# HandButton: Gesture Recognition of Transceiver-free Object by Using Wireless Networks

**Dian Zhang [1] and Weiling Zheng [2]**
[1] College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
[e-mail: Serena.dian@gmail.com]
[2] College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
[e-mail: zwlcle@hotmail.com]
*Corresponding author: Dian Zhang

## Abstract

Traditional radio-based gesture recognition approaches usually require the target to carry a device (e.g., an EMG sensor or an accelerometer sensor). However, such requirement cannot be satisfied in many applications. For example, in smart home, users want to control the light on/off by some specific hand gesture, without finding and pressing the button especially in dark area. They will not carry any device in this scenario. To overcome this drawback, in this paper, we propose three algorithms able to recognize the target gesture (mainly the human hand gesture) without carrying any device, based on just Radio Signal Strength Indicator (RSSI). Our platform utilizes only 6 telosB sensor nodes with a very easy deployment. Experiment results show that the successful recognition radio can reach around 80% in our system.

## 1. Introduction

**G**esture recognition usually aims to interpret various human gestures, e.g., the body motion usually originated from the hand or face. It attracts many researchers' attention, since it is very useful in many applications to understand body language.

Traditional technologies usually utilize computer vision algorithms to interpret the body language [4] [5] [6] [7]. However, they usually cannot work in dark area due to the light requirement of the surrounding cameras. Thus, these technologies are limited in some specific enviorments, e.g., in the dark area. Although some other technologies are able to recognize the human gesture in dark area, most of them require the target to carry an EMG sensor or an accelerometer sensor [1] [2]. Thus, it is still a big limitation for people to use. Considering the following scenario, when a person comes into a dark room, instead of trying to find the control button of the light in the dark, he/she wants to use his/her hand motion to turn on/off the light to without carrying an additional device. Such similar requirements are very important in many applications (e.g., the smart home) to control any electronic device. Traditional technologies usually cannot satisfy such requirements.

In order to fill the gap between traditional technologies and real-world requirements, we propose HandButton, which is able to recognize human gesture without carrying any device in such scenario. The basic idea is to utilize the signal dynamic information (containing their occurrence time) caused by the human body (e.g., the hand) to detect the gesture. In detail, some wireless nodes are deployed in advance in a specific area (e.g., the door area). Each node acts as both transmitter and receiver. Thus, there are many wireless links among these nodes. When a target gesture is performed, the Radio Signal Strength Indicator (RSSI) value of some wireless links will change. We referred the, as signal dynamics. Since the occurrence time of such signal dynamics are different for those wireless links, we utilize such difference to decide the target gesture.
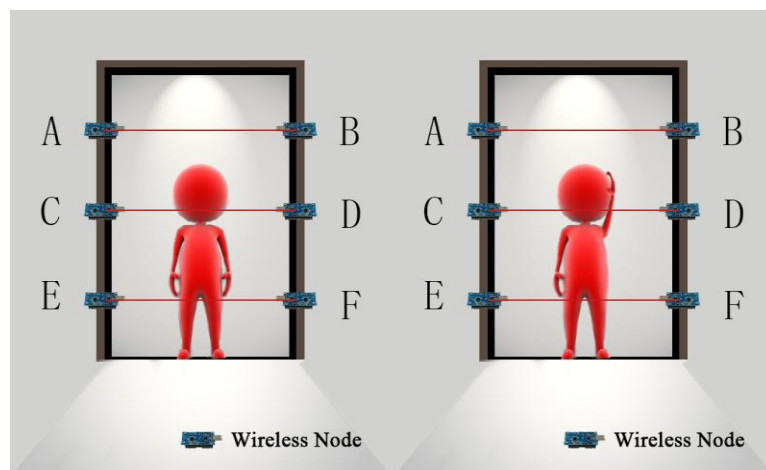


**Fig. 1.** An example of gesture recognition

**Fig. 1** gives an illustration of our idea, which contains six nodes marked *A*, *B*, *C*, *D*, *E* and *F*. The number of deployed node may vary according to different applications. Node *A*, *C* and *E*

are placed on the left side, while the other nodes are placed on the right side. Each node acts as both transmitter and receiver. When the target is motionless, as shown in the left part of **Fig. 1**, the RSSI value of each wireless link will be stable. When the target performs a gesture, the RSSI value of some wireless links will change. For example, in the right part of **Fig. 1**, when the target moves his/her hand from the top to down, the RSSI value of link *AB* usually will be first affected. Then it will be the other links, e.g., *CD* and *EF*. Our approach aims to recognize the target gesture without carrying any device in some specific scenarios, e.g., in the smart home to control the light on/off. It is easy to be extended to other applications.

Our HandButton system has three algorithms to determine the target gesture: Peak-Time, the Best-Fit and Dynamic Difference Algorithm. The first one leverages the time with the largest signal dynamic value of each wireless link, to determine the target gesture. It is convenient for users to apply. The second one utilizes localization algorithm to detect the target (e.g., the hand) trace, the target gesture is able to be derived. It has higher accuracy and can give more information to the gesture. The last one utilizes signal dynamic diffference of each pair of wireless links to decide the target gesture. It has higher accuracy and able to eliminate the noise behavior in the environment. Our experiments are based on 6 telosB sensors [3]. Experimental results show that the successful recognition radio can reach around 80% and the latency is only about 0.4*s*, which show immense potential in future applications.

The main contributions of this paper are as follows. First, we are able to recognize target gesture without carrying any device, and it can be applied in dark area. It eliminates the environment limitation of traditional technologies. Second, we propose HandButton, which contains three recognizing algorithms able to accurately recognize the target gesture. Third, we conduct real experiments and comprehensive analysis in this area. At last, the cost is very low and the deployment is easy.

The rest of the paper is organized as follows. In the next section, we will discuss the related work about traditional technologies. In the following, we will introduce our three recognizing algorithms. Section 4 will show the experiment setup and results. We will conclude the work in the last section. Some possible future work directions are also listed in the last section.

## 2. Related Work

Basically, there are two kinds of gesture recognition technologies: Non-radio based technologies and Radio based technologies.

### 2.1 Non-radio Based Technologies

In Non-radio based technologies, video technologies are very popular. Video technologies [4] [5] use image processing algorithms to recognize the target gesture. In the computer vision area, the gesture, especially the extremities and head gesture can be even recognized in 3-D by using multiple cameras [6]. The gesture can also be recognized by using RGB-D data from a Kinect sensor [7], it created the skeletal model from the depth data and the extractd the frame-level features from RGB frames. The depth data should be classified by multiple extreme learning machines. Although these technologies have high accuracy, they cannot be applied in the dark area. Thus, the application scenario is limited. New technologies, such as Kinect, have made a breakthrough in darkness. However, they should take a lot of training to recognize a gesture. On the other hand, the high cost prohibits the widespread use of these equipments. Moreover, they have major concerns on privacy and energy consumption.

## 2.2 Radio Based Technologies

Traditional radio-based technologies mainly use EMG sensor or Accelerometer sensor in recognition [8]. The gesture can be recognized based on the input signals from accelerometers [9]. EMG sensors measure the electrical activity produced by the muscles to recognize the gesture [10]. However, these technologies all require the target to carry a device (e.g., an EMG sensor or an accelerometer sensor).

