Long-term Correlation Tracking using Multi-layer Hybrid Features in Dense Environments

Nathanael L. Baisa[1], Deepayan Bhowmik[2] and Andrew Wallace[1]
[1]*Department of Electrical, Electronic and Computer Engineering, Heriot Watt University, Edinburgh, United Kingdom*
[2]*Department of Computing, Sheffield Hallam University, Sheffield, United Kingdom*
{*nb30, a.m.wallace*}*@hw.ac.uk, deepayan.bhowmik@shu.ac.uk*

Keywords: Visual tracking, Correlation filter, CNN features, Hybrid features, Online learning, GM-PHD filter.

Abstract: Tracking a target of interest in crowded environments is a challenging problem, not yet successfully addressed in the literature. In this paper, we propose a new long-term algorithm, learning a discriminative correlation filter and using an online classifier, to track a target of interest in dense video sequences. First, we learn a translational correlation filter using a multi-layer hybrid of convolutional neural networks (CNN) and traditional hand-crafted features. We combine the advantages of both the lower convolutional layer which retains better spatial detail for precise localization, and the higher convolutional layer which encodes semantic information for handling appearance variations. This is integrated with traditional features formed from a histogram of oriented gradients (HOG) and color-naming. Second, we include a re-detection module for overcoming tracking failures due to long-term occlusions by training an incremental (online) SVM on the most confident frames using hand-engineered features. This re-detection module is activated only when the correlation response of the object is below some pre-defined threshold to generate high score detection proposals. Finally, we incorporate a Gaussian mixture probability hypothesis density (GM-PHD) filter to temporally filter high score detection proposals generated from the learned online SVM to find the detection proposal with the maximum weight as the target position estimate by removing the other detection proposals as clutter. Extensive experiments on dense data sets show that our method significantly outperforms state-of-the-art methods.

1 INTRODUCTION

Visual target tracking is one of the most important and active research areas in computer vision with a wide range of applications like surveillance, robotics and human-computer interaction (HCI). Although it has been studied extensively during past decades as recently surveyed in (Smeulders et al., 2014), object tracking is still a difficult problem due to many challenges that cause significant appearance changes of targets such as illumination changes, occlusion, pose variation, deformation, abrupt motion, and background clutter. Tracking an interested target in dense or crowded environments is an important task in some security applications. However, it is very challenging due to heavy occlusions, high target densities, cluttered scenes and significant appearance variations of targets. Robust representation of target appearance is important to overcome these challenges.

Recently, convolutional neural networks (CNN) features have demonstrated outstanding results on various recognition tasks (Girshick et al., 2014; Simonyan and Zisserman, 2015). Motivated by this, a few deep learning based trackers (Wang and Yeung, 2013; Ma et al., 2015a; Wang et al., 2015) have been developed. In addition, discriminative correlation filters-based trackers have achieved state-of-the-art results as surveyed in (Chen et al., 2015) in terms of both efficiency and robustness due to three reasons. First, efficient correlation operations are performed by replacing exhausted circular convolutions with element-wise multiplications in the frequency domain which can be computed using the fast fourier transform (FFT) with very high speed. Second, thousands of negative samples around the target's environment can be efficiently incorporated through circular-shifting with the help of a circulant matrix. Third, training samples are regressed to soft labels of a Gaussian function (Gaussian-weighted labels) instead of binary labels alleviating sampling ambiguity. In fact, regression with class labels can be seen as classification.

In addition, the Gaussian mixture probability hypothesis density (GM-PHD) filter (Vo and Ma, 2006)

has the in-built capability of removing clutter while filtering targets with very efficient speed without the need for explicit data association. Though this filter is designed for multi-target filtering, it is even preferable for single target filtering in scenes with challenging background clutter, as well as clutter that comes from other targets not currently of interest.

In this work, we mainly focus on long-term tracking of a target of interest in crowded environments where an unknown target is initialized by a bounding box and then is tracked in subsequent frames. Without any constraint on the video scene of application, we develop an online tracking algorithm that can track a target of interest in dense scenes using the advantages of the correlation filter, hybrid of multi-layer CNN and hand-crafted features, an online support vector machine (SVM) classifier and Gaussian mixture probability hypothesis density (GM-PHD) filter.

We make the following three contributions. First, we integrate hybrid of multi-layer CNN and traditional features for learning translation correlation filter by extending a ridge regression for multi-layer features. Second, we include a re-detection module by learning an incremental (online) SVM for generating high score detection proposals. Third, we temporally filter the generated high score detection proposals using GM-PHD filter to find the detection proposal with maximum weight as the target position estimate, removing clutter in dense environments and re-initializing the tracker in case of tracking failures.

2 RELATED WORK

Various visual tracking algorithms have been proposed over the past decades to cope with tracking challenges, and they can be categorized into *generative* and *discriminative* methods depending on the learning strategy. *Generative* methods describe the target appearance using generative models and search for target regions that best-match the models. Various generative target appearance modelling algorithms have been proposed such as online density estimation (Han et al., 2008), sparse representation (Zhang et al., 2012), and incremental subspace learning (Ross et al., 2008). On the other hand, *discriminative* methods build a model that distinguishes the target from the background. These algorithms typically learn classifiers based on online boosting (Grabner et al., 2008), multiple instance learning (Babenko et al., 2011), P-N learning (Kalal et al., 2012), structured output SVMs (Hare et al., 2011) and combining multiple classifiers with different learning rates (Zhang et al., 2014). Discriminative methods are most competitive to the work presented here since they include background information, although hybrid generative and discriminative models can also be used (Dinh et al., 2014). However, sampling ambiguity in discriminative tracking methods results in drifting, which is a significant problems. Recently, correlation filters (Henriques et al., 2012; Henriques et al., 2015; Danelljan et al., 2014) have beenn introduced for online target tracking that can alleviate this sampling ambiguity.

There are about three tracking scenarios that are important to consider: short-term tracking, long-term tracking, and tracking in a crowded scene. If objects are visible over the whole course of the sequences, short-term model-free tracking algorithms are sufficient to track a single object though they can not re-initialize the trackers once they fail due to long-term occlusion and confusion from background clutters (Han et al., 2008; Danelljan et al., 2014). Long-term tracking algorithms are important for target tracking in a video stream that runs for indefinitely long handling long-term occlusions. A Tracking-Learning-Detection (TLD) algorithm has been developed in (Kalal et al., 2012) which explicitly decomposes the long-term tracking task into tracking, learning and detection. However, it is sensitive to background clutter although it works well in very sparse video. Long-term correlation tracking (LCT), developed in (Ma et al., 2015b), learns three different discriminative correlation filters: translation, appearance and scale correlation filters using hand-crafted features, however, it is not robust to long-term occlusions and background clutter.

