Hindawi
International Journal of Aerospace Engineering
Volume 2018, Article ID 6357185, 17 pages
https://doi.org/10.1155/2018/6357185

Research Article

Astronaut Visual Tracking of Flying Assistant Robot in Space Station Based on Deep Learning and Probabilistic Model

Rui Zhang[1], **Zhaokui Wang**[2] **and Yulin Zhang**[1,2]

[1]*College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China*
[2]*School of Aerospace Engineering, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Zhaokui Wang; wangzk@tsinghua.edu.cn

Received 3 January 2018; Revised 24 May 2018; Accepted 31 May 2018; Published 12 July 2018

Academic Editor: Franco Bernelli-Zazzera

Real-time astronaut visual tracking is the most important prerequisite for flying assistant robot to follow and assist the served astronaut in the space station. In this paper, an astronaut visual tracking algorithm which is based on deep learning and probabilistic model is proposed. Fine-tuned with feature extraction layers' parameters being initialized by ready-made model, an improved SSD (Single Shot Multibox Detector) network was proposed for robust astronaut detection in color image. Associating the detection results with synchronized depth image measured by RGB-D camera, a probabilistic model is presented to ensure accurate and consecutive tracking of the certain served astronaut. The algorithm runs 10 fps at Jetson TX2, and it was extensively validated by several datasets which contain most instances of astronaut activities. The experimental results indicate that our proposed algorithm achieves not only robust tracking of the specified person with diverse postures or dressings but also effective occlusion detection for avoiding mistaken tracking.

1. Introduction

The near-earth orbit space station serves as a microgravity research laboratory in which crew members conduct experiments in biology, physics, astronomy, and other fields. It is also suited for the testing of spacecraft systems and equipment required for missions to the Moon and Mars. Operating on orbit for more than ten years, the long-term care of the space station and the implementation of complex scientific studies will mainly depend on the astronauts. However, available astronaut time is usually limited. The improvement of astronauts' work efficiency becomes particularly important for space missions.

To help improving the astronauts' work efficiency, several in-cabin robots have been proposed or even flew on orbit in the International Space Station. PSA [1] was the first in-cabin assistant robot proposed by NASA's Ames Research Center. The program was aborted while its technologies inherited to the institute's related robot projects which called Astrobee [2]. Running in autonomous or tele-operated mode, Astrobee is designed to act the roles of mobile camera and mobile sensor in ISS. It is now under development and said to be sent to the ISS in the year 2018. Smart SPHERES [3] had been proposed in 2013 by upgrading the hardware and software of SPHERES [4] that was developed by MIT. It can be remotely operated by astronauts inside the spacecraft or by mission controllers on the ground, performing tasks such as environmental monitoring survey, inventory, and mobile camera work. Int-Ball is another in-cabin robot that has been sent to ISS in June 2017. It is designed by JAXA and acts as a mobile camera inside the ISS. Int-Ball can move on its own based on 3D target marker, or it can be controlled remotely by controllers and researchers on the ground. Apart from these robots, other astronaut assistant robots such as SHB [5] and AAR-2 [6, 7] were also proposed.

In this paper, an intelligent astronaut assistant robot called Intelligent Formation Personal Satellite (IFPS) is proposed and its prototype is shown in Figure 1. It weighs 2.6 kilogram and shapes in a sphere with a diameter of 20 centimeters. A RGB-D camera and IMU are chosen for the implementation of visual navigation. Six sonar sensors are laid out evenly on the sphere for obstacle avoidance.

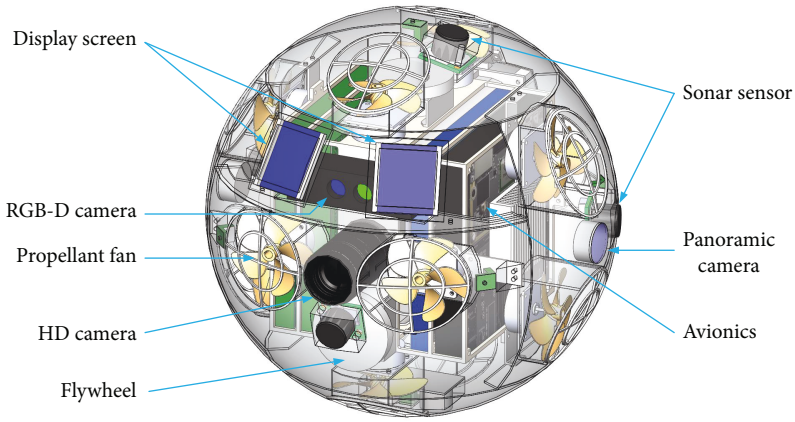


Figure 1: Prototype of the Intelligent Formation Personal Satellite.

Onboard information processing and computing hardware mainly consists of a Jetson TX2 and a FPGA. Six pairs of fans and three orthogonal flywheels are designed for position and attitude maneuvering. A docking station is designed to provide IFPS with support of docking and energy supplying. It will be mounted on the inner wall of the cabin. And it can also communicate with IFPS through high bandwidth wireless network, offering the robot extra off-board parallel computing resources with graphic processing units (GPU).

Based on the regularly updated map inside the cabin and the advanced 3D SLAM algorithm, the robot can fly autonomously everywhere in the space station without any cooperative identifiers. Formation hovering around the served astronaut and face-to-face interacting with him through gestures and voice, the robot offers assistance tasks such as data inquiry, panoramic video recording, and HD video recording in the space station. Compared with the robots mentioned above, our robot focuses more on autonomous flying and intelligent human-robot interaction. To enable the robot's capabilities of hovering around and interacting with the served astronaut, we should first address the problem of real-time astronaut visual tracking.

For astronaut tracking, the prior procedure of astronaut detection is needed. General people detection [8–10] methods often use features like Haar [11, 12], SIFT [13], HOG [14–16], and their combinations [17, 18] or variations [19, 20] to extract the most discriminative features, followed by a classifier such as SVM [14, 16], boosted classifiers [21], or random forests [22] to do classification. These methods are almost simply applied to detect pedestrians of standing or walking. They were designed to be robust to pedestrian detection of simplex postures while remaining sensitive to pedestrian detection of diverse postures. In recent years, deep learning models especially deep convolutional neural networks (DCNNs) have become state-of-the-art for many vision problems. In object detection research, deep convolutional neural networks such as Fast R-CNN [23], Faster R-CNN [24], and R-FCN [25] are proposed. While accurate, these networks are too computational. Improved through bounding box prediction and regression method, newer object detection models called YOLO [26] and SSD [27] achieve high accuracy while further increasing detection speed. Alone [28–31] or combined with the traditional methods [32], these novel deep learning methods are more and more widely used in people detection research. Based on the detection results, Kalman filter [33–35], particle filter [17, 36], and probabilistic model [32] are often used to accomplish accurate and consecutive people tracking task.

Under microgravity circumstance, astronaut's postures and gestures can be diverse during space activities in the space station. Postures like standing, headstand, climbing, crouch, or any others can be all possible. This differs quite a lot from the usual pedestrian detection problem. Here, our goal is to achieve accurate and consecutive tracking of the served astronaut in the space station, despite the diversity of astronaut's postures or dressings. And we focus on tracking of the served astronaut in RGB-D images which is captured by the RGB-D camera. Our proposed algorithm consists of a detection module and a tracking module. An improved SSD network is proposed as the detection module for detecting astronauts, followed by the tracking module which is based on a probabilistic model to do tracking. The algorithm runs 10 fps at Jetson TX2, providing robust tracking of the served astronaut as well as effective occlusion detection for avoiding mistaken tracking.

The paper is organized as follows. In Section 2, we will formulate the astronaut tracking problem and introduce the outline of our algorithm. In Section 3, we will introduce the astronaut detection module which is based on deep learning. Associated with the astronaut detection results, the astronaut tracking module will be presented in Section 4. In Section 5, we will evaluate our algorithm on real-world datasets which contain most instances of astronaut activities in our space station mockup.

2. Formulation of Astronaut Tracking

2.1. Problem Formulation and Analysis. Designed to be an in-cabin assistant robot in the space station, the Intelligent Formation Personal Satellite (IFPS) acts the role of astronaut's personal assistant, as shown in Figure 2. It flies alongside with the served astronaut and interacts with him

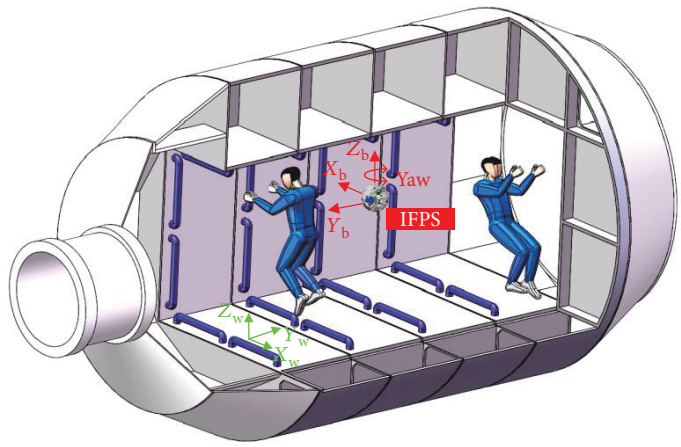


Figure 2: Intelligent Formation Personal Satellite in the space station.

through gestures or voice. To perform intelligent following and face-to-face interaction, we should first solve the problem of real-time astronaut tracking. Astronaut tracking, briefly speaking, is to recognize the served astronaut and locate him.

Such a special application scenario in the space station possesses special properties in the following aspects, which make astronaut tracking a distinctive and challenging task.

