Abnormal Crowd Behaviour Recognition in Surveillance Videos

Franjo Matković
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
franjo.matkovic@fer.hr

Darijan Marčetić
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
darijan.marcetic@fer.hr

Slobodan Ribarić
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
slobodan.ribaric@fer.hr

***Abstract*— The paper presents an approach to crowd behaviour recognition in surveillance videos. The approach is based on a 4-stage pipelined multi-person tracker adapted to microscopic crowd level representation and crowd behaviour recognition by the evaluation of fuzzy logic functions. The multi-person tracker combines a CNN-based detector and an optical flow-based tracker. The following tracker features are used: optical flow and histogram of optical flow orientation at the macroscopic level, and the tracklets and trajectories of a person and/or group of people at the microscopic level. The human interpretation of video sequences (real and/or video sequences obtained by simulators of crowds) is mapped into fuzzy logic predicates and fuzzy functions. Fuzzy logic predicates specify crowd motion patterns at the microscopic level for a person and/or group of people. They are building blocks of fuzzy logic functions which describe different scenarios of characteristic crowd behaviour. The preliminary results of three experiments for a runaway scenario show that the approach supports efficient and robust crowd behaviour recognition in surveillance videos.**

Keywords*— *Multi-person tracker, Crowd, Crowd behaviour recognition, Motion patterns, Fuzzy logic

I. INTRODUCTION

Video surveillance and automatic anomaly behaviour detection and interpretation have an important role in our information society in ensuring security and increasing safety in public and semi-public places where many people are gathered. One of the most active research areas in computer vision has been crowd anomaly detection, behaviour recognition and analysis [1-3]. A crowd is defined as a large number of individuals or mass of people gathered in the same physical environment and usually sharing common goals [4]. The main processes of crowd analysis are the following: crowd detection, crowd motion and tracking, crowd density estimation, and crowd behaviour recognition. Generally, the main crowd analysis processes may be conducted at different levels of crowd representations: i) the macroscopic level, where the global features of crowd motion and behaviour are used; ii) the microscopic level, where the features of the movements and behaviour of a person are the base for crowd analysis; and iii) the mesoscopic level which combines features from levels i) and ii). The processes are interdependent and mutually interwoven and pervade one another. For example, crowd behaviour recognition combines crowd detection, crowd motion, motion pattern generation, tracking and behaviour classification. Crowd behaviour is based on different models [2-5], such as social force models (SFM), cellular automata, agent-based models, Bayesian models, Hidden Markov models (HMM), models based on histograms of motion, and gas-kinematics or fluid dynamics models. In the paper, we deal with abnormal crowd recognition at the microscopic level by using the features of the agent-based approach (tracklets of a person or a group of people) and features which are usually inherent in the macroscopic level (dense optical flow, histogram of optical flow orientation). Furthermore, we use an approach which combines common-sense knowledge representation and fuzzy-logic-based inference to recognize the crowd behaviour.

The main contributions and/or novelties of the paper might be considered as follows: i) the design of a multi-person tracker that combines a CNN-based detector and an optical flow-based tracker. It is adapted to a microscopic level representation which combines features characteristic of the macroscopic level; ii) common-sense knowledge used for the interpretation of motion patterns and crowd behaviour; iii) fuzzy predicates for motion pattern detection and fuzzy logic functions used for crowd behaviour recognition.

The paper is organized as follows. Section II gives a short overview of relevant research work. Section III describes a model of the proposed system for crowd behaviour recognition. It deals with a pipelined structure of trajectory generation, the detection of crowd motion patterns and recognition of crowd behaviour. Section IV describes the experimental set-up and presents the preliminary results of crowd behaviour recognition. Finally, section V summarizes the results and gives directions for future research.

II. RELATED WORK

Crowd behaviour recognition is mainly based on crowd motion patterns, where the patterns are defined as any recognizable spatio-temporal regularity of a moving crowd. Depending on the level of crowd representation, motion patterns are represented based on: object/entity movement (microscopic level); the global movement of the crowd (macroscopic level); and a combination of object/entity and global movement (mesoscopic level).

Santoro et al. [6] used the macroscopic approach to crowd behavioural analysis of two or more groups of pedestrians. A sparse optical flow was computed using detected corner features by means of a pyramidal Lucas-Kanade tracker. Density-based clustering (DBSCAN) was used to group similar motion vectors. A crowd tracker was implemented based on the similarity function which combines the distance between centres of crowd masses, the differences between vectors which represent the average direction of all pixels in the cluster, and the difference between the areas of the clusters. The algorithm was tested on the PETS2009 database for the following scenarios: crowd merging; crowd splitting; and crowd collision events.

Solmaz et al. identified five crowd motion patterns (blocking, lane, bottleneck, arch, and fountainhead) by using the stability analysis of a fluid-dynamic model (Lagrangian particle model) where crowds are treated as collections of interacting particles [7]. The proposed method uses low-level local motion features obtained by optical flow and high-level

978-1-7281-5686-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SITIS.2019.00075

information obtained by analysing several regions of interest in the scene.

Ge et al. [8] described a vision-based analysis of small groups in pedestrian crowds. They combined a pedestrian detector, a particle filter tracker and a multi-object data association algorithm to extract long-term trajectories of people passing through the scene. Based on the tracklets of moving objects and the assignment of trajectories, the linear assignment problem was solved using the Hungarian algorithm.

