Toward Detection of Driver Drowsiness with Commercial Smartwatch and Smartphone

Liangliang Lin[1,2(✉)], Hongyu Yang[1], Yang Liu[1], Haoyuan Zheng[1], and Jizhong Zhao[1]

[1] School of Computer Science and Technology, Department of Telecommunications, Xi'an Jiaotong University, Xi'an 710004, People's Republic of China
Lin_LL@126.com
[2] Information Department, Xi'an Conservatory of Music, Xi'an 710061, People's Republic of China

Abstract. In the life, there are always many objects that are unable to actively contact with us, such as keychains, glasses and mobile phones. In general, they are referred to non-cooperative targets. Non-cooperative targets are often overlooked by users while being hard to find. It will be convenient if we can localize those non-cooperative targets. We propose a non-cooperative target localization system which based on MEMS. We detect the arm posture changes of the user by using the MEMS sensors which embedded in the smart watch. First distinguish the arm motions, identify the final motion, and then perform the localization. There are two essential models in our system. The first step is arm gesture estimation model which based on MESE sensor in smart watch. we first collect the MEMS sensor data from the watch. And then the arm kinematic model and formulate the mathematical relationship between arm degrees of freedom with and the gestures of watch. We compare the results of the four actions which are important in the later model with the Kinect observations. The errors in the space are less than 0.14 m. The second step is non-cooperative target localization model that based on the first step. We use the 5-degrees data of the arm to train the classification model and identify the key actions in the scene. In this step, we estimate the location of non-cooperative targets through the type of interactive actions. To demonstrate the effectiveness of our system, we implement it on tracking keys and mobile phones in practice. The experiments show that the localization accuracy is >83%.

Keywords: Arm gesture · Non-cooperative target · Localization · Smart-watch

1 Introduction

There are many small things in life which cannot interact with people actively, such as keychains, glasses and mobile phones. They are collectively referred to as non-cooperative targets. The location of non-cooperative targets is often forgotten by users. If these targets can be located through technology, it will greatly facilitate people's life.

This work is supported by NSFC Grants No.61802299, 61772413, 61672424.

Q. Li et al. (Eds.): BROADNETS 2019, LNICST 303, pp. 227–243, 2019.
https://doi.org/10.1007/978-3-030-36442-7_15

For non-cooperative targets, people often find them by recalling where they were last used (what scene) and then recalling what they were used for (what interactions). However, there are no relevant sensors on the non-cooperative objects, and people cannot live under constant surveillance instead relying on smart devices to record their interactions with the target and guess its location. Today's smart devices are equipped with commercial MEMS sensors, including accelerometers [1–3], gyroscopes [4] and magnetic induction meters [5]. If scenes and interactions could be recorded by using MEMS sensors, it would can help people find non-cooperative targets.

Meanwhile, with the development of science and technology in recent years, people's life has become more and more intelligent. Smart devices emerge in an endless stream. A series of devices begin to enter people's lives, such as smart phones [6], including smart wristbands [7], smart watches and smart glasses [8]. In this paper, we uses smart watch to achieve the localization of non-cooperative targets. The main idea is to record the latest interaction between users and non-cooperative targets through smart watch, and then guess the location of non-cooperative targets. This paper mainly consists of two steps: the arm posture estimation and the non-cooperative target localization.

The rest of this paper is organized as follow:

Section 2 Related work

Related work is mainly about Kinect development. Kinect is an image device with high reliability that can recognize motion. In this paper, we compare the results of skeletal tracking with the results of arm posture estimation model from Kinect.

Section 3 Model Design

This section mainly realizes arm posture estimation model through the data of smartwatch MEMS sensor and non-cooperative target localization model. We take the key tracking in the opening scene and the mobile phone tracking in the calling scene to illustrate the non-cooperative target localization.

Section 4 Simulation and Experiment

This section realizes the visual simulation of the arm posture estimation model and explain the results of our system. Firstly, the arm posture estimation model is tested with common images, and then the scene of non-cooperative target is visualized, and posture changes are also tested. Finally, the localization results of non-cooperative target are tested.

We summarize this paper in Sect. 5.

2 Related Work

2.1 Kinect

Kinect is released by Microsoft which is a device that interacts with camera sensors [9–11]. It is a 3D device that can recognize objects, perform language operations, and capture objects. Kinect is first announced in 2009, and after years of research and development, Kinect has a richer set of functions and interactive tools.

Kinect relies on several core sensors for its rich functionality. The positions of Kinect infrared (IR) projector, color camera and infrared camera are indicated respectively. Depth information is collected by an infrared projector combined with an infrared camera, both of which are CMOS sensors. An infrared camera is an instrument that emits infrared light using a diffraction grating.

2.2 Bone Tracking

Simple arm movement recognition work can be achieved with Kinect. It is done by means of images. When interacting with people, Kinect use bone tracking technology to locate key parts of the body.

