

Received November 13, 2019, accepted December 19, 2019, date of publication December 30, 2019, date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962843

Coarse-to-Fine Activity Annotation and Recognition Algorithm for Solitary Older Adults

XIN HU¹, ZHENGZUO LI¹, RUIXUAN DAI², YANG CUI³, ZHIYUAN ZHOU¹,
BOYANG AN¹, YONGQING HAN⁴, CHUNMAO JIANG⁵,
DEQIONG DING⁶, AND DIANHUI CHU¹

¹Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, China

²Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA

³Department of Engineering and IT, The University of Melbourne, Melbourne, VIC 3051, Australia

⁴Office of Retirement Services, Harbin Institute of Technology at Weihai, Weihai 264209, China

⁵School of Computer Science and Technology, Harbin Normal University, Harbin 150001, China

⁶Department of Mathematics, Harbin Institute of Technology at Weihai, Weihai 264209, China

Corresponding authors: Xin Hu (hithuxin@hit.edu.cn), Deqiong Ding (mathddq@hit.edu.cn), and Dianhui Chu (chudh@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (General Program) under Grant 61772159, and in part by the Natural Science Foundation of Shandong Province (in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript) under Grant ZR2017MF026.

ABSTRACT Older adults want to remain independent with dignity for as long as possible, especially the solitary older adults. Activity recognition plays an essential role in elderly care and rehabilitation by detecting activity shifts among the elderly population. Despite over a decade of research and development in activity recognition, accurate and reliable systems for older adults in use are few. We propose an automatic data collecting and labeling system by addressing the annotation issue, and a novel coarse-to-fine activities of daily living (ADLs) recognition algorithm for older adults, by combining supervised and unsupervised machine learning methods. The automatic data collecting and labeling system targets at the annotation issue caused by the diversity of ADLs in free-living situations. Multiple sensors fusion strategy is employed to interpret and annotate the ADLs. Leveraging supervised and unsupervised machine learning methods, we can discover and recognize ambulatory and trivial ADLs for older adults. The performance of the automatic data collecting and labeling system is double-checked in a four days long test. With the reliable ground truth, we evaluate the coarse-to-fine ADLs recognition algorithm. The performance of our algorithm is promising, the recognition accuracy is larger than 91%.

INDEX TERMS Automatic labeling, coarse-to-fine ADLs recognition, multiple sensors fusion strategy, solitary older adults.

I. INTRODUCTION

As the aging process of world's population is accelerating, the issue of population aging is an unavoidable social problem. Significant problems are reported in elderly care and rehabilitation, associated with the waste of public medical resources. Older adults, regardless of nationality, want to remain independent with dignity for as long as possible, especially the solitary older adults. Activity recognition plays an essential role in this area by detecting activity shifts

The associate editor coordinating the review of this manuscript and approving it for publication was Honggang Wang.

among the elderly population, which can alleviate stress on limited medical resources, and help the older adults to maintain functional ability and live independently longer, by detecting and diagnosing early illnesses for early warning [1]. Based on device diversity and sensor modality, previous researches in activity recognition can be roughly categorized into 4 classes: video-based, radar-based, WiFi signal-based, and inertial sensor-based. Several good reviews can be found in [1]–[6].

In the past decades, a large number of video datasets and benchmarks for activity recognition have been released, and various video-based activity recognition algorithms emerge

one after another. Thus, the video-based methods are more realistic and appropriate to recognize human postures and activities, comparing with the other three classes. The video data can be accurately labeled with the data itself manually. Methods of the other three classes may still need video or human journal for labeling. It is natural and intuitive to recognize human postures and activities from video data.

However, most existing video datasets are captured from surveillance cameras, which always suffer from privacy issue [2]. The privacy issue could be attended to with depth video and infrared thermal video, which are also more resistant to the illumination changes and the pool imaging light, comparing with the visible video (RGB or gray). Therefore, depth video and infrared video for action recognition have gained much attention. However, there are still several challenges remain unsolved in the video-based activity recognition area, which are occlusion, annotation, visibility range, and illumination variation. These challenges prevent the video-based activity recognition methods from wildly use, especially in the out-door situations.

Recent years, radar has been employed to detect, recognize, and understand human activities, because of the effective sensing capability and penetration through obstacles [5]. Passive radar techniques for indoor human activity recognition have been extensively investigated [5], [7]–[10]. Most of them show high precision performance (more than 85% accuracy), in the non-line-of-sight environment, which is the typical situation of activity recognition for older adults. To ensure the high accuracy, ambient radar solution is essential for the radar-based methods, which requires a set of ambient radar sensors. The number of sensors increases with the environment size, while the cost explodes with it. Moreover, the complexity of multiple radar system also roars with the number of radar, making it too sophisticated to handle. As a result, trained professionals are need to interpret the radar signals, which make the radar-based methods hard to be implemented [4].

Nowadays, researchers employ wireless remote sensing technologies with commercial off-the-shelf wireless fidelity (WiFi) devices, to perceive and identify activities [4]. Instead of special and professional devices, common WiFi devices are used in a passive manner for activity detecting, which means cheap, universal, convenient, and unobtrusive. WiFi signals propagate indoors and carry rich human body information, which can be modeled for human tracking and activity identifying. Multi-path fading rule is employed to extract activity features from the channel state and the frequency modulated wave [6]. Although the satisfactory accuracy and robustness are guaranteed for the WiFi signal-based methods, complex environment limits them from application in real-world. Device placement scheme, coverage, multi-target, and through-wall attenuation are all attached to the challenges of the WiFi signal-based methods [4].

Thanks to the ubiquity of smart devices, such as mobile phones, wrist bands, and smart watches, using inertial measurement unit (IMU) sensors embedded in the smart devices

dominates the activity recognition area [1], [11]. Obvious advantages of inertial sensors can be found over other sensor modalities, ubiquity, cheap installation, unobtrusiveness, usability, ease of use, and free of privacy issue. Since the inertial sensors are attached to human body, they carry much more body information than other sensors, which can infer human activities in detail. Moreover, there is no occlusion or visibility range issues exist in this case. Since activities of daily living (ADLs) are ambulatory and trivial with high level of diversity, unsupervised machine learning algorithms are unlikely to identify ADLs. Thus, most activity recognition algorithms are supervised learning based, where training data with reliable labels is critical. However, the annotation process of the training data requires huge human effort to scan through the raw data for manual labels. In order to model activities effectively and increase the generalization of the model, large dataset with various human activity modalities is needed, which makes the annotation process more challenging. The annotation issue is one of the major challenges remains unsolved in the inertial sensor-based activity recognition approaches.

In this paper, we propose an automatic data collecting and labeling system by addressing the annotation issue, and a novel coarse-to-fine ADLs recognition strategy for older adults, by combining supervised and unsupervised machine learning methods. Since older adults want to remain independent with dignity for as long as possible, especially solitary older adults, we aim at helping solitary older adults with functional ability, to live independently longer, by modeling their activities. For this purpose, we collect data from smart homes with multiple sensors, and annotate the data automatically, by using a multiple sensors fusion strategy. Previous studies tend to focus on differentiating solely among ambulatory activities in controlled environments, which is the most fundamental form of the problem in the activity recognition sense. Although success has been achieved in this regime, with reported high sensitivity rates and low misclassification rates, this sort of scenario is too idealized to be deployed. The diversity of ADLs in free-living situations may confuse the models trained in controlled environments. Addressing this challenge, we propose a coarse-to-fine strategy to discover activity clusters and recognize them then. An unsupervised machine learning method is employed to discover and segment the raw sensing data into big clusters. This is a natural step to group the raw sensing data into big clusters firstly, since large clusters would be much easy for identification [12]. Then, a Hidden Markov Model (HMM) is used to assign the clusters with a well-defined activity set. Since HMM employs a mathematical model based random process, to describe the courses of activities, the statistical property gives HMM the capability in modeling a random signal sequence (activity) with multiple features.

The rest of this paper is organized as follows. In Section II, we describe the automatic data labeling system and the coarse-to-fine ADLs recognition strategy. More specifically,

we introduce the data collection protocol and smart home settings for solitary older adults in Section II-A. In Section II-B, we present the details of the proposed automatic data collecting and labeling system. Section II-C describes the coarse-to-fine ADLs recognition strategy. The evaluation experiment setup and the experimental results are shown in Section III. Finally, we make discussion on the obtained results and draw conclusion in Section IV.

