Machine Learning for Trajectory Generation of Multiple-pedestrians

Hye-Yeon Yu(✉), Young-Nam Kim, and Moon-Hyun Kim

College of Information and Communication Engineering,
Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si
Gyeonggi-do 440-746, Republic of Korea
{yu0529,hwarangjin,mhkim}@skku.edu

Abstract. In this paper, we provide an algorithm for generating a trajectory in real time to identify the pedestrians. Typically, the contours for the extraction of pedestrians from the foreground of images are not clear due to factors including brightness and shade; furthermore, pedestrians move in different directions and interact with each other. These issues mean that the identification of pedestrians and the generation of trajectories are somewhat difficult. We propose a new method for trajectory generation regarding multiple pedestrians. The first stage of the method distinguishes between those pedestrian blob situations that need to be merged and those that require splitting, followed by the use of trained decision trees to separate the pedestrians. The second stage generates the trajectories of each pedestrian by using the point-correspondence method; however, we introduce a new point correspondence algorithm for which the A* search method has been modified. By using fuzzy membership functions, a heuristic evaluation of the correspondence between the blobs was also conducted. The proposed method was implemented and tested with the PETS 2009 dataset to show an effective multiple-pedestrian-tracking capability in a pedestrian interaction environment.

Keywords: Artificial intelligence · Multiple-target tracking · Machine learning · Decision tree

1 Introduction

For a long period of time, tracking has been an important theme in terms of computer vision, and it remains a challenging task. Computer-vision tracking involves the detection of the moving regions of images, the separation of foreground pedestrians from the background, and movement prediction. The tracking of a pedestrian is straightforward, as the pedestrian can easily be found in each frame, and the elapsed time can be used to determine the trajectory of the detected pedestrian; however, the generation of the trajectories of multiple tracked pedestrians is more complicated. During the tracking of multiple pedestrians, it is possible to mistakenly identify one of the other pedestrians as the target pedestrian, or to miss the detection of a pedestrian due to inter-pedestrian interactions such as occlusions.

J.J. (Jong Hyuk) Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*,
Lecture Notes in Electrical Engineering 421, DOI 10.1007/978-981-10-3023-9_63

The objects tracking and detection is the most important and essential step in computer vision. The main target of object tracking is to detect and track the moving objects such as pedestrians.

Occlusions and overlaps occur very commonly in pedestrian scenes. Depending on incomplete object detection, tracking algorithm is more complex. Calculation for connecting the tracklets belonging to the same individual is NP-complete problem.

Yilmaz et al. [1] categorize the tracking methods on the basis of the object and motion representations used. There are two main processes in motion-based recognition: One is a motion representation, another is the extraction of motion information from a sequence of images. In general, for extracting motion information, using the two methods as follows: motion correspondence and optical flow [2]. Eom et al. [3] proposed a heuristic search-based motion correspondence algorithm using fuzzy clustering. They focused on the motion corresponding algorithm. And in the experiment, corresponding points is a preprocessing. In this paper is an improved version of their pre-processing phase. Pasquale Foggia et al. [6] proposed system is able to simultaneously track single objects and groups of objects. The object classifier module classifies each single blob into different classes.

2 Proposed Method

The diagram of the proposed method is shown in Fig. 1. We separated the foreground from the background by using a Gaussian mixture model whereby the separated foreground consists of blobs. Each pedestrian blob is represented by features such as area, perimeter, aspect ratio, center of mass, average color, and the distance to the closest blob in the previous frame; based on training-set learning, these features are used for the construction of the decision trees. The resultant learned decision trees are used to classify the blobs as either "split" or "merged," and the splitting and merging processes assign blobs to each pedestrian.

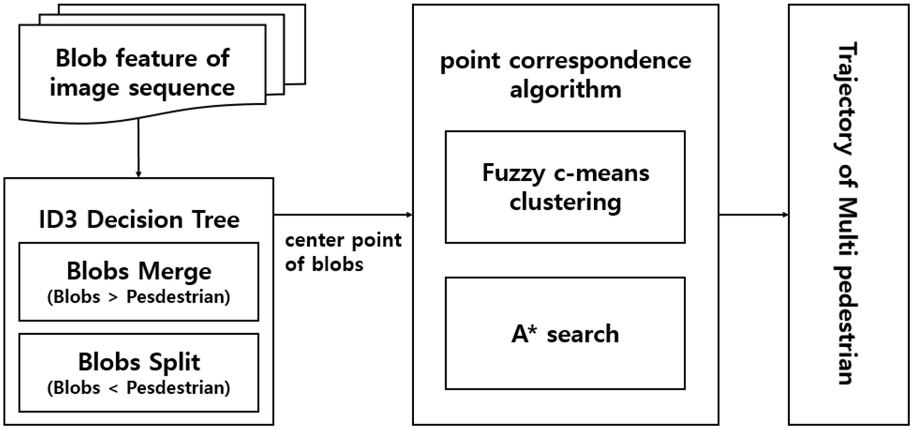


Fig. 1. Diagram of the proposed method

To associate a blob in the current frame with a tracked pedestrian from a previous frame, we devised a modified A* heuristic search algorithm. For each blob, the search for the most-probable pedestrian from the previous frame is based on the Euclidean distance and the difference between the velocity angles of the blob and the pedestrian. The required heuristic function that computes the possibility that a tracked pedestrian corresponds with a blob was designed using a fuzzy-C-means (FCM) clustering algorithm. It is the association of the blobs with the pedestrians over the entire frame sequence that produces the trajectories, each of which is a trajectory for a single pedestrian.

