Correlation-based self-correcting tracking

Tewodros A. Biresaw\textsuperscript{a,b,*}, Andrea Cavallaro\textsuperscript{b}, Carlo S. Regazzoni\textsuperscript{a}

\textsuperscript{a}DITEN, University of Genova, Italy
\textsuperscript{b}Centre for Intelligent Sensing, Queen Mary University of London, UK

Abstract

We present a framework for improving probabilistic tracking of an extended object with a set of model points. The framework combines the tracker with an on-line performance measure and a correction technique. We correlate model point trajectories to improve on-line the accuracy of a failed or an uncertain tracker. A model point tracker gets assistance from neighboring trackers whenever a degradation in its performance is detected using the on-line performance measure. The correction of the model point state is based on correlation information from the state of other trackers. Partial Least Square (PLS) regression is used to model the correlation of point tracker states from short windowed trajectories adaptively. Experimental results on data obtained from optical motion capture systems show the improvement in tracking performance of the proposed framework compared to the baseline tracker and other state-of-the-art trackers.

Keywords: Object tracking, data association, online performance measure, correction, trajectory correlation.

1. Introduction

Tracking plays a fundamental role in surveillance \cite{1}, computer vision \cite{2}, human-computer interaction \cite{3} and medical image processing \cite{4}. A wide variety of tracking techniques have been proposed such as Mean-Shift tracker \cite{5},
Kalman filter tracker [6] and KLT tracker [7] (surveys on tracking are available in [8, 9, 10]). The target state representation and tracking challenges vary from one application domain to another. For example, an extended object can be represented by a set of points estimated from multiple independent measurements [11, 8]. The movements of the points enable us to analyze the overall shape evolution and sub-part dynamics, such as the movements of hands and legs relative to other body parts of a person [12, 13]. Local appearance is modeled using pre-selected points on the object, which we refer to as model points, such as markers in a motion capture system [14], or features extracted from images using Scale-Invariant Feature Transform (SIFT) [15].

Tracking model points is achieved by estimating the state of each point individually, which generates a challenge for data association [11]. Moreover, performance degradation or failures in tracking can be generated by the challenges related to data association, missed and false detections, illumination changes and occlusions [16]. A tracking failure at any instant of time can generate a long-term failure due to the use of first-order Markov processes and model updates [17, 18]. Except [19, 20, 21], most trackers do not explicitly detect failures and correct them. A performance measure to detect tracking failures and a tracking correction step are desirable to obtain robust tracking [22, 23, 24]. Since comparing the tracker’s output to the ground-truth data is not applicable for real-time systems [25, 26], there is the need for an efficient and robust framework for online track verification and correction [19, 27].

We propose a Track-Evaluate-Correct (TEC) framework based on Bayesian filtering of model points. Tracking is obtained using a baseline tracker. Evaluation and correction judge the track quality and apply appropriate changes to the baseline tracker for improving its performance. As a novelty, we make model point trackers to assist each other based on their evaluation and a correlation model in the TEC framework. In particular, we propose a quality measure criterion for evaluation of each model point track to produce a decision for correction. Correction of low-quality model point tracker involves an estimation of a probable true state and a re-initialization of the tracker using the correlation model with other point trackers. The correlation between point trackers is modeled from observed trajectory histories adaptively based on the result of the quality measure. Unlike our previous work on performance evaluation using time-reversed Markov chain [28] and trajectory correlation [29], in this work we take a binary decision on the
trackers by examining their states, and we use an online modeling and a correlated trajectories selection criterion for effectively recovering low-quality trackers. The proposed TEC framework is shown in Fig. 1.

The paper is organized as follows: Section 2 formulates the problem. Related works on evaluation and correction for tracking are reviewed in Section 3. Section 4 discusses the algorithm for tracking model points. The proposed performance measure and correction technique are presented in Sections 5 and 6, respectively. Section 7 discusses the experiments and Section 8 concludes the paper.

2. Problem formulation

Tracking involves estimating the states $X = \{X_t\}_{t=1}^\tau$ of the target over time from a set of available measurements $Z = \{Z_t\}_{t=1}^\tau$. $\tau$ is the trajectory duration, while $X_t$ and $Z_t$ are the estimated state and the measurement, respectively, at time $t$. Let us define $T(\cdot)$ to represent a tracker that estimates $X_t$ as

$$X_t = T(Z_t, \zeta_t), \quad (1)$$
where $\zeta_t$ is other inputs to the tracker such as the previous state $X_{t-1}$ for Bayesian trackers (Fig. 1). For an extended object, the state $X_t$ consists of the state (and identity) of each model point

$$X_t = \{x^l_t : 1 \leq l \leq N_t, l \in \mathbb{N}^+\},$$

(2)

where $x^l_t$ is the state of model point $l$ and $N_t$ is the number of estimated model points. For implementations with initiation and termination of model points, $N_t$ is variable over time. Similarly, at each time the sensor or feature extractor produces $M_t$ point measurements

$$Z_t = \{\hat{z}^\ell_{t} : 1 \leq \hat{\ell} \leq M_t, \hat{\ell} \in \mathbb{N}^+\},$$

(3)

where $\hat{z}^\ell_{t}$ is the $\hat{\ell}th$ model point measurement. The measurements $Z_t$ are unlabeled, and are affected by potential misdetections and clutter.

