Sensors and Materials, Vol. 36, No. 5 (2024) 1881–1903
MYU Tokyo

S & M 3642

Vision System with Personnel Recognition and Tracking Functions for Use in a Space with Multiple Persons

Neng-Sheng Pai, Lian-Sheng Hong, Pi-Yun Chen,*
Zi-Heng Zhong, and Wei-Zhe Huang

Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

(Received May 25, 2023; accepted December 19, 2023)

Keywords: depth camera, space with multiple people, personnel tracking, unscented Kalman filter (UKF), FaceNet, decision tree

In this study, we propose a vision system for personnel recognition and tracking in a multiple-persons space. The personnel tracking function can be implemented in service vehicles or medical systems to identify specific users, thus facilitating service tasks such as follow-up care or support. User identification and tracking processes become highly challenging in environments that contain multiple people who are not wearing any external sensors. The proposed system involves three steps. First, the You Only Look Once version 4 (YOLOv4)-tiny model of object detection is used to extract a personnel-bounding box from an image. Second, the image coordinates and camera coordinates are converted into three-dimensional (3D) space coordinates on the basis of a depth map derived from a binocular camera to obtain 3D spatial information on the personnel. Finally, the unscented Kalman filter (UKF), FaceNet, or a combination of the UKF and FaceNet (UKF–FaceNet) produced through decision tree fusion, is used to identify or track the target personnel. Experimental results indicated that the proposed system can successfully track personnel. The UKF–FaceNet method can effectively improve the drawbacks of UKF- and FaceNet-based tracking. The UKF–FaceNet method can be adopted to achieve accurate and stable personnel identification in a multiple-persons environment. This method can reidentify and retrack a target user accurately when the user is obscured.

1. Introduction

Advances in deep learning and hardware have enabled the rapid growth of visual analytics technology, which is gradually replacing other sensing technologies. Such advances have also facilitated the development of low-power camera vision systems at a relatively low cost and with less installation difficulty. The identification and tracking of people by using images have potential uses in different applications, such as automatic driving assistance systems, service vehicles, human–machine interaction, and medical monitoring;[1–3] nevertheless, personnel identification and tracking are challenging tasks. Hence, developing a robust system for these tasks is imperative.

*Corresponding author: e-mail: chenby@ncut.edu.tw
https://doi.org/10.18494/SAM4527

ISSN 0914-4935

In this study, we propose a system that can accurately identify personnel in environments with multiple people. The proposed system can be implemented in service vehicles[4] and medical monitoring systems. This system can provide the information required by a vehicle to track a user and can be implemented in applications such as fall detection. Thus, this system can identify and interact with objects; object identification and interaction are the basic functionalities required in many service-related tasks. In traditional methods of tracking personnel in a space with multiple people, users are required to wear certain equipment or specific markers.[5–8] By contrast, the proposed system applies a purely visual approach, without the need for any equipment or markers, to track personnel. Thus, this system can increase the practicability of personnel tracking in environments with multiple people.

The proposed system involves three steps. First, all personnel in a space with multiple people are detected using depth camera images. Second, the image coordinates and camera coordinates are converted into three-dimensional (3D) spatial information in accordance with a depth map created using a binocular camera. Finally, the unscented Kalman filter (UKF),[9,10] FaceNet,[11] or a combination of the UKF and FaceNet (UKF–FaceNet) produced through decision tree fusion, is used to identify or track the target personnel.

The tracking of people in images using a Kalman filter[12,13] has been well researched and is one of the more commonly implemented solutions. Although extant systems provide a smooth tracking path by using recursive filtering to detect changes in object motion, they cannot predict target objects, and this could impair their accuracy in tracking target objects. To address this limitation, we implemented neural networks in the proposed system to enable personnel identification[14] through face recognition, thus ensuring accurate personnel tracking. Finally, we integrated the UKF and FaceNet through decision tree fusion to leverage their advantages while overcoming their drawbacks, yielding a UKF–FaceNet method suitable for identifying and tracking specific individuals in a multiple-persons space.

2. Related Work

Here, we review the literature on sensors, object detection, the UKF, and face recognition.

2.1 Sensors

Visual sensors are cheap and easy to install. In the proposed system, a ZED binocular camera[15] is adopted as the visual sensor. This camera performs passive distance measurements to calculate depth values. Although this camera is less effective in environments with low levels of ambient light, it has an effective distance of 20 m and a range of up to 40 m; thus, it can be effectively used in indoor and outdoor environments.

2.2 Object detection

The proposed system uses the You Only Look Once version 4 (YOLOv4)-tiny model[16] to perform personnel detection. YOLOv4-tiny, which is a simplified version of YOLOv4 proposed

by Bochkovskiy *et al.*,[16] can simultaneously predict the location of and classify the target personnel. The detection accuracy of the YOLOv4-tiny model is marginally lower than those of the region-based convolutional neural network (R-CNN) proposed by Girshick *et al.*[17] and its variants;[18–20] however, the YOLOv4-tiny model has a considerably higher execution speed. For systems that require real-time operation, a balance must be achieved between accuracy and execution speed.

2.3 UKF method

The Kalman filter is mainly used to estimate the state of an object. Because measurements are often inaccurate and noisy, the motion state of the target must be estimated and tracked on the basis of sensor measurements. However, the Kalman filter is unsuitable for nonlinear systems; moreover, such systems should generally be approximated using a first-order Taylor series (extended Kalman filter) to address the corresponding first-order biases in specific problems. Nevertheless, solving the Jacobi matrix is a time-consuming task. Hence, in the proposed system, the UKF method is adopted to overcome these drawbacks.

The UKF method applies a linearization method called unscented transformation. This method applies the linear regression of n sigma points obtained from the prior distribution to approximate a nonlinear function of random variables in systems with high nonlinearity. This linearization is more accurate and simpler than the Taylor series linearization approach.

2.4 Face recognition

The proposed system uses FaceNet to accomplish face recognition. FaceNet is a face recognition architecture that was developed by Google in 2015. It has been reported that among various face recognition models, FaceNet achieved the highest accuracy (99.63%) on data obtained from the LFW face database;[21] therefore, it has become one of the most commonly used face recognition models. Traditional face recognition methods usually apply the softmax function to output classification results; thus, when new data are added, the entire neural network must be retrained. FaceNet uses the feature vector of the output image rather than the classification result, and it compares the feature distance between the input image and the database image. The smaller this distance is, the higher is the probability of both images showing the same person. Therefore, when identifying new users, FaceNet requires only the photographs of these users and need not be retrained.

3. Methodology

The proposed system involves four processes: personnel detection using YOLOv4-tiny, personnel tracking using the UKF, visual tracking using FaceNet, and tracking using the UKF–FaceNet method. This system detects people and boundary frames from color images captured by the binocular camera, converts the image space into a 3D space to obtain the real space coordinates, and identifies or tracks specific users without the use of additional equipment or

specific markings (Fig. 1). This study is focused on fusion tracking methods; however, nonfusion methods can also be used in accordance with the relevant requirements.

3.1 Personnel detection using YOLOv4-tiny

Figure 2 illustrates the process of object detection by YOLOv4-tiny. After an image is input into the YOLOv4-tiny network, the original bounding box is predicted and then sorted through nonmaximum suppression (NMS) to obtain the final detection result.

YOLOv4-tiny has a simpler structure and simpler parameters than YOLOv4; thus, YOLOv4-tiny has a lower accuracy but a considerably higher execution speed than YOLOv4. A high execution speed is often more valuable than high accuracy in real-time detection tasks.

Figure 3 displays the architecture of YOLOv4-tiny, where 'Conv' represents convolution, 'BN' represents batch normalization, 'CBLR' represents convolution at 2, and 'Pooling' represents the pooling operation (i.e., max pooling in this study). YOLOv4-tiny outputs two feature sets for bounding box prediction (1/16 and 1/32 input image size features, respectively).

Because the original bounding boxes of the neural network output contain overlapping parts, the bounding boxes are sorted through NMS operations. After being sorted, each bounding box contains six predictions: the center coordinates of the bounding box, the width of the bounding box, the height of the bounding box, the confidence level of the bounding box, and the probability of each category.