There are some other technologies using WiFi [11], which doesn't require the target to carry a device. Most WiFi technologies use MIMO [12] technologies and USRP [13] to estimate the human gesture. However, they require devices with multiple antennas and should get physical layer information. It is a limitation to most common device. Our algorithms have no such requirements and they can be widely for almost all the wireless device.

# 3. Methodology

In this section, we will introduce three algorithms to recognize target gesture: Peak-Time, Best-Fit and DDA. The basic idea behind the three algorithms is to detect when the wireless link signal be affected by the target gesture, then derive the gesture direction from the time difference between links. The wireless signal is measured by RSSI information. If in a *static environment* where no object moves around and the target object is motionless, the RSSI value of each wireless will be stable. If in a *dynamic environment* where the target performs some gesture, the RSSI value of some wireless link will change (referred as RSSI dynamics). Those changed wireless links plus their change time will be leveraged to derive the gesture.

## 3.1 Peak-Time Algorithm

This algorithm is able to be applied in different node topologies and recognize simple target gestures. In this algorithm, we may recognize gesture $G_1$ (wave hand from top to down) and $G_2$ (wave hand from down to top), the other gestures in other scenarios can be easily derived accordingly. Without loss the generality, we introduce this algorithm based on the grid setting of the nodes.
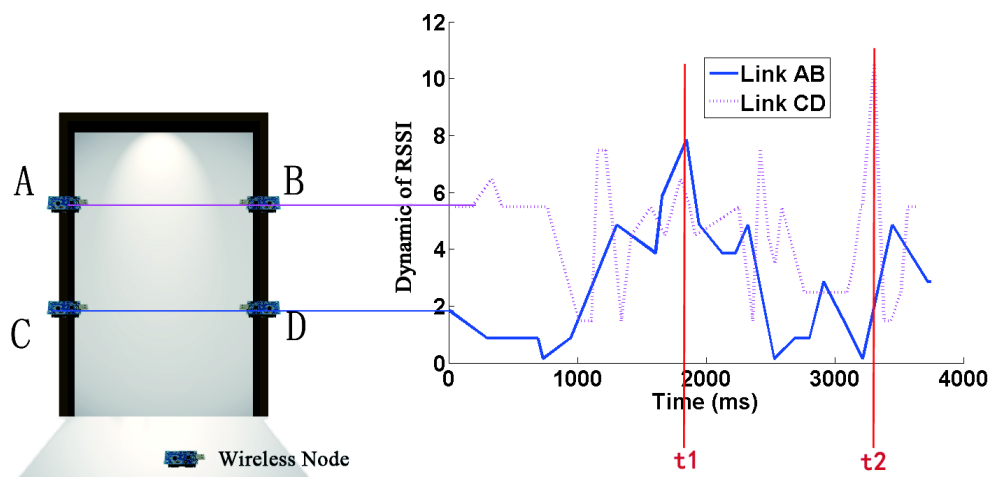


**Fig. 2.** An example of Peak-Time algorithm

Suppose the gesture recognition area is a door area, as shown in **Fig. 2**. In this example, we have 4 nodes deployed in this area. Each node will act as both transmitter and receiver. Therefore, in total we have 6 wireless links (we regard the symmetric links as one link). In this algorithm, we only utilize the horizontal wireless link of each pair of nodes (in the example, they are link *AB* and *CD*). For each horizontal wireless link, when the target is motionless, the RSSI value of each horizontal wireless link is stable. When the target gesture occurs (e.g., a human gesture from down to up), link *CD* usually will be affected first. Its RSSI value will change first. As shown in the right part of **Fig. 2**, the RSSI value of link *CD* changes dramatically. The change is the largest at time $t_1$ (referred to Peak-Time). The RSSI value of link *AB* changes dramatically with the largest at time $t_2$. Also we may find that, the time $t_1$ is earlier than time $t_2$. We may conclude that the gesture is $G_2$ (from down to up). In order to get higher accuracy, we may deploy more nodes for redundancy. How many number nodes should be used in our system will be discussed in the later section.

In general, for each horizontal wireless link *ij* between node *i* and *j*, its RSSI value in static environment is $r_{ij}$. Suppose

$$\bar{R}_{ij} = \frac{\sum_1^N r_{ij}}{N} \tag{1}$$

where $\bar{R}_{ij}$ is the average RSSI value of *N* RSSI static value obtained from the horizontal wireless link *ij*.

In dynamic environment, when the target gesture occurs, the RSSI dynamic value of each horizontal wireless link *ij* between the node *i* and *j* is $D_{ij}$. It is noted as $d_{ij}$, which is the RSSI difference between static environment and dynamic environment.

$$d_{ij} = \left| D_{ij} - \bar{R}_{ij} \right| \tag{2}$$

For each horizontal wireless link *ij* between node *i* and *j*, we will calculate its peak time $T_{ij}$ when the value of $d_{ij}$ is the largest. It can be expressed as

$$T_{ij} = T_{\max(d_{ij})} \tag{3}$$

Therefore, the target gesture can be derived by all the peak time of all the horizontal wireless links. Among the whole set, for each two horizontal wireless links *ij* between node *i* and *j* and link *uv* between node *u* and *v*, suppose their Y axis-coordinates are $y_{ij}$ and $y_{uv}$. Their corresponding peak times are $T_{ij}$ and $T_{uv}$, respectively. We will decide the gesture G for link *ij* and *uv* as follows.

$$G = \begin{cases} G_1 & \text{if } (y_{ij} > y_{uv} \text{ and } T_{ij} > T_{uv}) \text{ or } (y_{ij} < y_{uv} \text{ and } T_{ij} < T_{uv}), \\ G_2 & \text{if } (y_{ij} > y_{uv} \text{ and } T_{ij} < T_{uv}) \text{ or } (y_{ij} < y_{uv} \text{ and } T_{ij} > T_{uv}), \\ F & \text{if } (T_{ij} = T_{uv}). \end{cases} \tag{4}$$

where *F* means the failure of the gesture recognition.

If the number of horizontal wireless links *n* is more than two, we will have $C_n^2$ pair of horizontal wireless links (for example, if there are 3 horizontal wireless links, then we will have 3 pair of those links). Applying Eq. (4) to these $C_n^2$ pair of horizontal wireless links,

suppose there are $l$ number of gesture decision falls into gesture $G_1$, while $k$ number of gesture decision falls into gesture $G_2$, here $l + k \leq C_n^2$. By using the Equ.4, we will have the final gesture decision $G_{final}$ as

$$G_{final} = \begin{cases} G_1 & if \ l > k, \\ G_2 & if \ l < k, \\ F & if \ l < k. \end{cases} \tag{5}$$

Actually the Peak-Time algorithm is easy to be extended to well recognize other gestures, only if the target gesture crosses the line-of-sight link of some links. For example, we may deploy the node in different topologies, e.g., the star topology. In general, the gestures to be recognized are depended on the node topologies. Only if the target able to cross the line-of-sight link of each pair of nodes, the corresponding gesture can be well recognized by this algorithm. Considering our scenario in the door area to control the light, we only introduce the method to recognize two gestures. the other gestures in other scenarios can be easily derived accordingly.

