Available online: November 09, 2018

Commun.Fac.Sci.Univ.Ank.Series A2-A3
Volume 60, Number 2, Pages 63-82 (2018)
DOI: 10.1501/commua1-2_0000000116
ISSN 1303-6009 E-ISSN 2618-6462
http://communications.science.ankara.edu.tr/index.php?series=A2A3

COMMUNICATIONS
SERIES A2-A3

MOVING OBJECT DETECTION AND CLASSIFICATION IN SURVEILLANCE SYSTEMS USING MOVING CAMERAS

OZGE MERCANOGLU SINCAN, HACER YALIM KELES
and SULEYMAN TOSUN

ABSTRACT. In this paper, we present a novel method to detect and classify moving objects from surveillance videos that are obtained from a moving camera. In our method, we first estimate the camera motion by interpreting the movement of interest points in the scene. Then, we eliminate the camera motion and find candidate regions that belong to the moving objects. Considering these regions as priors, we apply an efficient segmentation algorithm to obtain accurate object boundaries for the moving objects. Finally, we classify the detected objects as people, vehicle, or others using some morphological features and the velocity vectors of moving objects. The evaluation of the proposed approach on our surveillance dataset shows that our approach is very effective for determining the classes of moving objects in a moving camera setting.

1. INTRODUCTION

Automatic surveillance analysis is an active research area because of the increased security demand and large datasets [1,2, 3]. Today, it is not feasible to track all the recorded data by human observers. Moving object detection, tracking, and classification without human intervention are the main problems to be solved in this domain.

Continuous surveillance video recording takes place in many places in modern societies such as airports, train and subway stations, hospitals, motorways, and highways. Moreover, country borders and military bases are security critical areas where surveillance is used widely. In most of these systems, e.g., border surveillance, a mounted camera traverses a predefined angular range while recording data. Although there are many works in the literature that tracks moving objects from the video frames recorded by stationary [4,5] or moving cameras [6-10], to the

Received by the editors: June 12, 2018; Accepted: October 17, 2018.
Key word and phrases: Moving object detection, classification, moving camera, surveillance system.

best of our knowledge there has been no prior work that focuses on both detection and classification of moving objects from the videos acquired by moving cameras in video surveillance systems.

In this study, we aim to classify objects in a border surveillance domain. In this domain the camera moves continually to scan the neighborhood using a large field-of-view (FoV). Due to this large FoV, the projected images usually contain objects with high variance in their appearance and their apparent velocity. Moreover, the outdoor environment, e.g. varying weather conditions and occlusions, make the problem more challenging. All these factors negatively affect the detection of moving objects borders accurately; hence influence the classification accuracy severely.

In our preliminary work [11], we proposed a moving object detection scheme for surveillance videos obtained from a moving camera. In this study, we expand our previous method (1) by improving the object segmentation and (2) by adding a classifier to the system so that the detected objects are labeled as people, vehicle, or others. We observed that our improved object segmentation greatly helped to obtain more accurate object boundaries than previous version, which also increased classification accuracy. In classification, we designed a decision tree based classifier that uses some morphological properties of the segmented objects and the velocities of the parts as primary features.

In order to evaluate our object detection and classification system, we generated several videos and created a new dataset, namely the Golbasi Surveillance Dataset (GSD), which can be used as a benchmark in the related researches[1]. Our experiments show that the proposed method achieves very promising results.

We organized the rest of the paper as follows: In Section 2, we review the related work. In Section 3, we present the details of the proposed approach for moving object detection and classification. In Section 4, we provide the performance of our approach with a detailed discussion on experiment results that are obtained using GSD benchmark. Finally, we conclude the paper in Section 5.

2. Related Work

There is a great deal of research going on for classifying objects in surveillance videos. Most of them [12-17] aim to classify objects as people and vehicles using a still camera since distinguishing people and vehicles is important in city surveillance. These previous studies use different types of features to classify objects; such as morphological characteristics, recurrent motion, and histogram of gradients. They choose the features depending on the application domain and the object categories.

[1] http://comp.eng.ankara.edu.tr/golbasi-dataset/

Bo and Heqin [12] aim to improve the performance of urban traffic monitoring system. They assume that there are only two types of objects; people or vehicles. Their method is computationally inexpensive. As a classifier, they create a decision tree using aspect ratio, compactness, and velocity features. In Ref. [13], the authors use background subtraction method to obtain moving objects. Then, they use the size, velocity, location, and difference of histogram of oriented gradients (DHoG) of objects to classify them as human or vehicle. DHoG is the difference between HoGs obtained in consecutive frames and it measures intra-object deformation. In Ref. [17], the authors use an adaptive background subtraction and foreground segmentation technique to obtain moving objects. Then, they remove shadow to get accurate detection. They use aspect ratio, affine moment-invariants, and vertical-horizontal projective histograms as features to classify objects as human or car.

Javed and Shah [15] classify objects as single person, group of people, or vehicle based on their motion characteristics. They use recurrent motion image (RMI) to calculate repeated motion of objects. If average recurrence value is greater than a threshold, then the object can be a single person or a group of people. People and vehicles are distinguished by the help of recurrence value in the middle and bottom sections of RMI. Then, single person and group of people are distinguished by two different strategies: (1) If people are not very close to each other, shape cues are used to count the number of heads. (2) If people are very close to each other, normalized area of recurrence in the top section of RMI is checked.