Tracking of a target of interest in a crowded scene is very challenging due to heavy occlusion, high target densities and clutter, and significant appearance variations. Person detection and tracking in crowds is formulated as a joint energy minimization problem by combining crowd density estimation and localization of an individual person in (Rodriguez et al., 2011). Though this approach doesn't require manual initialization, it has low performance for tracking a generic target of interest as it was mainly developed for tracking human heads. The method developed in (Kratz and Nishino, 2012) trained Hidden Markov Models (HMMs) on motion patterns within a scene to capture spatial and temporal variations of motion in the crowd which is used for tracking individuals. However, this approach is limited to a crowd with a structured pattern. The algorithm developed in (Idrees et al., 2014) used visual information (prominence) and spatial context (influence from neighbours) to develop online tracking in a crowded scene. This algorithm performs well in crowded scene but has low performance in

medium density scenes as influence from neighbours (spatial context) decreases in such scene.

Our proposed tracking algorithm tracks a target of interest in dense environments without using any constraint from the video scene using a correlation filter, sophisticated features and a re-detection scheme, and is robust to occluded and densely cluttered scenes.

3 OVERVIEW OF OUR ALGORITHM

CNN features have recently demonstrated outstanding results on various recognition tasks though traditional hand-engineered features are still important. Similarly, correlation filters are giving better results for online tracking problems in both efficiency and accuracy. Besides, the GM-PHD filter is efficient in removing clutter that originates from both the background scene and other targets not of interest. Having observed these factors, we develop a long-term online tracking algorithm that can be applied to track a target of interest in densely cluttered environments by learning a correlation filter using a hybrid of multi-layer CNN and hand-crafted features as well as including a re-detection module using an incremental SVM and GM-PHD filter.

Accordingly, first we learn a translation correlation filter ($\mathbf{w}_t$) using a hybrid of multi-layer CNN features from VGG-Net (Simonyan and Zisserman, 2015) and robust traditional hand-crafted features.

For the CNN part, we combine features from both a lower convolutional layer which retains more spatial detail for precise localization and a higher convolutional layer which encodes semantic information for handling appearance variations. This forms layers 1 and 2 in multi-layer features with multiple channels (512 dimensions) in each layer. Since the spatial resolution of the extracted features gradually reduces with the increase of the depth of CNN layers due to pooling operators, it is crucial to resize each feature map to a fixed size using bilinear interpolation.

For the traditional features part, we use the histogram of oriented gradients (HOG), in particular Felzenszwalb's variant (Felzenszwalb et al., 2010) and color-naming (van de Weijer et al., 2009) features for capturing image gradients and color information, respectively. Color-naming is the linguistic color label assigned by a human to describe the color, hence, the mapping method in (van de Weijer et al., 2009) is employed to convert the RGB space into the color name space which is an 11 dimensional color representation providing the perception of a target color. By aligning the feature size of the HOG variant with 31 dimensions and color-naming with 11 dimensions, they are integrated to make a 42 dimensional feature which forms the 3rd layer in our hybrid multi-layer features.

Second, we incorporate a re-detection module by learning incremental SVM from the most confident frames determined by the maximal value of the correlation response map. This uses HOG, LUV color and normalized gradient magnitude features for generating high-score detection proposals which are filtered using the GM-PHD filter to re-acquire the target in case of tracking failures. The flowchart of our method is given in Figure 1 and the outline of our proposed algorithm is given in Algorithm 1.

4 PROPOSED ALGORITHM

This section describes our proposed tracking algorithm which has three distinct functional parts: 1) correlation filters formulated for multi-layer hybrid features, 2) online SVM detector developed for generating high score detection proposals, and 3) GM-PHD filter for finding the detection proposal with maximum weight to re-initialize the tracker in case of tracking failures by removing the other detection proposals as clutter.

4.1 Correlation Filters for Multi-layer Features

To track a target using correlation filters, the appearance of the target should be modeled using a correlation filter $\mathbf{w}$ which can be trained on a feature vector $\mathbf{x}$ of size $M \times N \times D$ extracted from an image patch where M, N, and D indicate the width, height and number of channels, respectively. This feature vector $\mathbf{x}$ can be extracted from multiple layers, for example CNN features and/or traditional hand-crafted features, therefore, we denote it as $\mathbf{x}^{(l)}$ to designate from which layer l it is extracted. All the circular shifts of $\mathbf{x}^{(l)}$ along the M and N dimensions are considered as training examples where each circularly shifted sample $\mathbf{x}^{(l)}_{m,n}, m \in \{0,1,...,M-1\}, n \in \{0,1,...,N-1\}$ has a Gaussian function label y(m,n) given by

$$y(m,n) = e^{-\frac{(m-M/2)^2+(n-N/2)^2}{2\sigma^2}}, \qquad (1)$$

where σ is the kernel width, hence, $y(m,n)$ is a soft label rather than a binary label. To learn the correlation filter $\mathbf{w}^{(l)}$ for layer l with the same size as $\mathbf{x}^{(l)}$, we extend ridge regression (Rifkin et al., 2003), developed for a single-layer feature vector, to be used for a multi-layer hybrid feature vector with layer l, $\mathbf{x}^{(l)}$, as