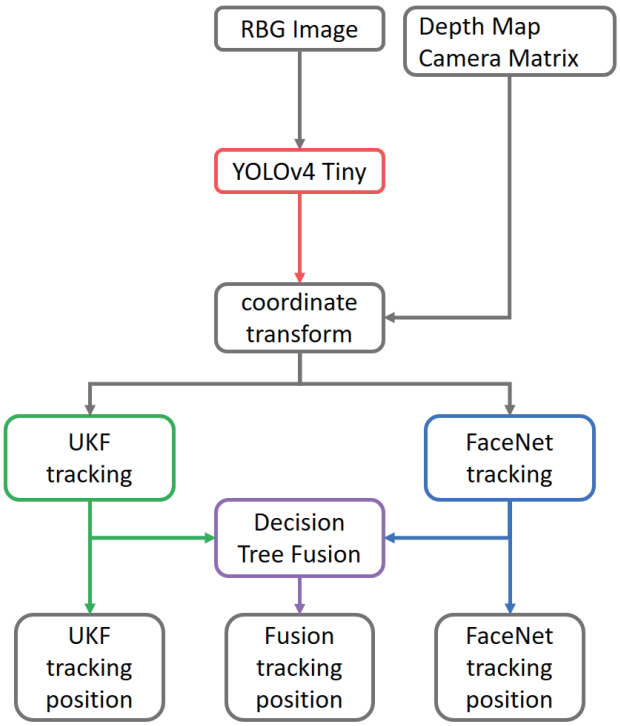


Fig. 1. (Color online) Architecture of proposed visual user tracking system.

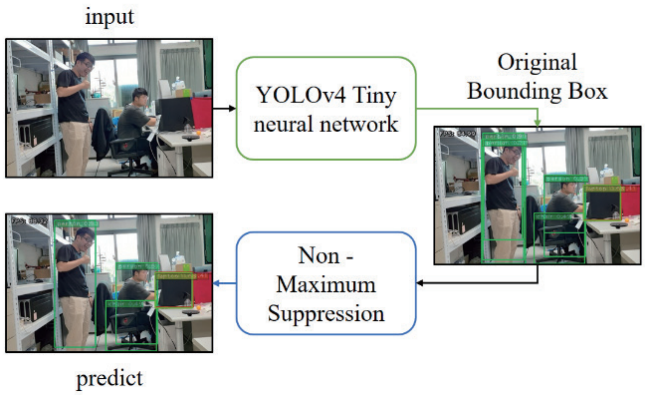


Fig. 2. (Color online) YOLOv4-tiny-based object detection.

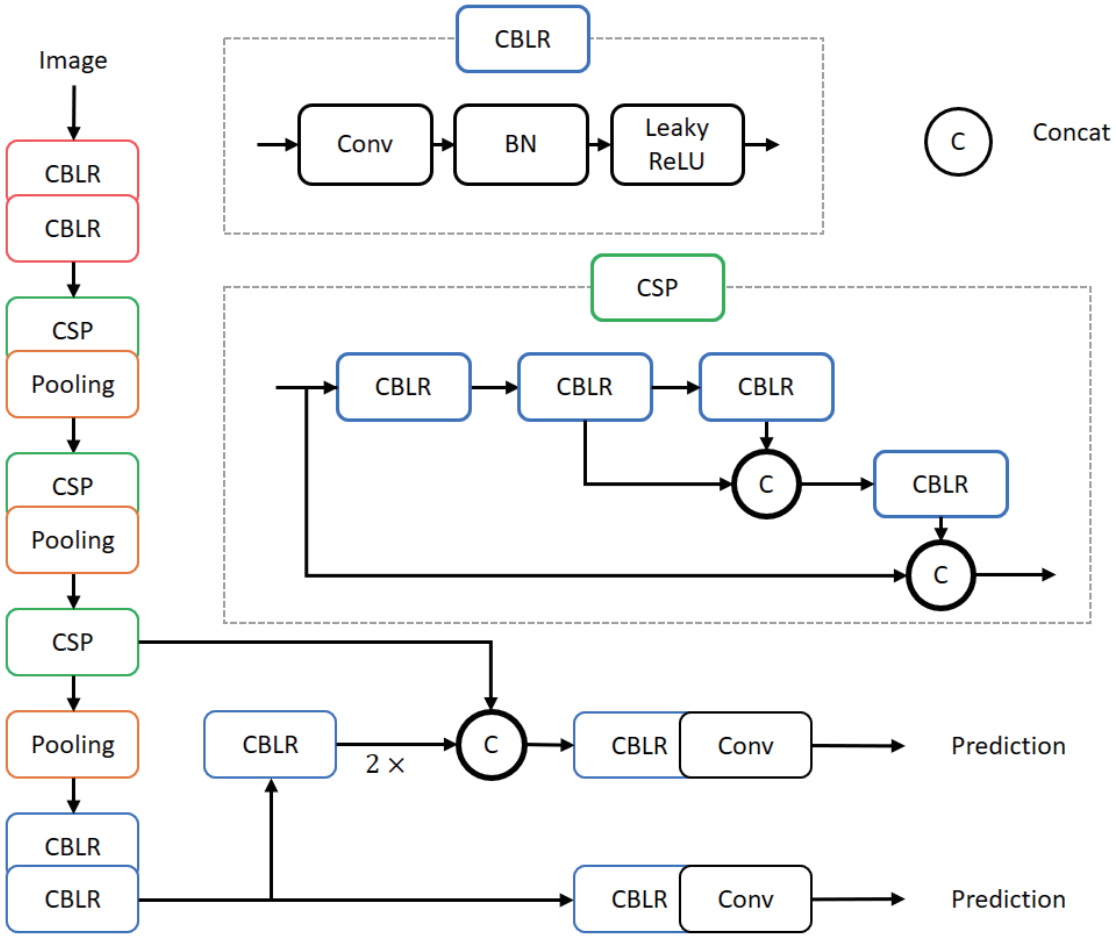


Fig. 3. (Color online) Architecture of the YOLOv4-tiny model.

A bounding box contains only 2D image coordinates (*uv* coordinates), and a depth map is used to convert the 2D image coordinates into 3D coordinates to obtain the temporal position of the human body in a 3D space. The *uv* coordinate system is derived by projecting the world coordinate system onto the camera coordinate system, as shown in Fig. 4.

In Fig. 4, F_c denotes the center of the camera, and X_c, Y_c, and Z_c denote the camera coordinates. The image coordinates (u, v) are transformed into world coordinates (X, Y, and Z) through a process called camera calibration, which is mathematically expressed as

$$m_{uv} = K[R|T]m_w, \tag{1}$$

where m_{uv} denotes the position on the projection plane, m_w denotes the corresponding position in a 3D space, K denotes the internal reference matrix for the camera, and $[R|T]$ denotes the matrix for the conversion of the world coordinates into camera coordinates. In the proposed system, the camera coordinates are used as a reference, and the pixel points on the projection plane are converted into a 3D space; therefore, the conversion equation can be modified as

$$m_c = K^{-1}m_{uv}. \tag{2}$$

The internal reference matrix of the ZED camera can be obtained from ZED SDK. After m_c is derived using the equation $m_c = [R|T]m_w$, the projection direction can be obtained by calculating the unit vector of m_c by using Eq. (3) and then multiplying this vector by the depth (m_{uv}), thereby completing the conversion from *uv* coordinates into 3D coordinates.

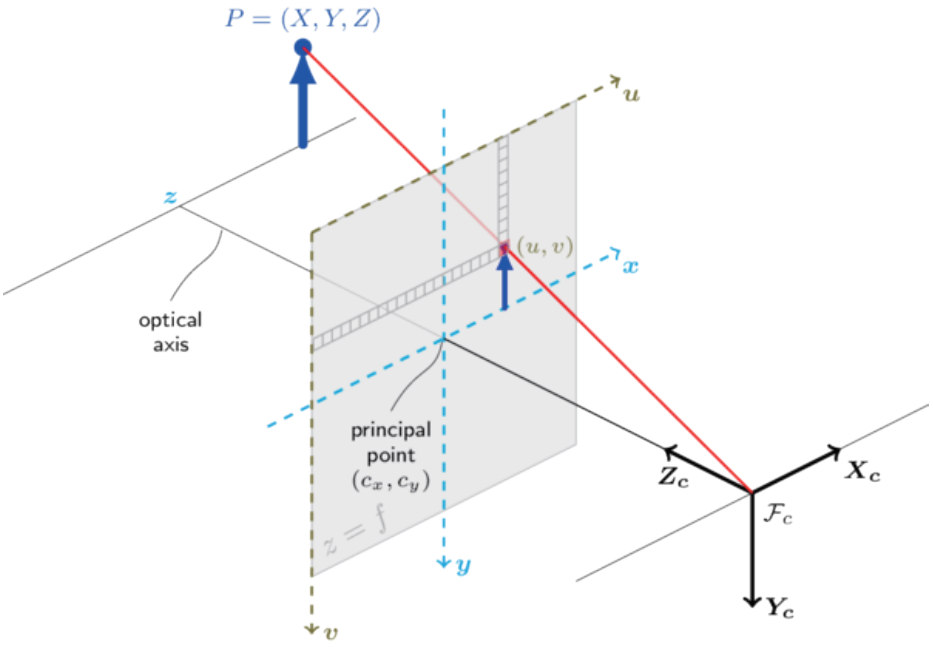


Fig. 4. (Color online) Conversion from the 2D image coordinate system to a 3D coordinate system.