Senior et al. [16] proposed a method for tracking objects, even when they are occluded, using appearance models. Background subtraction method is applied to extract foreground objects. A correspondence matrix is constructed whose rows are existing tracks and columns are foreground objects. Using this matrix, each foreground object is categorized as an existing object, a new object, merge, or split. Finally, each object is classified as a single person, a group of people, a vehicle, or other. A simple rule-based classifier is designed considering the area, the length and orientation of the principal axes, and dispersedness of the objects.

Elhoseiny et al. [14] proposed an object classification system for surveillance videos. Objects are classified into five classes: human, car, vehicle, object, and bicycle. Gurwicz et al. [18] aims to classify five types of objects: human, body organs, bag, group of people, and clutter. Both studies investigate different types of features such as luminance symmetry, cumulants, horizontal-vertical projection, morphological features, and 2D moment-based features. They both apply a feature-selection procedure to eliminate redundant features. Both research report that the highest classification accuracy is obtained with the geometric features. In Ref. [14], the experiments performed with VIRAT [19] dataset show that HoG feature does not perform well for surveillance videos. The reason is the low resolution of the detected objects. Although both support vector machine (SVM) and Adaboost classification techniques perform well, the performance of the Adaboost classifier is better than SVM in Ref. [14]. On the other hand, the highest classification accuracy is achieved by SVM classifier in Ref. [18].

Martín and Martínez [20] evaluate the state of the art people detection approaches in video surveillance. They group the approaches into two main categories according to appearance or motion information. They observe that the motion information is not adequate by itself to obtain good results for people detection. The authors selected eight different people detection approaches. First, they evaluate the appearance based approaches. Then, they extend the appearance based approaches by adding motion information. It is shown that combining appearance and motion information improves the results in all the cases.

All the aforementioned works use a stationary camera setting for object detection and classification. There are some new studies that use moving camera for traffic surveillance. Hua et al. [21] aim to detect pedestrians for assisting drivers to avoid vehicle-pedestrian accidents. Their purpose is to develop a warning system that detects people while the car is in motion. Their data is obtained from a moving camera as well. They find interest points using Lucas-Kanade algorithm [22] and estimate camera motion by the structure from motion (SfM) algorithm [23]. They use the spatio-temporal histogram of oriented gradient (STHoG), which includes pedestrian appearance and motion features to discriminate the pedestrian from background. Prioletti et al. [24] also proposed pedestrian detection system for driver assistance. They extract possible pedestrian candidates using the Haar cascade classifier. Then, they validate the candidates through part-based HoG classifier. Jegham and Khalifa [25] aim to detect pedestrians in poor weather conditions using a moving vehicle. Detecting pedestrians in a moving camera is a challenging problem. It provides a different field of view and object appearance/movement characteristics when it is compared to our problem domain. Hence, the methods presented in [21, 24, 25] are not directly applicable in our application domain due to mainly two reasons: (1) Objects are very small in our setting in which appearance based features fail to identify humans. (2) FoV is large, hence the regional coverage of the images in our domain are wider than the domains in [21,24]. Therefore, we need to utilize the coherency between consecutive frames for efficiency reasons. In our setting, small changes in the camera view direction influence the velocity vectors significantly; hence, camera needs to move slowly, rather than abruptly as may be the case in a car.

Meanwhile, some recent studies use deep learning methods for moving object detection in the presence of moving cameras [26].Babaee et al. [27] proposed a novel background subtraction method using Convolutional Neural Networks (CNNs).They evaluate their method on several datasets which includes different categories such as shadow, dynamic background, PTZ etc. For our problem, only PTZ category is suitable because it includes camera movement. However, in PTZ category the proposed CNN method gives poor results since PTZ category consists of insufficient data for training. Rozantsev et al. [28] detect flying unmanned aerial vehicles (UAVs) and aircrafts using a camera which is mounted on a drone or aircraft. Since there is no dataset available for detecting flying objects, they build two new datasets (UAV and aircraft) each including 20 videos. They propose a CNN based approach and compare it with relevant state-of-the-art techniques. They achieve about 15 percent increase on the average precision for detection.Deep learning is a promising research area and it has been used for detecting objects in various problems. However, large number of data requirement of these approaches makes them impractical to use in our research problem due to lack of insufficient labeled data.

3. The Proposed Scheme

The proposed framework is composed of four primary parts: camera motion estimation, moving object detection, improving detected objects boundaries, and classification of the detected objects as people, vehicle, or others. We show the overview of our framework in Figure 1. In the following subsections, we explain the design and implementation details for each component.

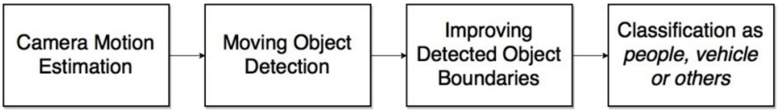

Camera Motion Estimation
Moving Object Detection
Improving Detected Object Boundaries
Classification as *people, vehicle or others*

FIGURE 1. Overview of the proposed scheme.

3.1. CAMERA MOTION ESTIMATION

In a video that is captured by a moving camera, the coordinates of all the pixels that belong to the background change in two consecutive frames. Therefore, moving objects cannot be obtained by a simple background subtraction method since the background is not fixed for two frames. In order to detect moving objects, we first need to estimate the camera motion and eliminate it from the frame so that we can determine the background pixels. Then, we can take the difference of two frames in order to detect moving objects.

In our camera motion estimation problem, we have two realistic assumptions for border surveillance: (1) We assume that the camera motion is slow and continuous, i.e. there is a spatial coherence between consecutive frames. (2) Moving objects occupy a small percentage of the whole scene. In another words, the FoV of the camera is large in the case of border surveillance systems. For videos conforming to these assumptions, we calculate the camera motion in three main steps: (1) interest point detection, (2) optical flow computation, and (3) camera motion vector calculation.