Bone Tracking Technology

In Kinect, human skeleton architecture can be extracted through more than 20 joints [12–14]. When the user enters the Kinect field of vision, the device can represent the user's joint position in space through coordinates (x, y, z). When the computer obtains these coordinates, it can calculate the posture of the body's main limbs in space. Kinect supports simultaneous detection of 6 people, but can only support two skeleton structures at most. In general, the user can track 20 joints while standing and 10 joints while sitting. More specifically, Kinect can provide three kinds of information: (a) the tracking status of relevant bones, only the position of the tester can be detected in passive mode, and the coordinates of the tester's 20 joints can be detected in active mode; (b) give each skeleton a test ID. This ID will be associated with several testers that can be detected to determine which one the skeleton data belongs to. (c) the specific location of the user which is actually the center of mass of the user.

Kinect Bone Tracking Development

The development of Kinect is described as follows: Firstly, it communicate with Kinect and then take the relevant data of the skeleton, so that skeletal events can be returned and detection functions can be started. Then, the bone tracking function is turned on, which processes the image information from the camera, and reads the bone data.

Secondly, data smoothing is carried out. Mutations in the data may occur while processing the data. For example, in the tracking process, some situations may lead to a large change in the detected position, which is caused by the incoherent actions of the user and the performance problems of the hardware device of Kinect, so jitter should be removed by filtering.

Finally, we transform the frame. Since the depth image data and color image data read by Kinect come from different cameras, and they face different scenes (they are in different positions), the generated images will be different, so the spatial coordinate system needs to be transformed. After that the depth information is removed from the final data and the data are drawn on the plane graph.

3 Model Design

3.1 Arm Gesture Estimation Model Based on Smart Watch MEMS Sensors

Arm gesture estimation model uses sensors of commercial smartwatches to model and estimate a series of user arm posture changes.

Kinematics Analysis Model of the Arm

First, we need to introduce the degree of freedom. In the kinematics model of the arm, the degree of freedom is the rotation angle of the joint which affects the arm posture.

There are seven degrees of freedom of the human arm. Since the smart watch is worn on the wrist, We only need to consider the motion state of the upper arm and the lower arm, so it only considers 5 degrees of freedom, including three on the shoulder and two on the elbow. We need to establish three coordinate systems to describe the posture of the arm relative to the human body. The three coordinate systems are: human body, shoulder and elbow.

This paper establishes the right hand system in the human body shoulder, namely human body coordinate system. The origin is the shoulder join. The positive direction of the X axis is from the left shoulder to the right shoulder. The positive direction of the Y axis is the back pointing to the chest. The positive direction of the Z axis is perpendicular to the other two axes. Shoulder coordinate system is the coordinate system established with the upper arm of human body as the reference point. The origin is the shoulder joint. The axis is parallel to the direction of the zupper arm, with the positive direction pointing from the shoulder to the elbow. In addition, we need to determine another axis for it. Here, when the arm is naturally drooping and the palm is facing the body, the X axis is perpendicular to the direction of the arm pointing to the human body. The elbow coordinate system is similar to the shoulder coordinate system. The origin of the coordinate system is the elbow joint, and the positive direction of the Z axis is from the elbow to the wrist, and the direction of the palm is the X axis.

Now we assume that the length of upper arm is lb, the length of lower arm is ls, the radius of lower arm is rs.

First, deduce the position of the smartwatch in elbow coordinates:

$$P_{we} = (-\mathrm{rs}, 0, \mathrm{ls}) \tag{1}$$

The subscript *we* indicates the information of the smart watches in the elbow. In order to determine the watch posture in the space, the X and Z directions of the watch's own coordinate system are introduced here as a description method of the watch posture. Two directions in the elbow coordinate system can be obtained [15]:

$$O_{we} = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} \tag{2}$$

At the same time, the vector $\overrightarrow{S_{we}}$ is introduced to describe the position and state of the smart watch:

$$\overrightarrow{S_{we}} = \begin{pmatrix} P_{we} \\ O_{we} \end{pmatrix} \tag{3}$$

Then, the state of the smartwatch in the shoulder coordinate system is deduced. Consider the rotation first, assuming that there is a transition coordinate system $\overrightarrow{O_t}$ whose origin is at the elbow, but the three axes are parallel to the three axes of the shoulder coordinate system, and the positive direction is the same. Then, according to the rotation matrix R (θ_{ef}, θ_{er}) corresponding to the degrees of freedom θ_{ef} and θ_{er}, the relationship of the state of the watch in the two coordinate systems is derived [16]:

$$\overrightarrow{S_{wt}} = \overrightarrow{S_{we}} * R(\theta_{ef}, \theta_{er})^{-1} \tag{4}$$

where $\overrightarrow{S_{wt}}$ is the state of the watch in the transition coordinate system. Then consider there is a displacement change between the actual shoulder coordinate system and the coordinate system used for the transition. So we can get:

$$\overrightarrow{S_{ws}} = \overrightarrow{S_{wt}} + \begin{pmatrix} 0 & 0 & lb \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{5}$$

Where $\overrightarrow{S_{ws}}$ is the state of the watch in the shoulder coordinate system.

