Behavior Recognition Using Multiple Depth Cameras Based on a Time-Variant Skeleton Vector Projection

Chien-Hao Kuo, Pao-Chi Chang, and Shih-Wei Sun, *Member, IEEE*

***Abstract*—User behavior recognition in a smart office environment is a challenging research task. Wearable sensors can be used to recognize behaviors, but such sensors could go unworn, making the recognition task unreliable. Cameras are also used to recognize behaviors, but occlusions and unstable lighting conditions reduce such methods' recognition accuracy. To address these problems, we propose a time-variant skeleton vector projection scheme using multiple infrared-based depth cameras for behavior recognition. The contribution of this paper is threefold: 1) The proposed method can extract reliable projected skeleton vector features by compensating occluded data using nonoccluded data; 2) the proposed occlusion-based weighting element generation can be employed to train support-vector-machine-based classifiers to recognize behaviors in a multiple-view environment; and 3) the proposed method achieves superior behavior recognition accuracy and involves less computational complexity compared with other state-of-the-art methods for practical testing environments.**

***Index Terms*—Behavior recognition, skeleton, joint, multiple cameras, depth camera, Kinect.**

I. Introduction

BEHAVIOR recognition for human subjects moving in an indoor environment such as an office has become increasingly crucial in recent years. For example, Zenonos *et al.* [1] proposed the use of wearable sensors and smartphones to recognize human behaviors and thereby monitor the health of office workers, which could ultimately reduce the cost to government of work-related stress, anxiety, and depression. Once the behavior or mood of a human subject can be recognized in a smart office environment, a proper response from a server can be suggested to a user. For example, Sun *et al.* [2] proposed a smart living space that allows users to enter using an RF-ID card, recognizes facial expressions using a static camera, and gauges users' mood using a heart-rate sensor on their smart watch, as illustrated in Fig. 1. Once the expression and external behavior of a user can be recognized, corresponding audiovisual responses can be made accordingly. For example, when a user enters the smart living space for the first time, the system can automatically display a map and highlight the user's physical location using an Arduino control board within the Internet of Things (IoT) environment.

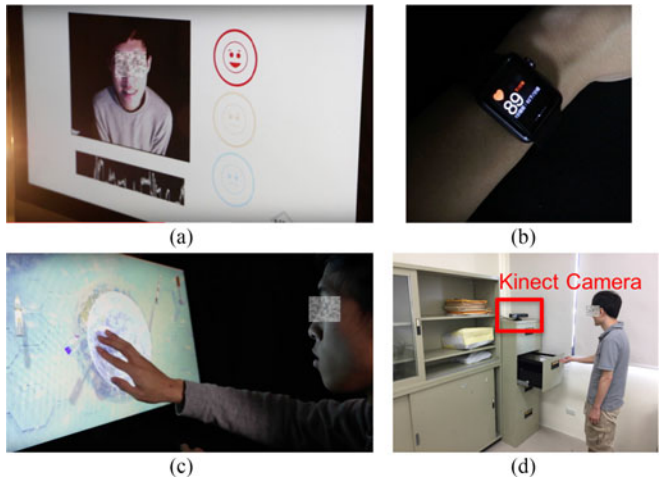


Fig. 1. Smart office scenario proposed by Sun *et al.* [2]: (a) facial expression recognition, (b) heart rate measurement using smart watch with built-in app, (c) audiovisual response from server, and (d) "open" gesture is recognized by the system and the drawer is opened by a servo motor controlled by the smart office server.

During the development of a smart living space [2] , cameras are one of the most critical sensors for human behavior recognition, and their use may obviate the necessity of users wearing or bringing sensor devices, thus leading to a natural and undisturbed user experience. Wearable sensors can precisely measure the internal state of a user, but once the smart watch or mobile device containing the sensors has been set aside by the user, they are unable to correctly recognize behaviors. Behavior recognition using camera devices also has drawbacks, with recognition becoming extremely challenging under changing lighting conditions or occlusions to the field of view. In this paper, we propose a time-variant skeleton vector projection scheme using multiple depth cameras for behavior recognition in a smart office environment; the proposed scheme avoids common recognition system drawbacks such as users forgetting to wear their wearable sensors and object occlusion within camera-based approaches. The contribution of the proposed method is threefold: 1) The proposed time-variant skeleton vector projection involving multiple cameras and a classifier training process can recognize human behaviors by using information from non-occluded

Manuscript received November 14, 2016; revised January 29, 2017; accepted February 21, 2017. Date of current version August 7, 2017. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 105-2221-E-119-001.

C.-H. Kuo and P.-C. Chang are with the Department of Communication Engineering, National Central University, Jong-Li 320, Taiwan (e-mail: chkuo@vaplab.ce.ncu.edu.tw; pcchang@ce.ncu.edu.tw).

S.-W. Sun is with the Department of New Media Art, Taipei National University of the Arts, Taipei 112, Taiwan (e-mail: swsun@newmedia.tnua.edu.tw).

Digital Object Identifier 10.1109/TETCI.2017.2674186

views to compensate for the occluded views; 2) the proposed occlusion-based weighting element generation process improves behavior recognition accuracy by analyzing the number of joints whose position is precisely known because no occlusion occurred; and 3) compared with state-of-the-art methods, the proposed method achieves higher recognition accuracy at a lower computational cost. The remainder of this paper is organized as follows. In Section II, related studies on behavior recognition are reviewed. The proposed method is described in detail in Section III, and the experimental results are presented in Section IV. Finally, Section V provides the conclusion.

II. Related Work

Existing methods for recognizing behavior in a smart office environment can be classified into four categories: wearable sensor, joint-based camera, partly camera-based, and multiple-camera approaches.

Wearable sensor techniques recognize behaviors by measuring the inertial signals of users. Zenonos *et al.* [1] used wearable sensors and smartphones to extract heart rate, acceleration, temperature, and pulse rate and train mood classification models to recognize the mood of users, gathering the information in a healthy office app that can improve an office's working efficiency. Sprint *et al.* [3] proposed the collection of sensor data from indoor environments for use in activity recognition by detecting behavior change. To recognize human activities, Shen *et al.* [4] proposed the use of smartphone motion sensors and Lee *et al.* [5] used a user's smartwatch and smart belt for behavior recognition. To understand a user's movement, Sun *et al.* [6] built a pressure sensing system using fiber-optic sensors mounted on the floor and a space encoding process, statistical modeling, and mixture learning. Human activities (walking, working, resting, and talking) can be recognized in an indoor office environment within an 8×8 block area. However, one of the mounted sensors can only respond with a binary result for a user's occupancy with limited spatial precision. Pham *et al.* [7] used an inertial measurement unit for motion data collection and distributed passive infrared sensors for human activity recognition, localization, and tracking. Tao *et al.* [8] employed wireless sensor networks from acceleration sensor signals for human behavior recognition. Tan and Yang [9] proposed the use of Wi-Fi signals to recognize user gestures. However, pressure sensors, fiber-optic sensors, passive infrared sensors, and Wi-Fi signals, as used in the aforementioned studies, only provide a rough estimation of a user's location and behavior recognition requires much higher spatial precision. Additionally, if a user forgets to wear their sensing device, any behavior recognition results will be incorrect.

Joint-based camera approaches utilize the geometrical relationships among the analytical joints of the body determined using motion capture (MoCap) or depth cameras. Lv and Nevatia [10] used the 3D positions of multiple joints as features and applied hidden Markov model (HMM) [11] weak classifiers in AdaBoost [12] to boost the classifiers in a single-MoCap-device environment. To utilize temporal information, Sheikh *et al.* [13] proposed the use of 4D space (with time as the fourth dimension) to recognize actions using joint angles in a MoCap environment. Employing the angular relationships among joint vectors, Hussein *et al.* [14] proposed a 3D joint covariance descriptor with the linear support vector machine (SVM) classifier for recognizing actions using a Kinect depth camera. By adopting MoCap and a Kinect depth camera, Wang *et al.* [15] extracted 3D joint features and used local occupancy patterns to generate spatial histograms for behavior recognition, using LIBSVM [16], a library for SVMs, to train the classifiers. By modeling the observed spherical coordinates of joints, Xia *et al.* [17] used linear discriminant analysis, vector quantization, and a discrete HMM to recognize behaviors. Yang and Tian [18] proposed the EigenJoints method based on a principal component analysis for behavior recognition. Furthermore, Zhu *et al.* [19] adopted multiple spatiotemporal features [20]–[23] and skeleton joint features [18] for feature quantization, and a random forest algorithm [24] was used for feature fusion and action classification in a Kinect camera environment.