3.1.1. Interest point detection

We detect the interest points in a frame using Shi-Tomasi algorithm [29].Interest points are the points that have good contextual properties for tracking. Since the interest point detection is computationally costly, we extract interest points once in every 30 frames. In other words, we track the same interest points for 30 frames and then we extract new interest points to replace the old ones.

3.1.2. Optical flow computation

In the second step, we find the new coordinates of the detected interest points in the next frame using pyramidal Lucas-Kanade method [22, 30]. We then compute the motion vectors for all interest point pairs.

3.1.3. Camera motion vector calculation

Within a frame, the motion vectors that belong to the background pixels will be similar, since the apparent motion at these points occurs solely as a result of the camera movement. On the other hand, the motion vectors of moving objects appear as a combination of their own motion and the camera motion. Moreover, since moving objects will occupy a small percentage of the whole scene, the motion vectors of the majority of the interest points will be belonging to the background. We use this information to compute the camera motion. Therefore, when we estimate the background motion, we essentially obtain the camera motion, which is in the opposite direction.

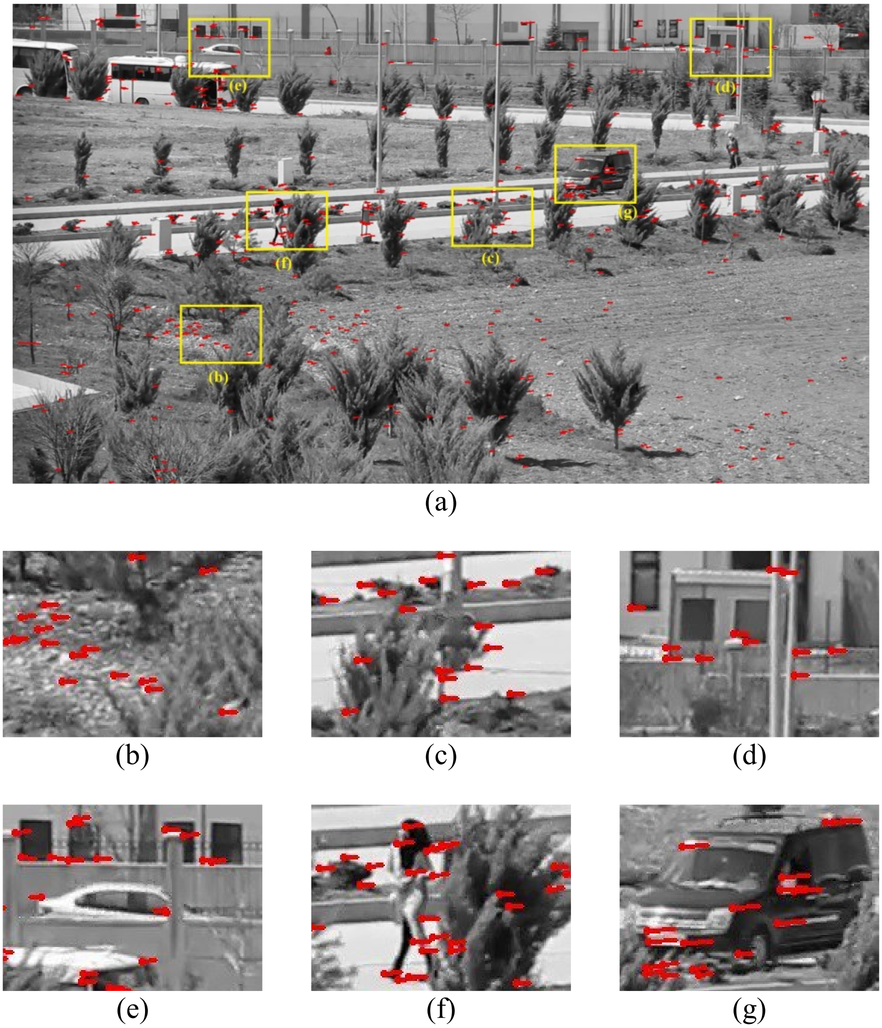
After detecting the motion vectors of all interest points in the previous step, we extract a histogram of motion vector magnitudes that will be used in background motion estimation. In this histogram, the majority of the similar values represent the common motion flow, which we identify as the background motion. Since the flow vectors of the moving objects will not be in this majority vector group, they will automatically be eliminated. Note that all these assumptions are attained after careful observations of our benchmark surveillance dataset.

In Figure 2, we give an example frame that shows the set of detected interest points together with their motion vectors. In Figure 2b-d, the depicted subregions belong to the background objects. The motion vectors in these example frames are very similar to each other in their direction and magnitude, since the apparent motion in these frames is caused by camera movement. Since the directions of the background motion vectors are from right to left, we conclude that the camera moves from left to right. In Fig 2e-g, we provide sample frames that contain motion vectors of moving objects. In Figure 2f,g, the magnitudes of motion vectors are larger than the background motion vectors, which demonstrate that the woman and the black car are moving in the opposite camera direction. Therefore, subtraction of the camera motion vector from the object motion vector significantly increases the magnitude of object motion vectors. On the other hand, in Figure2e, the car moves in the same direction with the camera, yet with a higher velocity compared to the camera motion. Hence, we can still detect the motion of the car in this example.