Finally, the state of the smart watch relative to the human coordinate system is calculated. There is three degrees of freedom rotation between the shoulder coordinate system and the human coordinate system, then according to the rotation matrix R (θ_{sa}, θ_{sf}, θ_{sr}) corresponding to the degrees of freedom θ_{sa}, θ_{sf} and θ_{sr}, the state of the watch in the human coordinate system can be derived:

$$\overrightarrow{S_{wh}} = \overrightarrow{S_{ws}} * R(\theta_{sa}, \theta_{sf}, \theta_{sr})^{-1} \tag{6}$$

Through the above derivation process, the relationship between the position of the watch and the 5 angles can be obtained. Finally, the posture of the smart watch in the human coordinate system can be derived as follows [17]:

$$\overrightarrow{S_{wh}} = \begin{pmatrix} P_{wh} \\ X \\ Z \end{pmatrix} = \left[\begin{pmatrix} P_{we} \\ O_{we} \end{pmatrix} * R(\theta_{ef},\ \theta_{er})^{-1} + \begin{pmatrix} 0 & 0 & lb \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right] \tag{7}$$

The position of the watch almost overlaps with the wrist, so the position of the wrist can be obtained as long as the position of the watch is available. But only one point in the space (the position of the wrist) is not enough to fully describe the posture of the entire arm, but also to obtain the position of the elbow. Here, the position of the elbow is also derived. In the shoulder coordinate system, the position of the elbow P_{es} is as follows:

$$P_{es} = (0,\ 0,\ \mathrm{lb}) \tag{8}$$

The subscript es indicates the information of the elbow in the shoulder coordinate system. Then, change it into the human coordinate system. The elbow involves two degrees of freedom, θ_{sa}, θ_{sf}, which gives the position of the elbow:

$$\overrightarrow{P_{eh}} = \overrightarrow{P_{es}} * R(\theta_{sa}, \theta_{sf})^{-1} \tag{9}$$

DOF and Arm Posture Mapping Relationship

In this section, the mapping relationship between the posture of the watch and wrist position, elbow position and 5 degrees of freedom will be established on the basis of last subsection. The mapping here is one to many. That is, given a watch posture, the corresponding other information value is a few. The set of values is called the solution space.

Steps of acquire the mapping relationship between the posture of the watch and other information in body coordinate system are: First, traverse the given 5 degrees of freedom, respectively, and then calculate the values of the corresponding three keys for a given degree of freedom, namely the wristwatch, wrist and elbow position, and finally use the watch posture as the key, and other information as the value to establish a mapping relationship.

Posture Acquisition

We obtain the real posture of the smartwatch in the body coordinate system through the data of the smartwatch sensor. It mainly includes two steps: first, obtain the direction of the smartwatch in the world coordinate system, and then obtain the direction of the user in the world coordinate system, so as to obtain the posture of the smartwatch in the user coordinate system.

We get the direction of the smartwatch by combining sensors such as gyroscope, accelerometer and magnetic induction meter. The main steps are: integral calculate the real-time Angle of gyroscope, then when the watch is still, use accelerometer and magnetic induction meter to calibrate Angle obtained by integral calculation.

Next, the connection between the smartwatch and smartphone is used to determine the direction the user is facing. Data from the smartphone's accelerometer, gyroscope and magnetic induction meter are measured and calibrated to determine the user's orientation in the world coordinate system. Use the watch posture in the world coordinate system "minus" user orientation in the world coordinate system, we can obtain the watch posture in the body coordinate system.

Constraints

In the mapping relationship, each state of the watch corresponds to the multiple freedom combination of arms and the position of the smartwatch. The data of degree of freedom is relatively abstract, which is not conducive to intuitive display, while the position of the wrist (or elbow) can be clearly visualized. In this paper, the wrist (or elbow) is selected to draw the state cloud.

The state cloud which corresponds to the wrist position in the circle motion, exbtract 20 time points in the whole action, calculate the 20 time points gesture of the watch, and draw the same color graph of multiple wrist positions corresponding to each time point. State clouds of 20 time points are drawn on the same graph to form the state cloud of this action. To distinguish, the state cloud at the same moment becomes a layer of state cloud.

There are multiple wrist positions at each moment, and the estimation results on this basis will be greatly deviated. If the number of corresponding state points at each moment can be reduced, the accuracy of the results will be greatly improved. Therefore, in this subsection, constraints are introduced to compress and further optimize the solution space to make the final "estimation" result more accurate.

The constraint of solution space in this paper mainly includes the following four points:

(1) Continuity of arm posture;
(2) Arm posture is limited by human structure;
(3) Arm posture changes can be captured by sensors;
(4) The frequency of different arm posture is different.

Estimation of Arm Posture Change

In the previous subsection, we obtained the state cloud corresponding to the arm posture change. This subsection will estimate on this basis. Here, we need to select the most likely state from each layer of state cloud as the current state, and then string the most likely state points at different consecutive moments to obtain the change trajectory of wrist position over a period of time. This trajectory is referred to the trajectory line of wrist in this paper. Similarly, there are trajectory lines of elbow.

The method of estimating trajectory lines is mainly divided into two steps. The first step is to use the particle points of the state cloud for estimation directly, and the second step is to smooth the trajectory lines of state changes by a filter.

3.2 Non-cooperative Target Localization Model Based on Arm Gesture Estimation Model

This subsection will use the DOF data obtained in the previous subsection to capture the historical interaction between the target object and the user's arm, and then estimate the destination of the non-cooperative target object in the scene.

Non-cooperative Target Localization Scenes

In this subsection, the scenes involved in non-cooperative target localization will be described in detail. This includes an introduction to the scenes of the non-cooperative target and a description of the types of actions in the scene.

