Sequential Sensor Fusion Combining Probability Hypothesis Density and Kernelized Correlation Filters for Multi-Object Tracking in Video Data

T. Kutschbach, E. Bochinski, V. Eiselein, T. Sikora
Communications Systems Group
Technische Universität Berlin
{kutschbach,bochinski,eiselein,sikora}@nue.tu-berlin.de

Abstract

This work applies the Gaussian Mixture Probability Hypothesis Density (GMPHD) Filter to multi-object tracking in video data. In order to take advantage of additional visual information, Kernelized Correlation Filters (KCF) are evaluated as a possible extension of the GMPHD tracking-by-detection scheme to enhance its performance. The baseline GMPHD filter and its extension are evaluated on the UA-DETRAC benchmark, showing that combining both methods leads to a higher recall and a better quality of object tracks to the cost of increased computational complexity and increased sensitivity to false-positives.

1. Introduction

Multi-object tracking is a challenging task in many applications, *e.g.* traffic monitoring or surveillance. The Gaussian Mixture Probability Hypothesis Density (GMPHD) Filter [8] has gained interest in the sonar / radar tracking community due to its low computational complexity and its ability to deal with high detection clutter. However, the constraints in computer vision applications differ from sonar / radar applications. Especially the comparatively low detection probability, due to occlusion or other visual disturbances, leads to problems with the pure tracking-by-detection scheme of the GMPHD. On the other hand, due to the availability of images, additional information is available and can be used to enhance its performance. Once the existence of an object is known, it can directly be relocalized by visual correlation. In recent years, starting from [2] to [3] and [6], the visual single-object tracking community has made significant advances, making it possible to robustly track single objects over a long time span based only on a single annotation in an initialization frame.

Based on a problem analysis, this work proposes a combination of GMPHD and Kernelized Correlation Filters (KCF), introduced in [6], by sequential multi-sensor fusion. Afterwards, the original and the extended version are evaluated on the UA-DETRAC benchmark. This evaluation includes some adjustments that have been made to comply with the requirements of the benchmark.

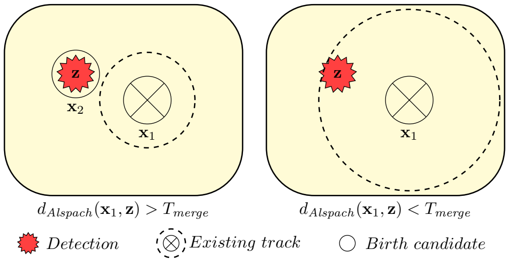

z
$\mathbf{x}_2$
$\mathbf{x}_1$
z
$\mathbf{x}_1$
$d_{Alspach}(\mathbf{x}_1, \mathbf{z}) > T_{merge}$
$d_{Alspach}(\mathbf{x}_1, \mathbf{z}) < T_{merge}$
Detection
Existing track
Birth candidate

Figure 1. Whether a detection $\mathbf{z}$ leads to an initialization of a new track or not, depends on the Alspach distances to the surrounding tracks (themselves depending on the covariances of their tracks). In the left example, the covariance of $\mathbf{x}_1$ is small, leading to the initialization of the new track $\mathbf{x}_2$. In the right example, the covariance of $\mathbf{x}_1$ is larger, such that no new track is initialized.

2. Multi-Object Tracking with a PHD Filter

The basis for the tracker in our work is a GMPHD filter which follows the tracking-by-detection paradigm and was presented in [4]. The PHD is the first statistical moment of a multi-target probability distribution of a random finite set (RFS) comprising potential multi-target states. It lives on the single-state space and assigns every point the probability for existence of a target in that point:

$$D(\mathbf{x}) = \sum_{n=0}^{\infty} \frac{1}{n!} \int p(\{\mathbf{x}, \mathbf{y}_1, ..., \mathbf{y}_n\}) d\mathbf{y}_1 ... d\mathbf{y}_n. \quad (1)$$

Summing the PHD gives an expectation value of the number of targets in a given area. Due to the underlying RFS-character, individual target identities are lost and target labels have to be obtained separately.

The GMPHD filter first proposed by [8] uses a set of Gaussian distributions in order to model the PHD:

$$D_k = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}(\mathbf{x}; \mu_k^{(i)}, C_k^{(i)}) \tag{2}$$

with D_k as the PHD at time k and $\mu_k^{(i)}, C_k^{(i)}, w_k^{(i)}$ as the mean, covariance and weight of the i-th Gaussian.