II. METHOD

In order to help solitary older adults remain independent, maintain functional ability, and live at home longer, we research the activity model of solitary older adults and secure their safety at home. Collaborating with several local independent aging facilities, we build a smart home for older adults with an unobtrusive and continuous monitoring system. This monitoring system consists multiple sensors, a two dimensional (2D) laser scanner, a visual-depth information (RGB-D) camera, and a smart watch. The 2D laser scanner and the RGB-D scanner are environmentally mounted, while the smart watch is worn on one hand.

Due to the limitations of environmentally mounted sensors, such as occlusion and coverage, signals describing activities in different angles are collected from different sensors, and then are fused to interpret and model activities. The labeling issue is one of the biggest challenges in interpreting and modeling activities. Thus, we propose an automatic labeling algorithm based on sensor fusion, and model the the activities of older adults with IMU signals, to avoid the limitations of environmentally mounted sensors.

A. SUBJECTS AND EXPERIMENT SETTING

15 healthy older adults: 8 males, 7 females, age 66.4 ± 5.6 years, height 165.2 ± 5.7 cm, body mass 68.6 ± 13.8 kg, and Body Mass Index (BMI) 24 ± 7.2 , entered our study non-consecutively from August 2019 to November 2019. They are recruited by the Office of Retirement Services of Harbin Institute of Technology at Weihai. Ethics Committee approval was obtained from the Institutional Ethics Committee of Harbin Institute of Technology at Weihai.

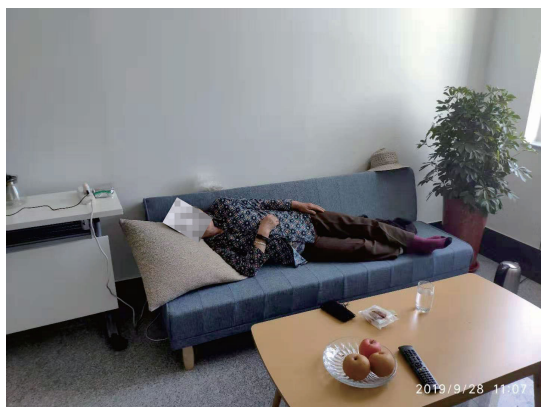


FIGURE 1. One apartment in the smart home with one older adult.

Since we target at modeling ADLs of older adults in free-living situations, we do not have any training process for the older adults. Each older adult was invited to an apartment in the smart home, and lived there for a week or more, as shown in Fig. 1. The smart watch was required to wear on the non-dominant hand, to reduce the motion noise level. And older adults were asked to charge the smart watches at any time convenient to them, like taking a shower or having dinner. Since several ADLs are rare in daily living, the environment and layout of smart home is designed to guide the activities of participants, by considering the integrality and balance of the dataset we are going to create. The end table is 0.55m away from the sofa, which is out of reach when sitting/lying on the sofa. And the end table is about 0.42m high, which means the participant has to get up and bend for the things on the end table. An electric kettle is placed on the ground. The participant needs to perform crouching activity for a while when operating the kettle.

B. MULTIPLE SENSORS FUSION BASED AUTOMATIC DATA COLLECTING AND LABELING SYSTEM

With great advances in embedded computing technologies, smart wearable devices with inertial sensors play significant roles in our daily living, especially in health care. Thanks to the pervasiveness of these wearable devices, great potential has been demonstrated in characterizing human activities and alleviating stress on limited medical resources. The sensing data, such as acceleration and gyroscope, can be explored to model the activities for older adults and uncover their health status. However, after a decade of research in modeling activities of older adults, accurate and reliable systems in use are few. In order to train models, tremendous efforts have been placed in data collecting and labeling. Many of them suffer from huge human effort, technical and privacy limitations [13]. It is still challenging to collect accurate and reliable activity data with labels. As a result, there is no commonly used dataset for older adults exists. Addressing this challenge, we propose a data collection system with an automatic annotation strategy, to collect the ADL sensing data of older adults and label it automatically. In the annotation process, we focus on an activity set: standing, walking, bending, lying, crouching, and sitting, which are ADLs closely related to basic independent living.

1) HARDWARE DEVICES

The automatic data collecting and labeling system consists of a 2D laser scanner (3irobotix lidar C0602), a RGB-D camera (Intel RealSense camera D415), a smart watch (Huawei watch 2), and a Raspberry Pi computer (Raspberry Pi 3B+), as shown in Fig. 2. The 2D laser scanner is horizontally mounted on the ground under the sofa, to capture the foot locomotion. Ranging laser technology provides a continuous stream of distance data, which takes advantages in wide field of view and consistency of any lighting condition. It gives us the potential to capture the foot locomotion rapidly and

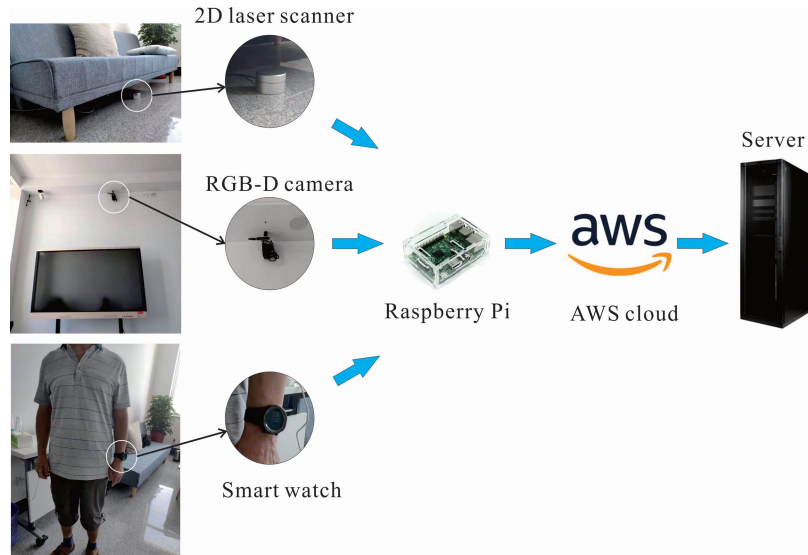


FIGURE 2. The automatic data collecting and labeling system for modeling ADLs of solitary older adults. The 2D range data (2D laser scanner), the depth images (RGB-D camera), and the IMU sensing data (smart watch) are collected and transferred to the AWS cloud for storage, by using a Raspberry Pi computer. The data is downloaded to a local server, annotated by using our automatic labeling system, subsequently.

measure walking speed precisely. Considering the high levels of occlusion and part self-occlusion, sofas with thin legs are chosen, and the laser scanners are mounted as far as possible from the feet. The RGB-D camera is on the ceiling above the television, facing the sofa in the living room. The depth images captured by the RGB-D camera are used to identify the posture of older adults. Since we intend to model the ADLs of solitary older adults with the lowest invasive, the smart watch is the only sensor to be amounted on the body. As we discussed previously, the laser scanner and the camera are environmentally mounted, they cannot collect the activity data continuously, when the participant gets out of the sights of the devices. Therefore, we tend to model the activities of older adults with the smart watches, by combining the laser scanners and the cameras. The Raspberry Pi computer transfers the data collected to the AWS cloud. And the de-identified and encrypted data on the AWS cloud can be used for labeling and activity modeling. Finally, the ADLs data is annotated by using our automatic labeling system, to form the ADLs dataset for activity modeling.

The 2D range data is generated by the laser scanner at 6.2Hz with an angular resolution of 0.5° , and stored in point clouds. The range radius of the point cloud is 10m. The RGB-D camera has two depth imagers, an infrared projector, and a RGB module. Two depth imagers enhance the RGB-D camera capability of capturing depth image with wide field of view. And the infrared projector enables the camera to work under poor lighting conditions, with a large distance range from 0.2m to over 10m. Considering the computational burden of Raspberry Pi computer, the transmission speed limit of USB2.0, and the network bandwidth, we empirically determine the resolution and the frequency of the camera to be 640×480 and 6Hz, respectively. We collect barometer signal

at frequency 20Hz, three axes of acceleration and gyroscope at frequency 50Hz, from the smart watch. By switching off the network and uploading the data when charging, we make the smart watch maintain a reasonable battery life, the watch operational cycle can reach as long as 28 continuous hours. With the long battery life, participants can charge the watches at any time convenient to them, like taking a shower. Moreover, the build-in memory of smart watch can store continuous data for 14 days without uploading.

2) RANGE DATA BASED WALKING SPEED ESTIMATION

Ranging laser technology provides a continuous stream of distance data, which gives us the potential of capturing foot locomotion rapidly and measuring walking speed precisely. When the point cloud is generated by the laser scanner, K-means classifier [14] is employed to partition the point cloud into different clusters (including the foot point cluster), as shown in Fig. 3. In our work, a foot is assumed to have an approximate circle shape. Since the laser scanner mounted on the ground under the sofa environmentally, we just have the foot silhouette in one direction. Thus, a point cluster has an arc or convex shape within a reasonable ratio will be classified as foot. Based on this foot shape assumption, several foot features are employed to identify and trace the two feet, which are detailed below.

