Online Adaptive Hidden Markov Model for Multi-Tracker Fusion

Tomas Vojir, Jiri Matas
The Center for Machine Perception, FEE CTU in Prague
Karlovo namesti 13, 121 35 Prague 2, Czech Republic
{vojirtom, matas}@cmp.felk.cvut.cz

Jana Noskova
Faculty of Civil Engineering, CTU in Prague
Thakurova 7/2077, 166 29 Prague 6, Czech Republic
noskova@fsv.cvut.cz

April 24, 2015

Abstract

In this paper, we propose a novel method for visual tracking called HMMTxD. The method fuses information from complementary trackers and a detector by utilizing a hidden Markov model whose latent states correspond to a binary vector expressing the failure of individual trackers. The Markov model is trained in an unsupervised way, relying on an online learned detector to provide a source of tracker-independent information for a modified Baum-Welch algorithm that updates the model w.r.t. the partially annotated data.

We show the effectiveness of this approach on combination of two and three tracking methods. The performance of HMMTxD is evaluated on two standard benchmarks (CVPR2013 and VOT) and on a rich collection of 77 publicly available sequences. The HMMTxD outperforms the state-of-the-art, often significantly, on all datasets in almost all criteria.

1 Introduction

In the last thirty years, a large number of diverse visual tracking methods has been proposed [28, 22]. The methods differ in the formulation of the problem, assumptions made about the observed motion, in optimization techniques, the features used, in the processing speed, and in the application domain. Some methods focus on specific challenges like tracking of articulated or deformable objects [14, 5, 3], occlusion handling [6], abrupt motion [31] or long-term tracking [18, 10].

Three observations motivate the presented research. First, most trackers perform poorly if run outside the scenario they were designed for. Second, some trackers make different and complementary assumptions and their failures are not highly correlated (called complementary trackers in the paper). And finally, even fairly complex well performing trackers run at frame rate or faster on standard hardware, opening the possibility for multiple trackers to run concurrently and yet in or near real-time.

We propose a novel methodology that exploits a hidden Markov model (HMM) for fusion and pose prediction of multiple complementary trackers using an on-line learned high-precision detector. The HMM, trained in an unsupervised manner, estimates the state of the trackers – failed, operates correctly – and outputs the pose of the tracked object taking into account the past performance and current observations of the trackers and the detector. The HMM treats the detector output as correct; for the potentially many frames where reliable detector output is not available, it combines the trackers. The detector is trained on the first image and interacts with the learning of the HMM by partially annotating the sequence of HMM states in the time of verified detections. The recall of the detector is not critical but it affects the learning rate of the HMM and the long-term properties of the HMMTxD method, i.e. its ability to reinitialize trackers after occlusions or object disappearance.

Related work. The most closely related approaches include Santner et al. [21], where three tracking methods with different rates of appearance adaptation are combined to prevent drift due to incorrect model updates. The approach uses simple, hard-coded rules for tracker selection. Kalal et al. [10] combine a tracking-by-detection method with a short-term tracker that generates so called P-N events to learn new object appearance. The output is defined either by the detector or the tracker based on visual similarity to the learned object model. Both these methods employ pre-defined rules to make decisions about object pose and use
one type of measurement, a certain form of similarity between the object and the estimated location. HMM-TxD learns continuously and causally the performance statistics of individual parts of the systems and fuses multiple “confidence” measurements in the form of probability densities of observables in the HMM. Zhang et al. [30] use a pool of multiple classifiers learned from different time spans and choose the one that maximize an entropy-based cost function. This method addresses the problem of model drifting due to wrong model updates, but the failure modes inherent to the classifier itself remains the same. This is unlike the proposed method which allows to combine diverse tracking methods with different inherent failure modes and with different learning strategies to balance their weaknesses.

