Robust False Positive Detection for Real-Time Multi-target Tracking

Henrik Brauer, Christos Grecos, and Kai von Luck

University of the West of Scotland, UK
University of Applied Sciences Hamburg, Germany
`Henrik.Brauer@HAW-Hamburg.de`

Abstract. We present a real-time multi-target tracking system that effectively deals with false positive detections. In order to achieve this, we build a novel motion model that treats false positives on background objects and false positives on foreground objects such as shoulders or bags separately. In addition we train a new head detector based on the Aggregated Channel Features (ACF) detector and propose a schema that includes the identification of true positives with the data association instead of using the internal decision-making process of the detector. Through several experiments, we show that our system is superior to previous work.

1 Introduction

Due to the availability of inexpensive cameras and high demand of home and public monitoring systems for applications such as surveillance, smart environments and ambient assisted living, visual surveillance is becoming a more and more important research topic in computer vision. The aims of such systems are to recognise humans and to build trajectories when they move through the scene. The data can then be used for further processing such as decision-making.

In this paper we adopt Markov Chain Monte Carlo Data Association (MCMCDA) to estimate a varying number of trajectories given a set of detections extracted from a video sequence. We focus on head detection, because heads are rarely obscured from overhead surveillance cameras. We describe a novel false positive model that allows us to filter out false positive detections in the background as well as false positive detections on foreground objects, e.g. head detections on other body parts such as shoulders or bags. The algorithm runs in real-time on high definition (1920x1080/25fps) cameras on a standard computer without the need to use a GPU.

Multi-target tracking has been an active research area for many years. Our work is based on MCMCDA [7,9,6,5,2], which was first introduced for tracking a single or fixed number of targets [7]. Later, MCMCDA was adapted specifically for visual tracking by associating object detections resulting from background subtraction [9] and a boosted Haar classifier cascade [6]. Ge and Collins [5] further developed this approach by using not only object detections but also

Fig. 1. Results from the work of Benfold and Reid¹. Left: the raw detections with a true positive, a background and a foreground false positive. Right: the results after data association; the algorithm was able to identify the background false positive as a false positive but not the foreground false positive.

tracklets, which were created by using a standard tracking algorithm for a short period after each detection.

In the most recent work, Benfold and Reid [2] combine asynchronous Histogram of Oriented Gradients (HOG) detections with simultaneous Kanade-Lucas-Tomas (KLT) tracking in an accurate real-time algorithm. In addition, they present a novel approach for false positive detection by creating a separate model for false positives and combining the identification of false positives with data association. In our work, we also use an asynchronous detection step as well as a separate false positive model; however, we make several improvements on the work of Benfold and Reid.

The first contribution involves the treatment of false positive detections. False positives are a frequent problem in multi-target tracking. They occur either on background objects or as part of a foreground object. False positives in background regions are stationary and often repeatedly occur in the same position. Benfold and Reid [2] have shown that such false positives can be filtered out by creating a separate model for false positives and then combining the identification of false positives with the data association. However, different to background false positives, foreground false positives are the result of incorrect detections of foreground objects, such as incorrect head detections on other body parts such as shoulders or bags.

We noticed that such incorrect detections have the same motion model as true positives and therefore can't be detected by the model of Benfold and Reid (see Fig. 1). However, we also noticed, that even when the motion is the same, they often have properties which label them as false positives. In order to filter out foreground false positives, we expand the approach of Benfold and Reid by creating separate models for background and foreground false positive rather than background false positives only.

The next contribution involves detection. We train a new detector based on the Aggregated Channel Features (ACF) detector [4] and propose a schema that includes the identification of true positives with data association instead of using the internal decision making process of the detector.

¹ http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009babenfold_headpose/project.html

The last contribution is the evaluation of the algorithm on the town centre benchmark [2] using the standard CLEAR MOT [3] evaluation criteria.

2 Multi-target Tracking

In agreement with the majority of recent multi-target tracking methods, e.g. [6,9,5,2], we pursue tracking by detection. Targets (here, pedestrians) are separated from the background in a preprocessing step and form a set of target hypotheses, which are then used to infer the targets trajectories. We thus run a novel sliding window head detector, based on the Aggregated Channel Features (ACF) detector. To estimate the location of pedestrians in the current frame and to ensure that data associations can be made correctly, a tracker is initialised to track the relative motion for a period d (in our case $d = 75$ frames). In order to achieve real-time performance we use a multi-threaded approach, in which one thread produces asynchronous detections, while a second thread applies the tracking algorithm and a third thread performs data association.