1) Reasonable distance ratio F_d . The central point of each point cluster is calculated. The foot point cluster should be within a reasonable Euclidean Distance, F_d , as follows:

$$F_d = |p - P_c| < 0.14 \quad (1)$$

where $|\cdot|$ denotes the L_2 norm; $p = x, y$ is one point in the point cluster P , $\{x, y\}$ is the coordinate of point p ; P_c is the central of the point cluster P ; the distance ratio is set to

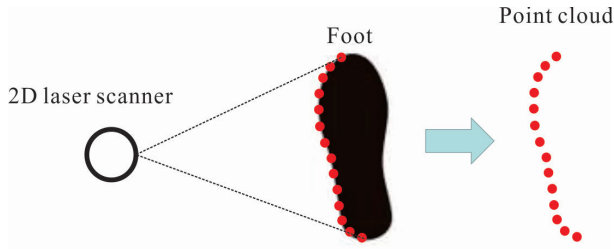


FIGURE 3. The foot silhouette in one direction generated by the laser scanner. A point cluster is assumed to have an arc or convex shape, which is a typical shape of foot. Other directions have the similar observations.

0.14m, which is the maximum distance from cluster edge to the cluster center.

2) Foot length F_l . The length of one point cluster in moving direction is estimated as foot length, as follows:

$$F_l = |p_f - p_b| \quad (2)$$

where p_f and p_b are the two points with maximum distance in the moving direction.

3) Foot circularity F_c . The radian of each point in the cluster edge is calculated. And the mean value of the radians is considered as approximate foot circularity, F_c , as follows:

$$F_c = \frac{1}{n-1} \sum_{i=2}^n \operatorname{acos} \frac{|p_i - P_c|^2 + |p_{i-1} - P_c|^2 - |p_i - p_{i-1}|^2}{2|p_i - P_c| \cdot |p_{i-1} - P_c|} \quad (3)$$

where p_{i-1} and p_i are two adjacent points in the cluster edge, n is the number of points in the cluster edge.

4) Foot arc length F_a . The sum of the Euclidean Distance between two adjacent points is calculated as foot arc length as follows:

$$F_a = \sum_{i=2}^n |p_i - p_{i-1}| \quad (4)$$

where p_{i-1} and p_i are two adjacent points in the cluster edge, n is the number of points in the cluster edge.

Due to the dynamics of walking speed and trivial foot motion in daily living, estimating walking speed in daily living is really challenging. Thus, we take a frame work combining Random Forest and Kalman Filter to detect feet and track them. In this frame work, we assume that the background is static, and the foot appearance is fixed. Since the real-world environment is with high diversity, accurate walking speed estimation asks for identifying feet from other moving objects, such as crutches and animals. Random Forest is employed to detect feet, and results in foot candidates. Then foot candidates are fed into Kalman Filter for foot tracking. Finally, the walking speed is estimated based on the tracking trajectory.

Random Forest is one of the top performers for classification tasks. By organizing multiple decision trees in a cascade manner, Random Forest embeds the randomness in the training phase, which gives the Random Forest capability of dealing missing data and unseen data by reducing overfitting.

In our research, missing data and unseen data were observed in occlusion and part self-occlusion situations. And Random Forest can help the foot tracking process keep accurate. There are many open source implementations of Random Forest, such as OpenCV for C language [15] and Scikit-learn for Python language [16]. We combine OpenCV and Scikit-learn to gain both detection accuracy and computational efficiency, by executing the OpenCV implementation of Random Forest algorithm, with partially modified Scikit-learn parameters. Here, we change the default setting of maximum tree depth to 20, and training iteration to 100.

In order to avoid human effort involved in the annotating process, we collect two training datasets, no-person and single-person datasets. Three indoor environments have been chosen. For each environment, single-person and no-person datasets are collected, 12 minutes duration each. Since the backgrounds are static and same in both single-person and no-person datasets for each environment, it is easy to annotate background (negative dataset) and human feet (positive dataset).

With the trained Random Forest classifier, the foot candidates are detected for each frame of the point cloud. And the centroids of the foot candidates are fed into the Kalman Filter for foot tracking. Each centroid will be assigned to a Kalman tracker (Kalman filter). The Kalman filter has been proved to be effective and accurate in estimating a target's velocity and position [17]. Addressing the occlusion issue in the foot tracking, we employ the Kalman filter [18] to model the walking activity as a dynamic system, by using the OpenCV implementation, as following:

$$\begin{aligned} X_k &= A_k X_{k-1} + B_k u_k + w_k \\ z_k &= H_k X_k + v_k \end{aligned} \quad (5)$$

where $X_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]$ is the state vector of a centroid c_k in frame k , and $z_k = [x_k, y_k]$ is the measurement vector. x_k and y_k are the position components, while \dot{x}_k and \dot{y}_k are the velocity components; A_k is a state transition matrix; u_k is the control vector containing acceleration force; B_k is the control matrix of acceleration effect; w_k is the process noise; H_k is the transformation matrix, mapping X_k into measurement z_k ; v_k is the observed noise. Since the frequency of the 2D laser scanner is 6.2Hz, the interval between two frames is 0.16s, which is small for walking activity. Thus, we assume that the foot velocity is constant. As a result, the control vector u_k is set to 0.

Here, the Kalman filtering process takes two stages: prediction and measurement. Take one tracker for example. For frame k , the centroid c_k is predicted, based on the transition matrix A_{k-1} and the centroid c_{k-1} . Then, we search for foot candidate around the predicted centroid c_k . If the foot candidate does exist in the predicted area, the centroid c_k will be updated. And the association between c_k and c_{k-1} is recorded in the transition matrix A_k , which will be used for prediction in next iteration. If the foot candidate does not exist in the predicted area, occlusion may happen.

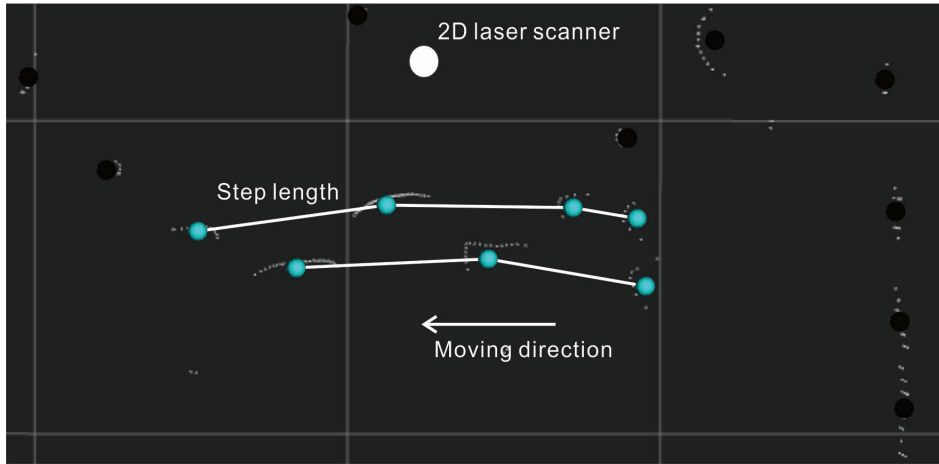


FIGURE 4. Foot tracking and walking speed estimation. Small white points are the point clouds, while dark points and blue points indicate the detected background and the central points of foot clusters, respectively. White lines are the distances between footprints. For demonstration purposes, point clouds are shown every 8 frames.

The predicted centroid c_k is reserved, and the transition matrix A_k is also updated based on c_k . This tracker is reserved waiting for a foot candidate which matches the prediction. If there is no match for the predictions in the following frames, the tracker will be removed, and the detected centroid c_{k-1} will be considered as the end of a walking activity. The step length between footprints in the foot trail is used to calculate the walking speed, as shown in Fig. 4. Left foot and right foot are identified according to the moving direction and the distance to the scanner. And the two feet are tracked separately. Finally, the average speed of the two feet is defined as the walking speed for each step in the tracking period. If one footprint is occluded (missing), another footprint is used to calculate the walking speed. In the worst case, two footprints are missing, next pair of footprints will be used for calculation. In our experiment, we observed only one pair of footprints missing. And it did not affect the accuracy in walking speed estimation. The foot candidates in the first frame are used as initial position, and the initial foot velocity is set to 0.