Similarly to the proposed method, Wang et al. [25] and Bailer et al. [1] fuse different out-of-the-box tracking methods. Bailer et al. combine offline the outputs of multiple tracking algorithms. There is no interaction between trackers, which for instance implies that the method avoids failure only if one method correctly tracks the whole sequence. Wang et al. use a factorial hidden Markov model and a Bayesian approach. The state space of their factorial HMM is the set of potential object positions, therefore it is very large. The model contains a probability description of the object motion based on a particle filter. Trackers interact by reinitializing those with low reliability to the pose of the most confident one.

In contrast, the HMM-TxD method is online with tracker interaction via a high precision object detector that supervises tracker reinitializations which happen on the fly. Moreover, the HMM-TxD confidence estimation is motion-model free and this prevents biases towards support of trackers with a particular motion model.

Yoon et al. [29] combines multiple trackers in a particle filter framework. This approach models observables and transition behavior of individual trackers, but the trackers are self-adapting which makes it prone to wrong model updates. The adaptation of HMM-TxD model is supervised by a non-adaptive detector method set to a specific mode of operation – near 100% precision – alleviating the incorrect update problem.

The contributions of the paper are: a novel method for fusion of multiple trackers based on HMMs, a simple, and so far unused, unsupervised method for HMMs training in the context of tracking, tunable feature-based detector with very low false positive rate, and the creation of a tracking system that shows state-of-the-art performance.

The observable values are allowed to be from the (0, 1) interval with either a known or unknown distribution. The observables with an unknown distribution are modeled as beta-distributed random variables (Eq. 1). The beta distribution was chosen for its versatility, where practically any kind of unimodal random variable on (0, 1) can be modeled by the beta distribution, i.e. for any choice of any lower

\[ \lambda = (A, F) \]

\[ F(d) = x_{1}x_{2}x_{3}x_{4} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]

\[ B_{d} \]
and upper quantiles, a beta distribution exists satisfying the given quantile constraint [7]. HMM is parameterized by the pair $\lambda = (A,F)$, where $A$ are the probabilities of state transition and $F$ are the beta distributions with parameters of shape $p,q > 0$ and density for $x \in (0,1)$

$$f(x|p,q) = \frac{x^{p-1}(1-x)^{q-1}}{\int_0^1 u^{p-1}(1-u)^{q-1} du}. \quad (1)$$

Since our goal is real-time tracking without any specific pre-processing, learning of HMM parameters has to be done online. We utilize a detector which is set to operating mode with approximately zero false positive rate to partially annotate the sequence of hidden states. In contrast to classical HMM, where only a sequence of observations $X = \{X_t\}_{t=1}^T, X_t = (x_1,x_2,\ldots,x_n)$ is available, we are in a semi-supervised setting and have a time sequence $0 = t_0 < t_1 < t_2 \ldots < t_K \leq T$ of observed states of a Markov chain $S = \{S_{t_k} = s_{i_k}, (t_k)_{k=1}^K\}$, and Markov chain starting again in state $s_1$ at any time $\{t_k + 1, 0 \leq k \leq K\}$. This information is provided by the detector, where $\{t_k\}_{k=1}^K$ is a sequence of detection times. The HMM parameters are learned by a modified Baum-Welch algorithm run on the observations $X$ and the annotated sequence of states $S$. The partial annotation and HMM parameter estimation update is done strictly online.

The output of the HMMTracker is an average bounding box of correct trackers of the current most probable state $s_t^*$. For $t_{(k-1)} < t < t_k, 1 \leq k \leq K$ the forward-backward procedure [2] for HMM is used to calculate the marginal probability of each state at time $t$ and the state $s_t^* \in \{0,1\}^n \setminus \{(0,0,\ldots,0)\}$ is the state for which

$$P(S_t = s_t | X_1,\ldots,X_t,S_{t_1},\ldots,S_{t_{(k-1)}},\lambda) \quad (2)$$

is maximal. For $t_K < t \leq T$ the Eq. 2 holds with $t_{(k-1)} = t_K$. This ensures that the algorithm outputs a pose for each frame which is required by most benchmark protocols. Illustration of the tracking process and HMM insight is shown in Fig. 2. Theoretically the parameters of HMM could be updated after each frame. However, in our implementation, learning takes place only at frames where the detector positively detects the object, i.e. the sequence of states starting and ending with observed state inferred by the detector. ¹