Individual allocated trackers for each model point are local trackers $T^l$. The state vector $x^l_t$ depends on the type of motion model used in the tracking method. A typical state vector for a $D$-dimensional tracking problem contains the position and velocity components of the model point as $x^l_t = [x^l_t, \dot{x}^l_t, ... x^l_t, \dot{x}^l_t, \dot{x}^l_t, ...]^T$.

The quality of the tracking result depends on how close the estimated state is to the actual (true) state. A performance measure on the tracker $T^l$ and estimated states $x^l_t$ is represented as

$$p^l_t = \Phi(x^l_t, T^l, I_p), \quad 1 \leq l \leq N_t,$$

(4)

where $\Phi(\cdot)$ is the operation made to obtain a set of predefined classes or numerical values $p^l_t$ for the track quality or the tracker performance measure. $I_p$ is any other information, other than the current estimated state of the tracker, such as pre-defined threshold values and reference data, which are used by the performance measure.

When a tracker generates a low-quality output, a correction mechanism can be employed. The correction is applied to the tracker and the track based on a decision from the result of the quality measure $p = \{p^l_t\}_{l=1}^{N_t}$. For the identified low-quality tracks, let us define their performance value as $p^l_t = 1$. The correction step aims to modify the tracker $T^l$ and to improve the accuracy of the estimated states $x^l_t$ as

$$\hat{T}^l, \hat{x}^l_t = \begin{cases} \Theta(x^l_t, T^l, p^l_t, I_c), & \text{if } p^l_t = 1, \\ T, x^l_t & \text{otherwise} \end{cases}$$

(5)
where $\Theta(\cdot)$ is the transformation made to obtain the corrected tracker $\hat{T}$ and the improved states $\hat{X}_t = \{\hat{x}_t^l\}_{l=1}^{N_t}$. $I_c$, which is similar to $I_p$, represents valid information such as a trajectory output and an online learned appearance model to assist the correction technique. Determining $\Phi(\cdot)$ and $\Theta(\cdot)$ for the tracker, together with the methods to obtain and use the side information, $I_p$ and $I_c$, plays an important role in the implementation of the overall TEC framework.

3. Prior work

The idea of track quality measurement and correction is used to improve the performance of baseline trackers. Various types of performance measures and correction techniques, i.e. $\Phi(\cdot)$ and $\Theta(\cdot)$, have been proposed for different trackers, and are discussed in this section. Table 1 summarizes the characteristics of trackers that use performance measures and/or correction techniques.

3.1. Performance measures

Quantifying the quality of a tracking result generally involves comparing one or more output variables of a tracker with reference data [25]. For an offline case, manually collected ground-truth data are used as a reference [25]. For online performance measures, standalone empirical methods judge the output of the tracker [26]. The characteristics of the output considered for evaluation include trajectory properties [19, 37, 39], objects color differences and boundary contrasts with background [33, 40, 41], observation likelihood [25, 32, 41] and innovation errors or covariances of the states [25, 31]. These characteristics are compared with thresholds and predefined target properties. Trajectory properties such as smoothness, length, change of direction, similarity with a predefined model and similarity with a reverse tracking result are considered for trajectory-based evaluation [31, 39, 42]. Histogram differences between the track output and the prior known reference of the target or temporal differences between track outputs are used as color-oriented performance measures. Although these performance measures are easy to implement, they may produce false positive results in the presence of illumination changes [31, 40]. Performance measures are comparable to online model validation of the assumed dynamic system [32, 39]. Model validation of the probabilistic trackers is directly estimated from their innovation error, likelihood and covariance magnitudes. However, similar available
Table 1: Comparison of trackers with performance measure and correction technique. State representations: P: point based, A: area based and O: other, “-”: not mentioned or not explicitly stated, Prop.: proposed framework

