International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 383–391

Using Feed-Forward Neural Networks for Data Association on Multi-Object Tracking Tasks

Uwe Jaenen, Carsten Grenz, Christian Paul and Joerg Haehner

Leibniz Universität Hannover, Institute of Systems Engineering - System and Computer Architecture,
Appelstrasse 4, Hannover 30167, Germany
{*jaenen, grenz, paul, haehner*}*@sra.uni-hannover.de*

Abstract: **This article presents an approach for data association in single camera, multi-object tracking scenarios using feed-forward neural networks (FFNN). The challenges of data association are object occlusions and changing features which are used to describe objects during the process. The presented algorithm within this article can be applied to any kind of object which has to be tracked, e.g. persons and vehicles. This approach arises within a project to detect critical behavior of persons. Besides, person tracking is one of the most challenging scenarios. People have different velocities and often change the moving direction. In addition, a variety of occlusions are caused by the movement as a group. Also in most surveillance scenarios the illumination conditions are not optimal. The usage of a feed-forward neural network is a mostly new approach in this research field. The advantage is the lightweight computational complexity and the fixed termination time in contrast to recursive neural networks like Hopfield networks which are used for plot association during radar tracking. FFNN is a non-probabilistic approach in contrast to common algorithms within this filed. They deliver decisions not probability values. The handling of the FFNN output will be presented in this article. During the evaluation we will show that the developed approach is capable to handle completely different scenarios like tracking people moving mostly straight forward but also complex scenarios like a soccer game.**

Keywords: single camera, multi-object tracking, data association, feed-forward, neural network

I. Introduction

This article deals with the data association problem (DAP) in multi-target single-camera tracking scenarios. Generally, tracking is used to extract trajectories of objects. The applications differ from tracking merchandises in controlled industrial environments, traffic analysis [1] up to sophisticated scenarios like pattern recognition on trajectory data or on selected images [2] to analyze behavior. This work arose in the context of the latter application. In automated and adaptive surveillance scenarios, recorded trajectories can be used to reconfigure camera systems e.g. the change of field of views. Therefore special demands arise on tracking algorithms. Multi-object tracking is a challenge of object detection and consistent labeling (data association) of objects. Due to the specialization of object detection like the histogram of oriented gradients detector [3] or sophisticated techniques using infrared images [4], it is necessary to divide detection and labeling. The challenges of DAP are on object occlusions and changing features which are used to describe objects during the process. The focus of this article is on a generic approach for DAP. Besides the development in a project with pattern recognition on people trajectories, we also choose person tracking because it is one of the most difficult applications. The objects have different velocities (pedestrians strolling through a mall) which changes over the time. Also, objects have different dimensions which make distance estimation on object size delicate. Using object-color is sophisticated because of the not optimal illumination conditions in most surveillance scenarios. In addition, a variety of occlusions are caused by the movement as a group of people. The Usage of feed-forward neural networks (FFNNs) is a mostly new approach on this field. The advantage is the light-weighted computational complexity and the fixed termination time in contrast to recursive neural networks (NN) like Hopfield networks. FFNNs in data association are a non-probabilistic approach. They provide a decision. The data association bases on these results. Therefore different realizations are possible as shown in the further sections.

The remainder of the paper is structured as follows: In Section II, we will investigate related work from the field of object tracking. In Sec. III to Sec. IV the architecture of the label distributor is deduced including the artificial neural network architecture. The robustness of the approach is demonstrated using two videos from a mall, two videos from the PETS2001 data set and a soccer game in the evaluation. The paper concludes with a summary and an outlook on future work.

II. State of the Art

Common object tracking algorithms can be differentiated in methods which interconnect detection and data association to a unit, so that these parts cannot be divided e.g. CamShift, and those approaches which use separate operations for object detection and data association (DA). The algorithm presented within this article, is a DA algorithm for the latter

class of object tracking approaches. The advantage of algorithms interconnecting detection and labeling is, that both can directly benefit from each other. The close coupling of both parts may speed up algorithms. The disadvantage is that most of these algorithms track a moving object without knowing what kind of object it is. From our point of view the interdependency of detection and labeling impedes the separate improvement of both. Based on this class of DA algorithms, it can be distinguished between methods which interpret the scene entirely (joint association) and methods based on the individual trajectory of objects (non-joint association). Often also the differentiation of probabilistic and non-probabilistic algorithm can be found. This classification is concurrent to the upper differentiation and also differs between joint and non-joint association. Our approach can be classified as a non-joint, non-probabilistic data association algorithm. The classification is depicted in Fig. 1. A

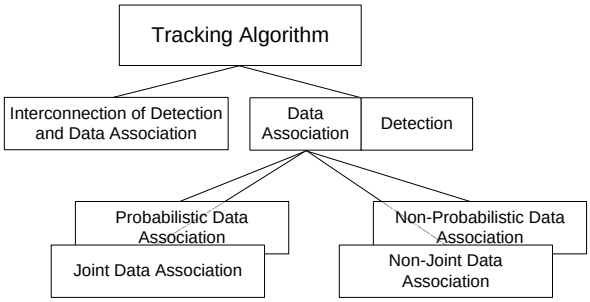

Figure. 1: Apportionment of different tracking-solutions.

