Review Article

Real-Time Height Measurement for Moving Pedestrians

Wenju Zhou,¹ Fulong Yao,¹ Wei Feng ,² and Haikuan Wang¹

¹School of Mechanical Engineering and Automation, Shanghai University, Shanghai 200444, China

²Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

Correspondence should be addressed to Wei Feng; wei.feng@siat.ac.cn

Received 11 June 2020; Revised 27 July 2020; Accepted 18 August 2020; Published 27 August 2020

Academic Editor: Kailong Liu

Copyright © 2020 Wenju Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Height measurement for moving pedestrians is quite significant in many scenarios, such as pedestrian positioning, criminal suspect tracking, and virtual reality. Although some existing height measurement methods can detect the height of the static people, it is hard to measure height accurately for moving pedestrians. Considering the height fluctuations in dynamic situation, this paper proposes a real-time height measurement based on the Time-of-Flight (TOF) camera. Depth images in a continuous sequence are addressed to obtain the real-time height of the pedestrian with moving. Firstly, a normalization equation is presented to convert the depth image into the grey image for a lower time cost and better performance. Secondly, a difference-particle swarm optimization (D-PSO) algorithm is proposed to remove the complex background and reduce the noises. Thirdly, a segmentation algorithm based on the maximally stable extremal regions (MSERs) is introduced to extract the pedestrian head region. Then, a novel multilayer iterative average algorithm (MLIA) is developed for obtaining the height of dynamic pedestrians. Finally, Kalman filtering is used to improve the measurement accuracy by combining the current measurement and the height at the last moment. In addition, the VICON system is adopted as the ground truth to verify the proposed method, and the result shows that our method can accurately measure the real-time height of moving pedestrians.

1. Introduction

Whether in reality or in virtual scene, it is crucial to evaluate the height of moving pedestrians. Although there are many works related to dynamic pedestrians, such as detection and recognition [1–3], positioning [4–6], and tracking [7–9], it is still a serious challenge to measure the human height accurately in the dynamic case. As a vital state attribute of pedestrians, the height can not only help locate dynamic pedestrians or track criminal suspects in reality but also help people get rid of the 3D glasses or helmets in virtual scene [10]. For example, Nilsson et al. adopted the pedestrian height and positions of feet as constraint factors to design a Kalman filter model [11, 12], which can be used for pedestrian positioning and navigation.

In the past years, some height detection methods are developed. Chen et al. developed a novel action-based pedestrian recognition method [13], which could get the rough height. Also, the pneumatic sensor-based height measurement methods are developed to detect the pedestrian height

in literatures [14, 15]. However, these methods did not take height measurement as the main research content, which leads to an overall low accuracy. Besides, a significant issue is often ignored that the pedestrian height is changing when the pedestrian is walking, which may reduce the accuracy. There are few state-of-the-art motion tracking systems (MTS), such as VICON, which can precisely detect the real-time height of the dynamic pedestrian [16]. Sheng et al. used the VICON motion capture system, one of the popular spatial positioning systems in the world, to evaluate the human behaviours by analysing pedestrian attributes including height [17]. However, the MTS's costs for installation and maintenance are extremely expensive. Therefore, it is necessary to develop a cheap system to accurately measure the real-time height for moving pedestrians.

With the rapid development of the visual sensing technology, TOF camera is widely used in many fields, such as robot research [18–20], object detection [21–23], 3D reconstruction, and gesture recognition [24, 25], due to its compact structure and stable characteristics. In this paper,

the TOF camera is adopted as the input, and a real-time height measurement is developed for moving pedestrians; the flowchart of the measurement is shown in Figure 1. Each frame of depth image in a continuous sequence is addressed by the proposed algorithms to obtain the real-time height of the pedestrian with moving. The algorithm can be roughly divided into two steps: image processing and data processing.

Image processing is dedicated to extract the regions of interest (ROI)—head region. When the TOF camera is adopted, the depth value in depth images may be large due to a huge conversion ratio occurring between the actual distance in the world coordinate and the depth data in the image coordinate. Also, the gap between the maximum depth value and minimum depth value is massive, which is not conducive to the subsequent extraction of ROI. To reduce computation and improve the efficiency, a normalization equation is developed in the paper. Then, a D-PSO evolutionary algorithm is developed to reduce the effects of the complicated background. In the D-PSO, the difference part is dedicated to removing the complex background in surroundings, while the PSO part is responsible for the noises that appear after applying the difference part. After that, an MSER-based segmentation algorithm is adopted to extract the head region. Image processing is devoted to height calculation and correction. In this step, a novel multilayer iterative average algorithm based on the actual situation is proposed to remove the outliers and possible noises among the head data. Then, the pinhole model proposed in our previous work [26] is adopted to allow our method to work for pedestrians who are not vertically below the TOF camera. After that, considering the continuity of the height change of the moving pedestrian, Kalman filtering is adopted to combine the current measurement and previous height to improve the accuracy. In addition, the VICON system whose measurement accuracy reached 0.01 mm [27] is used as the ground truth to verify the proposed method.

To this end, our main contributions are listed as follows:

- (1) A real-time height detection method is developed for dynamic pedestrians. It considers the fluctuation of height while the pedestrian is walking, which is scarcely mentioned in the existing paper.
- (2) A new D-PSO algorithm is proposed to reduce the effects of the complicated background.
- (3) A novel multilayer iterative average algorithm is developed to remove the outliers and possible noises for a better performance.
- (4) Kalman filtering is used to improve the accuracy of the real-time height by combining current measurements and the data at last moment.