2.1 Data Association

Assuming we have a set of detections $D = \{D_1, D_{1+\delta t}, \dots, D_\tau\}$ in the time interval $[1, \tau]$, where $D_t = \{d_{t1}, d_{t2}, \dots, d_{tn}\}$ are the detections obtained at the frame t . For each detection, a tracker is initialised to track the relative motion for a period d (in our case $d = 75$ frames). Our aim is to find the hypothesis, H_i , that divides the detections into a set of target trajectories $T = \{T_1, T_2, \dots, T_j\}$, so that each trajectory contains all observations of a single person. In order to represent false positive trajectories, each trajectory has a type c_j that can take the values c_p for true positive, c_{f1} for foreground false positive and c_{f2} for background false positive trajectories. We constrain each observation to be associated with at most one trajectory, and only one detection can be associated to a trajectory at each time step.

The tracking problem is then formulated as a Bayesian problem and we then take the maximum a posteriori (MAP) estimator of the posterior distribution as the optimal solution for the hypothesis H :

$$H^* = \arg \max(p(H|D)) = \arg \max(p(D|H)p(H)) \quad (1)$$

where $p(D|H)$ is the likelihood function that models how well the hypothesis fits the detections and $p(H)$ expresses our prior knowledge about desirable properties of good trajectories.

In order to represent both background and foreground false positives, we extend the posterior distribution of [2] with a new likelihood function:

$$p(D|H_i) = \prod_{T_j \in H_i} \left[p(d_1^j | c_j) \prod_{d_n^j \in T_j / d_1^j} p(d_n^j | d_{n-1}^j, c_j) \right] \quad (2)$$

where d_n^j is the n th detection in a track T_j , with n indicating only the order within the track. We define the link probability between two detections as the product of four probabilities, namely the probability for size s , location x , motion m , and detection score r .

$$p(d_1^j | c_j) = p(s_1 | c_j) p(x_1 | c_j) p(m_1 | c_j) p(r_1 | c_j) \quad (3)$$

$$p(d_n^j | d_{n-1}^j, c_j) = p(s_n | s_{n-1}) p(x_n | x_{n-1}, c_j) p(m_n | c_j) p(r_n | c_j) \quad (4)$$

The proposed link probability is designed to represent the correct data association and track types and allows different to the like probability of [2] to make a distinction between true positives and foreground false positives. In the following, we explain each probability in detail.

2.2 Feature Extraction and Modelling

Detection Score. Object detection algorithms such as the Histogram of Oriented Gradients (HOG) based detection algorithm used in [2] utilise only the information present in a single frame to decide if a possible head candidate is a true positive. However, in our case we have the knowledge that each true detection is part of a track of true detection. This additional information can be used to increase the accuracy in each frame. We therefore propose instead using the internal decision-making process of the detector to explicitly include the identification of true positives with the data association in our scheme, assuming that the average detection score of a true positive track is higher than the average detection score of a false positive track. This has two advantages: first we are improving the recognition rate of false positives, and second we include true positive detections with low confidence which would otherwise be removed.

The general idea is as follows. We train a detector that returns all possible head candidates, even those with low confidence which are most likely false positives. Then we assign to each detection a probability that describes how certain we are that this detection is a true positive and include this probability in the data association process. We therefore train a new detector based on the Aggregated Channel Features (ACF) detector [4]. The ACF detector has shown higher accuracy in pedestrian detection and is faster than the HOG detector [2] (a performance test can be found in [4]). For example on our test machine the algorithm needs 0.3 seconds to process a high-definition (HD) image on the CPU whereas the HOG detector needs 2.25 seconds on the CPU or 1.2 seconds on a GPU.

As training set, we use the CAVIAR Head Pose Dataset [8], the dataset is composed of 21326 head examples of 50x50 pixels. We train our detector so that 99% of the true detections are detected without taking false positives into account. This results in a detector that returns almost all true positives and an acceptable number of false positives. We then model the probability of each detection to be a true positive by a sigmoid function:

Fig. 2. Left: Output ACF detector trained with default configuration, no false positive detection but two missed detections (marked with a circle). Right: Output ACF detector trained with proposed configuration, no missed detections but three false positive detections (marked with a circle).

$$f(r_n) = \frac{1}{1 + \exp\left(\frac{-(r_n - \mu_r)}{\sigma_r^2}\right)} \quad (5)$$

r_n is the confidence value provided by the ACF detector. The probability that a detection is a true positive detection is then:

$$p(r_n|c_p) = f(r_n) \quad (6)$$

and a false positive detection is:

$$p(r_n|c_{f1}) = 1 - f(r_n) \quad (7)$$

$$p(r_n|c_{f2}) = 1 - f(r_n) \quad (8)$$

Figure 2 shows examples of the detection results of a detector trained with default parameters and a detector trained as proposed. The first detector missed two heads; in contrast, the second detector detected all heads but also detected three false positives. However, since the false positive detections are part of a false positive track, they can be filtered out in the data association step.