The GMPHD prediction step where new tracks are initialized and known tracks are propagated is formulated as

$$\begin{aligned} D_{k|k-1}(\mathbf{x}) &= b(\mathbf{x}) \qquad (3) \\ + \quad \sum_{j=1}^{J_{k-1}} p_S(\mathbf{x}) &\cdot w_{k-1}^{(i)} \cdot \mathcal{N}(\mathbf{x}; \mu_{S,k|k-1}^{(i)}, C_{S,k|k-1}^{(i)}) \end{aligned}$$

with $b(\mathbf{x})$ as the birth intensity and $\mu_{S,k|k-1}^{(i)}, C_{S,k|k-1}^{(i)}$ as the parameters for the i-th PHD component propagated using a linear motion model. $p_S(\mathbf{x})$ models a survival probability but is discarded in our implementation. $b(\mathbf{x})$ is a set containing a pre-defined Gaussian birth distribution D_{birth} in every position for which a detection has been received and which has a high distance to known tracks. This is the case if the Alspach distance [1] to all neighboring objects exceeds the threshold T_{merge}. The Alspach distance between $\mathbf{z}$ and $\mathbf{x}_i$ is the L2-norm of both, normalized by the covariance of $\mathbf{x}_i$. The covariance of a track thus highly impacts the initialization of other tracks (cf. Figure 1).

The update step then corrects the predicted Gaussians by taking into account the received measurement set Z_k:

$$\begin{aligned} D_{k|k}(\mathbf{x}) &= (1 - p_D) \cdot D_{k|k-1}(\mathbf{x}) \qquad (4) \\ + \sum_{\mathbf{z} \in Z_k} \sum_{j=1}^{J_{k-1}} & \frac{p_D(\mathbf{x}) \cdot L_z(\mathbf{x}) \cdot w_{k|k-1}^{(j)}}{\mathcal{C} + \sum_{l=1}^{J_{k-1}} p_D(\mathbf{x}) \cdot L_z^{(l)}(\mathbf{x}) \cdot w_{k|k-1}^{(l)}}. \end{aligned}$$

In this term, p_D is the detection probability of the sensor, L_z represents the likelihood for a given pair of detection and track and $\mathcal{C}$ is the sensor clutter. In order to maintain object labels, a label tree implementation [7] is used.

3. Visual Single-Object Tracking with Kernelized Correlation Filters

The visual tracking scheme used in this work is similar to the Fast Scale Space Tracking scheme of [3] and uses two separate models for estimating target translation and scale.

We use FHOG-features[5] from P. Dollár's computer vision toolbox[1], a Gaussian kernel-KCF for translation estimation and a linear kernel filter for scale estimation[2].

Initialization

Given the position and size of the target in the initial frame, FHOG features are extracted from a region of 2.5 times its size and weighted by a cosine window in order to highlight the target in the center and to avoid boundary issues. A circulant matrix is used to describe all possible shifts of the target from the initial base sample.

With $\boldsymbol{\alpha}$ as the coefficients to be learned and the discrete Fourier transform (DFT) $\mathfrak{F}$, the fast learning equation is

$$\hat{\boldsymbol{\alpha}} = \frac{\mathfrak{F}(y)}{\mathfrak{F}(k^{xx'}) + \lambda}, \tag{5}$$

where the division represents element-wise division and $\hat{()}$ is a DFT-transformed signal representation. $\boldsymbol{y}$ denotes the regression targets, *i.e*. Gaussian functions with their peaks at the cyclic shift of the corresponding training sample, slowly decaying to zero for other shifts. $\boldsymbol{k}^{\boldsymbol{x}\boldsymbol{x}'}$ denotes the kernel correlation function between two signals $\boldsymbol{x}$ and $\boldsymbol{x}'$. For Gaussian kernels, it is defined as

$$\boldsymbol{k}^{\boldsymbol{x}\boldsymbol{x}'} = e^{\left(-\frac{1}{\sigma^2}\left(\|\boldsymbol{x}\|^2 + \|\boldsymbol{x}'\|^2 - 2\,\mathfrak{F}^{-1}\left(\sum_c \hat{\boldsymbol{x}}_{\boldsymbol{c}}^* \odot \hat{\boldsymbol{x}}'_{\boldsymbol{c}}\right)\right)\right)}. \tag{6}$$

The symbol $\odot$ stands for element-wise multiplication and * for the complex-conjugate.