3) DEPTH IMAGE BASED POSTURE ESTIMATION

The posture estimation runs parallel to the walking speed estimation, recognizing activities from the perspective of complementarity. Addressing the privacy issue, depth image is chosen for posture estimation instead of RGB color image. The pre-trained Convolutional Neural Network (CNN) of OpenPose is fine-tuned to extract human skeleton structure, by using transfer learning from RGB images to depth images. OpenPose [19] is an open source library for multi-person skeleton detection. It is worth to mention that, this person skeleton detection system achieves high accuracy and real-time performance. Then, the skeleton-based features are fed into a Random Forest for posture recognition.

In the ADLs dataset we mentioned previously, depth images and aligned RGB images are collected simultaneously. We employ a pretrained CNN of OpenPose to extract

human skeleton structures, by using the RGB images. The extracted human skeleton structures are mapped to the corresponding depth images. The obtained human skeleton structures and the corresponding depth images form the posture dataset, to obtain our own CNN for depth images. Human skeleton structure with 25 keypoints is used in this work, which are nose, eyes, ears, neck, shoulders, elbows, wrists, mid-hip, hips, knees, ankles, big toes, small toes, and heels, as shown in Fig. 5. Less important keypoints are not shown for demonstration purposes, such as eyes and heels. In order to avoid confusion, the 6 activities in the activity set: standing, walking, bending, lying, crouching, sitting, are used as postures here. The postures are marked by 3 human raters independently. Since the dataset of depth images and aligned RGB images is huge, labeling the postures by going through the images frame by frame manually is time consuming and unrealistic. We take a semi-automatic method to mark the dataset, which is described later. Secondly, the pretrained CNN of OpenPose is fine-tuned by using Caffe fine-tuning toolkit [20] on our posture dataset. Our depth image based

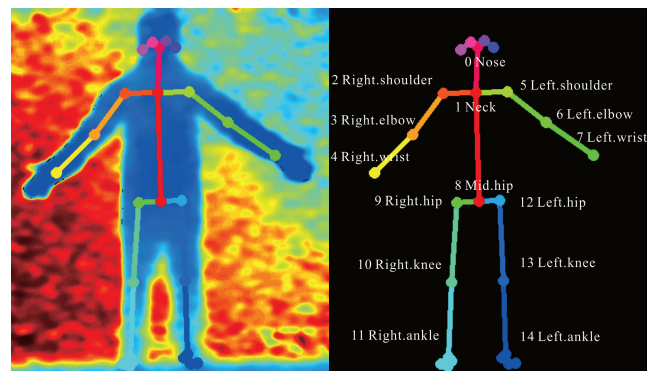


FIGURE 5. Skeleton structure with keypoints extracted from depth image. Keypoints of less importance, eyes, ears, toes, and heels, are not shown here for demonstration purposes.

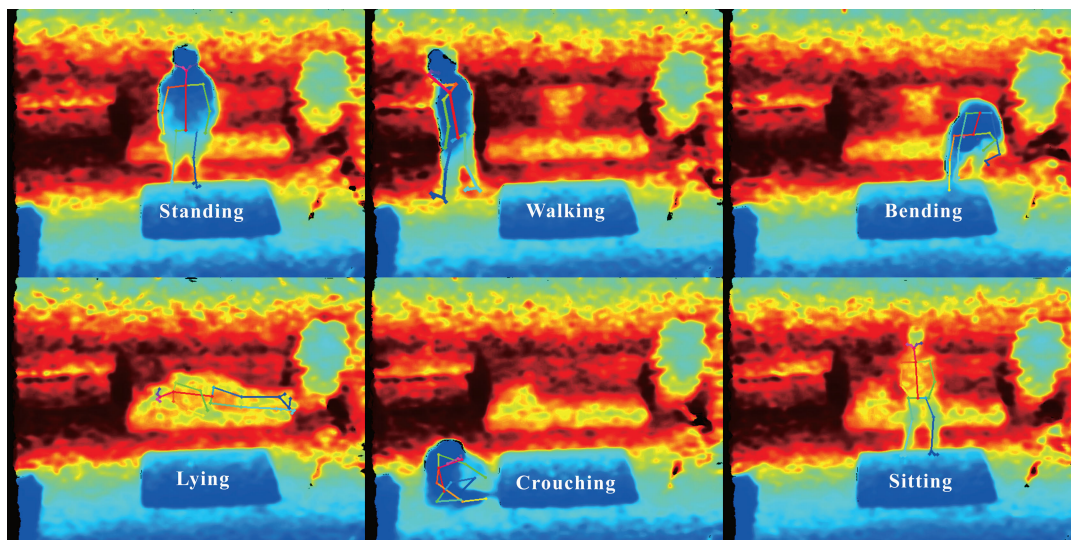


FIGURE 6. The skeleton structures extracted by our tuned CNN and the 6 postures estimated by our posture estimation algorithm.

task is related to the task of extracting human skeleton structures from RGB images. Thus, there is no need to train a new model with costly relearning. What’s more, our posture dataset is a relatively small collection, comparing with the datasets used to train the CNN of OpenPose, MPII [21] and COCO [22]. Due to a small size of our dataset, retraining a brand new CNN could largely increase the possibility of overfitting, whereas transfer learning requires much smaller dataset [23]. The fine-tuning process is implemented in Matlab 2017b with Caffe toolkit, and runs on a Titan GPU with 12GB video memory under Linux (Ubuntu 16.04).

Once the skeleton structure with 25 keypoints is extracted from the depth image, as shown in Fig. 6, geometrical angles and three dimensional (3D) pairwise distances are calculated as posture features. The body trunk angle, a_{trunk} , is designed to identify bending and lying from other postures, which is defined as the angle between the body trunk and the horizontal plane. The definition of a_{trunk} is given as following:

$$a_{trunk} = \text{acos} \frac{\vec{n} \cdot (p_{neck} - p_{mid.hip})}{|\vec{n}| \cdot |p_{neck} - p_{mid.hip}|} \quad (6)$$

where $\vec{n} = (1, 0, 0)$ is the normal vector of the horizontal plane; $p_{neck} = \{x_{neck}, y_{neck}, z_{neck}\}$ denotes the 3D coordinates of keypoint neck (Fig.), x_{neck} , y_{neck} , and z_{neck} are the x-coordinate, y-coordinate, and intensity of keypoint neck in the depth image.

The hip angle, a_{hip} , is designed to classify sitting and crouching from standing and walking, and the definition is given as following:

$$a_{hip} = \frac{a_{left.hip} + a_{right.hip}}{2} = \text{acos} \frac{a_{left.hip}(p_{neck}, p_{left.hip}, p_{left.knee}) \cdot (p_{left.knee} - p_{left.hip})}{|p_{neck} - p_{left.hip}| \cdot |p_{left.knee} - p_{left.hip}|} \quad (7)$$

where $a_{left.hip}(p_{neck}, p_{left.hip}, p_{left.knee})$ denotes the angle extracted from three adjacent keypoints, neck, left hip, and left knee (Fig.); $p_{left.hip}$ and $p_{left.knee}$ are defined in the same way of p_{neck} ; $a_{right.hip}$ can be calculated from another three adjacent keypoints, neck, right hip, and right knee. Considering occlusion or part self-occlusion, $a_{left.hip}$ or $p_{left.knee}$ is used as a_{hip} for further processing.

Defined in the same way, $a_{left.shoulder}(p_{neck}, p_{left.shoulder}, p_{left.elbow})$, $a_{right.shoulder}(p_{neck}, p_{right.shoulder}, p_{right.elbow})$, $a_{left.knee}(p_{left.hip}, p_{left.knee}, p_{left.ankle})$, and $a_{right.knee}(p_{right.hip}, p_{right.knee}, p_{right.ankle})$ are employed to identify standing from walking, bending from lying, and crouching from sitting.

Since occlusion and part self-occlusion frequently occur in ADLs, distances between paired keypoints are calculated for posture recognition in a complementary way. To calculate distance features, we create two keypoint clusters $L = \{\text{left knee, right knee, left ankle, right ankle}\}$ and $U = \{\text{left elbow, right elbow, left wrist, right wrist}\}$. Paired keypoints are selected randomly from these two clusters and keypoint neck. For example, right wrist and left knee are chosen from cluster L and U , respectively. The distance between right wrist and left knee, $d_{right.wrist\&left.knee}$, is calculated as following:

$$d_{right.wrist\&left.knee} = |p_{right.wrist} - p_{left.knee}|/s \quad (8)$$

where s is a scaling parameter. The 3D distances are invariant to rotation. However, they are with large variation in respect to different people. To ensure the scale invariant, we normalize the distances with a scaling parameter s . The distance between two shoulder keypoints is defined as the scaling parameter. The two shoulder keypoints are always visible for our RGB-D camera. The distance between them is proportionally related to the height of one particular person. And it is stable and does not change dramatically through frames. These are the reasons we choose two shoulder keypoints to calculate the scaling parameter. Please notice that, keypoint

missing caused by occlusion is inevitable. When keypoint missing happens, features are set to Null.