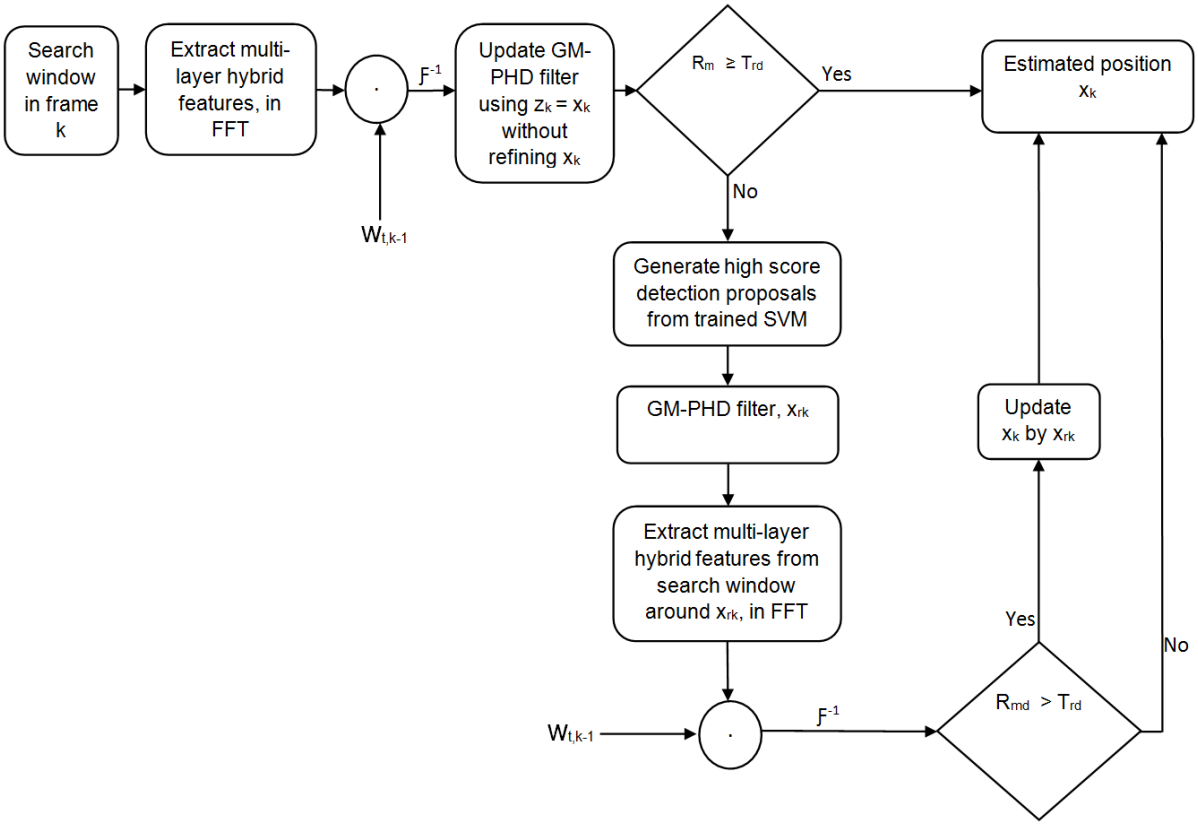

Figure 1: The flowchart of the proposed algorithm. It consists of two main parts: translation estimation and re-detection. Given a search window, we extract multi-layer hybrid features (in the frequency domain) and then estimate target position ($\mathbf{x}_k$) using a translation correlation filter ($\mathbf{w}_t$). This estimated position ($\mathbf{x}_k$) is used as a measurement ($\mathbf{z}_k$) for updating GM-PHD filter without refining $\mathbf{x}_k$, just to update its weight for later use during re-detection. Re-detection is activated if the maximum of the response map (R_m) falls below a pre-defined threshold (T_{rd}). Then, we generate high score detection proposals which are filtered by the GM-PHD filter to estimate the detection with the maximum weight as target position ($\mathbf{x}_{rk}$) removing the others as clutters. If the response map around $\mathbf{x}_{rk}$ (R_{md}) is greater than T_{rd}, the target position $\mathbf{x}_k$ is updated by a re-detected position $\mathbf{x}_{rk}$. In frame 1, we only train the correlation filter and SVM classifier using the initialized target; no detection is performed.

$$\min_{\mathbf{w}^{(l)}} \sum_{m,n} |\Phi(\mathbf{x}^{(l)}).\mathbf{w}^{(l)} - y(m,n)|^2 + \lambda |\mathbf{w}^{(l)}|^2, \quad (2)$$

where Φ denotes the mapping to a kernel space and λ is a regularization parameter ($\lambda \geq 0$). The solution $\mathbf{w}^{(l)}$ can be expressed as

$$\mathbf{w}^{(l)} = \sum_{m,n} \mathbf{a}^{(l)}(m,n)\Phi(\mathbf{x}^{(l)}_{m,n}), \quad (3)$$

This alternative representation makes the dual space $\mathbf{a}^{(l)}$ the variable under optimization instead of the primal space $\mathbf{w}^{(l)}$.

Training phase: The training phase is performed in the Fourier domain using the fast Fourier transform (FFT) to compute the coefficient $\mathbf{A}^{(l)}$ as

$$\mathbf{A}^{(l)} = \mathcal{F}(\mathbf{a}^{(l)}) = \frac{\mathcal{F}(y)}{\mathcal{F}\left(\Phi(\mathbf{x}^{(l)}).\Phi(\mathbf{x}^{(l)})\right) + \lambda}, \quad (4)$$

where $\mathcal{F}$ denotes the FFT operator.

Detection phase: The detection phase is performed on the new frame given an image patch (search window) which is used as spatial context i.e. the search window is larger than the target. If feature vector $\mathbf{z}^{(l)}$ of size $M \times N \times D$ is extracted from this image patch, the response map ($\mathbf{r}^{(l)}$) is computed as

$$\mathbf{r}^{(l)} = \mathcal{F}^{-1}\left(\tilde{\mathbf{A}}^{(l)} \odot \mathcal{F}\left(\Phi(\mathbf{z}^{(l)}).\Phi(\tilde{\mathbf{x}}^{(l)})\right)\right), \quad (5)$$

where $\tilde{\mathbf{A}}^{(l)}$ and $\tilde{\mathbf{x}}^{(l)} = \mathcal{F}^{-1}(\tilde{\mathbf{X}}^{(l)})$ denote the learned target appearance model for layer l, operator $\odot$ is the Hadamard (element-wise) product, and $\mathcal{F}^{-1}$ is the inverse FFT. Now, the response maps of all layers are summed according to their weight $\gamma(l)$ element-wise as

$$\mathbf{r}(m,n) = \sum_l \gamma(l)\mathbf{r}^{(l)}(m,n), \quad (6)$$

The new target position is estimated by finding the maximum value of $\mathbf{r}(m,n)$ as

$$(\hat{m},\hat{n}) = \underset{m,n}{\operatorname{argmax}}\ \mathbf{r}(m,n), \tag{7}$$

Model update: The model is updated by training a new model at the new target position and then linearly interpolating the obtained values of the dual space coefficients $\mathbf{A}_k^{(l)}$ and the base data template $\mathbf{X}_k^{(l)} = \mathcal{F}(\mathbf{x}_k^{(l)})$ with those from the previous frame to make the tracker more adaptive to target appearance variations.

$$\tilde{\mathbf{X}}_k^{(l)} = (1-\eta)\tilde{\mathbf{X}}_{k-1}^{(l)} + \eta\mathbf{X}_k^{(l)}, \tag{8a}$$

$$\tilde{\mathbf{A}}_k^{(l)} = (1-\eta)\tilde{\mathbf{A}}_{k-1}^{(l)} + \eta\mathbf{A}_k^{(l)}, \tag{8b}$$

where k is the index of the current frame, and η is the learning rate.