Partly camera-based approaches use the relationships between skeleton data and multiple joint sets for behavior recognition. Chaudhry *et al.* [25] utilized connectivity relationships among multiple joint locations to generate shape context features [26] and optimal multiple kernel learning weights [27] to generate SVM classifiers in an environment with a single MoCap and Kinect depth camera. To observe how 3D joint locations change with time, Ofli *et al.* [28] proposed a method of measuring the Levenshtein distance [29] between joints, which can then be used to train classifiers. To analyze depth information around the joints, Ohn-Bar and Trivedi [30] proposed a histogram of oriented gradient-based [31] method for training classifiers. Using four joints for behavior recognition, Evangelidis *et al.* [32] proposed a skeletal quads method using a Gaussian mixture model and Fisher score [33] to generate feature vectors. For modeling temporal information for behavior recognition, Vemulapalli *et al.* [34] adopted Lie algebra for mapping joints in the spatiotemporal vector space and generating SVM classifiers. In all the aforementioned joint-based or partly camera-based approaches, however, partial and self-occlusion reduces behavior recognition accuracy.

Recently, multiple-camera environment approaches have begun to be used for behavior recognition. Azis *et al.* [35] used fused skeleton data from two cameras, and a nearest-neighbor dynamic time warping method was adopted for behavior recognition in a two-Kinect-camera environment. Furthermore, the same research group proposed a weighted averaging fusion algorithm [36] for generating a multiview skeleton including 3D joint positions, pairwise joint distances, and histogram of cubes [35], which can be used as features to generate behavior classifiers.

The method proposed in this paper avoids issues of inconvenience regarding wearable sensors and also overcomes problems regarding occlusions that occur in joint-based and partly camera-based approaches. The method proposed herein achieves behavior recognition using a multiple-camera environment, compensating for occluded views using the other camera's non-occluded view. The training data obtained from the multiple views are properly analyzed to achieve high behavior recognition accuracy with low computational complexity.

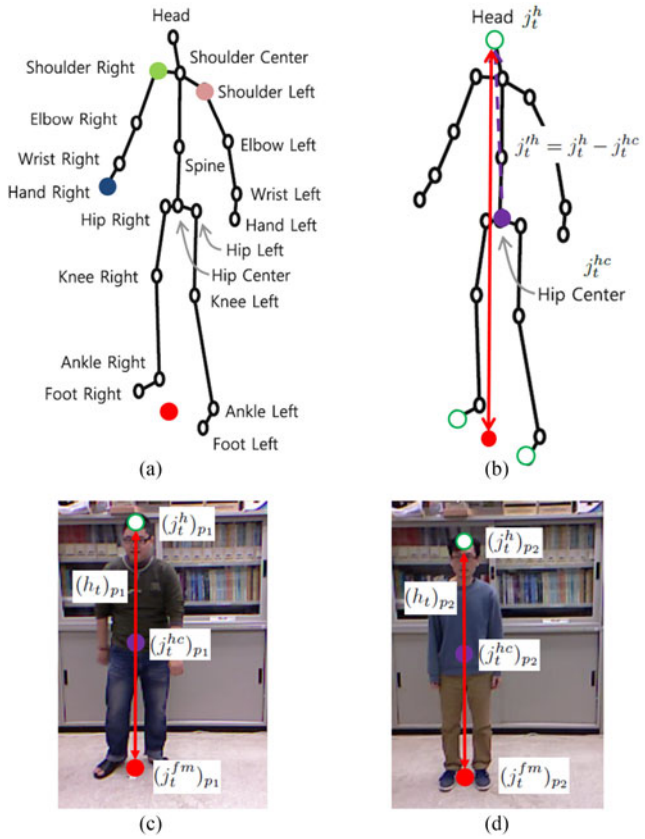


Fig. 2. Example joints and skeletons detected using Kinect SDK [37] and the corresponding measurements: (a) joints on human skeleton; (b) height measurement (red arrow) and the position of the head relative to the hip center joint (purple dashed arrow); (c) joints of p_1, (d) joints of p_2.

III. Time-Variant Skeleton Projection

Before behavior recognition can be attempted, skeleton joint data are obtained from the official Microsoft Kinect SDK 1.8 [37]. Each joint is represented by its real 3D position. For example, the top of the head $j_t^h = (x_t^h, y_t^h, z_t^h)$ (the top joint displayed in Fig. 2), obtained from the Kinect SDK, represents the physical position detected in the depth camera's field of view of the head point j_t^h at time t. When a behavior is performed by a user, the skeleton data are extracted from multiple views - for example, the center, right, and left views - which are recorded from consecutive frames, as illustrated by $Behavior$ 1 on the left-hand side of Fig. 3 (wherein joints and skeleton are obtained from the three views mentioned). The skeleton data are analyzed based on the proposed relative joint positions using a normalization process, basis vector generation, projection from a joint vector onto the basis vectors, and behavior classifier training, all of which are described in detail in the following sections and are represented by blocks in the central part of Fig. 3. In addition, the behavior features (right-hand side of Fig. 3) obtained from multiple views are further used for classifier training and behavior recognition.

A. Relative Joint Position With a Normalization Process

To deal with situations involving human subjects with different heights, which are inevitable in real applications of the method, a body height normalization process is necessary. As shown in Fig. 2(b), at time t, given the positions of the right foot joint $j_t^{fr} = (x_t^{fr}, y_t^{fr}, z_t^{fr})$ and left foot joint $j_t^{fl} = (x_t^{fl}, y_t^{fl}, z_t^{fl})$, a middle foot joint can be defined by taking their average: $j_t^{fm} = \frac{(j_t^{fr} + j_t^{fl})}{2}$. Furthermore, given the position of the head point $j_t^h = (x_t^h, y_t^h, z_t^h)$, the height of a subject can be measured by calculating the Euclidean distance from the head point j_t^h to the middle foot joint j_t^{fm}:

$$h_t = d\left(j_t^h, j_t^{fm}\right) = \sqrt{\left(x_t^h - x_t^{fm}\right)^2 + \left(y_t^h - y_t^{fm}\right)^2 + \left(z_t^h - z_t^{fm}\right)^2}, \quad (1)$$

which is depicted by the red line segment in Fig. 2(b). Taking the hip center joint j_t^{hc} [the purple circle in Fig. 2(b)] as the origin point of a relative coordinate system, the relative position of the head point normalized by the body height h_t:

$$j_t'^h = \frac{\left(j_t^h - j_t^{hc}\right)}{h_t} = \frac{\left(x_t^h - x_t^{hc}, y_t^h - y_t^{hc}, z_t^h - z_t^{hc}\right)}{h_t}, \quad (2)$$

can be calculated and is illustrated by the purple dashed line in Fig. 2(b). The relative positions of the rest of the joints can be similarly obtained. Hereafter, a prime symbol on a joint position is used to represent a normalized relative position; for example, $j_t'^h$ represents the relative position of the head point j_t^h with respect to the hip center j_t^{hc} [the purple dashed vector in Fig. 2(b)], normalized by h_t [the red line segment in Fig. 2(b)].

Even if users have different heights, as illustrated in Fig. 2(c) for the joints belonging to person p_1 and Fig. 2(d) for person p_2, the calculated $(j_t'^h)_{p_1} \approx (j_t'^h)_{p_2}$. Therefore, the proposed relative joint positions obtained through normalization can be treated as a user-invariant (regardless of users' height) feature.