2.1.1. Astronaut. The postures and gestures of an astronaut can be arbitrary under microgravity condition. Postures like standing, headstand, climbing, crouch, or any others can be all possible. Besides, the wearing of astronaut can be different according to different space activities. And multiple astronauts may dress the same, for example, they may all dress blue in-cabin work clothes during working hours, as shown in Figure 2.

2.1.2. Robot. Flying inside the space station under microgravity condition, the robot has 6-DOF movement capability. However, we constrain the robot's movement freedom to 4-DOFs for ease of orienting and autonomous control while following flight. The 4-DOFs consist of three translational DOFs along its $X_bY_bZ_b$ body coordinates and the yaw freedom. The X_bY_b plane parallels to the X_wY_w plane of the World Coordinate System while the Z_b axis parallels to the Z_w axis, as shown in Figure 2. Besides, the robot keeps about one meter away from astronaut for better interacting with each other. Therefore, it is urged that any other astronauts do not break into the interspace between the robot and the served astronaut. And the robot may take the initiative to avoid obstacles to prevent the tracking target image from being blocked.

2.2. Algorithm Outline. For astronaut visual tracking problem, our proposed tracking algorithm is based on robust and accurate astronaut detection. Astronaut detection module and tracking module are the two main parts of our proposed astronaut tracking algorithm, as shown in Figure 3.

A deep learning-based detection module is first designed to detect all astronauts and locate them by drawing rectangle bounding box β_i^k. For the kth frame, the detection module outputs several boxes $\beta_1^k, \beta_2^k, \ldots, \beta_m^k$, which are the bounding boxes of the detected astronauts. Here, m denotes the number of astronauts detected. And the served astronaut p that we will track is confirmed from the early detections.

For these bounding boxes detected in the detection module, we take the most likely one as the served astronaut which we confirmed in advance. Then, our proposed tracking module formulates the tracking problem to be a *maximum a posteriori* probability problem as follows:

$$\arg\max P^k\left(p \mid \beta_i^k\right). \tag{1}$$

These two modules are described in the two following sections, respectively.

3. Astronaut Detection Module Based on Deep Learning

In this section, we introduce our designed astronaut detection module which is based on deep learning. The detection module returns the detection results of each frame and locates them by drawing rectangle bounding boxes $\beta_1^k, \beta_2^k, \ldots, \beta_m^k$.

3.1. DCNN for Astronaut Detection. Allowing for the real-time application background, we hope that our network runs as fast as possible while retaining a higher accuracy. In object detection research area, the Single Shot Multibox Detector (SSD) [27] is significantly more accurate and is currently the best detector with respect to the speed-versus-accuracy trade-off. Inspired by the SSD network, an end-to-end DCNN for astronaut detection is proposed. The architecture of the network is shown in Figure 4.

3.1.1. Data Layer. Data layer is responsible for color image reading and preprocessing. The color images that fed into the network are all resized to a size of 300×300.

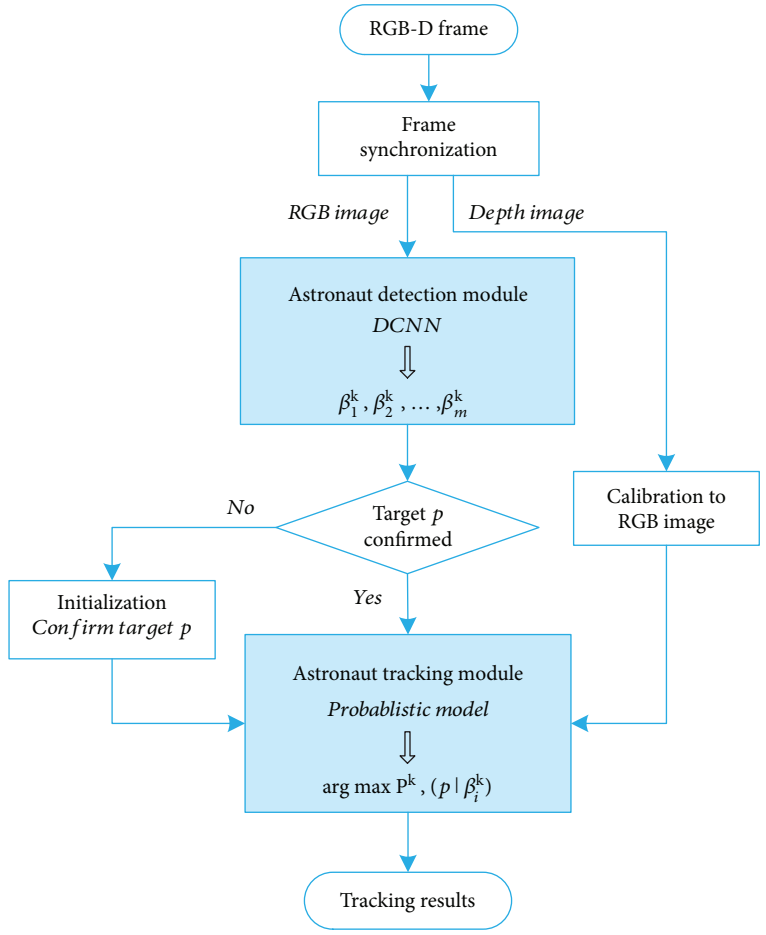


Figure 3: Outline of astronaut tracking algorithm.

3.1.2. Feature Extraction Layers. Robust and reliable feature extraction is critical for astronaut detection. Here, we used the fc-reduced VGG-16 network as feature extracting network. And transfer learning method was used when training the whole net as our training dataset is limited. We initialed our feature extraction layers' parameters with already-made model trained in object detection task, rather than in object classification task as the original SSD network did. The object detection network would converge well when it is fine-tuned from a pretrained object detection model which detecting more categories [37].

3.1.3. Multiscale Prediction Layers. Upon the feature extraction layers, multiscale layers are constructed to do multiscale prediction, as shown in Figure 4. Compared with the original SSD network, multiscale prediction layers are improved in three aspects allowing for the specificities of the astronaut detection task.

(1) Rationally designing the number of multiscale prediction layers and feature map size in each layer: from the original SSD network, we know that the six prediction layers predict objects in six different detection resolutions and scales. In astronaut detection task, astronauts present on the image within a certain scale as the cabin space is limited. Predictions on the 38×38 layer contribute little to the astronaut detection, however increased the computation burden. We predict our detections on 19×19, 10×10, 5×5, 3×3, and 1×1 layers. These five layers, namely, Conv6, Conv7, Conv8, Conv9, and Conv10, are first constructed through convolution upon the feature extraction layers, as shown in Figure 5.

(2) Sharing features that were extracted in lower layers to upper layer: allowing for the safe distance between robot and astronaut as well as the interaction distance demanded in ergonomics, astronaut image dimension varies within 10×10, 5×5, and 3×3 layers in most common situations, as shown in Figures 6(b)–6(d). Thus, we introduce features sharing mainly on these three layers. As shown in Figure 5, feature maps of 19×19 layer are first pooled to 10×10 and concatenate to the 10×10 Conv6 layer output (ReLU6) with a concatenation layer Concat7. It is worth noting that normalization is needed upon each layer before concatenation. Here, a batch norm layer followed with a scale layer is used for normalization. Same operations about sharing lower layers' features to the upper are

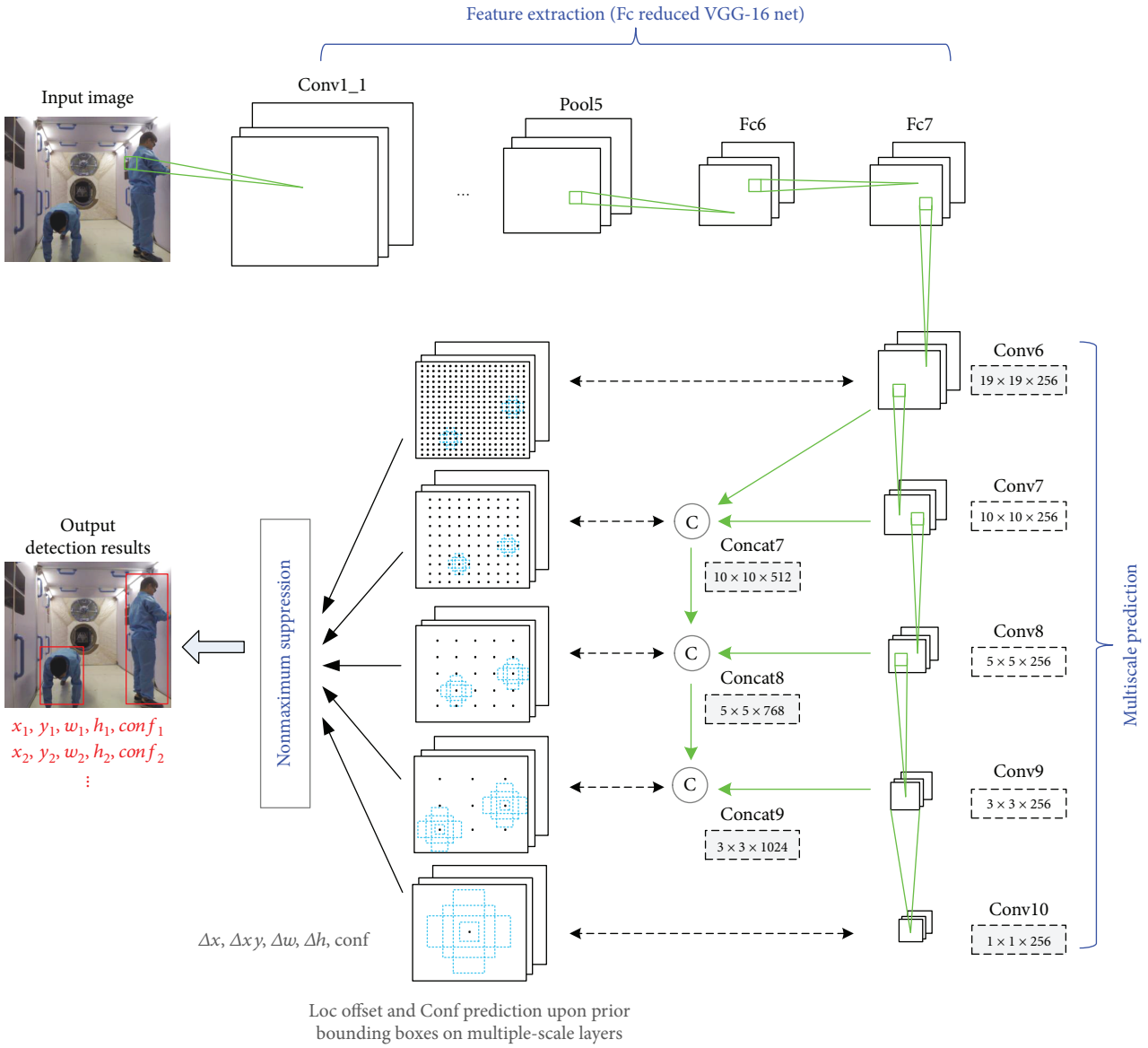


Figure 4: Deep convolutional neural network for astronaut detection.