Jodoin et al. [9] proposed a method to extract and recover global dominant motion patterns and the main entry/exit areas from a surveillance video. The proposed method is represented as a 4-stage pipeline: i) the computation of a motion histogram for each pixel by means of the Horn-Schunk /Lucas-Kanade method; ii) the computation of the orientation distribution function (ODF) from motion histograms; iii) meta-tracking based on the assignment of every pixel to an ODF; iv) clustering meta-data to obtain dominant motion patterns and main entry/exit areas. The datasets on which the authors tested the proposed approach consisted of varying crowd densities such as those found in the UCF database, changedetection.net, and their own database. The experiment showed that the method is fast and simple to implement and works on sparse and extremely crowded scenes.

Wang et al. [10] presented a method to detect global abnormal behaviour in video sequences. They introduced histograms of optical flow orientation (HOFO) as a descriptor encoding movement information in a video sequence. The method consisted of two main steps: i) the computation of histograms of optical flow orientation (HOFO) of the original global image and of the foreground image (obtained by applying background subtraction); ii) the use of a one-class support vector machine (SVM) frame classification (abnormal or normal). The algorithm was tested on UMN and PETS datasets.

Shantaiya et al. [11] presented an improvement of optical flow-based multi-object tracking by using the Kalman filter. The Kalman filter was used to handle occlusion that happens during tracking. The authors claimed that the improved optical flow-based multi-object tracker achieved better accuracy and robustness in handling the occlusion.

Colque et al. [12] introduced a novel spatio-temporal feature called Histograms of Optical Flow Orientation and Magnitude (HOFM) and entropy to detect anomalous events in videos. The experiments performed on UCSD and Subway data demonstrate that the model can handle different situations and is able to recognize anomalous events with success.

Recently, Yang et al. [13] proposed an approach to an online multi-object tracker combining optical flow and compressive tracking modelled by multiple Markov decision processes (MDPs). The approach was tested on the multi-object tracking (MOT) benchmark for pedestrian tracking and the results showed that the method had a superior performance compared to several state-of-the-art online multi-object trackers.

Besides our recent paper [14], there are only a few papers or research reports which deal with the knowledge-based approach to crowd modelling and classification [15, 16]. In [15], multi-agent-based pedestrian models for large-scale outdoor events are developed. Each pedestrian is viewed as an intelligent knowledge-based agent. Three specific models are developed based on the phases formed, including the model for crowd arrival, the model for crowd dispersal, and the model for crowd evacuation. In [16], the authors propose a formal method for knowledge representation and management in a crowded area. The presented methodology is based on ontology and a set of fuzzy rules, which provide crowd classification according to sociological theory. Ontology was implemented by Protégé editor which supports the OWL language. There are no implementation details or experimentally evaluated results. As far as we know, there have been no attempts to build a knowledge base and inference engine based on the human common-sense interpretation of crowd behaviour.

III. A MODEL OF THE CROWD BEHAVIOUR RECOGNITION SYSTEM

A model of the crowd behaviour recognition system consists of two main components (Fig. 1): i) 4-stage pipeline multi-person tracker TBD – tracking by detection; and ii) knowledge-based crowd behaviour classifier.

A. 4-stage pipeline multi-person tracker

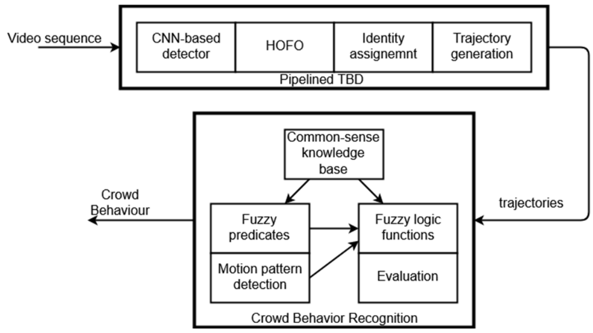

Fig. 1. A model of the crowd behaviour recognition system.

The multi-person tracking procedure is summarised in the Tracking Algorithm.

Algorithm: Tracking

INPUT: Video sequence.
OUTPUT: Person trajectories.
FOR each frame perform **STEPS 1-5**.
STEP 1: Detect person. For every detected person calculate following features: bounding box centre, dense optical flow, HOFO from optical flow, dominant motion vector (*dx, dy*) from HOFO and pdf of HOFO.
STEP 2: Regular tracking. For each active tracker whose bounding box (from a frame F_{k-1}) overlaps with only one detection (in the frame F_k), update the Kalman filter and update the tracker with features obtained in **STEP 1.**
Exceptions:
STEP 3: Multiple detection overlaps. For each active tracker whose bounding box (from a frame F_{k-1}) overlaps with multiple detections (in the frame F_k), use the Hungarian algorithm to solve the assignment problem.
STEP 4: Unassigned detections. For detections (in the frame F_k) that do not overlap with any bounding box of the active trackers (from a frame F_{k-1}), initialize a new tracker with features obtained in **STEP 1** and initialize the Kalman filter with the current detection centre position.
STEP 5: Unassigned trackers. For each active tracker whose bounding box (from a frame F_{k-1}) does not overlap with any of the detections (in the frame F_k), use the Kalman filter prediction to update the position and increment fail counter *fc*. If *fc* exceeds the threshold, the tracker is blocked.

The first stage of the pipelined multi-person tracker (Tracking Algorithm, STEP 1) is the CNN-based "off-the-shelf" YOLOv3 detector [17]. It is implemented in Keras [18] and trained on the COCO dataset with 80 classes of labelled objects. The YOLOv3 detector returns the following data:

(x_1, y_1, x_2, y_2, C, $conf$), where x_1, y_1, and x_2, y_2 are the top left corner and bottom right corner positions of the bounding box, respectively, C is one of many tens of object classes, i.e. "person", and $conf \in < 0, 1]$ is the confident level of detection. Fig. 2 illustrates the outputs of the YOLOv3 detector.