The rest of this paper is organised as follows. Section 2 presents a real-time extraction for head region. Section 3 shows a real-time correction and estimation for pedestrian height. Some experiments are introduced in Section 4 to show the feasibility and good performance of the proposed method, while the conclusion and further work are shown in Section 5.

2. Real-Time Extraction for the Pedestrian Head Region

The pedestrian head data are used in this paper to calculate the real-time height of dynamic pedestrians. Each frame of depth image in a continuous sequence is addressed by the following algorithms to extract the pedestrian head region.

2.1. Normalization. According to the imaging principle of the TOF camera, the depth value in depth image is large, and the gap among depth data is massive. Figures 2(a) and 2(b) show the depth images captured by TOF camera; the depth images are shown in Hue, Saturation, and Value (HSV) format for clarity; different colours represent different distances. Figure 2(a) is the background depth image captured in advance, while Figure 2(b) shows the depth image with the pedestrian.

To reduce computation and improve the efficiency of subsequent algorithms, the depth image is converted into the grey image by (1). Figures 2(c) and 2(d) are the grey image (image with pixel values between 0 and 255) corresponding to Figures 2(a) and 2(b), respectively:

$$p_i = 255 * \frac{d_i - d_{\min}}{d_{\max} - d_{\min}}, \quad (1)$$

where p_i is the pixel value of a point in the grey image corresponding to the depth value d_i in the depth image, and d_i is only related to the characteristics of the camera and the distance from the head to the camera; $d_{\max}$ and $d_{\min}$ are the maximum and minimum depth values in the depth image. Different depth values correspond to different pixel values between 0 and 255. The larger the depth value, the bigger the pixel value.

2.2. Difference-Particle Swarm Optimization (D-PSO) Denoising. It is obviously hard to obtain the accurate head data of the pedestrian due to disturbance from the complicated background. The difference algorithm, as shown in the following equation, is used in this paper to mitigate the effects of the complex background:

$$G_{\text{dif}} = G_{\text{back}} - G_{\text{ped}}, \quad (2)$$

where G_{back} shows the background grey image (such as in Figure 2(c)), G_{ped} represents the grey pedestrian image (such as in Figure 2(d)), and G_{dif} denotes the result of difference (such as in Figure 2(e)).

The difference algorithm produces a large amount of noises while extracting the human body region successfully. For clarity, the 3D perspective view of Figure 2(e) is shown in Figure 2(f). To eliminate the effect of noises, several common denoising algorithms have been applied with appropriate parameters. Figures 2(g)–2(j) show the results obtained by the common denoising algorithms along with the grey image (Figure 2(e)). To compare these algorithms clearly, the results of these algorithms are also shown here in 3D perspective view. In these figures, we can easily see the strength of the noise from the value of Pixel-axis. Therefore,

FIGURE 1: Overall framework of the proposed method.

FIGURE 2: Continued.

FIGURE 2: Representative images at each stage. (a, b) The depth images represented by HSV format. (c, d) The grey image corresponding to the depth images in (a, b), respectively. (e) The result of difference algorithm. (f) The 3D perspective view of the result in (e). The representative results (represented in 3D perspective view) using the different denoising algorithms: (g) average filtering, (h) median filtering, (i) Gaussian filtering, (j) bilateral filtering, and (k) two-stage PCA filtering. (l, m) The PSO denoising results in 3D and 2D, respectively. (n, o) The results obtained by the segmentation algorithm based on MSER.

the value of the Pixel-axis can be used as a criterion for evaluating the denoising effect. Although these algorithms can reduce the influence of noises to some extent, they may also blur the target contour and damage the pixel in head region, which is not conducive to the extraction of the head region. Figure 2(k) shows the result that is got by adopting the two-stage PCA filtering algorithm proposed in [28]. It can be seen from Figures 2(f)–2(k) that, compared to common filtering algorithms, PCA can reduce noise better and has little influence on target contour and head region. However, the average time consumed by the PCA algorithm is greater than 15 seconds, which is beyond our tolerance.

Particle swarm optimization (PSO), developed by Dr. Kenney and Dr. Eberhart [29], is an evolutionary algorithm

based on the study of bird or fish predation behaviour and mainly seeks an optimal global solution by following the searched optimal values of current particles [30]. Because of its fast speed, no need to manually set the threshold, etc.; it has been widely used in the field of image processing [31–33] and has achieved excellent results. Thus, PSO is adopted here to remove the background noises. In the PSO algorithm, each particle travels in a multidimensional search space and adjusts its position in search space based on the experience of itself and neighbouring particles [34]. The performance of each particle is evaluated by a predefined fitness function that encapsulates the core characteristics of the optimization problem.

In each iteration, every particle in the particle swarm gets its velocity and position by (3) and (4), respectively:

$$v_i^{k+1} = w^k v_i^k + c_1 r_1 (pbest_i - x_i^k) + c_2 r_2 (gbest - x_i^k), \quad (3)$$

$$x_i^{k+1} = x_i^k + v_i^k, \quad (4)$$

where k is the current number of iterations, x_i^k and v_i^k are, respectively, the position and velocity of the i th particle in the particle swarm during the k th iteration, r_1 and r_2 are two random numbers in $[0, 1]$, respectively, w^k is the inertia weight in the k th iteration, $pbest_i$ is the optimal solution available for the i th particle, $gbest$ is the optimal solution currently available for all particles, and c_1 and c_2 are individual learning factors and social learning factors, respectively, which are generally constant. As recommended by Dr. Kenney and Dr. Eberhart [29], we define learning factors $c_1 = c_2 = 2$. In this case, r_1 or r_2 multiplied by 2 to give it a mean of 1, PSO can well take into account both social learning and individual learning [35]. The scale of the particle swarm, called M , is directly related to the optimization result and time consumption. A small scale may cause the PSO to fail to find the optimal solution, and a large scale will cause unnecessary time costs [36]. Consider the two points; the particle swarm scale is defined as $M = 20$.