taken upon 5×5 and 3×3 layers. Feature maps integrating lower layers' features are concatenated at layers Concat8 and Concat9, respectively. Finally, we predict multiscale astronaut detections upon outputs of layers Conv6, Concat7, Concat8, Concat9, Concat10, and Conv10 in five different scales, as shown in Figure 5.

(3) Parameter optimization of prior bounding boxes: multiscale layers Conv6, Concat7, Concat8, Concat9, and Conv10 are used for astronaut prediction. Upon these layers, a set of prior bounding boxes is assigned as proposal regions, as shown in Figure 6(a). Aspect ratios of prior bounding boxes on layers Concat7, Concat8, and Concat9 are all designed as $A_R \in \{1, 2, 3, 1/2, 1/3\}$, while $A_R \in \{1, 2, 1/2\}$ on layers Conv6 and Conv10. Width and height of prior bounding boxes in the kth prediction layer can be calculated by $w^k = L^k_{\min} \times \sqrt{A_R}$ and $h^k = L^k_{\min}/\sqrt{A_R}$ as [27] did. For the aspect ratio of 1, another prior bounding box is added and its scale is calculated as $w^k = h^k = \sqrt{L^k_{\min} \times L^k_{\max}}$ with $L^k_{\max} = L^{k+1}_{\min}$. Parameters of prior bounding boxes upon the five prediction layers are summarized in Table 1. Our network reduced the amount of proposals about 75% compared with the original SSD network. The final detection results are given by nonmaximum suppression from the total 2252 candidate proposals, outputting locations of detected astronauts as well as their confidences.

3.2. Training and Validation. In order to be consistent with the application in the space station as much as possible, our

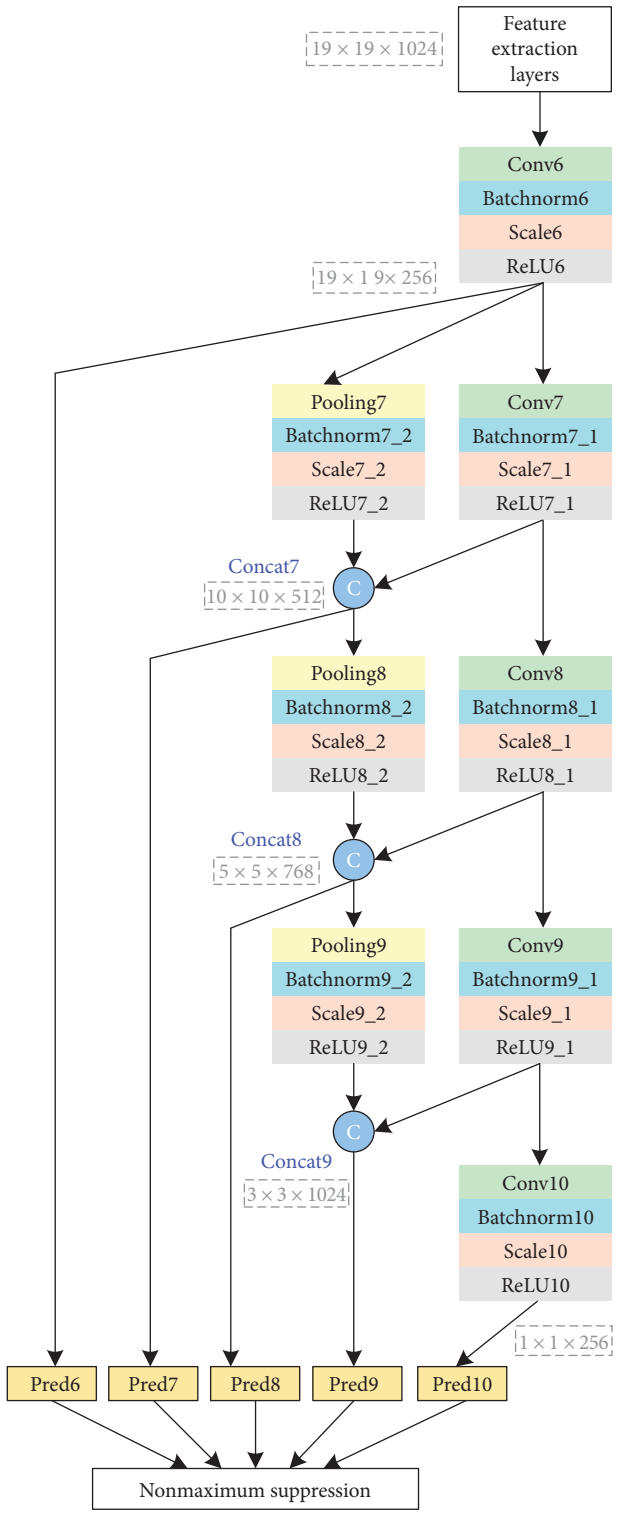


Figure 5: Architecture of multiscale prediction layers with features shared at main layers.

training dataset is obtained from videos recorded by mobile or security cameras in our ground space station mockup. Volunteers imitate the astronauts' operating and walking in the ground space station mockup arbitrarily, and their dressings can be arbitrary in three different types, as shown in Figure 7. We build our dataset by labeling the picked 4050 pictures from recorded videos, of which 3250 pictures are used as training dataset while the remaining 800 pictures are used as testing dataset. Then, we began training our astronaut detection network.

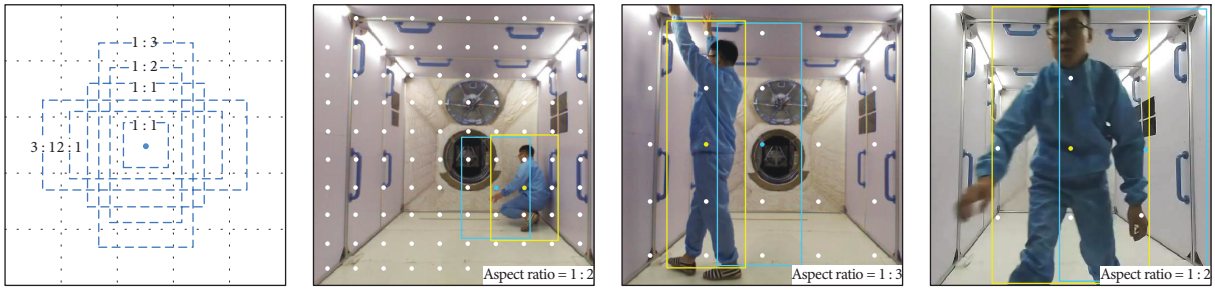
To train the astronaut detection network, we first match prior bounding boxes to any ground truth with jaccard overlap higher than 0.5. Figure 8(b) shows the matching of six prior bounding boxes at a cell in layer Concat8 to the ground truth. The training objective loss for an image is defined as a weighted sum of the localization loss and the confidence loss.

$$L(c,l) = \frac{1}{N}\sum_{i=1}^{2252}\sum_{j=1}^{G_N}\left(L_{\text{conf}}\left(c_i,\hat{c}_{ij}\right) + \lambda\hat{c}_{ij}L_{\text{loc}}\left(l_i,\hat{l}_{ij}\right)\right), \quad (2)$$

where N is the total number of the matched prior bounding boxes, G_N is the total number of the ground truth bounding boxes in this image, c_i is the confidence score of being an astronaut in the ith prior bounding box, $\hat{c}_{ij}$ is the ground truth matched label, $\hat{c}_{ij}$ is 1 when the overlap between the ith prior bounding box and the jth ground truth box is above 0.5, otherwise it is 0, l_i is the offset parameter for the ith predicted bounding box, $\hat{l}_{ij}$ is offset parameter between the ith prior bounding box and the jth ground truth box, and λ is a weighted parameter and it is set to 1 here. For the localization loss $L_{\text{loc}}(l_i,\hat{l}_{ij})$, smooth L1 loss function as described in [23] is adopted. For the confidence loss $L_{\text{conf}}(c_i,\hat{c}_{ij})$, the Softmax loss is used to identify whether an astronaut is detected in the prior bounding box or not.