well known algorithm for labeling objects using probability is the also so called Probabilistic Data Association (PDA) algorithm [5]. In PDA all candidates for association to a track are combined in a single statistically most probable solution. PDA premises that only one of the candidates is a trackable object, other candidates are clutter. A common usage scenario for PDA is plot association for radar tracker [6]. Joint Probabilistic Data Association (JPDA) [7] is an extension of the PDA algorithm which considers that more than one of the candidates is an object. The JPDA is tainted with the stigma of being computationally very intensive. The number of trackable objects has to be known a-priori [8], respectively all possible combinations have to been proven. This means the consideration of the assumption that all measurements are clutter up to all measurements are objects. So the computational effort increases very fast. An improvement of this was offered by the usage of neural networks. The problem of consistent labeling has similarities to the traveling salesman problem (TSP) which Hopfield tried to solve by neural networks [9]. In these Hopfield networks each neuron is connected to each other neuron within the network, so it is fully meshed. Our algorithm is based on a (progressive) feed-forward neural network architecture without recursion. Hopfield networks have to converge because of the recursion. The FFNN computing time is fix and depends only on the number of neurons. Hopfield networks perform well on tracking particle in a clutter environment using the position and the assumption of a nearly constant velocity. Because we want to track objects like people we decided to use additional information like color and object size to enhance the performance of the tracking result.

In recent years, several studies have already been made on object tracking algorithms using an artificial neural network (ANN).In [10] a camera is controlled by an ANN. In the first image an object is selected. This image is called master-image. The next frames from the camera are called slave-images. The master- and slave-image are compared pixel-wise by calculating the differences of normalized greyscale-values, gradient magnitude and gradient orientation. These values are relayed to a feed-forward neural network. The ANN provides information whether the camera should move left, stay, or move right to follow the object. The master-image is updated during the tracking process using the slave-image. This approach is a sophisticated template matching, whereas we work on only relevant feature data for multi-object tracking. In [11] the authors use a trained neural network. The system performs a foreground/background segmentation. The pixel values of the blobs in foreground are fed into a neural network, which outputs the probability of the blob being a person. Hypotheses are stated to describe the scene in a graph-structure. Each node represents an object with probability (outcome of the detection), size, position and appearance. Any connection between the nodes respectively the probability of connectivity is calculated from the weighted sum of similarity/density of the position, size and appearance. The hypotheses management calculates the likelihood of hypotheses and reduces their quantity to a limited number. The tracking result is passed to the detector in order to advance the detection property. The ramification of hypotheses models increases strongly with increasing number of objects which makes it practically not realizable in scenes with many objects.

III. Object Labeling

As described in Sec. I object tracking is a challenge of object detection and consistent object labeling. We propose a generic approach for object labeling where the object detection is exchangeable. During the label assignment process the algorithm holds two types of objects. The actual objects, which are located by the object detector in the actual frame, are denoted with o_t^d. Objects which have passed the labeling process and are labeled with a unique identification number, are held in the memory and are denoted with o^l. Each detected object $o_t^d \in O_t^d$ in the image at time t will be marked with a label $label(o^l, t)$ with $o^l \in O^l$ as labeled object. It is our goal to supply every object with a consistent label during the runtime T_{seq}.

$$\forall\, o^l \in O^l, t_1, t_2 \in T_{seq} : label(o^l, t_1) = label(o^l, t_2) \quad (1)$$

For the task of assigning labels to objects, we introduce the label distributor as depicted in Fig. 2. The architecture of the label distributor is designed with regard to consider the scalability. The scalability is related to the number of objects which have been detected in the scene o_t^d and the objects in the memory o^l. An ANN, which takes all labeled objects o^l and all actual objects o_t^d as input and delivers the correct assignment for each object, is not manageable due to the varying number of objects o^l and o_t^d. Thus, we chose an architecture where the ANN compares each actual object o_t^d with each labeled one. A collator will aggregate the results of the ANN.

A. Architecture of Object Labeling

The object detector is assumed to detect the objects present in the actual frames. The features describing the objects are extracted afterwards. The object labeling consists of a pre-processing element, an ANN and a collator, see Fig. 2. The pre-processing element prepares the actual object features and the features of labeled objects so that they can be handled by the ANN. The ANN compares the object features of

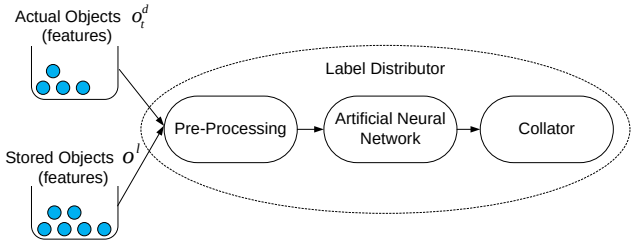

Figure. 2: Architecture of the label distributor, consisting of a pre-processing element, an ANN and a collator.

each o^l and o_t^d and evaluates the affinity of actual objects o_t^d to tracked objects o^l. The Collator aggregate the results of the ANN.