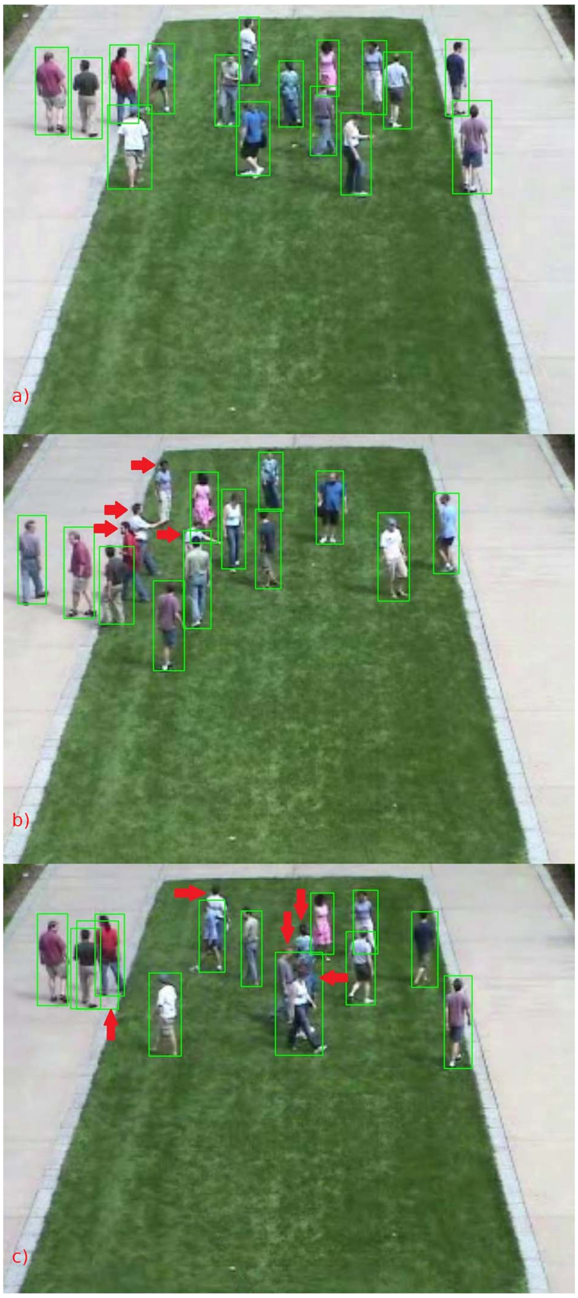

Fig. 2. Examples of the YOLOv3 detector output: a) There are no false positive (FP) or missed detections; b) missed detections; c) and false positive detections with missed detections.

For each detected person in the current frame, the following features are calculated: bounding box centre, dense optical flow in the bounding box, histogram of optical flow orientation (HOFO), probability density function of the HOFO, and a dominant motion vector.

In the second pipeline stage, a HOFO [12, 13] is calculated using dense optical flow extracted from corresponding bounding boxes from the F_k and F_{k-1} frames. Dense optical flow is obtained based on the method proposed by Farnebäck [19]. The procedure for the HOFO [12, 13] is similar to the HOG procedure [20]. Its inputs are a matrix that contains motion vectors for every pixel in the bounding box and the number of bins. Motion vectors obtained by dense optical flow are pixel movements in the x and y direction: $\mathbf{v} = (dx, dy)$. To obtain a polar histogram, the motion vectors are converted into a polar coordinate system $\boldsymbol{\rho}$ = (magnitude, angle). For every such motion vector, two neighbour bins that are closest to the angle are incremented with magnitude using linear interpolation. By normalizing the polar histogram, the probability density function (pdf) is obtained. Fig. 3 illustrates the steps of obtaining a polar histogram.

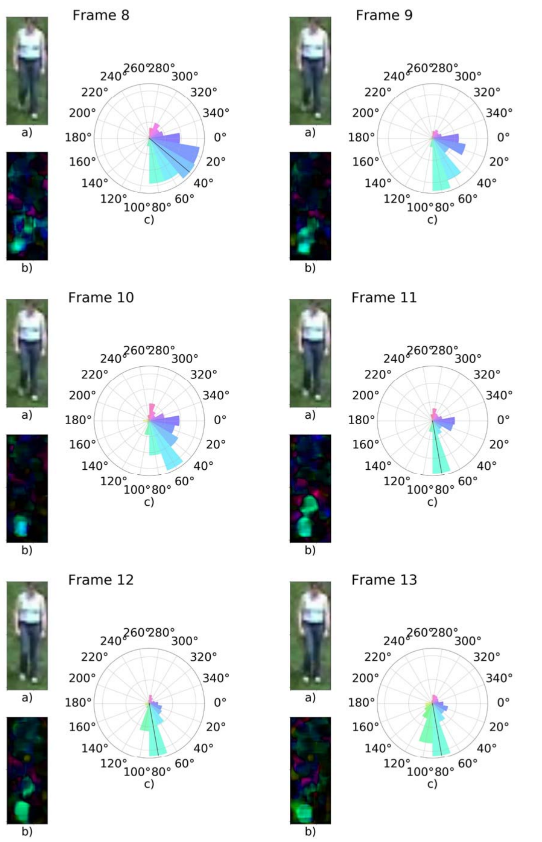

Fig. 3. Illustrations of the steps in obtaining a polar histogram: a) bounding box as a result of YOLOv3 detection; b) motion vectors of the corresponding bounding boxes based on frame F_{k-1} and F_k, where different colours represent directions of the pixel movements; c) a polar histogram of the bounding box (the number of bins is 18 – each bin corresponds to 20 degrees). A bold line going from the origin of the unimodal polar histogram to the outer circle marks the orientation of the dominant motion vector (frames 8, 11, 12, 13). Note that the polar histograms for frames 9 and 10 do not have bold lines (histograms are not unimodal) and dominant vectors are not defined.