Pretraining on a large-scale dataset and transferring the learned features to another task with smaller dataset became an effective strategy in deep learning-based applications. Fine-tuning from a pretrained model can combat overfitting effectively [38], leading to superior performance over training from scratch. Considering the limited size of our training dataset, we use transfer learning method to train the astronaut detection network. Before training, we initialize the feature extraction layers' parameters with a ready-made object detection model which is pretrained on the VOC07, VOC12, and COCO dataset. The pretrained model can be obtained from https://drive.google.com/file/d/0BzKzrI_SkD1_TkFPTEQ1Z091SUE/view. The other layers are randomly initialized with a zero-mean Gaussian distribution. The stochastic gradient descent (SGD) is used for training with a starting learning rate of 0.001, momentum of 0.9, and weight decay of 0.0005. The learning rate successively drops by 10 at 10,000, 20,000, 30,000, and 40,000 iterations. The training procedure consists of 50,000 iterations, and it is implemented with Caffe on NVIDIA TITAN X.

As a comparison, we modified the original SSD network to be a binary classification network for astronaut detection and fine-tuned it with the same dataset and solver parameters. Training loss and testing accuracy of the two networks during training are shown in Figures 9 and 10, respectively. Compared with the original SSD network, our proposed astronaut detection network not only converges faster and



(a) Prior bounding boxes at each cell

(b) Proposals on 10×10 layer Concat7

(c) Proposals on 5×5 layer Concat8

(d) Proposals on 3×3 layer Concat9

Figure 6: Prior bounding boxes and proposals on the main prediction layers.

Table 1: Parameters of prior bounding boxes in each prediction layer.

Layer	Scale	Prior bounding box parameters			
		L_{min}	L_{max}	Step	Aspect ratios
Conv6	19×19	60	115	16	1, 1/2, 2
Concat7	10×10	115	170	30	1, 1/2, 2, 1/3, 3
Concat8	5×5	170	225	60	1, 1/2, 2, 1/3, 3
Concat9	3×3	225	280	100	1, 1/2, 2, 1/3, 3
Conv10	1×1	280	320	300	1, 1/2, 2

Figure 7: Volunteers in three different kinds of dressings.

more stable but also results a better testing accuracy on the same testing dataset.

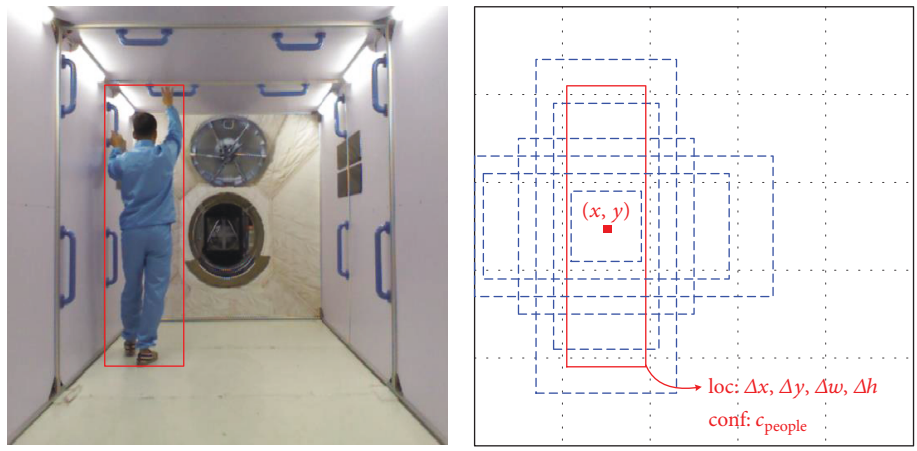
When the training procedure is finished, we validate our trained network's generalization ability by detecting people with different dressings and diverse postures in real application. Three different colors of clothes are used to represent different kinds of spacesuits, as shown in Figure 7. Several postures which may commonly occur under microgravity condition are chosen for testing. The volunteer turns around with these postures successively so as to exhaustively test the performance of detecting human body with different postures. Besides, people of diverse postures are detected from two different directions of view. The detection result is located on the image by a red rectangle. Figure 11 shows the result of detecting people in blue dressing with our proposed astronaut detection network. Comparison of the detailed detection results is illustrated in Table 2.

We treat the detection output whose confidence is above 0.5 as an effective output. And clear metrics are proposed to quantitatively evaluate the performance of our detection module. The metrics consist of five factors: true positives (TP), false positives (FP), false negatives (FN), maximum number of consecutive false negatives (CFN), and total detection accuracy (TDA). TP represents the number of successful and right detections. FP represents the number of mistaken detections. Astronaut detection is actually a binary classification problem as we just distinguish astronauts from the background. Thus, FN represents the number of missed detections. And CFN represents the maximum number of consecutive missed detections. TDA is determined by the ratio of TP to the total number of the experimental video frames. The experimental results show that our proposed astronaut detection network performs stronger generalization ability in real detection task, resulting in better detection accuracy than the original SSD network. Our proposed astronaut detection network is a highly effective and robust classifier for detecting people with diverse dressings and postures. Although the headstand posture was not tested here due to conditional restrictions, it is certain that our proposed DCNN is capable of such case if trained.

Running speed was also compared between our proposed detection network with the SSD network, as shown in Table 3. Our detection network runs at 66 fps speed on NVIDIA TITAN X, 6 fps faster than the SSD network. Considering the application in assistant robot IFPS, we transplant the algorithms to the embedded AI supercomputer Jetson TX2 designed as the robot's processor. Our detection network runs at 11~12 fps on TX2, 1 fps faster than the binary classification SSD network. It takes less than 91 milliseconds for our network to detect a picture on Jetson TX2.

4. Astronaut Tracking Module Based on Probabilistic Model

In this section, we introduce the astronaut tracking module. As the proposed astronaut detection module can produce relatively clean detection results, astronaut tracking becomes more manageable.



(a) Ground truth bounding box

(b) Match prior bounding boxes at a cell on layer Concat8 to the ground truth

Figure 8: Match prior bounding boxes to the ground truth.

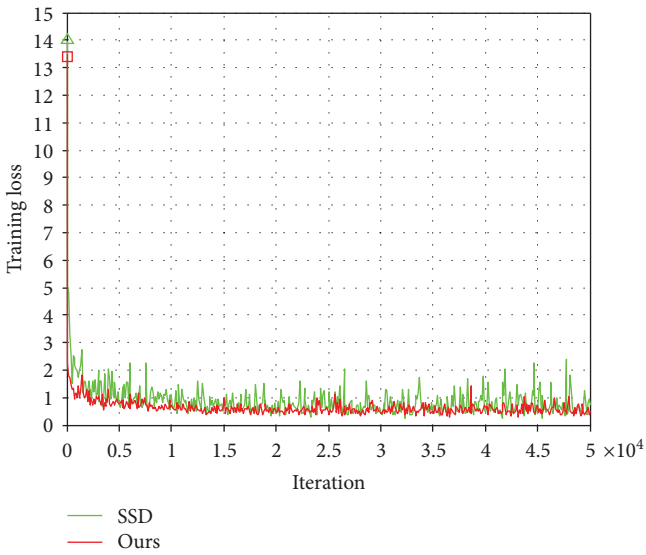


Figure 9: Comparison of training loss during training procedure.

4.1. Overview of Tracking Module. Figure 12 shows the flowchart of the tracking module. From Section 2.2, we know that our tracking problem is actually a *maximum a posteriori* problem where we want to maximize the probability $P^k(p \mid \beta_i^k)$. The probability can be estimated by many clues. Here, we aim to track astronaut in RGB-D data, so we have color image and depth information. Thus, the posteriori probability can also be formulated as

$$P^k\left(p \mid \beta_i^k\right) = P^k(p \mid L_i)P^k(p \mid I_i), \tag{3}$$

where L_i is the spatial location of the detected astronaut in bounding box β_i^k and I_i is the color image in bounding box β_i^k. $P^k(p \mid L_i)$ and $P^k(p \mid I_i)$ can be obtained by a matching with predicted spatial position and a geometric similarity matching of bounding box in color image, respectively.

4.1.1. Matching with Predicted Spatial Position. A modified Kalman filter is adopted to predict the spatial position of the served astronaut. The Kalman filter is initialized by five consecutive frames during the target confirm procedure mentioned in Section 2.2. The probability $P^k(p \mid L_i)$ is derived from the relation between the predicted position and the current measured position.

4.1.2. Geometric Similarity Matching of Bounding Box. Allowing for the probable same dressing and similar postures of astronauts, color information is not discriminative enough to do identification. Thus, we use a simpler geometric similarity method to describe the probability $P^k(p \mid I_i)$. It is derived from the geometric position and shape relations between the current detection and the last tracked result in color image.

Then, the astronaut tracking problem can be described as

$$\arg\max P^k\left(p \mid \beta_i^k\right) = \arg\max P^k(p \mid L_i)P^k(p \mid I_i). \tag{4}$$

Tracking procedure will stop and quit when qualified detection results or matching responses are not provided for a predefined period of time. Then, the robot will alarm and request for another opportunity to reconfirm the tracking target.

