Automatic Feature Point Detection and Tracking of Human Actions in Time-of-flight Videos

Xiaohui Yuan, *Senior Member, IEEE,* Longbo Kong, Dengchao Feng, and Zhenchun Wei

***Abstract*—Detecting feature points on the human body in video frames is a key step for tracking human movements. There have been methods developed that leverage models of human pose and classification of pixels of the body image. Yet, occlusion and robustness are still open challenges. In this paper, we present an automatic, model-free feature point detection and action tracking method using a time-of-flight camera. Our method automatically detects feature points for movement abstraction. To overcome errors caused by miss-detection and occlusion, a refinement method is devised that uses the trajectory of the feature points to correct the erroneous detections. Experiments were conducted using videos acquired with a Microsoft Kinect camera and a publicly available video set and comparisons were conducted with the state-of-the-art methods. The results demonstrated that our proposed method delivered improved and reliable performance with an average accuracy in the range of 90%. The trajectory-based refinement also demonstrated satisfactory effectiveness that recovers the detection with a success rate of 93.7%. Our method processed a frame in an average time of 71.1 ms.**

***Index Terms*—Feature point, human pose detection, joint detection, time-of-flight (ToF) videos.**

I. INTRODUCTION

HUMAN action tracking enables a wide range of applications such as virtual personal trainer and accident monitoring for elderly. In such a video-based tracking and analysis system, the geometric feature points of a human body, e.g., head, hands, feet, shoulders, and elbows, provide characteristic abstraction of various poses. A key step to track and recognize human movements is to automatically detect these feature points from video frames.

Accurate detection of the feature points in video frames, however, is a challenging problem when human subjects wear no markers. There are methods developed to detect feature points from videos of different modalities, e.g., range (or depth) videos and infrared videos [1]. Range videos exhibit great advantage in comparison to visual and infrared videos. Range videos are usually captured by depth camera, which can calculate the distance between a certain point in its field of view, and the camera itself. Different from the traditional optical cameras, depth cameras acquire point cloud instead of images made up with pixels. Each point in the point cloud carries a depth value, which provides an extra dimension to handle occlusions. The existing strategies for feature point detection and movement tracking can be categorized as model-based or model-free methods. In model-based methods, feature points are identified by fitting a model to the acquired point cloud [2]. Yet, difficulties arise from variations in human figures and occlusions. Model-free methods (also known as Data-driven methods), however, extract feature points from the acquired data with few or no assumptions using image features such as color and shape, which overcome the rigidness of using a model but demand accurate detection of feature points [3].

In this article, we propose a model-free method for detecting feature points from time-of-flight videos. Our proposed method automatically detects the extreme points from the three-dimensional point cloud of a human figure. The extreme points are a subset of the feature points, which include the ends of human body parts, i.e., head, hands, and feet. By initializing with a head-shoulder template, shoulders are detected, which serve as part of the references for the detection of the elbows. In the process of tracking, trajectory of the feature points is used to improve the precision of detection.

The contributions of this work is three-fold: first, a model-free, automatic extreme point detection strategy is proposed, which is robust to pose variations and occlusions; second, an optimization method for extraction of joints based on geometric constraints of human body parts is devised; third, our tracking method integrates the trajectory of the feature points for improved tracking precision.

The rest of this article is organized as follows: Section II reviews the related work on feature point detection from video frames and tracking of human movements. Section III presents our method for automatic feature point detection and action tracking. Section IV discusses our experimental results. Section V concludes this paper with a summary and future work.

II. RELATED WORK

There have been many studies on the detection of human pose in real-time and tracking human movements. Landmark detection, three-dimensional (3D) human body model fitting, and pixel classification are mostly used strategies. In the

Manuscript received June 7, 2016; accepted October 28, 2016. Recommended by Associate Editor Wei He. *(Corresponding author: Zhenchun Wei.)*

Citation: X. H. Yuan, L. B. Kong, D. C. Feng, and Z. C. Wei, "Automatic feature point detection and tracking of human actions in time-of-flight videos," *IEEE/CAA J. of Autom. Sinica*, vol. 4, no. 4, pp. 677−685, Oct. 2017.

X. H. Yuan is with China University of Geosciences (Wuhan), Wuhan 430074, China, and also with the University of North Texas, Texas 76203-1277, USA (e-mail: xiaohui.yuan@unt.edu).

L. B. Kong is with the University of North Texas, Texas 76203-1277, USA (e-mail: longbokong@my.unt.edu).

D. C. Feng is with the University of North Texas, Texas 76203-1277, USA (e-mail: df0158@unt.edu).

Z. C. Wei is with the Hefei University of Technology, Hefei 230009, China (e-mail: weizc@hfut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2017.7510625

methods that discover landmarks for tracking, the silhouette of human body is usually treated as a graph where pixels within the body are nodes and the immediate adjacency makes the edges. Baak *et al.* [3] applied a modified Dijkstra's algorithm to identify the extreme points in the human body silhouette. Furthermore, they also used Dijkstra's algorithm to detect extreme points and calculated the orientation of the extreme points by back-tracking the searching path. Plagemann *et al.* [4] used a body part identification method along with an extreme point searching method, which extracted the shape feature from data and classified them into body parts.

Landmarks give a rough description of a human pose. However, it is sometimes difficult to know the exact pose abstracted by the landmarks. Hence, many landmark-based methods rely on other sub-systems to provide additional information of human poses. Fitting a 3D human body model is usually used to estimate human poses. Baak *et al.* [3] developed a 3D model fitting and a database lookup system. In this method, a database of human poses was used to provide the closest poses for selecting landmarks. The retrievals from the database were used to compare with the results from the fitted 3D model and the previous poses. The final result was generated via a voting process. Handrich *et al.* [5] applied Dijkstra's algorithm to detect the extreme points. Because their focus was on the upper body, the proposed method only searched for the extreme points for hands and head, and a 3D skeleton was used to detect shoulders and elbows. Schwarz *et al.* [6] proposed a geodesic distance based algorithm to search for the extreme points as the primary landmarks, and registered a 3D skeleton model with the depth frames. Huang *et al.* [7] developed a method that employs a 3D human body model with cylindric body parts to fit with 3D point cloud from depth camera. Ganapthi *et al.* [8] also used a 3D human body model to fit into point cloud to estimate human poses.