The third stage is devoted to identity assignment. There are two main cases: regular tracking and exceptions. For regular tracking (Tracking Algorithm, STEP 2), where for each active tracker whose bounding box (from a frame F_{k-1}) overlaps with only one detection (in the frame F_k), update the Kalman filter [21], update the tracker with the features obtained in STEP 1, and assign the identity label from the previous frame (F_{k-1}).

The prediction of the next position of a person is performed by the Kalman filter. The elements of the Kalman control matrix are the components (*dx*, *dy*) of a dominant motion vector in the Cartesian coordinate system obtained based on the unimodal HOFO normalized by the number of components involved in the dominant bin. If the HOFO for the current frame is not unimodal, the last saved dominant motion vector from the unimodal HOFO is used. Note that the prediction of the next position of a person, based on an output of the Kalman filter, is used only when the YOLOv3 detector fails.

There are three tracking exceptions: i) multiple overlaps among the bounding box or boxes of the tracked person or persons and the detected bounding box(es) (Tracking Algorithm, STEP 3); ii) unassigned detections (Tracking Algorithm, STEP 4); iii) unassigned tracked person(s) (Tracking Algorithm, STEP 5). If there are multiple overlaps among bounding boxes (from a frame F_{k-1}) of the tracked person(s) and the detected bounding boxes in the frame F_k, the Hungarian algorithm is used. If the value of the intersection over union (IoU) of the overlapping bounding boxes (tracked and detected) is below the predetermined threshold, then these detections and active trackers are not entries in the Hungarian algorithm. The cost function used in the Hungarian algorithm consists of the weighted sum of the three components: normalized distances of the centres of the bounding boxes, 1.0 - IoU, and the difference of the HOFO pdfs between tracked and detected bounding boxes. Fig. 4 illustrates a tracking exception situation. Table 1 gives corresponding entries and values of the Hungarian matrix.

TABLE 1. HUNGARIAN MATRIX FOR TRACKING EXCEPTION

Detections / **Active trackers**	10	13
3	0.1081347 (0.048, 0.179, 0.117)	0.49367195 (0.578, 0.234, 0.640)
14	0.54492705 (0.663, 0.239, 0.692)	0.12865521 (0.059, 0.251, 0.098)

Table 1. Hungarian matrix for tracking exception depicted in Fig. 4. Values in brackets are corresponding components of the cost function (0.4 · (normalized distances of the centres of the bounding boxes), 0.3 · (1.0 – IoU), and 0.3 · (the difference of the HOFO pdfs), respectively). Detection 13 will be assigned to tracker 3 and detection 10 to tracker 14, which correspond to the minimum values of the entries of the Hungarian matrix. Note that the Hungarian matrix is 2 x 2 because the remaining 12 pairs of detection-tracker have IoU values below the threshold (0.4).

In the case of unassigned trackers (Tracking Algorithm, STEP 5), where the bounding box of an active tracker does not overlap with any of the detections, use the Kalman filter prediction to update the position, and increment a fail counter *fc*. The parameter *fc* represents the number of consecutive frames in which the position of the person is predicted without detection. If the *fc* exceeds the predefined threshold, the tracker is blocked, i.e. the person is not visible.

Fig 4. An illustration of the tracking exception situation i): a) YOLOv3 detections in the frame F_k (marked with light green colour); b) the bounding boxes of active trackers in the frame F_{k-1} (marked with yellow and purple colours); c) overlapping of detections in F_k with trackers in F_{k-1}.

The fourth stage is trajectory generation. The trajectory for each tracked person consists of the set of tracklets (x_k, y_k, *id*), where x_k and y_k are coordinates of the bounding box centre, and *id* is the identity number of the tracked person. The problem of YOLOv3 false positive detection is preliminarily solved in the postprocessing phase: if the trajectory of an object is shorter than the predefined threshold, it is neglected.

B. A knowledge-based crowd behaviour classifier

A knowledge-based crowd behaviour classifier consists of the following components:

i) a common-sense knowledge base, where the human-like descriptions of characteristic crowd behaviour are stored. This base is used for the definition of both fuzzy predicates for crowd motion patterns and fuzzy logic functions for crowd behaviour recognition;

ii) a motion pattern detector module, in which the fuzzy predicates for crowd motion patterns at microscopic level are evaluated. The fuzzy predicates are divided into two classes: fuzzy predicates related to the motion pattern for a person, and fuzzy predicates for a group of people in a crowd. Note that the assignment functions for the fuzzy predicates are determined based on expert knowledge and/or common-sense knowledge obtained by observing the real video sequences and/or sequences obtained by simulators of crowds with characteristic motion patterns [22, 23]. Fig. 5 depicts the taxonomy of the fuzzy predicates for motion patterns at the microscopic level. The outputs of the motion pattern detector module are the motion pattern classes with the values of the corresponding fuzzy predicates. The fuzzy predicates for motion patterns for a person and motion patterns for a group of people in a crowd are used for building fuzzy functions suitable for making inferences about complex crowd behaviour patterns. The fuzzy logic functions are application specific and their truth values are obtained with the composition of the fuzzy predicates and fuzzy logic operators [14, 24].