B. Feature Extraction

The label distributor uses low level features, because we want to design a generic label distributor which is independent from a particular detector type. It can be assumed that the following features are retrievable by almost every detector type:

- position $\vec{p_t}$ in frame coordinate system (fcs)
- object dimensions d_t (width, height) in fcs
- color histogram in HSV color space (hue, saturation, value)

The region on the object, which is used to create the histogram should contain mostly pixel of the object in each perspective. While a person is pictured at a sufficient size, a representative color histogram can be extracted from the upper-body, see Fig. 3. In cases with small objects it is reasonable to choose the whole object size to calculate a representative color histogram. The background in these cases conglomerate additional context information. In applications with occlusions and with a low object detection rate, using only the position is not sufficient. For this purpose the color of the object is an adequate supplemental information. In addition, the object dimensions are useful, because it is related to the object distance to the camera.
Known objects o^l have a history of features which can be considered as a First In - First Out Memory. Each time an object has been recognized, the actual features are stored. The history has to be set to a reasonable size N. Pre-Evaluations showed that a history of 10 slots per object provides good results. Additionally, known objects o^l have a time-to-live (TTL). The TTL is used to cleanup objects which have not been detected for a defined time. If the object o^l has been detected in the actual frame, the TTL is set to a maximum

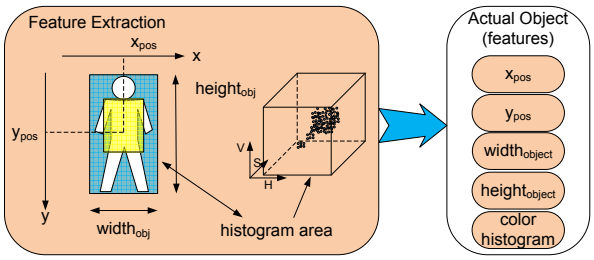

Figure. 3: The object dimensions, position and the color histogram are sufficient to describe an object. If the objects are pictured in a sufficient size, the marked upper-body (yellow) contains enough color values to create a sufficiently representative color-histogram. Alternatively the whole object size (blue) can be used. The background in these cases conglomerate additional context information.

value TTL_{max}. Every time an object o^l cannot be assigned to a detected object o_t^d, the TTL will be decremented. When reaching zero, the object will be removed from the list of known objects O^l.

C. Pre-Processing

The features of actual detected objects and the features of already known (tracked) objects have to be pre-processed in such way that they can be handled by the ANN. It makes sense to operate on comparison metrics of actual and known feature values, which means the difference of the positions, dimensions and the normalized correlation coefficient of the color-histograms.
There are various reasons why objects can be missed by the detector, caused by occlusions or detection failures. So the feature values of o^l can be outdated compared to feature values of actual objects o_t^d. The history is used to smooth the feature values.

Position Using the object history, the object position of o^l is estimated by linear interpolation (time-base is frame-count). The prediction is useful, because if an object has not been detected for some frames, the position will be interpolated in this way.

$$\forall\, o^l \in O^l : \vec{p}_{t+1}(o^l) = \vec{p}_t(o^l) + \vec{v}_t(o^l) \cdot \triangle t \qquad (2)$$

with

$$\vec{v}_t(o^l) = \frac{1}{N-1} \sum_{n=1}^{N-1} \frac{\vec{p}_{n+1}(o^l) - \vec{p}_n(o^l)}{t_{n+1} - t_n} \qquad (3)$$

The predicted position of object o^l compared to the position of the detected object o_t^d, in relation to the maximum assumable deviation $p_{const} \in \{X_{const}, Y_{const}\}$, is used as input to the ANN.

$$\triangle_p = \begin{cases} \dfrac{\triangle'_p}{p_{const}}, & \text{if } \triangle'_p < p_{const} \\ 1, & \text{else} \end{cases} \qquad (4)$$

Because the linear prediction is very sensitive, the distance from the last known position is used as $\triangle_p$ if the history N is not filled.

Dimension A prediction of the dimension in general is not possible. The values vary frame by frame because of the inaccuracy of most detectors. The detected object dimensions alternate in a deterministic non-descriptive way. This makes it reasonable to use the average.

$$\forall\, o^l \in O^l : \bar{d}(o^l) = \frac{1}{N}\sum_{n=1}^{N} d_n(o^l)$$

with

$$d_n(o^l) \in \{width(o^l, n), height(o^l, n)\} \tag{5}$$

The dimensions of object o^l compared to the dimensions of the detected object o_t^d, in relation to the maximum assumable deviation d_{const} (see Eq. 6), is used as input to the ANN.

$$\triangle_d = \begin{cases} \dfrac{\triangle'_d}{d_{const}}, & \text{if } \triangle'_d < d_{const} \\ 1, & \text{else} \end{cases} \tag{6}$$

To compare the color histograms, the normalized correlation coefficient ρ is used. The actual histogram is compared to each stored histogram. This approach is more effective than using only the last stored color-histogram. Between the color-histogram of the actual detected object and the last stored color-histogram of o^l can be a gap of several frames, because of missing detections by the detector. Also disturbances like light variations due to the missed detections can be smoothed, by using the average.

$$\triangle\rho = \bar{\rho}(o^l) = \frac{1}{N}\sum_{n=1}^{N} \rho_n(o^l) \tag{7}$$

The average correlation of the color-histograms of object o^l and the detected object o_t^d is used as input to the ANN.

