Recognition of Pedestrian Activity based on Dropped-object Detection

Weidong Min , Yu Zhang , Jing Li , Shaoping Xu

PII: S0165-1684(17)30346-8
DOI: 10.1016/j.sigpro.2017.09.024
Reference: SIGPRO 6616

To appear in: *Signal Processing*

Received date: 14 April 2017
Revised date: 21 September 2017
Accepted date: 22 September 2017

Please cite this article as: Weidong Min , Yu Zhang , Jing Li , Shaoping Xu , Recognition of Pedestrian Activity based on Dropped-object Detection, *Signal Processing* (2017), doi: 10.1016/j.sigpro.2017.09.024

Highlights

- Pedestrian activity is recognized based on dropped-object detection.
- The method analyzes the relationship between the dropped objects and pedestrians.
- The method is implemented in a distributed model.

Recognition of Pedestrian Activity based on Dropped-object Detection

Weidong Min [1], Yu Zhang [1], Jing Li [1,*], Shaoping Xu [1],

1. School of Information Engineering, Nanchang University, Nanchang 330031, China

* Corresponding author: Jing Li, jing.li.2003@gmail.com

Abstract: Aiming at recognizing dropped objects and matching their owners, this paper presents a method for analyzing pedestrian activity based on dropped-object detection in video surveillance. The recognition results may be applied to further analyzing human activity and intentions such as determining whether the dropped-objects are intentional hazardous or unconsciously lost articles according to the appearance of dropped-objects. The method consists of dropped-object detection and recognition. The dropped-object detection algorithm uses foreground detection based on bi-directional background modeling, MeanShift tracking, and pixel-based regional information at the drop-off point. It analyzes the relationship between the dropped objects and pedestrians at the pixel level in complex environments with noises and occlusions. Afterwards, an algorithm based on moment invariant and Principal Component Analysis (PCA) is proposed to further recognize the dropped-objects viewed from different directions and locations from video cameras. In addition, in order to solve the limitation of the centralized video processing model for large-scale video streams in real time, the proposed method is designed and accomplished in a distributed model. The experimental results showed that the proposed method can effectively and efficiently recognize the pedestrian activity through the dropped objects in real-time video data.

Keywords: Dropped-object detection; dropped-object recognition; relevancy analysis; distributed model; human activity analysis; video surveillance

1. Introduction

With the rapid development of urbanization, i.e., population growth and geographic expansion, it is nearly impossible to monitor and protect the public by security sectors. Especially at the level of terrorist attacks in crowded public places, such as airports, railway stations, busy streets, etc., it is difficult and tedious to recognize pedestrian behaviors in those complex environments. This situation has led to the necessity to develop a surveillance system with smart and precise analysis, which can help monitoring staffs quickly search some suspicious items or someone when dangerous or unusual things happen.

Recognition of pedestrian activity based on dropped-object (also called as abandoned-object) detection is one of important monitoring tasks in both real-time video surveillance systems and offline data mining from stored video data. The purpose of dropped-object detection is to automatically detect static objects, alert monitoring staffs, and identify the object owner's behaviors. Those dropped objects should be deemed as items such as backpacks, handbags, luggage, etc. keeping static in a defined period of time. Through automatic detection, staffs could be informed timely and distinguish the items' owners in a short time. What is more important, the recognition results may be applied to further analyzing human activity and intentions such as determining whether the dropped objects are intentional hazardous articles or unconsciously lost articles according to the appearance of dropped objects. Several challenges have been involved in the detection task like complex scenes with dynamically changing background, illuminations and lighting conditions, as well as object occlusions, inaccurate or incomplete information when people stand still by the objects. Despite of recent advanced facilities and computer vision technologies, there is little attempt to adopt effective object detection and identification techniques to help improve the accuracy of dropped-object detection and reduce false alarms.

In this paper, we propose an algorithm to detect dropped objects and analyze the relevancy between the objects and people nearby based on regional information. Furthermore, we propose a dropped-object recognition algorithm based on moment invariant [1] and Principal Component Analysis (PCA). In earlier works, Zhan [2] put forward a detection algorithm based on self-organizing map through artificial neural networks to effectively deal with noise and illumination changes. Multi-layer background subtraction based on RGB color space and local binary patterns [3] was proposed to detect dropped objects. This approach is able to handle local illumination changes such as cast shadows from moving objects. Moreover, it obtains higher detection accuracy by working in single views and exploiting prior information of the scene, but it requires additional computation in managing the background layer and sophisticated hardware **Error! Reference source not found.**. A dropped-object detection system was raised based on a dual-time background subtraction algorithm and an approximate median model [5]. A backtracking algorithm [6, 7] was proposed to analyze the relationship between the objects and the owners. After detecting the abandoned object, it checked each of the previous frames to find the time of object dropping and the pedestrian. Fan [8] analyzed the relevance of abandonment. However, it cannot track the owner to reduce the risks of left items. Tian et al. [9] modeled background by three Gaussian mixtures, which is able to adapt to quick lighting changes, fragment reduction, and keep a stable update rate for video streams with different frame rates. An effective approach was proposed to detect abandoned luggage by long-term and short-term background models for the premise of the backtracking verification method

[10], where each pixel in an input image is classified as a 2-bit code. A framework was introduced in this method to identify static foreground regions based on the temporal transition of code pattern. Jayasuganthi [11] proposed to detect protruding as well as non-protruding objects in sequences of walking pedestrians based on the texture of foreground objects, where k-means was used over data streams in order to find the outliers, i.e. the dropped objects. An improved semantic segmentation approach [12] based on Joint Systems Engineering Group (JSEG) algorithm and multiple region merging criteria generated meaningful regions and detected salient objects. Caterina [13] proposed an algorithm based on background elimination and a distributed architecture. A multimedia communication system based on direct sequence code-division multiple-access (DS/CDMA) techniques aims at ensuring secure and noise-robust wireless transmission links between guarded stations and the remote control center [14]. The frameworks of spatiotemporal background modeling were proposed for adapting to illumination and dynamic changes, which can detect foreground objects robustly [15]. Timofte [16] tracked the object through modeling it as a set of pixel-level templates with weak configuration constraints, using localization process and obtaining information constantly to increase robustness. For the saliency detection, Wang proposed multiple-instance learning, multi-cue information and multi-spectrum methods [17-19]. With the rapid improvement of neutral networks, they have been extensively used in object detection by combining other algorithms [20-28].

Nowadays, recognition is mainly conducted based on knowledge and algebraic features, especially Hu's moment invariants [29], SIFT [30], and their improved ones. For example, Hu's moment invariants were improved in precision [31] and practicability [32]. PCA-SIFT demonstrated that PCA-based local descriptors are more distinctive, more robust to image deformations, and more compact than the standard SIFT representation [33]. SVD-SIFT [34] is an easier and more compact algorithm that can be used to improve the recognition speed for severe scene variations. Also, SIFT-SPMK and TST-SIFT are the improvement of SIFT for increasing the accuracy [35]. Besides, a methodology called Weighted Gaussian Kernel Video Segmentation (WGKVS) was proposed to construct a background model and incorporate multiple information sources by a Multiple Kernel Learning (MKL) framework, which performs background subtraction to enhance the representation of each pixel. However, it is not a real-time algorithm [36]. Juang et al. [37] computed the object's color histograms and improved accuracy by histogram-based fuzzy classifier with support vector learning. Sheikh [38] proposed a detection criterion and modeled the background as a single probability density. In addition, Canny [39] and its improved algorithms were used to detect and match the edges of objects for the recognition purpose. Xie [40] proposed a multi-label material recognition algorithm by Directed Acyclic Graph (DAG) which is used to assist the inference of surface materials. The development of

machine learning introduces a new way to recognize and identify objects with a very high accuracy rate through big data training and feature extraction [41-43]. For multiple objects, a deep recurrent neural network was trained with reinforcement learning to focus on the most relevant regions of the input image, while using fewer parameters and requiring less computation [44]. Jarrett [45] showed that using non-linearities and two stages of feature extraction can achieve better accuracy. Yang [46] improved the recognition accuracy by CNN through extracting object proposals from each image and using nearest-neighbor relationships of local regions to form a multi-view pipeline.

The rest of the paper is organized as follows. In Section 2, we present the detail of the proposed detection and recognition algorithms, and the distributed architecture model. We apply our newly proposed method to some videos for dropped-object detection in Section 3, including the comparison with other state-of-art methods on their public available datasets. Conclusions and future works are given in Section 4.

2. Dropped-object Detection and Recognition Models on a Distributed Architecture

2.1 Dropped-object Detection

In order to solve the limitation of the centralized video processing model for large-scale video streams in real time, our proposed method detects and identifies dropped items on a distributed architecture. The model is mainly divided into three parts, i.e. server, agent and monitoring equipment, in which the processes of video collection, video detection, and video recognition are separated, as shown in Fig. 1.

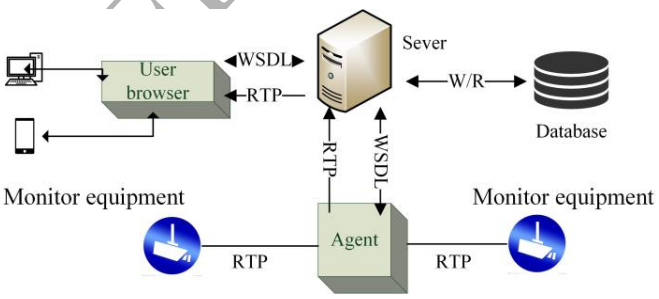


Fig. 1. A fundamental distributed model for video processing.

We propose a region-based algorithm for foreground detection and mobile pedestrian tracking of the target area in each video frame. The algorithm detects blob area (connected domain in a foreground image) to track the target. When a person abandons an item and once the item is separated from the person, a new non-tracking target (item) area appears in the foreground image of the current frame. According to the characteristics of MeanShift [Ref], it is very likely that the tracked

target (pedestrian) region will partially coincide with the non-tracking target (object) region at that moment with the shortest Euclidean distance between two centroids. On this point, the relevance between the object and the pedestrian can be deducted. The owner (pedestrian) of the abandoned item then continues to be detected and tracked until s/he leaves the video scene. After that time, the algorithm determines whether the item is a legacy or not based on its stationary time. In this way, we can find the owner of the dropped object and also know the time and location when/where the object was dropped.