The larger the inertia weight w is, the stronger the global optimization ability is, and the weaker the local optimization ability is [37]. Otherwise, the local optimization ability is stronger. In order to strike a balance between search speed and search accuracy, w should not be a fixed constant. A nonlinear decreasing function for w is adopted in the paper, as shown in the following equation:

$$w^k = w_{\max} - (w_{\max} - w_{\min}) * \frac{1}{1 + a * b^{m * (k/k_{\max})}}, \quad (5)$$

where $w_{\max}$ and $w_{\min}$ are the predefined maximum and minimum inertia weights, respectively, k and $k_{\max}$ are the current and maximum number of iterations, and $m \in N^*$, $a > 0$, and $0 < b < 1$ are adjustment factors of the polynomial. After trial and error, we define $w_{\max} = 0.9$, $w_{\min} = 0.3$, $k_{\max} = 100$, $a = 2$, $b = 0.6$, and $m = 10$. The inertia weight curve corresponding to the above parameters is shown in Figure 3. It guarantees that PSO has a high global searchability in the early stage to get the appropriate seed and has higher local searchability in the later stage to improve the convergence accuracy.

Besides, we adopted the maximum interclass variance equation (6) as the fitness function in this paper. The larger the value of the fitness function is, the closer to the optimal solution it will be:

$$f = v_0 * v_1 * (u_0 - u_1)^2, \quad (6)$$

where v_0 and v_1 are, respectively, the proportion of the foreground and background images to the image; u_0 and u_1 represent, respectively, the average grayscale of the foreground and background images.

Figures 2(l) and 2(m) show the denoising results of PSO algorithm in 3D and 2D perspective view, respectively. Compared with other denoising algorithms, this algorithm can achieve better denoising effect without blurring the

target contour. In this section, a D-PSO is introduced to remove the complicated background. Compared with using the difference algorithm alone, D-PSO can not only remove the complex background in surroundings, but it can also reduce the noises that appear after applying the difference algorithm.

2.3. Head Segmentation Based on Maximally Stable Extremal Regions (MSER). When the TOF camera is used, the depth value for different parts of the pedestrian body varies greatly. In order to extract the head region, the maximally stable extremal regions (MSER) algorithm is used in the paper. The MSER algorithm refers to performing successive binarization operations on a picture; the binarization threshold is continuously increased from 0 to 255 [38]. If a connected region in the image is changed a little or even is not changed within a wide range of the binarization threshold, this region is called the maximum stable extreme region. Figure 2(n) shows the result obtained by the MSER along with Figure 2(m). In the figure, different connected regions are marked with different colours for clarity. It is obvious that MSER can separate different levels of pedestrian body parts.

Fortunately, regardless of the height and position of pedestrians, the head shapes of pedestrians are relatively stable ellipse, even for pedestrians without hair. Thus, the circularity is used as a constraint to get the head region. The circularity of each region is calculated by the following equation:

$$C = \frac{4\pi * A}{l^2}, \quad (7)$$

where C represents the circularity of the connected region, l represents the number of pixels in the boundary of the connected region, and A represents the number of pixels within the connected region.

The standard circularity is 1 and the circularity of other noncircular objects is less than 1. According to the experimental equipment and environment, we had an empirical conclusion that the circularity of head region is better between 0.6 and 1.0. If a connected region's circularity is beyond this range, it would be remarked as the non-head region and deleted. Due to the size of the pedestrian head in practice, the number of pixels A is used as another constraint condition. After repeated tests, we conclude that the A of head region should be during (300, 900). In other words, it is possible to be a head region only if the A of the connected region is within the range. As stated above, the constraints can be summarized in the following equation:

$$\begin{cases} 300 \leq A \leq 900, \\ C = \frac{4\pi * A}{l^2}, \\ 0.6 \leq C \leq 1.0. \end{cases} \quad (8)$$

By calculating and comparing the above two parameters of each connected region in Figure 2(n), the head region is

FIGURE 3: The curve of inertia weight in PSO.

extracted, as shown in the yellow part of Figure 2(o). Figure 4 is the pixel distribution map of the extracted head region, where the black dots represent pixel points, and the coordinates represent the positions of the pixels in the image. From this figure, we can discover another advantage of the proposed MSER-based segmentation algorithm, which can remove the notable noises in the head region, such as salt-and-pepper noise. Since the notable noise is very different from its neighbour pixels, it will not be incorporated into the head region when the MSER algorithm is used to obtain the stable region. Therefore, the MSER-based segmentation can effectively filter out notable noises in the head region, as shown in the red rectangles in Figure 4. Note that the red rectangles are the manual markers for easy viewing.