## 3.2 Best-Fit Algorithm

Previous Peak-Time algorithm only can decide some basic target gestures and the accuracy depends on the node deployment. If users want to get more accurate information about the target gesture, they may use the Best-Fit algorithm. The basic idea behind this algorithm is to utilize localization algorithm to get the trace of target gesture. Then the gesture behavior is able to be decided.

According to the model of our previous research, for each wireless link, if the target is closer to the center of the link, the RSSI value will change more. Therefore, for each wireless link, once its RSSI value changes, we are able to estimate the possible target area for such link, which is able to be presented by a rectangle. The length and the width of the rectangle can be obtained from our previous model [14].
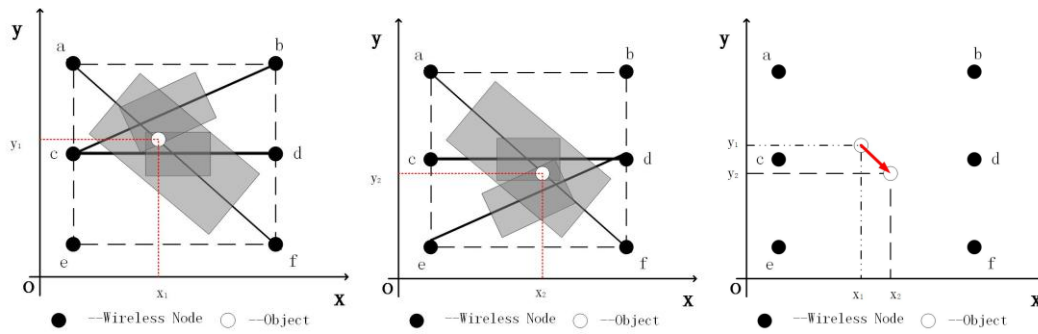


**Fig. 3.** An example of Best-Fit algorithm

The basic idea of our Best-Fit algorithm is illustrated in **Fig. 3**. As **Fig. 3** shown, we have deployed six nodes in a door area. Each node will act as both transmitter and receiver. Therefore, in total we have 15 wireless links (we regard the symmetric links as one link). For each wireless link, we measure the RSSI dynamic value, which is the RSSI difference between static environment and dynamic environment, as introduced before. If the RSSI dynamic value is larger than zero, we will estimate the possible target area for this link, as the rectangle for

link $cd$, $af$ and $cb$, colored in grey. A larger RSSI dynamic value will cause a smaller rectangle area but with a large weight. These rectangles may be overlap as shown in **Fig. 3**.

In the following, every fixed interval (in our experiment this value is 200$ms$), we calculate the intersection points of the rectangles. The estimated target position is the weight average position of these intersection points. As shown in **Fig. 3**, position $(x_1, y_1)$ is the first estimate result. Position $(x_2, y_2)$ is the second estimate result. So we may conclude that the target trace is from $(x_1, y_1)$ to $(x_2, y_2)$. Since $y_1 > y_2$, we may decide the gesture belongs to $G_1$. Moreover, it can give more information.

In general, suppose we have $m$ nodes. Therefore, we have $C_m^2$ wireless links. As mentioned before, the RSSI dynamic value of each wireless link $d_{ij}$ can be calculated from Eq. (1) and Eq. (2) (here we consider all the wireless links, not just the horizontal links). Each $d_{ij}$ creates a rectangle area, in which the object body is likely to reside. For each rectangle area of link $ij$, its corresponding rectangle area is identified as $A_{ij}$. The weight value $w_{ij}$ of $A_{ij}$ is calculated by the follow formula

$$w_{ij} = d_{ij} + \sum_{u,v=1}^{m} \frac{A_{ij} \bigcap A_{uv}}{A_{uv}} \times d_{uv} \, (\text{u} < \text{v}) \tag{6}$$

here $A_{uv}$ is the area of each rectangle overlapping with $A_{ij}$.

We choose the top $K$ rectangles with the largest weight values (in our experiment we choose this values as 5) and calculate their intersection points. Then we calculate their average position of these intersection points as the target location $(x_s, y_s)$. Such location estimation will be repeated every fixed interval (in our system this value is 200$ms$). For a two adjacent estimated target location $(x_s, y_s)$ and $(x_{s+1}, y_{s+1})$, we may decide the gesture $G$ as follows.

$$G = \begin{cases} G_1 \; if \; y_{s+1} - y_s < 0, \\ G_2 \; if \; y_{s+1} - y_s > 0, \\ F \; if \; y_{s+1} - y_s = 0. \end{cases} \tag{7}$$

If the number of gesture decision $p$ is more than two. Suppose there are $r$ number of gesture decision falls into gesture $G_1$, while $q$ number of gesture decision falls into gesture $G_2$, here $r + q \leq p$ , and the final gesture decision $G_{final}$ will be generated by following formula.

$$G_{final} = \begin{cases} G_1 \; if \; r > q, \\ G_2 \; if \; r < q, \\ F \; if \; r = q. \end{cases} \tag{8}$$

This Best-Fit algorithm can not only give the gesture decision, but also can give more information about the target trace. As shown in the **Fig. 4**. Moreover, the topology with more nodes can have high localization accuracy and benefit the successful recognition ratio.
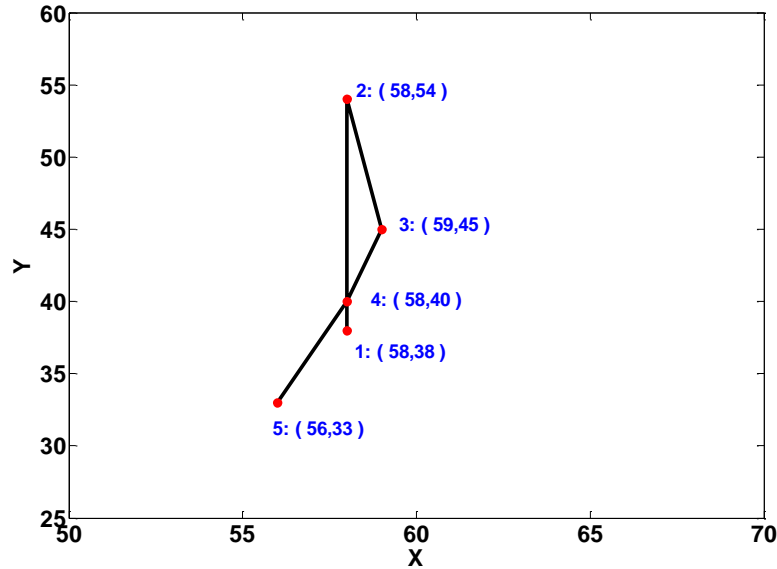
**Fig. 4.** the example of traget trace