In this paper, related definitions are given as follows.

Definition 1: We define a list of item areas used to store each non-tracking target area by $L = \{Ob_1, Ob_2, Ob_3, \cdots, Ob_n\}$. Here, L is defined as the circumscribed rectangular region of the connected region (blob) in the obtained binary image through foreground detection in the current frame, T is defined as the circumscribed rectangle area of the tracking target area obtained by tracking each pedestrian, and NT denotes an outer rectangle area with no blobs (non-tracking target area) which do not coincide with the tracking target area in the foreground image. As defined in Eq. (1)-(4), R_c is the intersecting region between several blobs or different circumscribed rectangular regions (e.g., R_1 and R_2) of the tracking target, P is the ratio of R_c to the sum of the areas, Q is the ratio of R_c to R_1, and D denotes the inter-region Euclidean distance between two centroids of the circumscribed rectangular area, where (x_1, y_1) is the centroid of R_1 and (x_2, y_2) is the centroid of R_2.

$$R_c = R_1 \bigcap R_2 \tag{1}$$

$$P = \frac{R_c}{R_1 \bigcup R_2} \tag{2}$$

$$Q = \frac{R_c}{R_1} \tag{3}$$

$$D = \sqrt[2]{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4}$$

Definition 2: Given a user-defined time threshold T, when the item's silent period $T' \geq T$, we judge this item as a dropped item according to the following Eq. (5) and (6).

$$FrameLast - FrameFirst \geq FrameCount \tag{5}$$

$$FrameCount = T \times FPS \tag{6}$$

In other words, when $T' \geq FrameCount/FPS$, the object is deemed as a dropped object.

We define *FrameCurrent* as the current frame number of the video, *RectCurrent* as the information of the external rectangle of each non-tracking object in the current frame, *FrameFirst* as the frame number when the object first appears, *FrameLast* as the frame number when the object last appears, *RectFirst* is the external rectangle of the area information for the first time, *RectLast* is the external rectangle of the area information for the last time. The variable *bLose* determines whether the object is a legacy, and *ownerindex* is the tracking serial number of the object. For NT_n (a component of *NT*), we set *FrameFirst=FrameCurrent*, *FrameLast=FrameCurrent*, *RectFirst=RectCurrent*, *RectLast=RectCurrent*, *bLose=false*, *ownerindex=n*. Besides, *B* and *T* are obtained by the foreground detection and tracking algorithm, *Q* can be calculated from T_n and B_n based on Eq. (1) and (3), *D* and *P* are calculated from NT_n and T_n, with the minimum D_{min} for *D* and the maximum P_{max} for *P*, respectively, and we obtain R_c by calculating the properties of Ob_n and T_n based on Eq. (1).

The pseudo code of our proposed dropped-object detection algorithm is shown in Table 1.

Table 1. Dropped-object Detection Algorithm

Input: An original image PImage

Output: A detected image PI

Begin

for $j \leftarrow 1$ **to** $|m|$ **do** // Get B_m **for** $i \leftarrow 1$ **to** $|n|$ **do** // Get T

if $B_m \cap T_n / T_n \leq 0.2$ **then** $NT_m \leftarrow B_m$; **end if**

end for

for $k \leftarrow 1$ **to** $|r|$ **do** // Get Ob_r

if $Ob_r \cap NT_m / Ob_r > 0.8$ or Distance(Ob_r , NT_m) ≤ 5 **then**

FrameLast $\leftarrow$ *FrameCurrent*; *RectLast* $\leftarrow$*FrameCurrentRect*;

else

for $i \leftarrow 1$ **to** $|n|$ **do**

$P = NT_m \cap T_n / (NT_m \cup T_n)$;

if $P = Max > 0$ **then**

NT_m relevance to T_n;

bLose = false;

Index(NT_m) ←Index(T_n);

Add NT_m to L;

r++;

else if P = Max = 0 **then**

$D = Distance(NT_m, T_n)$;

if $D = Min$ **then**

NT_m relevance to T_n;

bLose = false;

Index(NT_m) ←Index(T_n);

Add NT_m to L;

r++;

end if

else error processing;

end if

end for

for k'←1 **to** $|R|$ **do** // Get Ob_R

if $FrameCurrent - FrameLast > 5$ **then**

Delete Ob_R from L;

else if $FrameLast - FrameFirst > T * FPS$ **then**

bLose =true； // Ob_R is a dropped object

else Continue；

end if

end for

end for

End

At first, we apply foreground detection based on bidirectional background modeling to effectively detect a set of stationary objects in the video. Then, we use the improved MeanShift algorithm to track the moving pedestrians in the foreground image. According to the characteristic of MeanShift, at the moment of pedestrian abandonment (separation of dropped items and pedestrians), the tracking target (pedestrian) region is most likely to coincide with a non-tracking target (item) region or containing a non-tracking target (item) region. The mathematical expression is given in Eq. (7),

$$R_1 \bigcap R_2 > 0 \ \text{OR} \ R_1 \supset R_2 , \tag{7}$$

where R_1 is the tracking target area and R_2 is the non-tracking target area. Afterwards, we use the Kalman filter to eliminate the noise generated during the tracking and improve the tracking accuracy. According to the proportion of T_n in the overlap of B_n and T_n, we can determine whether B_n belongs to NT. By comparing the Euclidean distances between the centroids of NT_n and previously stored objects in the item list L, we can judge whether NT_n is new. If so, NT_n is stored as an object in L.

In general, we analyze the correlation and judge whether the object is a leftover item according to its still time. Therefore, this algorithm can quickly detect lost items and analyze the relevance, and rapidly handle real-time video streamd when applied to large-scale automated video surveillance.

2.2 Dropped-object Identification

In this paper, the recognition of legacy objects is achieved by combining the geometric affine invariant moment and PCA. Because the dropped-object recognition algorithm combines their characteristics, it can solve the problem of view-angle changes (the front view and the side view of the objects), including rotation, translation, zoom and tilting. The framework of our algorithm is given in Fig. 2.

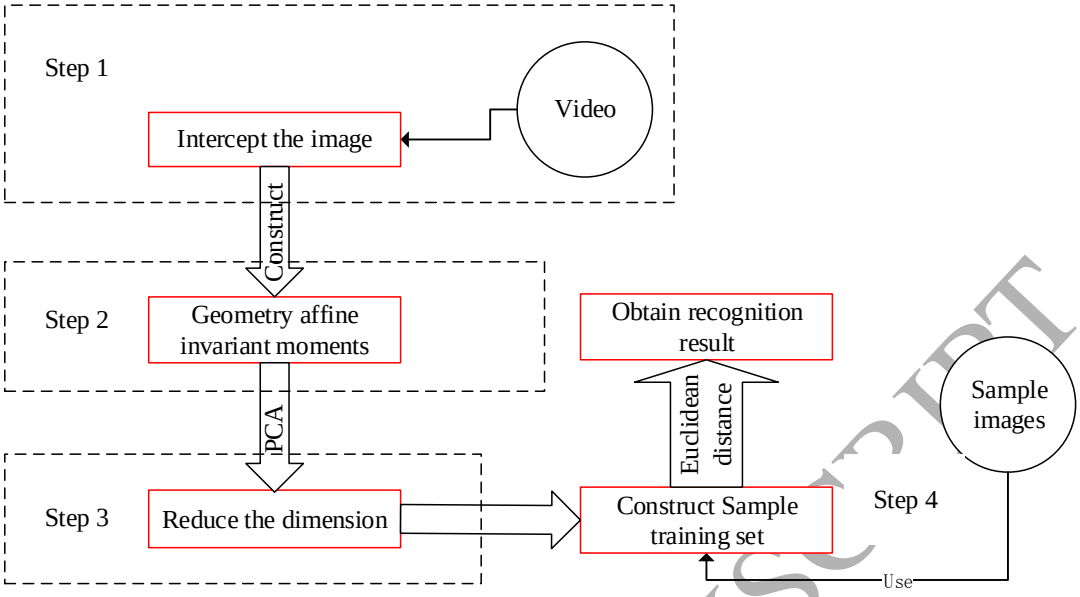


Fig. 2. The framework of dropped-object recognition.

First, the area of dropped-item detection results is stored and geometry affine invariant moments are constructed. Then, PCA is used to reduce the dimension of its invariant matrix, in order to quickly identify the dropped items in the video. Finally, the recognition results are obtained after matching the Euclidean distance between the test image and the sample images in the object library.

The procedure of the algorithm is described in detail as follows.