3. Real-Time Calculation for Pedestrian Height

3.1. Multilayer Iterative Average Algorithm for Pixel Value. Although the MSER algorithm can filter out the notable noises, there will still be some noises in the head region, as shown in the 3D representation of the head region in Figure 2(f). The typical height measurement of only using the head top is not accurate. Thus, a novel multilayer iterative average algorithm (MLIA) is proposed to get the pixel average for getting the pedestrian height. The MLIA algorithm not only can improve accuracy, but also can effectively remove some outliers that MSER cannot filter out. The MLIA can be broken down into the following steps:

- (1) Calculating the average of pixel value: adopting the following equation to get the average of pixel value in the head region, as

$$p_{ave} = \frac{1}{n} \sum_{i=1}^n p_i, \quad (9)$$

where p_{ave} is the pixel value average, n is the number of pixels in current head region, and p_i represents i -th pixel value in current head region.

- (2) Updating the head region: traverse all the pixels in the head region, and delete the pixels that do not meet the following equation. The remaining pixels are combined to update the head region:

FIGURE 4: The pixel distribution map of the extracted head region.

$$|p_i - p_{ave}| \leq T(p_{ave}), \quad (10)$$

where $T(p_{ave})$ is a threshold function related to the current average p_{ave} , and it is defined as follows:

$$T(p_{ave}) = \text{Min}(p_{\max} - p_{ave}, p_{ave} - p_{\min}), \quad (11)$$

where $p_{\max}$ and $p_{\min}$ are the maximum pixel value and the minimum pixel value in the head region, respectively.

- (3) Repeat step (1) and step (2) above until p_{ave} satisfy the following equation:

$$p_{ave} - \frac{p_{\min} + p_{\max}}{2} \leq \delta, \quad (12)$$

where δ is the empirical constant. In this paper, δ is selected as 2.0 according to the actual situation.

The above steps can be summarized as the following pseudocode (Algorithm 1).

By the way, the MLIA algorithm can also be applied to the multipedestrian situation. When the image contains more than one pedestrian, the MSER-based segmentation can get more than one head region. Meanwhile, the pixel value average of each head region needs to be calculated by the MLIA algorithm.

3.2. Height Calculation. Once p_{ave} is obtained, the average of the head region in original pedestrian grey image (such as in Figure 2(d)), defined as p_{avg} , can be obtained through the deformation of (2).

Then, substituting p_{avg} into (1) to replace p_i , we can obtain the following equation:

Input: S-initial head region extracted by the MSER-based segmentation.

Procedure:

- (1) $n = \text{count}(S)$;
- (2) $p_{\text{ave}} = (1/n) \sum_{i=1}^n p_i, p_i \in S$;
- (3) $p_{\text{max}} = \text{Max}(p_i), i = 1, 2, \dots, n$;
- (4) $p_{\text{min}} = \text{Min}(p_i), i = 1, 2, \dots, n$;
- (5) while $p_{\text{ave}} - ((p_{\text{min}} + p_{\text{max}})/2) > \delta$ do
- (7) $T(p_{\text{ave}}) = \min(p_{\text{max}} - p_{\text{ave}}, p_{\text{ave}} - p_{\text{min}})$;
- (8) $S = \{p_i \mid |p_i - p_{\text{ave}}| \leq T(p_{\text{ave}}), i = 1, 2, \dots, n\}$;
- (9) $n = \text{count}(S)$;
- (10) $p_{\text{ave}} = (1/n) \sum_{i=1}^n p_i, p_i \in S$
- (11) $p_{\text{max}} = \max(p_i), i = 1, 2, \dots, n$;
- (12) $p_{\text{min}} = \min(p_i), i = 1, 2, \dots, n$;
- (13) end while

Output: p_{ave} -the average of the pixels in the head region.

ALGORITHM 1: Multilayer iterative average algorithm (MLIA).

$$d_{\text{avg}} = \frac{p_{\text{avg}}(d_{\text{max}} - d_{\text{min}})}{255} + d_{\text{min}}, \quad (13)$$

where d_{avg} is the depth value corresponding to p_{avg} and d_{max} and d_{min} are the maximum and minimum depth values in the pedestrian depth image.

According to the physical properties of the TOF camera, the following conversion equation can be used to recover the physical distance from the depth data d_{avg} [39]:

$$D_{\text{dis}} = K_{\text{tof}} d_{\text{avg}} + E = K_{\text{tof}} \left(\frac{p_{\text{avg}}(d_{\text{max}} - d_{\text{min}})}{255} + d_{\text{min}} \right) + E, \quad (14)$$

where D_{dis} represents the physical distance between the TOF camera and the pedestrian head (unit: mm), E is the deviation constant associated with the physical structure and placement height of the TOF camera, while $K_{\text{tof}} = (5/12)$ is the conversion coefficient only associated with the physical structure of the TOF camera.

To allow our method to work for pedestrians who are not vertically below the TOF camera, the pinhole model proposed in our previous work [26] is adopted to correct D_{dis} :

$$D_{\text{co}} = D_{\text{dis}} \times \cos\left(\arctan\left(\frac{OM}{f}\right)\right), \quad (15)$$

where D_{co} is the corrected physical distance, f is the focal length, and OM is the distance between the centroid of the head region in the grey image M and the centre of the grey image O ; the coordinates of the centroid M can be got by the following equation. More detailed information about the pinhole model can be found in the literature [26]:

$$\begin{cases} h_p = \frac{1}{n} \sum_{i=1}^n m_i h_i, \\ w_p = \frac{1}{n} \sum_{i=1}^n m_i w_i, \end{cases} \quad (16)$$

where n is the number of pixels in current head region, w_p and h_p are the horizontal and vertical coordinates of the centroid M , and w_i and h_i are the horizontal and vertical coordinates of the i th pixel, respectively. m_i is the mass of the i th pixel, which is defined as $m_i = 1$ in this paper.