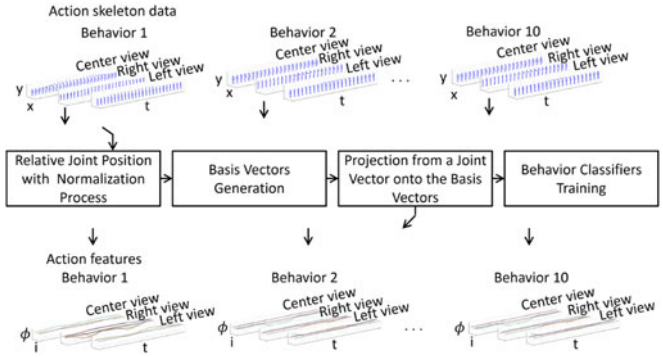


Fig. 3. Process for extracting features from skeleton data obtained from multiple-view depth cameras and the feature vector modeling process.

B. Basis Vectors Generation

Given the relative positions of the right shoulder joint $j_t'^{sr}$ and left shoulder joint $j_t'^{sl}$ [Fig. 2(a)], a shoulder vector $\overrightarrow{S_t} = \overrightarrow{j_t'^{sr} j_t'^{sl}}$ can be obtained, as illustrated by the green arrow in Fig. 4(a). Given the relative position of the foot middle joint $j_t'^{fm}$, a foot vector $\overrightarrow{F_t} = \overrightarrow{j_t'^{sr} j_t'^{fm}}$ can also be obtained

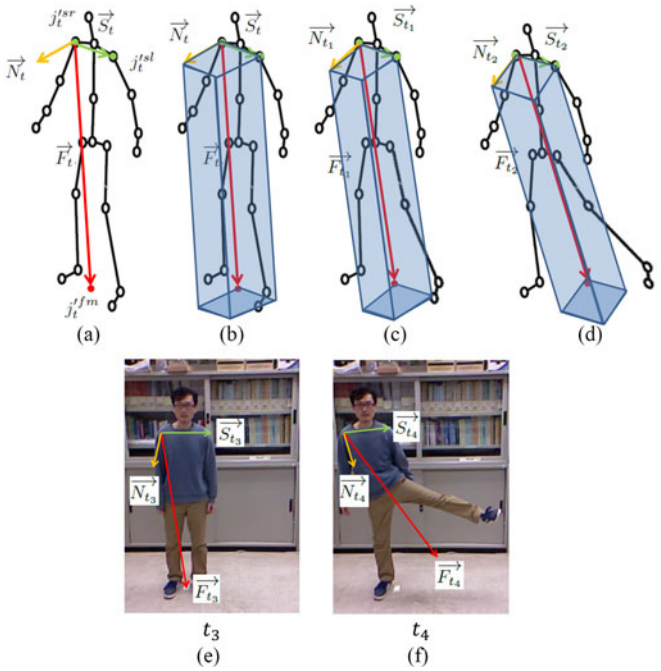


Fig. 4. Basis vector generation: (a) shoulder, foot, and normal vectors; (b) the obtained representative basis vectors; (c) the basis vectors at time t_1; (d) basis vectors at time t_2; (e) basis vectors in a real test at time t_3; and (f) basis vectors in a real test at time t_4.

[red arrow in Fig. 4(a)]. A normal vector $\overrightarrow{N_t}$:

$$\overrightarrow{N_t} = \overrightarrow{S_t} \times \overrightarrow{F_t}, \tag{3}$$

can subsequently be calculated from the outer product of $\overrightarrow{S_t}$ and $\overrightarrow{F_t}$. Notably, $\overrightarrow{N_t}$ is orthogonal to $\overrightarrow{S_t}$ and $\overrightarrow{F_t}$. Therefore, the vectors $\overrightarrow{N_t}$, $\overrightarrow{S_t}$, and $\overrightarrow{F_t}$ are treated as the basis vectors [Fig. 4(b)]. At time $t = t_1$, a set of basis vectors $\{\overrightarrow{N_{t_1}}, \overrightarrow{S_{t_1}}, \overrightarrow{F_{t_1}}\}$ is obtained [Fig. 4(c)], whereas at time $t = t_2$, another set of basis vectors $\{\overrightarrow{N_{t_2}}, \overrightarrow{S_{t_2}}, \overrightarrow{F_{t_2}}\}$ is obtained [Fig. 4(d)]. Thus, the obtained basis vectors are time dependent. For example, the basis vectors are different at times t_3 and t_4 because of the different postures assumed by the user. Hence, the proposed set of time-variant basis vectors – $\overrightarrow{N_t}$, $\overrightarrow{S_t}$, and $\overrightarrow{F_t}$ – can be applied for further feature description.

C. Projection of Joint Vector Onto the Basis Vectors

Given a right hand joint j_t^{ihr} and right shoulder joint j_t^{isr}, a hand joint vector $\overrightarrow{H_t} = \overrightarrow{j_t^{isr} j_t^{ihr}}$ can be obtained [blue arrow in Fig. 5(a)]. The geometric relationship between the hand joint vector $\overrightarrow{H_t}$ and basis vectors $\overrightarrow{N_t}$, $\overrightarrow{F_t}$, and $\overrightarrow{S_t}$ is depicted in Fig. 5(b). The projections of $\overrightarrow{H_t}$ onto the basis vectors $\overrightarrow{N_t}$, $\overrightarrow{F_t}$, and $\overrightarrow{S_t}$ are defined as $\|\overrightarrow{H_t}\| \cos\alpha$, $\|\overrightarrow{H_t}\| \cos\beta$, and $\|\overrightarrow{H_t}\| \cos\gamma$, respectively. The norms $\|\overrightarrow{N_t}\|$, $\|\overrightarrow{F_t}\|$, and $\|\overrightarrow{S_t}\|$ of the basis vectors can be substituted into these expressions to give $\|\overrightarrow{N_t}\|\|\overrightarrow{H_t}\| \cos\alpha$, $\|\overrightarrow{F_t}\|\|\overrightarrow{H_t}\| \cos\beta$, and $\|\overrightarrow{S_t}\|\|\overrightarrow{H_t}\| \cos\gamma$,

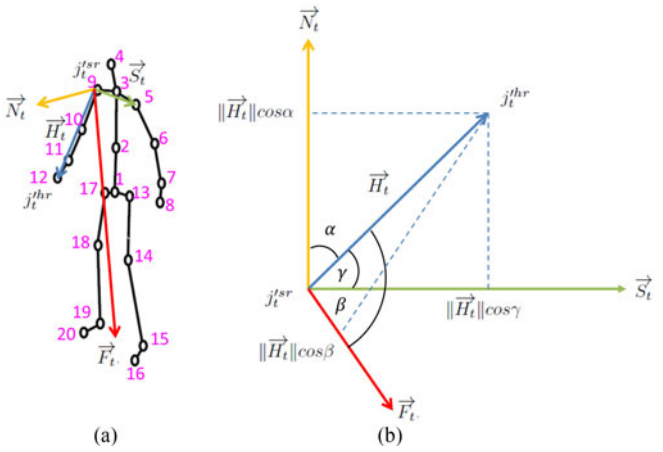


Fig. 5. Joint vector projections: (a) the indexes used for joints and the basis vectors; (b) the geometric relationships among the hand joint vectors and the basis vectors.

respectively. The derived expression

$$f_t^N = \|\overrightarrow{N_t}\|\|\overrightarrow{H_t}\| \cos\alpha = < \overrightarrow{N_t}, \overrightarrow{H_t} >, \tag{4}$$

$$f_t^F = \|\overrightarrow{F_t}\|\|\overrightarrow{H_t}\| \cos\beta = < \overrightarrow{F_t}, \overrightarrow{H_t} >, \tag{5}$$

$$f_t^S = \|\overrightarrow{S_t}\|\|\overrightarrow{H_t}\| \cos\gamma = < \overrightarrow{S_t}, \overrightarrow{H_t} > . \tag{6}$$

Therefore, for a given joint, the feature vector at time t is defined as

$$f_t = [f_t^N, f_t^F, f_t^S]. \tag{7}$$

The index i is used to represent the different joints of a human subject, and f_t^i is used to represent the feature vectors belonging to the ith joint. Thus, the complete set of features is represented as

$$\phi_t = \begin{bmatrix} f_t^{i=1,N} & f_t^{i=1,F} & f_t^{i=1,S} \\ f_t^{i=2,N} & f_t^{i=2,F} & f_t^{i=2,S} \\ \cdots & \cdots & \cdots \\ f_t^{i=20,N} & f_t^{i=20,F} & f_t^{i=20,S} \end{bmatrix}, \tag{8}$$

where the indexes $i = 1, \ldots 20$ are depicted in Fig. 5(a).