Besides landmark detection, classification methods have been applied to image pixels for body part identification. Wei *et al.* [9] used classification method in their work to initialize the system and reinitialized the system when failure occurred. Shotton *et al.* [10] applied decision trees to classify each pixel in a depth frame and calculated the probability of each pixel for being part of a joint. However, classification based methods demand high computation power.

In the detection of human poses and tracking of movements, joints are commonly used to provide a more accurate abstraction of human body. Many methods have been proposed to detect joints. When joints detection is required, a 3D human body model fitting is usually used. Weng *et al.* [11] used templates to detect body parts and applied a kinematic chain on the body to locate joints. The pixel classification method in [9] was used to determine the body parts and the location of the joints. Then, an inverse kinematic chain method is used to obtain the estimated human pose results.

III. Methodology

Our method consists of three key components: extreme point detection, joint detection, and feature point refinement. Fig. 1 illustrates an overview of the work flow. The time-of-flight (ToF) videos are processed to extract the human figure using background removal. Extreme points are then automatically detected based on geometric properties of the three dimensional silhouette. With a head-shoulder template, shoulders are detected, which serve as part of the references for the detection of elbows. The temporal locations of each feature point are used to form a profile to make prediction of its future location. This predicted location is then used as a constraint for refining the feature point detection. In the rest of this section, we present each component in detail.

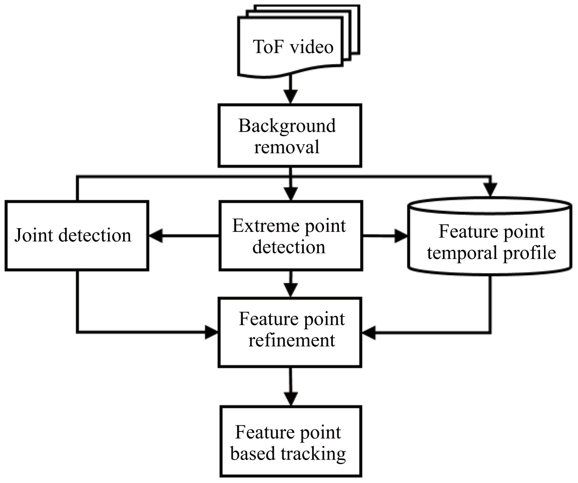


Fig. 1. An overview of the work flow of our proposed method.

A. Background Removal

To extract the silhouette of human body, background needs to be removed. Different from optical cameras, depth cameras acquire the distance between objects to the camera rather than the spectral information. Following the idea in [12], the average depth at a pixel across a temporal range is used to create a background model for human silhouette extraction. If the object remains stationary, its depth value remains the same. In reality, however, due to the presence of noise, the depth value at a pixel varies slightly in a certain range even though the scene remains unchanged. The following model is used to describe the stationary objects that form the background:

$$p_{i,j} = \sum_{n=1}^{N} \frac{p_{i,j}^{n}}{N} \quad \forall p_{i,j} \in \mathcal{B} \tag{1}$$

where $p_{i,j}$ is the depth value at a pixel location (i, j) of background model $\mathcal{B}$, $p_{i,j}^{n}$ is the depth value at a pixel location (i, j) of the nth frame, and N is the number of frames used to build the background model. For each pixel in the background model, we calculate the average sum of the depth of all pixels at the same position from the sample frames as its depth value. When an object appears in a frame m, the depth value of pixels in the area of the occluded background changes, and the changes should be greater than a threshold T as following:

$$\begin{aligned} P^{m} = \{p_{i,j}^{m} \in f^{m} | \triangle p_{i,j}^{m} > T\} \\ \forall p_{i,j}^{m} \in f^{m} \quad \text{and} \quad \triangle p_{i,j}^{m} = p_{i,j}^{m} - p_{i,j} \end{aligned} \tag{2}$$

where P^{m} is the set of foreground pixels and $\triangle p_{i,j}^{m}$ is the difference between pixel $p_{i,j}^{m}$ in frame f^{m} and pixel $p_{i,j}$ in background model.

B. Extreme Point Detection

The extreme points in a human silhouette include head, hands, and feet. Without occlusion, five extreme points are expected. The spatial distribution of extreme points roughly represents the general information of human activities. To simplify our explanation, we present our method in a 2D scenario. However, it is straightforward to extend to 3D cases.

Let I denote a 2D binary image that contains a human silhouette and the background. The background pixels are zeros and the foreground pixels (i.e., human body silhouette) are ones. Given a randomly selected point inside the silhouette, denoted with E_0, the geodesic distance between E_0 and a point is calculated as follows:

$$D_g(E_0, I(x,y)) = \sum \|I(x_p, y_p) - I(x_q, y_q)\| \tag{3}$$

where $I(x_p, y_p)$ and $I(x_q, y_q)$ are the nearest neighboring points on the shortest route between E_0 and $I(x,y)$. A distance map, denoted with M^i, is created, where i, $i = \{0, 1, \dots, N\}$, is the iteration number. Hence, an extreme point is the one that gives the farthest geodesic distance to the reference point E_0:

$$E_i = \arg\max_{I(x,y)} D_g(E_0, I(x,y)). \tag{4}$$

In search for other extreme points, (4) can be modified by replacing E_0 with the previously identified extreme point E_i as follows:

$$E_i = \arg\max_{I(x,y)} D_g(E_i, I(x,y)), \quad i \neq 0. \tag{5}$$

As the result, an updated distance map is generated.

Yet, to avoid repeatedly reaching the same farthest points on the silhouette, when an extreme point is identified, its geodesic distance to any existing extreme points is set to zero, i.e.,

$$\forall E_i, \quad D_g(E_i, E_j) \Leftarrow 0, \quad i \neq j. \tag{6}$$

Hence, for any newly identified extreme point, it must exhibit the farthest distance to all the existing ones, which is easily satisfied with (6).