Step 1: We compute the normalized invariant moments for n samples. The invariant matrix is expressed in Eq. (8).

$$Y = (F_1, F_2, \cdots, F_6) \tag{8}$$

where n represents the number of images and F_i = (I_{i1}, I_{i2}, $\cdots$, I_{in}) (i = 1, 2, $\cdots$, 6) denotes the i-th moment of different images, as defined in Eq. (9) and (10).

$$\begin{cases} I_5 = (u_{40} - 4u_{31}u_{13} + 3u_{22}{}^2)/u_{00}{}^6 \\ I_6 = (u_{40}u_{04}u_{22} + 2u_{31}u_{22}u_{13} - u_{40}u_{13}{}^2 - u_{04}u_{310}{}^2 - u_{22}{}^3)/u_{00}{}^9 \end{cases} \tag{9}$$

$$\Psi = \frac{1}{l}\sum_{i=1}^{l} x_i \tag{10}$$

Step 2: We calculate the overall average of the number of samples in the library using Eq. (11) and (12).

$$\mu_i = \frac{1}{n}\sum_{k=1}^{n} \mathrm{I}_{ik} \tag{11}$$

$$C = \frac{1}{l}\sum_{i=1}^{l} d_i d^T = \frac{1}{l} AA^T \tag{12}$$

Step 3: We obtain the covariance matrix *C* of *Y* using Eq. (13), where *C* is a 6 * 6 matrix.

$$C = \frac{1}{n}\mathrm{AA}^{\mathrm{T}} \tag{13}$$

Step 4: The eigenvalues and eigenvectors of *C* are obtained using Eq. (14) and (15),

$$\mathrm{U}^{\mathrm{T}} C \mathrm{U} = \Lambda \tag{14}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_6 \end{bmatrix}, \tag{15}$$

where the values in Λ (I = 1, 2, 3, 4, 5, 6) are the eigenvalues and $U = (\omega_1, \omega_2, \cdots, \omega_6)$ is the orthogonal matrix. The eigenvector corresponds to the eigenvalue with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \lambda_5 \geq \lambda_6$, spanning the feature space. To obtain the contribution rate of 99%, *j* eigenvalues are obtained eventually using Eq. (16):

$$\sum_{1}^{j} |\lambda_{\mathrm{i}}| \geq 0.99 \sum_{1}^{6} |\lambda_{\mathrm{i}}| . \tag{16}$$

Step 5: We project the original object sample set *X* into the feature space, where *j* eigenvalues correspond to *j* eigenvectors in the covariance matrix *C*. After applying PCA, the matrix that has the most of the features of *X* is denoted by Ω and computed using Eq. (17):

$$\Omega = \mathrm{W}^{\mathrm{T}}\left(\sum_{\mathrm{k}=1}^{\mathrm{n}} (\mathrm{X}_{\mathrm{k}} - u)\right), \tag{17}$$

where the dimension of X is $\mathrm{n} * 6$ and the dimension of Ω is $\mathrm{n} * j$ (0 <j <6) after dimension reduction.

Step 6: Suppose the test image of the dropped object collected in a random video be *X* '(the dimension if $1 * \mathrm{m}$), the Euclidean distance *D* can be obtained according to Eq. (18):

$$D = \sqrt{\sum_{c=1}^{n} \left\| \Omega' - \mathrm{N}_{\mathrm{c}} \right\|^2}, \tag{18}$$

where Ω' is obtained after dimension reduction according to Eq. (15), the row vector *Nc* (c = 1, 2, n) corresponding to each sample in the library is used to extract each row of Ω.

Step 7: Finding $D_{\min}$ from *D*. If $D_{\min} > D_f$ (a threshold), the image of an dropped item belongs to the category "other items". Otherwise, we find an approximate image of the abandoned object from the item library according to the matching rule based on the minimum distance classifier.

Because the recognition algorithm uses the geometric affine invariant moment as features, the front/side views of trunk, handbag and carton have the same feature quantity as the corresponding image is rotated in a certain direction. When only the front or side images of the trunk, handbag and carton are stored in the database, the algorithm can still correctly identify the luggage, handbags, and paper boxes with large affine transformations such as rotation, different lighting conditions, translation, zoom and tilting. In Figs. 3-5, the first column shows sample images in the database from the front view, whereas columns (a), (b), (c) are side-view screenshots with affine transformations in the test videos.

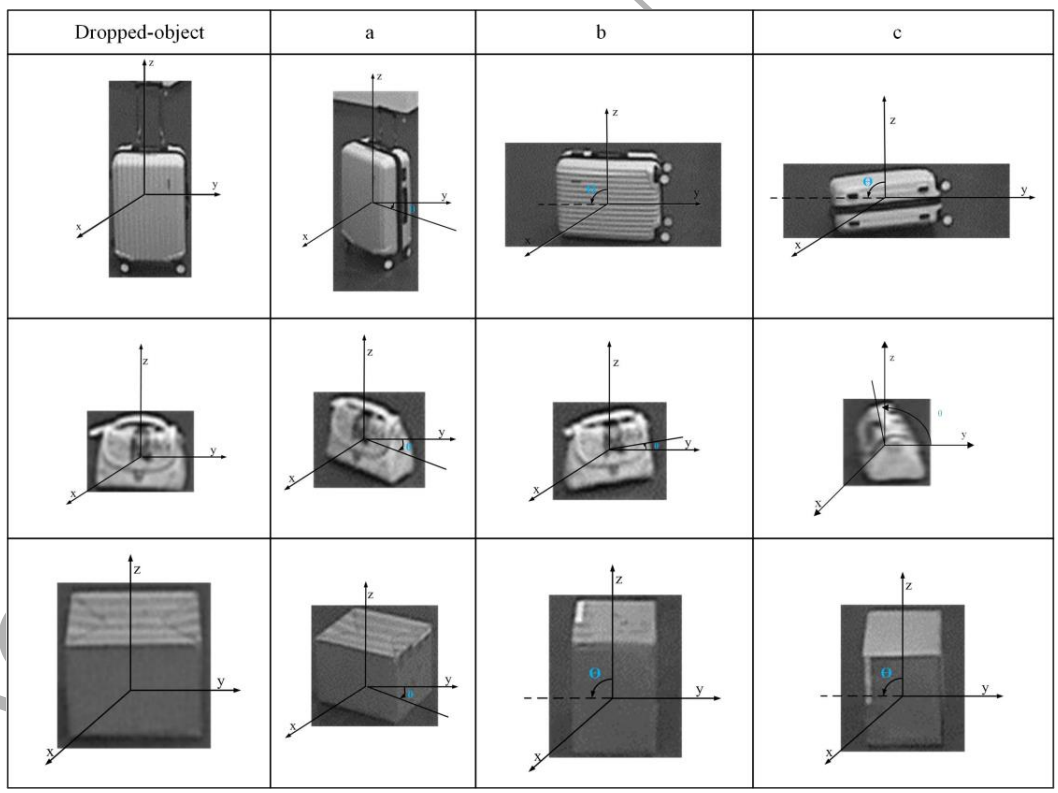


Fig. 3. Different items with affine transformation.

2.3 The Distributed Architecture

Our proposed method performs the dropped-object detection on a distributed architecture. The model is divided into the server, agent and monitoring device (e.g., webcam), where video acquisition, detection and recognition are separately implemented. As shown in Fig.1, the agent is connected to an IP-based webcam, receiving real-time video streaming data and deploying the dropped-object detection algorithms. At the same time, real-time video data and testing results are sent to the server. According to the user's request, the server sends the detection and recognition task to the agent, receives and stores the agent-transmitted dropped-item detection results into a database, and implements dropped-item recognition. While receiving the forwarded real-time video stream and storing it on the server's local disk as files, the recognition results are also stored in the database. The distributed model supports remote web-side access if the user enters the system through the browser. Moreover, in this system, the user can watch real-time streaming video, on-demand warnings, and task definition and query operations.

In this structure, we achieve the distributed model by using OpenCV and LIVE555 open source library. The agent and server were developed in the LIVE555 streaming media server with some modifications. We use the WSDL and SOAP protocols for the interaction of feature data, such as the transmission of abandoned-item detection results and dispatch of detection tasks, etc. The RTP protocol is used for real-time video streaming data transmission.

3. Experimental Results

We use OpenCV to build an experimental platform and implement all the experiments on Intel (R) Core (TM) i5-2400 3.10 GHz processor and 4G RAM computer. Axis 215 PTZ webcam is used to capture real-time video at 5 FPS. Three different datasets are used to test the performance of the proposed algorithms, including two publicly available dataset Video Surveillance Online Repository (VISOR) [47][48] and CAVIAR Test Case Scenarios[49], and our dataset. VISOR contains ten sequences with increasing complexity of a staged abandoned-object scenario at a classroom, where the corresponding annotations are given. The CAVIAR Test Case Scenarios were recorded for acting out different scenarios of interest and we select the "Leaving bags behind" scenario . Our dataset describes the pedestrians' activity of leaving objects both in indoor and outdoor environments, including 21 videos with single and multiplayer situation.

3.1 Experimental Results of Dropped-item Detection

In the following experimental results, the blue box indicates the tracking target (pedestrian) area T_n, the blue line indicates the trajectory of the tracking target (pedestrian), the green frame indicates the non-tracking target (item) area NT_n, and the red

box indicates a dropped item Ob_n. The "ownerindex" in the upper-right corner of the red box indicates the attributes of the legacy item Ob_n. If its attribute value and pedestrian are marked with the same number, it is indicated that the owner of the dropped item is the pedestrian, which means that the dropped object is associated with the pedestrian. If "ownerindex" is not shown at the upper right corner of the red box, it means that the host associated with the remaining item is not detected. The string "Dropped Object Number" indicates an alarm in the upper left corner of the figures and the red numbers behind the string indicate the number of detected dropped-objects.

Fig. 4 shows a video scene where a pedestrian left an item at a time. According to our algorithm, when the interregional distance $D \le 5$ or the ratio of R_c to R_1 is larger than 0.8, and the object satisfied with Eq. (5) and (6), the object is deemed as a dropped object. Frame 159 shows the tracking of Pedestrian 1 (area T_1); Frame 201 shows the moment when the pedestrian abandoned the item; Frames 229 and 251 indicate that after a lapse of time T, the dropped-object is detected. Simultaneously, the "ownerindex" of the legacy item and the pedestrian are marked with the same number 1, which means that the legacy item is associated with Pedestrian 1. The red number in the upper-left corner of Frame 251 indicates that the number of left-over items in the video is 1.