### 3 Learning the Hidden Markov Model

For learning of the parameters $\lambda$ of the HMM a MLE inference is employed, however maximizing the likelihood function $P(X,S|\lambda)$ is a complicated task that cannot be solved analytically. In the proposed method, the Baum-Welch algorithm [2] is used. The Baum-Welch algorithm is a widespread iterative procedure for estimating parameters of HMM where each iteration increases the likelihood function but, in general, the convergence to the global maximum is not guaranteed. The Baum-Welch algorithm is in fact an application of the EM (Expectation-Maximization) algorithm [4].

Let us assume the HMM with $N$ possible states $\{s_1,s_2,\ldots,s_N\}$, the matrix of state transition probabilities $A = \{a_{ij}\}_{i,j=1}^N$, the vector of initial state probabilities $\pi = (1,0,0,\ldots,0)$, the initial state $s_1 = (1,1,\ldots,1)$, a sequence of observations $X = \{X_t\}_{t=1}^T, X_t \in R^m$ and $F = \{f_t(x)\}_{t=1}^N$ the system of conditional probability densities of observations conditioned on $S_t = s_t$

$$f_t(x) = f(x|S_t = s_t) \quad (3)$$

where $S_t$ are random variables representing the state at time $t$, and $\lambda = (A,F)$ is denoting the parameter set of the model.

Let us denote

$$Q(\lambda, \lambda') = \sum_{s \in S} P(s|X, \lambda) \log[P(s|X|\lambda')], \quad (4)$$

where $S = \{s_1,s_2,\ldots,s_N\}^T$ is a set of all possible $T$-tuples of states and $s \in S, s = (s_1,\ldots,s_T)$ is one sequence of states. According to Theorem 2.1. in [2]

$$Q(\lambda, \lambda') \geq Q(\lambda, \lambda) \Rightarrow P(X|\lambda') \geq P(X|\lambda) \quad (5)$$

and the equality holds if and only if $P(s|X, \lambda) = P(s|X, \lambda')$ for $s \in S$. The classical Baum-Welch algorithm repeats the following steps until convergence:
1. Compute $\lambda^* = \arg \max_\lambda Q(\lambda_n, \lambda)$

2. Set $\lambda_{n+1} = \lambda^*$.

The modified Baum-Welch algorithm exploits the partially annotated sequence of states, where the known states are inferred from the detector output. Let $0 = t_0 < t_1 < t_2 \ldots < t_K \leq T$ be a sequence of detection times, $S = \{S_{t_k} = s_{i_k}, (t_k)_{k=1}^K\}$ be observed states of Markov chain, marked by the detector, and $S_{t+1} = s_1$ for $0 \leq k \leq K$. So the sequence of observations of the HMM is divided into $K + 1$ independent subsequences, each with a fixed initial state $s_1$, the first $K$ subsequences with a known terminal state defined by the detector and the last subsequence with an unknown terminal state.

The following equations are obtained by employing the modification to the Baum-Welch algorithm,

$$\log[P(s, X, S|\lambda)] = \sum_{t=1}^{T-1} \log a_{s_{t}, s_{t+1}} + \sum_{t=1}^{T} \log f_{n_s}(X_t),$$

$$Q(\lambda_n, \lambda) = \sum_{s \in \Theta} P(s|X, S, \lambda_n) \sum_{t=1}^{T-1} \log a_{s_{t}, s_{t+1}} +$$

$$\sum_{s \in \Theta} P(s|X, S, \lambda) \sum_{t=1}^{T} \log f_{n_s}(X_t).$$

The maximization of the $Q(\lambda_n, \lambda)$ can be separated to maximization w.r.t. $A = \{a_{ij}\}_{i,j=1}^N$ by maximizing the first term and w.r.t. $F = \{f_s(x)\}_{i=1}^N$ by maximizing the second term.