First of all, it is necessary to define the non-cooperative target localization scene that can be studied in this paper, that is, to explain which non-cooperative targets can be located and in which scene. The scenes in this paper contain two characteristics: 1. There are obvious scene triggering actions when someone enters the scene, so that the smartwatch knows what kind of environment it enters and what target it tracks. Here, the scene is bound to the tracking target. For example, in the opening scene, only the key is tracked; in the smoking scene, the lighter is tracked. 2. A series of interactive actions are carried out between the hand of the user wearing the smartwatch and the non-cooperative target, so that the smartwatch can estimate the destination of the target.

We picks two simple scenes for analysis.

Scene 1: Track the key chain in the opening scene.
Scene 2: Track the phone in the calling scene.

In general, the action is mainly divided into two categories in related scenes: trigger action and historical interactive action.

Trigger action: action indicates entry into a scene. When the trigger scene is detected, it means that the user has entered the scene and the watch can record the following actions. For example, the opening action in the opening scene.

Historical interaction action: after triggering the action, the user enters the scene, interacts with the non-cooperative target, and places it somewhere. The action of placing it somewhere is the historical interactive action studied in this paper. By identifying the historical interaction, we can guess the final destination of the detection target.

There are two scenes in this paper, namely, tracking the key chain in the opening scene and tracking the mobile phone in the calling scene. The trigger scene here is a door opening and a phone call.

(1) Opening action: it refers to the process of taking out the key, inserting it in the key hole and turning the key to open the door after the user arrives at the door.
(2) Call action: the user takes the phone from a certain location and holds it to his ear to hold the call.

There are two reasons for dividing actions into trigger actions and interactive actions. First, trigger actions represent a boundary that defines which actions will be related to non-cooperative goals of interest. The action after the trigger action is needed to predict the action of the non-cooperative target position. Then, the sampling rate of the system can be controlled. The strategy we adopt is to collect a small amount of data at the ordinary sampling rate for analysis to determine whether the scene is entered. If the scene is entered, the strategy of high sampling rate will be adopted.

The scenes in this paper are not complicated, and the historical interactive actions mainly include: putting them into coat pockets, putting them into trouser pockets, putting them on the table, and throwing them onto some objects nearby (such as sofa, bed, etc.). In addition to the four types of actions related to where the target is going, there are also some unrecognizable actions. These 5 types of actions are collectively referred to as historical interactive actions. They are: load top A1, load bottom A2, put on top of something A3, throw out A4, and other actions A5.

Action Segment Segmentation

This subsection will segment the action based on the data of the variation of the freedom of the arm. In the process of moving arms, the movement tends to move from one resting position to another, rather than moving at any given time. If these periods of rest throughout the moving progress can be found, then we can segment the action. The specific approach is as follows: first, track the rest point, extract a short period of sliding window. In this window, if a few degrees of freedom change is detected to be very small, then this segment can be used as a candidate set of rest points. Then the data changes between the rest points are analyzed. Data changes between rest points need to exceed a certain threshold to prevent small changes from affecting the final results. The non-conforming rest points will be excluded. Through the above algorithm, one action segment can be obtained.

Identification of Key Actions

When the action segment is obtained, each action segment can be identified. We will identify key actions in three steps.

Firstly, feature extraction is carried out to extract the most closely related parameters of posture change, and then dimensionality reduction of redundant features is carried out. Finally, the classifier's parameters are obtained by training the eigenvector.

In order to tell which, action each segment is, features need to be extracted. This paper makes use of the following features, which will play a key role in the later action recognition:

(1) mean

The mean is the mean of the signals. The mean value can reflect the approximate fluctuation position of signal, and its calculation formula is as follow:

$$\mu_x = \frac{1}{N}\sum\nolimits_{n=0}^{N-1} x_n \tag{10}$$

(2) variance
The variance can represent the fluctuation of the whole data above and below the mean value. Its calculation formula is as follow:

$$\sigma_X^2 = \frac{1}{N}\sum\nolimits_{n=0}^{N-1} |x_n - \mu_x|^2 \tag{11}$$

(3) kurtosis
Kurtosis represents the peak value of the image at the average value. Intuitively, it represents the height of the image tip. Its calculation formula is as follow:

$$\frac{\sum_{n=1}^{N}(x_n - x)^4}{(N-1)s^4} \tag{12}$$

(4) signal power RMS
Here RMS is actually the square root of signal power. Signals carry energy, and signal power is a measure of the energy carried by a signal. Its calculation formula is $\sqrt{P_x}$.

Where the * calculation is as follow:

$$P_x = \frac{E_x}{N} = \frac{1}{N}\sum\nolimits_{n=0}^{N-1} |x_n|^2 \tag{13}$$

We select 5 features, which means we need to reduce the dimension of the 25 dimensional vector corresponding to 5 degrees of freedom.

In this paper, principal component analysis (PCA) is used to realize data dimension reduction. PCA is an important algorithm in statistics, which is mainly used to reduce redundancy and extract main features. PCA can reduce the dimension of feature in the last subsection greatly. The next step is to use these features for identification.

The types of actions have been described in the previous subsection. What need to be identified here are the trigger actions such as opening the door, answering the phone, and the actions such as loading top, loading bottom, putting on something, and throwing out. Opening the door and answering the phone are identified as a group, and the last four actions are identified as a group, but the recognition method is universal.