Over a specific period, the 3D position of each joint can be tracked. Fig. 6 illustrates the spatiotemporal domain information for different joints, wherein the z-axis (depth) is ignored for display convenience. The right hand joint (green circles, i = 12) is seen to move in large motions, but the right elbow (pink circles) and left ankle (red circles) move little. The 3D positions of the joints are defined by the proposed basis vector projections. An example of the complete set of features [(8)] is depicted in Fig. 7. The orange circles depicted at $t = 1$ are the projection vectors of all the joint vectors ($i = 1, \ldots i = 20$) onto the basis vector $\overrightarrow{N_{t=1}}$, which are defined as $f_{t=1}^{i=1,N} \ldots f_{t=1}^{i=20,N}$, [i.e., the first column vector of (8)]. Similarly, the red and green circles represent the projection vectors onto the basis vectors $\overrightarrow{F_{t=1}}$ and $\overrightarrow{S_{t=1}}$, and they are the second and third column vectors in (8), respectively.

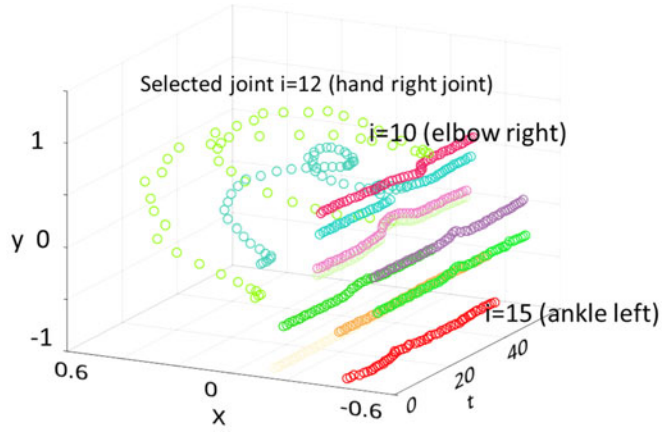


Fig. 6. Positions of the selected joints as detected by one of the multiple cameras in the spatiotemporal domain.

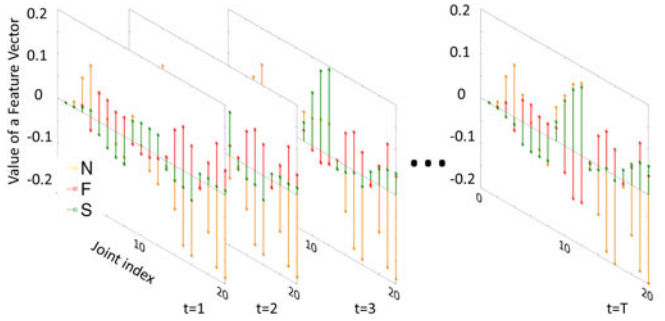


Fig. 7. Illustrative example of (8) for $t = 1, \ldots T$.

Given an obtained feature $\phi_{t,c}$ at camera c in a multiple-view environment, a feature set $\phi_{T,c} = [\phi_{t=1,c}, \phi_{t=2,c}, \cdots, \phi_{t=T,c}]$ can be obtained over a period T. Moreover, the features obtained from multiple views are concatenated, a cross-view feature can be obtained:

$$\Phi_T = [\phi_{T,c=1}, \phi_{T,c=2}, \cdots \phi_{T,c=C}], \tag{9}$$

where C is the number of cameras. The feature vectors found using (9) are depicted on the right-hand side of Fig. 3.

D. Behavior Classifier Training

Once the feature vectors from multiple views are obtained (examples of which are illustrated in Fig. 7), LIBSVM [16] is adopted for generating the classifiers for different behaviors in a multiple-view environment. For example (Fig. 8), the spatiotemporal features of $Behavior\ 1$ can be extracted from the multiple views of the depth cameras (in this example, the center, right, and left views), and the salient spatiotemporal features can then be selected from the pool of trajectories (set of curves in Fig. 9). The variance of each trajectory along the time axis is calculated and trajectories with variances larger than a threshold v_T (such as the bolder curves in Fig. 9) are kept as the salient spatiotemporal trajectories. Notably, the trajectories presented in Fig. 9 operate in the feature domain [i.e., (8)], not in the physical spatiotemporal domain.

Next, the adaptive weighting factors are calculated according to the occlusion situation analyzed from multiple views. Fig. 10 illustrates an example gesture (raised right hand) that can be observed from views $c1$ and $c2$ but not $c3$, in which it is occluded by the user's body. In the behavior classifier training, therefore, the views in which occlusion does and does not occur have different weights. In this paper, we propose an occlusion-based weighting element generation process that assesses the

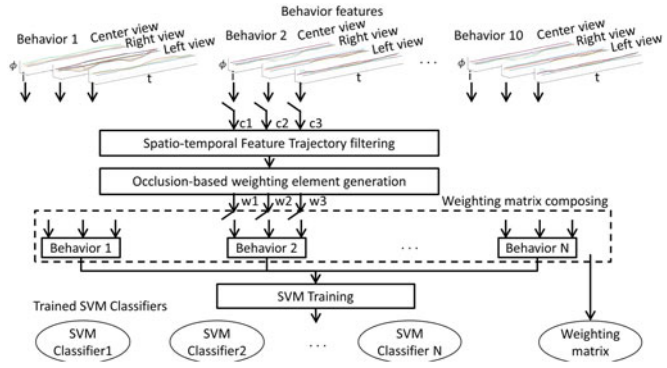


Fig. 8. Training of behavior classifiers using the features extracted from multiple views.

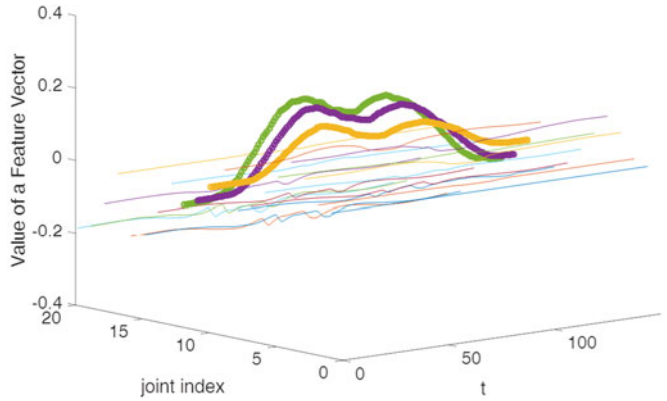


Fig. 9. Proposed trajectory filtering process, with plots of 20 joints projected on one of the basis vectors in the spatiotemporal feature domain; the three filtered feature trajectories are presented in bold.

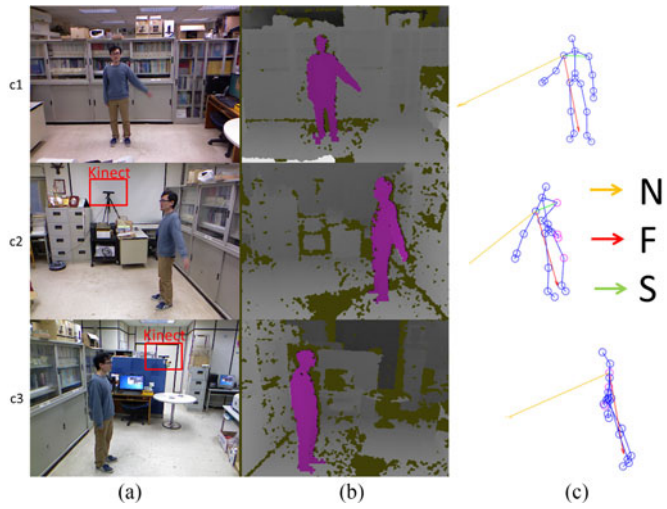


Fig. 10. Captured depth camera data: (a) color frames; (b) depth frames with subjects detected; and (c) the resultant skeletons and basis vectors.