Size. Benfold and Reid [2] assumed that the size of the first detection has a global prior log-normal distribution that is independent of the image location. However, in most scenarios we examined, the size of a true positive detection strongly correlates with the image location x . For example, in the town centre scene (see Fig. 3) the average head size of a person on the left side is 50.3 pixels whereas the average size on the right side is 18.7 pixels. We model this relation with a probability map that depends on the image location x_n, y_n and assume a normal distribution:

$$\ln(s_1) \sim N(\mu_{map}(x_1, y_1), \sigma_{map}^2(x_1, y_1)) \quad (9)$$

In contrast, for the foreground and background false positives it is assumed that the size is uniformly distributed over the set of possible sizes S :

$$p(s_1) = \frac{1}{|S|} \quad (10)$$

Fig. 3. Left: Mean detection size in the town centre scene summed over 100×100 pixel blocks, in which it can be seen that the size strongly correlates with the location in the scene. Right: Entry map in the town centre video.

For the following detections in the track the size is then encoded by the ratio to the previous detection:

$$\ln \frac{s_n}{s_{n-1}} \Big| c_p \sim N(0, \delta_t \sigma_{sp}^2) \quad (11)$$

$$\ln \frac{s_n}{s_{n-1}} \Big| c_{f1} \sim N(0, \delta_t \sigma_{sf1}^2) \quad (12)$$

$$\ln \frac{s_n}{s_{n-1}} \Big| c_{f2} \sim N(0, \delta_t \sigma_{sf2}^2) \quad (13)$$

where δ_t is the time difference between the frames in which the detections were made.

Location. Recent approaches [5,2] have assumed that the locations of both pedestrians and false positives are uniformly distributed around the image; however, in the case of a stationary camera this is not true for pedestrians. A pedestrian always has to enter the scene at some point, and therefore the first detection has to be next to an entry point. In order to model this fact, we build an entry map (see Fig. 3). The probability that a track is a true positive track depends on the distance from the first detection x_n to the next entry point ep divided by the detection size s_1 :

$$\frac{\|x_1 - ep\|}{s_1} \Big| c_p \sim N(0, \sigma_p^2) \quad (14)$$

where $\|\cdot\|$ is the Euclidean distance.

For false positives, it is assumed that the location of the first detection is uniformly distributed around the image, therefore the probability density of x_1 on the image area α is relative to the object size s_1 in pixels:

$$p(x_1) = \frac{s_1^2}{\alpha} \quad (15)$$

For the following detections, the probability depends on the track type. For true positives and foreground false positives, the probability depends on the estimated location x_{est} of the previous detection x_{n-1} at the time t , where t is the time at which the following detections x_n were made. In order to estimate the location x_{est} we use the tracker that was proposed recently in [1], the tracker combines template matching and an adaptive Kalman filter and is able to deal with temporary occlusions. We assume a normal distribution:

$$||x_n - x_{est}|| \Big| c_p \sim N(0, \sigma_{tp}^2 + 2\sigma_d^2) \quad (16)$$

$$||x_n - x_{est}|| \Big| c_{f1} \sim N(0, \sigma_{fp1}^2 + 2\sigma_d^2) \quad (17)$$

where $2\sigma_d$ is an additional uncertainty that models the error in the two detection locations. Background false positives are the result of background objects, and therefore the location is assumed stationary:

$$||x_n - x_{n-1}|| \Big| c_{f2} \sim N(0, 2\sigma_d^2) \quad (18)$$

Motion Vector. As the last feature we use a motion vector histogram similar to [2], included to distinguish between background false positives which are expected to have no movement and true positives which are expected to have at least a small amount. The motion vector histogram has four bins with boundaries representing movement of $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$ pixels per frame where the motion vector is calculated from the result of the tracking in the first five frames immediately after the detection. A multinomial distribution is then used to model the probability:

$$m_n | c_p \sim Mult(m_p) \quad (19)$$

$$m_n | c_{f1} \sim Mult(m_{f1}) \quad (20)$$

$$m_n | c_{f2} \sim Mult(m_{f2}) \quad (21)$$

Evaluating the space of hypotheses is extremely challenging and a close form solution is usually not available in practise. We therefore use Markov Chain Monte Carlo Data Association (MCMCDA) to estimate the best hypotheses H^* in the same way as [2].