4.2. Matching with Predicted Spatial Position. To predict motion of the served astronaut, we should first know the astronaut position relative to the assistant robot. To begin with, we define the center point (u_{center}, v_{center}) of the bounding box in detection result to be astronaut's 2D location in color image, as shown in Figure 13(a). After calibrating the perspective difference between the color and depth camera, we acquire fused color and depth image. Thus, for each pixel in color image, the corresponding depth value can be

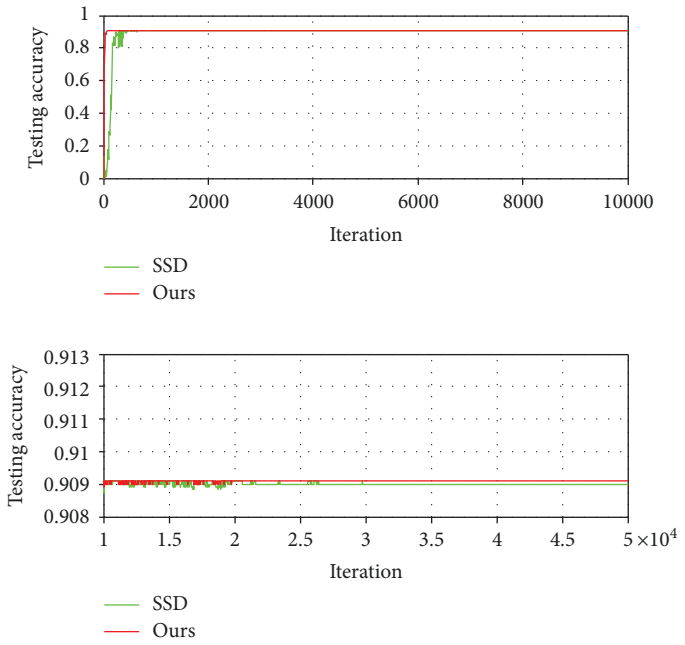
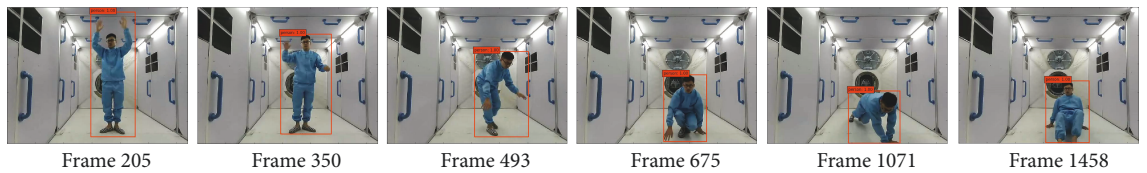


Figure 10: Comparison of testing accuracy during training procedure.



(a)

(b)

Figure 11: Detecting people of diverse postures from different views.

Table 2: Result comparison of detecting people with diverse postures and dressings.

Dataset settings			Detection results									
			SSD					Ours				
Clothes color	View type	Total frames	TP	FP	FN	CFN	TDA	TP	FP	FN	CFN	TDA
Green	1	1850	1844	0	6	4	99.7%	1850	0	0	0	100%
	2	1853	1779	0	74	23	96.0%	1850	0	3	3	99.8%
Blue	1	2131	2117	0	14	5	99.3%	2122	0	9	5	99.6%
	2	2050	2049	0	1	1	99.9%	2041	0	9	9	99.6%
Indigo	1	1675	1675	0	0	0	100%	1675	0	0	0	100%
	2	1893	1893	0	0	0	100%	1893	0	0	0	100%

obtained. Astronaut's depth value relative to color camera is the average value of the smallest six sparse sampling depth values, as shown in Figure 13(b). Based on the color camera model, 2D location in color image $(u_{\text{center}}, v_{\text{center}})$, and the depth value, astronaut spatial position $(X_{\text{RGB}}, Y_{\text{RGB}}, Z_{\text{RGB}})$ relative to color camera can be obtained. Furthermore, we

Table 3: Comparison of running speed.

Network	Speed	
	Titan X	Jetson TX2 (Max-N@2035 MHz)
SSD	60 fps	10~11
Ours	66 fps	11~12

can easily obtain astronaut spatial position (X_b, Y_b, Z_b) in robot body coordinate system from a simple coordinate transformation, as shown in Figure 13(c). Then, a modified Kalman filter is proposed to predict astronaut's spatial position and velocity in the robot's body coordinate system.

As astronaut motion model, we choose a constant velocity model as the common solution. The motion velocity of an astronaut is assumed to be constant between frames, so the state equation is simplified and does not include the acceleration term. The behavior of moving astronaut can be characterized by the following models of motion and measurement, respectively.

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_k, \tag{5}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k. \tag{6}$$

In the above equations, $\mathbf{A}$ represents the state transition matrix, which determines the relationship between the current state vector $\mathbf{x}_k$ and the previous $\mathbf{x}_{k-1}$. $\mathbf{H}$ represents the measurement matrix, which determines the relationship between the measurement vector $\mathbf{z}_k$ and the state vector $\mathbf{x}_k$. The vector $\mathbf{w}_k$ is process noise while $\mathbf{v}_k$ is measurement noise. These two noise terms are assumed to be White Gaussian Noise with zero mean and covariance matrices $\mathbf{Q} = E[\mathbf{w}_k\mathbf{w}_k^T]$ and $\mathbf{R} = E[\mathbf{v}_k\mathbf{v}_k^T]$, respectively.

The state vector in the presented motion model is described as

$$\mathbf{x}_k = [\mathbf{S}_k \ \mathbf{V}_k]^T, \tag{7}$$

where the vector $\mathbf{S}_k = [X_k \ Y_k \ Z_k]$ and $\mathbf{V}_k = [V_{Xk} \ V_{Yk} \ V_{Zk}]$ represent the spatial position and velocity of astronaut in the robot's body coordinate system, respectively. By considering the consecutive frames as motion of constant velocity, the state transition matrix $\mathbf{A}$ can be achieved from the kinematic equations as follows:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{S}_{k-1} + \mathbf{V}_k \times \Delta t, \\ \mathbf{V}_k &= \mathbf{V}_{k\text{-}1}, \end{aligned} \tag{8}$$

where Δt represents the sampling interval. From Section 3.2, we know that it takes less than 91 ms for our astronaut detection network to detect an image on TX2. It is enough for us to predict and update the state vector every 100 ms, namely, $\Delta t = 100$ ms. Our astronaut visual tracking algorithm runs at 10 fps on Jetson TX2.

As introduced above, the position of an astronaut is obtained from the detection results. The position coordinate X_k and Y_k are calculated from the bounding box of astronaut detection, while the coordinate Z_k is smoothed value of the corresponding depth sampling points. To avoid large discrepancies that could arise between frames due to the vision-based astronaut detection, we replace the measurement vector $\mathbf{z}_k$ in (6) with smoothed measurement vector $\mathbf{z}_k^s$. The smoothed measurement vector $\mathbf{z}_k^s$ takes the previous measurement into account. It is calculated by the following equation:

$$\mathbf{z}_k^s = \left(\mathbf{S}_k + \left(\mathbf{z}_{k\text{-}1}^s + \frac{(\mathbf{S}_k - \mathbf{S}_{k-1})}{\Delta t} \times \Delta t\right)\right) \div 2. \tag{9}$$

The presented Kalman filter consists of a prediction stage and a correction stage. The prediction stage can be expressed as follows:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k\text{-}1}, \tag{10}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}, \tag{11}$$

where $\hat{\mathbf{x}}_k^-$ is the a priori estimate state and $\mathbf{P}_k^-$ is the a priori estimate error covariance matrix in frame k. The correction stage can be expressed as follows:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}\right)^{-1}, \tag{12}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k^s - \mathbf{H}\hat{\mathbf{x}}_k^-), \tag{13}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-, \tag{14}$$

where $\hat{\mathbf{x}}_k = \begin{bmatrix}\hat{\mathbf{S}}_k & \hat{\mathbf{V}}_k\end{bmatrix}^T$ is the a posteriori estimate state at frame k, $\mathbf{P}_k$ is the a posteriori estimate error covariance matrix at frame k, and $\mathbf{K}_k$ is the Kalman gain. Here, we replace the measurement vector $\mathbf{z}_k$ with the smoothed measurement vector $\mathbf{z}_k^s$. The Kalman-based motion prediction is a recursive process. The state estimated in (13) will be used in the right side of (10) to calculate the next frame $k+1$, and also in (9) for smoothed measurement vector calculating.

Predicted by the above Kalman filter, we can obtain a robust position prediction $\hat{\mathbf{S}}_k^-$ of the served astronaut for each frame. Then the probability $P^k(p \mid L_i)$ can be defined as follows:

$$P^k(p \mid L_i) = \begin{cases} 1, & \left\|\mathbf{S}_k - \hat{\mathbf{S}}_k^-\right\| \le R_{\text{thre}}, \\ 0, & \left\|\mathbf{S}_k - \hat{\mathbf{S}}_k^-\right\| > R_{\text{thre}}, \end{cases} \tag{15}$$

where $\mathbf{S}_k$ is the measured position of the served astronaut. R_{thre} is a key threshold value, and selection of its value needs trade-off consideration. If the parameter value is too small, the tracking procedure may be easily affected by the consequent missed detection. If the value is too big, mistaken tracking may occur when multiple astronauts are much closer together. Here, we set R_{thre} to be 30 centimeters after weighing the above two factors.