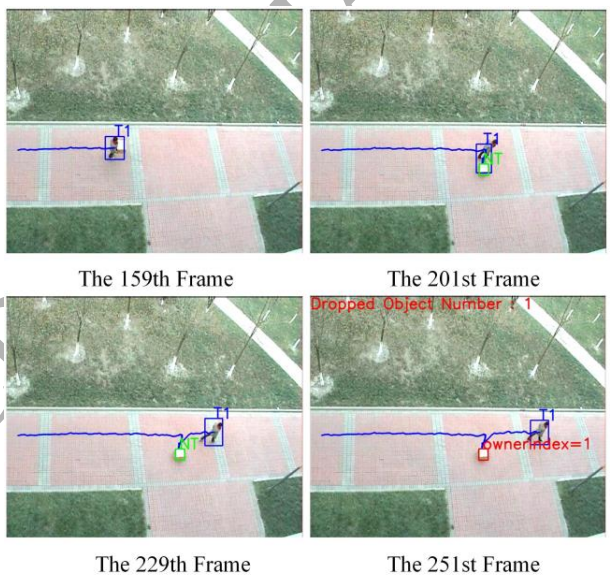
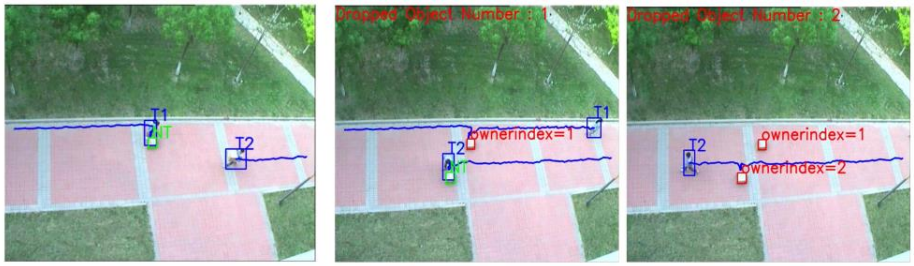


Fig. 4. A pedestrian left an item in the video scene.

Fig.5 shows that two pedestrians left two items at different times. In Frame 182, Pedestrian 1 (area T_1) and Pedestrian 2 (area T_2) are tracked separately. The 365th frame indicates that the left-over item of Pedestrian 1 has been detected, while Pedestrian 2 just abandoned an item at the same time. After computing the Euclidean distance between the dropped object and

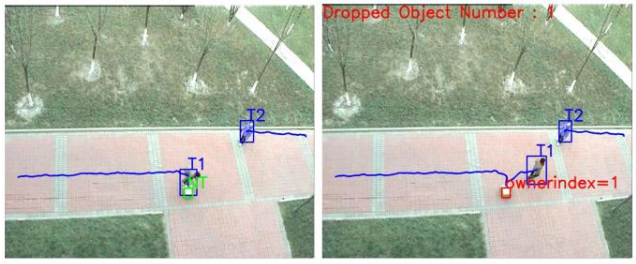
the owner, the detected dropped item is associated with Pedestrian 1. As shown in the figure, abandoned items of Pedestrian 1 and Pedestrian 2 are both detected, while the two remaining items are correctly associated with their respective masters. The 365th frame and the 462nd frame represent abandonment moments of the two pedestrians.



The 182nd Frame The 365th Frame The 462nd Frame

Fig. 5. Two pedestrians left two items at different times.

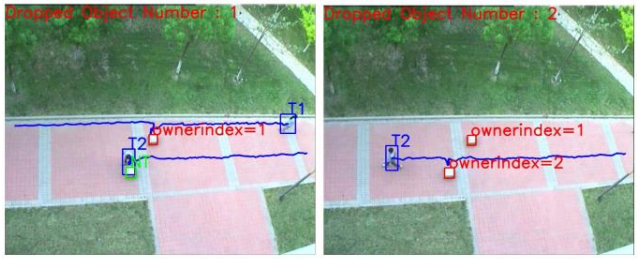
In Fig. 6, a pedestrian left an item at that moment, and another pedestrian remained stationary for a period of time. While the abandoned item from Pedestrians 1 is detected, the stationary Pedestrian 2 continues to be tracked without being detected as a stationary object.



The 192nd Frame The 219th Frame

Fig. 6. One pedestrian dropped one item and another pedestrian remained stationary for a period of time.

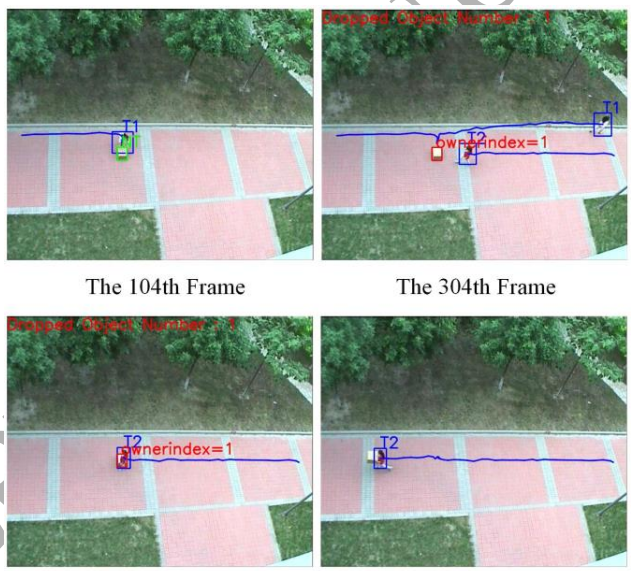
In Fig. 7, a pedestrian left an item at that moment, and another pedestrian in front of the pedestrian remained stationary at same time. At the moment when Pedestrians 1 abandoned the item, the circumscribed rectangular area T_2 of Pedestrian 2 partially coincides with the area T_1 of Pedestrian 1, but it also partly overlaps with the non-tracking objects outside the rectangular area *NT*. That is, it is still associated with Pedestrian 1 when the dropped item is detected.



The 269th Frame The 382nd Frame

Fig. 7. A pedestrian dropped an item at a time when another pedestrian remains stationary in front of the pedestrian.

The left-over item shown in Fig. 8 is with almost complete occlusion for a period of time, and then it is removed by a pedestrian. The object can still be detected and associated with Pedestrian 1 for a period of time when it is obscured by Pedestrian 2 who remains to stand still. The 412th frame shows that when the left object is removed by Pedestrian 2 from its original location, it is not further detected. At the same time, the alarm (the red words) is immediately reset.



The 104th Frame The 304th Frame

The 330th Frame The 412th Frame

Fig. 8. The dropped item is blocked for a period of time and then removed by a pedestrian.

We tested the performance of the proposed algorithm on the recorded 10 videos in 4 different scenes (4 groups of experiments). The statistical test results for each frame are summarized in Fig. 9, showing the detection rate of the dropped objects, the detection rate of the owner corresponding to the dropped objects in all the video frames of a scene, and the detection accuracy of the owner after finding the dropped objects.

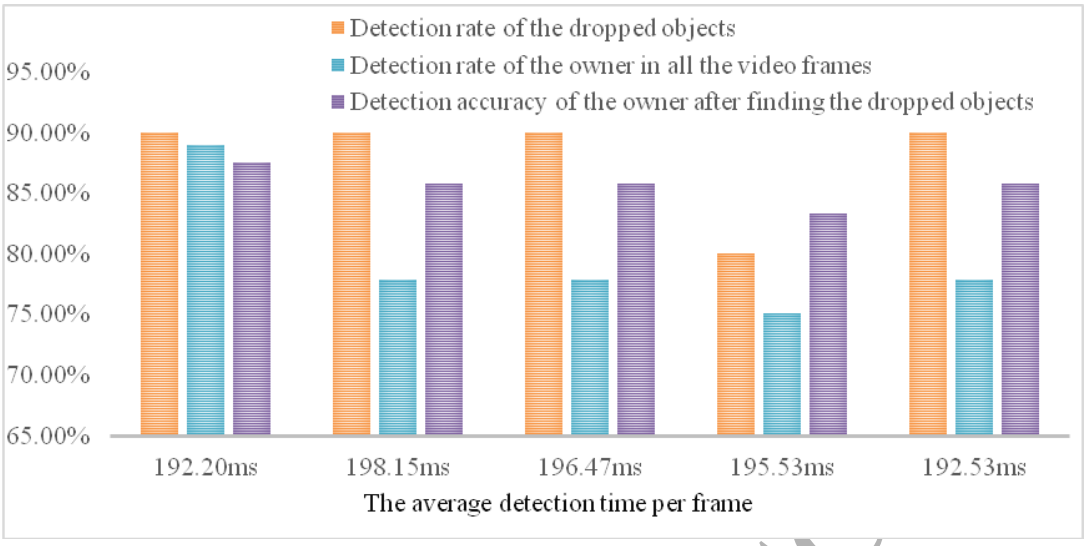


Fig. 9. The statistical results of the test data in different video scenes. The orange column shows the detection rate of dropped objects; the blue one shows the detection rate of the owner in all the video frames; the purple one shows the detection accuracy of the owner after finding the dropped objects.

As shown in Fig. 9, it is more accurate to use the new algorithm for detecting the presence of a legacy object and its owner in the video scenes. The average detection time per frame is 200ms (calculated from 1 frame / frame rate), which is less than the time difference between the frames of the video and the reference frame. Therefore, the proposed algorithm can quickly detect and analyze the relevance in the real-time video. In real-time automated video surveillance, the algorithm can provide a high detection rate of legacy items, which lays the foundation for later legacy-item identification and also guarantees the accuracy and availability of the whole algorithm.

To present the qualitative results, we evaluate our method in the VISOR and the CAVIAR test scenarios. In those two datasets, our system can accurately detect the abandoned object with no false negatives. Figure 10 and 11 show the results of abandoned object detection in public datasets. In Fig. 10, a person walking from one end of classroom to the other and dropping objects were successively detected as abandoned ones. Fig. 11 shows in a complicated indoor scene, one man is walking from a tunnel and leaving a bag, which is detected as an abandoned object. Furthermore, to demonstrate the superiority of our proposed algorithm, we provide the comparison of recall and precision with other dropped-object detection method [11], as shown in Table 2. Here, recall is the fraction of detected dropped objects among all the objects in the scene; precision is the fraction of detected dropped objects over the total amount of detected objects. It is obvious that our algorithm performs better than k-means used in [11] on recall and precision.