## 3.3 Dynamic Difference Algorithm

In this subsection, we propose an algorithm named Dynamic Difference Algorithm (DDA) in our gesture recognition. The Basic idea of this algorithm is to first utilize the difference of RSSI Dynamic Value between the Top Link and the Bottom Link, then choose the maximal difference to estimate the gesture. The algorithm detail is as follows.
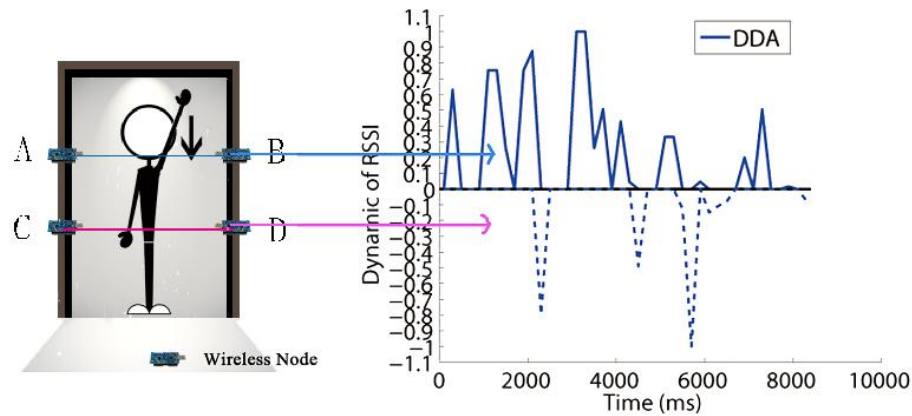


**Fig. 5.** the example of DDA algorithm

We illustrate the DDA algorithm in the **Fig. 5**. Suppose the gesture is waving hand from top to down. For each timestamp, we subtract the dynamic value of RSSI of Link *AB* to that of Link *CD*. The result is shown in the right part of **Fig. 5**. The upper part above the zero line represents the scenario that Link *AB* is influenced by the gesture more than Link *CD*, vise verse for the lower part below the zero line. Therefore, for the upper part, we choose the time with the maximum RSSI dynamic value as the gesture cross Link *AB*. for the lower part, we

choose the time with the minimum RSSI dynamic value as the gesture cross Link *CD*. So the gesture is able to be estimated.

In general, suppose we have $s$ wireless links. As mentioned before, the RSSI dynamic value of each wireless link $d_{ij}$ can be calculated from Eq. (1) and Eq. (2) (here we consider just the horizontal links). For every two wireless links $d_{ij}$ and $d_{uv}$, we may compute the difference of RSSI dynamic value $H_{ij,uv}$ between the two links as follows.

$$H_{ij,uv} = d_{ij} - d_{uv} \tag{9}$$

In total, we have $C_s^2$ pair of wireless links and their corresponding $H_{ij,uv}$ values. Then we will get time $T_{ij}$ when the gesture crossing the link $ij$ as the time with the maximum $H_{ij,uv}$ value. Also, we will get time $T_{uv}$ when the gesture crossing the link $uv$ as the time with the minimum $H_{ij,uv}$ value. It can be expressed as

$$T_{ij} = T_{\max(H_{ij,uv})} \quad \text{and} \quad T_{uv} = T_{\min(H_{ij,uv})} \tag{10}$$

For each pair of wireless link $ij$ and $uv$, we may derive the gesture $G_{ij,uv}$ as follows.

$$G_{ij,uv} = \begin{cases} G_1 & \text{if } T_{ij} - T_{uv} < 0, \\ G_2 & \text{if } T_{ij} - T_{uv} > 0, \\ F & \text{if } T_{ij} - T_{uv} = 0. \end{cases} \tag{7}$$

In total, we will have $C_s^2$ gesture decision $G_{ij,uv}$. Suppose there are $l$ number of gesture of gesture decision falls into gesture $G_1$, while $k$ number of gesture decision falls into gesture $G_2$, and the final gesture decision $G_{final}$ will be generated by following formula.

$$G_{final} = \begin{cases} G_1 & \text{if } l > k, \\ G_2 & \text{if } l < k, \\ F & \text{if } l < k. \end{cases} \tag{5}$$

The advantage of DDA algorithm is that it can overcome the drawbacks of previous two algorithms. The Peak-Time algorithm does not work well when the environmental noise influence dramatically on the RSSI Dynamic values of both Top Link and Bottom Link, making the peak time easily influenced by such noise behavior. Our DDA algorithm can eliminate such noise behavior to the both links. The Best-fit algorithm leverages the localization algorithm to recognize the gesture. It works well when there is enough number of wireless links. However, the area for human to performed the gesture is limited, since the human hand can cover only a limited range. So the accuracy is limited.
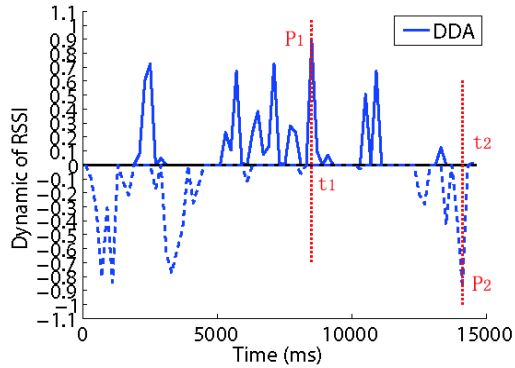
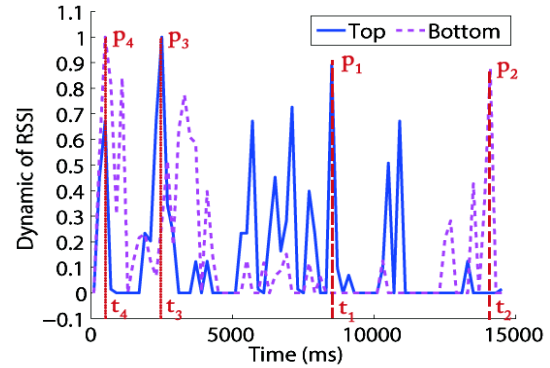**Fig. 6.** the figure capture of  DDA algorithm



**Fig. 7.** the influence of noise behavior to Peak-Time algorithm

For example, **Fig. 7** shows an example when the human gesture is from top to down, based on 4 wireless sensor nodes deployed in the door area (as **Fig. 5** shows). We may see that, according to previous Peak-Time algorithm, the peak time of Top Link is $t_3$ with RSSI dynamic peak value $p_3$, while the peak time of Bottom Link is $t_4$ with RSSI dynamic peak value $p_4$. However, $t_1$ is the real the peak time of the Top Link, and $t_2$ is the real the peak time of the Bottom Link, respectively. The two peak RSSI dynamic values $p_3$ and $p_4$ are very possible caused by the noise, since they are very close to each other and both have sharp variance around time $t_3$ and $t_4$. If using DDA algorithm, the result is shown in **Fig. 6**. We may see that the noise behavior is eliminated at time $t_3$ and $t_4$. $p_1$ and $p_2$ is the real peak time of the Top Link and Bottom Link, respectively. This algorithm is also easily to be extended to recognize other gestures based on different node topologies.