After the model $\boldsymbol{x}$ and coefficients $\boldsymbol{\alpha}$ for translation estimation have been obtained, the same is done for scale estimation, but using a linear kernel defined as

$$\boldsymbol{k}^{\boldsymbol{x}\boldsymbol{x}'} = \mathfrak{F}^{-1}\left(\sum_c \hat{\boldsymbol{x}}_{\boldsymbol{c}}^* \odot \hat{\boldsymbol{x}}'_{\boldsymbol{c}}\right). \tag{7}$$

Training samples for scale estimation are obtained as features of variable patch sizes with the target as center.

Update

Given the translation model $\tilde{\boldsymbol{x}}_{k-1}$ and coefficients $\tilde{\boldsymbol{\alpha}}_{k-1}$, learned during initialization or obtained by updates in the previous frame, the learned object can be found in the next image patch at time k by extracting the features $\boldsymbol{z}_k$ in this patch and calculating the full detection response $\boldsymbol{r}(\boldsymbol{z}_k)$, *i.e*. the output for each location in this patch:

$$\boldsymbol{r}(\boldsymbol{z}_k) = \mathfrak{F}^{-1}\left(\hat{\boldsymbol{k}}^{\tilde{\boldsymbol{x}}\boldsymbol{z}} \odot \hat{\tilde{\boldsymbol{\alpha}}}\right)\Big|_{\substack{\tilde{\boldsymbol{\alpha}} = \tilde{\boldsymbol{\alpha}}_{k-1} \\ \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_{k-1} \\ \boldsymbol{z} = \boldsymbol{z}_k}} \tag{8}$$

The position of the target can be found by a peak in this response matrix. As proposed in [2], the Peak to Sidelobe Ratio PSR $= \frac{r_{\max} - \mu_{s1}}{\sigma_{s1}}$ serves for detecting tracking failures, with $r_{\max}$ as the maximum value of the detection response $\boldsymbol{r}$ and μ_{s1}, σ_{s1} as mean and standard deviation of the sidelobe.

If the PSR is above a certain threshold, then the target is considered found. The model $\hat{\tilde{\boldsymbol{x}}}$ and coefficients $\hat{\tilde{\boldsymbol{\alpha}}}$ are updated by linear interpolation at the target position found. Scale estimation and model update is done analogously.