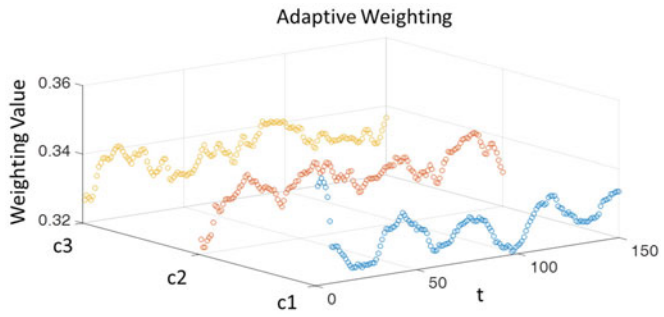


Fig. 11. Example of the proposed occlusion-based weighting element generation.

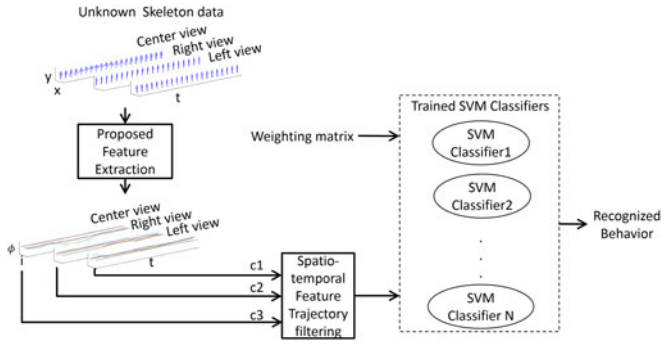


Fig. 12. Process of behavior recognition from unknown skeleton data extracted from multiple depth cameras.

occlusion situation and generates suitable SVM-based behavior classifier (middle of Fig. 8).

1) Occlusion-Based Weighting Element Generation: During the detection of skeletons and joints with multiple depth cameras in practice, joints may be occluded by other parts of the user's body when a behavior is being performed. In Kinect SDK, the detected joints are classified into three categories: "tracked," indicating a tracked joint that the camera can see; "inferred," indicating a tracked joint that the camera cannot see but the position of which can be estimated by surrounding joint data; and "not tracked," indicating the lack of joint data. The "inferred" and "not tracked" joints are possibly occluded (by the user or other objects) in the field of view. In Kinect SDK1.8 [37], 20 joints are detected in each frame, and more "inferred" and "not tracked" joints indicate that there are less "tracked" joints. Thus, the reliability of a representation is proportional to the number of "tracked" joints $\tau_{t,c}$ at camera c. Views with less occlusion (higher $\tau_{t,c}$) should have a greater effect on the feature generation process and the following behavior recognition process. Therefore, through the consideration of the occlusion effects for different views at different time points, a time-variant weighting function can be independently defined in terms of the feature vector:

$$\Phi_T^w = [w_{c=1} \cdot \phi_{T,c=1}, w_{c=2} \cdot \phi_{T,c=2}, \cdots w_{c=C} \cdot \phi_{T,c=C}]. \tag{10}$$

Equation (10) reveals the effect of occlusions at different time points. By considering the computational complexity and accuracy in the feature generation and recognition processes, we propose the designation of the weighting elements as follows:

$$w_c = \frac{\tau_{t_c}}{\sum_{c=1}^{C} \tau_{t_c}}, \tag{11}$$

where $\tau_{t,c=1}$ is the number of "tracked" joints in each view. A three-view example of the proposed weighting elements is illustrated in Fig. 11. At different time instances, the weights of the multiple cameras (i.e., $c1$, $c2$, and $c3$,) are determined by the number of "tracked" joints in the three views.

E. Behavior Recognition

LIBSVM [16] is adopted for classifier training and testing for behavior recognition (Fig. 8, bottom; Fig. 12, right-hand side). The proposed feature extraction process using multiple views (Fig. 12, upper-left) and the feature trajectory filtering process (e.g., Fig. 9) are operated for unknown skeleton data. The calculated weight matrix (e.g., Fig. 11) is sent as side information together with the SVM classifiers (Fig. 12, right-hand side) so that the behaviors of a user can be recognized.

IV. Experimental Results

In the experiment, three Kinect v1 depth cameras were used to capture a user's behavior with the official Kinect SDK 1.8 [37], and this served as the raw skeleton data. The Kinect cameras (two marked by red rectangles in Fig. 10) were mounted at a height of approximately 1.6 m from the floor. One of the depth cameras was positioned to capture the front view, and the others were mounted to capture the two side views. Color frames, depth frames wherein the subject has been detected, and skeletons (blue line segments) with joints (blue circles) are presented in Fig. 10(a)–(c), respectively. Through the proposed method, the three basis vectors $\overrightarrow{N_t}$, $\overrightarrow{F_t}$ and $\overrightarrow{S_t}$ were calculated using (3) and are depicted by the orange, red, and green arrows in Fig. 10(c). Furthermore, the calculated vector projections [through (4), (5), and (6)] for the three views over a specific period are plotted in Fig. 13(a) and (b) for $Person$ 1 and $Person$ 2, respectively. These two sets of vector projections have similar distributions in the environment with multiple depth cameras. Therefore, the obtained feature vectors were used to train the classifiers for behavior recognition. The trajectory threshold, defined in Section III-D, was empirically determined as $v_T = 0.02$, and this value was used in all subsequent tests.

Ten behaviors were recorded in the environment for evaluation, with a total of 15,169 frames recorded in 505.63 s, and 10 volunteer users were asked to perform the behaviors (examples in Fig. 14). The 10 users performed each behavior three times; therefore, $10 \times 30 \times 3 = 300$ video clips were generated, with a manual time synchronization process used. Half of the uniformly sampled video clips were used as the training dataset for behavior recognition, and the other half were used as the testing dataset. The proposed method was compared with other state-of-the-art methods using the multiple-view test video sequences provided by Azis *et al.* [36] . The performance level of the proposed method was compared with those of the methods proposed by Azis *et al.* [36] (multiple-view approach) and Vemulapalli *et al.* [34] (partly camera-based single-view approach).

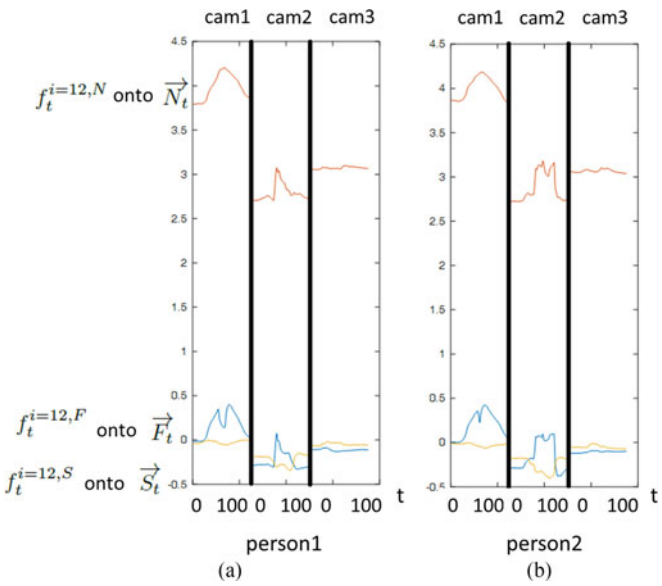


Fig. 13. Feature vectors among three views for the same behavior as performed by (a) person 1 and (b) person 2.