The final updated distance map will take the minimum value for each pixel among all distance maps M^i:

$$M^i(x,y) = \min(M^1(x,y), M^2(x,y), \dots, M^n(x,y)). \tag{7}$$

To handle self-occlusion, during the process of updating the distance map M^i, we compute the depth difference (i.e., z difference) between adjacent points (adjacent in the X-Y plane). If the difference is less than a threshold δ, these two points are considered as being on the surface of the same body part; otherwise, they belong to different body parts, i.e., occlusion occurs. In the case of occlusion, the occluded surface is filled by interpolation. Hence, to update the geodesic distance of a point, the neighboring points on the same body part surface are used.

C. Joint Detection

1) Shoulders: Our method uses a head-shoulder (HS) template to search the position of shoulders. A head-shoulder template is to provide a consistent means for finding the locations of shoulders. In the extreme point detection, head and hands are identified. Ideally, when there is a prominent curvature near the shoulders, the shoulders are easily detected. However, in many poses, e.g., the upper arms being leveled with the shoulders, there exist no distinctive marks to differentiate the shoulders from the arms. The template is hence useful to estimate the locations of the shoulders. To obtain the HS template, an initialization pose is used, the silhouette of which is shown in Fig. 3 (a).

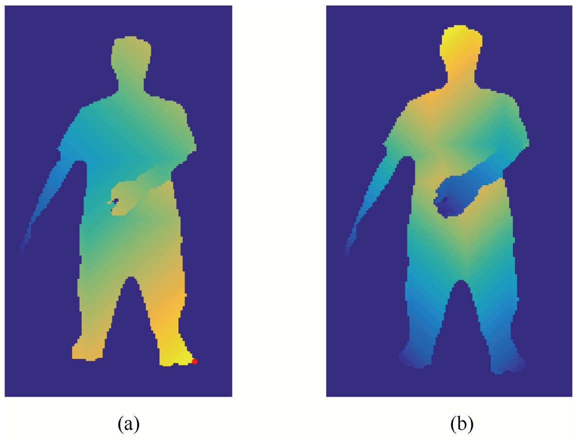


Fig. 2. (a) An example of a distance map for one extreme point. (b) An example of updated distance map (in the 4th iteration). The extreme points on limbs are detected and their distance value is set to 0, the extreme point of head has been calculated (brightest pixel).

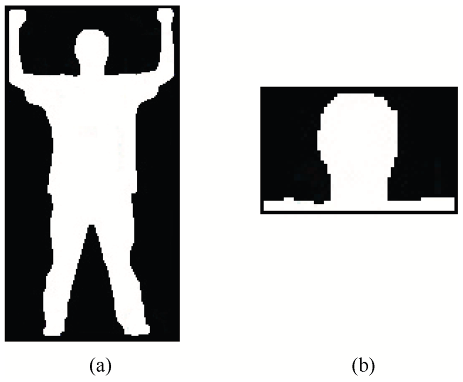


Fig. 3. Head-shoulder template. (a) Example pose. (b) Head-shoulder template captured from the pose in (a).

This initialization pose is used to invoke the detection method and to capture the head-shoulder (HS) template. If the pose of the human subject matches the initialization pose, the silhouette of the head and shoulders are captured and used as the HS template. The similarity of two objects A and B is calculated as follows:

$$S(A,B) = \sum_{i=1}^{7} |m_i^A - m_j^B| \tag{8}$$

where

$$m_i^A = \text{sign}(\eta_i^A) \times \log \eta_i^A$$

$$m_j^B = \text{sign}(\eta_j^B) \times \log \eta_j^B$$

and η_i^A, η_j^B are Hu moments [13] of objects A and B, respectively. Seven moments are used to compute the similarity to the template. An example of the head-shoulder (HS) template is shown in Fig. 3 (b).

The distance between the two shoulders is estimated by calculating the average width of the torso. With the HS template, shoulders are detected. After locating the position with the greatest similarity in the current frame, a bounding box is used to locate the shoulders. The two bottom corners of the bounding box are proposed as the shoulder positions. However, when a proposed shoulder position falls into the background (a side bend posture), a nearest foreground point is located and used as the updated shoulder position. An example is shown in Fig. 4. The position of the extreme point, which represents the head, is refined by this HS template.

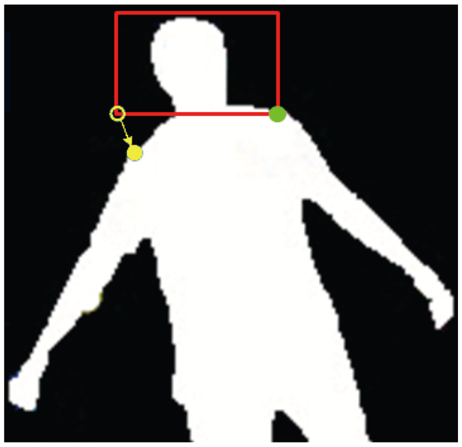


Fig. 4. Detection of shoulders. The green dot marks the left shoulder. The yellow circle marks the proposed location of the right shoulder. However, since the proposed location is in the background, a nearest foreground point is identified as the proposed location of the right shoulder, as depicted by yellow dot.

2) *Elbows:* Identifying the joints improves the accuracy of tracking body movements. Body joints, however, vary from person to person, and depend on the perspective of the imaging device even without occlusion. The intuition of treating the geometric mid-point as the joint location gives a rough estimation; yet, it could be far off as shown in Fig. 5 (b), in which the ideal joint locations are marked with red circle. In Fig. 5 (a), the mid-point between two feature points (in red dots) is a good estimation of the joint; yet, in Fig. 5 (b), the joint is much farther away from the tip of a hand.