In the last step of pre-processing, the range of $\triangle$'s has to be adopted to the range of the used artificial neuronal network library. We implemented the network using the FANN library [12]. Because we used the symmetrical tangent hyperbolic sigmoid function as activation function we transformed the $\triangle$'s input-values into a [-1, 1] range, with -1 for bad match and 1 for good match.

$$\triangle_{feature} \quad \xrightarrow{range} \quad e_{feature} \tag{8}$$

IV. Architecture of the Artificial Neural Network

Artificial neural networks are part of the artificial intelligence. In our application, we have chosen a feed-forward neuronal network. The FFNN consists of neurons which can be divided into three classes: input neurons, hidden neurons and output neurons.
This is depicted in Fig. 4. The connecting lines represent the weighted link of an output gate to the input gate of a neuron of the following layer. The output value O is defined by an activation function A. We chose the symmetrically tangent hyperbolic sigmoid-function for the hidden and output neurons as proposed in [13]. The parameters of A_j of a neuron j is the weighted sum of the output e_i of a neuron i of the former layer. Additional a bias Θ_j has to be considered. A formal description is depicted by Eq. 9.

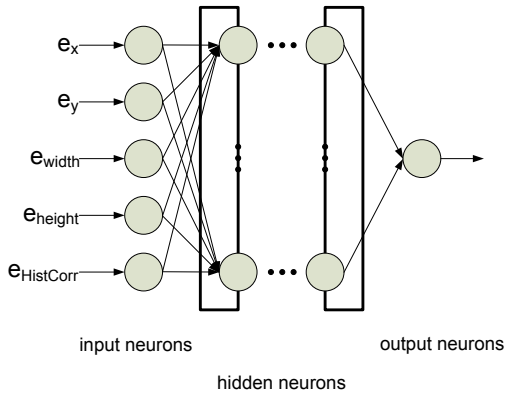

Figure. 4: Architecture of a feed-forward neuronal network with regard to the presented approach.

$$O_j = A_j\left(-\Theta_j + \sum_i w_{j,i}\cdot e_{j,i}\right) \tag{9}$$

During the training the training-algorithm explores the weighting values $w_{j,i}$ of each connection and the bias values Θ_j to get the maximum performance. The parameters and the performance constitute a fitness landscape. Common training algorithms are able to find local maxima within these landscapes. There are many training algorithms for disposal, e.g. Backpropagation, Resilient Propagation, Quick Propagation etc. During the training process we considered that these algorithms did not have a significant effect on the quality of the results of the training, rather on the time required for training. The training of neuronal networks is a trial and error process based on experience. We selected Backpropagation as online variant, which means the network will be adapted after each training datum. The training is done on three self-made videos. Two of them display a floor with two persons. The other one shows a mall. These videos include difficult scenes with persons who move through the camera's field of view (FoV) in different directions and temporary occlusions. This ensured that many difficult situations could be presented to the network such as occlusions, position jumps, as well as objects which are positioned very close. The training-dataset consists of comparisons of all detected objects with all known objects. The used three self-made videos deliver approx. 10.000 entries to the dataset.

A. Evaluation of Different Architectures of Feed-Forward Neural Networks

In the following annotations, the layer will separated by "-" and the hidden layer will be marked by "[ ]". Several architecture models for the ANN were analyzed differing in the number of hidden layers and the neurons per layer. For the evaluation of the chosen architectures we used two benchmark videos [14] and [15] which display a scene of a parking area. The different network architectures which we evaluated are listed on the left in Tab. 1.
Tab. 1. The corresponding number of failures for each architecture is listed on the right. The networks 5-[8]-1, 5-1

ANN	Number of Failures in video [14]	Number of Failures in video [15]
5-1	11	3
5-[8]-1	9	3
5-[15]-1	17	5
5-[3]-1	12	7
5-[5]-[2]-1	31	14

Table 1: Different network types depicted against their number of failures in video [14] and [15]. The used parameters are listed in Tab. 2.

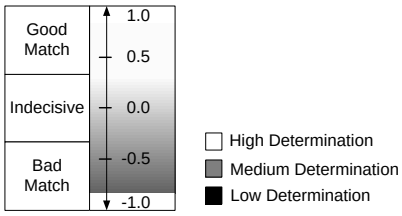

Figure. 5: The result of the FFNN can be classified into three rough different credibility levels

and 5-[15]-1 provide the highest accuracy. Network 5-1 is a good indicator for a performance estimation of this approach. The architecture without hidden-layer is only capable to use linear separation for classification. But during the training it is easier to find the maximum performance, because the fitness landscape is only determined by six parameter (five input weights and one bias), whether the fitness landscape of the 5-[8]-1 network is determined by 49 parameters. The fact that the 5-1 network can achieve better results than 5-[8]-1 in other videos shows that the performance of 5-[8]-1 could be increased even further by training. The result of network 5-[15]-1 shows that an increasing number of hidden neurons does not correlate with an increasingly correct classification rate. That one hidden layer is sufficient, corresponds to the design recommendations for continuous functions in [13]. We chose network 5-[8]-1 for the following evaluations.