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Tracker</th>
<th>State</th>
<th>Performance measure</th>
<th>Correction method</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>[25]</td>
<td>Unscented Kalman Filter</td>
<td>P</td>
<td>detection performance, track accuracy (innovation error), track quality</td>
<td>-</td>
<td>relies on sensor’s measurement accuracy</td>
</tr>
<tr>
<td>[29]</td>
<td>PDAF [30]</td>
<td>P</td>
<td>trajectories correlation between trackers</td>
<td>-</td>
<td>states of trackers providing correlation information are not checked a priori</td>
</tr>
<tr>
<td>[31]</td>
<td>Particle Filter</td>
<td>A</td>
<td>spatial state uncertainty and time-reversed constraint</td>
<td>-</td>
<td>applied to trackers using multiple hypotheses strategy</td>
</tr>
<tr>
<td>[32]</td>
<td>Probabilistic Trackers</td>
<td>P</td>
<td>model validation, change detection</td>
<td>-</td>
<td>largely dependent on the system characteristics</td>
</tr>
<tr>
<td>[33]</td>
<td>Any Tracker</td>
<td>A</td>
<td>color change properties, log polar transformation</td>
<td>-</td>
<td>dependent of background color, sensitive to illumination changes</td>
</tr>
<tr>
<td>[34]</td>
<td>Mean-Shift</td>
<td>A</td>
<td>correlation information</td>
<td>-</td>
<td>correlation information fused to the tracker with no stated criteria</td>
</tr>
<tr>
<td>[27]</td>
<td>Particle Filter</td>
<td>A</td>
<td>appearance model update</td>
<td>-</td>
<td>update performed with no performance measure of the tracker</td>
</tr>
<tr>
<td>[18]</td>
<td>Particle Filter</td>
<td>A</td>
<td>appearance and motion model update</td>
<td>-</td>
<td>update done with no performance measure of the tracker</td>
</tr>
<tr>
<td>[37]</td>
<td>KLT</td>
<td>P</td>
<td>time-reversed constraint</td>
<td>-</td>
<td>spatial information is not preserved</td>
</tr>
<tr>
<td>[20]</td>
<td>Mean-Shift</td>
<td>A</td>
<td>compare to first detected model</td>
<td>information from detector</td>
<td>evaluation not robust for appearance changes</td>
</tr>
<tr>
<td>[21]</td>
<td>Particle Filter</td>
<td>O</td>
<td>misdetection</td>
<td>correlation information</td>
<td>correlation model learned off-line</td>
</tr>
<tr>
<td>[38]</td>
<td>Particle Filter</td>
<td>A</td>
<td>trained classifier for occlusion detection</td>
<td>use knowledge of unoccluded region</td>
<td>performance of tracker is measured indirectly by detecting occlusions</td>
</tr>
<tr>
<td>Prop.</td>
<td>PDAF [30]</td>
<td>P</td>
<td>predefined failure models and tracking challenges</td>
<td>trajectory correlation between trackers</td>
<td>performance measure based on the baseline tracker, adaptive correlation modeling for correction</td>
</tr>
</tbody>
</table>
measurements from other objects or clutter usually results in low-quality performance measures [31].

For tracking objects by a set of model points, the performance measure is based on the model point detection and matching quality [35, 36, 41] or their trajectory properties [19, 28, 37]. Performance measures of extended objects are based either on local point trackers [37] or on a global-object tracker [19]. A local point tracker has a low-quality performance when no new measurements are matched with it [35]. However, matching-based performance evaluation generates false positives and negatives due to similarities between descriptors, clutter and misdetections. Trajectory-based performance measures use time-reversed tracking to obtain a backwards trajectory for comparison with the original one [39]. The differences between the forward and the backward trajectories are used as a performance measure [28, 37, 39]. However, tracking back to the initial (or previous) frame leads to delays; moreover it is expensive for long-term trackers and for targets with large numbers of model points, such as markers for the human body in a motion capture system.

3.2. Correction

The correction step involves changing a tracking parameter [18], applying a suitable tracking principle or constraint over the baseline tracking algorithm in order to improve performance [38]. The most common correction techniques use adaptive methods [18, 27, 43]. The adaptive component of the system changes its tracking parameters, and the target appearance and motion models based on the result obtained from the tracker, usually from the most recent frames. These methods are prone to accumulated errors and cause drifts due to background clutter and false detections [16, 17]. Alternatively, tracker re-initialization is used for correction [19, 20, 44]. This approach needs the knowledge of the target, such as online learned detectors, while the tracker is well on the target [19] or known characteristics of the target at the initial frame [20].

Terminating uncertain and low-quality local tracks in the extended object is used to achieve robust tracking [37]. This process avoids likely tracking failures and wrong state information by the low-quality local trackers as part of the global object. This approach usually leads to the loss of structural information of the object when model points are assumed to be a persistent part of the object and the track of the points are inputs for high-level tasks such as activity analysis. After terminating poorly performing local trackers, new trackers are initiated [35, 36].
In a tracking system consisting of multiple trackers, correlation information among the trackers has been used as a correction technique [28, 34, 21]. In 3D articulated human motion tracking, correlation information is obtained from symmetric portions of the human body [21]. This information is used to estimate the motion prior and to constrain the proposal distribution of the particle filter when no measurement data are available. The model used to obtain correlation information is learned offline, which limits the generality of the solution. Although not explicitly stated as a correction technique, a similar type of correlation information is used to obtain robust tracking with a context-aware tracker [34]. The source of correlation information is tracking of other objects in the scene (auxiliary objects).