When a joint is visible, e.g., in a bent arm, the geodesic distance between two extreme points is greater than their Euclidean distance:

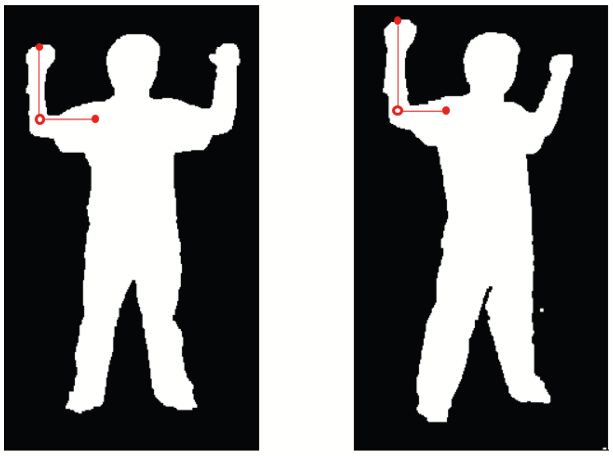
$$D_g(E_i, E_j) > D_e(E_i, E_j) \tag{9}$$

where $D_e(\cdot)$ denotes the Euclidean distance. This can be generalized to any two points on the distinct sections of a body part, e.g., forearm and upper arm. Given that each limb section is rigid and approximately straight, the joint point must satisfy the following relation:

$$D_g(E_i, \hat{E}) + D_g(\hat{E}, E_j) = D_e(E_i, \hat{E}) + D_e(\hat{E}, E_j) \tag{10}$$

$$\text{s.t.} \quad \min_X [D_g(E_i, X) + D_g(X, E_j) - D_e(E_i, X) - D_e(X, E_j)] \tag{11}$$

where $\hat{E}$ is the ideal point for joint location. Given the width of silhouette, there are more than one point that satisfies (10). To break the tie, the one that gives the shortest geodesic distance is used.



(a) (b)

Fig. 5. The geometric properties of joint and extreme points vary according to the imaging perspective. The red dots mark the extreme points and the circle marks the joint location. (a) front view and (b) angular view of the human body. When the human body turns sideway, the geometric properties of the joint and extreme points change.

To find the ideal joint location, an iterative search method is proposed. Our objective function is to minimize the difference between the total geodesic distance between two extreme points through the joint point. According to the updated distance map, the geodesic distance between hand and shoulder will not be changed during the iterative search. Thus, the goal of the iterative search is changed to maximize the Euclidean distance between a hand and the corresponding shoulder. To avoid the searching result going too far outside the arm region, the joint point must be in a distance to the corresponding hand or shoulder:

$$\hat{E} = \sum_k D_g(E_k, \hat{E}) - \sum_k D_e(E_k, \hat{E})$$
$$\text{s.t.} \quad \max_X [D_e(E_i, X) + D_e(X, E_j)]. \tag{12}$$

Fig. 6 illustrates a scenario of finding the elbow in an arm. The feature points and the ideal joint point are marked with red dots. The estimated joint location X in the current iteration is marked with a red circle.

D. *Feature Point Refinement Using Trajectory*

When occlusion occurs or there exists a significant amount of noise, detection of feature points could be very difficult. To avoid failure of detection or missing feature points due to occlusion, we keep track of the trajectory of each feature point to predict its most likely spatial location in the next frame. The predicted location is used as a constraint in our detection. That is, a feature point shall lie in the vicinity of the predicted location based on its historical trajectory. Our refinement process consists of three components: feature point correspondence, location prediction, and verification.

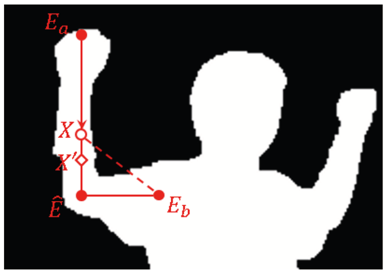


Fig. 6. An example of finding the elbow in an arm.

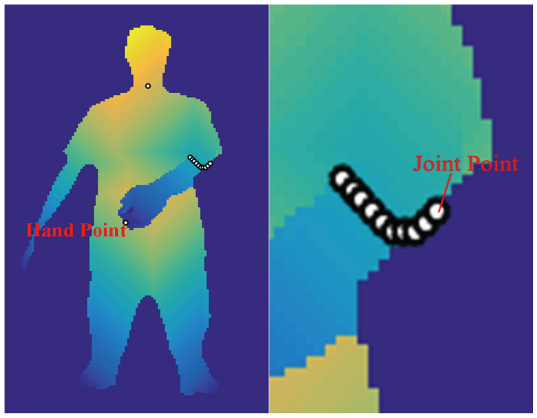


Fig. 7. An example of a searching pattern for an elbow point under self-occlusion situation.

In a feature point detection method, the order of the detection of feature points varies from frame to frame. Thus, the correspondence of feature points between two adjacent frames must be found so that they are tracked correctly in a video. Due to the continuity of human motion, the corresponding feature points have very small changes in position between two adjacent frames. Hence, we impose the minimum distance constraint in the correspondence identification:

$$p_j^n = \arg\min_j \sum D(p_i^{n-1}, p_j^n) \tag{13}$$

where p_i^{n-1} is the ith feature points in frame f^{n-1} and p_j^n is the jth feature points in frame f^n. $D(p_i^{n-1}, p_j^n)$ calculates the spatial distance between the feature point p_i^{n-1} and a feature point p_j^n in frame f^n. In case where more than one extreme point in frame f^n satisfies the above constraint, a mutual exclusive method is used to decide the points as follows:

$$\arg\min_{k,h} \sum D(p_k^{n-1}, p_h^n) \tag{14}$$

where p_k^{n-1} represents the extreme point in frame f^n that has same correspondence point in frame f^{n-1} with another extreme point in frame f^n. p_h^n represents the extreme point from frame f^{n-1}, which is put into correspondence with multiple points in f^n or not in correspondence with any point at all. When unresolved correspondence exists, the predicted result based on the point trajectory is used instead.

To predict the likely location of an extreme point, its positions in the previous frames are used following a linear extrapolation method as follows:

$$\begin{aligned} \bar{p}_i^{n+1} &= p_i^n + \triangle p_i^{n-1} \\ \triangle p_i^{n-1} &= p_i^n - p_i^{n-1} \end{aligned} \tag{15}$$

where p_i^n represents the position of the ith feature point in frame f^n and $\bar{p}_i^{n+1}$ represents the predicted position for the ith extreme point in frame f^{n+1}.

The predicted position based on trajectory may not coincide with the detected one based on the geodesic distance. To refine the position of the feature points, a searching window centered at the predicted position is used. The size of this search window depends on the velocity of the feature point. If the detected feature point is outside the search window, the detection is considered as erroneous and is hence replaced with the predicted position or a nearest foreground point around the predicted position. The size of the search window is defined as follows:

$$r = \alpha(p_i^n - p_i^{n-1}) \tag{16}$$

where α characterizes the velocity of the feature point.