3.2. Moving Object Detection

After estimating the camera motion, we can determine the overlapping parts of two consecutive frames. To do this, we crop the unmatching parts from both of the frames using the camera motion vector. For example, if the camera moves in the north-east direction, frame $T$is cropped from its bottom and left; frame $T + 1$is cropped from its top and right, at an amount that is equal to the camera motion vector magnitude. In this way, we obtain two consecutive frames that share the same background view (Figure 3). This region is considered as the actual background for the scene.

After detecting the background, we simply take the differences of the two frames to detect the moving parts inside the overlapping area. We use an adaptive thresholding method to get better results considering different lighting conditions in different areas within a frame [11].



(a)

(b) (c) (d)

(e) (f) (g)

FIGURE 2. A set of motion vectors that belong to background (b-d) and moving objects (e-g).

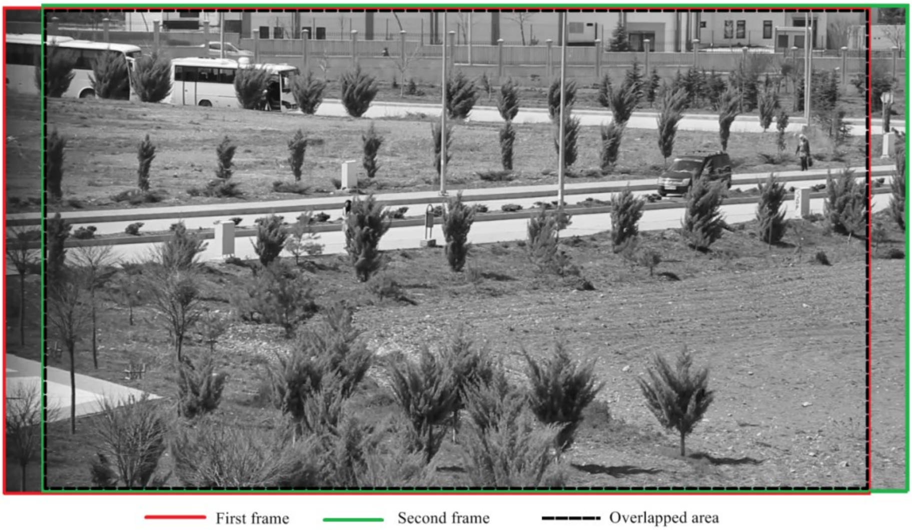


FIGURE 3. Detecting the background by cropping two consecutive frames.

3.3. Improving Detected Object Boundaries

Although the locations of moving objects are detected correctly, it is hard to detect the whole coverage of the moving objects, accurately. Most of the times, the area that belong to the same object is obtained as a set of disconnected small segments rather than one single segment. When we enclose each detected parts within a bounding box, there are a lot of overlapping bounding boxes that belong to different parts of same objects (Figure 4a). Therefore, we need to merge these segments to find a single bounding box for each object. This is a difficult problem especially when some parts of the object are occluded.

In order to solve this problem, we developed a heuristic method to determine all segments that belong to a single object. In this method, after we find all moving parts of the scene, we create a bounding box around each disconnected segment and augment each box with its width, height, diagonal length, and a representative motion vector of the box. The representative motion vector of a bounding box is calculated by averaging the motion vectors of interest points in the bounding box. If there are no interest points in the box, motion is considered as the displacement of the bounding box center in two consecutive frames. Then, we sort all bounding boxes by their diagonal length from largest to smallest. Starting from the biggest bounding box, we examine every two bounding boxes. If the distance between the centers of the two parts is less than the sum of the diagonals of the two parts, then we define them as close-by parts. If two parts are close-by and their motion vectors are similar, we assume that these two parts belong to the same object. In this case, we merge the two bounding boxes and update the augmented data of the new bounding box corresponding to the new region. The merge operation deletes the previous bounding boxes from the list and inserts the newly created box for the following iterations. The algorithm iterates this step until no merging is possible. As it is shown in Figure 4a, there are many small parts that are marked using independent bounding boxes. On the other hand, Figure 4b shows that our method correctly groups all the parts that belong to the same objects. As it is visible in the same figure, even if we could group the bounding boxes correctly, which covers the area of the moving objects almost completely, we still could not obtain the whole object boundary accurately.

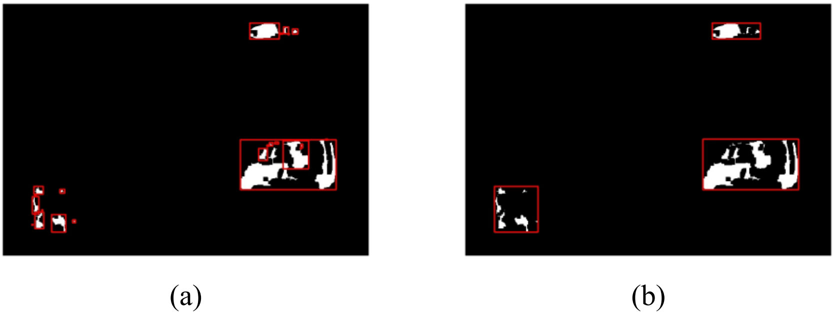

(a) (b)

FIGURE 4. a) Initial and b) final bounding boxes of each objects.

In order to solve this problem, we utilize a segmentation algorithm that considers only the regions corresponding to the estimated bounding boxes in the original image. This helps us to obtain better segmentations of the objects that are determined in the previous step. For this purpose, we use GrabCut object segmentation method to get more accurate object boundaries [31]. In our implementation, we provide a rectangle that marks the object regions for segmentation and run the GrabCut algorithm in each region, separately. Although GrabCut method performs well most of the times, it may generate inaccurate segmentations when the region is too small or the intensity values of an object and its surrounding is very similar. Therefore, we combine our initial motion based estimations with the segments that we obtain using the GrabCut algorithm in a hybrid solution. This method invalidates a result that is obtained from GrabCut method if it is less than half of the area in our method. In such cases, we keep the original region as it is. Figure 5 shows the obtained moving objects by applying our previous method [11] and our extended hybrid approach using the GrabCut method. As it is seen from the figure, object silhouettes are improved significantly in the combined approach.