The seven geometrical angle features are used to mark the postures semi-automatically, under the supervision of the human raters. a_{trunk} , $a_{left.knee}$, $a_{right.knee}$, $a_{left.hip}$, and $a_{right.hip}$ can be employed to divide standing, lying, sitting, and crouching into sever categories roughly according to continuous time, by using a threshold based method. Thresholds for the features are decided empirically. a_{trunk} , $a_{left.shoulder}$, and $a_{right.shoulder}$ are used for posture bending, while $a_{left.hip}$, $a_{right.hip}$, $a_{left.knee}$, and $a_{right.knee}$ are for posture walking. Since the postures of older adults are continuous in time, the human raters focus on the beginning and end of each category, and mark the RGB images. The labels for depth images can be obtained subsequently.

Once the features are calculated for one frame, a feature vector is formed and fed to train a Random Forest classifier, as we introduced in Section II-B.2. Since the Random Forest implementation does not accept Null data, the Null features are set to 2π and 100 for geometrical angles and 3D pairwise distances, respectively. In our experiment, we found that the posture estimation subsystem cannot identify standing from walking accurately. As we can observe from Fig. 6 (standing and walking), two postures standing and walking are very similar to each other. The walking speed estimation subsystem can help in recognizing these two postures, where walking speed more than 0.05m/s is considered as walking. Moreover, the posture estimation subsystem also has difficulties in identifying bending and crouching, because of part self-occlusion. It's worth mentioning that small part of occlusion would not affect the skeleton extraction, see Fig. 6 standing and sitting.

4) AUTOMATIC IMU DATA LABELING

In this research, we try to model ADLs of older adults with the streaming barometer, acceleration, and gyroscope data, which are collected from smart watches. As we all know, ADLs are more ambulatory and trivial than the activities in labs. There is no clearly distinct boundary to separate them from each other. In daily living, there are numerous situations, in which one activity may be interrupted by other activities, and divided into small activity segments. For example, walking is interrupted by short stops, such as standing. Traditional activity recognition algorithms cluster the walking segments and the stops as different activities, and fragment the walking into several parts. The fragmenting walking activities with different interruptions have different motion features, which make the modeling algorithm confused.

Inspired by the unsupervised activity discovery algorithm, Unbounded Unsupervised Activity Discovery using the Temporal Behaviour Assumption (UnADeVs) [24], we classify the sensing data into big clusters, by defining the minimum duration of ADLs. The UnADeVs algorithm has the capability in discovering activity clusters, corresponding to periodic and stationary activities in sensing data, not limited to acceleration and gyroscope data. There are three key parameters

of UnADeVs, the number of active cluster, the duration that a cluster can remain active (tolerance), and the minimum duration of a cluster. The UnADeVs discovers activities in a growing way. Firstly, the data is segmented into overlapping windows by using a sliding window technique [27]. Then, feature vector $[b, SMV_{acc}, a_x, a_y, a_z, SMV_{gyro}, g_x, g_y, g_z]$ is calculated for each window, where b is the barometer, SMV_{acc} is the signal magnitude vectors (SMV) [13] of acceleration, a_x , a_y , and a_z are the three axes of acceleration, SMV_{gyro} , g_x , g_y , and g_z are with the same definitions for gyroscope. For one window waiting for clustering, the distances between active clusters and the window are calculated. The cluster with minimum distance will be chosen for growing, by adding the window into the cluster. The number of active cluster determines the deviation of the target window. The tolerance parameter decides which cluster should be turned into inactive, and moves a new cluster into active. At the meantime, the minimum duration parameter prevents some small clusters from creating. The size of the sliding window is set to 2s, and two consecutive windows overlap with each other for 1s.

A voting strategy is employed to label the sensing data, as shown in Fig. 7. For each discovered activity cluster, the sliding window technique is used to segment the cluster into overlapping windows. And the windows of walking speed estimation vote for walking and still, while the windows of posture estimation vote for other activities in a complementary way, as we described at the end of Section II-B.3. Since the walking speed estimation is with high accuracy, the voting from walking speed estimation is with high priority. Although the walking speed estimation and the posture estimation are combined to improve the labeling accuracy, misclassifications were reported in several sling windows. Because of the interruptions of other activities, the misclassifications in several windows are reasonable. What's more important, misclassifications with much lower detection rate do not affect the final labeling results (more details in Section III-B and III-C).

Before the labeling process, the voting strategy is used to tune the parameters of the UnADeVs algorithm. The initial parameters are set to 3, 22s, and 16s for the number of active cluster, the tolerance, and the minimum duration of a cluster, respectively. For each cluster, activity with highest percentage wins the vote. And the mean value of the highest percentages for all clusters is used to tune the three key parameters. The larger the mean value is, the better the parameters are. We empirically change the value of the parameters, to achieve the highest mean value. As a result, we got the parameter set: 7, 30s, and 25s for the three key parameters.

With the tuned parameters, the sensing data is classified into clusters, as discovered activities. The voting strategy is used again to label the clusters, as we described previously. For one cluster, the activity with the highest voting percentage will be assigned to this cluster. Finally, the sensing data clusters with activity labels form the ADL dataset for older adults.

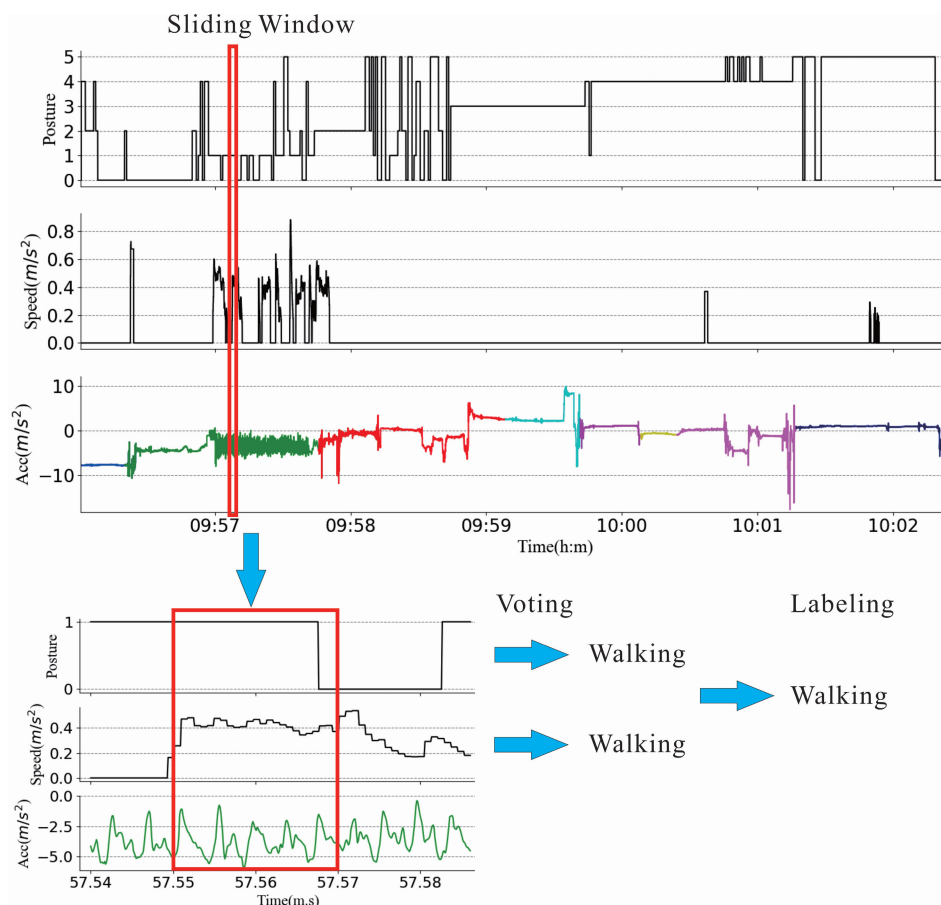


FIGURE 7. Voting strategy for labeling the sensing data. Y axis of acceleration is shown in different colors (bottom), corresponding to the discovered activity clusters. Walking speed estimation is shown in the middle, while the posture estimation is plotted on the top. Time axes are shown in hour:minute and minute:second. The numbers {0,1,2,3,4,5} in the top line plot indicate the postures standing, walking, bending, lying, crouching, and sitting, respectively. The red rectangles are the sliding windows.