V. The Collator

The ANN calculates based on the input $e_{feature}$ the decision if a detected object o_t^d has similarity to a known object o^l. Each detected object will be compared with each known object in the memory which TTL has not expired. The result is a two-dimension ranking matrix R of assignments $r_{o_t^d,o^l} \in R$.

$$o^l \in \{o^l \in O^l | TTL > 0\}, o_t^d \xrightarrow{ANN} r_{o_t^d,o^l} \qquad (10)$$

An example for a ranking matrix is R_1, see Equation 11. The rows are the detected objects and the columns are the known objects. In this example the system knows three objects and detected four. The fourth object e.g. could have currently entered the scene. A high similarity is represented by a '1' and a mismatch by '-1', so the decision-margin of the trained artificial neural network is binary.

$$R_1 = \begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix} \qquad R_2 = \begin{bmatrix} +1 & +1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \\ -1 & -1 & -1 \end{bmatrix} \qquad (11)$$

Because of the symmetrically tangent hyperbolic sigmoid-function the output of the FFNN is not binary in that strict way. The results of the neural network can be interpreted as determination, which means that they can draw conclusions about the "credibility".
The credibility indicates the degree of trust, which can be assumed if an association is correct in the ranking matrix. It should be pointed out again that the results of the neural network do not reflect absolute probability-values. Rather, it can be assessed based on these values if an association $r_{o_t^d,o^l}$ is more likely than a comparable assignment. The credibility-validation includes the elements of the ranking matrix and whether there are further assignments in a row or column of the matrix, as shown in R_2. A credibility matrix can be calculated based on the following equation 12.

$$c(r_{o_t^d,o^l}) = \frac{r_{o_t^d,o^l}}{2 - r^{max}_{o_t^d,o^l}} \qquad (12)$$

with $r^{max}_{o_t^d,o^l}$ as the maximum value (greater than zero) of the considered row respectively column of the ranking matrix excluding the considered $r_{o_t^d,o^l}$ itself. This credibility matrix can be included as a quality measure in the behavior analysis of people, which is one research goal of the project described in the introduction. Evaluations based on these credibility matrices have shown that the decision based solely on the FFNN results are sufficient and that the inclusion of the credibility has no further effort on the assignment decision. They rather assess the risk of a label permutation. Based on these pre-evaluations the assignment algorithm is structured as follows. For the assignment, the collator takes the best-hit $max\{r_{o_t^d,o^l}\}$ in the ranking matrix R, after that the next best-hit and so on of the matrix R. Actual objects which cannot be assigned will be included as new objects. An object cannot be assigned if the number of new objects is greater than the number of known objects, or if the result of the ANN is lower than a defined threshold $threshold_{rmin}$. If formerly known objects cannot be assigned, their TTL will be decreased.

VI. Evaluation

In this section the robustness of the algorithm will be evaluated. We used five benchmark videos which cover different complex scenarios. The first two videos show a mall in Portugal with two different scenes, from the CAVIAR project. The challenge is the view on the scene. The videos were recorded along a longitudinal corridor, so many occlusions arise. Additional two videos from the PETS2001 dataset were used. This videos show the same scene from different views. It shows a wide area with moving objects like cars and people. The most sophisticated benchmark video shows a soccer game. The video includes many object occlusions and fast changes of the moving directions. In Figure 6 a snapshot of each video is depicted.
To evaluate the videos it is necessary to use the ground truth data for object detection. This makes the results of the data association discussible to similar approaches. The ground

Figure. 6: The figures show from left to right the benchmark videos Mall1 [16], Mall2 [17], PETS1 [14], PETS2 [15] and the soccer game [18].

truth data also contains objects which cannot be seen by the camera, e.g. they are occluded. Therefore we wanted to evaluate the videos at real constraints, the objects which cannot be detected by a HOG-Detector [3] were not used. The goal of the algorithm is to continuously assign labels to corresponding objects, see Equation 1. A change of the label is unwanted. A failure of the label distributor is defined as a change of the label of an object while it is present in the scene. E.g. if the label of an object changes and the label will be reassigned to the object after a few frames it will cause two failures.
There are several parameters which affect on the algorithms performance. One of these parameters is the $threshold_{r_{min}}$. Output values of the FFNN below this threshold will not be considered. This parameter describes the minimum determination of a correct assignment. Other important parameters are the time-to-live TTL, the history size N which determines the maximum amount of features which can be stored of each object o^l.

A. *Exemplary Result*

To demonstrate the capacity of this algorithm we selected a scene of the mall video [16] exemplarily. During the first 300 frames two persons revolve around each other, see Fig. 7, and the labels were assigned correctly.

B. *Dependency on Detection Rate*

The data association depends on the object detection. Because we used the ground-truth data we are able to evaluate the performance on several detection rates. In Fig. 8 the amount of false assignments of the label distributor are depicted on a varying probabilistic detection rate decreasing from 100% to 20%. The probability is assumed to be uniform. Therefore we used a Mersenne Twister implementation. A detection rate of e.g. 20% does mean that on average a object is detected every fifth frame, but because of the probabilistic uniform distribution the gap between detetections can be much longer. Each measurement was repeated ten times. We used the parameters listed in Tab. 2. Fig. 8 shows the results of the mall videos. The list above represents the failure ratio to all assignments. These videos contains many occlusions because of the longitudinal viewing direction to the corridor. It is evident, that even at a detection rate of 20% the failure ratio is lower than 1.6%. In Fig. 9 the results of the parking scenes (PETS2001) are depicted. These videos are challenging because of the bad illumination conditions. At a detection rate lower than 50% the correct assignment of the people in the dark regions becomes difficult. The persons revolve around each other and at a low detection rate this cannot be recognized. Fig. 10 shows the results of the soccer game benchmark video. It is the most challenging video, because the objects sharply change their movement direction. But even at a detection rate of 50% the failure ratio is lower than 0.5%. These results show that the failure ratio of false assignments about all benchmark videos is even at a detection rate of 40% much lower than 1.0%.