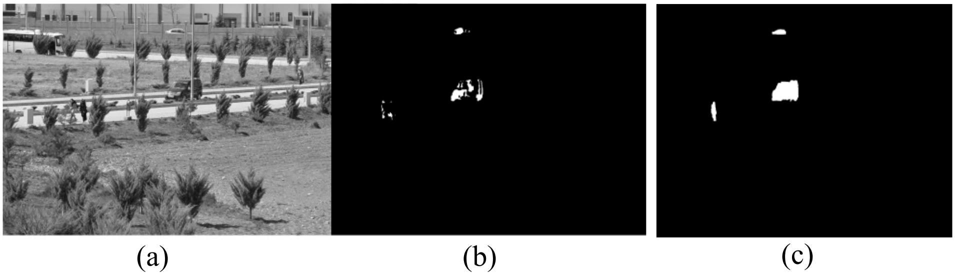


FIGURE 5. a) A video frame, b) moving objects obtained by [11] c) moving objects obtained our improved object boundary detection approach.

3.4. CLASSIFICATION

The main motivation of this research is automatic detection of the trespassing across the country borders and classification of these trespassing objects as people, vehicle, or others. We developed a working decision tree based approach for this purpose. We use aspect ratio, compactness, and the velocity vectors of the detected regions as features. Our classifier is configured based on three main observations:

- People tend to move more slowly than vehicles.
- People have more complex boundary structure than vehicles. Therefore, compactness values of people are lower than vehicles.
- People have aspect ratios between 1.8 and 3.7 as shown in [17].

We implemented a rule-based decision tree for object classification considering these observations. The pseudocode of the classification method is shown in Algorithm 1. Here, V is the velocity, C is the compactness, and $AR$is the aspect ratio of the moving object. $C$and AR are defined by equations (1) and (2), respectively.

$$C = \frac{Area}{Perimeter^2} \tag{1}$$

$$AR = \frac{\text{Height}}{\text{Width}} \tag{2}$$

ALGORITHM 1. Classification

```
if $V > T_{v1}$ then
  object ← others
else
  if $V < T_{v2}$ then
    if $AR \geq 1.8$ and $AR \leq 3.7$ then
      Look at 5 past frames. If it was a vehicle in past frames, object ← vehicle otherwise object ← people
    else
      if $C > T_C$ then
        object ← vehicle
      else
        Look at 5 past frames. If it was a vehicle in past frames, object ← vehicle otherwiseobject ← people
      end if
    end if
  else
    if !($AR \geq 1.8$ and $AR \leq 3.7$ ) then
      object ← vehicle
    else
      if $C \leq T_C$ then
        object ← people
      else
        Look at 5 past frames. If it was a vehicle in past frames, object ← vehicle otherwise object ← others
      end if
    end if
  end if
end if
```

Velocity is the displacement of an object in consecutive frames as we mentioned before. However, because of the perspective projection, using only the amount of displacement to estimate the velocity can be misleading most of the times. Therefore, we normalize the velocity of an object with the object size approximately by dividing it to its diagonal length. This interpretation is more robust for discriminating objects according to their velocities in different spatial positions in the image.

In order to classify objects, we first consider the velocity feature. If an object appears in one frame and disappears in consecutive frames, it is considered as a false alarm and ignored by setting its velocity to zero. On the other hand, if the velocity of an object is more than a predetermined threshold,T_{v1}, it is classified as others, e.g. a bird passing in front of the camera.

In order to distinguish people and vehicle, we determine a velocity threshold valueT_{v2}. If the object velocity is lower than T_{v2}, object can be a person, a group of people, or a slow vehicle. Considering these possibilities, we check the aspect ratio of the object as the second criterion. If it matches with the aspect ratio of people, there are two possible categories: people or slow vehicle which is partially occluded. To resolve this ambiguity, we consider the history of detected objects by utilizing a small (e.g. five frame length) buffer. If it is marked as a vehicle in the history buffer, it is classified as vehicle; otherwise, it is classified as people. If the aspect ratio does not match with the specified range, there is still a possibility that the object belongs to people category since the aspect ratio of a group of people is different from the aspect ratio of a single person. Therefore, to separate a group of people and a vehicle, we use the compactness feature. Compactness of a group of people is lower than compactness of a vehicle because they have more complex structure than vehicles. Therefore, if the compactness value of an object is higher than T_c, we classify it as a vehicle, else we check the history buffer to avoid occlusion based mistakes. If it is marked as a vehicle in the history buffer, it is classified as a vehicle; otherwise, it is classified as people.

If the object velocity is higher than T_{v2}, the probability of the object being a vehicle is higher. However, we need to consider the possibility of the object being people, due to the perspective projection. If the object aspect ratio is different from the people aspect ratio range, the object is classified as a vehicle. Otherwise, if both the aspect ratio and the compactness values are in the ranges that are defined for people, the object is classified as people. Otherwise, we look at our history buffer again. If it is marked as a vehicle in the history buffer, it is classified as a vehicle; otherwise, it is classified in the others category.