2.1 ID3 Decision Trees

The ID3 decision tree is an information-theoretical algorithm invented by Ross Quinlan [7]. Each iteration of the ID3 involves the attainment of the largest information gain (or smallest entropy value), and a probability of p is determined from the occurrence frequency. The entropy of a given B_M is computed as follows:

$$E(B_M) = -\sum_{C_i \in C} p(C_i) \log_2 p(C_i) \tag{1}$$

where C is a set of classes in B_M that is either of a positive (merging) class or a negative (not-merging) class, and p(C_i) is the proportion of the number of pairs labeled as class C_i in B_M to $|B_M|$. The *information gain(IG)* from the partitioning of B_M according to the value of the feature f_M is computed using Eq. (2), as follows:

$$IG(f_{M_i}, B_M) = E(B_M) - \sum_{v_j \in v_{M_i}} p(v_j) E(B^M_{v_j}) \tag{2}$$

In Eq. (2), $E(B^M_{v_j})$ is the entropy of the subset $B^M_{v_j}$ that is constructed through the collection of pairs, each of which has a value of v_j for the feature f_{M_i}, from B_M. V_{M_i} denotes a set of values for f_{M_i}; therefore, $B_M = \bigcup_{v_j \in M_i} B^M_{v_j}$. The probability $p(v_j)$ is computed using Eq. (3), as follows:

$$p(v_j) = \frac{\left|B^M_{v_j}\right|}{|B^M|} \tag{3}$$

At each iteration of the construction of a decision tree, the feature that provides the maximum information gain is selected as a decision node. To deal with the numeric features, the C4.5 [8] ID3 algorithm is used.

2.2 Trajectory Generation

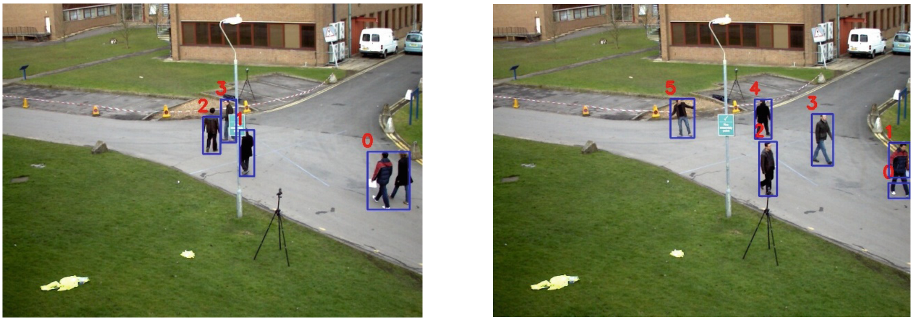
A* algorithm is a typical heuristic shortest path search algorithm. A* algorithm select the node with the minimum value of the values that have been calculated by guidance function that uses a heuristic estimate to a depth first search [5]. The evaluation function in search node n is given in Eq. (4).

$$f(n) = g(n) + h(n) \tag{4}$$

With g(n): path cost to the n node from the starting node. h(n): estimated cost of the path to the target node from the n node. h(n) usually use the manhattan distance. The Algorithm progress that the node with a value of shortest guidance function in node to be connected form the starting node is to be a node that make up the shortest path, but it will end by selecting the target node. A* algorithm has the advantage that a search time is shortened in accordance with finding the target node without comparison with other paths. However, in a complex and unstable network, there is the potential for that to fail the search for shortest path. The number of clusters and the initial cluster center value at the approximate point are calculated by using a Mountain method [5]. With many changes by the FCM algorithm repeatedly, the positions of the cluster centers that are determined roughly is determined the final cluster center. Using the degree of affiliation calculated by the FCM algorithm, cost of all correspondence are calculated then the cost is used in making the correspondence matrix. Finally, an optimal trajectory is generated by the correspondence matrix and the A* algorithm.

2.3 Experiment Results

We used the Caviar "Crowd of four people meet, walk and split" sequence, PETS 2009 dataset S2 from the L1 sequence view 001 and the L2 sequence view 001 as shown in Fig. 2. The Caviar sequence is used only for tracking in clean environment. WEKA (Waikato Environment for Knowledge Analysis) version 3.7, written in JAVA, is an experimental tool used for machine learning [11] that we used for ID3 machine learning and the visualization of the decision trees. Blob detection and a trajectory-generation algorithm were implemented in OpenCV 2.4.2 with Visual Studio 2010.