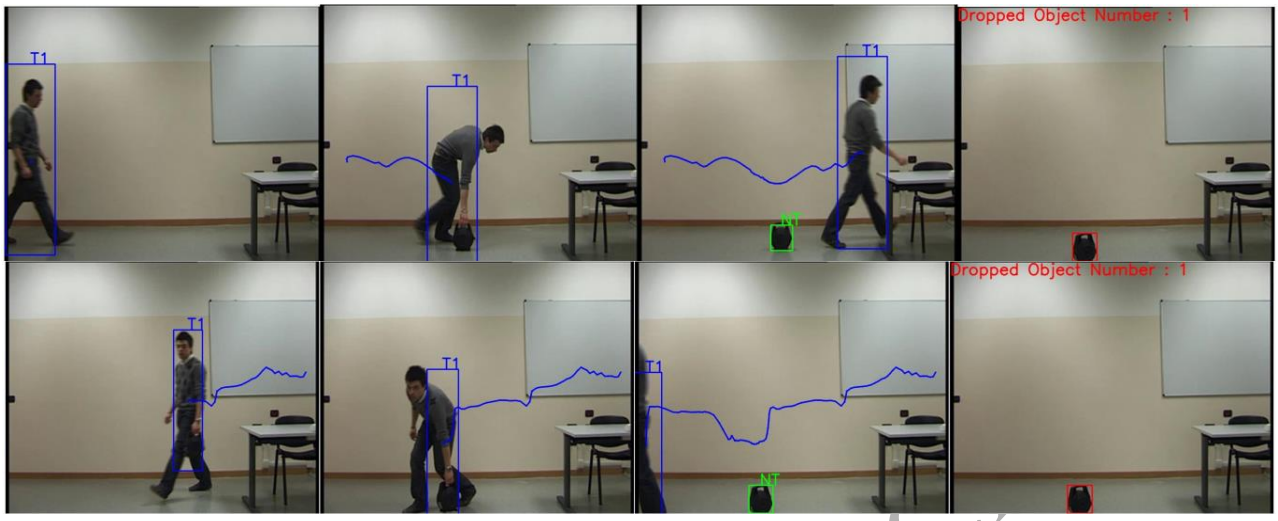


Fig. 10. Sample results from abandoned Object 1 and 2 from VISOR.

Fig. 11. Sample results from CAVIR.

Table 2. The comparison of recall and precision with [11].

Parameters	VISOR dataset		Our dataset	
	K-means [11]	Our algorithm	K-means	Our algorithm
Recall (%)	60	70	40	83.33
Precision (%)	50	80	40	100

Compared with other algorithm used VISOR [50] and CAVIR [51]，our algorithm not only could detect the abandoned object with high accuracy, but also can track the owner of the abandoned object and show the owner number on the screen.

3.2 Experimental Results of Dropped-object Recognition

We recorded 16 real-time videos with three different items for legacy-item identification in the same scene. We selected the front view and side view of each item, i.e., "trunk", "handbag" and "carton" (as shown in Fig. 12) as sample images. Then, we build geometric affine invariant moment and use PCA to reduce the dimension of the invariance meshing matrix to 5.

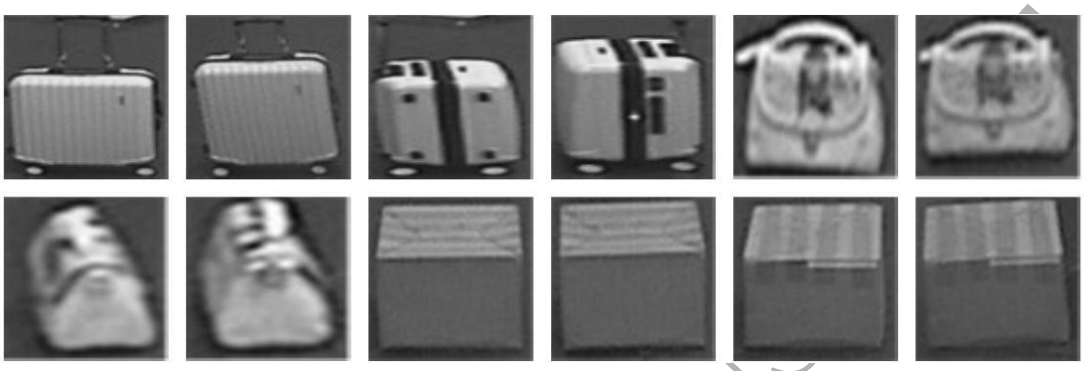


Fig. 12. Sample images after preprocessing in the item library.

In the figures of results, the green box indicates the tracking target (pedestrian) area and the red box indicates the detected item OB_n. If the number above the red box is marked as the same as that of the pedestrian, it indicates that the pedestrian is the owner of the remnant. The letters "lb", "hb" and "cb" above the red box respectively stand for "luggage box", "hand bag" and "cardboard box", which are the identification results of the legacy items. The "Dropped Object Number" in the upper-left corner of the figure indicates an alarm, the red numbers that followed indicate the number of items left in the video that were detected. There are 12 sample images for each category of the suitcase, handbag and carton in the library, where three sample training sets are constructed using the dropped-object recognition algorithm. The similarity is based on the Euclidean distance as defined in Eq. (19), where X is the n-dimensional sample matrix and Y is the matrix of the left item.

$$\text{Similarity} = 1 - \frac{\sqrt{\sum_{i=1}^{n} \left\| X_i - Y_i \right\|^2}}{\sqrt{\sum_{i=1}^{n} \left\| X_i \right\|^2}} \tag{19}$$

Experiment 1: (Trunk Experiment) We recorded two videos for experiments in which a pedestrian left a trunk. Frames 299 and 308 respectively show that the trunk is left in the video scene with front and side views. As shown in Fig. 13, the trunk is detected and correctly identified with a red box marked as "lb". The similarity and processing time are shown in the figure.

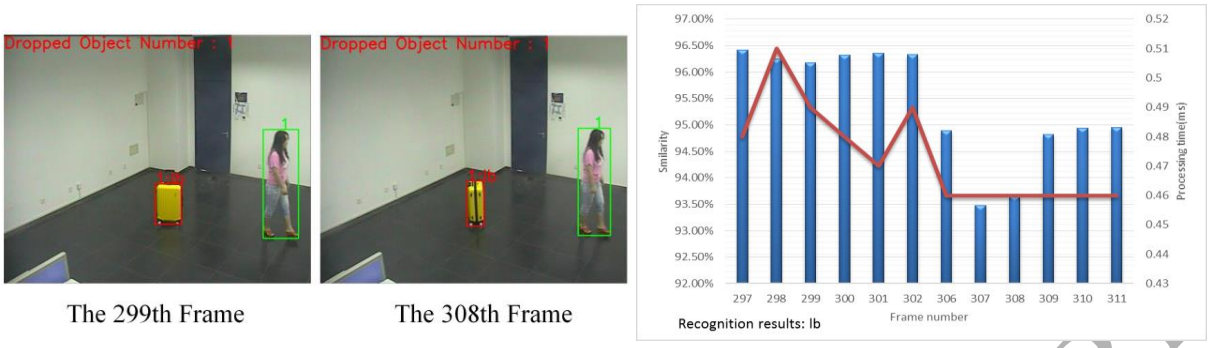


Fig. 13. A pedestrian left a trunk in the video scene and the recognition results of each frame of 6 consecutive images in two videos in Experiment 1

Experiment 2: (Handbag Experiment) We recorded two videos for experiments in which a pedestrian left with a handbag. In Fig. 14, Frames 270 and 282 respectively show that the handbag is left in the video scene with the front and side views. The handbag is detected and correctly identified with a red box marked as "hb". And it also shows the similarity and processing time.

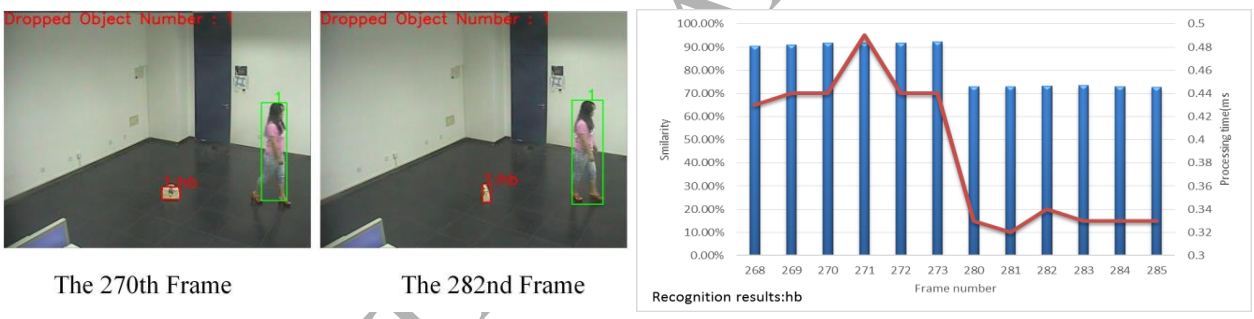


Fig. 14. A pedestrian left a handbag in the video scene and the recognition results of each frame of 6 consecutive images in two videos in Experiment 2

Experiment 3: (Carton Experiment) We recorded two videos for experiments in which a pedestrian left a carton in the video scene. In Fig. 15, Frames 323 and 347 indicate that the cartons are left in the video scene with the front and side views. Carton is detected and correctly identified with a red box labeled as "cb".

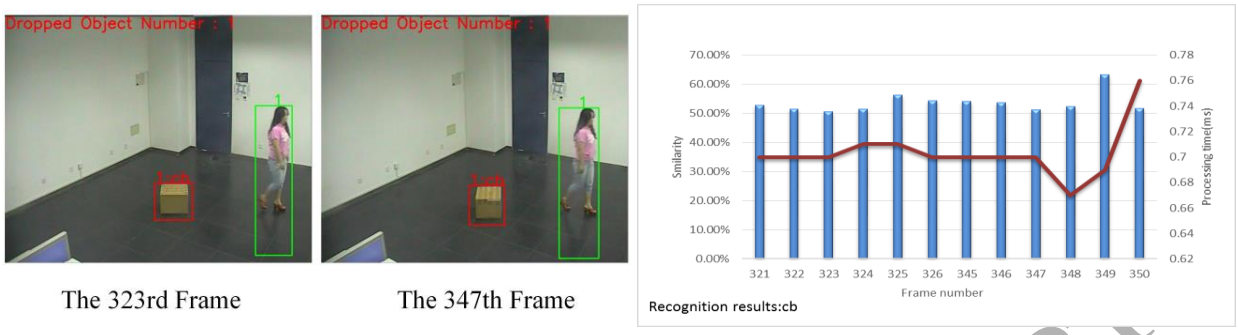


Fig. 15. A pedestrian left a carton of boxes in the video scene and the recognition results of each frame of 6 consecutive images in two videos in Experiment 3.