C. COARSE-TO-FINE ADLS RECOGNITION STRATEGY

Since it is difficult to separate ADLs from each other as we discussed previously, the coarse-to-fine strategy is employed to classify ADLs into big clusters roughly, and refine them subsequently. Firstly, the sensing data goes through the tuned UnADeVs algorithm. Clusters are generated as discovered activities. Secondly, the clusters are divided into overlapping windows, $W = \{..., w^i, ...\}$ by using the sliding window technique. The overlapping windows are further divided into small slots, which are defined as sampling periods. Let us denote the slots as $w^i = \{..., s_j^i, ..\}$, where s_j^i is the j th slot in window w^i . We divide the window into ten slots in this work. Since acceleration signal cannot identify activities with small motions, such as standing, lying, and sitting, gyroscope signal is combined with acceleration signal to detect the pose change of forearm. And the barometer signal is employed to recognize activities with different altitudes. Feature vectors with nine features (we mentioned in Section II-B.4) are calculated for slots. Thirdly, the feature vectors of slots are fed into the HMM training process [29]. Finally, posterior probabilities of activities in the activity set are estimated by the trained HMM for the slots. For one slot, the activity with highest probability

will win and be assigned to this slot. Slots vote for windows, and windows votes for clusters, subsequently.

The HMM can naturally identify the ADLs, by modeling temporal dependencies between consecutive activities. And numerous strong results have been obtained by using HMM for modelling ADLs [28]. Thus, a HMM is employed to recognize activities by using the sensing IMU data in this research.

The HMM for activity recognition can be expressed in a five item array as: $\phi = (M, N, \pi, A, B)$, where M is the number of invisible states (activities). Since there are 6 activities in the activity set, the number of invisible states is set as $M = 6$; N is the number of observation values, which will be described later; π is the initial state distribution corresponding to the invisible states, $\pi = \{\pi_m\}, m = 1, \dots, M, \sum_m \pi_m = 1$; A is the state transition matrix with size $M \times M$, which describes the transition probability between two states; B is the emission matrix with size $N \times M$, which describes the emission distribution of the HMM. The ADLs dataset for older adults we created is used to train the HMM.

The implementation of the HMM for our research has been carried out by using Seqlearn for Python language.

The initialization condition for training a HMM is summarized as following: 1) The number of invisible states $M = 6$, since we take 6 activities into consideration; 2) The number of observation values $N = 26$; 3) The initial state distribution: $\pi_1 = 1, \pi_j = 0, j = 2, \dots, 6$, corresponding to the 6 activities; 4) The initial state transition matrix A obeys uniform distribution by general principle [10]; 5) The initial emission matrix B also obeys uniform distribution.

III. EXPERIMENTS AND RESULTS

A. EVALUATION ON WALKING SPEED ESTIMATION SUBSYSTEM

Various protocols in measuring walking speed exist in the literature, such as Timed-Up and Go (TUG) and Timed 25 Feet Walk (T25FW). These physical performance instruments are frail for elderly populations, suffering from significant intra-individual test-retest variability [25]. Habitual Gait Speed (HGS) is reported to be reliable and considered as an useful indicator in clinical trails [30]. The measuring process of HGS is easy in implementation, which requires no doctor or clinical equipment. Therefore, HGS is chosen and used as ground truth, to evaluate the range data based walking speed estimation subsystem. As we proposed an automatic labeling system, which consists of two subsystems, walking speed estimation subsystem and posture estimation subsystem. And we are going to evaluate these two subsystems separately.

15 older adults were recruited to take part in the evaluation experiment. Distance in HGS measurement is the main fact in influencing the accuracy of measuring gait speed. HGS over 4 metres has been reported excellent reliability in clinical trails [30]. In our experiment, the participants were asked to walk a 5.5 metres path with their normal speed, and repeat the test 5 times. Before the experiment, the participants could practice walking on the path. The 2D laser scanner was mounted on the ground just beside the walking path, as shown in Fig.8. The range data was collected, when the participants were walking on the path. At the mean time, we timed the walkings to calculate the ground truth walking speed by using a stopwatch.

Since we estimate the walking speed of each step, absolute error range, mean absolute error, and error variance are employed to validate the walking speed estimation subsystem, as shown in Table 1. All intraclass mean absolute error is 0.06m/s. Slightly higher mean absolute error value was reported for the youngest female participant (age 57, walker), with the highest error of 0.11m/s. However, the highest error is small according to the mean absolute errors (0.06m/s), which is an evidence of the accuracy of the walking speed estimation subsystem. The slower the walking is, the more accurate the estimation is. As far as we know, most walking activities of older adults are slow in free-living situations, slower than 0.60m/s. Considering manual operation error (timing with a stopwatch), the accuracy of our walking speed estimation subsystem is reasonable and acceptable for labeling the dataset in the complementary way.

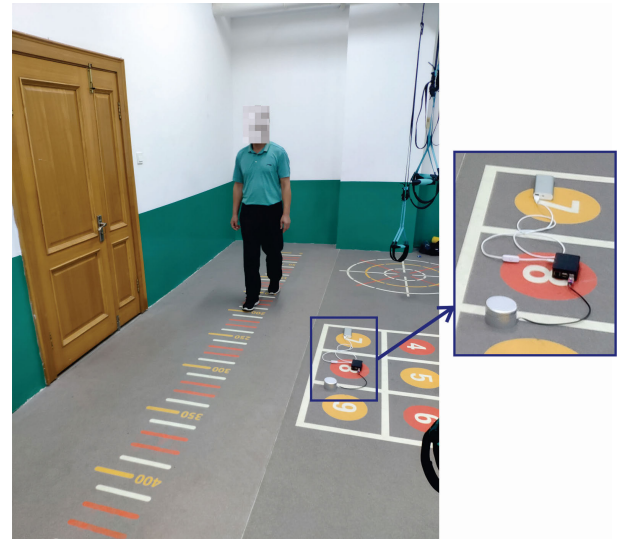


FIGURE 8. Range data based walking speed estimation subsystem. The 2D laser scanner was mounted on the ground just beside the walking path. A Raspberry Pi computer was used to upload the range data, which was supplied by a power bank. Please see the enlarged area (blue rectangle) for details.

TABLE 1. Mean absolute errors and error variance for evaluation on walking speed estimation. Male (M) and Female (F).

| Sex | Age | Timed Walking Speed(m/s) | Error Range (m/s) | Mean±Std (m/s) |
|-------------------------|-----|--------------------------|-------------------|----------------|
| M | 63 | 0.66 | [0.03, 0.07] | 0.05±0.02 |
| F | 67 | 0.72 | [0.05, 0.08] | 0.06±0.02 |
| F | 69 | 0.57 | [0.02, 0.08] | 0.04±0.01 |
| M | 70 | 0.58 | [0.02, 0.08] | 0.04±0.02 |
| M | 68 | 0.65 | [0.05, 0.06] | 0.05±0.01 |
| M | 65 | 0.74 | [0.05, 0.08] | 0.06±0.03 |
| F | 57 | 0.95 | [0.04, 0.11] | 0.09±0.04 |
| F | 68 | 0.69 | [0.05, 0.07] | 0.05±0.03 |
| F | 73 | 0.64 | [0.04, 0.06] | 0.04±0.02 |
| F | 71 | 0.71 | [0.05, 0.06] | 0.05±0.01 |
| M | 58 | 0.79 | [0.05, 0.07] | 0.08±0.02 |
| M | 64 | 0.58 | [0.03, 0.05] | 0.04±0.03 |
| F | 59 | 0.67 | [0.04, 0.06] | 0.04±0.02 |
| M | 77 | 0.72 | [0.04, 0.08] | 0.07±0.01 |
| M | 67 | 0.69 | [0.03, 0.08] | 0.06±0.02 |
| All Intraclass Mean±Std | | | | 0.06±0.02 |

B. EVALUATION ON THE POSTURE ESTIMATION SUBSYSTEM

The posture estimation is designed as a subsystem of the automatic labeling system, and runs parallel to the walking speed estimation in a complementary manner, since the posture estimation is assumed to be not accurate due to the slow movements and the occlusion issue. Thus, the evaluation here is designed to check the complementary between the walking speed estimation and the posture estimation. Firstly, the classification accuracy is employed to locate the limitations of the posture estimation subsystem, by using the ADLs dataset we created. Then, the complementary is checked in

TABLE 2. Confusion matrix, illustrating the accuracy of the posture estimation. The classification accuracies of the 6 postures are shown in bold, while noticeable misclassification rates are shown in gray areas.

| | | Target Posture (%) | | | | | |
|--------------------|-----------|--------------------|-------------|-------------|-------------|-------------|-------------|
| | | standing | walking | bending | lying | crouching | sitting |
| Output Posture (%) | standing | 92.5 | 7.9 | 1.6 | 0 | 1.4 | 2.4 |
| | walking | 6.5 | 78.5 | 7.8 | 0.7 | 5.2 | 1.1 |
| | bending | 0 | 4.1 | 84.1 | 0 | 6.6 | 0.6 |
| | lying | 0 | 1.3 | 0.4 | 99.3 | 0 | 0 |
| | crouching | 0.3 | 6.9 | 4.1 | 0 | 85.2 | 0 |
| | sitting | 0.7 | 1.3 | 2.0 | 0 | 0.3 | 95.9 |



FIGURE 9. Problematic areas of posture estimation in identifying standing and walking. Red rectangles indicate the misclassification areas. The speed estimation can help in refining the estimated postures. Time axis is shown in minute.second.

the misclassification areas, to find out whether the posture estimation can be refined by the estimated walking speeds.