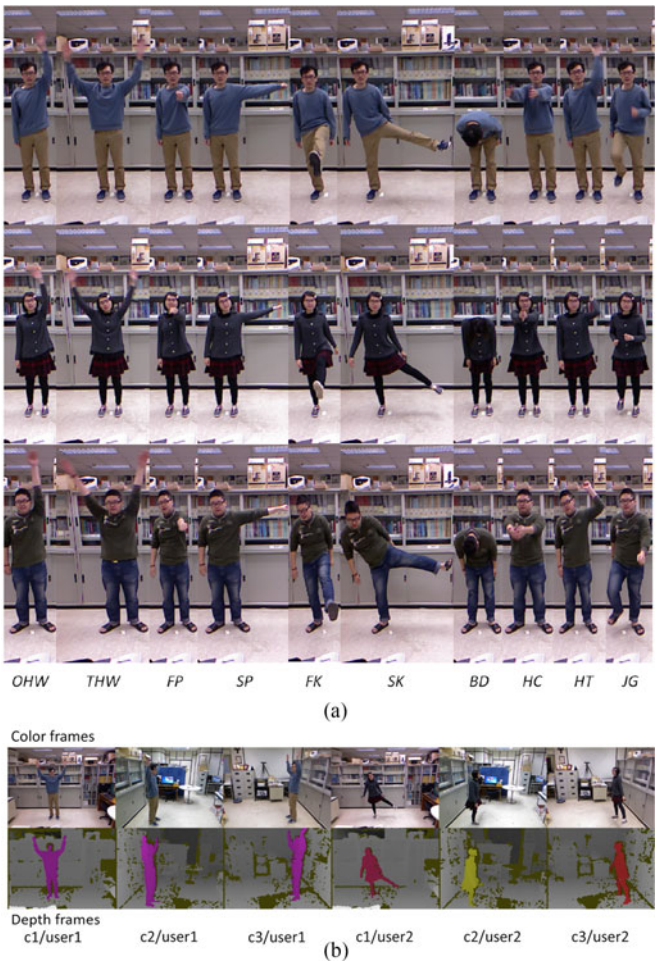


Fig. 14. Part of the behavior database recorded with volunteer users. Behaviors included were the one hand wave (OHW), two hand wave (THW), forward punch (FP), side punch (SP), forward kick (FK), side kick (SK), bend (BD), hand clap (HC), high throw (HT), and jog (JG). (a) Ten different behaviors behaved by three volunteers. (b) Color and depth frames from the three views.

TABLE I
COMPARISON RESULTS OF MEAN AVERAGE PRECISION (MAP)

	proposed dataset	[36] dataset
Vemulapalli's method [34], center view	$\mathbf{100 \pm 0.00}\%$	$91.67 \pm 10.94\%$
Vemulapalli's method [34], right view	$93.33 \pm 14.74\%$	–
Vemulapalli's method [34], left view	$87.25 \pm 14.56\%$	–
Vemulapalli's method [34], side view	–	$87.12 \pm 17.63\%$
Azis's method [36]	$86.29 \pm 9.34\%$	$85.60 \pm 13.12\%$
Proposed Method	$99.33 \pm 2.11\%$	$\mathbf{91.68 \pm 19.65}\%$

A. Quantitative Evaluation

The mean average precision (mAP) results associated with the proposed method (multiple-view) and the methods from [36] (multiple-view) and [34] (partly camera-based single-view) are compared in Table I. The single-view method [34] was applied to the frames obtained from each view. Because the center view is less frequently occluded, the single-view [34] results derived for the center view were more accurate than the side-view results. As revealed by the second and third rows of Table I, occlusions that occurred in the side views resulted in decreased accuracies of 93.33% (right view) and 87.25% (left view) for the method of [34], whereas the proposed method achieved the best mAP of 99.33% when all views were considered.

Nevertheless, a center view cannot always be obtained in practical applications. Of the two multiple-view approaches, the proposed method had superior overall performance to the method of Axis *et al.* [36] for the proposed dataset. For the more challenging dataset provided by [36] (Table I), the proposed method also had the best behavior recognition capability when compared with the two other methods. The detailed behavior recognition results for different behaviors are presented through confusion matrices in Fig. 15. The label of each row in the confusion matrices is the actual behavior label (i.e., the behavior that was performed), and the labels at the foot of each column are the predicted behavior labels. The corresponding element of the confusion matrix is the number of times the predicted label was identified by the method divided by the total number of times the actual behavior was performed in the dataset (such that the sum of the elements in each row must equal 1). The last row of Fig. 15(f) corresponds to the "side kick," which was incorrectly recognized as other behaviors because only the front view and one side view could be obtained in the dataset of [36]. Similar performance degradation was observed when the method of Axis *et al.* [36] (multiple-view approach) was used, as shown by the last row of Fig. 15(d). Therefore, if information from less-occluded views could be obtained, the behavior recognition capability of the proposed method would be improved because the occluded data could be compensated.

B. Qualitative Evaluation

Fig. 16(a) and (c) present depth frames, wherein the subject has been detected (the first three rows), and skeletons (blue line segments in the subsequent three rows) obtained using Kinect SDK; from these, the basis vectors $\overrightarrow{N_t}$ (orange arrows), $\overrightarrow{F_t}$

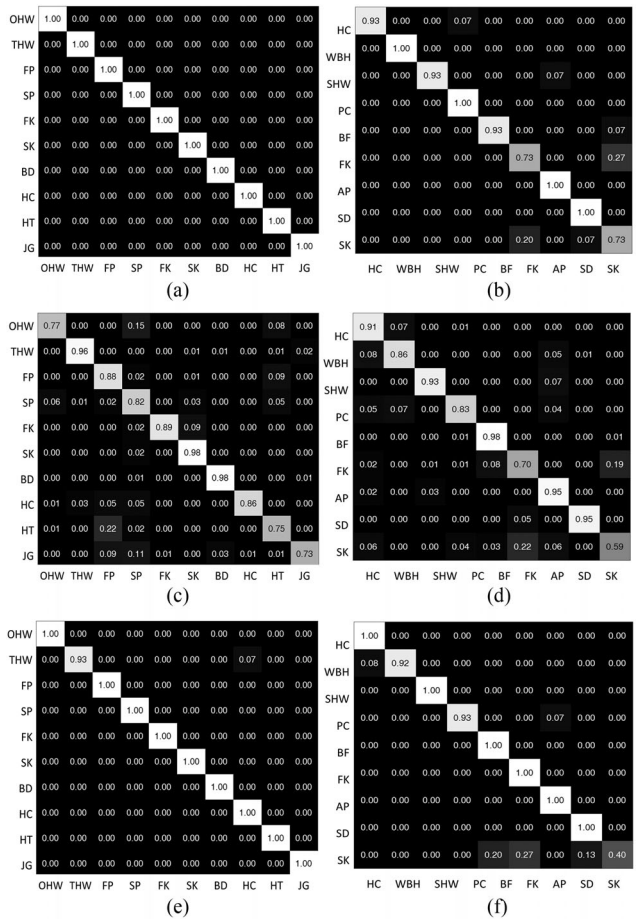


Fig. 15. Confusion matrix for the behavior recognition accuracy of Vemulapalli's method [34] (single-view), Azis's method [36] (multiple-view), and the proposed method (multiple-view). The behavior labels are as listed in the caption of Fig. 14, and the abbreviations used in the [36] dataset are: hand clapping (HC), waving both hands (WBH), single hand waving (SHW), punching (PC), bend forward (BF), forward kick (FK), answering a phone call (AP), sitting down (SD), and side kick (SK). (a) Vemulapalli's method [34] from the center view; (b) Vemulapalli's method [34] with the front view of the [36] dataset; (c) Azis's method [36] using three views; (d) Azis's method [36] using two views of the [36] dataset; (e) proposed method using three views; (f) tproposed method using two views of the [36] dataset.

(red arrows), and $\overrightarrow{S_t}$ (green arrows) could be obtained using the proposed method. For example, the movement of a foot could be identified from depth frames 1515–1520 in view $c1$. However, the foot is occluded by the subject in view $c2$. The foot joints marked by the green dashed circles in view $c1$ in Fig. 16(b) could be used to reliably determine $\overrightarrow{F_t}$ (red arrow) in view $c1$ according to the definition given in Section III-B. Nevertheless, because of the occlusion of the foot in views $c2$ and $c3$, the obtained $\overrightarrow{F_t}$ (red arrows) could be erroneous (with the magenta circles being the "inferred" joints). In the proposed method, the weights of the joints reduce the effect of the "inferred" joints. Thus, view $c1$ could have had higher weighting and relative angles, as illustrated by the skeleton identified in view $c1$ in Fig. 16(a).