iii) a crowd behaviour recognition module contains the fuzzy logic functions which describe the classes of crowd behaviour, and a module for evaluation of the fuzzy logic functions. The outputs of the knowledge-based crowd behaviour classifier are the values of the fuzzy logic functions for every crowd behaviour class at every frame for a video sequence. The crowd behaviour class with the maximum value of the corresponding fuzzy logic function value is a candidate for the final crowd behaviour classification. In the case of binary classification (normal behaviour vs. abnormal behaviour), classification into abnormal behaviour is based

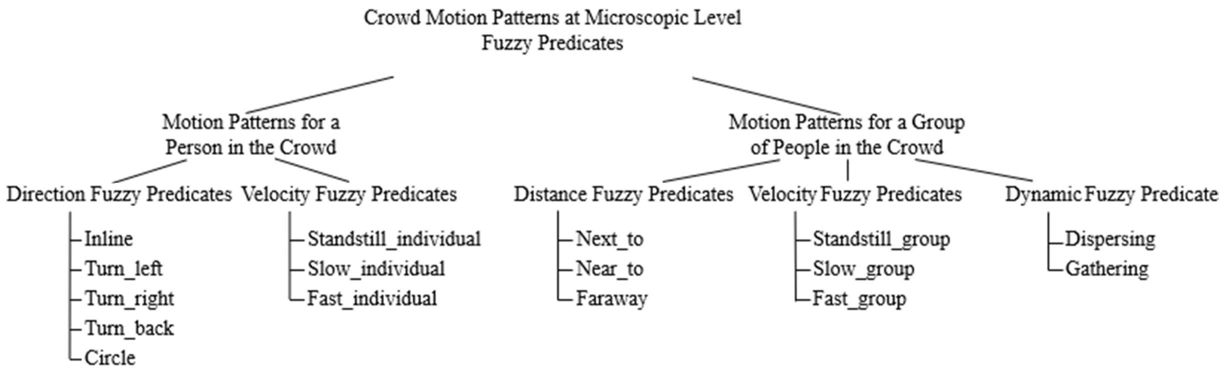

Fig 5. A taxonomy of the fuzzy predicates for motion patterns at the microscopic level

on a predefined threshold: if the value of the fuzzy logic function exceeds the threshold, then an abnormal behaviour pattern is detected.

1) *Crowd motion pattern detection*

To classify crowd motion patterns at the microscopic level, which is suitable for low-density crowds where the movement of each person or small group of people is concerned, we use two types of fuzzy predicates: fuzzy predicates for the motion patterns of a person, and fuzzy predicates for the motion patterns of a group of people (Fig. 5). The fuzzy predicates are based on the trajectory of a person and a set of trajectories of the tracked people, respectively.

In general, a fuzzy predicate is defined as a mapping (assignment function): O × T × R → [0,1], where O is a set of $n \geq 1$ tracked individual(s), T is a set of k time points, and R is a set of n trajectories of individuals. The assignment functions are determined based on human common-sense knowledge and an interpretation of the ground truth annotations of a training set of video sequences [14].

For example, the truth value estimation steps of the procedure for the group dynamic fuzzy predicate Dispersing(O, t_j, R) is presented in the Dispersing Algorithm.

Dispersing Algorithm

INPUT: n position points for the time point t_{j-m} and n position points for the time point t_j from the set R:

$$((x_1^{j-m+1}, y_1^{j-m+1}), \ldots, (x_n^{j-m+1}, y_n^{j-m+1}), \ldots, (x_1^j, y_1^j), \ldots, (x_n^j, y_n^j)),$$

where n is the number of individuals in a group. The m is constant, and it is experimentally determined based on a learning video dataset. It defines a time window (m frames in the past, starting from the current frame with index j).

STEP 1: At the time point t_{j-m} for the set of corresponding n position points

$$((x_i^{j-m}, y_i^{j-m}))_{i=1}^n$$

the average value of $n(n-1)/2$ distances (in pixels) between all possible point pairs is determined.

The same procedure is performed at time point t_j for corresponding n position points $((x_i^j, y_i^j))_{i=1}^n$.

STEP 2: The difference of the average value of distances for the time point t_j and the average value of distances for the time point t_{j-m} is calculated.

STEP 3: The estimated truth value of the fuzzy predicate Dispersing(O, t_j, R) is obtained by mapping the difference of the average value of distances to an interval [0,1] with an experimentally determined assignment function (Figure 6.).

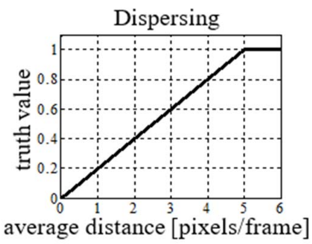

Fig. 6. Assignment function for the fuzzy predicate Dispersing(O, t_j, R).

The truth value estimation procedure for the fuzzy predicate Gathering(O, t_j, R) (which is used for bottleneck behaviour) is the same as that for the Dispersing(O, t_j, R) predicate, except that in STEP 3 an appropriate experimentally determined function is used.