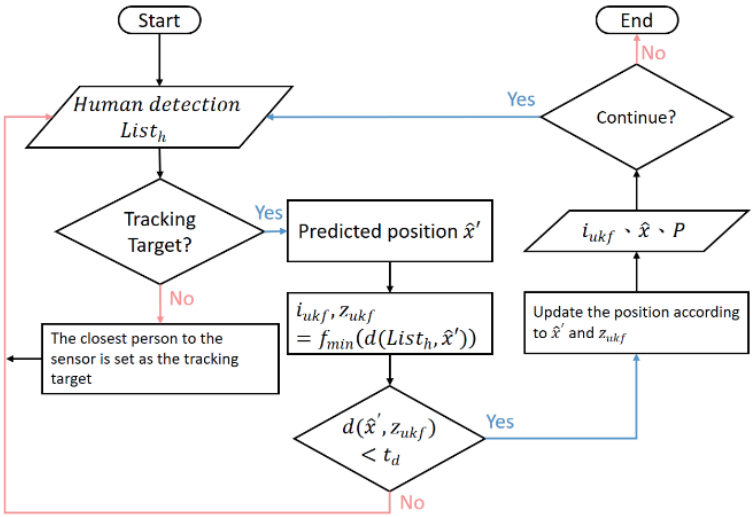
$$m_{\mathcal{F}_c} = \frac{m_c}{\|m_c\|_2} \times depth\left(m_{uv}\right) \tag{3}$$

Here, $m_{\mathcal{F}_c}$ is the vector position in a 3D space, $\|m_c\|_2$ is the Euclidean norm of m_c, and $depth(m_{uv})$ is the depth of m_{uv} obtained from the depth map. The proposed system extracts the bounding box and calculates the object position in the 3D space; however, for simplicity, only the median is considered to be representative of the 3D space. Thus, four items are derived for personnel detection: bounding box coordinates, bounding box confidence, category, and 3D space position.

3.2 Tracking using the UKF

The UKF-based tracking method mainly involves three steps: position prediction, measurement, and update (Fig. 5). After the tracking system is initiated, the nearest person is set as the target object, and the predicted value $\hat{x}'$ is obtained on the basis of the target's position and speed. The best measurement value z_{ukf} is then selected on the basis of the predicted value $\hat{x}'$, and the position update is completed.

The measured values are filtered on the basis of a threshold t_d. If $d\left(\hat{x}', z_{ukf}\right)$ is less than t_d, z_{ukf} is used as the measured value to reduce the possibility of tracking the wrong person because of



$List_h$	List of human bounding boxes detected by YOLOv4-tiny
$d()$	Euclidean distance
$f_{min}(d(List_h, \hat{x}'))$	Find the closest person to $\hat{x}'$ in the $List_h$
i_{ukf}, z_{ukf}	Human index, measurement position
t_d	Distance threshold
$\hat{x}, P$	Updated position, covariance

Fig. 5. (Color online) Flowchart of UKF-based personnel tracking method.

occlusion. Finally, the number (i_{ukf}) and position ($\hat{x}$) of the tracking target are sent after the data are updated.

The entire tracking process relies heavily on the predictive ability of the UKF. A real-world tracking scenario typically involves numerous nonlinear components. The predicted linear Gaussian distribution provided by a linear Kalman filter diverges to varying degrees under the existence of nonlinear measurement data, which results in unstable or failed tracking. The UKF method uses sigma points to solve this problem, and the sigma points are obtained using[10]

$$\begin{aligned}
&\mathcal{S}^{[0]} = \hat{x}_{k-1|k-1}, \\
&\mathcal{S}^{[i]} = \hat{x}_{k-1|k-1} + \left(\sqrt{(n+\lambda)P_{k-1|k-1}}\right)_i, \qquad \text{for } i = 1,\cdots,n \\
&\mathcal{S}^{[i]} = \hat{x}_{k-1|k-1} - \left(\sqrt{(n+\lambda)P_{k-1|k-1}}\right)_{i-n}, \quad \text{for } i = n+1,\cdots,2n
\end{aligned} \tag{4}$$

where $\mathcal{S}$ represents the sampled sigma points, n represents a parameter that is used to control the number of sigma points with the same task dimension, and $\lambda = \alpha^2(n+\kappa) - n$, where α and κ represent parameters that are used to adjust the distribution and weight of the sigma points, respectively. The higher the value of α is, the larger is the distribution of $\mathcal{S}$ values and the higher is the weight of the central sigma point. In this study, both α and κ were set to 1. The term i in $\left(\sqrt{P}\right)_i$ denotes the ith row of the root of the matrix. The weight $\mathcal{S}$ (w_m) and the weight of covariance w_c are calculated as[10]

$$\begin{aligned}
&w_m^{[0]} = \frac{\lambda}{n+\lambda}, \\
&w_c^{[0]} = \frac{\lambda}{n+\lambda} + \left(1-\alpha^2+\beta\right), \\
&w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)}, \qquad \text{for } i = 1,\cdots,2n.
\end{aligned} \tag{5}$$

The parameter β can be obtained on the basis of prior knowledge under the assumption that the distribution of $\mathcal{S}$ follows a Gaussian distribution. Subsequently, $\mathcal{S}$ can be substituted into the nonlinear state transfer function to obtain the nonlinear sigma point X, and the prediction step is completed using w_m. The mathematical expressions in the prediction stage are presented in Table 1.[22]

After $\hat{x}'$ is derived, the Euclidean distance is used to find person i_{ukf} and their closest position to $\hat{x}'$ (z_{ukf}). The visual system's field of view might be occluded, causing the system to track the wrong object; hence, z_{ukf} must be within a certain distance of $\hat{x}'$. Otherwise, the calculation is skipped. After z_{ukf} is obtained, the measured value is calculated (Table 2).[22] The $h()$ function shifts the sigma point X in the prediction stage to the target position z_{ukf}, and Z denotes the processed data. The weighting process in the measurement stage is approximately the same as that in the prediction stage.

Table 1
Mathematical expressions for the prediction step.

	Prediction
	$X = f(\mathcal{S})$
	$\hat{x}' = \sum_{i=0} w_m^{[i]} X^{[i]}$
	$P' = \sum_{i=0} w_c^{[i]} \left(X^{[i]} - \hat{x}'\right)\left(X^{[i]} - \hat{x}'\right)^T + Q$

$\hat{x}'$: Prediction position	$f()$: Transfer Function	Q: Noise
P': Prediction Covariance	$\mathcal{S}$: Sigma points	w_m, w_c : weight

Table 2
Mathematical expressions for the measurement step.

	Measurement
	$Z = h(X, z_{ukf})$
	$\hat{z}' = \sum_{i=0} w_m^{[i]} Z^{[i]}$
	$P_z' = \sum_{i=0} w_c^{[i]} \left(Z^{[i]} - \hat{z}\right)\left(Z^{[i]} - \hat{z}\right)^T + R$

$\hat{z}'$: Measurement position	$h()$: Sigma points Transfer Function	R: Noise
P_z': Measurement Covariance	w_m, w_c : weight	

After the predicted and measured data are processed, they can be corrected and updated as presented in Table 3,[22] where $\hat{x}$ represents the position coordinates of the current tracked object.