The maximization w.r.t. $A$ constrained by $\sum_{j=1}^N a_{ij} = 1$ for $1 \leq i \leq N$ is obtained by re-estimating the parameters $\hat{a}_{ij} = \frac{\text{expected number of transitions from state } s_i \text{ to state } s_j}{\text{expected number of transitions from state } s_i}$.

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} (t \neq t_k, 1 \leq k \leq K) P(S_t = s_i, S_{t+1} = s_j|X, S, \lambda)}{\sum_{t=1}^{T-1} (t \neq t_k, 1 \leq k \leq K) P(S_t = s_i|X, S, \lambda)}.$$ 

This equation is computed using modified forward and backward variables of the Baum-Welch algorithm to reflect the partially annotated states. Due to paper page limitation, for the exact formulation of formulas for computation of $\hat{a}_{ij}$ please see supplementary material.

The maximization of Eq. 7 w.r.t. $F = \{f_s(x)\}_{i=1}^N$ depends on assumptions on the system of probability densities $F$. It is usually assumed (e.g. in [19, 2]) that $F$ is a system of probability distributions of the same type and differ only in their parameters.

In the HMMtxD the $m$-dimensional observed random variables $X_t = \{X^1_t, X^2_t, \ldots, X^m_t\} \in \mathbb{R}^m$ are assumed conditionally independent and to have the beta-distribution, so $f_i(x), 1 \leq i \leq N$ are products of $m$ one-dimensional beta distributions with parameters of shape $\{(p^i, q^i)\}_{i=1}^N$. In this case maximization of the second term of the Eq. 7 is an iterative procedure using inverse digamma function which is very computationally expensive [7].

We estimate the shape parameters of the beta distributions with a generalized method of moments. The classical method of moments is based on the fact that sample moments of independent observations converge to its theoretical ones due to the law of large numbers for independent random variables. In the HMMtxD observations $X = \{X_t\}_{t=1}^T$ are not independent. The generalized method of moments is based on the fact that $\{X_t - E(X_t|X_1, X_2, \ldots, X_{t-1})\}_{t=1}^T$ is a sequence of martingale differences for which the law of large numbers also holds. Using the modified method of moments gives estimates of the parameters of shape

$$\hat{p}^i_i = \hat{\mu}^i_i \left(\frac{\hat{\mu}^i_i (1 - \hat{\mu}^i_i)}{(\hat{\sigma}^i_i)^2} - 1\right)$$

and

$$\hat{q}^i_i = (1 - \hat{\mu}^i_i) \left(\frac{\hat{\mu}^i_i (1 - \hat{\mu}^i_i)}{(\hat{\sigma}^i_i)^2} - 1\right)$$

where

$$\hat{\mu}^i_i = \frac{\sum_{t=1}^{T} X^i_t P(S_t = s_i|X, S, \lambda)}{\sum_{t=1}^{T} P(S_t = s_i|X, S, \lambda)}$$

and

$$(\hat{\sigma}^i_i)^2 = \frac{\sum_{t=1}^{T} (X^i_t - \hat{\mu}^i_i)^2 P(S_t = s_i|X, S, \lambda)}{\sum_{t=1}^{T} P(S_t = s_i|X, S, \lambda)}.$$ 

The generalised method of moments is described in detail in a supplementary material. Let us denote the system of probability densities with re-estimated parameters $\hat{F} = \{\hat{f}_i(x)\}_{i=1}^N$.