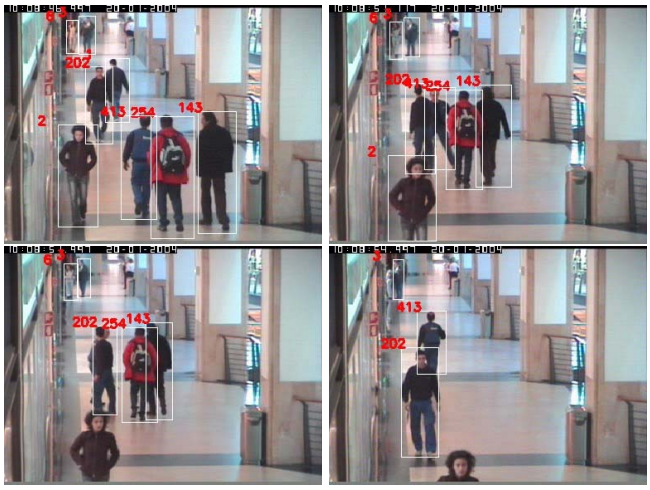

Figure. 7: The person with the label 413 will be occluded by person 202, because of a revolve. In the lower right figure both persons 202 and 413 kept their labels.

Parameter	Value
$threshold_{r_{min}}$	0.0
N	10
TTL	75
X_{const}	100
Y_{const}	75
d_{const}	1

Table 2: Parameters used within the evaluation.

C. *Dependency on a Minimum Recognition Amount*

In most applications the recorded trajectories are used to accomplish further analysis on these data. In huge observation scenarios those systems cannot be administrated manually.

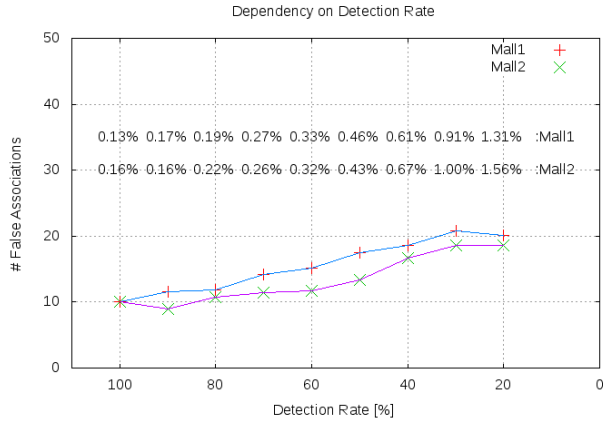

Figure. 8: Amount of failures of the mall videos. The list above represents the failure ratio to all assignments.

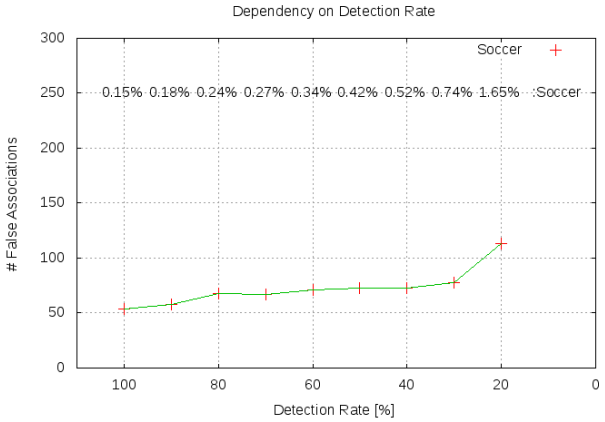

Figure. 10: Amount of failures on the soccer video. The list above represents the failure ratio to all assignments.

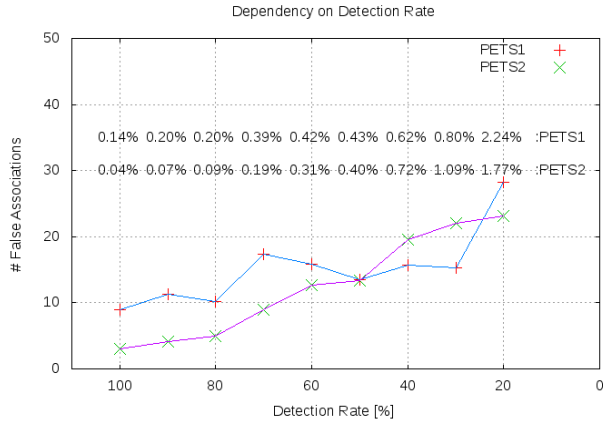

Figure. 9: Amount of failures of the PETS2001 benchmark videos. The list above represents the failure ratio to all assignments.