3.3 Visual tracking using FaceNet

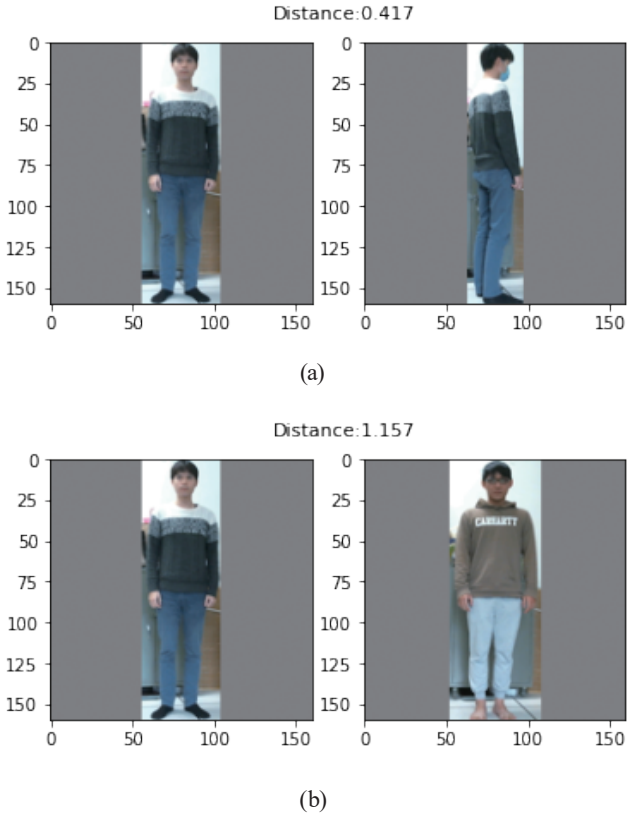
FaceNet is a face recognition model proposed by Google. This model calculates a set of eigenvalues for an input image. FaceNet can calculate the Euclidean distance between the features of two images to determine the feature similarity. As displayed in Fig. 6(a), a small feature distance can be obtained for the same person, even at different angles.

A threshold value can be set to determine whether the identities of the people in two bounding boxes are the same. The complete user identification and tracking process is illustrated in Fig. 7. When a user image is input into the proposed tracking system, the image is processed, its feature vector is calculated, and this vector is stored in $EList_t$. When a new list of people ($List_h$) is input into the tracking system, the system reads the bounding box information corresponding to each person and calculates the feature vector for each user image. Subsequently, it compares the Euclidean distances between the original input image and the new images, and the image pair with the smallest distance is selected to determine whether their Euclidean distance is smaller than the threshold t_E. If this distance is smaller than t_E, the coordinates of the user are output; otherwise, the next list ($List_h$) is input into the tracking system.

Table 3
Mathematical expressions for the update step.

	Update
	$K = \left[\sum_{i=0} w_c^{[i]} \left(X^{[i]} - \hat{x}'^{[i]}\right)\left(Z^{[i]} - \hat{z}'^{[i]}\right)^T\right] P_z'^{-1}$ $\hat{x} = \hat{x}' + K\left(z_{ukf} - \hat{z}'\right)$ $P = P' - KP_z'K^T$

K: Kalman Gain $\hat{x}$: Update position P: Update Covariance



(a)

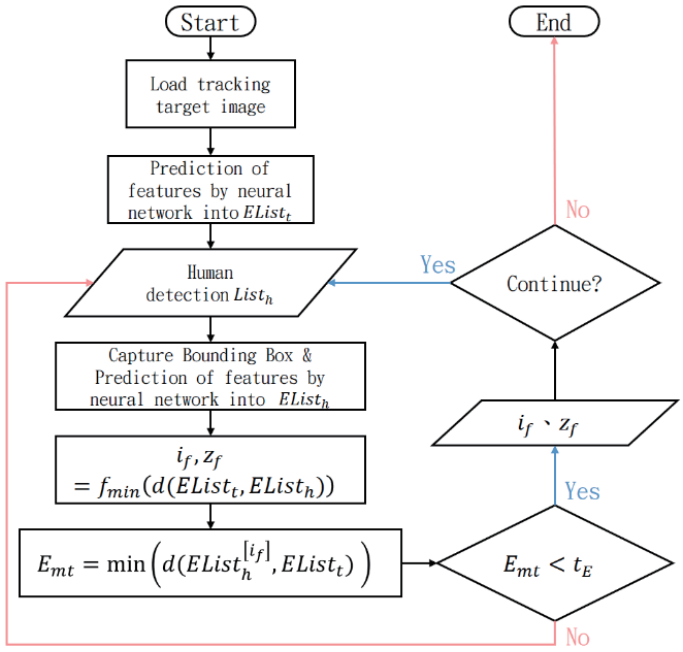
(b)

Fig. 6. (Color online) Schematic of the feature distances between two images of (a) the same person and (b) different people.

3.4 Tracking using the decision tree

The UKF can be used to track personnel by using limited resources and without classifying specific personnel. Using the UKF, the proposed system first tracks personnel closest to the camera; however, tracking failure, in which the system cannot identify personnel characteristics accurately, might occur in the following scenarios.

- The tracking target is blocked for a long time.
- The tracking target is outside the visual range.
- The tracking target is in close contact with another person.
- The tracking target and other people are moving in similar directions and meet each other.



$List_h$	List of human bounding boxes detected by YOLOv4-tiny
$d()$	Euclidean distance
$f_{min}(d(EList_h, EList_h))$	Detects the person with the minimum feature distance in both lists
i_f, z_f	Human index, measurement position
E_{mt}	Minimum feature distance between the person i_f and the tracking target feature
t_E	Feature distance threshold

Fig. 7. (Color online) Flowchart of FaceNet-based user identification and tracking.

Therefore, the proposed system is suitable for use only in low-interference environments or in environments where an additional transmitting device is attached to the tracking object.

FaceNet affords a simple means of detecting and tracking the location of a specific person. In FaceNet, the detection process is not continuous; thus, any user action is acceptable as long as the recognition capability of this network is sufficient. However, because FaceNet requires a relatively large amount of computational resources to achieve extensive user detection and tracking, inaccuracies in user detection and tracking might occur in the following scenarios.

- An object or human posture is not detected. The tracking target is outside the visual range.
- A pedestrian with the same body shape and wearing the same clothes (such as a uniform) as the tracking target appears.
- A user wears an object or accessory that obscures a feature.

Because the user detection process of FaceNet relies heavily on the bounding box, if the user's image cannot be extracted from the bounding box owing to obstruction or other conditions, the user's coordinates cannot be determined. When a pedestrian wearing similar clothes to the target user is encountered or a target user carrying a large backpack is moving, FaceNet might not be able to obtain the best feature vector. Consequently, the Euclidean distance

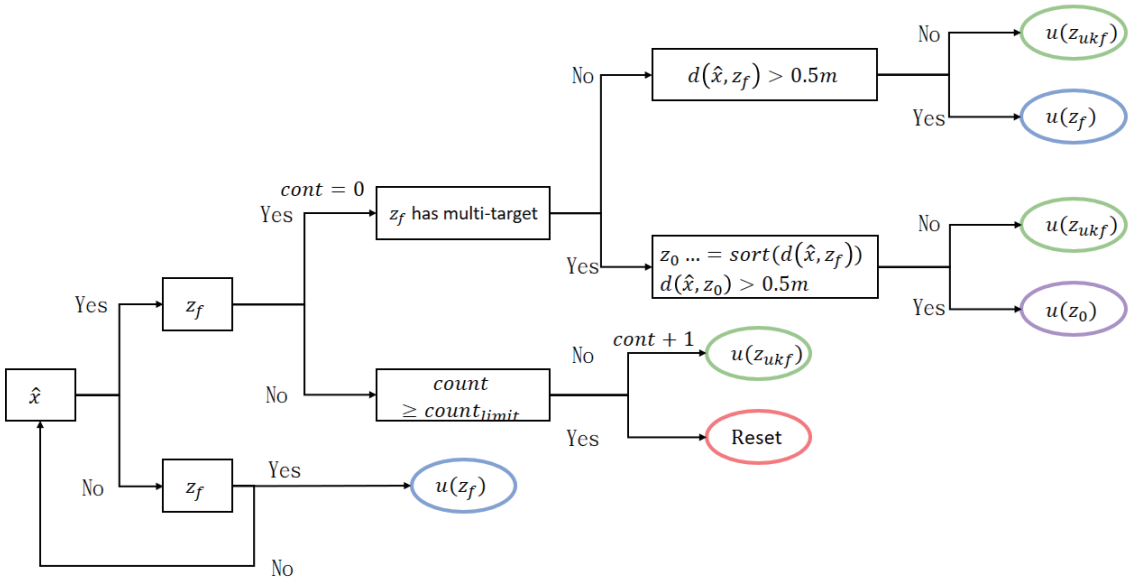
between the feature vectors of two people might be smaller than the threshold value, which would result in an incorrect detection.

A decision tree is a decision tool that uses tree-like graphs or decision models. In this study, FaceNet and the UKF (the backbone of the combined network) were integrated through decision tree fusion to leverage their advantages in tracking and address their drawbacks. As displayed in Fig. 8, when the personnel list $List_h$ that contains 3D location information is updated, the UKF–FaceNet tracking process is updated, as described below.

I. When the tracking system is started or reset, the UKF is not used to track the target ($\hat{x}$ has no data), FaceNet is used to visually confirm the user identity, and the user position (z_f) serves as the initial tracking target of the UKF.