In some situations the information used for performance measures and corrections is used interchangeably. In [29], trajectory correlation information between feature trackers is used to produce the performance measures. From the observed trajectory correlation, reference data is estimated for a feature tracker using the states of other feature trackers. The evaluation compares error distances between the output state and the estimated reference data. However, in order to calculate the reference data for a particular feature tracker, the performance of other feature trackers needs to be checked.

4. Extended object tracker

For tracking multiple local points on an extended object, Bayesian filtering is used as a baseline tracker [30]. Bayesian tracking involves the estimation of the posterior distribution \( p(x^l_t|Z_{1:t}) \) for a model point \( l \) using a prediction step

\[
p(x^l_t|Z_{1:t-1}) = \int p(x^l_t|x^l_{t-1})p(x^l_{t-1}|Z_{t-1})dx^l_{t-1},
\]

followed by an update step

\[
p(x^l_t|Z_{1:t}) = \frac{p(Z_t|x^l_t)p(x^l_t|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})},
\]

where \( x^l_t \) and \( Z_t \) are defined in Eq. 2 and Eq. 3, respectively. The prediction and update formulations assume a first-order Markov process for \( x_t \) and conditionally independence for \( Z_t \) when the current state \( x_t \) is available. An analytical online solution to the above integral is obtained by assuming a linear-Gaussian model as a Kalman filter. For non-linear and non-Gaussian models numerical approximations such as Monte Carlo methods are applied [45].
For marginalizing analytically the state of the extended object, which is proportional to the number of local trackers, we choose Kalman filter as the baseline tracker, i.e. $T$ in Eq. 5, for each model point. Moreover, we assume that the motion of model points is not complex (highly non-linear), and a linear motion model and a linear measurement model with Gaussian noise are adequate for state estimation [30, 46]. The position and velocity components of the states for each local point $l$ in the Kalman filter are represented by the random variable $\mathbf{x}_l^t = [x_{l,1}, \dot{x}_{l,1}, \ldots, x_{l,D}, \dot{x}_{l,D}]^T$ for a $D$-dimensional tracking problem. For the state models a constant velocity is assumed. The prediction model (state transition) is defined by the Gaussian distribution as

$$p(\mathbf{x}_l^t | \mathbf{x}_{l}^{t-1}) = \mathcal{N}(F_{t} \mathbf{x}_{l}^{t-1}, Q_t),$$

where $\mathcal{N}(\cdot)$ is a $D$-dimensional Gaussian probability density function with mean $F_{t} \mathbf{x}_{l}^{t-1}$ and covariance matrix $Q_t$. The constant velocity model is defined by the matrix $F_{t}$ as

$$F_{t} = I_D \otimes \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},$$

where $I_D$ is a $D \times D$ identity matrix and $\otimes$ represents the tensor product on matrices. The measurement model is defined as

$$p(\mathbf{Z}_t | \mathbf{x}_l^t) = \mathcal{N}(H \mathbf{x}_l^t, R_t),$$

where $R_t$ is the covariance matrix that represents the uncertainty in the measurement and $H$ is the observation matrix, defined as

$$H = I_D \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}.$$
Validation region (Gaussian)

\[ \bar{x}_t \] - Predicted state
\[ \triangledown \] - Mean of Gaussian
\[ Z_t \] - Measurements
\[ \square \] - for state update
\[ \blacksquare \] - not for state update

Figure 2: Data association problem with multiple measurements in the validation region at the update stage of the Kalman filter.

each model point independently. The separate PDAF implementation to a model point considers the other model points’ measurements as clutter. PDAF allows the tracker to take into account the association uncertainty for measurements in the validation region. The soft decision of PDAF is based on Minimum Mean Square Error (MMSE), between the predicted state and measurements in the validation region. JPDAF is effective for tracking a known number of targets, which is considered in this paper, and in the presence of clutter [47]. However, its performance degrades significantly when neighboring targets and clutter produce persistent interference, and when misdetections occur. We improve the performance of JPDAF in such conditions with the proposed TEC framework.

5. Quality measure

Sources of performance degradation are clutter (false measurements), absence of measurements (misdetections) and association errors. Fig. 3 shows the proposed tracking quality estimation for a model points tracker taking into consideration the aforementioned sources of failure, which are represented by \( I_p \) in Eq. 4. The details of the quality estimation are described below.