The confusion matrix of classification accuracy is used to locate the limitations, as shown in Table 2. The classification accuracies of the six postures are shown in bold, while noticeable misclassification rates are shown in gray areas. It can be easily observed that, the accuracy of the posture estimation subsystem is high (average accuracy 89.3%). As the posture estimation is designed as a subsystem of a ground truth labeling system, let us look into the gray areas (misclassification areas), to check the complementary between the walking speed estimation and the posture estimation. The biggest limitation of the posture estimation subsystem focuses on identifying walking from standing, bending, and crouching, which is with the lowest recognition accuracy (78.5%). A graphical user interface (GUI) was developed to help us look into the misclassification areas, as shown in Fig. 9. It can be easily observed that, with the help of speed estimation, walking postures can be easily identified from the other three postures (walking speed larger than 0.1m/s). With the still estimation, walking speed 0m/s, standing posture can be refined from walking misclassification, Fig. 9 the first two rectangles.

The remaining limitations focus on identifying bending and crouching. Since these two postures rest at relatively quiet status, the walking speed could not refine the posture estimation. We further looked into the classification areas of these two postures, and found that the misclassifications did not gather together. With much lower misclassification rates than the accuracy rates, the misclassifications of these two postures do not affect the final voting labels.

C. DOUBLE-CHECK ON THE AUTOMATIC LABELING SYSTEM

Since the automatic labeling system is designed to be with extremely high accuracy, we double-checked the reliability of the system by running a long test (four days) under the supervision of the three human raters, before generating the reliable ground truth. The three raters went through the labels generated by the automatic labelling system with the help of the GUI. The GUI organizes the labels along with the discovered activity clusters, the voting percentages, the walking speeds, and the estimated postures, as shown in Fig. 10. The raters can zoom in and out to check the details or skip quickly. They can also click at one point to take a view of RGB images,

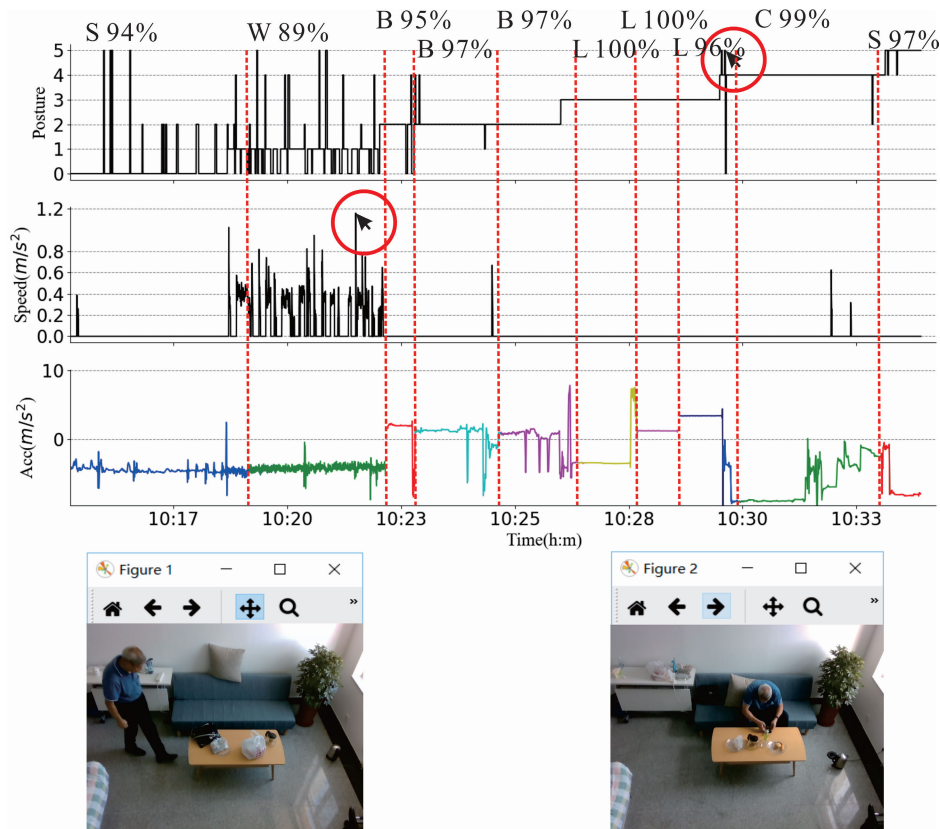


FIGURE 10. Double-check process with the GUI. The GUI organizes the labels along with the discovered activity clusters, the voting percentages, the walking speeds, and the estimated postures. Two red circles indicate two clicks, corresponding to the following two RGB images. Labels with the voting percentages are shown in acronyms: standing (S), walking (W), bending (B), lying (L), crouching (C), and sitting (S).

TABLE 3. Confusion matrix, illustrating the accuracy of the the coarse-to-fine ADLs recognition algorithm.

| | | Target Activity (%) | | | | | |
|---------------------------|-----------|---------------------|---------|---------|-------|-----------|---------|
| | | standing | walking | bending | lying | crouching | sitting |
| Output Activity (%) | standing | 93.2 | 2.3 | 0 | 0 | 2.6 | 0 |
| | walking | 0 | 94.5 | 0 | 0 | 0 | 0 |
| | bending | 2.8 | 2.2 | 92.0 | 0 | 3.6 | 0 |
| | lying | 0 | 0 | 1.3 | 100 | 1.4 | 1.3 |
| | crouching | 4 | 1.0 | 4.8 | 0 | 91.2 | 1.4 |
| | sitting | 0 | 0 | 1.9 | 0 | 1.2 | 97.3 |

to confirm the final label. In the long test, only 32 cases with lower largest voting percentages (around 74%) were reported, due to occlusion and part self-occlusion, as shown in Fig. 11. In this long test, we confirmed that they did not affect the final labels. We believe that the automatic labeling system is reliable to generate ground truth for training and testing.

D. EVALUATION ON THE COARSE-TO-FINE ADLs RECOGNITION ALGORITHM

Table 3 gives the evaluation on the coarse-to-fine ADLs recognition algorithm in confusion matrix form. It can be easily observed that, the recognition algorithm outputs promising results, accuracy over 91%. Lower accuracies are obtained in recognising bending and crouching, which are

similar to the results generated by the posture estimation. The similar results proves that it is challenging in identifying bending and crouching. As we know, older adults pay most of their time in lying and sitting, standing and walking come the second, bending and crouching the least. According to the results shown in the table, we come to a conclusion, the longer the activity duration is, the more accurate the activity recognition is. Comparing with bending and crouching, accuracy in identifying standing and walking is higher. Since recognizing these two activities is more important in modeling activity shifts among the elderly population, higher accuracy makes our ADLs recognition algorithm more valuable for implementation. The highest accuracies are obtained in recognizing lying and sitting, which are rest-activities with small motions

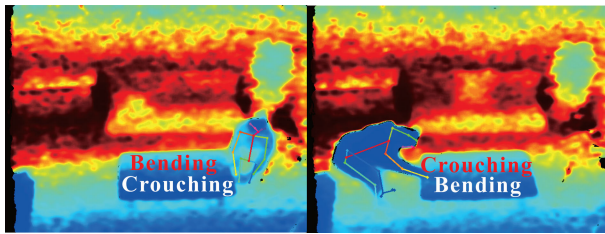


FIGURE 11. Misclassifications reported in identifying bending and crouching. Red words are the misclassifications, and white words are the manual labels.

(less energy consumption). Duration statistics of these two activities are useful for habit preference modeling.