2) *Common-sense knowledge database*

In the context of common-sense knowledge, by observing specific video sequences [25] and/or multi-agent simulators for crowds [22, 23], scenarios of anomalous crowd behaviour, such as runaway, bottleneck, barrier breaking scenarios, a person rushing through a group of standing group of people (with or without a collision), crowd or large group waves and crowd merging can be described. Due to the limited space of this paper, we illustrate only two scenarios which are characteristic of crowd behaviour:

i. Runaway:

   A runaway occurs when, at the beginning, several individuals that are close to each other (i.e. they form a group of people) suddenly start to run away, the velocity of each individual increases, the group disperses, and the individuals become more and more distant.

ii. Bottleneck:

   A bottleneck occurs when, at the beginning, several individuals are near each other and moving fast in the same direction. With the passing of time, the individuals become closer and closer to each other, and they move slowly or stand still.

Based on the above common-sense knowledge descriptions of the above scenarios, the following fuzzy logic functions are defined:

Runaway(t_j) = Near_to($O,t_{(j-w)},R$) ∧ Fast_group(O,t_j,R) ∧ Dispersing(O,t_j,R) ∧ Faraway(O,t_j,R), where $j > w$.
Parameter w defines a time point in the past (the number of frames) which is determined by means of expert knowledge and training video sequences.

Bottleneck(t_j) = Near_to(O,t_j,R) ∧ Gathering(O,t_j,R) ∧ Fast_group($O,t_{(j-w)},R$) ∧ [Slow_group(O,t_j,R) ∨ Standstill_group(O,t_j,R)], where $j > w$, and ∧ is a conjunctor which corresponds to the minimum value of fuzzy predicates.

Note that Dispersing(O,t_j,R) and Gathering(O,t_j,R) are the dynamic fuzzy predicates from the set of fuzzy predicates for a group of people, while Near_to($O,t_{(j-w)},R$) is from the set of distance fuzzy predicates, and Fast_group(O,t_j,R), Slow_group(O,t_j,R), and Standstill_group(O,t_j,R) are from the set of velocity fuzzy predicates.The fuzzy predicates Near_to($O,t_{(j-w)},R$), Fast_group(O,t_j,R) and Faraway(O,t_j,R), as well as corresponding assignment functions, are given in [14].
By evaluating the fuzzy logic functions, searching for their maximum value and comparing them with the predetermined threshold value (see Section IV), abnormal behaviour can be detected in every frame in a video sequence.

IV. Experimental setup and preliminary results

The preliminary testing of the proposed system for the detection of abnormal crowd behaviour is performed on the crowd video dataset UMN [25]. The dataset displays a group of people that move at normal walking speed in different directions and abnormal behaviour consisting of people running away from a scene. In this case, there is a binary classification: normal and abnormal behaviour (runaway). If the value of the fuzzy logic function Runaway exceeds the predefined threshold, then a runaway situation is detected.

Three experiments were performed.

Experiment 1- Ground truth person trajectory annotations and the crowd person behaviour recognition module are used to evaluate crowd behaviour recognition (i.e. ideal trajectories - no errors from the CNN-based detector stage or the remaining stages of the tracker).

Experiment 2 - Ground truth person detection annotations in combination with the remaining stages of the tracker and the crowd person behaviour recognition module are used (i.e.

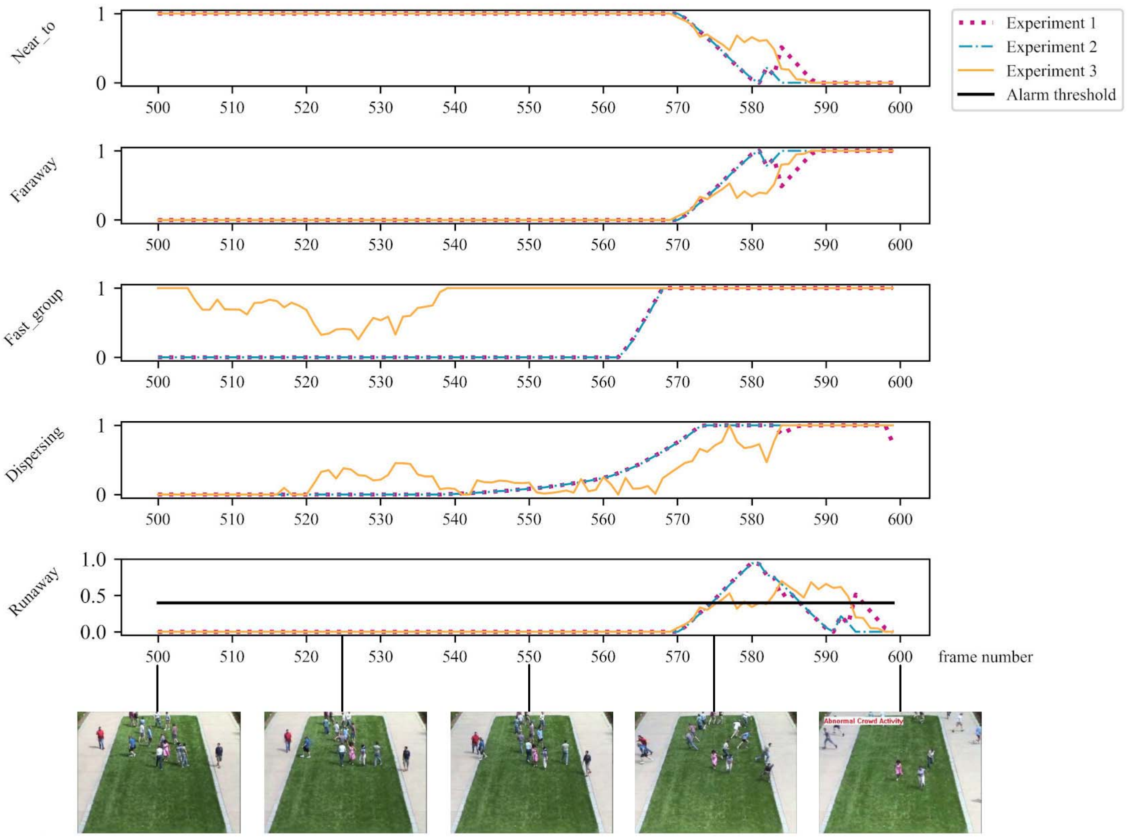

Fig. 7. Results of three experiments for Runaway behaviour

ideal detections - no errors from the CNN-based detector stage, but there are errors from the remaining stages of the tracker). The idea of this experiment is to evaluate the influence of the remaining stages of the tracker on the result of crowd behaviour recognition.