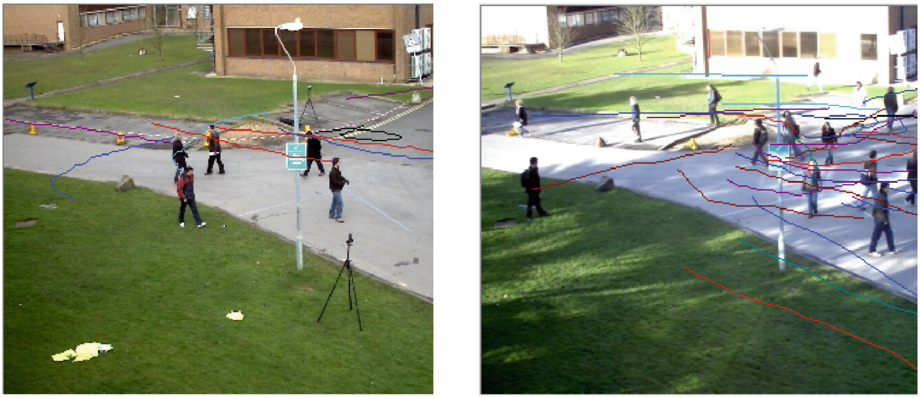


(a) In the merging situation (b) In the splitting situation

Fig. 2. Merge or split situation

Figure 2 shows splitting or merging situations for the detected blobs in the image. Two pedestrians are detected as one blob, so the blob number 0 of Fig. 2 (a) needs to be split; furthermore, one pedestrian is detected as two blobs, so the blob number 0 and the blob number 1 of Fig. 2 (b) need to be merged.

Figure 3 shows the generated trajectories for the S2 views of the L1 and L2 sequences from the PETS 2009 dataset. The pedestrian group consists of nine people in the (a) L1 sequence and 33 people in the (b) L2 sequence. The pedestrians of (a) frequently change their moving direction in comparison with those of (b). The L1 and L2 sequences generated trajectories of 30 frames and 26 frames, respectively.



(a) L1 ground truth trajectories

(b) L2 ground truth trajectories

Fig. 3. Multiple trajectories of pedestrians

3 Conclusion

Generating a trajectory in the multiple pedestrian have a lot of difficult factors. In the previous research, presented excellent results from trajectory generation of multiple target. The biggest problem for the application of previous research to multiple pedestrian tracking is the interaction between each other. We apply the ID3 algorithm to solve the problem of the interaction of the pedestrian. ID3 algorithm, the blobs overlap between the pedestrian was used to define the split and merge situations. Redefine the blobs to display a multiple-pedestrian in the experiment, to calculate the corresponding cost. Multiple-pedestrian trajectory is connection of the minimum cost of the node in A* search algorithm. Our experiments solved several constraints problems for the applied of point correspondence method to trajectory generation of multiple-pedestrian. The experimental results are classified as unclear blob splitting and merging situations. Thereafter, to generate a trajectory of each multiple-pedestrian as FCM based point correspondence method. The results of the research was successful in generating the trajectory of the multiple-pedestrian to the frequent interaction. In the future, to improve the accuracy of classification, via the ID3 algorithm for improved accuracy, generating the trajectory of each multiple-pedestrian with Datasets in diversity of experiments is expected to proceed to the next research challenge.

Acknowledgement. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIP) (NRF-2014R1A2A1A11053902).

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. (CSUR) **38** (4), 13 (2006)
2. Cedras, C., Shah, M.: Motion-based recognition a survey. Image Vis. Comput. **13**(2), 129–155 (1995)
3. Eom, K.-Y., Jung, J.-Y., Kim, M.-H.: A heuristic search-based motion correspondence algorithm using fuzzy clustering. Int. J. Control Automat. Syst. **10**(3), 594–602 (2012)
4. Yu, H.-Y., Kim, Y.-N., Kim, M.-H.: Temporal search algorithm for multiple-pedestrian tracking. KSII Trans. Internet Inf. Syst. **10**(5), 2310–2325 (2016)
5. Yager, R., Filev, D.: Approximate clustering via the mountain method. IEEE Trans. Syst. Man Cybern. **24**(8), 1279–1284 (1994)
6. Foggia, P., et al.: Real-time tracking of single people and groups simultaneously by contextual graph-based reasoning dealing complex occlusions. In: 2013 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), IEEE (2013)
7. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
8. Salzberg, S.L.: C4.5: programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers Inc, 1993. Mach. Learn. **16**(3), 235–240 (1994)
9. Berclaz, J., et al.: Multiple object tracking using K-shortest paths optimization. IEEE Trans. Pattern Anal. Mach. Intell. **33**(9), 1806–1819 (2011)
10. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
11. Weka 3 - Data Mining With Open Source Machine Learning Software In Java. www.cs.waikato.ac.nz
12. Ferryman, J., Shahrokni, A.: PETS 2009 (2009). http://www.cvg.rdg.ac.uk/PETS2009
13. Homepages.inf.ed.ac.uk, CAVIAR Test Case Scenarios (2015). http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/