5.1. Addressing performance degradation

An increase in the error covariance of the Bayesian tracker is an indication of low confidence for the local tracker [32]. The increase is a result of associating more than one measurement data \( Z_t \) in the validation region.
of the Kalman filter at the update step. The local points in the extended object come close to each other and lead to the observation of one model point in the validation region of the other model point trackers. The multiple measurements in the validation region are also a result of clutter. Let $T^i$ represent the $i^{th}$ local point tracker. Fig. 4 shows the covariance characteristics for local point trackers under clutter (the position of the points are manually collected from a walking person for simulation). $T^3$ and $T^6$ have clutter at frames 17 and 50, respectively. The association uncertainties from the clutter increase the covariances of the local point trackers and finally lead to a track loss.

Let $\Sigma$ be the covariance matrix in the Kalman filter and $p_{c,t}$ be the quality measure against the error in covariance (in the case of double subscript for $p$, the first subscript identifies the particular type of performance measure). We estimate $p_{c,t}$ by observing the magnitudes of the covariance matrix $|\Sigma|$.
Figure 4: Increase in covariance of local point trackers due to multiple measurements in the validation region. (a) sample track, (b) covariance magnitude and (c) track error. $T^3$ and $T^6$ incur track losses due to clutter appearance in their validation region. The Euclidean distance is measured between local point track and ground-truth position.

as

$$p_{c,t} = \begin{cases} 1 & \text{if } |\Sigma| \geq \Sigma_T \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

where $\Sigma_T$ is a threshold covariance magnitude used to estimate the trajectory. The magnitude $|\Sigma|$ is obtained by taking the absolute values for the matrix components. For selecting the threshold, factors such as spatial distribution
of points in the extended object are considered.

*Misdetections* due to sensor resolution or the feature extraction process cause an absence of measurement for the update stage. It is not possible to distinguish online if the absence of measurement is either from a tracking failure or from a misdetection. For such a reason, any absence of measurement at the update stage is used to estimate the track quality as a performance due to absence of measurement $p_{o,t} = 1$. Fig. 5 shows the results of $T^5$ and $T^7$ when they undergo misdetections at frames 30 and 50, respectively. Both trackers failed in the subsequent frames.

*Coalescence* occurs when two or more local trackers are tracking the same model point. For example, after a crossover between targets, one tracker assumes the wrong model point and continues to track. Coalescence also occurs due to an increase in covariance, that in turn is caused by a misdetection and clutter, and the tracker starts to consider data from other model points than its own. The other performance measures mentioned, $p_{c,t}$ and $p_{o,t}$, do not handle such situations. The increase in covariance may not be noticeable before it is detected by $p_{c,t}$. Therefore, a performance measure against coalescence $p_{s,t}$ is used to estimate the track quality. We estimate $p_{s,k}$ by examining the local trackers and their associated measurements. Whenever a single measurement alone is associated to two or more trackers, we set $p_{s,k} = 1$.

### 5.2. Strong and weak trackers

Considering the aforementioned low-quality tracking performance and sources of failure, the performance measure conducted on the output of the local trackers produces two classes: strong trackers and weak trackers. Weak trackers are not certain about the state they have estimated. Let $p_t = 1$ and $p_t = 0$ represent the decision given for weak and strong trackers, respectively. For each model point tracker the performance value $p^t_l = \{0, 1\}$ is determined as

$$p^t_l = p^c_{e,t} \lor p^o_{o,t} \lor p^s_{s,t},$$

(13)

where $\lor$ is a logical OR operator for combining performance measures (Fig. 3).

It is important to note that a weak tracker, for instance with large covariance, does not always imply tracking failure. The increase in covariance leads to the inclusion of other model points and also exposes it to a large amount of clutter in its validation region. Correcting this uncertainty, i.e. the low
Figure 5: Performance of local point trackers against misdetections. (a) sample track, (b) misdetections duration: D=detection and MD=misdetection, and (c) track error. A failure happens in $T^5$ and $T^9$ due to the misdetections of their model point measurement. The Euclidean distance is measured between local point track and ground-truth position.

confidence, will avoid associating multiple points in the subsequent frames to improve the accuracy of the model point trackers. The information to make corrections on the identified weak trackers is obtained from either an offline constructed model [21] or learned online [19]. Online learned methods are based on decisions from the performance measure. Our correction step for weak trackers is based on states of strong trackers, and details are described in the next section.
Figure 6: Correction step. The correlation model $\beta_t$, constructed from short windowed trajectories, and its modeling error covariance $C_t$ allow to select correlated local trackers $s_t$ and their corresponding weight $\tilde{\alpha}_t$ according to the degree of correlation. The selected trackers with a high degree of correlation allow the estimation of the corrected state $\hat{X}_t$ and tracker $\hat{T}$.

6. Correction

The correction step estimates the most probable state of the weak tracker from the state of strong trackers. A correlation model from trajectories is used to estimate the state of one tracker from the other. The state estimation using the correlation model enables $\Theta(\cdot)$, where trajectories from trackers represent $I_c$ (Eq. 5). Trajectories allow us to observe motion correlation of the state of model points over time. The correlation existing between local trackers of a generic extended object, such as articulated structure, is time dependent and has different degrees of correlation. The degree of correlation is the confidence in accurately estimating the state of one local tracker from the state of the others. The adaptive modeling of this correlation over time is important for estimating the probable state of the weak tracker.