II. If $List_h$ contains no user that matches the FaceNet tracking criteria (z_f) after several consecutive updates ($count_{limit}$), the user is highly likely to be outside the visual sensing range; thus, the tracking system is reset to reduce the possibility of the system tracking the wrong person through the UKF.

III. In nonspecial cases, when $List_h$ is updated, an analysis is conducted to determine whether the difference between the location of the target tracked using the UKF ($\hat{x}$) and the location of the user identified using FaceNet (z_f) is too large. Because the UKF-based tracking process



$\hat{x}$	UKF tracking output	$d()$	Euclidean Distance
z_{ukf}	UKF tracking measurement	$sort()$	Sort small to large
z_f	FaceNet tracking measurement	$u()$	Update UKF tracking
$count$	count variable	Reset	Reset UKF tracking
$count_{limit}$	Counting limit		

Fig. 8. (Color online) UKF–FaceNet tracking method.

cannot carry out self-correction, corrections cannot be performed if tracking fails or the wrong object is tracked. Therefore, if the aforementioned difference is too large, the FaceNet recognition result is used.

IV. In the FaceNet-based identification process, multiple eligible targets might exist when users have extremely similar physical appearances and clothing. However, images of the same user might be captured from different angles exhibiting different postures, accessories, or behaviours; thus, the smallest Euclidean feature distance might not correspond to the same user. Accordingly, in contrast to the FaceNet-based tracking process (Fig. 7), which considers only the user with the smallest feature distance to the target user, the integrated method records all eligible users; in the integrated method, the user with the closest distance to the UKF tracking position ($\hat{x}$) is identified, and this user's position (z_0) is retained. If the aforementioned distance is larger than the threshold, the user at z_0 is used as the tracking target for UKF correction. Hence, the integration of the UKF with FaceNet can enhance the accuracy and stability of user detection and tracking.

Table 4 presents the main program code run in accordance with the UKF–FaceNet method displayed in Fig. 8, and Tables 1–3 present the mathematical expressions for the *ukf* predictions, measurements, and updates. The code rospy.is_shutdown() is used to check whether or not the Robot Operating System (ROS) node is closed, and the result 'True' is returned if this node is closed.

$EList_m$represents the list of users whose feature distances are smaller than the threshold in the FaceNet model. Ideally, only one person should be on this list; however, when multiple possible targets exist, the person whose location is closest to $\hat{x}$ is selected rather than on the basis of the information of the UKF. The UKF–FaceNet method improves on the individual shortcomings of the UKF and FaceNet and thus enhances the accuracy and robustness of tracking in a multiple-persons space.

4. Experiments

4.1 Experimental steps and objectives

An edge computing platform (Jetson AGX Xavier, NVIDIA) and a binocular camera (ZED) constituted the main hardware (Fig. 9) used in this study.

The ROS system was used to implement relevant functions in this study, and ZED SDK and zed-ros-wrapper were installed on the edge platform to capture the sensor data of the ROS system (Fig. 10).

The sensor applied in the study experiments had a resolution of 720 pixels. Moreover, in the experiments, the *K*-matrix was used to transform the 2D image space coordinates into 3D space coordinates as

$$K = \begin{bmatrix} 521.3756 & 0.0 & 645.8579 \\ 0.0 & 521.3756 & 352.7648 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. \quad (6)$$

Table 4
Fusion code for the UKF–FaceNet tracking method.

INPUT: RGB imgahe i_{rgb}, human detection list $List_h$
OUTPUT: UKF tracking target position $\hat{x}$

$ukf \leftarrow$ Initialization UKF tracking filter
$model \leftarrow$ FaceNet TensorRT Inference model
$Elist_t \leftarrow$ Load the user's image and predict the feature list from the model
$count \leftarrow 0$, $count_{\text{limit}} \leftarrow$ Counting limit

while not rospy.is_shutdown( ) do
 foreach val in $List_h$ **do**
 The images are retrieved fromi_{rgb} on the basis of the bounding box coordinates of val. Use the $model$ to predict its feature and calculate the Euclidean distance with $Elist_t$.
 $d_{val} \leftarrow$ Keep the smallest distance data.
 if $d_{val} < t_E$ **then**
 $EList_m$ append ($[val, d_{val}]$)
 end if

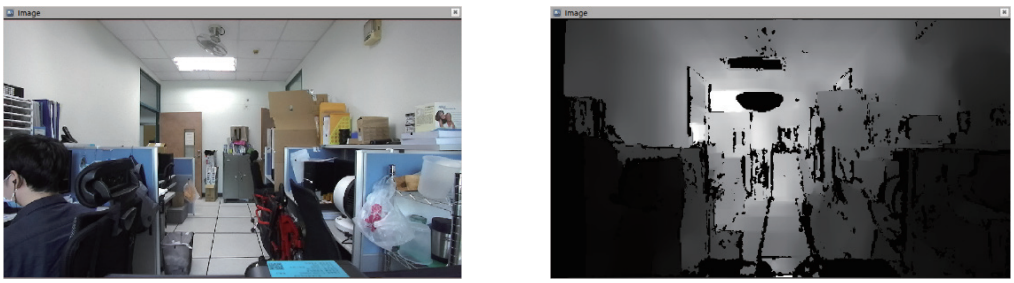
 if $ukf\ \hat{x}$ no date **then**
 if $len(EList_m) \geq 1$ **then**
 ukf predict
 $z_f \leftarrow$ Take the position of the smallest d_{val} in the $EList_m$ list in the three-dimensional position.
 $\hat{x} \leftarrow ukf$ update(ukf measurement(z_f))
 end if
 else
 if $EList_m$ has date **then**
 $count \leftarrow 0$
 if $EList_m$ has multitarget **then**
 $z_0 \dots z_{len(EList_m)} \leftarrow$ Sort $EList_m$ according to the Euclidean distance between z and $\hat{x}$, and save the three dimensions of EList_m from smallest to largest.
 if Euclidean distance $(z_0, \hat{x}) > 0.5m$ **then**
 ukf predict
 $\hat{x} \leftarrow ukf$ update (ukf measurement(z_0))
 else
 ukf predict
 $\hat{x} \leftarrow ukf$ update (ukf measurement(z_{ukf}))
 end if
 else
 $z_f \leftarrow$3D positions of target in $EList_m$
 if European distance $(z_f, \hat{x}) > 0.5m$ **then**
 ukf predict
 $\hat{x} \leftarrow ukf$ update (ukf measurement(z_f))

Table 4
(Continued) Fusion code for the UKF–FaceNet tracking method.

```
        else
         ukf predict
         x̂ ← ukf update (ukf measurement(z_ukf))
        end if
       end if
      else
       if count ≥ count_limit then
        ukf ← Initialize UKF tracking filter
       else
        count ← count + 1
        ukf predict
        x̂ ← ukf update (ukf measurement(z_ukf))
       end if
      end if
     end if
    end while
```

(a) (b)

Fig. 9. (Color online) Core hardware used in the study experiments. (a) Jetson AGX Xavier and (b) ZED binocular camera.



(a) (b)

Fig. 10. (Color online) Imagery generated by the ZED depth camera. (a) Color image and (b) depth image.

4.2 Personnel detection using YOLOv4-tiny

Figure 11 illustrates the visualization process executed in RViz, an ROS-based environment, for personnel detection. The resolution of the pretrained YOLOv4-tiny model used in this study was 288 × 288 × 3 pixels. This model was trained on the COCO data set, which contains 80 classes. To achieve the optimal runtime, the aforementioned model was converted into a half-precision TensorRT inference engine and designed as a resource package for ROS after confirming that it ran appropriately.

Figure 12 presents the visualization results obtained in the RViz environment. RViz was used to convert 2D information into 3D information on the basis of the depth map and the internal reference of the camera.

In Fig. 12, the top-left panel indicates the bounding box drawn by YOLOv4-tiny, the bottom-left panel indicates the result of human pose estimation, and the right panel indicates the visualization results in 3D space. In this figure, base_link represents the center coordinates of the sensor, and the red dot on the 3D human figure represents the median coordinate of the person obtained from the bounding box. This coordinate and the 3D space were used to conduct a tracking task in this study. The human skeleton is mainly used for illustrating visual effects suitably.

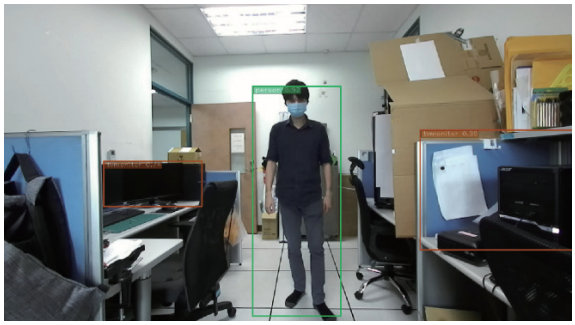


Fig. 11. (Color online) Visualization effect achieved with YOLOv4-tiny.