The mappings to the kernel space (Φ) used in (4) and (5) can be expressed using the kernel function as $K(\mathbf{x}_i^{(l)},\mathbf{x}_j^{(l)}) = \Phi(\mathbf{x}_i^{(l)}).\Phi(\mathbf{x}_j^{(l)}) = \Phi(\mathbf{x}_i^{(l)})^T\Phi(\mathbf{x}_j^{(l)})$. If the computation is performed in frequency domain, the normal transpose should be replaced by the Hermitian transpose i.e. $\Phi(\mathbf{X}_i^{(l)})^H = (\Phi(\mathbf{X}_i^{(l)})^*)^T$ where the star ($*$) denotes the complex conjugate. A linear kernel is used and is given as

$$K(\mathbf{x}_i^{(l)},\mathbf{x}_j^{(l)}) = (\mathbf{x}_i^{(l)})^T\mathbf{x}_j^{(l)} = \mathcal{F}^{-1}(\sum_d (\mathbf{X}_{i,d}^{(l)})^* \odot \mathbf{X}_{j,d}^{(l)}), \tag{9}$$

where $\mathbf{X}_i^{(l)} = \mathcal{F}(\mathbf{x}_i^{(l)})$.

This formulation is a generic formulation for multiple channel features from multiple layers as in the case of our multi-layer hybrid features, i.e. where $\mathbf{X}_{i,d}^{(l)},\ d \in \{1,...,D\},\ l \in \{1,...,L\}$. This is an extended version of the one given in (Henriques et al., 2015) that takes into account features from multiple layers. The linearity of the FFT allows us to simply sum the individual dot-products for each channel $d \in \{1,...,D\}$ in each layer $l \in \{1,...,L\}$.

4.2 Online Detection

We include a re-detection module, D_r, to generate high score detection proposals in case of tracking failures due to long-term occlusion. Instead of using a correlation filter to scan across the entire frame which is computationally expensive, we learn an incremental (online) SVM (Diehl and Cauwenberghs, 2003) by generating a set of samples in the search window around the estimated target position from the most confident frames and scan through the window when it is activated to generate high score detection proposals. These most confident frames are determined by the maximum translation correlation response in the current frame, i.e. if the maximum correlation response of an image patch is above the trained detector threshold (T_{td}), we generate samples around this image patch and train the detector. This detector is activated to generate high score detection proposals if the maximum of the correlation response becomes below activate detector threshold (T_{ad}). We use HOG (particularly Felzenszwalbs variant (Felzenszwalb et al., 2010)), LUV color and normalized gradient magnitude features to train this online SVM classifier. We use different visual features from the ones we use to learn the correlation filter. Since we can select the feature representation for each module independently (Danelljan et al., 2014; Ma et al., 2015b), this greatly reduces the computational cost.

We want to update the weight vector $\mathbf{w}$ of the SVM by providing a set of samples with associated labels, $\{(\acute{\mathbf{x}}_i,\acute{\mathbf{y}}_i)\}$, obtained from the current results. The label $\acute{\mathbf{y}}_i$ of a new example $\acute{\mathbf{x}}_i$ is given by

$$\acute{\mathbf{y}}_i = \begin{cases} +1, & \text{if } IOU(\acute{\mathbf{x}}_i,\ddot{\mathbf{x}}_t) \geq \delta_p \\ -1, & \text{if } IOU(\acute{\mathbf{x}}_i,\ddot{\mathbf{x}}_t) < \delta_n \end{cases} \tag{10}$$

where $IOU(.)$ is the intersection over union (overlap ratio) of a new example $\acute{\mathbf{x}}_i$ and the estimated target bounding box in the current most confident frame $\ddot{\mathbf{x}}_t$.

SVM classifiers of the form $f(\mathbf{x}) = \mathbf{w}.\Phi(\mathbf{x}) + b$ are learned from the data $\{(\mathbf{x}_i,\mathbf{y}_i) \in \Re^m \times \{-1,+1\} \forall i \in \{1,...,N\}\}$ by minimizing

$$\min_{\mathbf{w},b,\xi}\ \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i^p \tag{11}$$

for $p \in \{1,2\}$ subject to the constraints

$$y_i(\mathbf{w}.\Phi(\mathbf{x_i}) + b) \geq 1 - \xi_i, \xi_i \geq 0\ \forall i \in \{1,...,N\}. \tag{12}$$

Hinge loss ($p = 1$) is preferred over quadratic loss ($p = 2$) due to its improved robustness to outliers. Thus, the offline SVM learns a weight vector $\mathbf{w} = (w_1, w_2, ..., w_N)^T$ by solving this quadratic convex optimization problem (QP) which can be expressed in its dual form as

$$\min_{0\leq a_i\leq C} W = \frac{1}{2}\sum_{i,j=1}^{N} a_i Q_{ij} a_j - \sum_{i=1}^{N} a_i + b\sum_{i=1}^{N} y_i a_i, \tag{13}$$

where $\{a_i\}$ are Lagrange multipliers, b is bias, C is a regularization parameter, and $Q_{ij} = y_i y_j K(\mathbf{x_i},\mathbf{x_j})$. The kernel function $K(\mathbf{x_i},\mathbf{x_j}) = \Phi(\mathbf{x_i}).\Phi(\mathbf{x_j})$ is used to implicitly map into a higher dimensional feature space and compute the dot product. It is not straightforward for conventional QP solvers to handle the optimization problem in (13) for online tracking tasks as the training data are provided sequentially, not all at once.