Finally, the pedestrian height H is calculated by the following equation:

$$H = H_{\text{tof}} - D_{\text{co}}, \quad (17)$$

where H_{tof} is the distance between the TOF camera and the ground.

3.3. Kalman Estimation of Real-Time Height. In the experiments, we found that the fluctuations of the pedestrian heights all approximately conform to the Gaussian distribution with variance 256 (unit: mm^2), and the variance did not change with the state of the system. Therefore, Kalman filtering is further introduced to estimate the pedestrian heights got by (17) to achieve the more accurate real-time heights. Kalman filtering is a highly efficient recursive filter that can estimate the state of a dynamic system from a series of measurements containing redundant noise [40]. It can generate estimates of unknown variables, which have proven to be more accurate than those only based on a single measurement [4, 41]. The Kalman filter can be implemented in two stages: time update stage and measurement update stage [42].

The time update stage is dedicated to predicting the currently a priori estimates through past state and the error covariance. Equations (18) and (19) are responsible for predicting the a priori state estimate $\hat{x}_k$ and the a priori error covariance estimate $\hat{P}_k$ in current (k th) frame, respectively:

$$\hat{x}_k = A_{k-1} x_{k-1} + B u_{k-1}, \quad (18)$$

$$\hat{P}_k = A_{k-1} P_{k-1} A_{k-1}^T + Q, \quad (19)$$

where x_{k-1} and P_{k-1} are, respectively, the state and the error covariance of the previous step, A_{k-1} is the transfer matrix that relates the state of the previous step to the state of the

current step, B is the control matrix that relates the previous input u_{k-1} , and Q is the variance of the Gaussian process noise. Based on the actual situation of pedestrians during the movement (no external input, Gaussian distribution of the height fluctuation, and continuity of the height change), the parameters in time update stage are defined as follows: $u_{k-1} \equiv 0$, $Q \equiv 256$, $A_{k-1} \equiv 1$; $\hat{x}_k$ is the a priori height estimate from the current depth image.

The measurement update stage is devoted to combining actual measurements with a priori estimates to get the improved posteriori estimates [42]. It can be achieved by the following equations:

$$K_k = \hat{P}_k H_k^T (H_k \hat{P}_k H_k^T + R)^{-1}, \quad (20)$$

$$x_k = \hat{x}_k + K (Z_k - H_k \hat{x}_k), \quad (21)$$

$$P_k = (I - K H_k) \hat{P}_k, \quad (22)$$

where x_k and P_k are the posteriori state estimate and the posteriori error covariance estimate in current (k th) step, K_k is the Kalman gain in current step, H_k is the matrix that relates the state to the measurement Z_k , I is a unit matrix, and R is the variance of the Gaussian measurement noise. Based on the actual situation of measurements (camera accuracy and measurement process), the parameters in measurement update stage are defined as follows: $R \equiv 144$, $H_k \equiv 1$; x_k is the posteriori height estimate from the current depth image, and Z_k is the pedestrian heights got by (17). In addition, the initialization is defined as $x_1 = Z_1$ and $P_1 = 10$.

4. Experiments and Analysis

4.1. Experimental Setup. In this paper, an EPC660 is used as the TOF chip to offer a fully digital interface for the control circuitry, and the communication between computer and camera is realized through Gigabit network. In addition, the experiment is completed with the support of the computer with Windows 10 OS, Intel® Core™ i3-8100 3.60 GHz CPU and 8 GB RAM. The campus corridor is selected as the first test site, and the experimental scene is shown in Figure 5(a). Then, considering the fluctuation of pedestrian height in dynamic situations, the research room is chosen as the second test site, and the VICON system fixed in this site is adopted as the ground truth to confirm the feasibility of the proposed method. The experimental scene in research room is shown in Figure 5(b), where a portion of the VICON system, two of the 12 infrared cameras, is shown. While the VICON is running, four lightweight reflective balls are stuck to the pedestrian's head; the placement layout of the balls is shown in Figure 5(c). And the average height of the four balls is adopted as the real-time height of the pedestrian.

4.2. Comparison with Other Popular Algorithms. Before the PSO algorithm is adopted to process the images with unwanted noise, other popular algorithms are deployed to process the same images for a comparison. More specifically, three algorithms are implemented for comparison here:

- (1) *Maximum Connected Region (MCR)*. As the name implies, MCR refers to the method of extracting the largest connected region in an image. When only a single person appears in the field of view, such as in Figure 2(e), MCR is more likely to get desirable results than PSO. In the actual situation, however, we do not know in advance how many people will go through the test site. Take Figure 6(a) as an example; when two people go through the test site at the same time, MCR may get a wrong result, as shown in Figures 6(b) and 6(c).
- (2) *Edge Threshold Method (ETM)*. In ETM, the edge operators such as Canny is firstly used to obtain the possible target contours, and the number of pixels in these contours is then calculated, respectively. Once the number is bigger than a specific threshold, the region enclosed by the corresponding contour is considered as the useful region and is retained; otherwise, this region is considered as the useless region and is removed. In the paper, the boundary between the target person and the redundant noise is usually solid, which makes it possible to split the target from the background with the ETM. More importantly, the ETM can also get good results in multipedestrian images with appropriate parameters. However, it is a very difficult task for the ETM to adaptively select parameters. Once the test environment changes, the parameters of ETM need to be reselected, which limits the application of the ETM.
- (3) *Reaction Diffusion-Level Set Evolution (RD-LSE)*. The RD-LES proposed by Zhang et al. [43] is an improved level set algorithm, which is widely used in the field of image segmentation. Figure 6(d) shows the search process using the RD-LSE algorithm for the Figure 6(a), in which the yellow curves show the evolution processes, the green curve represents the initial contour, and the red curve represents the final contour. This algorithm can achieve a better result than PSO algorithm even in the case of multiple pedestrians, as shown in Figures 6(e) and 6(f). In the paper, we take 4 different types of pictures as examples, to compare the performance of RD-LES and PSO in terms of converged iterations and CPU time. The experimental results are shown in Table 1, where images 1–4 represent Figures 2(e), 6(a), 7(a) and 7(g), respectively. The values in table are the average of 100 experiments. Table 1 shows that the computational efficiency of the PSO algorithm far exceeds the RD-LES, which is the main reason why we choose PSO.