In Figure 6, we present some exemplary results, where the bounding boxes of the objects are depicted using different colors depending on their classes, i.e. red box is used for people, green box is used for vehicles and blue box is used for others classes. These results belong to the GSD benchmark. In our implementation, we determined the threshold values experimentally by observing the object behaviours of Golbasi1-5 videos. In all experiments reported in this research we set the threshold values T_{v1} as 0.51, T_{v2} as 0.020, and T_c as 0.031 based on the observations.

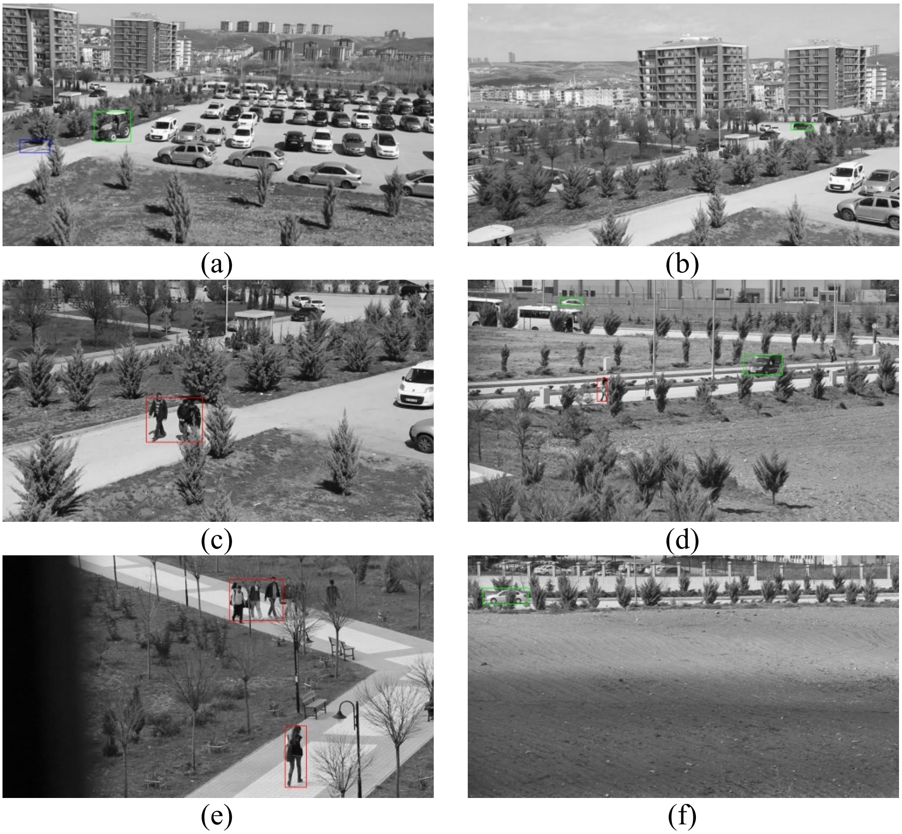

(a) (b) (c) (d) (e) (f)

FIGURE 6. Classification result samples from Golbasi1-6 videos respectively.

4. Results And Discussions

In this research, we provided a solution to the challenging problem of automatic object detection and identification from surveillance videos that are recorded using moving cameras. To the best of our knowledge, there is no published benchmark dataset that is suitable for testing our method. Therefore, we created a new dataset, which is named as Golbasi surveillance dataset (GSD). GSD contains video records of some regions in Golbasi Campus at Ankara University. The environment around the campus is similar to a country border region that is largely composed of wild landscapes with small bushes and wild birds and contains only a few moving objects around. Moving objects are generally vehicles and people walking alone or in groups. Thus, the dataset is suitable for our purposes. The dataset contains two sets of recordings in different qualities: Some videos are recorded by Canon 650D, with 1920x1080 resolution while some of them are recorded by a webcam with 640x480 resolution.

In order to evaluate the performance of our object detection approach, we determine precision (P) and recall (R) rates.

Table 1 shows the total number of frames and moving objects in each video of our Dataset. We give the precision and recall rates of our object detection method in the last two columns of Table 1. We calculate these rates using the ground truth data of the corresponding videos. In these calculations, we use the following procedure: If the bounding box of a moving object overlaps with the bounding box in ground truth file, we treat it as a true positive (TP). If it is classified incorrectly by our method, we take it as a false positive (FP). If an object in the ground truth file is not detected by our method, i.e. missed, we mark it as a false negative (FN). The system sums all TP, FP, and FN objects in all the frames in a video and then calculates the precision and recall rates using Eq. (3) and (4), respectively.

$$P = TP/(TP + FP) \tag{3}$$

$$R = TP/(TP + FN) \tag{4}$$

TABLE 1. Precision and recall rates for moving object detection.

Video	# of Frames	Resolution	# of Objects	Precision Rate	Recall Rate
Golbasi1	60	1920 x 1080	81	97,46	95,06
Golbasi2	60	1920 x 1080	94	89,18	70,21
Golbasi3	60	1920 x 1080	60	100	100
Golbasi4	60	1920 x 1080	237	100	72,57
Golbasi5	60	1920 x 1080	160	95,93	73,75
Golbasi6	60	1920 x 1080	60	79,16	95
Golbasi7	66	1920 x 1080	66	90,27	98,48
Golbasi8	160	640 x 480	131	94,02	96,18
Golbasi9	109	640 x 480	102	100	88,23
Golbasi10	235	640 x 480	235	91,39	94,89
Golbasi11	100	640 x 480	100	97,05	99
Golbasi12	172	640 x 480	210	97,96	91,90