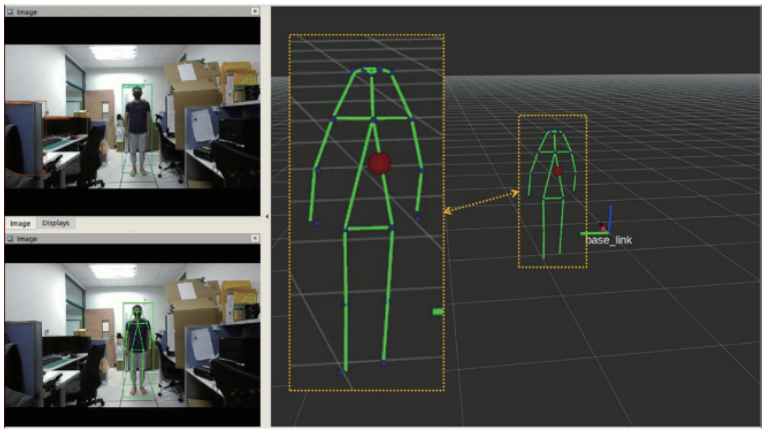
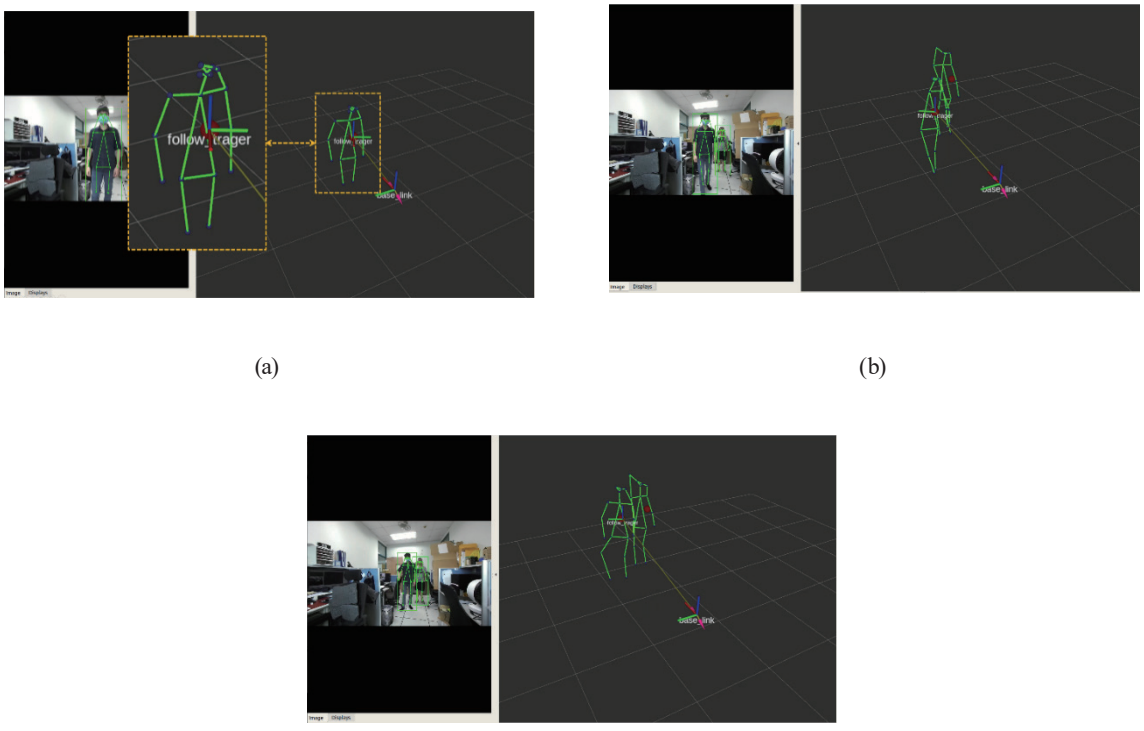


Fig. 12. (Color online) Visualization of 3D information.

4.3 UKF-based tracking

When the proposed tracking system is initiated, the person closest to the sensor is the initial tracking target, and the tracking process (Fig. 5) is executed. The steps in this tracking process are illustrated in Fig. 13, in which follow_target represents the visualization of the tracking target and its position, and the red axis represents the direction of movement of the target. As shown in Fig. 13(a), the system starts tracking the nearest person to it. Figure 13(b) depicts the tracking of individuals standing one behind the other, and Fig. 13(c) depicts the tracking of two individuals standing side by side.

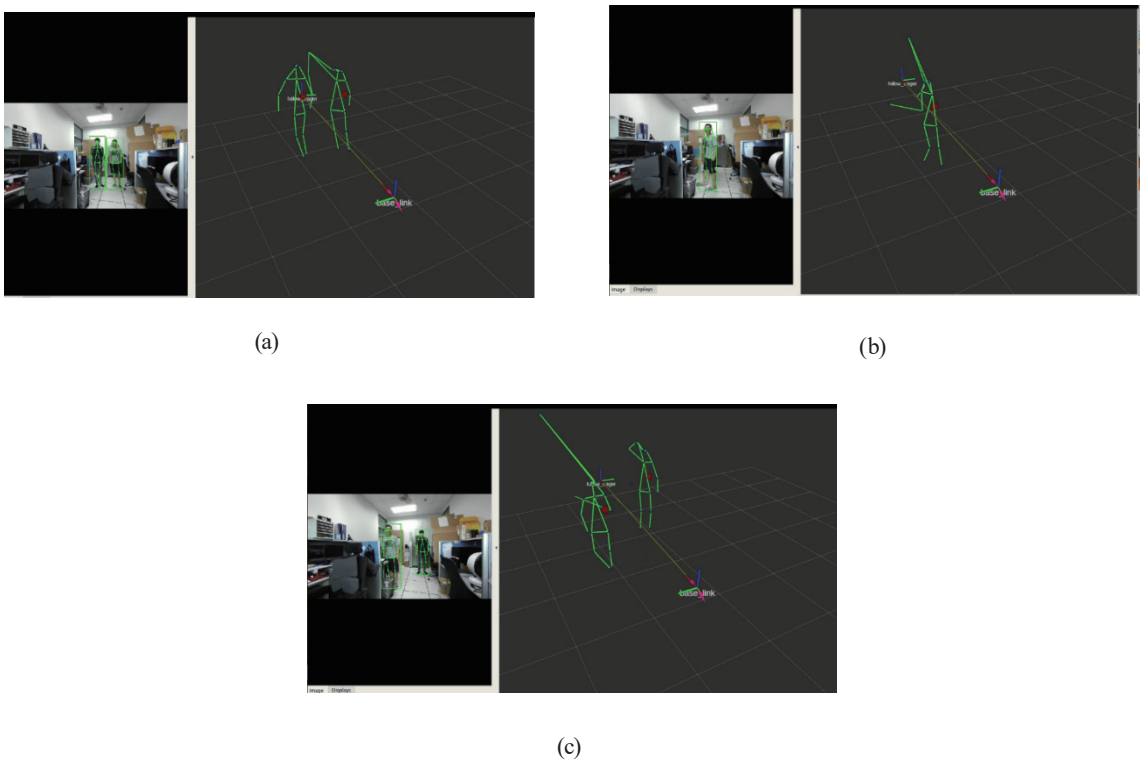
Because the proposed system currently does not allow for users to wear additional equipment, it might lose the target if its sensor is blocked briefly. Therefore, to reduce instances of failed or incorrect tracking, a threshold t_d is used to ignore observations with an excessively large position difference. If the target is blocked briefly during the tracking process, the target can be retraced; however, if the blocking time is long or the target position and direction change considerably, the tracking process is highly likely to fail or be erroneous (Fig. 14). If the tracking target cannot be reidentified in the UKF method, the UKF-based tracking process must be restarted.



(a) (b)

(c)

Fig. 13. (Color online) UKF-based tracking. (a) The nearest person is the initial tracking target, (b) tracking objects by moving backwards, and (c) tracking different targets simultaneously.