4.3. Experimental Results. Apart from the multipedestrian cases such as in Figure 6, many other cases with the pedestrian in different states are studied to verify the effectiveness and robustness of the proposed method. In Figure 7(a), the pedestrian raised his left hand above his head. Figures 7(c), 7(e), and 7(f) show the experimental process and result of adopting the proposed method for Figure 7(a). For clarity, the 3D

FIGURE 5: Experimental setup. (a) Site: campus corridor. (b) Site: research room. (c) Placement layout of the lightweight reflective balls.

FIGURE 6: Experiments with two-pedestrian image. (a) Original image with two pedestrians. (b) Image obtained by the MCR algorithm along with the original image. (c) Image obtained by the MSER-based segmentation along with (b). (d, e) The processes and result images obtained by the RD-LSE along with the original image. (f) Image obtained by the MSER-based segmentation along with that in (e).

TABLE 1: Iterations (Iter) and CPU time (Time) by FRFLS and PSO methods.

Methods	Image 1		Image 2		Image 3		Image 4	
	Time (s)	Iter	Time (s)	Iter	Time (s)	Iter	Time (s)	Iter
FRFLS	5.21	643	6.01	800	5.32	665	4.87	611
PSO	0.049	12.9	0.057	8.6	0.053	18.5	0.050	14.7

Image size: 320 * 240 pixels.

FIGURE 7: Experiments with the pedestrian in different states. (a) Original image with the pedestrian raising his left hand. (c) Image obtained by the PSO algorithm along with that in (a). (e, f) Images obtained by the MSER-based segmentation along with that in (c). (b, d) The 3D representation of images in (a, c), respectively. (g) Original image with the pedestrian who is kneeling. (i) Image obtained by the PSO algorithm along with that in (g). (k, l) Images obtained by the MSER-based segmentation along with those in (i). (h, j) The 3D representation of images in (g, i), respectively.

representations of Figures 7(a) and 7(c) are shown in Figures 7(b) and 7(d), respectively. Although the height of the head is lower than that of the left hand, the proposed method can still get the correct result. Figures 7(i), 7(k), and 7(l) show the experimental process and result of adopting the proposed

method for Figure 7(g), in which a pedestrian is kneeling. Although the proposed D-PSO algorithm does not eliminate all redundant noises, as shown in Figure 7(j), it also yields ideal experimental results due to MSER's insensitivity to a small amount of the sporadic noise. All the above experiments

FIGURE 8: The height results got from the six subjects using the VICON alone in several continuous seconds.

FIGURE 9: Experimental results of men with different heights in the six consecutive seconds.

show that the performance of our method is very stable and reliable.

To further verify the accuracy of the proposed method, a lot of experiments are conducted based on 6 subjects: four men and two women, who are asked to walk through the test sites at the usual speed. Here we take a set of data obtained from the research room as an example to analyse the results. Figure 8 shows the height results obtained from the six subjects using the VICON alone in several continuous seconds; the sex and static height of the six subjects are presented in the legend. It explains that it is unrealistic to

keep the height on the static level when the pedestrian is walking. Thus, it is essential to study the pedestrian height in the dynamic situation.

Due to the high speed of pictures taken by VICON and TOF cameras and the slowness of pedestrian movement (0.7–1.2 meters per second), we only select 5 height data per second to show a real-time height comparison between the VICON and the proposed method. Every fifth of one second, an image is collected with the TOF camera. The pedestrian height in the image is obtained by the proposed method and compared with the height collected with VICON at the same

FIGURE 10: Experimental results of women with different heights in the six consecutive seconds.

(a)

(b)

(c)

(d)

FIGURE 11: The error plot of men in the six consecutive seconds. (a-d) The men with static heights of 1760, 1676, 1761, and 1728.

FIGURE 12: The error plot of women in the six consecutive seconds. (a) The woman with static height of 1648. (b) The woman with static height of 1629.

TABLE 2: The sum of errors per second of the algorithms with and without Kalman filtering.