Incremental SVM (Diehl and Cauwenberghs, 2003) is tailored for such cases which retain the Karush-Kuhn-Tucker (KKT) conditions on all the existing examples while updating the model with a new example so that the exact solution at each increment of the dataset can be guaranteed. KKT conditions are the first-order necessary conditions for the optimal unique solution of dual parameters $\{a,b\}$ which minimizes (13) and are given by

$$\frac{\partial W}{\partial a_i} = \sum_{j=1}^{N} Q_{ij}a_j + y_i b - 1 \begin{cases} > 0, \text{if } a_i = 0 \\ = 0, \text{if } 0 \leq a_i \leq C \\ < 0, \text{if } a_i = C, \end{cases} \quad (14)$$

$$\frac{\partial W}{\partial b} = \sum_{j=1}^{N} y_j a_j = 0, \quad (15)$$

Based on the partial derivative $m_i = \frac{\partial W}{\partial a_i}$ which is related to the margin of the i-th example, each training example can be categorized into three: $\mathcal{S}_1$ support vectors lying on the margin ($m_i = 0$), $\mathcal{S}_2$ support vectors lying inside the margin ($m_i < 0$), and the remaining $\mathcal{R}$ reserve vectors (non-support vectors). During incremental learning, new examples with $m_i \leq 0$ eventually become margin ($\mathcal{S}_1$) or error ($\mathcal{S}_2$) support vectors. However, the rest of the new training examples become reserve vectors as they do not enter the solution so that lagrangian multipliers (a_i) are estimated while retaining the KKT conditions. Given the updated Lagrangian multipliers, the weight vector $\mathbf{w}$ is given by

$$\mathbf{w} = \sum_{i \in \mathcal{S}_1 \cup \mathcal{S}_2} a_i y_i \Phi(\mathbf{x_i}), \quad (16)$$

It is important to keep only a fixed number of support vectors with the smallest margins for efficiency during online tracking.

Thus, using the trained incremental SVM, we generate high score detections as detection proposals during the re-detection stage which are filtered using the GM-PHD filter to find the best possible detection that can re-initialize the tracker.

4.3 Temporal Filtering using GM-PHD Filter

Once we generate high score detection proposals using the online SVM classifier during the re-detection stage, we need to find the most probable detection proposal for the target state (position) estimate by finding the detection proposal with maximum weight using the GM-PHD filter (Vo and Ma, 2006). Though the GM-PHD filter is designed for multi-target filtering with the assumptions of a linear Gaussian system, in our problem (re-detecting a target in cluttered scene), it is used for removing clutter that come from the background and other targets not of interest as it is equipped with such a capability. Besides, it provides motion information for the tracking algorithm. More importantly, using the GM-PHD filter to find the detection with the maximum weight from the generated high score detection proposals is more robust than relying only on the maximum score of the classifier.

The detected position of the target in each frame is filtered using the GM-PHD filter, but without refining the position states until the re-detection module is activated. This updates the weight of the GM-PHD filter corresponding to a target of interest giving sufficient prior information to be picked up during re-detection among candidate high score detection proposals. If the re-detection module is activated (correlated response of the target becomes below a pre-defined threshold), we generate high score detection proposals (in this case 5) from the trained SVM classifier which are filtered using the GM-PHD filter. The Gaussian component with maximum weight is selected as the position estimate, and if the correlated response of this estimated position is greater than the pre-defined threshold, the estimated position of the target is refined.

The GM-PHD filter has two steps: prediction and update. Before stating these two steps, certain assumptions are needed. 1) Each target follows a linear Gaussian model:

$$y_{k|k-1}(x|\zeta) = \mathcal{N}(x; F_{k-1}\zeta, Q_{k-1}) \quad (17)$$

$$f_k(z|x) = \mathcal{N}(z; H_k x, R_k) \quad (18)$$

where $\mathcal{N}(.;m,P)$ denotes a Gaussian density with mean m and covariance P; F_{k-1} and H_k are the state transition and measurement matrices, respectively. Q_{k-1} and R_k are the covariance matrices of the process and the measurement noise respectively.

2) A current measurement driven birth intensity inspired by but not identical to (Ristic et al., 2012) is introduced at each time step, removing the need for prior knowledge (specification of birth intensities) or a random model, with a non-informative zero initial velocity. The intensity of the spontaneous birth RFS is a Gaussian mixture of the form

$$\gamma_k(x) = \sum_{v=1}^{V_{\gamma,k}} w_{\gamma,k}^{(v)} \mathcal{N}(x; m_{\gamma,k}^{(v)}, P_{\gamma,k}^{(v)}) \quad (19)$$

where $V_{\gamma,k}$ is the number of birth Gaussian components, $w_{\gamma,k}^{(v)}$ is the weight accompanying the Gaussian

component v, $m_{\gamma,k}^{(v)}$ is the current measurement and zero initial velocity used as mean, and $P_{\gamma,k}^{(v)}$ is the birth covariance for Gaussian component v. In our case, $V_{\gamma,k}$ equals 1 unless in the re-detection stage, at which it becomes 5 as we generate 5 high score detection proposals to be filtered.

3) The survival and detection probabilities are independent of the target state: $p_{s,k}(x_k) = p_{s,k}$ and $p_{D,k}(x_k) = p_{D,k}$.

Prediction: It is assumed that the posterior intensity at time $k-1$ is a Gaussian mixture of the form

$$\mathcal{D}_{k-1}(x) = \sum_{v=1}^{V_{k-1}} w_{k-1}^{(v)} \mathcal{N}(x; m_{k-1}^{(v)}, P_{k-1}^{(v)}), \tag{20}$$

where V_{k-1} is the number of Gaussian components of $\mathcal{D}_{k-1}(x)$. This is equal to the number of Gaussian components after pruning and merging at the previous iteration. Under these assumptions, the predicted intensity at time k is given by

$$\mathcal{D}_{k|k-1}(x) = \mathcal{D}_{S,k|k}(x) + \gamma_k(x), \tag{21}$$

where

$$\mathcal{D}_{S,k|k-1}(x) = \quad p_{s,k} \sum_{v=1}^{V_{k-1}} w_{k-1}^{(v)} \mathcal{N}(x; m_{S,k|k-1}^{(v)}, P_{S,k|k-1}^{(v)}),$$

$$m_{S,k|k-1}^{(v)} = F_{k-1} m_{k-1}^{(v)},$$

$$P_{S,k|k-1}^{(v)} = Q_{k-1} + F_{k-1} P_{k-1}^{(v)} F_{k-1}^{T},$$

where $\gamma_k(x)$ is given by (19).