This paper uses support vector machine (SVM) for classification. This is a multi-classification problem. For example, in the first group, three categories of actions, such as opening the door, answering the phone and other actions, need to be distinguished. In this paper, one-to-one support vector machine model is used. It trains the support vector machine model for any two kinds of actions in the same group and finally judges the categories according to the voting results of all the models.

Non-cooperative Target Localization
In the previous section, several actions that need to be recognized are identified by SVM. In this section, the location/destination of non-cooperative targets is tracked according to the recognition results in the previous subsection.

In this paper, Fig. 1 is used to represent the relationship between motion recognition and the location/destination of non-cooperative targets (i.e. keys and mobile phones). In Fig. 1, Scene judgement identifies which scene it is, and the location is determined by the action, including clothes, table/cabinet, sofa/bed. In this paper, the smartwatch is first used to collect data for action judgment to determine whether a scene is entered. After judging the scene, the following actions can be judged, and these historical interactive actions will be directly linked to the location of the non-cooperative target.

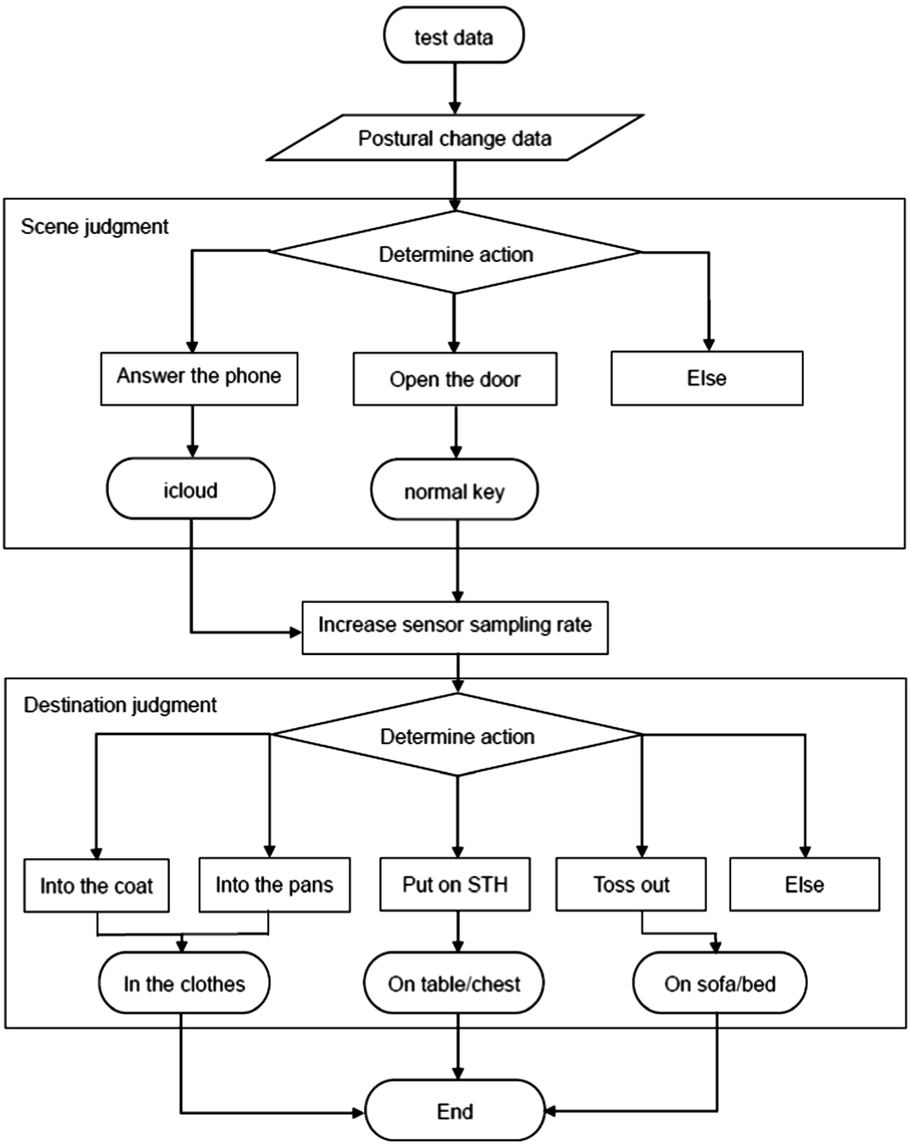

Fig. 1. Non-cooperative target location flow chart

So far, this paper has connected the data of arm posture change, classification model and the location of non-cooperative targets together, so as to realize the localization of non-cooperative targets in the scene.

4 Simulation and Experiment

In Sect. 3, the arm posture estimation model based on smartwatch MEMS sensor and the non-cooperative target model based on arm posture estimation are respectively established. This section will carry on the simulation, the experiment, and carry on the analysis to the result.

4.1 Analysis of Arm Posture Estimation Model Results

This subsection will analyze the reliability and accuracy of the arm posture estimation model. This paper will compare the results of arm posture estimation model based on smartwatch with the results of Kinect.

The whole testing process is carried out in the laboratory environment. A total of 5 participants were invited to test each action multiple times. The testing process mainly includes 3 steps: collecting data of smartwatch, collecting images of Kinect device, and comparing the calculated results of the computing platform with the results of Kinect.