Heights (mm)	Kalman filtering	Sum of errors per second (%)						Sum
		1st second	2nd second	3rd second	4th second	5th second	6th second	
1760*	Yes	1.202	0.956	1.836	1.242	1.611	1.525	8.372
	No	1.868	1.003	2.013	1.362	1.898	1.758	9.902
1676*	Yes	2.002	1.799	1.977	0.863	1.648	2.137	10.426
	No	2.249	1.968	2.087	1.602	1.827	3.261	12.994
1761*	Yes	1.282	1.483	0.963	1.132	0.632	1.487	6.979
	No	1.562	1.702	1.333	1.617	1.234	1.714	9.162
1728*	Yes	1.629	1.652	1.354	1.453	1.224	0.902	8.214
	No	2.201	2.159	1.912	1.592	1.984	1.336	11.184
1648#	Yes	2.006	1.194	1.818	1.014	1.585	1.693	9.310
	No	2.488	1.245	2.152	1.906	2.078	2.087	11.956
1629#	Yes	1.509	1.838	0.652	2.344	1.398	1.109	8.850
	No	1.632	2.536	1.328	2.508	1.497	1.340	10.841

*Male; #female.

time. Figures 9 and 10 show the experimental results of four men and two women in six consecutive seconds. In the figures, the dotted line represents our algorithm without Kalman filtering, the solid line represents our algorithm with Kalman filtering, and the dotted line with the mark “+” indicates the VICON. The waveforms show the real-time height value in 6 consecutive seconds; the static heights of men are 1760 mm, 1676 mm, 1761 mm, and 1728 mm, as shown in the legend of Figure 9, while the static heights of women are 1648 mm and 1629 mm, as shown in Figure 10.

It can be seen from the curves that the height data measured by our algorithm is almost consistent with the data obtained by VICON. In order to analyse the error of our algorithm, we sort out the errors of all the data in the six consecutive seconds; the results are shown in Figures 11 and 12. The figures show that Kalman filtering can effectively improve the accuracy of height measurement, which indicates the pedestrian height at the preceding moment facilitates the estimate of the pedestrian height in the latter moment.

Also, the sums of errors per second of the algorithms with and without Kalman filtering are given in Table 2 where the subscript “*” represents male and “#” represents female. Table 2 shows that our algorithm with Kalman filtering has a smaller cumulative error and can more accurately measure the real-time height of the moving pedestrians, which proves the feasibility and validity of the proposed method.

5. Conclusion and Future Work

In this paper, a real-time height measurement based on the TOF camera is proposed for moving pedestrians. To get the target region, a new D-PSO denoising algorithm and a segmentation algorithm based on MSER are developed in the paper. In addition, a novel multilayer iterative average algorithm is designed for calculating the pedestrian height. Also, the Kalman filtering is used to improve the measurement accuracy. The experimental results demonstrate the effectiveness and practicability of

the proposed method. Our future work is going to further improve the measurement accuracy and focus on tracking pedestrians in real time by using the real-time height of moving pedestrians.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are grateful to the financial support from the Natural Science Foundation of China (61877065), the National Key Research and Development Program of China (2019YFB1405500), the National Natural Science Foundation of Guangdong (2016A030313177), Guangdong Frontier and Key Technological Innovation (2017B090910013), and the Science and Technology Innovation Commission of Shenzhen (JCYJ20170818153048647 and JCYJ20180507182239617).

References

- [1] J. Li, X. Liang, S. Shen et al., "Scale-aware fast R-CNN for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2017.
- [2] F. P. An, "Pedestrian re-recognition algorithm based on optimization deep learning-sequence memory model," *Complexity*, vol. 2019, Article ID 5069026, 16 pages, 2019.
- [3] J. Cao, Y. Pang, and X. Li, "Learning multilayer channel features for pedestrian detection," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3210–3220, 2017.
- [4] M. Ji, J. Liu, X. Xu, Y. Guo, and Z. Lu, "Improved pedestrian positioning with inertial sensor based on adaptive gradient descent and double-constrained extended kalman filter," *Complexity*, vol. 2020, Article ID 4361812, 11 pages, 2020.
- [5] C. Li, Z. Su, Q. Li, and H. Zhao, "An indoor positioning error correction method of pedestrian multi-motions recognized by hybrid-orders fraction domain transformation," *IEEE Access*, vol. 7, pp. 11360–11377, 2019.
- [6] H. Zhao, W. Cheng, N. Yang et al., "Smartphone-based 3D indoor pedestrian positioning through multi-modal data fusion," *Sensors*, vol. 19, no. 20, Article ID s19204554, 2019.
- [7] B. Wang, T. Su, X. Jin, J. Kong, and Y. Bai, "3D reconstruction of pedestrian trajectory with moving direction learning and optimal gait recognition," *Complexity*, vol. 2018, Article ID 8735846, 10 pages, 2018.
- [8] Y. Jiang, Z. Li, and J. B. Wang, "Ptrack: enhancing the applicability of pedestrian tracking with wearables," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 431–443, 2018.
- [9] W. Xu, L. Liu, S. Zlatanova, W. Penard, and Q. Xiong, "A pedestrian tracking algorithm using grid-based indoor model," *Automation in Construction*, vol. 92, pp. 173–187, 2018.
- [10] L. Bozgeyikli, A. Raij, S. Katkoori, and R. Alqasemi, "A survey on virtual reality for individuals with autism spectrum disorder: design considerations," *IEEE Transactions on Learning Technologies*, vol. 11, no. 2, pp. 133–151, 2017.
- [11] J. O. Nilsson, D. Zachariah, I. Skog, and P. Händel, "Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 164, 2013.
- [12] I. Skog, J.-O. Nilsson, D. Zachariah, and P. Handel, "Fusing the information from two navigation systems using an upper bound on their maximum spatial separation," in *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation*, Article ID 6418862, Sydney, Australia, November 2012.
- [13] S.-B. Chen, Y. Xin, and B. Luo, "Action-based pedestrian identification via hierarchical matching pursuit and order preserving sparse coding," *Cognitive Computation*, vol. 8, no. 5, pp. 797–805, 2016.
- [14] B. Shin, C. Kim, J. Kim et al., "Motion recognition based 3D pedestrian navigation system using smartphone," *IEEE Sensors Journal*, vol. 16, no. 18, pp. 6977–6989, 2016.
- [15] M. Romanovas, V. Goridko, A. Al-Jawad et al., "A study on indoor pedestrian localization algorithms with foot-mounted sensors," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10, Sydney, Australia, November 2012.
- [16] A. Azaman, "Comparative study on gait kinematics between microsoft kinect and vicon across different anthropometric measurements," *Journal of Tomography System and Sensor Application*, vol. 2, no. 2, pp. 12–17, 2019.
- [17] W. Sheng, A. Thobbi, and Y. Gu, "An integrated framework for human-robot collaborative manipulation," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2030–2041, 2014.
- [18] S. Tsuji and T. Kohama, "Proximity skin sensor using time-of-flight sensor for human collaborative robot," *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5859–5864, 2019.
- [19] C. Oprea, I. Pirnig, I. Marcu, and M. Udrea, "Robust pose estimation using Time-of-Flight imaging," in *Proceedings of the IEEE International Semiconductor Conference*, pp. 301–304, Sinaia, Romania, January 2019.
- [20] A. Vysocký, R. Pastor, and P. Novák, "Interaction with collaborative robot using 2D and TOF camera," in *International Conference on Modelling and Simulation for Autonomous Systems*, pp. 477–489, Springer, Cham, Switzerland, 2018.
- [21] M. Gao, Y. Du, Y. Yang, and J. Zhang, "Adaptive anchor box mechanism to improve the accuracy in the object detection system," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27383–27402, 2019.
- [22] A. Anwer, S. S. Azhar Ali, A. Khan, and F. Mériaudeau, "Underwater 3-d scene reconstruction using kinect v2 based on physical models for refraction and time of flight correction," *IEEE Access*, vol. 5, pp. 15960–15970, 2017.
- [23] A. R. García, L. R. Miller, C. F. Andrés, and P. J. N. Lorente, "Obstacle detection using a time of flight range camera," in *Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 1–6, Madrid, Spain, September 2018.
- [24] N. Zengeler, T. Kopinski, and U. Handmann, "Hand gesture recognition in automotive human-machine interaction using depth cameras," *Sensors*, vol. 19, no. 1, Article ID s19010059, 2019.
- [25] M. A. Garduño-Ramón, I. R. Terol-Villalobos, R. A. Osornio-Rios, and L. A. Morales-Hernandez, "A new method for inpainting of depth maps from time-of-flight sensors based on a modified closing by reconstruction algorithm," *Journal of*