Fig. 8 depicts examples of the refinement of hands' images. The top panel visualizes the depth image in false color and the bottom panel shows the human silhouette with hands marked with circles where the center of the circle represents the hand.

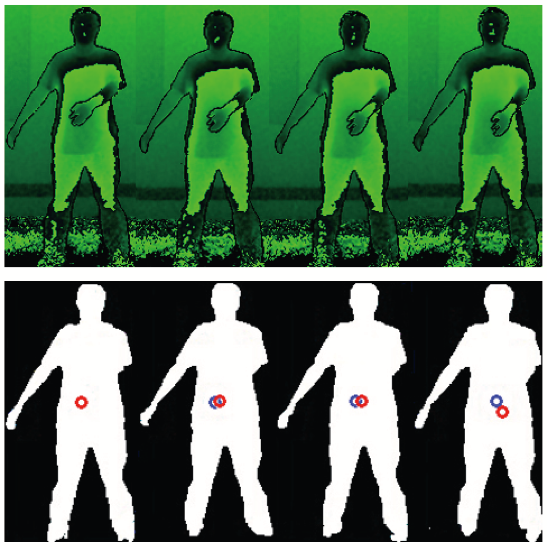


Fig. 8. Tracking result of a hand. The top panels show the visualized depth image. The bottom panels show the human silhouette and the location of the hand. The red circle represents the current position, and the blue circle represents the position in the previous frame.

Algorithm 1 summarizes the steps of our method. Note that the auxiliary data structures, e.g., the one for computing trajectories of feature points, are implicitly expressed in this algorithm.

IV. Experimental Results and Discussion

To evaluate our method, we acquired 10 videos using the second generation Microsoft Kinect and used the SMMC-10

```
Algorithm 1   Tracking with automatic feature point detection
1. Construct a background model $\mathcal{B}$ following (1).
2. for each video frame do
3.     Remove background from a video frame using $\mathcal{B}$ and (2).
4.     Detect the extreme points $E$ from the point cloud of human figure
       following (5).
5.     Detect the joints using the extreme points $E$ following (10).
6.     Find the correspondence of the feature points between two adjacent
       video frames following (13).
7.     Refine the feature point detection by applying the spatial constraint
       described in (15).
8. end for.
```

data set [8]. In the videos we acquired, each frame is of a resolution of 512×424 pixels and the frame rate is 30 fps. Our videos contain various human poses and actions such as walking, kicking, turning, waving hands, and jumping. There are 8 males and 2 females in our video sets with different body shapes. The SMMC-10 data set consists of 28 real-world depth video clips, which was acquired with a Mesa Swiss-Ranger Time-of-Flight camera at a frame rate of 25 fps and a resolution of 176×144 pixels. Each video clip contains human actions such as limb motions, kicks, swings, self-occlusions, and full body rotations. We manually prepared the reference frames to evaluate the accuracy of the feature point detection and tracking. In our evaluation, if a feature point is within 6 cm to the reference point and located inside the body part, the detection is considered correct.

A. Accuracy of Feature Point Detection and Tracking

Fig. 9 illustrates examples of the results from our method using two sets of videos. The feature points detected from the video frames are marked with circles in different colors. These exemplar frames show various poses including rotating the body (the first and second rows in columns (a) and (b)), kicking (the last row in columns (c) and (d)), standing on one foot (the second row in columns (c) and (d)), and squatting (first row of columns (c) and (d) and last row of columns (a) and (b)). It is clear that all feature points were correctly detected with excellent precision.

To evaluate the accuracy of our method, we compared the detected feature points against the hand-marked references using the videos we acquired with Kinect. Table I lists the overall average accuracy for detecting feature points.

TABLE I
OVERALL AVERAGE ACCURACY (%)

Feature points	Head	Shoulder		Elbow		Hand		Foot	
		left	right	left	right	left	right	left	right
Accuracy	84.7	95.1	91.7	68.4	69.0	82.3	85.3	96.6	96.3

Among all feature points, detection of shoulders and foot achieved the greatest accuracy in the range of upper 90 %. The detection of head and hands is also satisfactory, which is in the range of upper 85 %. The detection of elbows is difficult because arms are straight in many frames and the geometric property of the hand-elbow-shoulder is insignificant to differentiate the feature points. In addition, the detection of elbow is based on the detected hands and shoulders. The errors of hand/shoulder detection are hence inherited to the elbow detection. In cases when a hand is occluded, the location of elbow can only be estimated from the trajectory. The relative low accuracy of head is due to the existence of several equally distant points in the detection process and random selection was applied. The usage of head-shoulder template helped to stabilize the feature point of the head. The overall accuracy of detection of all feature points is 85.5 %.

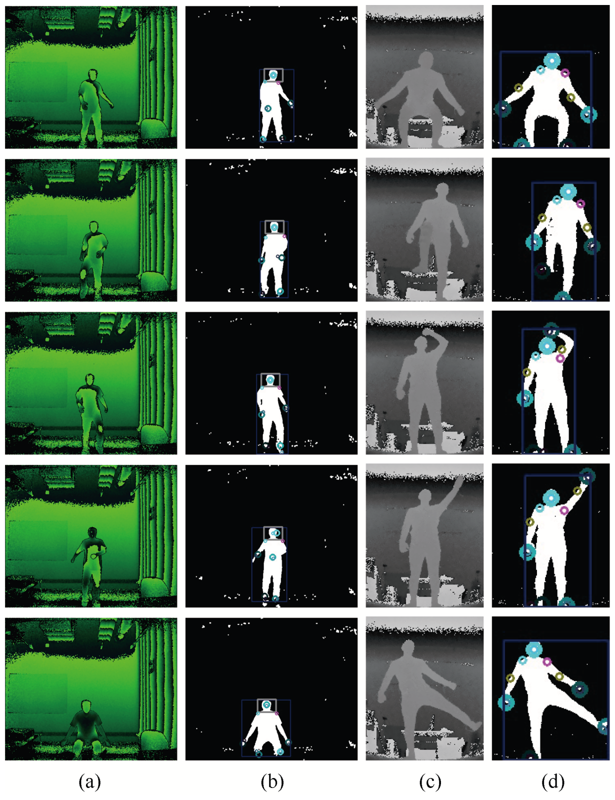


Fig. 9. Exemplar frames and the results from our method. Column (a) shows the frames from the videos of our collection and column (b) shows the results of the detected feature points. Column (c) shows the frames from the videos of SMMC-10 and column (d) shows the results of the detected feature points.