IV. DISCUSSION AND CONCLUSION

Targeting at the challenges in ground truth generation, we propose an automatic data collecting and labeling system. An unsupervised machine learning method is employed for activity discovery, to deal with the ambulatory and trivial ADLs. Then, a multiple sensors fusion strategy is used to interpret and annotate the discovered ADLs. This system is double-checked to be reliable in a long test. The unsupervised machine learning method is the key to the automatic labeling system. Due to the ambiguous boundary in the definition of ADLs, it is impossible for human rater to discover and annotate ADLs. The unsupervised machine learning method gives us the opportunity to interpret ADLs as a whole.

The unsupervised machine learning method is also the key to the coarse-to-fine ADLs recognition algorithm, since the fragmenting activities make the ADLs modeling unrealistic. Moreover, discovering and recognising activity is more natural than recognising activity directly. Another advantage of our coarse-to-fine recognition algorithm is the high accuracy. As discovering activities into big clusters, misclassifications of several sliding windows would not affect the final detection results.

Although the automatic data collecting and labeling system is considered to be reliable, the activity discovering model does not change among people. In order to deploy personalized activity discovery and target identification, we plan to model the three key parameters of the unsupervised machine learning method associated to other sensing signals, such as heart rate and blood pressure. Moreover, we plan to work with local independent aging facilities to create a big ADLs dataset, and make it available online.

REFERENCES

- [1] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, Sep. 2018.
- [2] T. Singh and D. K. Vishwakarma, "Video benchmarks of human action datasets: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1107–1154, Aug. 2019.
- [3] G. Yao, T. Lei, and J. Zhong, "A review of convolutional-neural-network-based action recognition," *Pattern Recognit. Lett.*, vol. 118, pp. 14–22, Feb. 2019.
- [4] C. Wang, S. Chen, Y. Yang, F. Hu, F. Liu, and J. Wu, "Literature review on wireless sensing-Wi-Fi signal-based recognition of human activities," *Tinshhua Sci. Technol.*, vol. 23, no. 2, pp. 203–222, Apr. 2018.
- [5] R. G. Hussain, M. A. Ghazanfar, M. A. Azam, U. Naeem, and S. Ur Rehman, "A performance comparison of machine learning classification approaches for robust activity of daily living recognition," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 357–379, Jun. 2019.
- [6] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-Garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," *Inf. Fusion*, vol. 46, pp. 147–170, Mar. 2019.
- [7] K. Zeng, S. Ding, and W. Jia, "Single image super-resolution using a polymorphic parallel CNN," *Appl. Intell.*, vol. 49, no. 1, pp. 292–300, Jan. 2019.
- [8] B. Erol and M. G. Amin, "Radar data cube processing for human activity recognition using multisubspace learning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 3617–3628, Dec. 2019.
- [9] S. Zhu, J. Xu, H. Guo, Q. Liu, S. Wu, and H. Wang, "Indoor human activity recognition based on ambient radar with signal processing and machine learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [10] W. Li, B. Tan, and R. Piechocki, "Passive radar for opportunistic monitoring in E-health applications," *IEEE J. Transl. Eng. Health Med.*, vol. 6, pp. 1–10, 2018.
- [11] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors J.*, vol. 17, no. 2, pp. 386–403, Jan. 2017.
- [12] D. M. Sherrill, M. L. Moy, J. J. Reilly, and P. Bonato, "Using hierarchical clustering methods to classify motor activities of COPD patients from wearable sensor data," *J. Neuroeng. Rehabil.*, vol. 2, no. 1, 2005, Art. no. 16.
- [13] X. Hu, R. Dor, S. Bosch, A. Khoong, J. Li, S. Stark, and C. Lu, "Challenges in studying falls of community-dwelling older adults in the real world," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2017, pp. 1–7.
- [14] S. Mathavan, A. Kumar, K. Kamal, M. Nieminen, H. Shah, and M. Rahman, "Fast segmentation of industrial quality pavement images using laws texture energy measures and k -means clustering," *J. Electron. Imag.*, vol. 25, no. 5, Sep. 2016, Art. no. 053010.
- [15] B. R. Felton, G. L. O'Neil, M.-M. Robertson, G. M. Fitch, and J. L. Goodall, "Using random forest classification and nationally available geospatial data to screen for wetlands over large geographic regions," *Water*, vol. 11, no. 6, p. 1158, Jun. 2019.
- [16] J. Montiel, J. Read, and A. Bifet, "Scikit-multiflow: A multi-output streaming framework," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 2914–2915, 2018.
- [17] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham, "MID: Tracking and identifying people with millimeter wave radar," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 33–40.
- [18] H. C. Kam, Y. K. Yu, and K. H. Wong, "An improvement on ArUco marker for pose tracking using Kalman filter," in *Proc. 19th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, Jun. 2018, pp. 65–69.
- [19] Z. Cao, G. Hidalgo Martinez, T. Simon, S.-E. Wei, and Y. A. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
- [20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 675–678.
- [21] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.
- [22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [23] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proc. ACM Int. Conf. Multimodal Interact. (ICMI)*, 2015, pp. 443–449.
- [24] H. Gjoreski and D. Roggen, "Unsupervised online activity discovery using temporal behaviour assumption," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, 2017, pp. 42–49.
- [25] E. Stone, M. Skubic, M. Rantz, C. Abbott, and S. Miller, "Average in-home gait speed: Investigation of a new metric for mobility and fall risk assessment of elders," *Gait Posture*, vol. 41, no. 1, pp. 57–62, Jan. 2015.

- [26] D. R. Lynch, J. M. Farmer, A. Y. Tsou, S. Perlman, S. H. Subramony, C. M. Gomez, T. Ashizawa, G. R. Wilmot, R. B. Wilson, and L. J. Balcer, "Measuring friedreich ataxia: Complementary features of examination and performance measures," *Neurology*, vol. 66, no. 11, pp. 1711–1716, Jun. 2006.
- [27] L. Tong, Q. Song, Y. Ge, and M. Liu, "HMM-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1849–1856, May 2013.
- [28] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 339–367, May 2017.
- [29] X. Hu, D. Ding, and D. Chu, "Multiple hidden Markov model for pathological vessel segmentation," *BioMed Res. Int.*, vol. 2018, pp. 1–14, Dec. 2018.
- [30] S. S. C. Kon, "Habitual gait speed: A novel prognostic marker in COPD?" Ph.D. dissertation, Nat. Heart Lung Inst., Imperial College London, London, U.K., 2015.



ZHIYUAN ZHOU is currently pursuing the bachelor's degree with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. His research interests are concentrated on activity recognition and wearable computing.



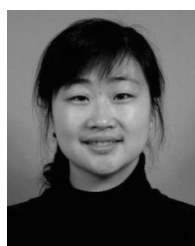
BOYANG AN is currently pursuing the bachelor's degree with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. His research interest includes wearable computing.



YONGQING HAN is currently an employee with the Office of Retirement Services, Harbin Institute of Technology at Weihai. Her research interest is concentrated on wearable computing.



CHUNMAO JIANG received the M.S. degree in software engineer from the Harbin Institute of Technology, in 2004, and the Ph.D. degree from the School of Computer Science and Technology, Harbin Engineering University. His research interests include cloud computing, embedded computing, and three-way decision.

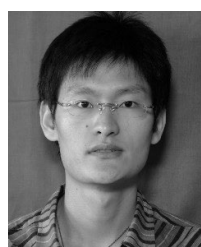


DEQIONG DING received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2011. She is currently an Associate Professor with the Department of Mathematics, Harbin Institute of Technology at Weihai. Her research interests are concentrated on nonstandard finite-difference method, numerical dynamical systems, and activity recognition.



DIANHUI CHU received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2014. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. His research interests are concentrated on fuzzy systems and knowledge discovery.

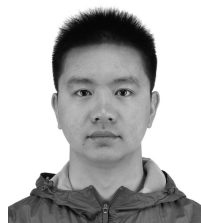
...



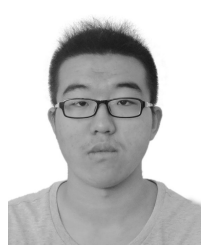
XIN HU received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2015. He is currently a Lecturer with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. His research interests are concentrated on activity recognition and analysis, wearable computing, medical image processing, and computer integrated surgical systems.



ZHENGZUO LI is currently pursuing the master's degree with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. His research interests are concentrated on activity recognition and analysis, and wearable computing.



RUIXUAN DAI is currently pursuing the Ph.D. degree with the School of Engineering, Washington University in St. Louis. His research interests are concentrated on activity recognition and analysis, and computer-aided medical diagnosis.



YANG CUI is currently pursuing the master's degree with the Department of Engineering and IT, The University of Melbourne. His research interest is concentrated on activity recognition.