## 4. Experimental Classification Results and Analysis

### 4.1 Experiment Set Up



**Fig. 8.** Experimental environment

We run the experiments in our lab. The lab area is 80 square meters. The wireless nodes we use are telosB sensor nodes with Chipcon CC2420 radio chips. TelosB is composed of the MSP430 (the MSP430F1611) microcontroller and the CC2420 radio chip. The microcontroller of this mote operates at 4.15$MHz$ and has a 10$kBytes$ internal RAM and a 48$kBytes$ program Flash memory. The working band stands on 2.4$GHz$. The sensors are deployed in a regular door area as shown in **Fig. 8**. The width and the height of the door area are about 110.2$cm$, and 213.8$cm$, respectively. In detail, the sensors are deployed on either side of the door with fixed distance between them. We program all the sensor nodes to broad cast beacons periodically with the same interval. Each node broadcasts beacon messages periodically and listens to the beacons from its neighbors as well. The transmission power is defaulted at 0$dBm$.

## 4.2 Infrastructure

In general, our HandButton system consists of two phases. First, in the initialization phase, each node builds a static table to store the static RSSI values for all its neighbors after receiving a few numbers of beacons (5 in our experiment). The average RSSI values of each pair of two nodes will be adopted as the benchmark to estimate the gesture. The initialization phase has to be carried out in the static environment. After initializing all the nodes, the system enters the recognition phase. Each node measures the RSSI dynamic value (compared with the static value) caused by the target gesture and report back to the sink node. After receiving the RSSI dynamic value and its occurrence time, we can estimate target gesture by leveraging our proposed algorithms.

## 4.3 The Impact of Node Distance

**Table 1.** Comparison of different node distance

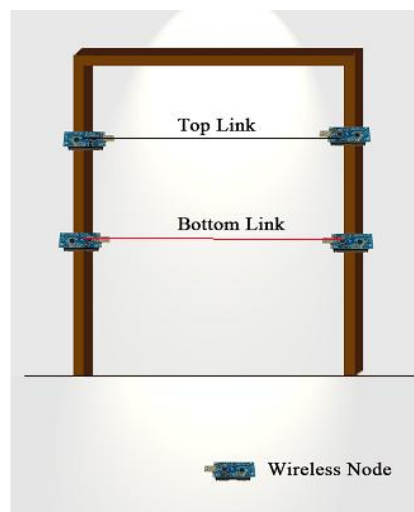| Node distance | Accuracy (%) |
|:---:|:---:|
| 30$cm$ | 82.35 |
| 40$cm$ | 86.20 |
| 50$cm$ | 72.73 |
| 60$cm$ | 72.73 |



**Fig. 9.** The example of node deployment

In this experiment, we will investigate how the node distance will impact the successful recognition radio. Since the sensors are deployed in the door area to recognize the human hand gesture and the width of the door is fixed, we only test the vertical node distance along each side of the door. Therefore, we adjust the perpendicular distances of each node from 30*cm* to 60*cm*, with each step separating by 10*cm*. We leverage 4 sensors in the door area, with two on each side as shown in **Fig. 9**. The line between the upper two sensors is the top link. The line between the lower two sensors is the bottom link. We test the human hand gesture from top to down. For each node distance, we tested 15 rounds. The experiment result is shown in **Table 1**. We find that, when node distance is set to 40*cm*, the result is the best, whose successful recognition ratio is 86.2%. **Fig. 10** is one of the results when node distance is 40*cm*.

We also find that, the nodes distance less than 30*cm* will not be benefit to the gesture recognition successful ratio. The reason may be the following. If the distance between the top link and bottom link is too close, the time difference between these two links caused by the hand gesture is not obvious. Thus, the gesture is more difficult to recognize. We also skip testing those node distances larger than 60*cm*. The reason is that, in general, the human hand gesture is unlikely to cover a very long distance. Therefore, in the following experiment, we will set the node distance default as 40*cm*. We mainly compare the following results with the best setting in our experiment.
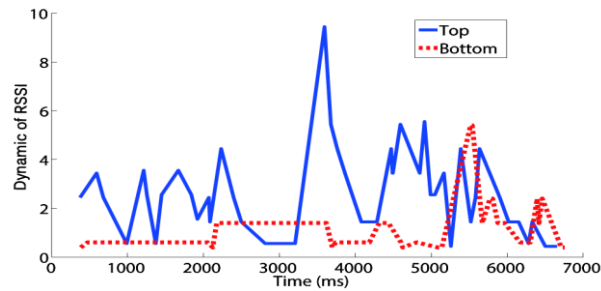


**Fig. 10.** The example of a figure caption (when node distance is 40cm)

## 4.4 The Impact of Node Numbers in a 2D Area

**Table 2.** Comparison of different node number

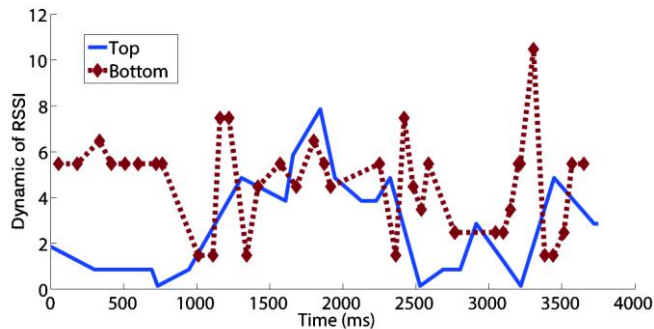| Node number | Accuracy (%) |
|---|---|
| Four nodes | 73.33 |
| Six nodes | 80.00 |



**Fig. 11.** The example of figure caption (with four nodes)

We totally tested 20 rounds for each kind of human hand gesture. **Fig. 11** is one of the results when the node number is 4. **Fig. 12** is one of the results when the node number is 6. The top link, middle link and the bottom link represent the horizontal wireless link between the top two nodes, the two middle nodes, and the two bottom nodes, respectively.
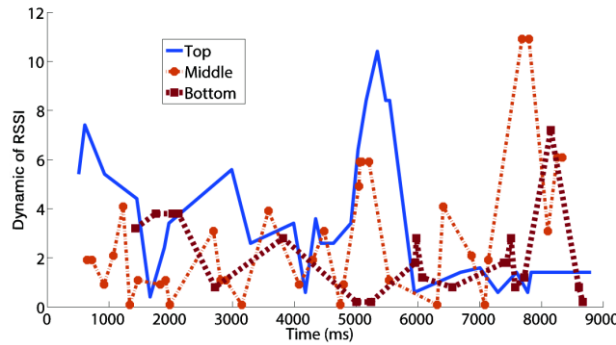


**Fig. 12.** The example of figure caption (with six nodes)

Base on all the samples, as **Table 2** shows, the successful recognition ratio with four nodes is 73.3% and with six nodes is 80%, which is improve by 6.7%. Therefore, in the later experiments, we utilize this setting with 6 nodes in the 2D door area. The reason why we do not use more nodes is that, since the vertical node distance we choose is 40cm, totally the vertical distance this setting will cover is 80cm. A typical human hand gesture will not cover longer than this distance.