In our experiments, we compared the accuracy of feature point detection with and without occlusion. Fig. 10 shows a bar plot of accuracies of these two situations. The detection of both shoulders and feet demonstrated the greatest robustness whereas the detection of hands and elbows faced performance degradation when occlusion occurred. Our justification of the cause of performance drop for hands and elbows is that in many cases when an arm is in front of the body and hence caused partial occlusion, the distance between the surface of the arm to the body is below the threshold δ used to avoid occlusion, which induced error in computing the geodesic distance. Making this threshold greater or smaller results in different types of errors. When the threshold is small, a body part that is along the sight of the imaging device could be treated as different part. When it is set to much greater, the occlusion is not detected and hence the geodesic distance could be erroneously shortened. Based on our empirical evaluation,

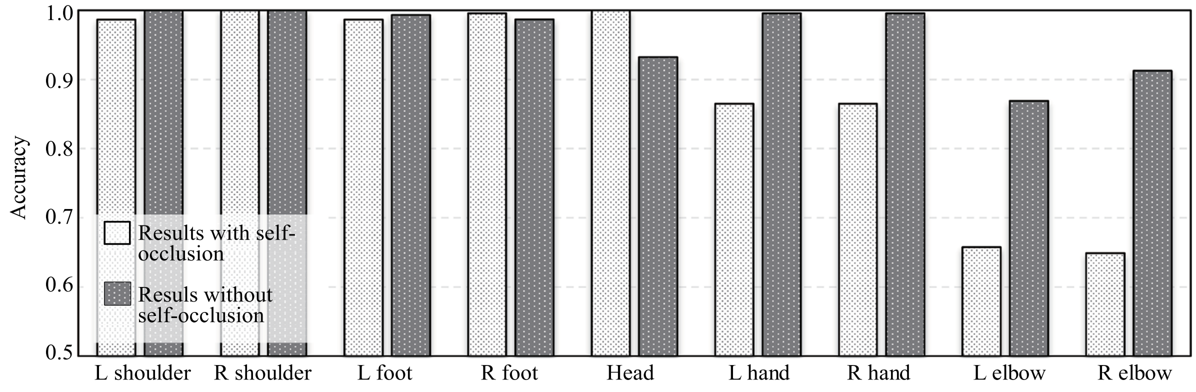


Fig. 10. A comparison of the average accuracy of feature point detection in the presence of self-occlusion (light bars) and without (dark bars).

a reasonable threshold is 4.5 cm.

We compare our method with the latest version of Microsoft Kinect SDK. Fig. 11 depicts a comparison of results from our method with those from the SDK. The left panels show our results and the right ones show results from the SDK. Fig. 11 (a) shows a pose that was correctly detected by both methods. Yet, the detection results of the elbows by the SDK were far off the correct locations. Fig. 11 (b) shows a case where the left elbow was completely missed by SDK. In contrast, our method correctly located the elbows in both cases.

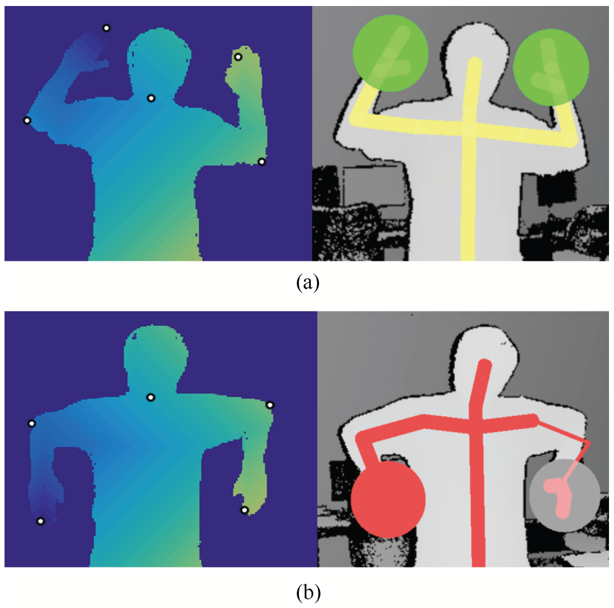


Fig. 11. A comparison with Microsoft Kinect SDK. Left: results from our method; Right: results from Microsoft Kinect SDK.

We examined the stability of the detections. Fig. 12 shows a set of frames when a human stands still. The joint position in the results of the SDK varied from frame to frame. The left elbow had significant changes. When the video is played in real-time, the detected elbows were constantly shaking. In contrast, our method delivered more stable results. It is demonstrated that our method delivers more stable detections. Using SMMC-10 data set, we compared the accuracy of our method with that of the methods by Ganapathi *et al.* [8] and Kim and Kim [14]. Fig. 13 shows the average accuracy of detecting feature points in video frames. In the detection of head, shoulders, and feet, all methods performed well. The average accuracies of these three categories are 99 %, 96 %, and 98.2 % for our method, method by Ganapathi *et al.* [8] and method by Kim and Kim [14], respectively. The accuracy in detection of hands by our method is significantly better than the others and the average accuracies are 97.7 %, 75.6 %, and 89.5 %. In contrast to the second best performer, our method improved the accuracy by 9 %. In the detection of elbows, our method achieved a competitive accuracy at 88.7 %. The best average accuracy by Kim and Kim [14] is at 94 %.