[1] https://github.com/pdollar/toolbox/

[2] https://github.com/klahaag/cf_tracking/

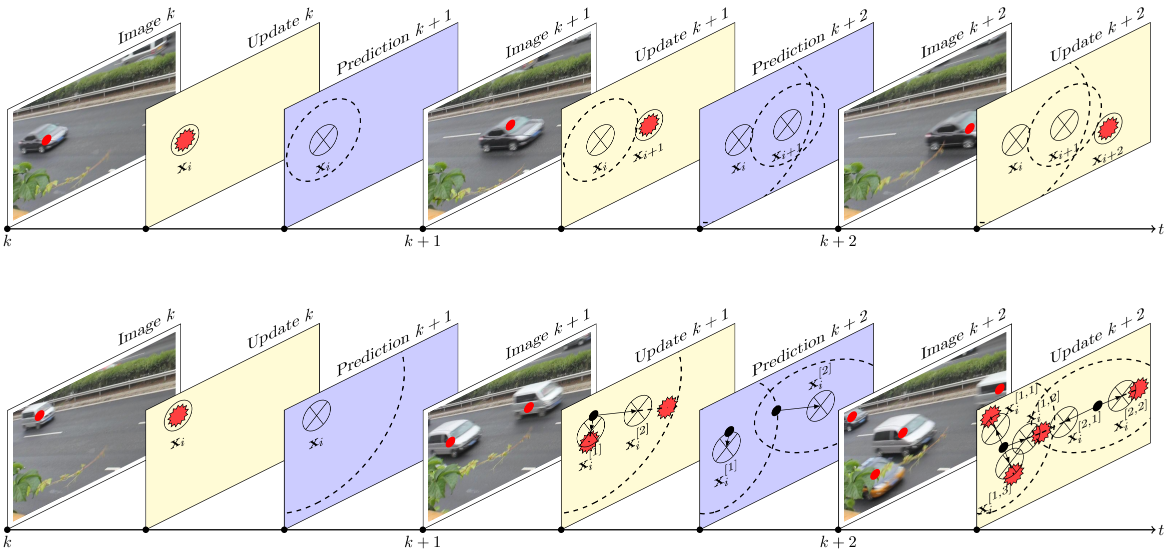


Figure 2. In scenarios with a high density and fast objects, track initialization is hampered in the GMPHD tracking scheme. Top: With the birth covariance chosen too small, the corresponding detections are not associated to each other. Bottom: With the birth covariance chosen properly for fast objects, the first track might suppress the initialization of other tracks in dense scenes (tracks denoted by $\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots$ while different hypotheses for the same track i are denoted as $\mathbf{x}_i^{[1]}, \mathbf{x}_i^{[2]}, \mathbf{x}_i^{[1,1]}, \mathbf{x}_i^{[2,1]}, \ldots$).

4. Enhanced Multi-Object Tracking with Combined GMPHD and Correlation Filters

The GMPHD filter is a very sophisticated method for tracking multiple objects, but it also shows some issues.

First, although it can handle missed detections, it is still prone to them. In case an object has not been detected, only the part with $(1-p_D)$ in equation (4) contributes to the new state estimation, leading of a divergence of the track covariance. In case detections of other object tracks are nearby, this divergence leads to confusion which detections belong to which object tracks, and finally results in ID-switches.

Second, in scenarios with very fast objects, the distance between the first detection of a new object and its detection in the next frame can be very large. Therefore, the birth covariance has to be set to a large value in every possible direction in order to associate the second detection with the track. If set too small, single detections will not be associated with each other, such that for every one of them a new track is initialized (cf. Figure 2 top).

However, in scenarios with fast objects and a high object density, the necessarily high birth covariance of a new track keeps not only the second detection of the desired object in its merging radius, but also the detections of other newly appeared objects nearby. The large birth covariance of the first track thus prevents the initialization of new tracks for following objects (cf. Figure 2 bottom). The different detections will be considered as multiple hypotheses tracked at the same time until the algorithm finally decides for one of them by a converging covariance. This leads to a delayed track extraction and lower tracking performance.

In order to resolve these issues, the main idea is to use visual correlation filters and their capability to rediscover previously known objects in subsequent frames as an additional information source. Naive inclusion of visual object trackers into a multi-object tracking scheme (*e.g.* simply adding the correlation tracker results to the general detection set Z_k) is dangerous, because it almost naturally makes the system prone to false-positive detections. Once a visual tracker is initialized on static false-positive detection, it tracks it forever, if not handled properly.

To avoid this problem, we propose to introduce an additional update step particularly for the visual tracking results right before the regular update step with regular detections. In this way, the correlation filter and the regular detector work as in sequential multi-sensor fusion. Because each instance of the correlation filter is dedicated to the specific object track that it has been initialized on, it can precisely correct its state prediction and sharpen its covariance. More precisely, for each detection, a dedicated instance of the correlation tracker is initialized. Each object track using a particular detection as a hypothesis in its label tree also holds a reference to the correlation tracker of the detection. After the prediction step and when the following image is received, this reference is used to perform a correlation filter update. If the target is found, the result is used to correct the previous state prediction using the standard Kalman filter update equations. In case the visual correlation tracker fails to find its target, nothing changes.

Afterwards, the regular update step with external detec-

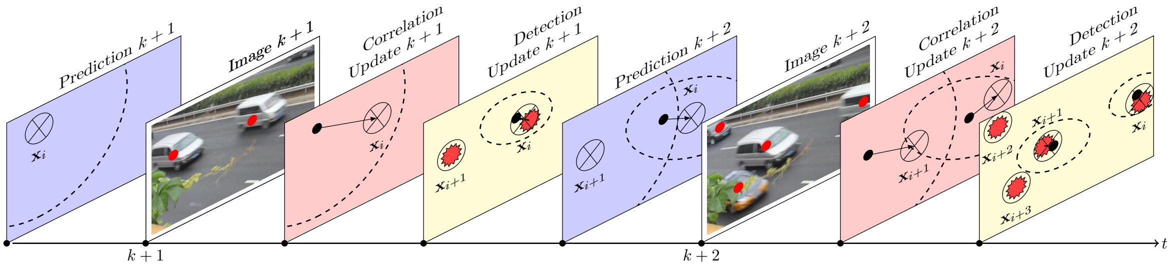


Figure 3. The proposed extension of the GMPHD tracking scheme uses visual correlation filters to perform an additional update step. This reduces track uncertainty before associating detections and allows successful tracking of fast objects even in dense scenes.

tions follows. Because the correlation filter update is executed on the same image as the external detector, both sources of information are synchronized, such that no additional prediction step between the correlation filter update and the regular detection update is necessary.