The complete modified Baum-Welch algorithm is summarized in Alg. 1, where after each iteration $P(X, S|\lambda_{n+1}) \geq P(X, S|\lambda_n)$ and we repeat these steps until convergence. Note that $A_n$ is a maximum likelihood estimate of $A$ therefore always increases $P(X, S|\lambda_n)$ (shown in [19]) but $F_n$ is estimated by the method of moments so the test on likelihood increase is needed (if statement in Alg. 1). In fact this algorithm structure corresponds to the generalised EM algorithm (GEM) introduced in [4].

### 4 Feature-Based Detector

The requirements for the detector are: adjustable operation mode (e.g. set for high precision but possibly low recall), (near) real-time performance and the ability to model
Algorithm 1: Algorithm for HMM parameters learning

| Input: $\mathcal{X}, S, \lambda_n = (A_n, F_n)$ |
|-----------------------------------------------|
| Output: $\lambda_{n+1} = (A_{n+1}, F_{n+1})$ |
| repeat |
| Compute likelihood $P(\mathcal{X}, S | \lambda_n)$ |
| Estimate $\hat{A}_n$ by Eq. 8 and $\hat{F}_n$ by Eq. 9, 10 |
| if $P(\mathcal{X}, S | \hat{A}_n, \hat{F}_n) < P(\mathcal{X}, S | A_n, F_n)$ then |
| $\lambda_{n+1} = (\hat{A}_n, \hat{F}_n)$ |
| else |
| $\lambda_{n+1} = (A_n, F_n)$ |
| $\lambda_n = \lambda_{n+1} = (A_{n+1}, F_{n+1})$ |
| until convergence $\lor$ max number of iteration |

pose transformations up to at least similarity (translation, rotation, isotropic scaling). Basically, any detector-like approach can be used and it may vary based on application. We choose to adapt a feature-based detector which has been shown to perform well in image retrieval, object detection and object tracking [18] tasks.

There are many possible combinations of features and their descriptors with different advantages and drawbacks. We exploit multiple feature types: specifically, Hessian key-points with the SIFT [15] descriptor, ORB [20] with BRISK and ORB with FREAK [17]. Each feature type is handled separately, up to the point where point correspondences are established. A weight is assigned to each feature type $w^g$ and is set to be inversely proportional to the number of features on the reference template, to balance the disparity in individual feature numbers.

The detector works as follows. In the initialization step, features are extracted from the inside and the outside of the region specifying the tracked object. Descriptors of the features outside of the region are stored as the background model. Usually, the input region is not 100% occupied by the target; therefore, fast color segmentation [12] is employed to remove the features that are most likely not on the target. Additionally, for each target feature, we use a normal distribution to model the similarity of the feature to other features. This allows defining the quality of correspondence matches in a probabilistic manner for each feature, thus getting rid of global static threshold for the acceptable correspondence distance.

In the detection phase, features are detected and described in the whole image. For each feature $g_i$ from the image the nearest neighbour (in Euclidean space or in Hamming distance metric space, depending on the feature type) feature $b^*$ from the background model and the nearest neighbour feature $f^*$ from the foreground model are computed. A tentative correspondence is formed if the feature match passes the second nearest neighbour test and a probability that the correspondence distance belongs to the outlier distribution is lower than a predefined significance set to 0.1%. So

$$\frac{d(g_i, f^*)}{d(g_i, b^*)} < 0.8 \land F(d(g_i, f^*) | \mu_f^*, \sigma_f^*) < 0.1\% \quad (12)$$

where $F(d | \mu_f^*, \sigma_f^*)$ is a c.d.f. of the normal distribution with parameters $\mu_f^*$ and $\sigma_f^*$ of a distance distribution of features not corresponding to $f^*$. The 0.1% significance corresponds to the $\mu - 3\sigma$ threshold. Finally, RANSAC estimates the target current pose using a sum of weighted inliers as a cost function

$$\text{cost} = \sum_i w^{g_i} * [g_i == \text{inlier}], \quad (13)$$

which takes into account the different numbers of features per feature type on the target.