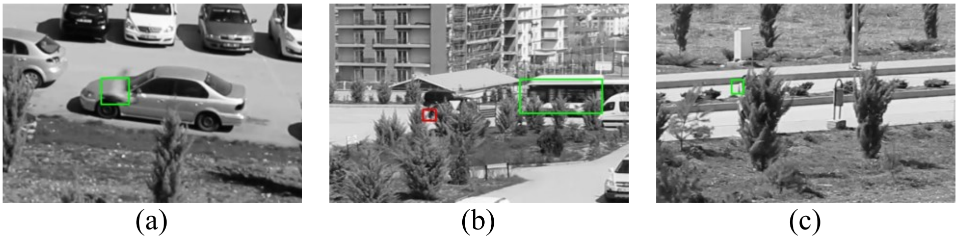
The determined precision rates in Table 1 show, our approach achieves very high TPs and low FPs in most videos. Even in some cases (e.g. Golbasi3 and Golbasi4), there is no observed FP at all. On the other hand, we observe low recall rates in some videos (e.g. Golbasi2, 4 and 5). When we analyzed these videos, we realized that some of the moving objects are occluded by other objects such as bushes, branches of trees. Therefore, our method was not able to detect them in such cases and as a result, recall rate decreases. The average precision and recall values are 94,36% and 89,60%, respectively. As can be seen in Table 1, despite the resolution differences, precision and recall results are close to each other. We believe that it is due to the large field of view; although the resolutions are different, moving objects occupy only a small percentage of the whole scene and the majority of the scene is background in both settings. Therefore, differences in the resolution do not significantly affect the camera motion computation.

Table 2 shows the number of correctly detected moving object and precision rates for our classification method. The precision rates are calculated only for the correctly detected objects. For example, in Golbasi1, there are 81 moving objects and the recall rate of the video is 95,06% as seen in Table 1. This means 77 of objects are detected correctly and 4 of them are missed. Therefore, the classification precision rate for this video is calculated based on these 77 objects. In general, if an object is occluded by the environment or only a part of an object appears in the field of view, it is sometimes misclassified. However, when the object is not occluded any more, the classification error is corrected immediately. As seen from Table 2, the *P* value for classification step is in 80.51% and 100% range. For twelve videos, the average *P* is 90,03%. The precision results of our object detection and classification approaches show that the proposed method works effectively in this domain.

TABLE 2. Presicion rates for classification.

Video	# of Objects	Precision Rate
Golbasi1	77	80,51
Golbasi2	66	83,33
Golbasi3	60	91,66
Golbasi4	172	81,97
Golbasi5	118	86,44
Golbasi6	57	91,22
Golbasi7	65	96,92
Golbasi8	126	96,82
Golbasi9	90	100
Golbasi10	223	98,20
Golbasi11	99	87,87
Golbasi12	193	85,49

In Figure 7, we illustrate some challenging samples from the GSD that are misclassified by our method. In the first example, a bird that is flying over a car is classified as a vehicle since its velocity and aspect ratio is similar to a vehicle. In the second figure, a car, which is occluded by bushes, is classified as people. In the last figure, a woman who is walking behind a tree is classified as a vehicle. Due to the high amount of occlusion, she is barely noticeable even by a human observer. Note that these misclassifications are valid only for the depicted video frames. In the following frames, the occlusion problem is resolved and these misclassifications are corrected.



(a) (b) (c)

FIGURE 7. Wrong classification examples. a) A bird is classified as vehicle. b) A car is classified as people. c) A woman is classified as vehicle.

5. Conclusion

In this paper, we presented a method for detecting and classifying moving objects using a moving camera. In our method, we calculate the camera motion by assuming that the majority of the motion vectors in the scene are generated by the camera motion. After we eliminate the camera motion, we use the differences between the consecutive frames to detect the moving objects. Since, we are primarily concerned with the detection and identification of the trespassings across the country borders, we developed a classification scheme on top of our moving object detection framework. In this framework, we classified objects as people, vehicle, or others using a rule based decision tree method. We utilize aspect ratio, compactness, and the normalized velocity of moving objects as the features in our classifier. We also use previous frame information to make the right decisions when some parts of the objects are temporarily occluded. In the context of this study, we generated a new dataset that can be used as a benchmark in this problem domain for future researches. The experiments that we perform using this dataset show that the proposed scheme is very effective for detection and classification of objects.

References

[1] P. Remagnino, S. A. Velastin, G. L.Foresti, and M. Trivedi, Novel concepts and challenges for the next generation of video surveillance systems, *Machine Vision and Applications,* 18/3 (2007) 135-137.

[2] R. Vezzani and R. Cucchiara, Video surveillance online repository (visor): an integrated framework, *Multimedia Tools and Applications*, 50/2 (2010) 359-380.

[3] A. S. Murugan, K. S. Devi, A. Sivaranjani, P. Srinivasan, A study on various methods used for video summarization and moving object detection for video surveillance applications, *Multimedia Tools and Applications*, (2018) 1-18.

[4] K. A. Joshi and D. G. Thakore, A survey on moving object detection and tracking in video surveillancesystem, *International Journal of Soft Computing and Engineering*, 2/3 (2012) 44-48.

[5] M. Chate, S. Amudha, V. Gohokar, Object detection and tracking in video sequences, *ACEEE International Journal on signal & Image processing*, 3/1(2012).

[6] W.-C. Hu, C.-H. Chen, T.-Y. Chen, D.-Y. Huang, and Z.-C. Wu, Moving object detectionand tracking from video captured by moving camera, *Journal of Visual Communication and Image Representation,* 30 (2015) 164-180.