Experiment 3 - The 4-stage multi-person tracker and the crowd person behaviour recognition module are used to evaluate crowd behaviour recognition (i.e. realistic evaluation - errors of the CNN-based detector stage and the remaining stages of the tracker are present).

For all the above experiments, the following parameters are used: $fc = 10$, $m = 20$, $w = 10$ and the threshold is 0.4.

The aim of the first experiment is to evaluate the assignment functions of the fuzzy predicates and the fuzzy logic function for a runaway crowd behaviour pattern, as well as to test the threshold value defined by an expert for the runaway fuzzy function, i.e. the runaway is recognized for the frames where the value of the function exceeds the threshold. The above testing scheme is used to evaluate crowd behaviour recognition in the context of: i) ground truth trajectories; ii) ground truth detections; and iii) the proposed multi-person tracker. The results of the three experiments are shown in Fig. 7.

The first four rows in Fig. 7 depict the values of the fuzzy predicates (Near_to, Faraway, Fast_group, and Dispersing) and the fifth row depicts the values of the Runaway fuzzy function. Additionally, in the fifth row the threshold value 0.4 and the frames for which the Runaway fuzzy function has values above the threshold are marked. Each row depicts corresponding values obtained in all three experiments: a dotted line for Experiment 1, a dot-dashed line for Experiment 2, and a solid line for Experiment 3.

The first experiment with ground truth trajectories results in the most satisfactory outputs and an almost perfect match with the ground truth annotations of abnormal (i.e. runaway) crowd behaviour (beginning in frame 588 in the UMN dataset; Fig 7). The second two experiments show that the output of the system (i.e. crowd behaviour recognition) is robust to errors of the detector stage and remaining stages in the TBD tracker. The results of Experiment 3 show that the beginning of Runaway behaviour is detected in frame 575 (Fig. 8) which corresponds to the ground truth annotation.

Significant deviations from ground truth annotations for some fuzzy predicates (especially for the Fast_group predicate for frame 500 to frame 545) indicate that YOLOv3 detections (precision 0.9675 and recall 0.884 for a training video (subset of the UMN)) have a negative impact on the tracker which further induces variations in the fuzzy predicates and the fuzzy function output values.

V. Conclusion

An approach to motion pattern detection and abnormal crowd behaviour recognition in surveillance videos at the microscopic level is presented. It is based on an evaluation of fuzzy predicates and fuzzy logic functions defined based on common-sense knowledge and/or human interpretation of real video sequences and (multi-agent) simulators for crowds. Through the evaluation of fuzzy logic predicates for the motion patterns of a person or a group of people, motion patterns are detected and classified according to the proposed taxonomy of fuzzy logic predicates. Fuzzy logic functions are used for the detection and classification of anomalous

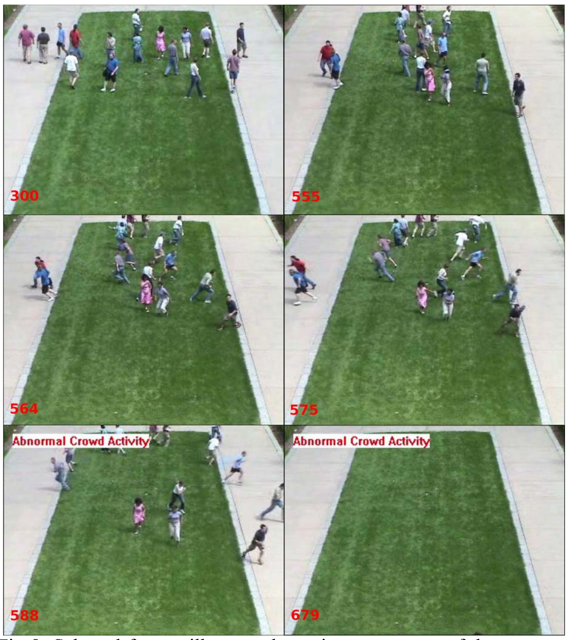

Fig 8. Selected frames illustrate the active occurrence of the runaway event; frame 575 – the beginning of the runaway event detected by the proposed approach; frame 588 – the beginning of the runaway event (annotated in the UMN dataset).

behaviour of a crowd. The building blocks of the fuzzy logic functions are the fuzzy predicates for which the assignment functions are determined based on an expert interpretation of training video sequences, connected by fuzzy logic operators [4]. The proposed approach is evaluated on simulated crowd events [4], ground truth annotations of real video sequences, and real trajectories obtained by the proposed 4-pipelined multi-person tracker. The preliminary experiments show promising and encouraging results.

Research in the near future will be oriented to: i) improving the CNN-based detector of the multi-person tracker by increasing its precision and recall; ii) extending the proposed approach to the multiclass recognition problem; iii) exhaustive testing of the proposed approach on video sequences of real crowd scenes; iv) developing heuristic procedures to determine the regions of interest and to identify multiple crowd behaviours in one scene.