The decision whether the detected pose is considered correct depends on the number of weighted inliers that supports the RANSAC-selected transformation and it controls the trade-of between precision and recall of the method. The threshold for this decision was set experimentally to have the false positive rate close to zero for the majority of the testing sequences. Furthermore, majority voting is used to verify that the detection is not in contradiction to the estimated HMM state, i.e. if we are in the state where two or more trackers are correct and the detector is not consistent with them, the detection is not used. The true and false positives for 77 sequences are shown in Fig. 3, where the detector works on almost all sequences with zero false positive rate (0.34% average false positive rate on the dataset) and 30% recall rate. The failure cases of this feature-based detector are mostly caused by the imprecise initial bounding box, which contains large portion of structured background (i.e. background where the detector finds features) and due to the presence of similar object in the scene, e.g. sequences hand2, basketball, football.

5 HMMTx3D Implementation

To demonstrate the performance of the proposed framework, a pair and a triplet of published short-term trackers were plugged into the framework to show the performance gain by combination of a different number of trackers. As Bailer et al. [1] pointed out, not all trackers when combined can improve the overall performance (i.e. adding tracking method with similar failure mode will not benefit).

We therefore choose methods that have a different designs and work with different assumptions (e.g. rigid global motion vs. color mean-shift estimation vs. maximum correlation response). These trackers are the Flock of Trackers (FoT) [23], scale adaptive mean-shift tracker (ASMS) [24] and kernelized correlation filters (KCF) [9]. This choice
shows that superior performance can be achieved by using simple, fast trackers (above 100fps) that may not represent the state-of-the-art. The trackers can be arbitrarily replaced depending on the user application or requirements.

Trackers

The Flock of Trackers (FoT) [23] evenly covers the object with patches and establishes frame-to-frame correspondence by the Lucas-Kanade method [16]. The global motion of the target is estimated by RANSAC. The second tracker is a scale adaptive mean-shift tracker (ASMS) [24] where the object pose is estimated by minimizing the distance between RGB histograms of the reference and the candidate bounding box. The KCF [9] tracker learns a correlation filter by ridge regression to have high response to target object and low response on background. The correlation is done in the Fourier domain which is very efficient.

These three trackers have been selected since they are complementary by design. FoT enforces a global motion constrain and works best for rigid object with texture. On the other hand, ASMS does not enforce object rigidity and is well suited for articulated or deformable objects assuming their color distribution is discriminative w.r.t. the background. KCF is basically tracking-by detection using sliding window like scanning.

For each tracker position, two global observable measurements are computed, namely the Hellinger distance between the target template histogram and the histogram of the current position and normalized cross-correlation score of the current patch and the target model patch. These target models are initialized in the first frame and then updated exponentially with factor of 0.5 during each positive detection of the detector part. Additionally, each tracker produces its own estimate of performance. For FoT it is the number of predicted correspondences (for details please see [23]) that support the global model. For ASMS it is the Hellinger distance between its histogram model and current neighborhood background (i.e. color similarity of the object and background) and for KFC it is a correlation response of the tracking procedure.

6 Experiments

The HMMtxD was compared with state-of-the-art methods on two standard benchmarks and on a dataset TV77 containing 77 public video sequences collected from tracking-related publications. The dataset exhibits wider diversity of content and variability of conditions than the benchmarks. Parameters of the method were fixed for all the experiments. In the HMM, the initial beta distribution shape parameters were set to $2, 1$ for all observations and the transition matrix was set to prefer staying in the current state (for exact numbers please see supplementary materials).

First, we compare the performance of individual parts of the HMMtxD framework (i.e. KCF, ASMS, FoT trackers) and their combination via HMM as proposed in this paper. Two variants of HMMtxD are evaluated – 2-HMMtxD refers to combination of FoT and ASMS trackers and the 3-HMMtxD to combination of all mentioned trackers. The Figure 4 shows the benefits gained by the combination of the trackers and a consistent increase in performance when

Figure 3: Frames with the detections for 77 sequences dataset. The green marks show the true positive detection and red marks are false positive. The blue line shows the recall of the detector and blue dashed line shows the average recall over all sequences. The length of each sequence is normalized to range $(0, 100)$. 