## 4.5 The Impact of 3D Deployment

**Table 3.** Comparison of different deployment

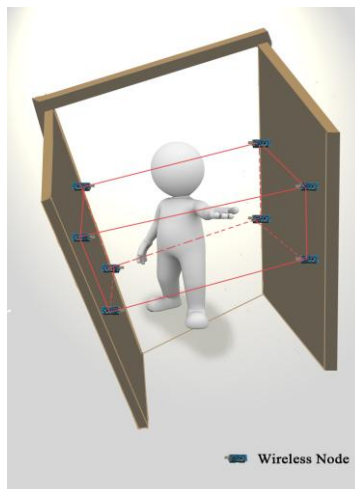| Arrangement | Accuracy (%) |
|---|---|
| 2D(4 nodes) | 73.33 |
| 2D(6 nodes) | 80.00 |
| 3D(8 nodes) | 60.00 |
| 3D(12 nodes) | 65.00 |



**Fig. 13.** Wave hand in 3D deployment

In order to investigate whether a 3D node setting will improve the successful recognition radio, we perform our experiment with the setting shown in **Fig. 13**. Considering the covered area of common human hand, we only consider the 3D node setting with 8 nodes and 12 nodes, as shown in **Fig. 14** and **Fig. 15**, respectively. The distance between nodes is 40 *cm*. In the 3D setting, for example with the 8 nodes setting as shown in **Fig. 14**, we have two Top Links (Top-Front link and Top-Back link) and two Bottom Links (Bottom-Front link and Bottom-Back link), as described in **Fig. 16**. For each pair of Top and Bottom links, we will estimate the gesture, then conclude the final gesture based on the decision of all these pair of links, as introduced before.
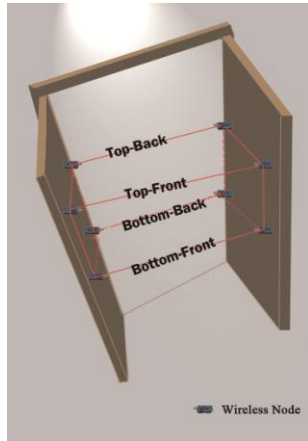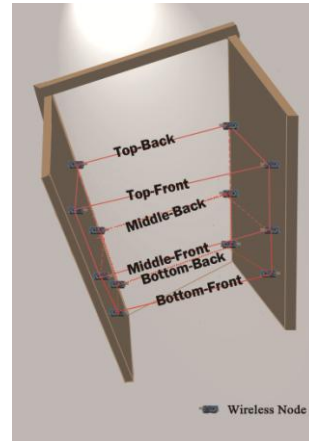


**Fig. 14.** The example of 3D deployment (with 8 nodes)



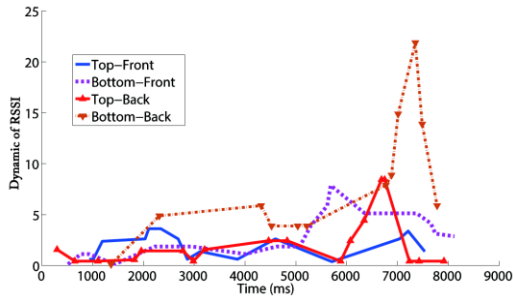**Fig. 15.** The example of 3D deployment (with 12 nodes)



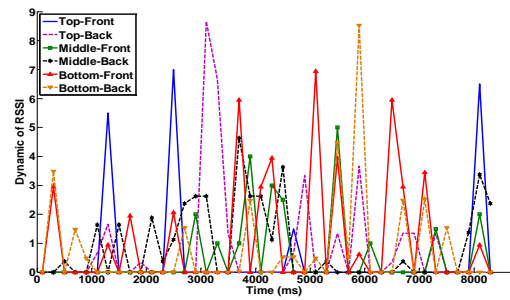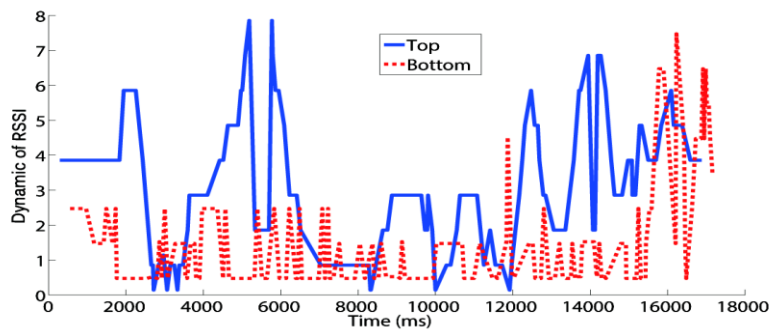**Fig. 16.** The example of figure caption(with 8 nodes in a 3D area)



**Fig. 17.** The example of figure caption(with 12 nodes in a 3D area)

In total we tested 20 rounds. The successful recognition rate of 3D node setting with 8 nodes and 12 nodes are 60% and 65%, respectively, as **Table 3** shows. **Fig. 17** is one of the results with the 3D node setting with 12 nodes. Surprisingly we find that, compared to the 2D node setting, the successful recognition rate is decreased in the 3D setting. Furthermore, increasing the number of nodes does not improve the accuracy a lot. The reason may be as follows. First, more communicating nodes may easily cause interference to each other. Second, since the length of human arm is limited, the links far away from the human hand (close to the finger part) may not be influenced as good as the closer links. Therefore, 2D node setting is more suitable in our application scenario of human gesture recognition. In the later experiments, we only utilize the 2D node setting.
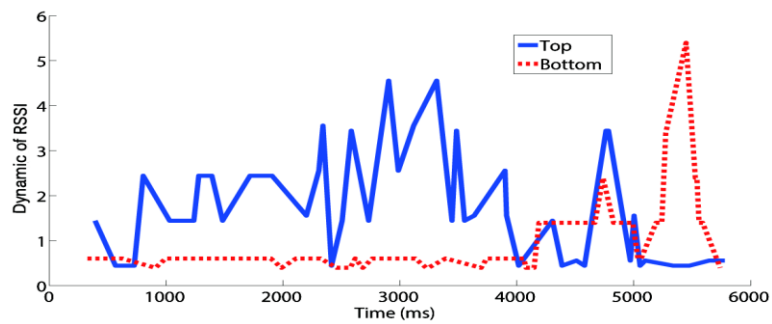
## 4.6 The Impact of Moving Speed
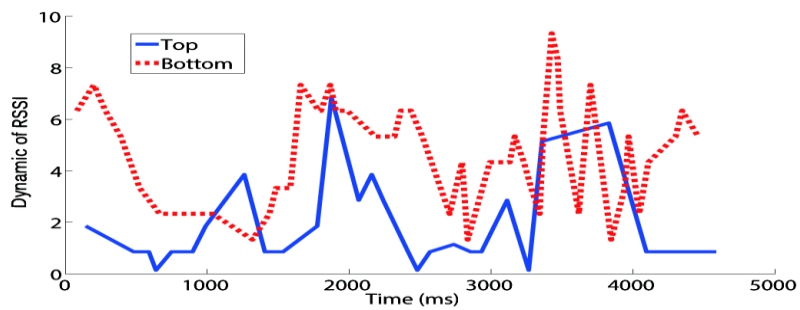
**Table 4.** Comparison of different moving speed

| Moving speed | Accuracy (%) |
| --- | --- |
| Slow | 71.43 |
| Normal | 88.89 |
| Quick | 66.67 |



(a) Slow



(b) Normal



(c) Quick

**Fig. 18.** The example of figure caption (slow, normal and quick)

To learn the speed of the target gesture will influence the successful recognition rate, we arrange a person to wave his hand at a fixed trace between the two sensor grids on the either side of the door. The vertical distance of the nodes are fixed at 40*cm* and in total we use 6 nodes, which has been introduced in the previous subsection. We test three kinds of moving speed: slow (about 2.27 *m/s*), normal (about 4.48 *m/s*), and fast (about 10.02 *m/s*). The examples of three moving speed are shown in **Fig 18**. The Experiment results are shown in **Table 4**. We find that, when the gesture is performed at a normal speed, the successful recognition ratio is the best.