Similarly, the raised hand in frame 2341 in Fig. 16(d) is shown to be more stably tracked in view $c1$ but severely occluded in

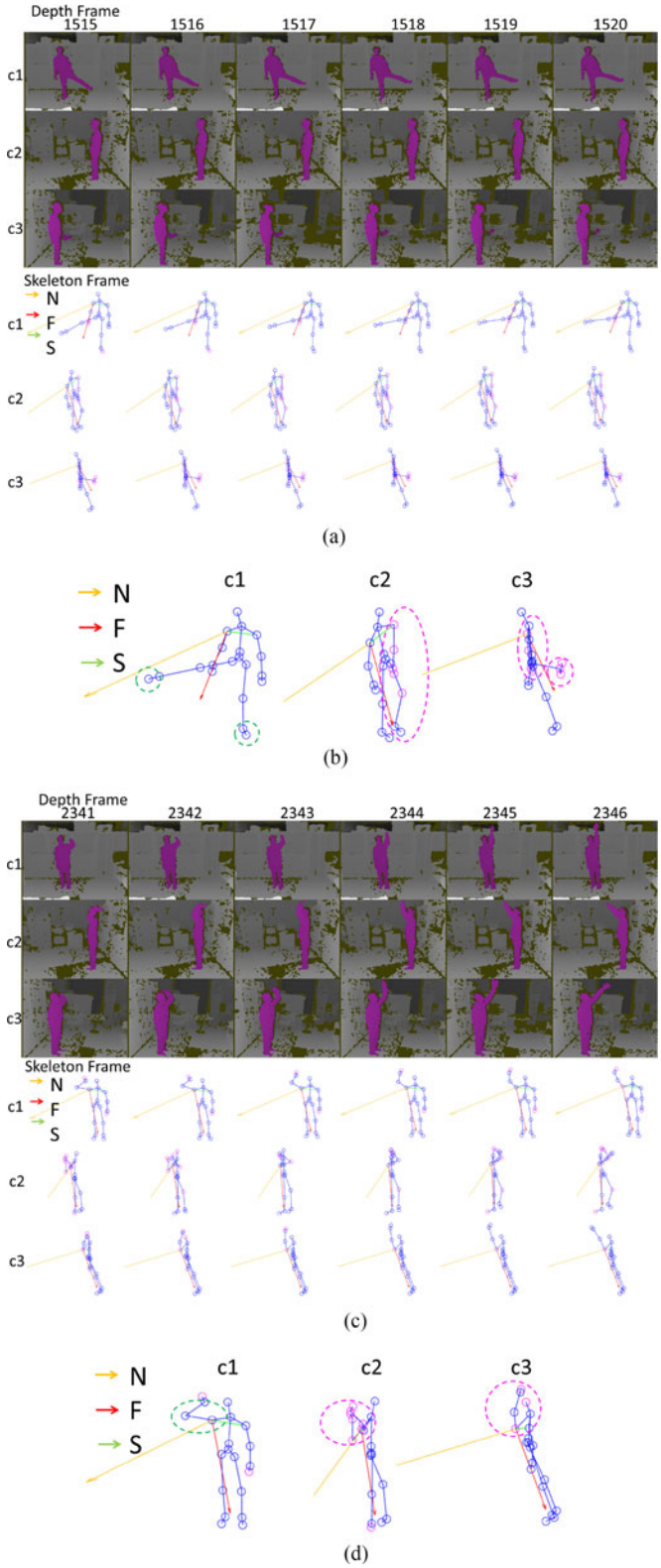


Fig. 16. Time-variant skeleton vector projection for a foot/hand movement behavior. (a) Depth frames 15151520 (foot movement) from views c1, c2, and c3, and the corresponding skeletons; (b) skeletons obtained from frame 1517: blue and magenta circles are the tracked and inferred joints, respectively. (c) Depth frames 23412346 (hand movement) from views c1, c2, and c3, and the corresponding skeletons.

TABLE II
TIME COMPLEXITY WHEN VEMULAPALLI'S METHOD [34] (SINGLE-VIEW), AZIS'S METHOD [36] (MULTIPLE-VIEW), AND THE PROPOSED METHOD (MULTIPLE-VIEW) WERE USED

time (s)	proposed dataset		[36] dataset	
	Modeling	Classification	Modeling	Classification
[34]	1673.79	946.63	1727.71	1178.23
[36]	142.40	143.55	139.18	152.94
Proposed	**59.48**	**38.60**	**21.18**	**7.46**

TABLE III
RESULTS OF THE PROPOSED METHOD WITH DIFFERENT NUMBERS OF VIEWS USED FOR THE PROPOSED DATASET; MEAN AVERAGE PRECISION (MAP), MODEL TIME (MT), AND CLASSIFICATION TIME (CT)

used views	mAP	MT (s)	CT (s)
center + right + left (3 views)	$\mathbf{99.33 \pm 2.11\%}$	**59.48**	**38.60**
center + right (2 views)	$99.33 \pm 2.11\%$	42.63	27.86
center + left (2 views)	$96.00 \pm 7.17\%$	42.98	27.97
left + right (2 views)	$90.00 \pm 10.06\%$	42.60	27.78
center (1 view)	$98.67 \pm 2.81\%$	18.79	13.21
right (1 view)	$85.33 \pm 12.09\%$	20.29	13.80
left (1 view)	$78.00 \pm 16.35\%$	20.87	13.68

views $c2$ and $c3$. The basis vector $\overrightarrow{S_t}$ is correct in view $c1$ but erroneously detected in view $c2$ because of occlusion. In view $c2$, the erroneous basis vector $\overrightarrow{N_t}$ (orange arrow) is due to the erroneous $\overrightarrow{S_t}$ (green arrow). Similar results can be observed in view $c2$ for frames 2341–2346. However, the proposed method could reduce the effect of the erroneous basis vectors because view $c1$ had a larger weighting factor.

C. Complexity Comparison

The proposed method (multiple-view), the method of Axis *et al.* [36] (multiple-view), and the method of Vemulapalli *et al.* [34] (single-view) were executed on the same computer, which had an Intel Core i7, a 2.67-GHz CPU, and 12Gb of RAM. The total computational time required for each method with each dataset is presented in Table II. The previously proposed methods in [36] and [34] involve dynamic time warping operations but the proposed method does not and thus requires the least computational time for its modeling and classification operations. The proposed method only needs to project the obtained skeleton vector onto the proposed basis vectors, after which the weighting factors are determined by any occlusion detected by the Kinect SDK, which occurs in real time. Therefore, the proposed method has the potential to be applied in real-time applications.

In the proposed method, the number of depth cameras used for the behavior recognition can affect the overall recognition accuracy. As revealed in the first row of Table III, the mAP was 99.33% when all three views were used in the behavior recognition process. When only two views were used, the mAP was reduced to 90.00%–99.33%, and the modeling and classification times were also reduced. Furthermore, when only one view was used, the mAP was further reduced to 78.00%–98.67%, with further savings in modeling and classification times. However, the mAP results were highly correlated with the view selected. In our tests, the left view included severe occlusions; therefore, when the left view was selected for behavior recognition using a single view or two views, the mAP was reduced. Nevertheless, occlusions can be considered by determining the weighting factors proposed in (10) in the proposed method. The detailed mAP results in Table III are also illustrated by the confusion matrices in Fig. 17.