Since $\mathcal{D}_{S,k|k-1}(x)$ and $\gamma_k(x)$ are Gaussian mixtures, $\mathcal{D}_{k|k-1}(x)$ can be expressed as a Gaussian mixture of the form

$$\mathcal{D}_{k|k-1}(x) = \sum_{v=1}^{V_{k|k-1}} w_{k|k-1}^{(v)} \mathcal{N}(x; m_{k|k-1}^{(v)}, P_{k|k-1}^{(v)}), \tag{22}$$

where $w_{k|k-1}^{(v)}$ is the weight accompanying the predicted Gaussian component v, and $V_{k|k-1}$ is the number of predicted Gaussian components, equal to the number of born targets (1 unless in case of re-detection in which case it is 5) added to the number of persistent components, actually the number of Gaussian components after pruning and merging in the previous iteration.

Update: The posterior intensity (updated PHD) at time k is also a Gaussian mixture and is given by

$$\mathcal{D}_{k|k}(x) = (1 - p_{D,k}) \mathcal{D}_{k|k-1}(x) + \sum_{z \in Z_k} \mathcal{D}_{D,k}(x; z), \tag{23}$$

where

$$\mathcal{D}_{D,k}(x; z) = \sum_{v=1}^{V_{k|k-1}} w_k^{(v)}(z) \mathcal{N}(x; m_{k|k}^{(v)}(z), P_{k|k}^{(v)}),$$

$$w_k^{(v)}(z) = \frac{p_{D,k} w_{k|k-1}^{(v)} q_k^{(v)}(z)}{c_{s_k}(z) + p_{D,k} \sum_{l=1}^{V_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(z)},$$

$$q_k^{(v)}(z) = \mathcal{N}(z; H_k m_{k|k-1}^{(v)}, R_k + H_k P_{k|k-1}^{(v)} H_k^T),$$

$$m_{k|k}^{(v)}(z) = m_{k|k-1}^{(v)} + K_k^{(v)}(z - H_k m_{k|k-1}^{(v)}),$$

$$P_{k|k}^{(v)} = [I - K_k^{(v)} H_k] P_{k|k-1}^{(v)},$$

$$K_k^{(v)} = P_{k|k-1}^{(v)} H_k^T [H_k P_{k|k-1}^{(v)} H_k^T + R_k]^{-1},$$

The clutter intensity due to the scene, $c_{s_k}(z)$, in (23) is given by

$$c_{s_k}(z) = \lambda c(z) = \lambda_c A c(z), \tag{24}$$

where $c(.)$ is the uniform density over the surveillance region A, and λ_c is the average number of clutter returns per unit volume i.e. $\lambda = \lambda_c A$. We set the clutter rate or false positive per image (fppi) $\lambda = 4$ in our experiment.

After update, weak Gaussian components with weight $w_k^v < 10^{-5}$ are pruned, and Gaussian components with Mahalanobis distance less than $U = 4$ pixels from each other are merged. These pruned and merged Gaussian components are predicted as existing (persistent) targets in the next iteration. Finally, the Gaussian component of the posterior intensity with mean corresponding to the maximum weight is selected as a target position estimate when the re-detection module is activated.

5 IMPLEMENTATION DETAILS

The main steps of our proposed algorithm are presented in Algorithm 1. Parameter settings are given as follows. To learn the translation correlation filter, we extract features from VGG-Net (Simonyan and Zisserman, 2015) trained on a large set of object recognition data from (ImageNet) (Deng et al., 2009) by first removing the fully convolutional layers. Particularly, we use the outputs of *conv*4-4 and *conv*5-4 convolutional layers as features ($l \in \{1,2\}$ and $d \in \{1,...,D\}$) i.e. the outputs of rectilinear units (inputs of pooling) layers must be used to keep more spatial resolution. Hence, the CNN features we use have 2 layers ($L = 2$) and multiple channels ($D = 512$) for *conv*4-4 and *conv*5-4 layers. For hand-crafted features, the HOG variant with 31 dimensions and color-naming with

Algorithm 1: Proposed tracking algorithm

Input: Image $\mathbf{I}_k$, previous target position $\mathbf{x}_{k-1}$, previous correlation filter $\mathbf{w}_{t,k-1}^{(l)}$, previous SVM detector D_r

Output: Estimated target position $\mathbf{x}_k = (x_k, y_k)$, updated correlation filter $\mathbf{w}_{t,k}^{(l)}$, updated SVM detector D_r

repeat

Crop out the searching window in frame k according to (x_{k-1}, y_{k-1}) and extract multi-layer hybrid features and resize them to a fixed size;

// Translation estimation

foreach *layer l* **do**

compute response map $\mathbf{r}^{(l)}$ using $\mathbf{w}_{t,k-1}^{(l)}$ and (5);

end

Sum up the response maps of all layers element-wise according to their weight $\gamma(l)$ to get $\mathbf{r}(m,n)$ using (6);
Estimate the new target position (x_k, y_k) by finding the maximum response of $\mathbf{r}(m,n)$ using (7);

// Apply GM-PHD filter

Update GM-PHD filter using the estimated target position (x_k, y_k) as measurement but without re-fining it, just to update weight of GM-PHD filter for later use;

// Target re-detection

if $\max\left(\mathbf{r}(m,n)\right) < T_{ad}$ **then**

Use the detector D_r to generate detection proposals Z_k from high scores of incremental SVM;

// Filtering using GM-PHD filter

Filter the generated candidate detections Z_k using GM-PHD filter and select the detection with maximum weight as a re-detected target position (x_{rk}, y_{rk}). Then crop out the searching window at this re-detected position and compute its response map using (5) and (6), and call it $\mathbf{r}_{rd}(m,n)$;

if $\max\left(\mathbf{r}_{rd}(m,n)\right) \geq T_{ad}$ **then**

$(x_k, y_k) = (x_{rk}, y_{rk})$ i.e. re-fine by the re-detected position;

end

end

// Translation correlation model update

Crop out new patch centered at (x_k, y_k) and extract multi-layer hybrid features and resize them to a fixed size;

foreach *layer l* **do**

Update translation correlation filter $\mathbf{w}_{t,k}^{(l)}$ using (8);

end

// Update detector D_r

if $\max\left(\mathbf{r}(m,n)\right) \geq T_{td}$ **then**

Generate positive and negative samples around (x_k, y_k) and then extract HOG, LUV color and normalized gradient magnitude features to train incremental SVM for updating its weight vector using (16);

end

until *End of video sequences*;

11 dimensions are integrated to make a 42 dimensional feature which makes the 3rd layer in our hybrid multi-layer representation. Given an image frame with a search window size of $M \times N$ which is about 2.8 times the target size to provide some context, we resize the multi-layer hybrid features to a fixed spatial size of $\frac{M}{4} \times \frac{N}{4}$. These hybrid features from each layer are weighted by a cosine window (Henriques et al., 2015) to remove the boundary discontinuities, and then combined later in (6) for which we set γ as 1, 0.4 and 0.1 for the *conv*5-4, *conv*4-4 and hand-crafted features, respectively. We set the regularization parameter of the ridge regression in (2) to $\lambda = 10^{-4}$, and a kernel bandwidth of the Gaussian function label in (1) to $\sigma = 0.1$. The learning rate for model update in (8) is set to $\eta = 0.01$. We use a linear kernel (9) to learn the translation correlation filter.