## 4.7 Algorithm Comparison

**Table 5.** Comparison of different algorithm

| Algorithm | Accuracy (%) |
|-----------|--------------|
| Peak-Time | 75.56 |
| Best-Fit | 76.67 |
| DDA | 77.78 |

As introduced in the previous subsection, we choose 6 nodes and the vertical node distance as 40*cm* in a 2D area as the final gesture recognition deployment. We compare our Peak-Time Best-Fit algorithms and DDA algorithms, based on 55 rounds of two human hand gesture tests ($G_1$ and $G_2$). The experiment result is shown in **Table 5**. We can find that, the successful recognition ratio of the Peak-Time algorithm, Best-Fit algorithm and DDA algorithm are 75.56%, 76.67% and 77.78%, respectively. We may see that, DDA algorithm performs the best in terms of the successful recognition rate. Moreover, the Best-Fit algorithm is able to find more complicate target gesture than Peak-Time algorithm. **Fig. 19** shows an example of the gesture trajectory when the moving speed is slow, **Fig. 20** shows and example with the normal speed. We may see that, the estimated trace is able to give more information of the target gesture.
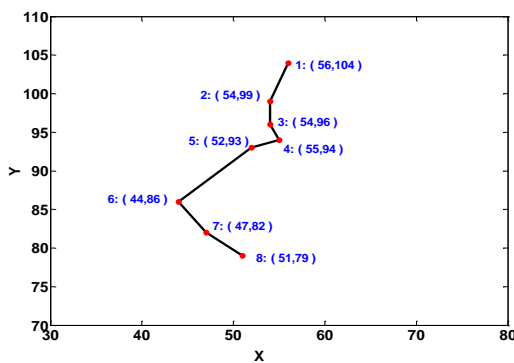


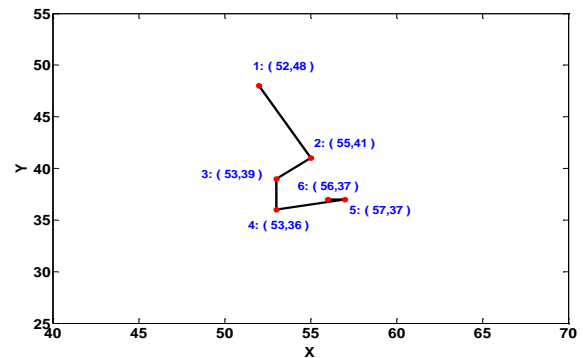**Fig. 19.** The example of gesture trajectory with slow speed



**Fig. 20.** The example of gesture trajectory with normal speed

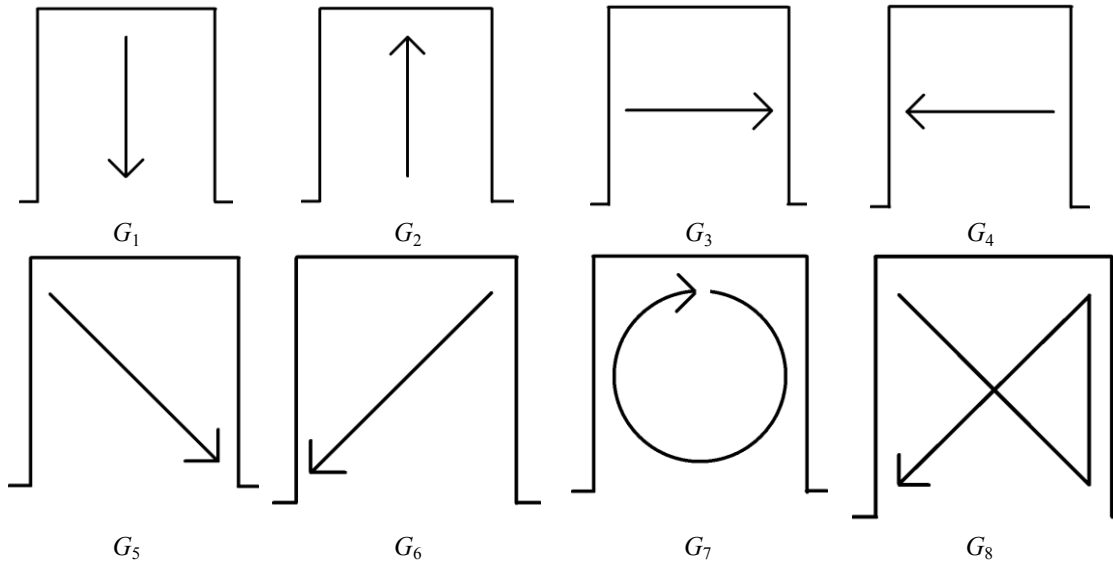## 4.8 Potential to recognize other Gestures



**Fig. 21.** Notations of the tested gestures

The three algorithms introduced before are able to recognize target gestures G1 and G2 with high success recognition ratio. Actually these algorithms have immense potential to recognize other target gestures. The Peak-time and DDA algorithms works well if the target gesture may cross the Line-Of-Sight (LOS) path of the fix pair of nodes. Since Best-Fit algorithm utilizes localization algorithm to perform gesture recognition. It is more likely to be leveraged to recognize more complicated gestures.

In this subsection, we use Best-fit algorithm and compare 8 gestures (G1 to G8, as shown in Figure 21) with the same deployment as introduced before, and the experiment results of each gesture are based on 10 rounds tests. We find that, the Best-Fit algorithm is able to recognize some complicated gestures despite of the low successful recognition ratios in this subsection. **Fig. 22** shows a graphic representation when gesture $G_7$ is successfully be recognized. The experiment result is shown in **Table 6**. We can find that, the successful recognition ratio of all gestures but $G_1$ and $G_2$ are not good enough. The reasons are as follows.

At first, in our scenario, we deploy the nodes in the door area. Due to the limitation of the door area, the human hand gesture is impossible to cross the door frame. Thus, for those gestures parallel to the ground, e.g., $G_3$ and $G_4$, the successful recognition ratio will be influenced unless the deployment could be changed.

Second, since the distance between sensor nodes is small than 1m, the signals of these sensor nodes are much affected by noise, especially when the other 5 gestures cause little impact to the LOS path link between the two parallel sensor nodes.

Third, since the number of deployed nodes is only 6, more nodes can contribute more on the localization accuracy as well the successful recognition rate.[15] [16]

The successful recognition rate can be improved if we change the deployment, increase the number of deployed nodes and change the deployment area in other application. We will try to improve the accuracy by rearranging the deployment and reducing the noise interference in

our future work. For the current application of control the light, high successful recognition rate for gesture G1 and G2 can satisfy such requirement.