3.3 Experiments for the Dropped-object Identification under Different Perspectives

Because our dropped-object recognition algorithm uses geometric affine invariant moments as features which are robust to affine transformations such as translation, rotation and scaling, the feature quantity is not changed with transformations and there is little influence on the recognition results. In this subsection, our experiments test the anti-rotation properties of the algorithm.

Experiment 4 This experiment is conducted to confirm that the algorithm is able to deal with rotations of the luggage, handbags and cartons. In Fig. 16, a pedestrian left a trunk with a different perspective from the training trunks. We recorded four videos, where Frame 253, 247, 252 and 236 are sample images of luggage as the benchmark after being rotated and left in the video scene with different perspectives. As shown in Fig. 16, the luggage compartment is detected and correctly identified in the perspective of a certain angular transformation with a red box marked as "lb".

In Fig. 17, a pedestrian left a handbag with a different perspective from the training handbags. We recorded two videos, where Frames 232 and 184 show sample images of the handbag as the benchmark after being rotated and left in the video scene with different perspectives. These two images correspond to Figs. 5 (b) and (c). As shown in Fig. 17, the handbag is detected and correctly recognized in the perspective of a certain angular transformation with a red box marked as "hb".

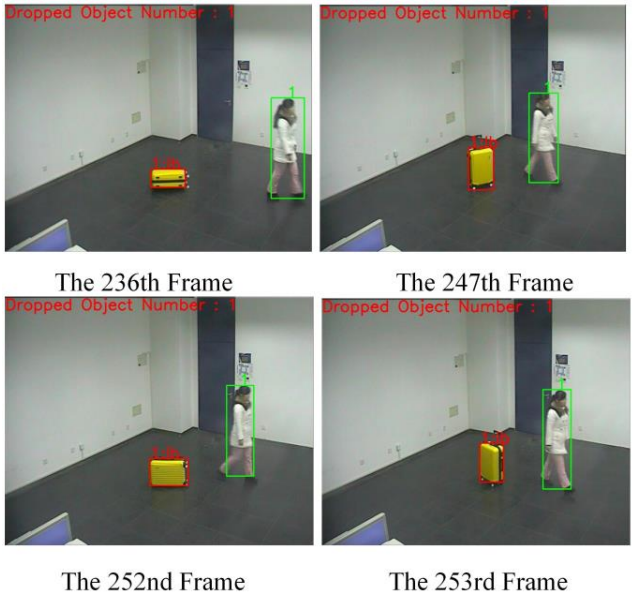


Fig. 16. A pedestrian left a trunk with a different perspective in the video scene.

Fig. 17. A pedestrian left a handbag with a different perspective in the video scene.

Fig. 18 shows a pedestrian leaving a carton box with a different perspective from the training carton boxes. We recorded four videos, where the 280th, 308th, 279th and 299th frames indicate that the sample images of carton as rotated through a different perspective. The carton is detected and correctly recognized at a certain angle of view of the angle conversion, which is labeled as "cb".

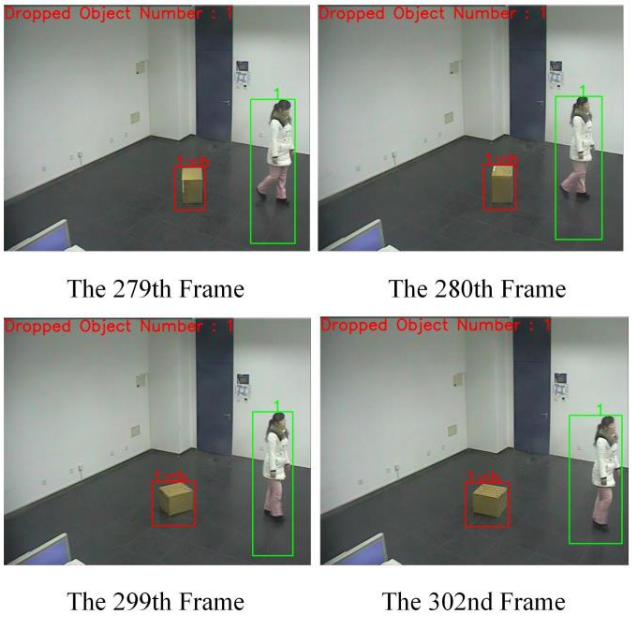


Fig.18. A pedestrian left a carton with a different perspective on the video scene.

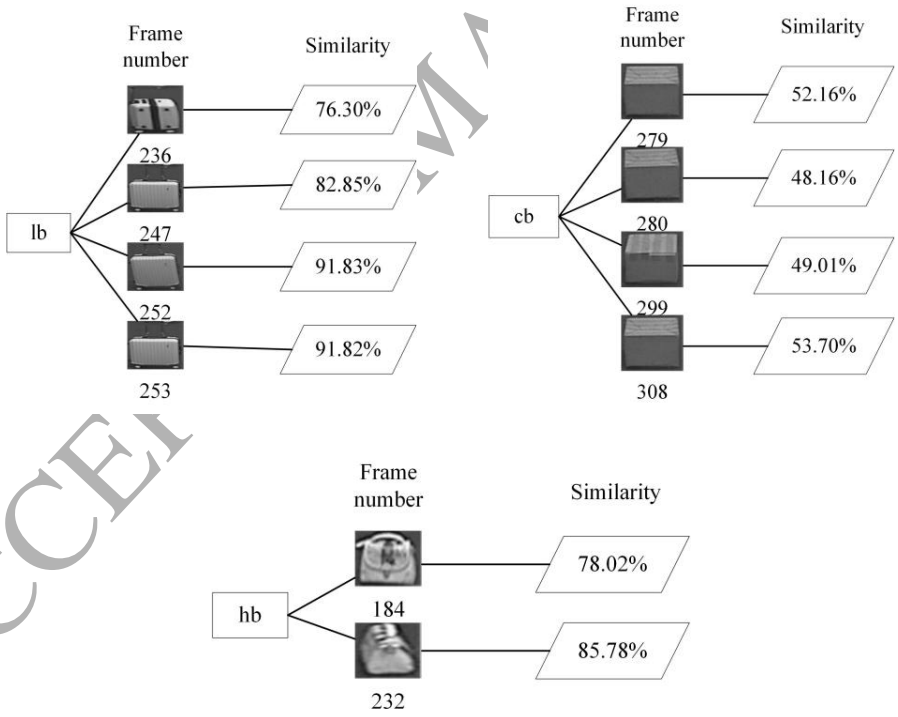


Fig. 19. The identification results (rates) of ten frames of the reference image.

From Fig. 19, the same recognition results are obtained corresponding to ten frames from Fig. 16, Fig. 17 and Fig. 18. In this figure, we can see that the front view of the object and its corresponding image have the same identification

characteristics as the images are rotated in a certain direction. When the item library only has the front or side sample images of the item, the algorithm still can correctly identify the objects from different perspectives.

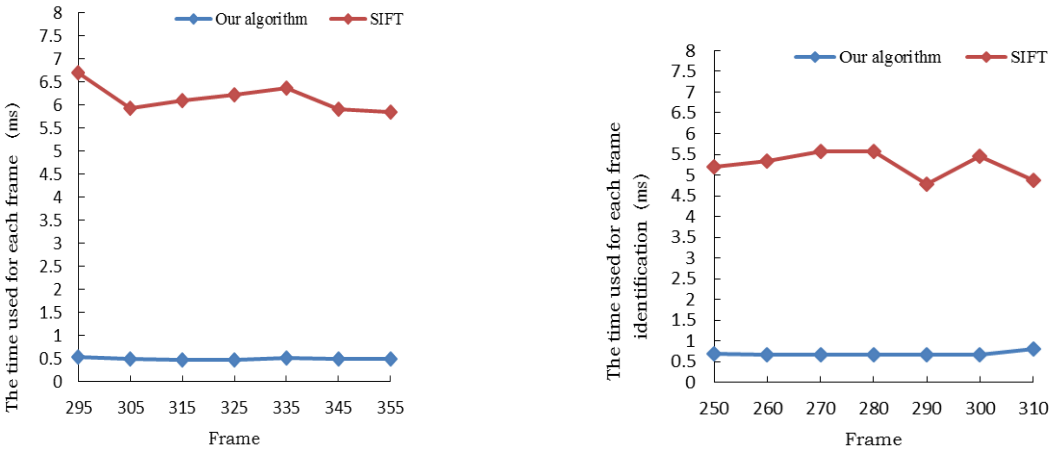


Fig. 20. The comparison of computation time between our algorithm and SIFT in object recognition

Fig. 20 shows the contrast between these two algorithms, it is clearly to find out that our algorithm uses less time than SIFT in object recognition, which are 10 times faster.

3.4 Performance Comparison of Centralized and Distributed Models

For a centralized architecture, the server directly receives live video streams sent by the webcam with the real-time video streams stored in the local disk. In our experiments, we made a computer as a server, used the algorithm proposed in this paper to simultaneously deal with different numbers of detection and recognition tasks in real-time video streams in this server, and then recorded the performance of the server.

For the distributed architecture, we used a local computer as a server for receiving the real-time video streams forwarded by the agent and storing the detection results in the local database. We set up four remote computers with four agents installed on each computer to simulate 16 agents in total. The detection algorithm of this paper is deployed in agents to carry out the dropped-objects detection in real-time video streams, with 2 live video streams for each agent.

Experiment 5 Fig. 21 shows a comparison of server CPU utilization when different numbers of video streaming tasks are processed at the same time in both centralized and distributed architectures. CPU utilization refers to the percentage of total CPU resources used in the execution of one or more video streaming tasks. According to the CPU utilization for each frame, the performance of dropped-object detection and recognition algorithms in the centralized and distributed models is analyzed

by calculating the average number of CPU usage in one cycle as an index. The horizontal axis in the figure indicates the number of tasks that the server processes at the same time and the vertical axis represents the CPU utilization rate. Checking the blue line in the figure for the performance of a centralized architecture, we can see that with the increase of the number of video streaming tasks, the computer's CPU utilization in the detection of four video streams at the same time has reached the peak of 100%. From the red line in the figure for the performance of a distributed architecture, as the number of video streaming tasks on the server increases, the CPU utilization of the server is in a slow upward trend. When the number of video streaming tasks reaches to 28, the CPU utilization increases very slowly; when the number of video streaming tasks reaches to 32, the CPU utilization of the server is only 37%.