- Visual Communication and Image Representation*, vol. 47, pp. 36–47, 2019.
- [26] L. Wang, Y. Luo, H. Wang, and M. Fei, “Measurement error correction model of TOF depth camera,” *Chinese Journal of System Simulation*, vol. 29, no. 10, pp. 2323–2329, 2017.
 - [27] VICON, “Official website of oxford metrics company,” 2020, <https://www.vicon.com/>.
 - [28] L. Zhang, W. Dong, D. Zhang, and G. Shi, “Two-stage image denoising by principal component analysis with local pixel grouping,” *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.
 - [29] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-International Conference on Neural Networks (ICW)*, vol. 4, pp. 1942–1948, Perth, Australia, November 1995.
 - [30] M. A. M. De Oca, T. Stutzle, M. Birattari, and M. Dorigo, “Frankenstein’s PSO: a composite particle swarm optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, 2009.
 - [31] Z. Zhen, S. Pang, F. Wang et al., “Pattern classification and PSO optimal weights based sky images cloud motion speed calculation method for solar PV power forecasting,” *IEEE Transactions on Industry Applications*, vol. 55, no. 4, pp. 3331–3342, 2019.
 - [32] X. Wang, J.-S. Pan, and S.-C. Chu, “A parallel multi-verse optimizer for application in multilevel image segmentation,” *IEEE Access*, vol. 8, pp. 32018–32030, 2020.
 - [33] Z. A. Bashir and M. E. El-Hawary, “Applying wavelets to short-term load forecasting using PSO-based neural networks,” *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 20–27, 2009.
 - [34] L. Liu, Y. Wang, F. Xie, and J. Gao, “Legendre cooperative PSO strategies for trajectory optimization,” *Complexity*, vol. 2018, Article ID 5036791, 13 pages, 2018.
 - [35] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings*, pp. 69–73, Anchorage, AK, USA, May 1998.
 - [36] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” in *International Conference on Evolutionary Programming*, pp. 591–600, Springer, Berlin, Germany, 1998.
 - [37] X. Lv, D. Zhou, Y. Tang, and L. Ma, “An improved test selection optimization model based on fault ambiguity group isolation and chaotic discrete PSO,” *Complexity*, vol. 2018, Article ID 3942723, 10 pages, 2018.
 - [38] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
 - [39] H. Shim and S. Lee, “Recovering translucent objects using a single time-of-flight depth camera,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 841–854, 2015.
 - [40] C. K. Chui and G. Chen, *Kalman Filtering*, pp. 19–26, Springer International Publishing, Berlin, Germany, 2017.
 - [41] L. Cui, X. Wang, Y. Xu, H. Jiang, and J. Zhou, “A novel switching unscented Kalman filter method for remaining useful life prediction of rolling bearing,” *Measurement*, vol. 135, pp. 678–684, 2019.
 - [42] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, Macmillan, New York, NY, USA, 1995.
 - [43] K. Zhang, L. Zhang, H. Song, and D. Zhang, “Reinitialization-free level set evolution via reaction diffusion,” *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 258–271, 2012.