Smartwatch data collection: the tester wears the LG G Watch on the left wrist, with the surface facing the same direction as the back of the hand; The tester stands in a room with few electronic devices (to reduce the effect of magnetic fields) and faced the Kinect camera to perform the relevant action tests.

Calculation results of the platform are compared with the results of Kinect: after collecting all sensor data of the smartwatch, it is transmitted to the computing platform; Calibrate and process sensor data of smartwatch on Matlab, and obtain relevant parameters; The results of relevant parameters were compared with the data collected by Kinect in Matlab.

This paper selects four actions for verification, namely circle, “S” curve, straight line and square. Take a set of test data of drawing circle for display. The tester wears the watch and draws an arbitrary circle in the space. Figure 2 is the trajectory line of the elbow and wrist corresponding to the circle drawing, where the figure on the left is the result of estimation and the figure on the right is the result from Kinect.

In Fig. 2, you can see that the right and left trajectories have the same trend, although they differ in subtle shapes. In the later part of this paper, when locating the non-cooperative target, it is enough to identify the action itself as long as the trend is consistent, without requiring the coordinate points to be exactly identical.

Here, the errors in the circle drawing process were counted, and the cumulative distribution errors of the circle drawing results were randomly counted for 5 times respectively.

The errors are calculated on three axes, and the results are shown in Fig. 3. It can be seen that the wrist error and elbow error are respectively 0.27 m and 0.15 m, the accuracy is well guaranteed. When drawing circles, the wrist error is larger than the elbow error, because when doing the action, the wrist motion range is larger than the

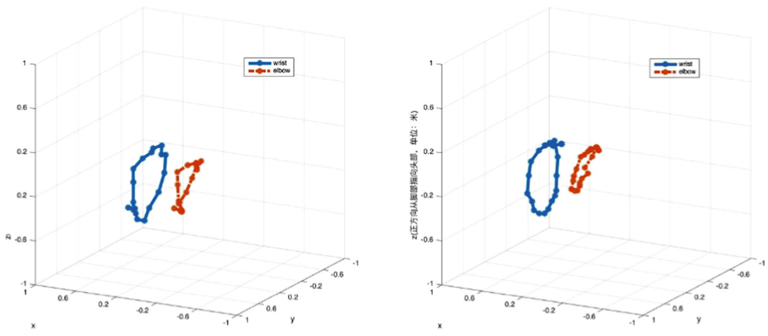

Fig. 2. Model estimation results and Kinect observation results of drawing circle trajectory line (unit: meter)

elbow, the error has such a relationship is not surprising. And Z axis error than the other two axes error is large, it is related to the error of Kienct in depth.

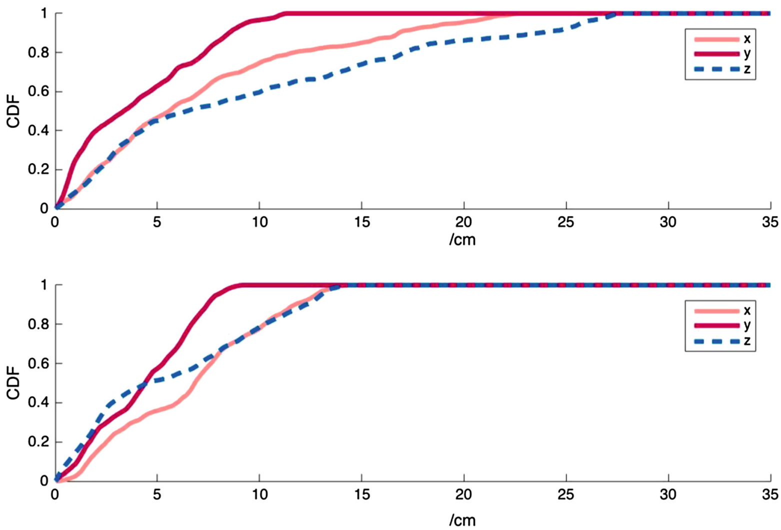

Fig. 3. Cumulative distribution of wrist and elbow position errors in circle drawing

4.2 Visualization and Recognition Results of Non-cooperative Target Locating Scene

In this subsection, the scene of non-cooperative target locating will be specified to make it visible, and the rationality of the result can be judged directly through the visualization results of the model. This subsection will examine the door opening and phone calling scenarios, as described in Sect. 3.

Opening the Door Scene

Compare the estimated results of the smartwatch with the results of Kinect, as shown in Fig. 4. In Fig. 4, blue is the track line of the wrist and red is the track line of the elbow. It can be seen that the trend of the two segments of track lines remains consistent, and the errors of the whole process on the three axes are 0.12 m, 0.14 m and 0.14 m respectively. The estimated results of the smartwatch on the left clearly show the process of turning a key or a door handle, while Kinect's description of the process is not accurate enough. This is because when the wrist and elbow are at the same height, the key points of the wrist and elbow are next to each other in the bone tracking, and the interaction between the two makes it impossible to describe the state in detail.