[7] S. W. Kim, K. Yun, K. M. Yi, S. J. Kim, and J. Y. Choi, Detection of moving objects with a moving camera using non-panoramic background model, *Machine Vision and Applications*, 24/5(2013) 1015-1028.

[8] N. Thakoor, J. Gao, and H. Chen, Automatic object detection in video sequences with camera in, *Proceedings of Advanced Concepts for Intelligent Vision Systems*, Citeseer, 2004.

[9] Y. Wu, X. He, and T. Q. Nguyen, Moving object detection with a freely moving camera viabackground motion subtraction, *IEEE Transactions on Circuits and Systems for Video Technology*, 27/2 (2017) 236-248.

[10] K. Yun, J. Lim, and J. Y. Choi, Scene conditional background update for moving object detection in a moving camera, *Pattern Recognition Letters*, 88 (2017) 57-63.

[11] O. M. Sincan, V. B. Ajabshir, H. Y. Keles, and S. Tosun. Moving object detection by a mounted moving camera, *EUROCON 2015 - International Conference on Computer as a Tool (EUROCON), IEEE*, Sept 2015.

[12] L. Bo and Z. Heqin, Using object classification to improve urban traffic monitoring system, *InternationalConference on Neural Networks and Signal Processing,* 2 (2003) 1155-1159.

[13] L. Chen, R. Feris, Y. Zhai, L. Brown, and A. Hampapur, An integrated system for moving object classification insurveillance videos, *2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance,* (2008) 52-59.

[14] M. Elhoseiny, A.Bakry, and A.Elgammal, Multiclass object classification in video surveillancesystems-experimental study, *Proceedings of the IEEE Conference on Computer Vision and Pattern RecognitionWorkshops*, (2013) 788-793.

[15] O. Javed and M. Shah, Tracking and Object Classification for Automated Surveillance, Springer Berlin Heidelberg, (2002) 343-357.

[16] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, R. Bolle, Appearance models for occlusion handling, *Image and Vision Computing Performance Evaluation of Tracking and Surveillance,* 24/11(2006) 1233-1243.

[17] A. A.Shafie, A. B. M. Ibrahim, and M. M. Rashid, Smart objects identificatio nsystem for robotic surveillance, *International Journal of Automation and Computing,* 11/1 (2014) 59-71.

[18] Y.Gurwicz, R. Yehezkel, B. Lachover, Multiclass object classification for real-time video surveillance systems, *Pattern Recognition Letters*, 32/6 (2011) 805-815.

[19] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, A large-scale benchmark dataset for event recognition insurveillance video, *Computer vision and pattern recognition (CVPR)*, (2011) 3153-3160.

[20] A. García-Martín and J. M. Martínez, People detection in surveillance: classification and evaluation, *IET ComputerVision*, 9/5 (2015) 779-788.

[21] C. Hua, Y. Makihara, Y. Yagi, S. Iwasaki, K. Miyagawa, and B. Li, Onboard monocularpedestrian detection by combining spatio-temporal hog with structure from motion algorithm, *Machine Vision and Applications,* 26/2 (2015) 161-183.

[22] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, *IJCAI*, (1981) 674-679.

[23] R. Hartley, R. Gupta, and T. Chang, Stereo from uncalibrated cameras, *Proceedings 1992 IEEE ComputerSociety Conference on Computer Vision and Pattern Recognition,* (1992) 761-764.

[24] A. Prioletti, A. MGelmose, P. Grisleri, M. M. Trivedi, A. Broggi, and T. B. Moeslund, Part-based pedestrian detection and feature-based tracking for driver assistance: Real-time, robust algorithms, and evaluation, *IEEE Transactions on Intelligent Transportation Systems,* (2013) 14/3 1346-1359.

[25] I.Jegham and A. B. Khalifa, Pedestrian detection in poor weather conditions using moving camera, *In Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference*, (2017) 358-362.

[26] M. Yazdi, T. Bouwmans, New trends on moving object detection in video images captured by a moving camera: a survey, *Computer Science Review*, 28(2018) 157–177.

[27] M. Babaee, D. T. Dinh, G. Rigoll, A deep convolutional neural network for video sequence background subtraction, *Pattern Recognition,* 76 (2018) 635-649.

[28] A. Rozantsev, V. Lepetit, and P. Fua, Detecting Flying Objects using a Single Moving Camera, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39 (2017) 879-892.

[29] J. Shi and C. Tomasi. Good features to track, *In 1994 Proceedings of IEEE Conference on Computer VisionPattern Recognition,* (1994) 593-600.

[30] G. Bradski and A.Kaehler, Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media,Inc., 2008.

[31] C. Rother, V. Kolmogorov, and A. Blake, Grabcut: Interactive foreground extraction usinggraph cuts, *In ACM transactions on graphics (TOG)*, 23/3 (2004) 309-314.

Current Address: OZGE MERCANOGLU SINCAN: Ankara University, Faculty of Engineering, Department of Computer Engineering, 06830, Gölbaşı, Ankara, TURKEY
E-mail: omercanoglu@ankara.edu.tr,
ORCID: https://orcid.org/0000-0001-9131-0634
Current Address: HACER YALIM KELES: Ankara University, Faculty of Engineering, Department of Computer Engineering, 06830, Gölbaşı, Ankara, TURKEY
E-mail: hkeles@ankara.edu.tr
ORCID: https://orcid.org/0000-0002-1671-4126
Current Address: SULEYMAN TOSUN Hacettepe University, Faculty of Engineering, Department of Computer Engineering, 06800, Beytepe, Ankara, TURKEY
E-mail:stosun@hacettepe.edu.tr
ORCID: https://orcid.org/0000-0002-3708-2009