(a) (b) (c)

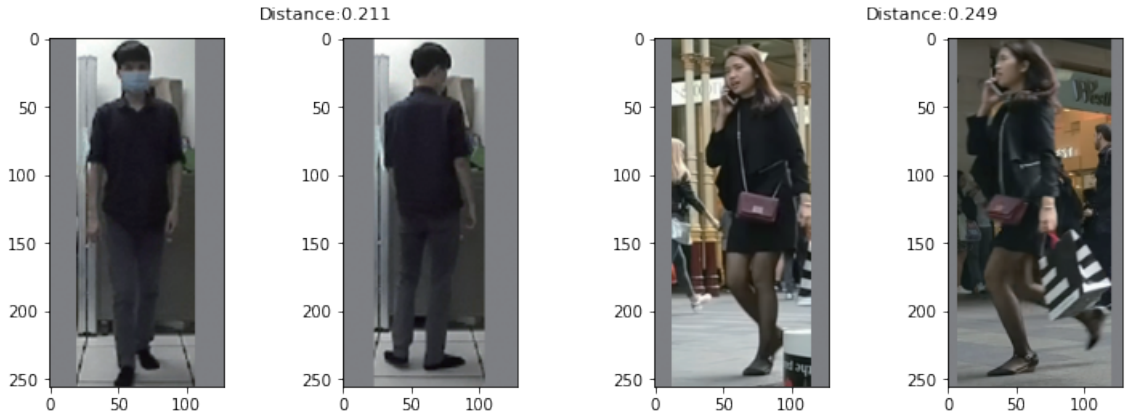
Fig. 14. (Color online) Failure of tracking after the tracking object was blocked. (a) Officers in pursuit, (b) blocking of the tracking object, and (c) tracking failure.

4.4 FaceNet-based visual tracking

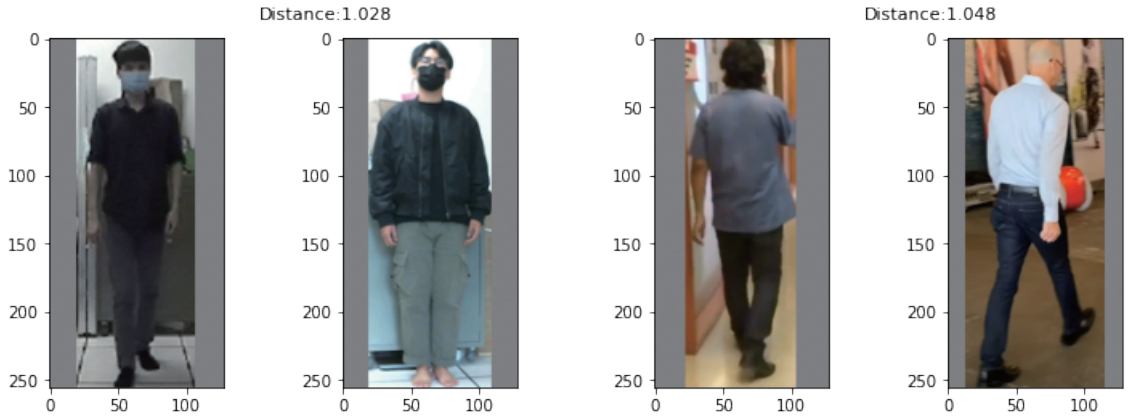
FaceNet contains ResNet-18[23] as its backbone. Its input size is 256 × 128 × 3 pixels, and its output feature size is 512 pixels. In this study, FaceNet was trained using a data set containing 658 full-body color images of 210 persons. The training results obtained for this network are displayed in Fig. 15, in which 'Distance' represents the Euclidean distance between two image features.

After the FaceNet model was trained, the user's image was stored in a folder and compared with the border object on the next screen. For images showing only the front of users, the highest detection accuracy was achieved when the feature distance threshold t_E was set to 0.75; however, under this setting, the system could easily misidentify the object. For images showing multiple angles of users (e.g., the front, side, and back), the highest recognition accuracy was achieved when t_E was set to 0.5; however, under this setting, the computation time was long. Figure 16 displays photographs of a single individual captured from different angles.

The FaceNet-based recognition and tracking processes are relatively straightforward; they involve calculating the distance between the image features of the tracking target and user separately for each frame and selecting the features with the smallest distance. Therefore, the detection and tracking processes encounter no problem if the user leaves the screen or is blocked for a long time. Figure 17 displays the elimination of the bounding box of the user after the user was occluded and the reappearance of the box after the user was no longer occluded.



(a)



(b)

Fig. 15. (Color online) Measured feature distances. (a) Euclidean feature distance between two images of the same person and (b) Euclidean feature distance between two images of different people.

(a) (b) (c)

Fig. 16. (Color online) Photographs of a single individual captured from different angles. (a) Front, (b) side, and (c) back.

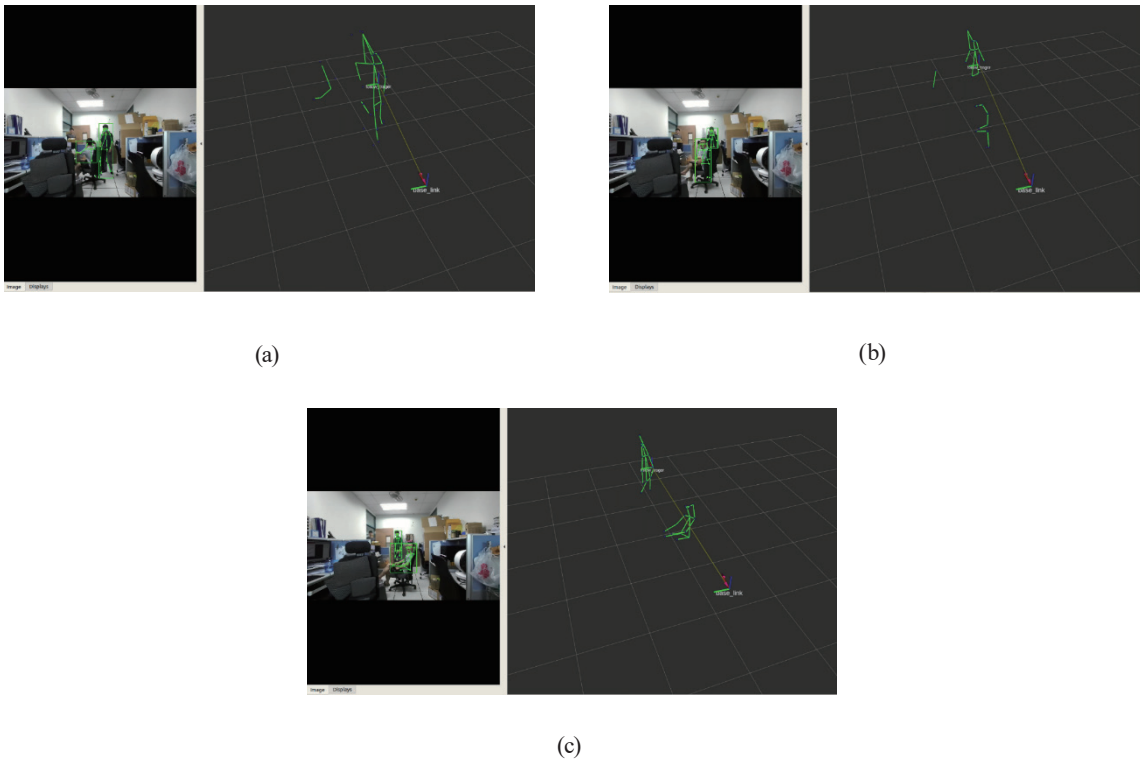


Fig. 17. (Color online) FaceNet-based tracking. (a) Tracking process, (b) occurrence of occlusion, and (c) identification of the tracking object.

Because of the intermittent occlusion of the tracking target, the output of the FaceNet-based tracking process was subject to abrupt changes, and the movement direction of the tracking target was not accurately detected. The detection and tracking processes become unstable when the system is fed images of multiple people who have similar body shapes and wear similar clothes. In such situations, a high image resolution is required to achieve accurate tracking and detection.

As mentioned, in this study, we integrated the UKF and FaceNet through decision tree fusion to leverage their advantages and overcome their drawbacks. The study results revealed that the UKF–FaceNet method achieved improved detection and tracking results in scenarios in which UKF-based tracking failed and in those in which the actual target was occluded for a prolonged period (Fig. 18).

4.5 Tracking by UKF–FaceNet method

As mentioned, we integrated the UKF and FaceNet through decision tree fusion to leverage their advantages and overcome their drawbacks. The study results revealed that the UKF–FaceNet method achieved improved detection and tracking results in scenarios in which UKF-based tracking failed and in those in which the actual target was occluded for a prolonged period (Fig. 18).