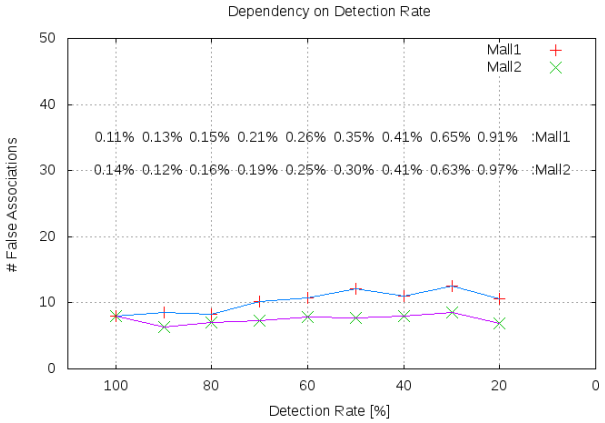

Figure. 11: Amount of failures of the mall videos. The list above represents the failure ratio to all assignments.

Current research is towards automated surveillance camera networks. These networks of smart cameras use the trajectory data for self-configuration. Those systems will assume, that a new object has entered the scene if the algorithm looses an object and assigns it with a new label. Therefore it could be necessary to report objects only if the object has been recognized for a minimum amount h. In Fig. 11, 12, 13 the amount of failures within the benchmark videos is depicted over the detection rate using a minimum h of 25. It is shown that the amount of failures is lower than using non minimum recognition length. Even in the soccer video the amount of failures has been halved at a detection rate of 20%. It is significant that the amount of failures does not increase with a decreasing detection rate. This is an advantage for a selforganzing surveillance system.

D. Dependency on the Determination Threshold

The output values of the FFNN are set into relation to a minimum determination threshold $threshold_{r_{min}}$. To set $threshold_{r_{min}} = 0.0$ is only one solution. In Fig.14 the failure rate is depicted over the determination rate and the minimum determination threshold in the soccer game example. It is evident that the system performs best with low thresholds, around 0.0 This is caused by the training of the FFNN. Most of the training data are false-positive matches. Using -1.0 as threshold respectively non threshold is not an option because this causes a lot of false associations.

VII. Conclusion and Outlook

In this article we presented a new approach for solving the data association problem using feed-forward networks. FFNNs have an advantage in computational speed compared to Hopfield networks. The termination time is fixed by the amount of neurons and hidden layer within the network. We have shown that the combination of error-prone object describing features as input to the feed-forward network is sufficient to reach a high association rate. In the evaluation the robustness of the algorithm was analyzed using five benchmark videos with different scenes and different complexity. We could show that the label distributor even works well with low detection rates. The evaluation on a minimum determination threshold on the output of the FFNN demonstrated that this threshold depends on the detection rate, but even at a detection rate of 40% the correct association rate is higher than 99%. In a scenario with a object detector which can ensure no detection failures the threshold can reduced to a very low level. In future work we want to include this algorithm

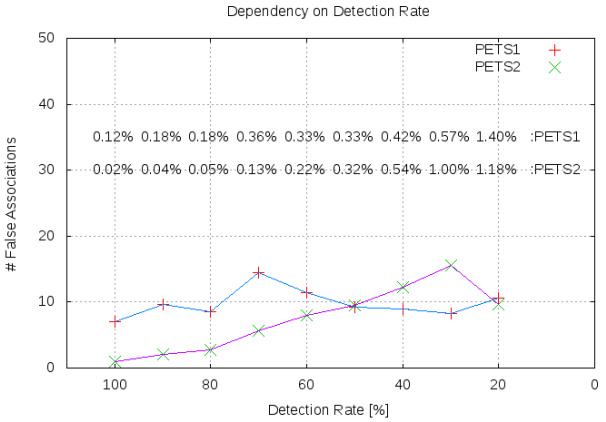

Figure. 12: Amount of failures of the PETS2001 benchmark videos. The list above represents the failure ratio to all assignments.

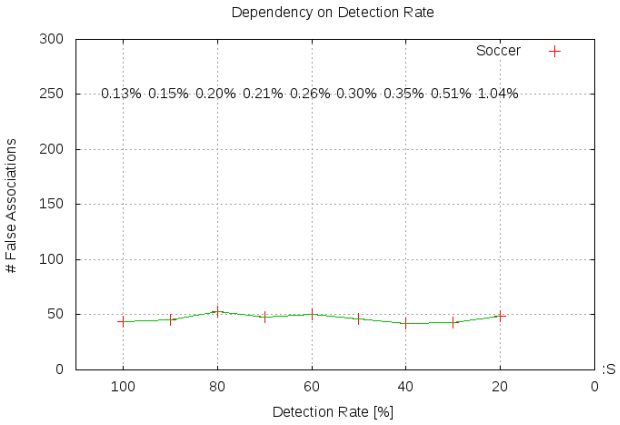

Figure. 13: Amount of failures on the soccer video. The list above represents the failure ratio to all assignments.

into an automated surveillance system. The system will reconfigure itself by using the trajectory data. This will require a detailed analysis of the credibility measurement which was introduced in section V.

VIII. Acknowledgement

The work of Uwe Jaenen has been founded by the German Research Foundation (DFG) within the project "QTrajectories". Carsten Grenz has been founded by the Federal Ministry of Education and Research (BMBF) within the "CamInSens" project.