HOG, LUV color and normalized gradient magnitude features are used to train an incremental (online) SVM classifier for the re-detection module. For the objective function given in (13), we use a Gaussian kernel, particularly for $Q_{ij} = y_i y_j K(\mathbf{x_i}, \mathbf{x_j})$, and the regularization parameter C is set to 2. Empirically, we set the activate detector threshold to $T_{ad} = 0.15$ and the train detector threshold to $T_{td} = 0.40$. The parameters in (10) are set as $\delta_p = 0.9$ and $\delta_n = 0.3$. For negative samples, we randomly sampled 3 times the number of positive samples satisfying $\delta_n = 0.3$ within the maximum search area of 4 times the target size. In the re-detection phase, we generate 5 high-score detection proposals from the trained online SVM around the estimated position within the maximum search area of 6 times the target size which are filtered using the GM-PHD filter to find the detection with the maximum weight removing the others as clutter.

6 EXPERIMENTAL RESULTS

We evaluate our proposed tracking algorithm on dense environments (medium and dense PETS 2009 data sets[1]), and compare its performance with state-of-the-art trackers using the same parameter values for all the sequences. We quantitatively evaluate the robustness of the trackers using two metrics, average precision and success rate based on center location error and bounding box overlap ratio respectively, using the one-pass evaluation (OPE) setting, running the trackers throughout a test sequence with initialization from the ground truth position in the first frame. The center location error computes the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truth positions of all the frames whereas the bounding box over-

[1] http://www.cvg.reading.ac.uk/PETS2009/a.html

lap ratio computes the intersection over union of the tracked target and ground truth bounding boxes.

We label the upper part (head + neck) of representative targets in both medium and dense PETS 2009 data sets to analyze our proposed tracking algorithm. In this experiment, our goal is to determine whether our and other methods can successfully be applied to track a target of interest in occluded and cluttered environments. Accordingly, we compare our proposed tracking algorithm with 6 state-of-the-art trackers including CF2 (Ma et al., 2015a), LCT (Ma et al., 2015b), MEEM (Zhang et al., 2014), DSST (Danelljan et al., 2014), KCF (Henriques et al., 2015) and SAMF (Li and Zhu, 2015), as well as 4 more top trackers included in the Benchmark (Wu et al., 2013), particularly SCM (Zhong et al., 2012), ASLA (Jia et al., 2012), CSK (Henriques et al., 2012) and IVT (Ross et al., 2008) both quantitatively and qualitatively.

Quantitative Evaluation: The precision (top) and success plots (bottom) based on center location error and bounding box overlap ratio, respectively, are shown in Figure 2. Our proposed tracking algorithm, denoted by LCMHT, outperforms the state-of-the-art trackers by large margin on PETS 2009 data sets in both precision and success rate measures. The rankings are given in distance precision of threshold scores at 20 pixels and overlap success of AUC score for each tracker as given in the legends.

The second and third ranked trackers are CF2 (Ma et al., 2015a) and MEEM (Zhang et al., 2014) for precision plots, respectively, and vice versa for success plots. Attention is focussed on the performance of LCT. It performs least well on the precision plots and second from the lowest on success plots on these data sets. Surprisingly, this algorithm was developed by learning three different discriminative correlation filters and even included a re-detection module for long-term tracking problems. Its performance on occluded and cluttered environments such as the PETS 2009 data sets is poor due to using less robust visual features in such environments. Even CF2 which uses CNN features has low performance compared to our proposed algorithm on these data sets. However, since our proposed tracking algorithm integrates a hybrid of multi-layer CNN and traditional features to learn the translation correlation filter and a GM-PHD filter for temporally filtering generated high score detection proposals during a re-detection phase for removing clutters, it outperforms all the available trackers significantly.

Qualitative Evaluation: Figure 3 presents the performance of our proposed tracker qualitatively compared to the state-of-the-art trackers. In this case,

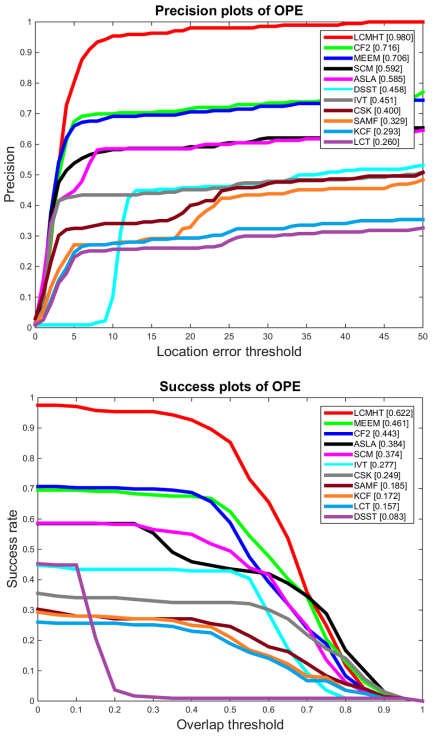

Figure 2: Distance precision (top) and overlap success (bottom) plots on PETS 2009 data sets using one-pass evaluation (OPE). The legend for distance precision contains threshold scores at 20 pixels while the legend for overlap success contains the AUC score of each tracker; the larger, the better.

we show the comparison of four representative trackers in addition to our proposed algorithm: CF2 (Ma et al., 2015a), MEEM (Zhang et al., 2014), LCT (Ma et al., 2015b), and KCF (Henriques et al., 2015). On the medium density data set (left column), LCT and KCF lose the target even in the first 16 frames. Though CF2 and MEEM trackers track the target well, they couldn't re-detect the target after occlusion i.e. only our proposed tracking algorithm tracks the target till the end of the sequence by re-initializing the tracker after the occlusion. We show the cropped and enlarged re-detection just after occlusion in Figure 4. On the dense data set (right column), all trackers track the target for the first 20 frames but LCT and KCF lose the target before 73 frames. Similar to the medium density data set, the CF2 and MEEM trackers track the target before they lose it due to occlusion. Only our proposed tracking algorithm, LCMHT, re-detects the target and tracks it till the end of the sequence in such a dense environment due to two reasons. First, it incorporates both lower and higher CNN layers in combination with traditional features (HOG and color-naming) in a multi-layer framework to learn the translation correlation filter that is ro-

Figure 3: Qualitative results of our proposed LCMHT algorithm, CF2 (Ma et al., 2015a), MEEM (Zhang et al., 2014), LCT (Ma et al., 2015b) and KCF (Henriques et al., 2015) on PETS 2009 medium (left column) and dense (right column) data sets.

bust to appearance variations of targets. Second, it includes a re-detection module which generates high score detection proposals during a re-detection phase and then filter them using the GM-PHD filter to remove clutter due to background and other targets so that it can re-detect the target of interest.