4.3. Geometric Similarity Matching of Bounding Box. Spatial motion prediction described above treats astronaut as a particle model. However, taking into account the limited space in the space station, the body size of astronaut should not

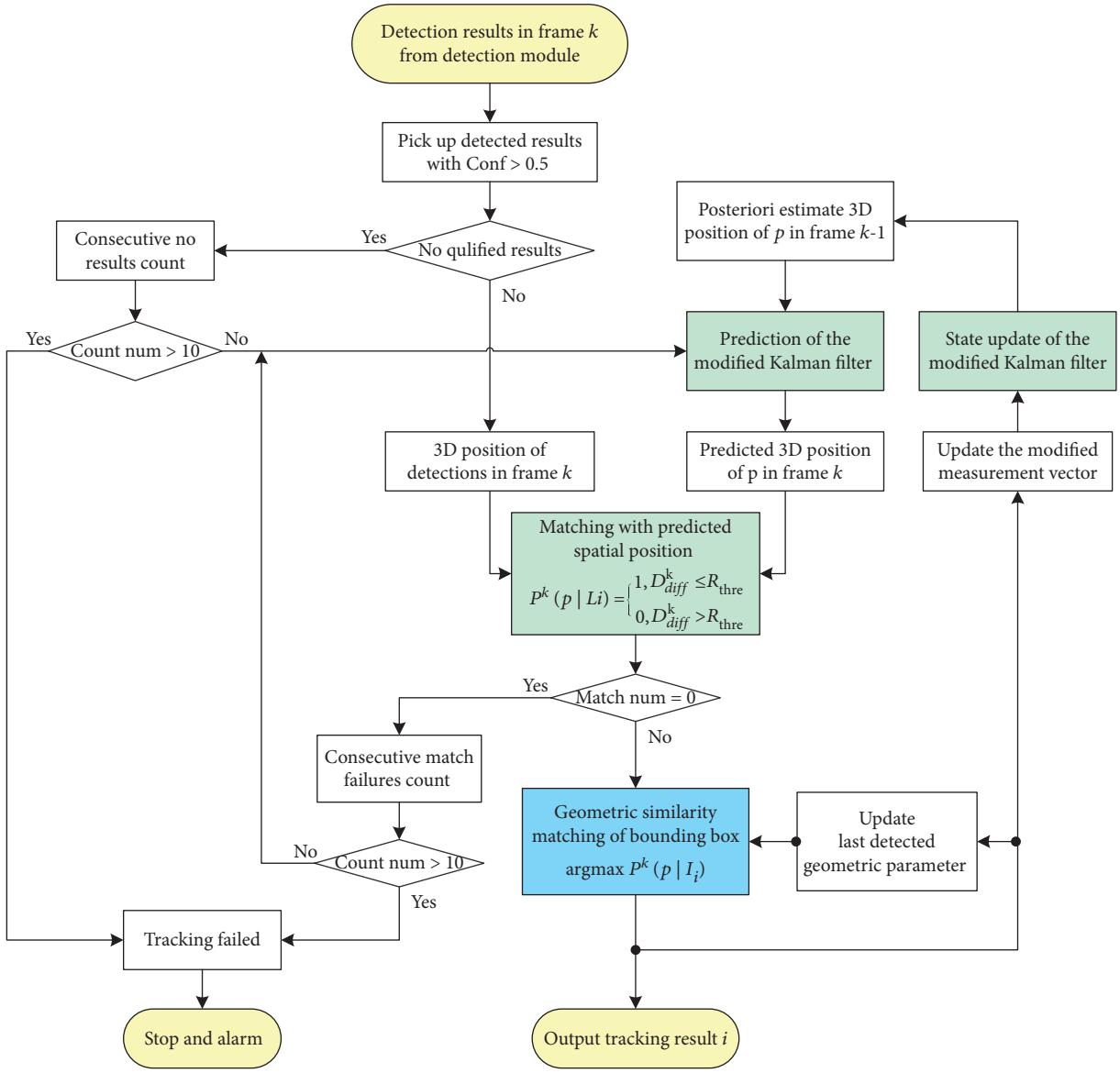


Figure 12: Flowchart of the astronaut tracking module.

be ignored. Body size range of astronaut under movement is actually the bounding box detected in color image. Therefore, the probability $P^k(p \mid I_i)$ is described by the similarity of color image in bounding boxes between the current detection and the previous tracked one.

In Figure 14, the solid rectangle represents the bounding box of the tracked astronaut in the last frame $k-1$, while the dashed rectangles represent the detected astronauts in the current frame k. We know that the bounding box of the tracked astronaut changes and moves little between sampling intervals in color image. Then, the bounding box which is closer to and more similar in shape with the tracked bounding box is more likely to be the tracked astronaut.

Then, the probability $P^k(p \mid I_i)$ can be defined as follows:

$$P^k(p \mid I_i) = e^{-\left[\alpha_0 \cdot \Delta D_i^k + \alpha_1 \left(\sqrt{\Delta S_i^k} + \alpha_2 \cdot \Delta K_i^k\right)\right]}. \quad (16)$$

In (16), $i = 1, 2, \ldots, m$ and m denotes the number of detected results in the frame k. α_0, α_1, and α_2 are constant parameters and $\alpha_0 = 1$, $\alpha_1 = 1$, and $\alpha_2 = 0.8$. ΔD_i^k is the distance between the ith detected bounding box in the frame k and the tracked bounding box in the frame $k-1$. It can be calculated by

$$\Delta D_i^k = \sqrt{\left(u_i^k - u^{k-1}\right)^2 + \left(v_i^k - v^{k-1}\right)^2}, \quad (17)$$

where (u^{k-1}, v^{k-1}) is the center point of the tracked bounding box in frame $k-1$. (u_i^k, v_i^k) is the center point of the $i$thdetected bounding box in frame k, and its coordinate values can be easily calculated from the coordinate values of the bounding box. ΔS_i^k is the difference in area between the ith detected bounding box in frame k and

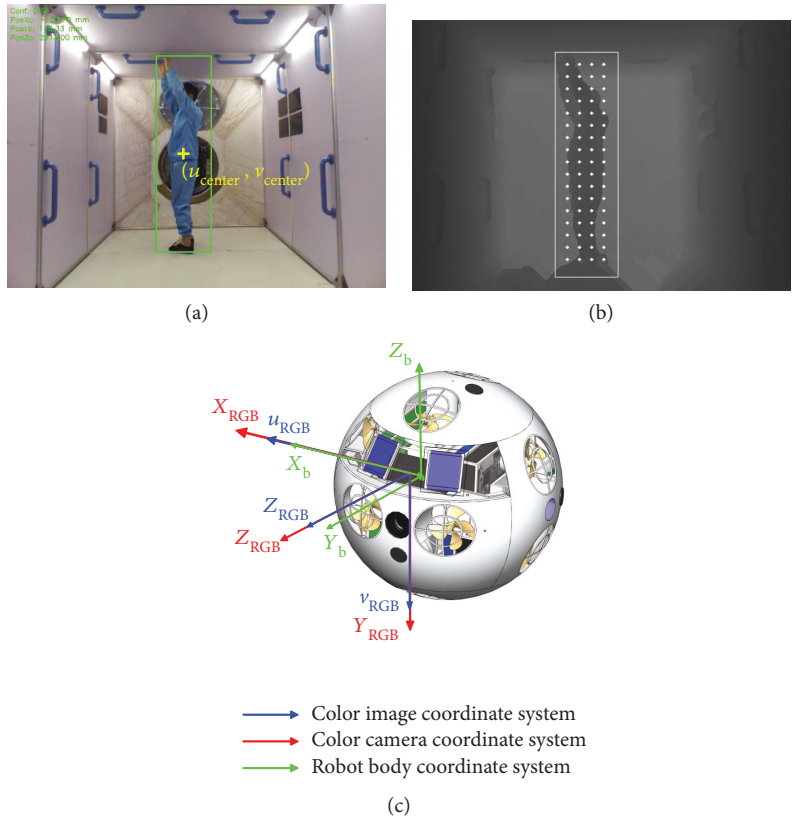


Figure 13: Images of RGB-D camera and coordinate system definition.

the tracked bounding box in frame $k-1$. It can be calculated by

$$\Delta S_i^k = \left| w_i^k h_i^k - w^{k-1} h^{k-1} \right|, \tag{18}$$

where w_i^k and h_i^k are width and height of the ith detected bounding box, respectively. w^{k-1} and h^{k-1} are width and height of the tracked bounding box. ΔK_i^k is the difference in aspect ratio between the ith detected bounding box in the frame k and the tracked bounding box in the frame k-1. It can be calculated by

$$\Delta K_i^k = \left| \frac{w_i^k}{h_i^k} - \frac{w^{k-1}}{h^{k-1}} \right|. \tag{19}$$

5. Experiments and Results

In order to verify our proposed astronaut visual tracking algorithm, we present our experiments and results in this section. The experimental videos which contain most instances of astronaut activities are recorded by a Kinect RGB-D camera with a resolution of 640×480 at 30 fps speed. Our proposed algorithm is implemented under Ubuntu 16.04 with C++ language and runs 10 fps on Jetson TX2, detecting and tracking the served astronaut every other two input images. Two sets of experiments are carried out to validate the proposed astronaut visual tracking algorithm.

In the first experiment, one subject is asked to follow a reference path of $1.4\,\text{m} \times 2\,\text{m}$ square with various postures. We evaluate the tracking performance of our proposed tracking algorithm by whether accurate and consecutive tracking is obtained or not, despite the subject's diverse postures or various distance relative to the robot. Besides, the reconstruction performance of the subject's spatial position relative to the camera is also tested in this experiment. We evaluate it by comparing the experimental results with the reference path.

The reference path following experiment is carried out in our space station mockup. Although the subject's dressings can be arbitrary in three different types, we just discuss one kind of them here as an illustration and the other cases have the similar results. As described in Section 2.2, the astronaut detection module is activated first when the tracking task begins. The detection results are located by red bounding boxes in each image frame. Here, we ignore the initial 2 frames of poor quality. Then, the first 5 consecutive frames in which human detection is successful are used to confirm tracking target and initialize the modified Kalman filter. After that, the astronaut tracking module is activated to track the predefined target. Bounding box of the detection will change green if the detection is judged as the target that we are tracking. Results of the five tests in Figure 15 show that our proposed tracking algorithm achieves accurate and consecutive tracking performance in all the instances.