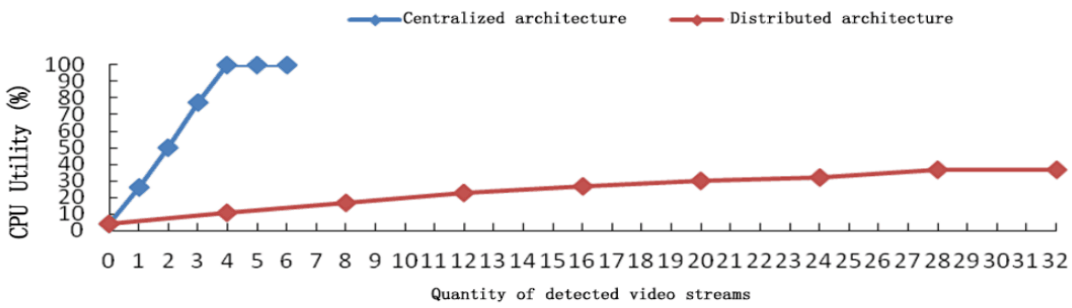


Fig. 21. Comparison of CPU utilization of the server for centralized and distributed architectures.

Experiment 6 Fig. 22 shows a comparison of the time spent for each task when a number of video streaming tasks are processed in the centralized and distributed architectures at the same time. The horizontal axis indicates the number of detected video stream tasks that the server is simultaneously processing, whereas the vertical axis represents the time taken for each frame to process each video stream. The blue line represents the elapsed time of the centralized architecture. As we can see from the graph, as the number of video streams increases, the time taken to process each video stream task increases. Time used here mainly refers to the completion of each detection and recognition task in the video stream of legacy items. Checking the red line for the time utilization of the distributed architecture, before the number of tasks in the video streams increases to 20, the processing time for each frame of the server in each task is basically unchanged, showing a moderate trend. When the number of video streams increases to 20, the processing time for each frame of the server is slowly increasing in each task. The time taken here means the total time of three progresses, which is each frame in the video streaming real-time video data is stored in the server's disk completed, each frame of kitsch detection results are stored in the database and server complete identification algorithm. The results of the detection of the dropped items per frame are stored in the database

and the server for the use in the recognition algorithm. The amount of time (less than 1 ms) for the server to complete the recognition algorithm is very little compared to the others, or even can be ignored. The green line shows the time difference between two sequential frames, i.e., 200ms because we choose 5fps as the video frame rate in our experiments. When the processing time of the algorithm on each frame is more than 200ms, it will delay the real-time detection of the next frame. As seen from the figure, the centralized model (blue Line) is processing four video streaming tasks while the time to deal with each video stream task per frame has exceeded 200ms, and therefore it will delay the detection of legacy items of the next frame. In such a case, real-time detection and identification of legacy items cannot be performed anymore. This may result in a program crash after a certain period of time has elapsed. The distributed model (red line) is in the simultaneous processing of 32 video streaming tasks, the processing time of the server to deal with each video stream task per frame is 100ms, far less than 200ms, which ensures real-time detection and identification of legacy items.

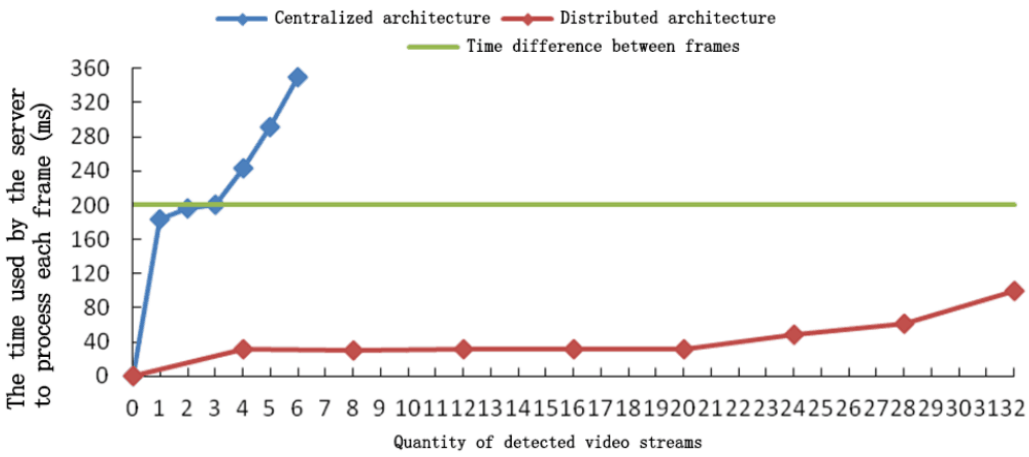


Fig. 22. The time taken by the server to process one frame per task by centralized and distributed architectures.

Experiment 7 Fig. 23 shows the time required for a centralized architecture to simultaneously detect and identify the legacy items in four video streams and to process the current frame. When the first 10 frames arrive, we need to wait at least 400ms until the current video frame data are being processed. Delaying will produce video stuttering, frame dropping, etc., resulting in the failure of detection and identification. Moreover, the delay time is continuously accumulated. When it is accumulated to a certain extent, a lot of video data will be automatically discarded. The algorithm cannot continue to run, leading to program collapse. In practice use, we should avoid this because excessive monitoring tasks could cause the server unable to allocate resources due to the delay situation.

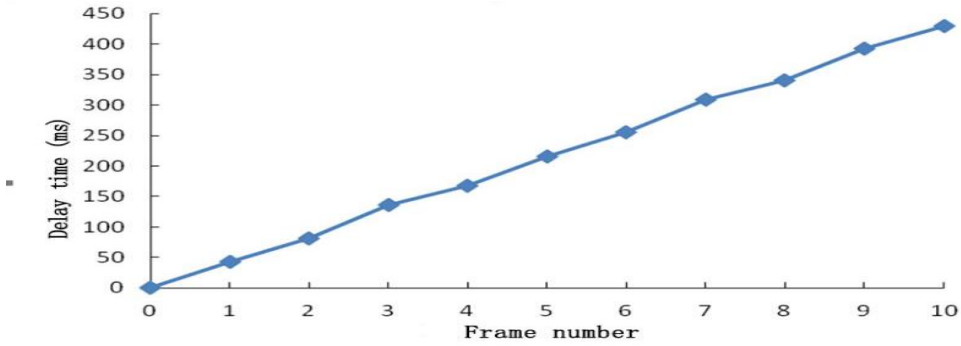


Fig. 23. The delay time trend of the centralized architecture in the server for processing video per frame.

Compared with the existing centralized processing methods, our distributed structure of dropped-object detection and identification method has the following advantages. It can effectively reduce server pressure, improve system performance, ensure real-time detection of multiple video streams, and be suit for large-scale real-time automatic video monitoring. It both improves the stability of the system operations and increases the scalability of the system.

4. Conclusions

We proposed a method for analyzing pedestrian activity based on dropped-object detection and recognition in video surveillance. The method consists of two parts: dropped-object detection and dropped-object recognition. A dropped-object detection algorithm based on regional information is proposed, which can detect dropped objects in complex environments and match the owner of dropped objects. An algorithm based on moment invariant and PCA is proposed to further recognize the dropped-objects, which can recognize objects viewed from different directions and locations from video cameras. In addition, in order to solve the limitation of the centralized video processing model for large-scale video streams in real time, the proposed method is designed and accomplished in a distributed architecture. The experimental results on representative databases showed that the proposed method can effectively and efficiently recognize the pedestrian activity of dropping objects in real-time video data. The recognition results from the proposed method may be applied to further analyzing human activities and intentions such as determining whether the dropped objects are intentional hazardous articles or unconsciously lost articles according to the appearance of the dropped objects. Compared with the existing dropped-object detection methods, our method has advantages of matching the owner of dropped objects, accurate and robust object recognition from different perspectives, capability of analyzing pedestrian activity to determine potentially dangerous dropping behavior and helping find the owner of lost properties. Experiment results also showed that our distributed model has better performance, stability and scalability than the traditional centralized method. Next challenges in this research field is applying GPU to the

task of human behavior analysis based on dropped-object detection and recognition with enhancement of distributed parallel hardware framework, comparing the processing stages in GPU and CPU, and involving the design of deep learning networks to encounter more complex scene, e.g., multiple pedestrians with dropped objects in crowded environments.

Acknowledgements

This research was supported by National Natural Science Foundation of China under Grants 61463032, 61762061, 61703198, and 61662044; Scientific Research Foundation for Returned Scholars, Ministry of Education of China ([2014] 1685); and the Natural Science Foundation of Jiangxi Province, China under Grant 20161ACB20004.

Reference

[1] Flusser, Jan, B. Zitova, and T. Suk. Moments and Moment Invariants in Pattern Recognition. Wiley Publishing, 2009.

[2] Zhan, C., Duan, X., Xu, S., Song, Z., & Luo, M. (2007). An Improved Moving Object Detection Algorithm Based on Frame Difference and Edge Detection. International Conference on Image and Graphics (pp.519-523). IEEE Computer Society.

[3] Yao, Jian, and J. Odobez. "Multi-Layer Background Subtraction Based on Color and Texture." (2007):1-8.

[4] Lopez-Mendez, Adolfo, F. Monay, and J. M. Odobez. "Exploiting scene cues for dropped object detection." International Conference on Computer Vision Theory and Applications IEEE, 2014:14-21.

[5] Singh, A., Sawan, S., Hanmandlu, M., Madasu, V. K., & Lovell, B. C. (2009). An Abandoned Object Detection System Based on Dual Background Segmentation. IEEE International Conference on Advanced Video and Signal Based Surveillance (pp.352-357). IEEE.