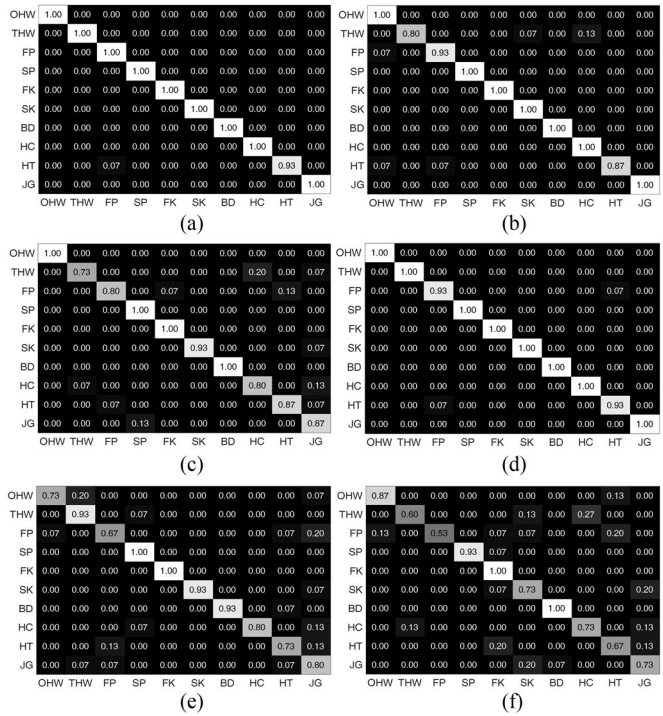


Fig. 17. Confusion matrix for the behavior recognition accuracy of the proposed method when two views and a single view were used. (a) center and right views used; (b) center and left views used; (c) left and right views used; (d) center view used; (e) right view used; (f) left view used.

V. CONCLUSION

In conclusion, the proposed behavior recognition scheme in a multiple-depth-camera environment recognizes behaviors with higher accuracy and lower computational complexity than the state-of-the-art methods of [34] and [36]. The contribution of this paper is thus threefold: 1) The proposed time-variant skeleton projection using multiple views can compensate for occluded views and identify features reliably; 2) the proposed SVM-based classifier accurately recognizes behaviors; and 3) the execution of the proposed method has low computational complexity compared with the other methods. As revealed in Table III, the addition of more cameras increases the accuracy of the method, but using fewer cameras saves computational power, which could make the method suitable for real-time applications. Therefore, the proposed method can be adopted in the future for smart office applications, recognizing user behaviors and then triggering corresponding IoT-based applications.

As illustrated by the example in Fig. 1(d), a smart office may have different IoT-connected devices such as lights or a servo motor controlled by an Arduino controller. Once a user's forward punch gesture (which does not involve physical touch) is recognized in a multiple-depth-camera environment according to the method proposed in this paper, the server can automatically turn on the light using a switch and trigger the servo motor to open a drawer, all controlled by an Arduino controller. In the future, the proposed behavior recognition scheme using multiple depth cameras can be extended to multiple skeleton-based devices such as the Kinect v2 and Leap Motion and motion capture devices.

REFERENCES

[1] A. Zenonos, A. Khan, G. Kalogridis, S. Vatsikas, T. Lewis, and M. Sooriyabandara, "Healthyoffice: Mood recognition at work using smartphones and wearable sensors," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2016, pp. 1–6.

[2] "Moving to 2026," the Innovation, Creation, and Entrepreneurial Contest, Ministry of Education, Taiwan, (the Best Innovation Award, the Best Technology Integration Award, Advisor: Prof. S.W. Sun) 2016. [Online]. Available: https://youtu.be/YIjtJJ8yXVA

[3] G. Sprint, D. Cook, R. Fritz, and M. Schmitter-Edgecombe, "Detecting health and behavior change by analyzing smart home sensor data," in *Proc. IEEE Int. Conf. Smart Comput.*, May 2016, pp. 1–3.

[4] C. Shen, Y. Chen, and G. Yang, "On motion-sensor behavior analysis for human-activity recognition via smartphones," in *Proc. IEEE Int. Conf. Identity, Security Behavior Anal.*, Feb. 2016, pp. 1–6.

[5] J. G. Lee, M. S. Kim, T. M. Hwang, and S. J. Kang, "A mobile robot which can follow and lead human by detecting user location and behavior with wearable devices," in *Proc. IEEE Int. Conf. Consum. Electron.*, Jan. 2016, pp. 209–210.

[6] Q. Sun, F. Hu, and Q. Hao, "Human movement modeling and activity perception based on fiber-optic sensing system," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 6, pp. 743–754, Dec. 2014.

[7] M. Pham, D. Yang, W. Sheng, and M. Liu, "Human localization and tracking using distributed motion sensors and an inertial measurement unit," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2015, pp. 2127–2132.

[8] D. Tao, L. Jin, Y. Wang, and X. Li, "Rank preserving discriminant analysis for human behavior recognition on wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 813–823, Feb. 2014.

[9] S. Tan and J. Yang, "Fine-grained gesture recognition using wifi," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2016, pp. 257–258.

[10] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using HMM and multi-class adaboost," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 359–372.

[11] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[12] Y. Freund and R. Schapire, "A decision theoretic generalization of on-line learning and application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1995.

[13] Y. Sheikh, M. Sheikh, and M. Shah, "Exploring the space of a human action," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, vol. 1, pp. 144–149.

[14] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2466–2472.

[15] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1290–1297.

[16] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[17] L. Xia, C. C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 20–27.

[18] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 14–19.

[19] Y. Zhu, W. Chen, and G. Guo, "Fusing spatiotemporal features and joints for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 486–491.

[20] I. Laptev and T. Lindeberg, "Velocity adaptation of space-time interest points," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2004, vol. 1, pp. 52–56.

[21] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis., Part II*, 2008, pp. 650–663.

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

[23] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 275:1–10.

[24] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[25] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 471–478.

[26] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[27] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, 2008.

[28] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 8–13.

[29] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Phys. Doklady*, vol. 10, 1966, Art. no. 707.

[30] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and hog2 for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 465–470.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, pp. 886–893.

[32] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 4513–4518.

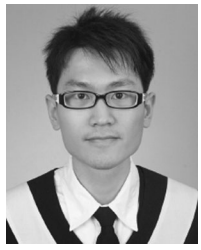
[33] T. Jaakola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. Conf. Adv. Neural Inf. Process. Syst. II*, 1999, pp. 487–493.

[34] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[35] N. A. Azis, H. J. Choi, and Y. Iraqi, "Substitutive skeleton fusion for human action recognition," in *Proc. Int. Conf. Big Data Smart Comput.*, Feb. 2015, pp. 170–177.

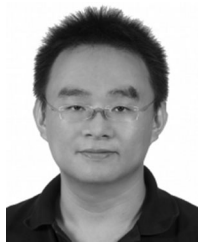
[36] N. A. Azis, Y. S. Jeong, H. J. Choi, and Y. Iraqi, "Weighted averaging fusion for multi-view skeletal data and its application in action recognition," *IET Comput. Vis.*, vol. 10, no. 2, pp. 134–142, 2016.

[37] "Kinect for windows sdk 1.8," 2013. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=40278



Chien-Hao Kuo received the B.S. degree in communication engineering from National Central University (NCU), Taiwan, in 2009. He is currently working toward the Ph.D. degree in the Video-Audio Processing Laboratory, Department of Communication Engineering, NCU, Taiwan. His research interests include video/image processing and video compression.

Pao-Chi Chang received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, and the Ph.D. degree from Stanford University, Stanford, CA, USA, 1986, all in electrical engineering. From 1986 to 1993, he was in research staff at IBM T.J. Watson Research Center, New York, NY, USA. In 1993, he joined the faculty of NCU, Taiwan, where he is presently a Professor in the Department of Communication Engineering. His main research interests include speech/audio coding, video/image compression, and multimedia retrieval.



Shih-Wei Sun (M'12) received the B.S. degree from Yuan-Ze University and the Ph.D. degree from National Central University, Taiwan, in 2001 and 2007, respectively, both in electrical engineering. From 2007 to 2011, he was a Post Doctoral Research Fellow in the institute of information science, Academia Sinica. In 2012, he joined the Department of New Media Art, Taipei National University of the Arts, Taiwan, as an Assistant Professor. Since 2016, he is an Associate Professor. He is the founding leader with the Ultra-Communication Vision Laboratory (ucVision Lab). His research interest includes visual content analysis, computer vision and the application for interactive technologies, and sensor applications for mobile devices. He received the Research Award from the Prize Award of Multimedia Grand Challenge from the 2014 ACM Multimedia Conference. He published more than 40 international journal papers and conference papers. He serves as the Reviewers and technical program committee Members for many journals and conferences.