References

[1] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 4, pp. 271 – 288, 1998.

[2] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "A fast statistical approach for human activity recognition," *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, vol. 4, pp. 334–340, 2012.

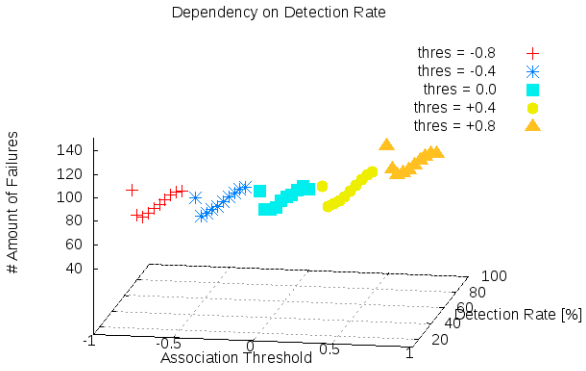

Figure. 14: Apportionment of different tracking-solutions.

[3] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *CVPR '06*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 1491–1498.

[4] M. Azarbad, A. Ebrahimzade, and V. Izadian, "Segmentation of infrared images and objectives detection using maximum entropy method based on the bee algorithm," *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, vol. 3, pp. 026–033, 2011.

[5] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 1, no. 1, pp. 451–460, 1975.

[6] H. Leung and M. Blanchette, "Data association for multiple target tracking using hopfield neural network," in *Speech, Image Processing and Neural Networks, 1994. Proceedings, ISSIPNN '94., 1994 International Symposium on*, apr 1994, pp. 280–283 vol.1.

[7] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, pp. 173–184, 1983.

[8] D. Musicki and R. Evans, "Joint integrated probabilistic data association - jipda," in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 2, 2002, pp. 1120–1125 vol.2.

[9] J. Hopfield and D. Tank, "Neural computation of decisions in optimization problems," *Biological Cybemetics*, vol. 52, pp. 141–152, 1985.

[10] B. Resko, P. Szemes, P. Korondi, P. Baranyi, and H. Hashimoto, "Artificial neural network based object tracking," *SICE -ANNUAL CONFERENCE-*, vol. 2, pp. 1398–1403, 2004.

[11] M. Han, A. Sethi, W. Hua, and Y. Gong, "A detection-based multiple object tracking method," 2004, pp. V: 3065–3068.

[12] S. Nissen, "Fast artificial neural network library (fann)," http://leenissen.dk/fann/index.php, 2010.

[13] T. M. Mitchell, *Machine Learning*. Mcgraw-Hill Higher Education, 1997, chapter 4.6.2 Representational Power of Feedforward Networks, page:105.

[14] "Benchmark-video-parking-area 1: Dataset 1," http://www.cvg.rdg.ac.uk/PETS2001/, 2011.

[15] "Benchmark-video-parking-area 2: Dataset 2," http://www.cvg.rdg.ac.uk/PETS2001/, 2011.

[16] "Benchmark-video-mall: Twoentershop2cor.mpg," groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/, 2010.

[17] "Benchmark-video-mall: Threepastshop2cor.mpg," groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/, 2010.

[18] "Benchmark-video-soccer-game: Test datasets of camera 3," http://www.cvg.cs.rdg.ac.uk/VSPETS/vspets-db.html, 2010.

Author Biographies

Uwe Jaenen was born in 1980 in Meppen (Germany). He received the Dipl.-Ing.(FH) in electrical engineering from the University of Applied Science Osnabrück in 2004, and the M.Sc. in electrical engineering from the Leibniz Universität Hannover (LUH) in 2007. After gaining experience in industry, he currently is Ph.D. student at the Institute of Systems Engineering at the Department of System and Computer Architecture at the LUH. His research focuses on distributed smart camera systems including computer vision and job scheduling.

Carsten Grenz was born in 1983 in Hannover (Germany). He received his M.Sc. degree in computer science from the Leibniz Universität Hannover in 2010. Currently, he works as Ph.D. student at the Institute of Systems Engineering (Department of System and Computer Architecture). Continuing his work in the field of mobile ad-hoc networks in smart camera systems, his research focuses on distributed data management algorithms in heterogeneous smart sensor systems.

Christian Paul was born in 1980 in Hannover (Germany). He received the M.Sc. in computer science from the Leibniz Universität Hannover (LUH) in 2009. After gaining experience as assistant at the Institute of Systems Engineering at the Department of System and Computer Architecture at the LUH he is now working in industry. His research focused on computer vision.

Joerg Haehner received the M.Sc. degree in computer science from the Darmstadt University of Technology, Darmstadt, Germany, in 2001 and the Dr. rer. nat. degree in computer science from the Universitt Stuttgart, Stuttgart, Germany, in 2006. He worked in the area of data management in mobile ad-hoc networks (MANETs) and in 2006, was appointed Assistant Professor in the System and Computer Architecture Group at Leibniz Universitt Hannover, Hannover, Germany. His research focuses on architectures and algorithms in the field of organic computing (e.g., distributed smart camera systems, mobile ad-hoc and sensor networks, and global scale peer-to-peer systems).