The overall correction step comprises of three sub-steps (Fig. 6): correlation model constructor, correlated trackers selector and predictor/corrector. A correlation model is constructed for each set of weak trackers against a set of existing strong local trackers. From the correlation model, a degree of correlation between the weak tracker and the strong trackers is estimated by assigning normalized weights to the strong trackers. Finally, the state of the weak tracker is corrected by combining the outputs from the aforementioned
two components.

6.1. Partial Least Square regression

Correlation is modeled using Partial Least Square (PLS) regression [21]. PLS regression models the relation between dependent and independent variables. The independent variables are the predictors and the dependent variables are the responses. Trajectories from strong and weak trackers are used as predictor and response variables, respectively.

In order to formulate the correlation model at each frame, let $w_t$ and $s_t$ be the set of indices for weak and strong local trackers, respectively, from their performance evaluation $p_t$ according to

$$N_t = \{w_i : 1 \leq i \leq N_w, i \in \mathbb{N}^+\} \text{ and } s_t = \{s_j : 1 \leq j \leq N_s, j \in \mathbb{N}^+\},$$

where $N_w$ and $N_s$ are the number of weak and strong trackers, respectively, such that $N_w + N_s = N_t$. Trajectories are constructed from the position components of the tracker state. Let the trajectories at the current frame $\Gamma^w_t$ and $\Gamma^s_t$ for a weak and a strong tracker, respectively, be defined as

$$\Gamma^w_t = [x^w_{t-1}, x^w_{t-2}, \ldots, x^w_{t-m}], \quad w_i \in w_t,$$

$$\Gamma^s_t = [x^s_{t-1}, x^s_{t-2}, \ldots, x^s_{t-m}], \quad s_j \in s_t,$$

where $m$ is the trajectory length corresponding to a temporal window for modeling the correlation. The state $x_t$ is a $D \times 1$ vector, and hence $\Gamma_t$ has a dimensions of $D \times m$.

PLS regression allows the correlation model to be learnt by using the observed trajectories as a training data. The correlation model $\beta^{w_i,s_j}_t$ between a weak tracker $w_i$ and strong tracker $s_j$ is estimated using PLS as

$$\beta^{w_i,s_j}_t = \Gamma^s_t U^T \Gamma^s_t \Gamma^w_t V (\Gamma^w_t U^T \Gamma^s_t \Gamma^s_t V)^{-1} \Gamma^w_t,$$

where $U$ and $V$ are the component matrices for the predictor variable ($\Gamma^w_t$) and the response variable ($\Gamma^s_t$), respectively from principal component analysis and the superscript $T$ represents the transpose operator [49].

The correlation model allows the estimation of the corrected state of weak trackers $\hat{x}^{w_i}_t$ (Eq. 5) given the state of strong trackers through a probability
density function \( p(\hat{x}_t^{w_i} | x_t^{s_j}, \beta_t^{w_i}) \). The probability density is estimated from a motion model for predicting the weak tracker \( w_i \) from the strong local tracker \( s_j \), and is given as

\[
\hat{x}_t^{w_i} = A_t^{w_i} x_t^{s_j} + b_t^{w_i}
\]  

(17)

The matrix \( A \) and translation vector \( b \) are obtained directly from the model \( \beta^{w_i} (\beta_t^{w_i} = [A_t^{w_i}, b_t^{w_i}]) \). \( A^{w_i} \) and \( b^{w_i} \) have dimensions of \( D \times D \) and \( D \times 1 \), respectively.

Due to absence of perfect correlations in the training data (i.e. between \( \Gamma_t^{s_j} \) and \( \Gamma_t^{w_i} \)), the model \( \beta_t^{w_i} \) has a residual fitting error \( E_t^{w_i} \), and is calculated as

\[
E_t^{w_i} = \Gamma_t^{w_i} - (A_t^{w_i} \Gamma_t^{s_j} + B_t^{w_i})
\]  

(18)

where \( B_t^{w_i} \) is a matrix of dimension \( D \times m \) obtained by concatenation of \( m \) vectors of \( b_t^{w_i} \). \( E_t^{w_i} \) has a dimension \( D \times m \), where the \( D \) rows correspond to the dimension of the states and the \( m \) columns correspond to the trajectory history considered from the current frame of reference.

PLS estimates the correlation model by minimizing the variance of \( E_t^{w_i} \). As a result, the mean of \( E_t^{w_i} \) is very small compared to the individual \( m \) errors of the trajectory component. Assuming the small mean error value of \( E_t^{w_i} \), the covariance matrix \( C_t^{w_i} \) for estimating the state of weak trackers from the state of strong trackers in the training data is calculated as

\[
C_t^{w_i} = \frac{1}{m} E_t^{w_i} E_t^{w_i}^T
\]  

(19)

\( C \) has dimensions \( D \times D \).