[6] Bhargava, M., Chen, C. C., Ryoo, M. S., & Aggarwal, J. K. (2007). Detection of abandoned objects in crowded environments. Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on (pp.271-276). IEEE.

[7] Bhargava, M., Chen, C. C., Ryoo, M. S., & Aggarwal, J. K. (2009). Detection of object abandonment using temporal logic. Machine Vision & Applications, 20(5), 271-281.

[8] Fan Quanfu, Pankanti S. ssss [C]. IEEE International Conference on Advanced Video and Signal-Based Surveillance '9. IEEE Press, 2012: 58-63.

[9] Tian, Y. L., Feris, R. S., Liu, H., Hampapur, A., & Sun, M. T. (2011). Robust detection of abandoned and removed objects in complex surveillance videos. IEEE Transactions on Systems Man & Cybernetics Part C, 41(5), 565-576.

[10] Lin, K., Chen, S. C., Chen, C. S., Lin, D. T., & Hung, Y. P. (2015). Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance. IEEE Transactions on Information Forensics & Security, 10(7), 1359-1370.

[11] Jayasuganthi, P., Jeyaprabha, V., Kumar, P. M. A., & Vaidehi, V. (2014). Detection of dropped non protruding objects in video surveillance using clustered data stream. International Conference on Recent Trends in Information Technology (pp.371-375). IEEE.

[12] Liu, Y., Zhang, J., Tjondronegoro, D., Geva, S., & Li, Z. (2009). An improved image segmentation algorithm for salient object detection. Image and Vision Computing New Zealand, 2008. Ivcnz 2008. International Conference (pp.1-6). IEEE.

[13] Caterina, Gaetano Di, and J. J. Soraghan. "An abandoned and removed object detection algorithm in a reactive smart surveillance system." Digital Signal Processing (DSP), 2011 17th International Conference on IEEE, 2011:1-6.

[14] Sacchi, C, and C. S. Regazzoni. "A distributed surveillance system for detection of abandoned objects in unmanned railway environments." IEEE Transactions on Vehicular Technology 49.5(2000):2013-2026.

[15] Yoshinaga, S., Shimada, A., Nagahara, H., & Taniguchi, R. I. (2014). Object detection based on spatiotemporal background models. Computer Vision & Image Understanding, 122(5), 84-91.

[16] Timofte, Radu, J. Kwon, and L. V. Gool. "PICASO: PIxel correspondences and SOft match selection for real-time tracking." Computer Vision & Image Understanding 153(2016):151-162.

[17] Wang, Q., Yuan, Y., Yan, P., & Li, X. (2013). Saliency detection by multiple-instance learning. IEEE Transactions on Cybernetics, 43(2), 660-672.

[18] Wang, Q., Fang, J., & Yuan, Y. (2014). Multi-cue based tracking. Neurocomputing, 131(7), 227-236.

[19] Wang, Q., Yan, P., Yuan, Y., & Li, X. (2013). Multi-spectral saliency detection. Pattern Recognition Letters, 34(1), 34-41.

[20] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 38(1), 142.

[21] Zhu, Y., Urtasun, R., Salakhutdinov, R., & Fidler, S. (2015). Segdeepm: exploiting segmentation and context in deep neural networks for object detection. , 84(84), 4703-4711.

[22] Zhang, Y., Sohn, K., Villegas, R., & Pan, G. (2015). Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction. Computer Vision and Pattern Recognition (Vol.8, pp.249-258). IEEE.

[23] Models, P., however, Ours, C. T., & Way, O. I. T. (2013). Scalable object detection using deep neural networks. 2155-2162.

[24] Zagoruyko S, Lerer A, Lin T Y, et al. A MultiPath Network for Object Detection[J]. 2016.

[25] Jiang H, Wang J, Yuan Z. Salient Object Detection: A Discriminative Regional Feature Integration Approach[C]// Computer Vision and Pattern Recognition. IEEE, 2013:2083-2090.

[26] Dai J, Li Y, He K and Sun Jian. R-FCN: Object Detection via Region-based Fully Convolutional Networks[J]. 2016.

[27] Song, Shuran, and J. Xiao. "Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images." Computer Science 139.2(2016):808-816.

[28] Zhang, Y., Sohn, K., Villegas, R., & Pan, G. (2015). Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction. Computer Vision and Pattern Recognition (Vol.8, pp.249-258). IEEE.

[29] Hu, Ming Kuei. "Hu, M.K.: Visual Pattern Recognition by Moment Invariants. IRE Transaction of Information Theory IT-8." Information Theory Ire Transactions on 8.2(1962):179-187.

[30] Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints." International Journal of Computer Vision 60.2(2004):91-110.

[31] Flusser, Jan, and T. Suk. "Suk, T.: Pattern recognition by affine moment invariant. Pattern Recognit. 26(1), 167-174." Pattern Recognition 26.1(1993):167-174.

[32] Liu, J., Li, D., Tao, W., & Yan, L. (2007). An automatic method for generating affine moment invariants. Pattern Recognition Letters, 28(16), 2295-2304.

[33] Ke, Yan, and R. Sukthankar. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on IEEE, 2004:II-506-II-513 Vol.2.

[34] Delponte, Elisabetta, et al. "SVD-matching using SIFT features." Graphical Models 68.5–6(2006):415-431.

[35] Lee, Taehee, and S. Soatto. "Video-based Descriptors for Object Recognition ☆." Image & Vision Computing 29.10(2011):639-652.

[36] Molina-Giraldo, S., Carvajal-González, J., Álvarez-Meza, A. M., & Castellanos-Domínguez, G. (2015). Video segmentation framework based on multi-kernel representations and feature relevance analysis for object classification.

[37] Juang, Chia Feng, W. K. Sun, and G. C. Chen. "Object Detection by Color Histogram-based Fuzzy Classifier with

Support Vector Learning." Neurocomputing 72.10-12(2009):2464-2476.

[38] Sheikh, Y, and M. Shah. "Bayesian Modeling of Dynamic Scenes for Object Detection." IEEE Transactions on Pattern Analysis & Machine Intelligence 27.11(2005):1778-1792.

[39] Canny, J., A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698..

[40] Xie, Xiaohua, L. Yang, and W. S. Zheng. "Learning Object-specific DAGs for Multi-label Material Recognition." Computer Vision & Image Understanding 143(2016):183-190.

[41] Ramík, Dominik Maximilián, et al. "A machine Learning based Intelligent Vision System for Autonomous Object Detection and Recognition." Applied Intelligence 40.2(2014):358-375.

[42] Fujiyoshi, Hironobu. "Object Recognition with Depth Image by Machine Learning." (2015).

[43] Nigatu, Hassen. "Machine Learning. Real Time Object Detection Algorithm. Adaboost, Integral Image, Cascading." Machine learning algorithm implementation 2015.

[44] Ba, Jimmy, V. Mnih, and K. Kavukcuoglu. "Multiple Object Recognition with Visual Attention." Computer Science (2015).

[45] Jarrett, K., Kavukcuoglu, K., Ranzato, M., & Lecun, Y. (2010). What is the best multi-stage architecture for object recognition?. IEEE, International Conference on Computer Vision (Vol.30, pp.2146 - 2153). IEEE.

[46] Yang, H., Zhou, J. T., Zhang, Y., Gao, B. B., Wu, J., & Cai, J. (2016). Exploit bounding box annotations for multi-label object recognition. Computer Science.

[47] R. Vezzani, R. Cucchiara,"Video Surveillance Online Repository (ViSOR): an integrated framework" in Multimedia Tools and Applications, vol. 50, n. 2, Kluwer Academic Press, pp. 359-380, 2010

[48] Video Surveillance Online Repository, http://www.openvisor.org.

[49] EC Funded CAVIAR project/IST 2001 37540,Website, 2007, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[50] jayasuganthi, P., Jeyaprabha, V., Kumar, P. M. A., & Vaidehi, V. (2014). Detection of dropped non protruding objects in video surveillance using clustered data stream. International Conference on Recent Trends in Information Technology (pp.371-375). IEEE.

[51] Muchtar, K., Lin, C. Y., & Yeh, C. H. (2014). Grabcut-based abandoned object detection. IEEE, International Workshop on Multimedia Signal Processing (pp.1-6). IEEE.

Weidong Min obtained his BE, ME and PhD of computer application at Tsinghua University in China in 1989, 1991 and 1995, respectively, on the research subjects of computer graphics, image processing and computer aided geometric design. He was an assistant professor of Tsinghua University from 1994 to 1995. From 1995 to 1997 he was a postdoctoral researcher at University of Alberta, Canada. From 1998 to 2014 he worked as a senior researcher and senior project manager at Corel and other companies in Canada. In recent years, he cooperated with School of Computer Science & Software Engineering, Tianjin Polytechnic University, China. From 2015 he is a professor at School of Information Engineering, Nanchang University, China. He is a Member of "The Recruitment Program of Global Expert" of Chinese government. He is an executive director of China Society of Image and Graphics. His current research interests include computer graphics, image and video processing, distributed system, software engineering and network management.

Yu Zhang obtained his ME of information science at Nanchang University in China in 2013. He is a PhD at Nanchang University in China now, on the research subject of abnormal behavior detection in video surveillance.

Jing Li obtained her PhD degree in Electronic and Electrical Engineering from the University of Sheffield, UK, in 2012. Before joining Nanchang University as an Associate Professor, she was a Research Associate at the University of Sheffield. Her research interests include content-based image retrieval, object recognition, visual tracking and scene understanding in complex environments. She has authored or coauthored in various journals, such as IEEE Transactions on Industrial Informatics, Information Sciences (Elsevier), etc.

Shaoping Xu received the M.S. degree in Computer Application from the China University of Geosciences, Wuhan, China, in 2004 and the Ph.D. degree in Mechatronics Engineering from the University of Nanchang, Nanchang, China, in 2010. He is currently a Professor in the Department of Computer Science, School of Information Engineering, at the Nanchang University, Nanchang. Dr. Xu has published more than 30 research articles and serves as a reviewer for several journals including IEEE Transactions on Instrumentation and Measurement. His current research interests include digital image processing and analysis, computer graphics, virtual reality, surgery simulation, etc.