Figure 16 shows the reconstruction of the subject's spatial position relative to the camera in each test. The left pictures

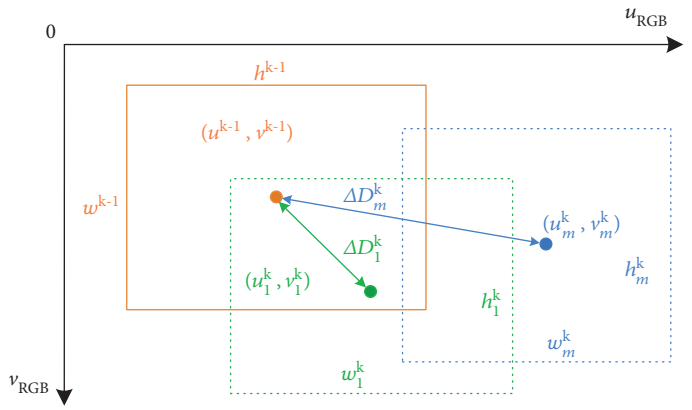
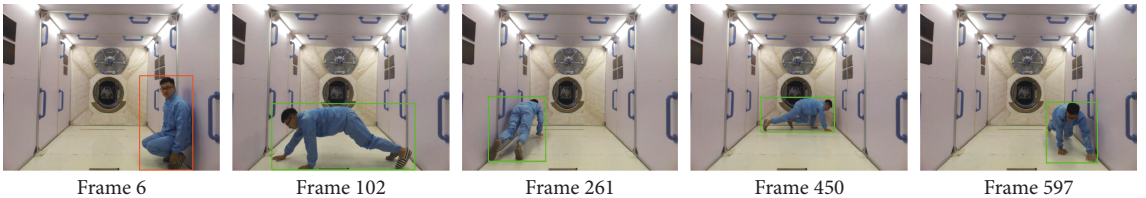


Figure 14: Geometric similarity matching of bounding box in color image.

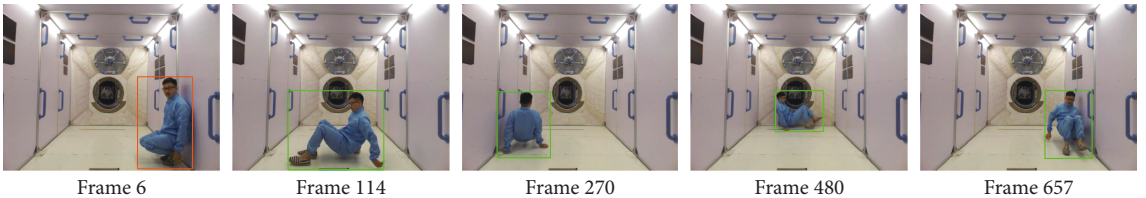
(a)

(b)

(c)



(d)



(e)

Figure 15: Follow a reference path with five different postures.

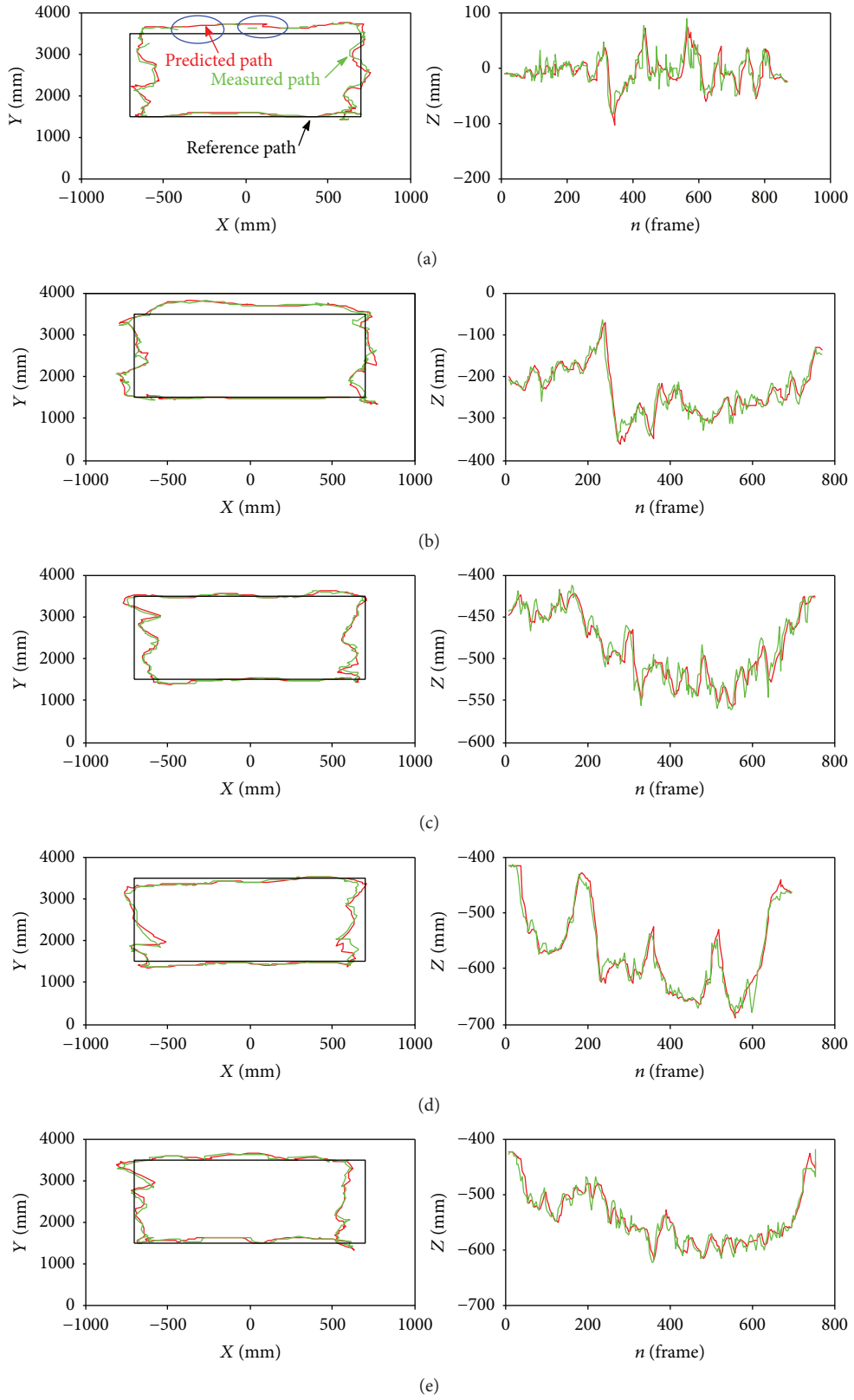


Figure 16: Prediction and measurement results of the tracked subject's spatial position.

indicate the people's moving paths projected to the camera's XY coordinate plane. The right pictures indicate the people's spatial position in the camera's Z coordinate. Black rectangles in the left pictures represent the 1.4 m × 2 m rectangle reference path. Green points represent the measured position while red points represent the predicted position. It can be

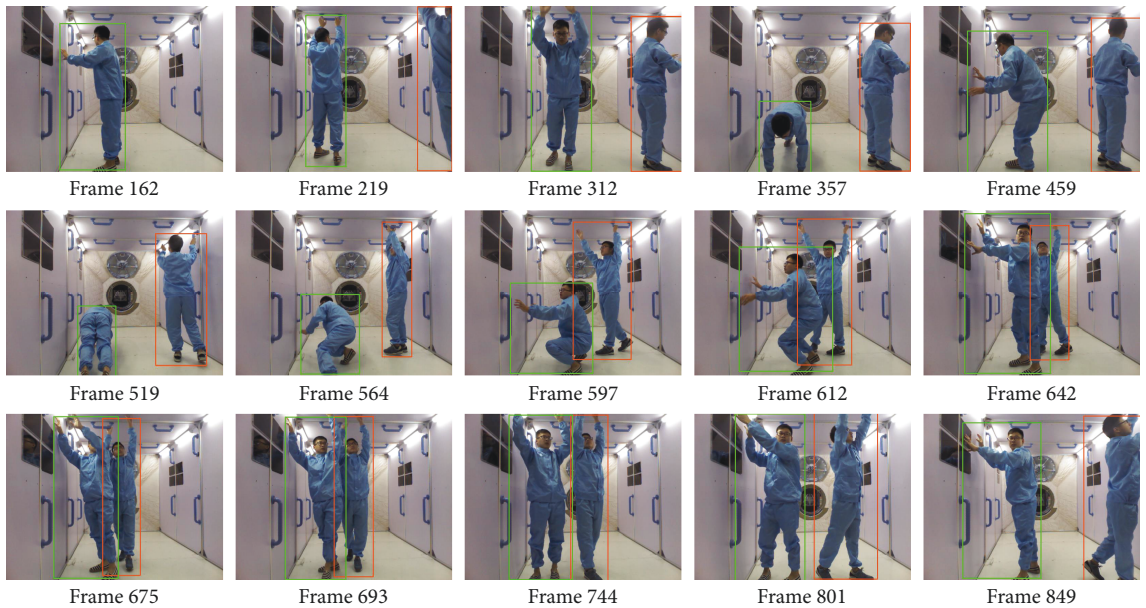


Figure 17: Tracking with other subject entering and moving very close to the target subject.