Figure 4 shows the benefits gained by the combination of the trackers and a consistent increase in performance when

---

6 Please see supplementary material.
the new complementary state of the art tracker (KCF) is added. In all other experiments, the abbreviation HMMTxD refers to the combination of all 3 trackers.

**Evaluation on the CVPR2013 Benchmark** [26] that contains 50 video sequences. Results on the benchmark have been published for about 30 trackers. The benchmark defines three types of experiments: (i) one-pass evaluation (OPE) – a tracker initialized in the first frame is run to the end of the sequence, (ii) temporal robustness evaluation (TRE) – the tracker is initialized and starts at a random frame, and (iii) spatial robustness evaluation (SRE) – the initialization is perturbed spatially. Performance is measured by precision (spatial accuracy, i.e. center distance of ground truth and reported bounding box) and success rate (the number of frames where overlap with the ground truth was higher than a threshold). The results are visualized in Fig. 5 where only results of the 10 top performing trackers are plotted. The proposed HMMTxD outperforms all trackers in the success rate in all three experiments. Its precision is comparable to MEEM [30], the top performing tracker. HMMTxD outperforms significantly the OPE results reported in Wang et al. [25], where 5 top performing trackers from this particular benchmark were used for combination (other experiments were not reported in the paper).

**VOT2013 benchmark** [13] evaluates trackers on a collection containing 16 sequences carefully selected from a large pool by a semi-automatic clustering method. For comparison, results of 27 tracking methods are available. The performance is measured by accuracy, average overlap with the ground truth, and robustness, the number of re-initialization of the tracker so that it is able to track the whole sequence. Average rank of trackers is used as an overall performance indicator.

In this benchmark, the proposed HMMTxD achieves clearly the best accuracy (Fig. 6). With less than one re-initialization per sequence it performs slightly worse in terms of robustness due to two reasons.

Firstly, the HMM recognizes a tracker problem with a delay and switching to other tracker (here even one frame where the overlap with ground truth is zero leads to penalization) and secondly the VOT evaluation protocol, which require re-initialization after failure and to forget all previously learned models (the VOT2013 refer to this as causal tracking), therefore the learned performance of the trackers is forgotten and has to be learned from scratch.

The results for the baseline experiment are shown in Fig. 6. Note that the ranking of the methods differs from the original publication since new methods (HMMTxD and MEEM) were added and the relative ranking of the methods changed. The top three performing trackers and their average ranks are HMMTxD (8.77), PLT (9.24), LGTpp [27] (10.11). MEEM tracker ends up at the fifth place with average rank 10.87. The rankings were obtained by the toolkit provided by the VOT in default settings for baseline and region noise experiments.

The second best performing method on the VOT2013 is the unpublished PLT for which just a short description is available in [13]. PLT is a variation of structural SVM that
uses multiple features (color, gradients). STRUCK [8] and MEEM [30] are similar methods to the PLT based on SVM classification. We compared these methods with HMMtxD on the diverse 77 videos along with the TLD [10] which has a similar design as HMMtxD. HMMtxD outperforms all these methods by a large margin on average recall – measured as number of frames where the tracker overlap with ground truth is higher than 0.5 averaged over all sequences. Results are shown in Fig. 7.

7 Conclusions

A novel method called HMMtxD for fusion of multiple trackers has been proposed. The method utilizes an on-line trained HMM to estimate the states of the individual trackers. The HMMtxD outperforms its constituent parts (FoT, ASMS, KCF, Detector) by a large margin and shows the efficiency of the HMM tracker fusion on combination of two and three trackers.