Our proposed tracking algorithm is implemented in MATLAB on 4 cores of a 3.0 GHz Intel Xeon CPU E5-1607 with 16 GB RAM. We also use the MatConvNet toolbox (Vedaldi and Lenc, 2015) for CNN feature extraction where its forward propagation computation is transferred to a NVIDIA Quadro K5000, and our tracker runs at 1 fps on this setting. The re-detection and forward propagation for feature extractions step are the main computational loads of our tracking algorithm.

7 CONCLUSIONS

We have developed a novel long-term visual tracking algorithm by learning a discriminative correlation filter and an incremental SVM classifier for tracking a target of interest in dense environments. We learn the translation correlation filter for which we combine a hybrid of multi-layer CNN (both lower and higher convolutional layers) and traditional (HOG and color-naming) features in proper proportion. We also include a re-detection module using HOG, LUV color and normalized gradient magnitude features for re-initializing the tracker in the case of tracking failures due to long-term occlusion by training an incremental SVM from the most confident frames. When activated, the re-detection module generates high score detection proposals which are temporally filtered using a GM-PHD filter for removing clutters. Extensive experimental results on PETS 2009 data sets show that our proposed algorithm outperforms the state-of-the-art trackers in terms of both accuracy and robustness. We conclude that learning a correlation filter using an appropriate combination of CNN and traditional features as well as including a re-detection module using incremental SVM and GM-PHD filter can give better results than many existing approaches.

Acknowledgment

We would like to acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC), grant references EP/K009931, EP/J015180 and a James Watt Scholarship.

REFERENCES

Babenko, B., Yang, M. H., and Belongie, S. (2011). Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632.

Chen, Z., Hong, Z., and Tao, D. (2015). An experimental survey on correlation filter-based tracking. *CoRR*, abs/1509.05520.

Danelljan, M., Hager, G., Shahbaz Khan, F., and Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255.

Diehl, C. P. and Cauwenberghs, G. (2003). SVM incremental learning, adaptation and optimization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2685–2690 vol.4.

Dinh, T. B., Yu, Q., and Medioni, G. (2014). Co-trained generative and discriminative trackers with cascade particle filter. *Comput. Vis. Image Underst.*, 119:41–56.

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

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*.

Grabner, H., Leistner, C., and Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 234–247.

Han, B., Comaniciu, D., Zhu, Y., and Davis, L. S. (2008). Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1186–1197.

Hare, S., Saffari, A., and Torr, P. H. S. (2011). Struck: Structured output tracking with kernels. In *2011 International Conference on Computer Vision*, pages 263–270.

Henriques, J. a. F., Caseiro, R., Martins, P., and Batista, J. (2012). Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 702–715.

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

Idrees, H., Warner, N., and Shah, M. (2014). Tracking in dense crowds using prominence and neighborhood

Figure 4: Qualitative results of our proposed LCMHT algorithm, CF2 (Ma et al., 2015a), MEEM (Zhang et al., 2014), LCT (Ma et al., 2015b) and KCF (Henriques et al., 2015) on PETS 2009 medium (left, frame 78) and dense (right, frame 85) data sets, just after occlusion by cropping and enlarging.

motion concurrence. *Image and Vision Computing*, 32(1):14.

Jia, X., Lu, H., and Yang, M. H. (2012). Visual tracking via adaptive structural local sparse appearance model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829.

Kalal, Z., Mikolajczyk, K., and Matas, J. (2012). Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422.

Kratz, L. and Nishino, K. (2012). Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):987–1002.

Li, Y. and Zhu, J. (2015). *A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration*, chapter Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II.

Ma, C., Huang, J. B., Yang, X., and Yang, M. H. (2015a). Hierarchical convolutional features for visual tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3074–3082.

Ma, C., Yang, X., Zhang, C., and Yang, M. H. (2015b). Long-term correlation tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5388–5396.

Rifkin, R., Yeo, G., and Poggio, T. (2003). Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, 190:131–154.

Ristic, B., Clark, D. E., Vo, B.-N., and Vo, B.-T. (2012). Adaptive target birth intensity for PHD and CPHD filters. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1656–1668.

Rodriguez, M., Sivic, J., Laptev, I., and Audibert, J.-Y. (2011). Density-aware person detection and tracking in crowds. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1):125–141.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR*.

Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2014). Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468.

van de Weijer, J., Schmid, C., Verbeek, J., and Larlus, D. (2009). Learning color names for real-world applications. *Trans. Img. Proc.*, 18(7):1512–1523.

Vedaldi, A. and Lenc, K. (2015). MatConvNet – convolutional neural networks for matlab. In *Proceedings of the 25th annual ACM international conference on Multimedia*.

Vo, B.-N. and Ma, W.-K. (2006). The Gaussian mixture probability hypothesis density filter. *Signal Processing, IEEE Transactions on*, 54(11):4091–4104.

Wang, L., Ouyang, W., Wang, X., and Lu, H. (2015). Visual tracking with fully convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3119–3127.

Wang, N. and Yeung, D.-Y. (2013). Learning a deep compact image representation for visual tracking. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 809–817.

Wu, Y., Lim, J., and Yang, M. H. (2013). Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418.

Zhang, J., Ma, S., and Sclaroff, S. (2014). MEEM: robust tracking via multiple experts using entropy minimization. In *Proc. of the European Conference on Computer Vision (ECCV)*.

Zhang, T., Ghanem, B., Liu, S., and Ahuja, N. (2012). Robust visual tracking via multi-task sparse learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2042–2049.

Zhong, W., Lu, H., and Yang, M. H. (2012). Robust object tracking via sparsity-based collaborative model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845.