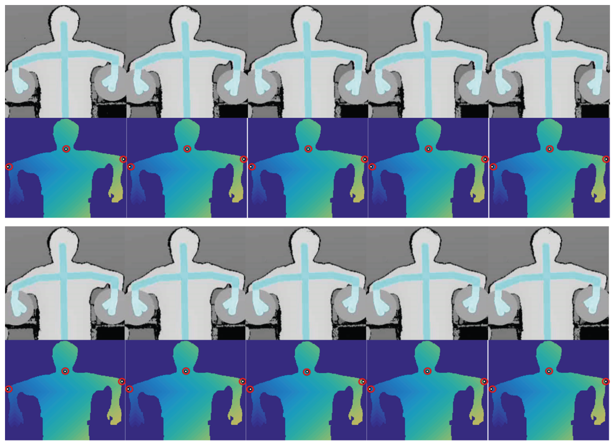


Fig. 12. Stability comparison with Microsoft Kinect SDK. Ten frames are split into two rows. The top panel in each row depicts results from the SDK and the bottom row depicts results from our method.

Using SMMC-10 data set, we compared the accuracy of our method with that of the methods by Ganapathi *et al.* [8] and Kim and Kim [14]. Fig. 13 shows the average accuracy of detecting feature points in video frames. In the detection of head, shoulders, and feet, all methods performed well. The average accuracies of these three categories are 99 %, 96 %, and 98.2 % for our method, method by Ganapathi *et al.* [8] and method by Kim and Kim [14], respectively. The accuracy in detection of hands by our method is significantly better than the others and the average accuracies are 97.7 %, 75.6 %, and 89.5 %. In contrast to the second best performer, our method improved the accuracy by 9 %. In the detection of elbows, our method achieved a competitive accuracy at 88.7 %. The best average accuracy by Kim and Kim [14] is at 94 %.

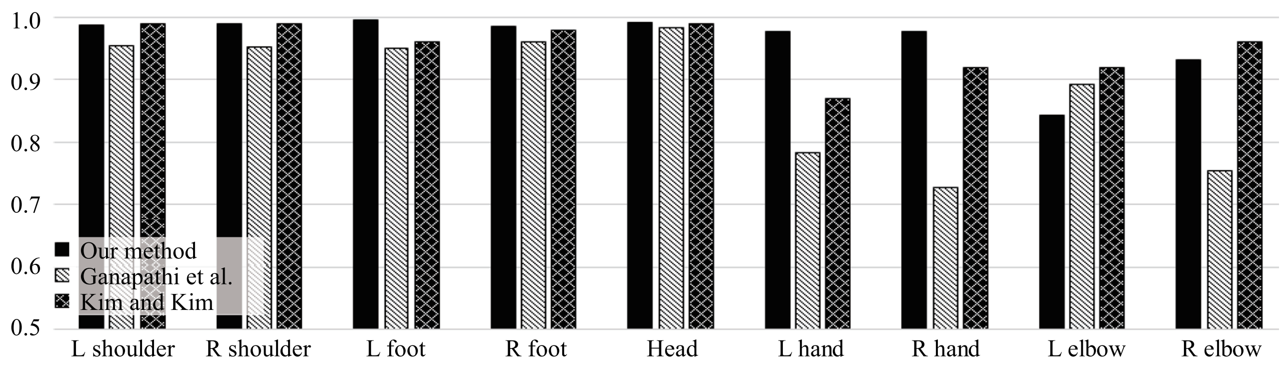


Fig. 13. A comparison of accuracy in feature point detection using SMMC-10 data set.

B. An Evaluation of Feature Point Refinement

To evaluate the robustness of our refinement method, we repeated the experiments with and without the refinement process based on the trajectory of the feature points. Fig. 14 shows exemplar cases of the refinement results. The green circle marks the initial detection of a body part and the yellow circle marks the refinement result. In these cases, the refinement successfully recovered the correct feature points.

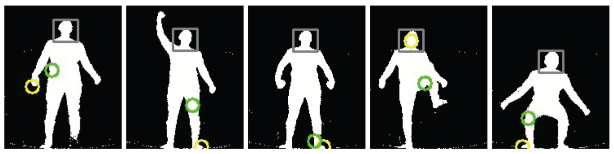


Fig. 14. Feature point refinement.

Fig. 15 shows the average accuracy of detection of different feature points with and without trajectory based refinement. The light shade bars are the average accuracy produced by our method with the refinement component; the dark shade bars are the average accuracy produced by our method without the refinement component. It is clear that the refinement process improved the accuracy and the overall improvement was 2.77 %. The accuracy of shoulder detection was least affected whereas the accuracy of elbows was improved by 6.69 % on average. The accuracy of feet detection had little improvement because there are very few frames showing complicated foot movement (e.g., stepping, jumping, and kicking). In cases when a refinement was used, the success rate of identifying the correct feature point is 93.7 %.

C. Efficiency Analysis

Our method was implemented with C++ and the experiments were conducted on a computer with an Intel dual core CPU at 3.4 GHz and 8 GB RAM running Windows 8.1. Table II lists the average processing time of each body part as well as a whole frame. The numbers within the parenthesis are the standard deviation. The overall average processing time for a frame was 71.1 ms, which is sufficient for processing real-time videos. The processing time for each component shows that detecting shoulders and the extreme points consumed most time. This is due to the calculation of the geodesic distance and searching for the shoulder points. Detecting elbows and refinement using trajectory is much faster.

TABLE II
FRAME PROCESSING TIME IN MILLISECOND

Steps	Ext. Pt.	Shoulder	Elbow	Refinement	Overall
Time	35.7	40.0	1.3	2.1	71.1
STD	(12)	(9.3)	(1.0)	(0.8)	(2.0)

V. CONCLUSION

We present a model-free method for detecting feature points from time-of-flight videos. Our method automatically detects the extreme points from the three-dimensional point cloud of a human figure based on geometric properties of the three dimensional silhouette. With a head-shoulder template, shoulders are detected, which serve as part of the references for the detection of elbows. The temporal locations of each feature point are used to form a profile to make prediction of its future location. This predicted location is then used as a constraint for refining the feature point detection.

The experimental results demonstrated that our proposed method achieved improved accuracy in comparison to the state-of-the-art methods including Microsoft Kinect SDK, methods by Ganapathi *et al.* [8] and Kim and Kim [14]. The overall accuracy of the feature point detection is 85.5 % based on our time-of-flight videos and 94.4 % based on the SMMC-10 video set. In the presence of occlusion, our method demonstrated exceptional robustness for all the extreme points and shoulder joints. The trajectory-based refinement also demonstrated satisfactory effectiveness that can recover the detection with a success rate of 93.7 %. Our method processed a frame in an average time of 71.1 ms. In our experiments, we observed that our method fails when body parts are invisible for a relatively long period. For instance, when a hand is occluded by the torso, our method resulted in an incorrect detection. We also found that the data acquired by depth cameras could be contaminated with noise from multi-path scattering, which appear as disproportionate measurements.