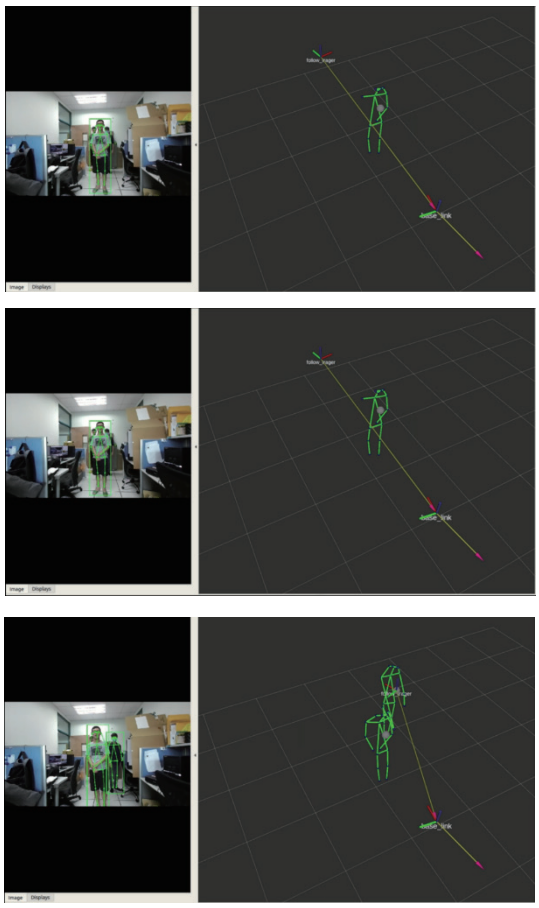


Fig. 18. (Color online) Retracing the tracking target after a prolonged occlusion period by using the UKF–FaceNet method.

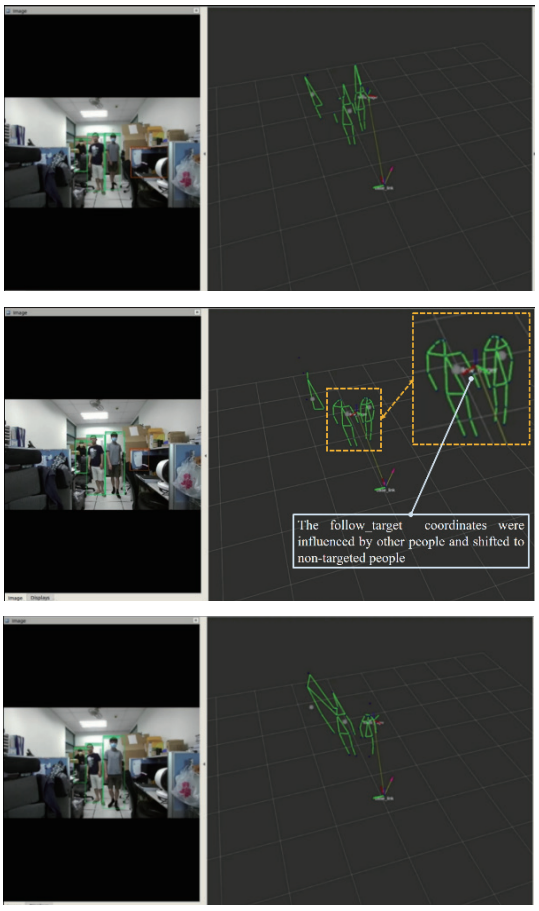


Fig. 19. (Color online) Results obtained with the UKF–FaceNet method when the tracking target crossed a narrow space.

In Fig. 18, the first panel illustrates the stoppage of the tracking process owing to the complete occlusion of the target, the second panel illustrates the reappearance of the target, and the third panel illustrates the regeneration of the bounding box and successful retracking. Figure 19 depicts the results obtained using the UKF–FaceNet method when the tracking target traversed a narrow space.

For the scenario illustrated in Fig. 19, the UKF method predicted the position and momentum of the tracking target and then detected the closest person to the predicted position. In this scenario, the tracking target and other people moved along the same path. Therefore, the UKF method tracked the wrong person (middle images). Because the UKF is suitable for nonlinear data, its updated value remained between those of the tracking target and other people. The updated value of the UKF was then successfully corrected using FaceNet (bottom images), which enabled the target to be correctly redetected and retracked. These results indicate that the UKF–FaceNet method can accurately detect and track targets in dynamic scenes with limited space.

5. Conclusion

In this study, we developed a camera vision system for accurately detecting and tracking a specific individual in a multiple-persons space without requiring the individual to wear external equipment or specific clothing. This system can obtain the location coordinates of individuals and send these coordinates to service vehicles or medical monitoring equipment so that suitable services can be provided to the individuals in a timely manner.

The proposed system uses the YOLOv4-tiny model to detect people and converts 2D image coordinates into 3D camera coordinates on the basis of a depth map and the internal reference of the adopted camera. The proposed system uses UKF, FaceNet, and UKF–FaceNet (integrated through decision tree fusion) methods to conduct user detection and tracking. Among these processes, the UKF–FaceNet method provided the best results. In contrast to traditional tracking methods, the proposed system requires only user photographs for user detection and tracking and does not require the use of additional sensing devices. Thus, the proposed system can provide accurate and stable user identification and tracking in a multiperson environment. This system can be used to conduct redetection and retracking when the target individual is occluded or tracking is interrupted.

Acknowledgments

This study was supported by the Ministry of Science and Technology of Taiwan under contract number MOST 110-2221-E-167-034 (duration: August 1, 2021–July 31, 2022).

References

1 X. Chen, Z. Qin, L. An, and B. Bhanu: IEEE Trans. Circuits Syst. Video Technol. **26** (2016) 2226. http://doi.org/10.1109/TCSVT.2015.2511480
2 I. Ahmed, A. Ahmad, F. Piccialli, A. K. Sangaiah, and G. Jeon: IEEE Internet Things J. **5** (2018) 1598.
3 H. Nodehi and A. Shahbahrami: IEEE Trans. Circuits Syst. Video Technol. **32** (2022) 147.
4 Panasonic: https://www.panasonic.com/jp/company/ppe/piimo.html (accessed June 2022).
5 Y. Sun, L. Sun, and J. Liu: 12th World Congress on Intelligent Control and Automation (WCICA) (2016) 1514.
6 J. Jommuangbut and K. Sritrakulchai: Int. Electrical Engineering Congress (iEECON) (2018) 1.
7 M. Sharikmaslat, R. Sidhaye, and A. Narkar: 3rd Int. Conf. Electronics, Communication and Aerospace Technology (ICECA) (2019) 702.
8 J. Chen and W. J. Kim: IEEE/ASME Trans. Mechatronics **24** (2019) 2377.
9 S. J. Julier and J. K. Uhlmann: Proc. IEEE **92** (2004) 401.
10 S. Julier, J. Uhlmann, and H. F. Durrant-Whyte: IEEE Trans. Automatic Control **45** (2000) 477.
11 F. Schroff, D. Kalenichenko, and J. Philbin: IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (2015) 815.
12 P. R. Gunjal, B. R. Gunjal, H. A. Shinde, S. M. Vanam, and S. S. Aher: Int. Conf. Advances in Communication and Computing Technology (ICACCT) (2018) 544.
13 M. Mirabi and S. Javadi: 3rd Int. Conf. Intelligent Systems Modelling and Simulation (2012) 303.
14 L. Li, X. Mu, S. Li, and H. Peng: IEEE Access **8** (2020) 139110.
15 Stereolabs - Capture the World in 3D: https://www.stereolabs.com/ (accessed June 2022).
16 A. Bochkovskiy, C. Y. Wang, and H. Y. Mark Lia: Vision and Pattern Recognition (cs.CV) (2020) 1.
17 R. Girshick, J. Donahue, T. Darrell, and J. Malik: IEEE Conf. Computer Vision and Pattern Recognition (2014) 580.
18 R. Girshick: IEEE Int. Conf. Computer Vision (ICCV) (2015) 1440.
19 S. Ren, K. He, R. Girshick, and J. Sun: IEEE Trans. Pattern Analysis and Machine Intelligence **39** (2017) 1137.

20 K. He, G. Gkioxari, P. Dollár, and R. Girshick: IEEE Int. Conf. Computer Vision (ICCV) (2017) 2980.
21 LFW Face Database: Main - Computer Vision Lab: http://vis-www.cs.umass.edu/lfw/ (accessed Jun. 2022).
22 R. labbe, Kalman-and-Bayesian-Filters-in-Python: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python (accessed Jun. 2022).
23 K. He, X. Zhang, S. Ren, and J. Sun: Computer Vision and Pattern Recognition (cs.CV) (2015) 1.