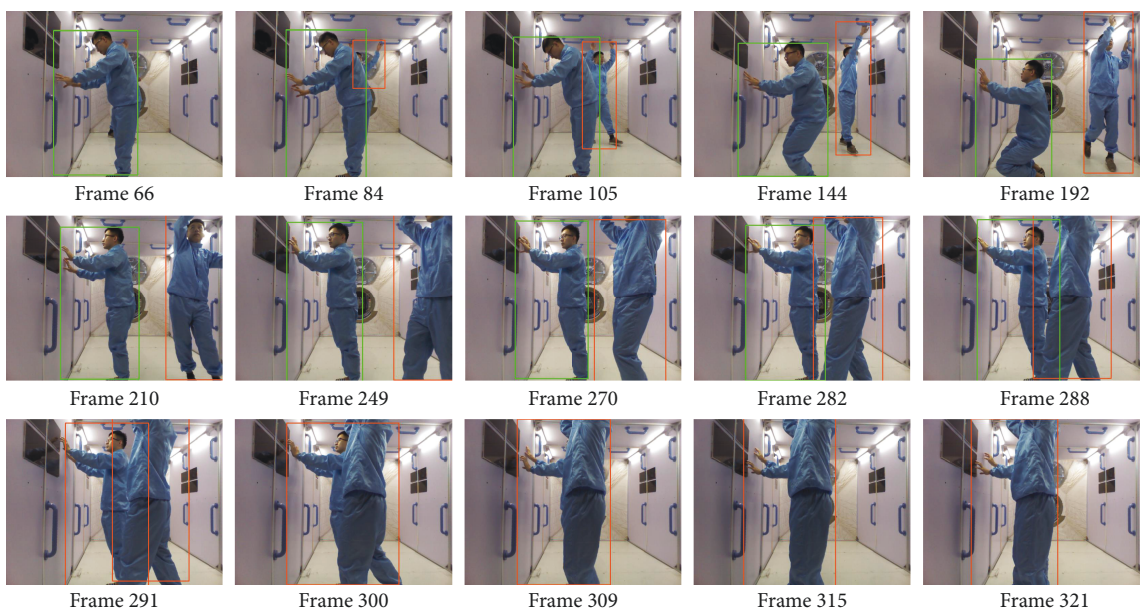


Figure 18: Occlusion detection for avoiding mistaken tracking.

observed that in the blue ellipses of Figure 16(a), no qualified detection has been tracked in several consecutive frames. Our proposed Kalman filter provides well predictions thus avoiding possible tracking failures in such situations. Moreover, the reconstruction of the subject's spatial position relative to the camera is basically the same as the ground truth. The deviation is no more than 20 cm. It is acceptable allowing for the path following error caused by the subject and the interference of his arbitrary body movements.

In the second experiment, we test the more common situations that the robot hovers or follows the served astronaut during assistance offering period. Here, we consider the most common case that there are two crews in the cabin and they are both dressed in blue in-cabin work clothes. We first set a subject as the tracking target. The subject is asked to keep a certain distance from the camera and imitate astronaut's working or walking activities. Meanwhile, another subject may enter or exit the scene. The two subjects may move very close in position, and the tracking target may be partially or entirely occluded by the other one.

Figure 17 shows the tracking image sequences with other subject entering and moving very close to the target subject. The target subject stands by the console panel performing operation or walks in the cabin by climbing. Another subject

enters at the 219th frame and moves very close to the target subject at the 675th frame. Our proposed astronaut tracking algorithm achieves stable and accurate tracking performance. The successful tracking result is located by green bounding box as shown in the image sequences.

Figure 18 shows the tracking image sequences with occlusion which happened between the target subject and other subject. The other subject enters the scene at the 84th frame and moves very close to the target subject at the 282th frame. He then breaks into the interspace between the camera and the target subject. We begin to lose track with the target subject at the 291th frame. The tracking algorithm causes a stop at the 321th frame when successive 10 tracking failures are detected. As a result, our proposed astronaut tracking algorithm successfully detects the occlusion and avoids the mistaken tracking that treats other subject as the target subject.

6. Conclusion

Astronaut visual tracking is the most important prerequisite for in-cabin assistant robot to formation fly with the served astronaut and offers assistance in the space station. In this paper, a deep learning and probabilistic model-based visual tracking algorithm is proposed for Intelligent Formation Personal Satellite to track the served astronaut in RGB-D videos. An improved SSD network-based detection module was first proposed for detecting people with arbitrary postures, gestures, or dressings. Our proposed network performs stronger generalization ability in real detection task, resulting better detection accuracy than the original SSD network. And it runs faster than the original SSD network, supporting real-time applications on embedded AI computer Jetson TX2. Based on the relatively clean detection results, a probabilistic model-based tracking module is proposed for accurate and consecutive tracking with the specified person. A modified Kalman filter is designed to predict the spatial position of the astronaut relative to the robot. Then, a matching with predicted spatial position and a geometric similarity matching of bounding box are associated for robust tracking. We extensively validated our algorithm with several datasets that contain most instances of astronaut activities. The algorithm runs at 10 fps on Jetson TX2. The experimental results show that our proposed astronaut visual tracking algorithm achieves not only robust tracking of the specified person with diverse postures or dressings but also effective occlusion detection for avoiding mistaken tracking. The study in this paper provides a powerful means for Intelligent Formation Personal Satellite to realize the real-time astronaut visual tracking in future application on orbit. Besides, our proposed astronaut tracking algorithm can also be applied in any other people-following robots as it could be easily constructed by common RGB-D camera and embedded GPUs.

Data Availability

We regret that access to data is restricted for the time being. The data used to support the findings of this study have not been made available because the protection of technical privacy and confidentiality.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] G. A. Dorais and Y. Gawdiak, "The personal satellite assistant: an internal spacecraft autonomous mobile monitor," in *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, pp. 333–348, Big Sky, MT, USA, 2003.

[2] T. Smith, J. Barlow, M. Bualat et al., "Astrobee: a new platform for free-flying robotics on the international space station," in *13th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, pp. 83–86, Beijing, China, 2016.

[3] T. Fong, M. Micire, T. Morse et al., "Smart SPHERES: a telerobotic free-flyer for intravehicular activities in space," in *AIAA Space 2013 Conference and Exposition*, pp. 5338–5352, San Diego, CA, USA, 2013.

[4] A. Saenz-Otero and D. W. Miller, "SPHERES: a platform for formation-flight research," in *UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts II*, p. 58990O, San Diego, CA, USA, 2005.

[5] Y. Tsumaki and I. Maeda, "Intra-vehicular free-flyer with manipulation capability," *Advanced Robotics*, vol. 24, no. 3, pp. 343–358, 2010.

[6] J. Liu, Q. Gao, Z. Liu, and Y. Li, "Attitude control for astronaut assisted robot in the space station," *International Journal of Control, Automation and Systems*, vol. 14, no. 4, pp. 1082–1095, 2016.

[7] Q. Gao, J. Liu, T. Tian, and Y. Li, "Free-flying dynamics and control of an astronaut assistant robot based on fuzzy sliding mode algorithm," *Acta Astronautica*, vol. 138, pp. 462–474, 2017.

[8] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[9] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: a survey," *Pattern Recognition*, vol. 51, pp. 148–175, 2016.

[10] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 973–986, 2018.

[11] G. Monteiro, P. Peixoto, and U. Nunes, "Vision-based pedestrian detection using Haar-like features," *Robotica*, vol. 24, pp. 46–50, 2006.

[12] X. Cui, Y. Liu, S. Shan, X. Chen, and W. Gao, "3D Haar-like features for pedestrian detection," in *2007 IEEE International Conference on Multimedia and Expo*, pp. 1263–1266, Beijing, China, 2007.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, San Diego, CA, USA, 2005.

[15] D. Sugimura, T. Fujimura, and T. Hamamoto, "Enhanced cascading classifier using multi-scale HOG for pedestrian detection from aerial images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 3, article 1655009, 2016.

[16] C. H. Chuang, Z. Y. Lian, P. R. Teng, and M. J. Lin, "Human detection for video surveillance in hospital," *Journal of Electronic Science and Technology*, vol. 2, 2017.

[17] M. Munaro, C. Lewis, D. Chambers, P. Hvass, and E. Menegatti, "RGB-D human detection and tracking for industrial environments," in *Intelligent Autonomous Systems 13*, pp. 1655–1668, Springer, 2016.

[18] L. Qu and J. S. Lim, "A novel way of pedestrian detection using neural network with a weighted fuzzy membership function," *Advanced Science Letters*, vol. 22, no. 11, pp. 3516–3519, 2016.

[19] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838–3843, San Francisco, CA, USA, 2011.

[20] X. Chen, K. Henrickson, and Y. Wang, "Kinect-based pedestrian detection for crowded scenes," *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 3, pp. 229–240, 2016.

[21] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Procedings of the British Machine Vision Conference 2009*, pp. 1–11, London, UK, 2009.

[22] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Computer Vision – ECCV 2012*, Lecture Notes in Computer Science, pp. 645–659, Springer, 2012.

[23] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[25] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pp. 379–387, 2016.

[26] D. Impiombato, S. Giarrusso, T. Mineo et al., "Characterization and performance of the ASIC (CITIROC) front-end of the ASTRI camera," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 794, pp. 185–192, 2015.

[27] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pp. 21–37, Springer, 2016.

[28] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4073–4082, Boston, MA, USA, 2015.

[29] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing LIDAR and images for pedestrian detection using convolutional neural networks," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2198–2205, Stockholm, Sweden, 2016.

[30] W. Ouyang, X. Zeng, and X. Wang, "Learning mutual visibility relationship for pedestrian detection with a deep model," *International Journal of Computer Vision*, vol. 120, no. 1, pp. 14–27, 2016.

[31] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang, "Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[32] H. Xue, Y. Liu, D. Cai, and X. He, "Tracking people in RGBD videos using deep learning and motion clues," *Neurocomputing*, vol. 204, pp. 70–76, 2016.

[33] E. Petrović, A. Leu, D. Ristić-Durrant, and V. Nikolić, "Stereo vision-based human tracking for robotic follower," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 230, 2013.

[34] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5636–5643, Hong Kong, 2014.

[35] J. Liu, Y. Liu, G. Zhang, P. Zhu, and Y. Q. Chen, "Detecting and tracking people in real time with RGB-D camera," *Pattern Recognition Letters*, vol. 53, pp. 16–23, 2015.

[36] H. Liu, J. Luo, P. Wu, S. Xie, and H. Li, "People detection and tracking using RGB-D cameras for mobile robots," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, 2016.

[37] Q. Zhong, C. Li, Y. Zhang et al., "Cascade region proposal and global context for deep object detection," 2017, https://arxiv.org/abs/1710.10749.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, https://arxiv.org/abs/1409.1556v6.