The state \( \hat{x}_t^{w_i} \) is determined from the state of a strong tracker, \( x_t^{s_j} \), as

\[
p(\hat{x}_t^{w_i} | x_t^{s_j}) = \mathcal{N}(A_t^{w_i} x_t^{s_j} + b_t^{w_i}, C_t^{w_i})
\]  

(20)

where \( \mathcal{N} \) is a \( D \)-dimensional Gaussian density function (Eq. 8). The correction of a weak tracker is done in a Gaussian function form, due to the fact that the baseline tracker is the Kalman filter. The mean value of the Gaussian is the most probable position as estimated by other trackers’ states, and the covariance is proportional to \( E_t^{w_i} \).
6.2. Correlated trackers selection

The degree of correlation for a particular weak tracker \( w_i \in \mathbf{w}_t \) to a set of strong trackers \( \mathbf{s}_t = \{s_j\}_{j=1}^{N_{St}} \) varies from one strong tracker to another. For articulated structures, subsets of model points with good correlation are those laying in the same rigid structure relative to the overall object. In order to obtain the most accurate state estimation of the weak tracker, the best correlations need to be identified among the set of strong trackers. Fig. 7 shows an example of a trajectory for a particular weak tracker \( w \) and a list of available strong trackers \( (s_1, s_2, s_3, s_4 \text{ and } s_5) \): among the existing strong trackers \( s_1 \) shows the best correlation to the weak tracker while \( s_5 \) is the worst.

For each candidate weak tracker \( w_i \in \mathbf{w}_t \), a new set of strong trackers \( \mathbf{g}_i^{w_i} = \{g_n^{w_i} : 1 \leq n \leq N_{St}, n \in \mathbb{N}^+\} \) (the superscript identifies the set particular to weak tracker \( w_i \)) are selected from the original list \( \mathbf{s}_t \). The new array \( \mathbf{g}_i^{w_i} \) contains indices of strong trackers in decreasing order of correlation with tracker \( w_i \). \( \mathbf{g}_i^{w_i} \) is obtained by considering the PLS correlation model co-

![Figure 7: Trajectory correlations between trackers. (a) A weak tracker and a set of strong trackers. (b) A weak tracker against a list of strong trackers.](image)
Figure 8: Trajectories $\Gamma_t^{s_j}$ and $\Gamma_t^{w_j}$ from a strong and a weak tracker, respectively, used for training data in order to construct the correlation model $\beta_t^{w_j/s_j}$. The correlation model has a covariance $C_t^{w_j/s_j}$. The fitting error from the constructed model is used to measure the correlation between strong and weak local trackers.

variance $C_t^{w_j/s_j}$ (Eq. 19). Strong trackers in $g_t^{w_i}$ are ordered according to the formulation

$$|C_t^{w_i,g_{(n)}^{w_i}}| < |C_t^{w_i,g_n^{w_i}}|, \quad 1 \leq n \leq N_{St}.$$  \hspace{1cm} (21)

Ranking the correlation level is important since the level is directly proportional to the prediction accuracy. Fig. 8 shows how the correlation between two local tracks is estimated from the regression model. The magnitude of the red arrows indicates how large the fitting error between the constructed model and the observed trajectories is. The correlation level is determined by considering the magnitudes of these residual errors for the trackers.

For accurate state estimation, only the first $\gamma$ elements of $g_t^{w_i}$ are selected as predictor by considering the magnitude of the covariance in the PLS regression model:

$$|C_t^{w_i,g_p^{w_i}}| < C_T, \quad 1 \leq p \leq \gamma, p \in \mathbb{N}^+,$$  \hspace{1cm} (22)

where $C_T$ is a threshold covariance magnitude. The scale of the object and the spatial distribution of the model points are considered as factors for determining the magnitude of $C_T$ for a particular application. Additionally, the optimized value of the state covariance in the baseline tracker (Eq. 8) allows $C_T$ to be selected.
The selected $\gamma$ strong local trackers from $g_{w_i}^{w_i}$ are assigned a weight $\alpha_t$ which is inversely proportional to the magnitude of their covariance in the PLS model

$$\alpha_t^{w_i g_p} \propto \left| C_{w_i g_p}^{w_i} \right|^{-1}. \quad (23)$$

The estimated weight gives the goodness of correlation level and accuracy of prediction from the learned model. In order to have a proper probability density, the weights for the selected local strong trackers are normalized at each time step as

$$\tilde{\alpha}_t^{w_i g_p} = \frac{\alpha_t^{w_i g_p}}{\sum_{p=1}^{\gamma} \alpha_t^{w_i g_p}}. \quad (24)$$