The whole tracking scheme of the proposed extended GMPHD filter is visualized in Figure 3. Please note that the additional update step influences only the state prediction and its covariance. The weights of the state predictions are still only effected by regular detections, *i.e.* the GMPHD filter equations remain unchanged and only hypotheses confirmed by external detections can get extracted. This avoids the problem of infinite false-detection tracks.

5. Experimental Results

The pure GMPHD filter and the proposed extension have both been evaluated on the full and the ”Experienced” test set of the UA-DETRAC benchmark introduced in [10].

The GMPHD filter depends on robust estimates for certain parameters and values. For example, the detection probability p_D, *i.e.* the probability that a real object in an image is detected by the sensor, is crucial. Furthermore good estimates for the clutter intensity C and the measurement noise $\vec{\eta}_D$ of the detector are essential. Additionally, also the value for the extraction threshold $T_{extract}$ depends on the quality of the input detections. Usually, those values are chosen depending of the detection threshold used for the detector. However, because the UA-DETRAC metrics rely on a evaluation scheme with changing detection thresholds, p_D, C, $\vec{\eta}_D$, and $T_{extract}$ cannot be considered constant.

As a remedy, those values are dynamically estimated from the used detection threshold $T_D \in [0,1]$ before each evaluation run by linear relationships:

$$p_D(T_D) = -m_p \cdot T_D + p_{D,0} \tag{9}$$

$$C(T_D) = p_D(T_D) \cdot C_0 \tag{10}$$

$$\vec{\eta}_D(T_D) = -m_\eta \cdot T_D + \vec{\eta}_{D,0} \tag{11}$$

$$T_{extract}(T_D) = -m_{extract} \cdot T_D + T_{extract,0} \tag{12}$$

Together with all other parameters of both methods, the linear coefficients have been obtained by multidimensional nonlinear optimization of the *PR-MOTA* value on a subset of the UA-DETRAC training set.

The results of the pure GMPHD and the proposed extension on the UA-DETRAC benchmark are shown together with the performance of other state-of-the art trackers in Table 1, where they are denoted as *GMPHD* and *GMPHD-KCF*, respectively. Considering only the *PR-MOTA* results, the performance of the plain GMPHD filter and the GMPHD-KCF is very similar.

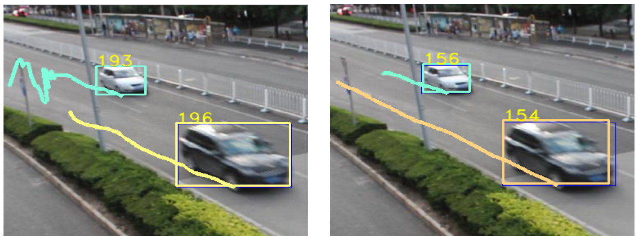
The *PR-MOTA* metric is a metric that combines false-positives, false-negatives and track-ID-switches equally. A closer look onto those specific values shows that the equivalence in performance is mainly due to the sensibility of the GMPHD-KCF to false-positives. Although the GMPHD-KCF is able to significantly reduce the number of false-negatives compared to the plain GMPHD filter, this positive effect is canceled out by the large increase of false-positives. Considering the order of magnitude of false-negatives and false-positives, the number of ID-switches is nearly insignificant for the difference in the *PR-MOTA*. However, the positive effect of extending the GMPHD filter with visual correlation filters is seen when comparing the results for *PR-MT*, *PR-ML*, and *PR-FRAG* (*i.e.* mostly-tracked, mostly-lost, and fragmented). The GMPHD-KCF shows a significant improvement in the stability of tracked objects and recall. Furthermore the extracted tracks are less fragmented compared to the plain GMPHD filter.

Generally, the GMPHD-KCF extracts not only more tracks than the plain GMPHD filter, the tracks are also more stable, less fragmented, and extracted earlier. This leads to lower false-negatives rates and a higher recall. However, due to the included visual tracking scheme, also false-positive detections (especially false-positives that repeatedly appear in different frames) are tracked and extracted instead of suppressed as in the plain GMPHD filter. Additionally, the increased computational complexity leads to reduced speed.