**Table 6.** Comparison of different gesture by using Best-Fit Algorithm

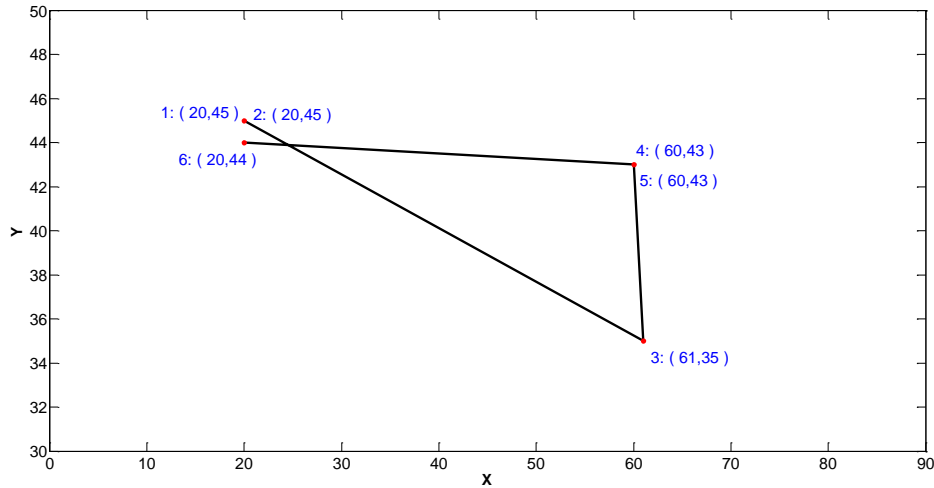| Gesture | Accuracy (%) |
|---------|--------------|
| $G_1$ | 90 |
| $G_2$ | 80 |
| $G_3$ | 30 |
| $G_4$ | 40 |
| $G_5$ | 40 |
| $G_6$ | 40 |
| $G_7$ | 30 |
| $G_8$ | 30 |



**Fig. 22.** The figure capture of gesture $G_7$

### 4.9 Latency

The latency of gesture recognition system mainly depends on how much time for the data has been collected. In our experiment, to avoid collision among 6 nodes, we set the beacon interval as 200*ms* to transmit a packet with 51*bytes*. For Peak-Time algorithm and DDA algorithm, the system latency is 200*ms*. But for Best-Fit Algorithm, we should estimate the target location over 2 times to decide the gesture. Therefore, the latency should be larger than $2 \times 200ms = 400ms$.

## 5. Conclusion

In this paper, we have presented three algorithms to recognize target object gesture behavior (mainly the humans) by using wireless sensor networks. At the same time, the target does not require to carry any device. The first Peak-Time algorithm uses the time difference of RSSI dynamics among difference links for recognition. It is easy to perform and with well successful recognition probability. The second Best-Fit algorithm introduce localization

algorithm, which has higher successful recognition probability and able to recognize more complicate gesture. The last DDA algorithm has the highest successful recognition probability through eliminating the noise behavior in the environment. We perform our algorithm in real environment. The experiment results show that we may recognize target gesture with the successful probability up to about 80%. As future work, we will try to recognize more different gestures. Furthermore, we may try to recognize gestures of multiple objects. More nodes deployment will be also under consideration in our future work. We also may try to reduce the noise interference.

## Acknowledgement

## References

[1] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Transactions on Systems*, Man, and Cybernetics, Part A, pp.1064-1076, 2011. Article (CrossRef Link).

[2] V. Kosmidou and L.J. Hadjileontiadis, "Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data," *IEEE Trans. Biomed. Engineering*, pp.2879-2890, 2009. Article (CrossRef Link).

[3] "XBOW Corporation," TelosB Mote Specifications. Article (CrossRef Link)

[4] R.Z. Khan and N.A. Ibraheem, "Survey on gesture recognition for hand image postures," *Computer and Information Science*, pp.110-121, 2012. Article (CrossRef Link).

[5] Z. Ren, J. Meng, J. Yuan and Z. Zhang, "Robust hand gesture recognition with kinect sensor," in *Proc. of ACM Multimedia*, pp.759-760, 2011. Article (CrossRef Link).

[6] C. Tran and M.M. Trivedi, "3-D posture and gesture recognition for interactivity in smart spaces," *IEEE Trans. Industrial Informatics*, pp.178-187, 2012. Article (CrossRef Link).

[7] X. Chen and M. Koskela, "Online RGB-D gesture recognition with extreme learning machines," in *Proc. of ICMI*, pp.467-474, 2013. Article (CrossRef Link).

[8] O.D. Lara and M.A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys and Tutorials*, pp.1192-1209, 2013. Article (CrossRef Link).

[9] R.Z. Xu, S.L. Zhou and W.J. Li, "MEMS accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors Journal*, vol.12, no.5 pp.1166-1173, May 2012. Article (CrossRef Link).

[10] Z. Lu, X. Chen, Q. Li, X. Zhang and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE T. Human-Machine Systems*, pp.293-299, 2014. Article (CrossRef Link).

[11] F. Adlib, Z. Kabelac and D. Katabi, "Multi-person motion tracking via RF body reflections," *Computer Science and Artificial Intelligence Laboratory Technical Report*, 2014.

[12] Q. Pu, S. Gupta, S. Gollakota and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. of MOBICOM*, pp.27-38, 2013. Article (CrossRef Link).

[13] F. Adib and D. Katabi, "See through walls with WiFi!," in *Proc. of SIGCOMM*, pp.75-86, 2013. Article (CrossRef Link).

[14] D. Zhang, J. Ma, Q. Chen and LM. Ni, "An RF-based system for tracking transceiver-free objects," in *Proc. of Fifth Ann. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '07)*, pp. 135-144, 2007. Article (CrossRef Link).

[15] D. Zhang, K. Lu, R. Mao, Y. Feng, Y. Liu, M. Zhong, LM. Ni, "Fine-Grained localization for multiple transceiver-free objects by using RF-Based technologies," *IEEE Transactions on Parallel and Distributed System (IEEE TPDS)* (ISSN:1045-9219), 25(6): 1464-1475, 2014.2. Article (CrossRef Link).

[16] D. Zhang, Y. Liu, LM. Ni, "RASS: A real-time accurate and scalable system for tracking transceiver-free Objects," *IEEE Transactions on Parallel and Distributed System (IEEE TPDS)* (ISSN: 1045-9219), 24(5): 996-1008, 2013. Article (CrossRef Link).

**Dian Zhang** received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2010. After that, she worked as a Research Assistant Professor at Fok Ying Tung Graduate School, Hong Kong University of Science and Technology, Hong Kong. She is currently an Assistant Professor at Shenzhen University. Her research interests include wireless sensor networks, pervasive computing and networking and distributed systems. She is amember of the IEEE.

**Weiling Zheng** received the B.E. degree in software engineering from the Shenzhen University of Computer Science and Software Engineering, Shenzhen, China, in 2015. She is currently a Research Assistant at Dian Zhang's reseach group.