3 Experimental Setup and Evaluation

The model parameters we use in our test, such as μ_r and σ_r were learned automatically by interleaving the MCMCDA sampling with an additional Metropolis Hastings update, similar to the approach in [5]. The entry points for the entry map were manually defined. Since detections occur delayed and not in every frame, the current location is estimated with the tracker of the last detection in each track. All experiments use the town centre benchmark [2] (Fig. 4). The experiments are performed on a desktop computer with an Intel Core i5-3470 CPU with 3.2 GHz and 8GB RAM. All experimental results are shown in Table 1.

Fig. 4. Sample video frames from the town centre sequence

Table 1. Benchmark results

Exp.	Method	MOTA	MOTP	Prec.	Rec.	False Pos.	Missed
1	Benfold [2]	45.40%	50.80%	73.80%	71.00%	18374	20427
	Ours	81.55%	64.51%	90.87%	91.35%	6500	6127
2	Test A	74.61%	64.53%	91.29%	82.71%	5586	12247
	Test B	76.53%	64.35%	85.63%	92.73%	11021	5147

In the first experiment (1), we compare our tracking output to the results of [2]. Similar to them, we use four metrics to evaluate tracking performance, Multiple Object Tracking Precision (MOTP), Multiple Object Tracking Accuracy (MOTA) and detection precision and recall. The MOTA is a combined measure which takes into account false positives, false negatives and identity switches, and MOTP measures the precision with which objects are located using the intersection of the estimated region with the ground truth region (see [3] for details). In addition, we also include the number of false positives and the number of missed detections. The results show the advantage of our approach, it can be seen that we reduce the number of missed detections and simultaneously reduce the number of false positive detections.

Our next experiment (2) shows the impact of the different contributions for false positive detection. Therefore, two tests are done; test A illustrates the impact of including the identification of true positives with the data association by comparing the tracking accuracy of experiment (1) to the tracking accuracy if the score probability is not included and the ACF detector with default configuration (see Section 2.2) is used. The results show that the overall tracking accuracy drops, the number of missed detections increases from 6127 to 12247

and but the number of false positives drops slightly from 6500 to 5586. This is due to the fact the proposed detector, in contrast to the ACF detector with default configuration, is explicitly trained such that all possible head candidates get returned, even those with low confidence (see Section 2.2). As a consequence, the number of missed detections drops but also some additional false positives appear that could not be filtered in the data association step.

Test *B* shows the advantage of creating a separate model for background false positives by testing the proposed schema without background false positives. The resulting schema is then similar to the schema of [2]. In this test, the number of false positives increases from 6500 to 11021, possibly because foreground false positives get classified as true positives since they have the same motion model. Simultaneously the number of missed detections drops 6127 to 5147, probably because true positive tracks can't be erroneously classified as foreground false positives.

4 Conclusion

In this paper, we presented a real-time multi-target tracking algorithm that effectively deals with false positive detections. In order to achieve this, we build a novel motion model that treats false positives on background objects and false positives on foreground objects separately. The novel model makes the tracker robust against false positives and simultaneously reduces the number of missed detections. In addition, we train a new detector and propose a schema which includes the identification of true positives with the data association instead of using the internal decision-making process of the detector. Several experiments validated the arguments in this paper and showed that our model is superior to previous work.

References

1. Ben-Ari, R., Ben-Shahar, O.: A computationally efficient tracker with direct appearance-kinematic measure and adaptive Kalman filter. *Journal of Real-Time Image Processing* (2013)
2. Benfold, B., Reid, I.: Stable Multi-Target Tracking in Real-Time Surveillance Video. In: *Conference on Computer Vision and Pattern Recognition*, pp. 3457–3464 (2011)
3. Bernardin, K., Stiefelwagen, R.: Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *J. Image Video Process.*, 1:1–1:10 (2008)
4. Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2014)
5. Ge, W., Collins, R.: Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation. In: *British Machine Vision Conference* (2008)
6. Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H., Yang, B., Sun, L., Yang, S.: Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video. In: *British Machine Vision Conference* (2007)
7. Pasula, H., Russell, S., Ostland, M., Ritov, Y.: Tracking Many Objects with Many Sensors. In: *International Joint Conference on Artificial Intelligence*, pp. 1160–1167 (1999)

8. Tosato, D., Spera, M., Cristani, M., Murino, V.: Characterizing Humans on Riemannian Manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8), 1972–1984 (2013)
9. Yu, Q., Medioni, G.G., Cohen, I.: Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association. In: *Conference on Computer Vision and Pattern Recognition* (2007)