Acknowledgment

This work has been supported by the Croatian Science Foundation under project IP-2018-01-7619 “Knowledge-based Approach to Crowd Analysis in Video Surveillance” (KACAVIS).

References

[1] S. Ali, K. Nishino, D. Manocha, and M. Shah, Modeling, “Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective”, In: Ali S., Nishino K., Manocha D., Shah M. (eds.) Modeling, Simulation and Visual Analysis of Crowds. The International Series in Video Computing, Vol 11, 2013, Springer, New York

[2] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, “Crowded Scene Analysis: A Survey”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 25, No. 3, March 2015, pp. 367-386.

[3] H. Y. Swathi, G. Shivakumar, and H. S. Mohana, “Crowd Behavior Analysis: A Survey”, in Proc. of the 2017 Int. Conference on Recent Advances in Electronics and Communication Technology, 2017, pp. 169-178.

[4] D. Marčetić, L. Maleš and S. Ribarić, “Crowd Motion Pattern Detection at the Microscopic Level”, Proceed. of the MIPRO 2019 conference, Opatija, 2019, pp. 1-5.

[5] X. Zhang, Q. Yu, H. Yu, “Physics Inspired Methods for Crowd Video Surveillance and Analysis: A Survey”, IEEE Access, November 2018, pp.66816-88830.

[6] F. Santoro, S. Pedro, Z. Tan and T. B. Moeslund, "Crowd analysis by using optical flow and density based clustering," 2010 18th European Signal Processing Conference, Aalborg, 2010, pp. 269-273.

[7] B. Solmaz, B. E. Moore, and M. Shah, “Identifying Behaviors in Crowd Scenes Using Stability Analysis for Dynamical Systems”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, Issue: 10, Oct. 2012, pp. 2064- 070.

[8] W. Ge, R. T. Collins, R. B. Ruback, “Vision-Based Analysis of Small Groups in Pedestrian Crowds”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, Vol.34, No. 5, pp. 1003-1016.

[9] P. Jodoin, Y. Benezeth and Y. Wang, "Meta-tracking for video scene understanding," 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, Krakow, 2013, pp. 1-6.

[10] T. Wang and H. Snoussi, "Detection of Abnormal Visual Events via Global Optical Flow Orientation Histogram," in IEEE Transactions on Information Forensics and Security, vol. 9, no. 6, June 2014, pp. 988-998.

[11] D. Shantaiya, K. Verma, K. Mehta, “Multiple object tracking using Kalman filter and optical flow”. European Journal of Advances in Engineering and Technology, 2015, 2, pp. 34-39.

[12] R. V. H. M. Colque, C. A. C. Júnior and W. R. Schwartz, "Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos," 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, 2015, pp. 126-133.

[13] T. Yang, C. Cappelle, Y. Ruichek, M. El Bagdouri,”Online multi-object tracking combining optical flow and compressive tracking in Markov decision process”, Journal of Visual Communication and Image Representation, Volume 58, 2019, pp 178-186.

[14] D. Marčetić, S. Ribarić, “A Fuzzy Logic-based Approach to Detection of Abnormal Crowd Behaviour”, Proceed. of the IEEE 61st ELMAR conference, Zadar, 2019, pp.1-4.

[15] J. Shi, Crowd management for large-scale outdoor events: multi-agent based modelling and simulation of a crowd behaviors, Doctoral Dissertation, The Chinese University of Hong Kong, 2006.

[16] Bandini S., Manenti L., Manzoni S. and Sartori F. “A Knowledge-based Approach to Crowd Classification”. In: Peacock R., Kuligowski E., Averill J. (eds) Pedestrian and Evacuation Dynamics. Springer, Boston, MA, 2011, pp 515-525.

[17] J. Redmon, A, Farhadi, “YOLOv3: An Incremental Improvement”, Computer Vision and Pattern Recognition, arXiv.org > cs > arXiv:1804.02767, 2018.

[18] GitHub. (2019). qqwweee/keras-yolo3. [online] Available at: https://github.com/qqwweee/keras-yolo3 [Accessed 8 Jul. 2019].

[19] G. Farnebäck, “Two-Frame Motion Estimation Based on Polynomial Expansion”. In: Bigun J., Gustavsson T. (eds) Image Analysis. SCIA 2003. Lecture Notes in Computer Science, vol 2749. Springer, Berlin, Heidelberg, 2003.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

[21] D. Shantiya, K. Verma, and K. Mehta, “Multiple object tracking using Kalman filter and optical flow", European Journal of Advances in Engineering and Technology. 2015, pp. 34-39.

[22] T. Weiss, A. Litteneker, C. Jiang, and D. Terzopoulus, “Position-Based Multi-agent Dynamics for Real-Time Crowd Simulation”, Proceeding of the SCA '17 Proceedings of the SCA '17 ACM SIGGRAPH / Eurographics Symposium on Computer AnimationLos Angeles, California — July 28 - 30, 2017; arXiv:1802.02673v2 [cs.GR] 20 Feb 2018. pp. 1-8.

[23] S. Kim, S.J. Guy, B. Zafar, A. Gutub, D. Manocha, “Velocity-Based Modeling of Physical Interactions in Dense Crowds”, The Visual Computer 31(5), June 2014, DOI: 10.1007/s00371-014-0946-1, 10 pages.

[24] I. Perfilieva, “Fuzzy Function: Theoretical and Practical Point of View”, Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-11), https://doi.org/10.2991/eusflat.2011.142, August 2011, pp. 480-486.

[25] UMN dataset. http://mha.cs.umn.edu/proj_events.shtml