HMMtxD outperforms all 29 methods included in the CVPR2013 benchmark in success rate while performing competitively in terms of precision. The HMMtxD also outperforms all method of the VOT2013 benchmark in accuracy, while maintaining very good robustness, and ranking in the first place in overall ranking. Experiments conducted on a diverse dataset TV77 show that HMMtxD outperforms state-of-the-art MEEM, STRUCK and TLD methods, which are similar in design, by a large margin. The processing speed of the HMMtxD is 5 – 10 frames per second on average, which is comparable with other complex tracking methods.

References

[1] C. Bailer, A. Pagani, and D. Stricker. A superior tracking approach: Building a strong tracker through fusion. In European Conference on Computer Vision, Lecture Notes in Computer Science, 2014. 2, 5
[2] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The Annals of Mathematical Statistics, 41(1):164–171, 02 1970. 3, 4
[3] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(4):941–953, Apr. 2013. 1
[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, series B, 39(1):1–38, 1977. 3, 4
Figure 7: Evaluation of state-of-the-art trackers on the TV77 dataset in terms of recall, i.e. number of correctly tracked frames. The average recall is shown by the dashed lines (precise number is in the legend). Black circles mark the grayscale sequences. The sequences are ordered by HMMTxD performance.

[5] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In International Conference on Computer Vision, 2011.

[6] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In Computer Vision and Pattern Recognition, pages 1285–1292, June 2010.

[7] A. Gupta and S. Nadarajah. Handbook of Beta Distribution and Its Applications. Statistics: A Series of Textbooks and Monographs. 2004.

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

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. Pattern Analysis and Machine Intelligence, 2015.

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

[11] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin. A Novel Performance Evaluation Methodology for Single-Target Trackers. ArXiv e-prints, Mar. 2015.

[12] M. Kristan, J. Perš, V. Sulic, and S. Kovacic. A graphical model for rapid obstacle image-map estimation from unmanned surface vehicles. In Asian Conference on Computer Vision. Accepted, to be published, 2014.

[13] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir, and et al. The visual object tracking vot2013 challenge results. In The IEEE International Conference on Computer Vision (ICCV) Workshops, June 2013.

[14] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In Computer Vision and Pattern Recognition, pages 1208–1215. 2009.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, Nov. 2004.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In International Joint Conference on Artificial Intelligence, 1981.

[17] R. Ortiz. Freak: Fast retina keypoint. In Conference on Computer Vision and Pattern Recognition, pages 510–517. Washington, DC, USA, 2012.

[18] F. Fernici and A. D. Bimbo. Object tracking by oversampling local features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 99(PrePrints):1, 2013.

[19] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, Feb 1989.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In International Conference on Computer Vision, ICCV ’11, pages 2564–2571. Washington, DC, USA, 2011.

[21] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST Parallel Robust Online Simple Tracking. In Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010.

[22] A. Smeulder, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah. Visual tracking: an experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov. 2013.

[23] T. Vojir and J. Matas. The enhanced flock of trackers. In Registration and Recognition in Images and Videos, volume 532 of Studies in Computational Intelligence, pages 113–136. January 2014.

[24] T. Vojir, J. Noskova, and J. Matas. Robust scale-adaptive mean-shift for tracking. In Image Analysis, volume 7944 of Lecture Notes in Computer Science, pages 652–663. 2013.

[25] N. Wang and D. yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In T. Je-
bara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning*, pages 1107–1115, 2014.

[26] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition*, pages 2411–2418, June 2013.

[27] J. Xiao, R. Stolkin, and A. Leonardis. An enhanced adaptive coupled-layer tracker++. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 137–144, Dec 2013.

[28] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), Dec. 2006.

[29] J. H. Yoon, D. Y. Kim, and K.-J. Yoon. Visual tracking via adaptive tracker selection with multiple features. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV, ECCV’12*, pages 28–41, 2012.

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

[31] X. Zhou and Y. Lu. Abrupt motion tracking via adaptive stochastic approximation Monte Carlo sampling. In *Computer Vision and Pattern Recognition*, pages 1847–1854, June 2010.