Due to the evaluation scheme used for the benchmark, trackers have to work reliably for both low and high de-

Detector	Tracker	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FRAG	PR-FP	PR-FN	Speed(fps)
Overall Test Set (Easy + Medium + Hard)										
CompACT	GMPHD-KCF	**14.5%**	36.0%	14.0%	**18.1%**	798.8	1606.8	38596.6	174042.7	24.60
CompACT	GOG	14.2%	**37.0%**	13.9%	19.9%	3334.6	7172.4	**32092.9**	180183.8	**389.51**
CompACT	GMPHD	14.1%	36.3%	13.2%	19.0%	797.2	2143.8	38032.4	177215.1	45.24
CompACT	CMOT	12.6%	36.1%	**16.1%**	18.6%	**285.3**	1516.8	57885.9	**67110.8**	3.79
CompACT	H2T	12.4%	35.7%	14.8%	19.4%	852.2	**1117.2**	51765.7	173899.8	3.02
Experienced Test Set (Medium + Hard)										
EB	GMPHD	**14.4%**	26.5%	12.3%	18.8%	994.3	1660.4	**19627.3**	139807.3	41.30
EB	GMPHD-KCF	14.1%	25.9%	**12.5%**	**18.5%**	909.9	1437.2	21863.7	**139245.4**	7.74
CompACT	GMPHD-KCF	12.0%	33.8%	10.8%	19.5%	648.8	**1300.2**	30518.1	140669.4	23.10
CompACT	GMPHD	11.7%	**33.9%**	10.0%	20.5%	**631.1**	1685.2	29574.2	143007.8	**48.50**

Table 1. Tracking results on the UA-DETRAC benchmark. Methods proposed in this paper are highlighted. CompACT, GOG, CMOT and H2T are provided by the benchmark. EB has been published in [9], the detections have been generated by its code (available online[4]).



(a) Plain GMPHD filter

(b) GMPHD-KCF

Figure 4. Example situation in the UA-DETRAC training set. Two cars are close to each other and appear at almost the same time. The plain GMPHD filter, shown in (a), initialized the track #193 on the first detection of the black car, but gets confused by detections of the newly appearing white car, leading to several switches, until it finally settles on the wrong car. The GMPHD-KCF, shown in (b), is able to distinguish both objects and track them separately.

tection thresholds. As mentioned, the PHD filter requires a very careful parametrization, especially regarding clutter and detection probability, and the used linear dependencies to the detection threshold are only very rough estimates. Table 1 still shows that the overall results are on a similar level as other state-of-the-art methods. The GMPHD-KCF even outperforms the best tracker of the benchmark baseline.

6. Conclusions

This work evaluated different issues of a GMPHD filter regarding typical situations in the UA-DETRAC benchmark. In order to resolve them, an extension using a kernelized correlation tracker has been proposed. Both the baseline and the extended method have been evaluated on the test set of the benchmark and show very promising results. The extended GMPHD filter offers a higher recall and tracks with a higher quality, but this comes at the cost of increased runtime and a higher sensitivity to false-positives. Future work will include how the robust visual object representations, learned by the correlation filters, can be used to further improve the algorithm.

Acknowledgment. The research leading to these results has received funding from the European Community's FP7 and BMBF-VIP+ under grant agreement number 607480 (LASIE) and 03VP01940 (SiGroViD).

References

[1] D. Alspach. *A Bayesian Approximation Technique for Estimation and Control of Time Discrete Stochastic Systems*. University of California, San Diego, 1970. 3

[2] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2, 3

[3] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proc. of the British Machine Vision Conference BMVC*, 2014. 2, 3

[4] V. Eiselein, D. Arp, M. Pätzold, and T. Sikora. Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors. In *12th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 325–330, 2012. 2

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 3

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015. 2

[7] K. Panta, D. Clark, and B.-N. Vo. Data association and track management for the gaussian mixture probability hypothesis density filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 45(3):1003 –1016, july 2009. 3

[8] B.-N. Vo and W.-K. Ma. The gaussian mixture probability hypothesis density filter. *Signal Processing, IEEE Transactions on*, 54(11):4091 – 4104, nov. 2006. 2, 3

[9] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue. Evolving boxes for fast vehicle detection. *CoRR*, abs/1702.00254, 2017. 6

[10] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136, 2015. 5

[4] http://zyb.im/research/EB/