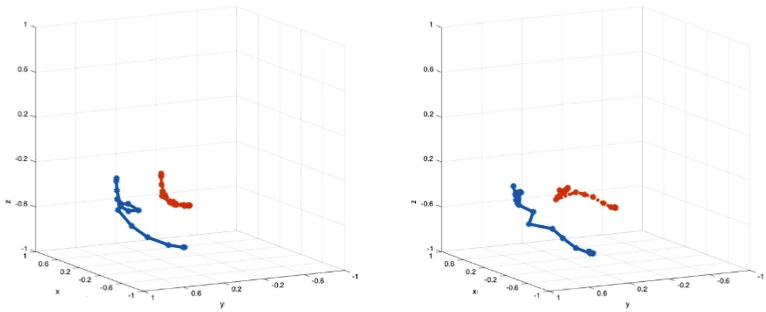

Fig. 4. Comparison figure of track lines of two devices (unit: meter) (Color figure online)

Calling Scene

From Fig. 5, it can be seen that the track line of the whole phone call is relatively simple, and Significant stagnation can be seen at the top of the trajectory line, which corresponds to the state of the phone at the ear. Compared with the results of Kinect, the errors of the three axes were 0.11 m, 0.13 m and 0.15 m respectively.

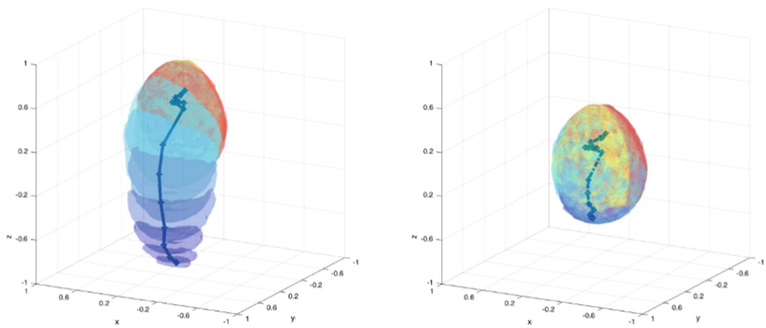

Fig. 5. The trajectory of the wrist and elbow during a phone call (unit: meter)

Historical Interactive Action

The results of the historical interaction actions involved in the following scenes are described here. Several related historical interactions have been illustrated in the previous section, they respectively are load top, load bottom, put on something and throw. It's also necessary to show the accuracy of their recognition, to prove later non-cooperative target test is conducted on the basis of the accuracy of guarantees. The results are shown in the following table (Table 1):

Table 1. The result of historical interactions

Action	Average wrist distance			Average elbow distance		
	X	Y	Z	X	Y	Z
Into the coat	0.10	0.08	0.14	0.14	0.09	0.12
Into the pants	0.09	0.09	0.15	0.09	0.07	0.14
Put on STH	0.04	0.12	0.14	0.11	0.08	0.12
Toss out	0.08	0.09	0.12	0.06	0.06	0.09

It can be seen from the above table that smartwatches have high accuracy in identifying the following actions. Next, this paper will use the data of 5 degrees of freedom changes obtained by the estimation model to locate the non-cooperative targets.

4.3 Results Analysis of Non-cooperative Target Locating

Test Preparation

This section describes the preparation of a test for a non-cooperative target location. It describes the entire test process and details.

The whole testing process of this paper is carried out in the laboratory environment. The main points include two steps: the smartwatch collects data, and the computing platform trains the data.

Data collection of smartwatch: the experiment invited 5 testers to perform 70 times of 2 trigger actions and 4 historical interactive actions respectively. They also did 70 other random actions. So there are seven types of actions, and each type of action has 350 sets of data. The LG G Watch was worn on the left wrist in the same direction as the back of the hand.

The computing platform trains the data: the collected data are preprocessed, reduced and put into model training on the computing platform, and the parameters are calculated to distinguish the actions. For 7 types of actions, 250 groups are randomly selected from 350 groups of data as the training set and 100 groups as the test set. During the extraction of these 250 groups, the data of each tester should be distributed evenly, so as to facilitate later analysis.

Each of the 5 participants was invited to perform 40 door opening and phone call simulations. That is 200 sets of complete test data for each scene, a total of 400 sets of data.

Test Results and Analysis

This subsection will process 350 sets of data for each of 7 types of actions. After dividing the data into training set and test set, the training set is used to train the SVM model, and then test the data and explain the results.

First, 350 groups of data were randomly assigned. Using random Numbers, 250 groups were selected as the training set and 100 groups as the test set. Each group corresponds to 7 actions, namely 7 pieces of data. Here, the tester is required to make a single action each time when collecting data, so as to avoid the workload of dividing the action segments and directly conduct follow-up training.

Then, the arm posture estimation model is used to obtain the changes of the 5 degrees of freedom of these 350 sets of data. The input of non-cooperative target locating is the change data of 5 degrees of freedom. Next, it is necessary to extract features from the data of these degrees of freedom changes and conduct dimensionality reduction processing on these features.