In our future work, we plan to improve joints' detection and address issues from complete occlusion of body parts for an extended period. The strategy for updating the distance map could be optimized as well. We currently update every node in the human body graph in four-neighbor iteration, which may cause error of feature point drifting.

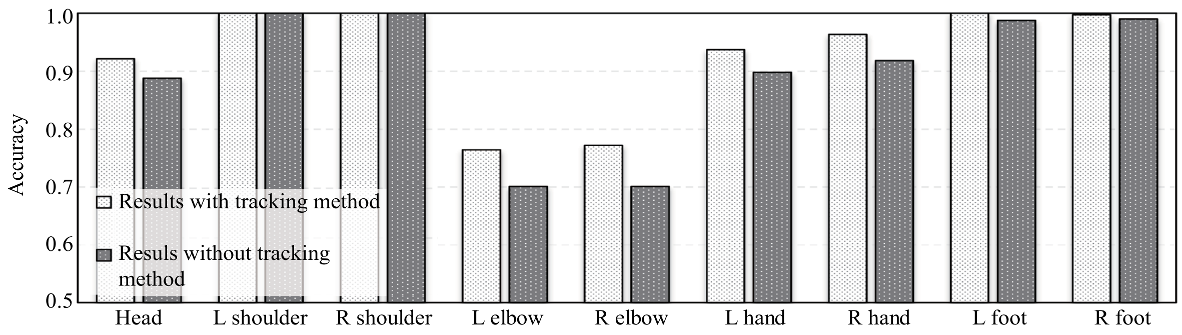


Fig. 15. Average accuracy of feature points with (light shade bars) and without (dark shade bars) the refinement component.

References

[1] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comp. Vis. Image Underst.*, vol. 104, no. 2−3, pp. 90−126, Nov.−Dec. 2006.

[2] J. X. Chen, S. Q. Nie, and Q. Ji, "Data-free prior model for upper body pose estimation and tracking," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4627−4639, Dec. 2013.

[3] A. Baak, M. Müller, G. Bharaj, H. P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Proc. IEEE Int. Conf. Computer Vision*, Barcelona, Spain, 2011, pp. 1092−1099.

[4] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 3108−3113.

[5] S. Handrich and A. Al-Hamadi, "A robust method for human pose estimation based on geodesic distance features," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Manchester, UK, 2013, pp. 906−911.

[6] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow," in *Proc. IEEE Int. Conf. Automatic Face & Gesture Recognition and Workshops*, Santa Barbara, CA, USA, 2011, pp. 700−706.

[7] P. C. Huang and S. K. Jeng, "Human body pose recognition from a single-view depth camera," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Seoul, Korea, 2012, pp. 2144−2149.

[8] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010, pp. 755−762.

[9] X. L. Wei, P. Z. Zhang, and J. X. Chai, "Accurate realtime full-body motion capture using a single depth camera," *ACM Trans. Graph. Proc. ACM SIGGRAPH Asia*, vol. 31, no. 6, pp. Article ID 188, Nov. 2012.

[10] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821−2840, Dec. 2013.

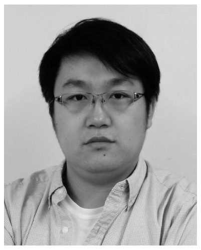
[11] E. J. Weng and L. C. Fu, "On-line human action recognition by combining joint tracking and key pose recognition," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vilamoura, Portugal, 2012, pp. 4112−4117.

[12] K. Greff, A. Brandão, S. Krauß, D. Stricker, and E. Clua, "A comparison between background subtraction algorithms using a consumer depth camera," in *Proc. Int. Conf. Computer Vision Theory and Applications*, Rome, Italy, 2012, pp. 431−436.

[13] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179−187, Feb. 1962.

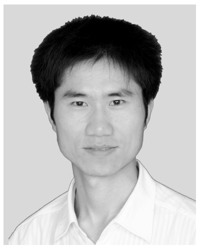
[14] Y. Kim and D. Kim, "Efficient body part tracking using ridge data and data pruning," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*, Seoul, Korea, pp. 114−120, 2015.

Xiaohui Yuan received the B.S. degree in electrical engineering from Hefei University of Technology, China in 1996 and the Ph.D. degree in computer science from Tulane University in 2004. He is an Associate Professor at the University of North Texas and a visiting Professor at the China University of Geosciences. He is the director of the Computer Vision and Intelligent Systems Laboratory at the University of North Texas. His research interests include computer vision, machine learning, image processing, data mining, and artificial intelligence. His research findings have been reported in over 100 peer-reviewed papers. He is a recipient of Ralph E. Powe Faculty award.



Longbo Kong received the bachelor of science degree in computer science in 2010 from Dalian University of Technology, China and received the master degree in computer science at the University of North Texas (UNT), USA. in 2013. He is currently the Ph.D. candidate at UNT. He is a member of the Computer Vision and Intelligent Systems Laboratory, and is working as a teaching assistant. His research interests include computer vision, data mining, and artificial intelligence.

Dengchao Feng graduated from Taiyuan University of Technology, China, in 1999. He received the M.S. degree from Taiyuan University of Technology in 2004 and the Ph.D. degree from Tianjin University, China, in 2008. He is currently an Associate Professor at the Laboratory of Remote Sensing, GNSS, GIS (3 S Lab) in North China Institute of Aerospace Engineering. His research interests include flight safety in low altitude airspace and intelligent information process.



Zhenchun Wei graduated from the Hefei University of Technology, China with a bachelor of science degree in computer science in 2000 and received the Ph.D. degree from the Hefei University of Technology, China in 2007. He is currently an Associate Professor with the School of Computer and Information at the Hefei University of Technology. His research interests include Internet of things, wireless sensor networks, machine learning, and distributed system.