Finally, the training set is put into the support vector machine to train, record the parameters, and use the test set to verify the accuracy and recall rate of the model. Here, 3 support vector machines are trained by using door opening data, phone call data and other movement data, and 10 support vector machines are trained by using 4 movements and other movement data. The final test results are shown in the following table:

Table 2. Action test results (unit: %)

Action	Open the door	Answer the phone	Into the coat	Into the pants	Put on STH	Toss out	Else
Open the door	93	2					5
Answer the phone	2	91					7
Into the coat			95	1	1	0	3
Into the pants			2	91	3	0	4
Put on STH			0	3	88	3	6
Toss out			1	2	2	87	8

Where the columns represent the actual categories of data and the rows represent the categories assigned. Among them, opening the door, answering the phone and other actions are the first group, indicating the distinction of scenes. Putting in coat, putting in trousers, putting it on something, throwing it out and other actions are in the second group to indicate the division of direction. You can see that the classification has good accuracy, which is mainly related to the complexity of the scene. Among them, more than 87% of the samples were correctly judged, and the misjudged movements were mainly other movements, but the proportion was relatively small. Next, the whole process will be tested with the complete scene test data.

It can be seen that the accuracy of Table 3 is lower than that of Table 2, mainly because the overall error is jointly determined by the error of scene judgment and the error of direction judgment. Among them, the main reason for wrong judgment results is that the other actions of the tester are casual, and some actions are similar to the key actions concerned in this paper, which leads to the inaccurate classification results of the classifier. However, the final accuracy rate was more than 76% and the overall accuracy rate was 83%, which enabled the estimation of the destination of non-cooperative targets.

Table 3. Scene test results (unit: %)

Action	Open the door				Answer the phone			
	Into the coat	Into the pants	Put on STH	Toss out	Into the coat	Into the pants	Put on STH	Toss out
Accuracy rate	90	86	84	80	88	82	78	76

5 Conclusion

This paper realizes the locating of non-cooperative targets based on smartwatch, which mainly includes the arm posture estimation model and the non-cooperative target locating model.

This paper verifies and tests the model, and the results are as follows:

(1) arm attitude estimation model results

In the experimental part of this paper, 5 testers are invited to draw 4 kinds of images in the air, and the triaxial error is no more than 0.30 m. This paper also identifies the scene actions that need to be involved in the locating of non-cooperative targets, and the error is no more than 0.14 m.

(2) results of non-cooperative target locating

In this paper, 5 testers were invited to perform a total of 350 times of 7 types of actions involved in the scene, 7*250 groups of data were used as the training set, and 7*100 groups of data were used as the test set, with the accuracy of action is more than 87%. In this paper, the testers demonstrated a total of 400 scenes, and the average accuracy of non-cooperative target locating is no less than 83%.

We believe that the model in this paper will be applied to more scenes in the future.

References

1. Nelson, E.C., Verhagen, T., Noordzij, M.L.: Health empowerment through activity trackers: an empirical smart wristband study. Comput. Hum. Behav. **62**, 364–374 (2016)
2. Li, S.Y., et al.: An exploration of fall-down detection by smart wristband. Appl. Mech. Mater. **687–691**, 805–808 (2014)
3. Cheng, L., Shum, V., Kuntze, G., et al.: A wearable and flexible Wristband computer for on-body sensing. In: 2011 IEEE Consumer Communications and Networking Conference (CCNC), pp. 860–864. IEEE (2011)
4. Al-Nasser, K.: Smart watch: US, US8725842 (2014)
5. Rauschnabel, P.A., Brem, A., Ivens, B.S.: Who will buy smart glasses? Empirical results of two pre market-entry studies on the role of personality in individual awareness and intended adoption of Google Glass wearables. Comput. Hum. Behav. **49**(8), 635–647 (2015)
6. Ji, S., Xu, W., Yang, M., et al.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221 (2013)
7. Lei, J., Li, G., Zhang, J., Guo, Q., Dan, T.: Continuous action segmentation and recognition using hybrid convolutional neural network-hidden Markov model model. Comput. Vis. IET **10**(6), 537–544 (2016)
8. Martínez, F., Manzanera, A., Romero, E.: Spatio-temporal multi-scale motion descriptor from a spatially-constrained decomposition for online action recognition. Comput. Vis. IET **11**(7), 541–549 (2017)
9. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 357–360 (2007)
10. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 1395–1402, 17–20 October 2005 (2005)
11. Csurka, G., et al.: Visual categorization with bags of keypoints. In: ECCV (2004)
12. Ogata, M., Imai, M.: SkinWatch: skin gesture interaction for smart watch. In: Proceedings of the 6th Augmented Human International Conference, pp. 21–24. ACM (2015)
13. Laput, G., Xiao, R., Chen, X.A., Hudson, S.E., Harrison, C.: Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, Honolulu, Hawaii, USA, 05–08 October 2014 (2014)
14. Weigel, M., Mehta, V., Steimle, J.: More than touch: understanding how people use skin as an input surface for mobile computing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Toronto, Ontario, Canada, 26 April–01 May 2014 (2014)
15. Parate, A., Chiu, M.C., Chadowitz, C., et al.: RisQ: recognizing smoking gestures with inertial sensors on a wristband. In: International Conference on Mobile Systems, MobiSys, p. 149 (2014)
16. Xu, C., Pathak, P.H., Mohapatra, P.: Finger-writing with smartwatch: a case for finger and hand gesture recognition using smartwatch. In: International Workshop on Mobile Computing Systems and Applications, pp. 9–14. ACM (2015)
17. Komninos, A., Dunlop, M.: Text input on a smart watch. IEEE Pervasive Comput. **13**(4), 50–58 (2014)