6.3. Correction

For each weak tracker $w_i$, the selected $\gamma$ strong trackers jointly contribute to estimate $\hat{x}_t^{w_i}$ as

$$p(\hat{x}_t^{w_i}) = p(\hat{x}_t^{w_i} | x_t^{g_i}) = N(x_t^{w_i, mean}, C_t^{w_i}), \quad (25)$$

where

$$x_t^{w_i, mean} = \sum_{p=1}^{\gamma} \tilde{\alpha}_t^{w_i g_p} \left( A_t^{w_i g_p} x_t^{g_p} + b_t^{w_i g_p} \right),$$

and

$$C_t^{w_i} = \sum_{p=1}^{\gamma} \tilde{\alpha}_t^{w_i g_p} C_t^{w_i g_p}.$$ 

It is worth mentioning that the set of strong tracker indices for the available two or more weak trackers are the same at each step of correction. However, the selected $\gamma$ strong predictor trackers for a particular weak local tracker are different and are assigned a weight according to their constructed regression models. Eq. 25 represents the function $\Theta(\cdot)$ in Eq. 5, and the state $p(\hat{x}_t^{w_i})$ is used as the corrected state output and prior information for the execution of the baseline tracker. The correction is a form of Kalman filter re-initialization, which estimates $\hat{x}_t$ and changes the prior information of $T$ in order to obtain $\hat{T}$. 

20
Figure 9: Motion capture pipeline. 2D points $y_{t}^{i=1:M_t}$ obtained by each camera are used to estimate the absolute 3D position of each marker indices. The obtained 3D points $z_{t}^{i=1:M_t}$ are associated and tracked at each time to create the markers’ trajectories. Misdetections, false detections and data association problems generate errors. Finally, off-line post-processing remove the outliers.

7. Experimental results

We present a quantitative performance evaluation of the proposed TEC framework by tracking markers from a motion capture system. Tracking of the markers involves matching corresponding marker indices at different frames from the list of unlabeled points. Fig. 9 shows a typical motion capture system [50]. At each frame, in the measured two-dimensional and three-dimensional points there are misdetections and false detections (clutter) of markers. Misdetections are the result of occlusions and wrong spatial alignment of the two-dimensional points detected by cameras in the motion capture system. False detections are common when multiple cameras are used and their detected two-dimensional points are cluttered. Using the markers, we compare the results of TEC framework with the baseline tracker JPDAF [51] [47]. JPDAF and TEC for 30 targets run on average 30 and 26 frames per second, respectively, (without clutter) and 16 and 12 frames per second (with 100 as the amount of clutter) on an Intel i5-3570k, 3.4GHz CPU with 8GB RAM on Windows 7 (non-optimized MATLAB2013b code). We also compare TEC with other state-of-the-art multi-target trackers: two online methods - Kalman filters using Hungarian

\(^{1}\text{code: http://www.mathworks.co.uk/matlabcentral/fileexchange/34146, Last accessed: March 2014}\)
algorithm and nearest-neighbor data associations HDAF [52] and NNDAF, respectively, and two offline methods - Hungarian-assessment-based particle tracker PT [53] and motion dynamics-based assessment for tracking targets with similar appearances SA [54].

7.1. Experimental setup

We use the Carnegie Mellon University (CMU) [55] and HumanEva [56] Motion Capture database. The data is from a human performing different actions. For the CMU dataset, the front snapshot of the human body with the marker indices are shown in Fig. 10. The subject has 41 markers, and the data include absolute position (three-dimensional) and orientation information. We select 39 markers, and their three-dimensional absolute positions are read from C3D file format in the database. Two of the markers are discarded in the experimentation since their initial frame measurements are either incorrect.

---

3code: http://www.mathworks.co.uk/matlabcentral/fileexchange/34040, Last accessed: March 2014
4code: https://bitbucket.org/cdicle/smot, Last accessed: March 2014
Table 2: Dataset used for validating the proposed Track-Evaluate-Correct approach. Legend- 3D: three-dimensional dataset from CMU [55] and HumanEva [56], 3D-m: 3D with misdetections, 3D-c: 3D with clutter, 2D-m and 2D-c: two-dimensional of the 3D dataset with misdetections and clutter, respectively. FL: frame length, NM: number of markers, SF: source of failure.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Motion</th>
<th>FL</th>
<th>NM</th>
<th>SF</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>Walking</td>
<td>342</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>172</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dancing</td>
<td>867</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WalkingEva</td>
<td>1877</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JoggingEva</td>
<td>1701</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>3D-m</td>
<td>3D</td>
<td></td>
<td></td>
<td>Misdetection</td>
</tr>
<tr>
<td>3D-c</td>
<td>3D</td>
<td></td>
<td></td>
<td>Clutter</td>
</tr>
<tr>
<td>2D</td>
<td>Walking</td>
<td>342</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>172</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dancing</td>
<td>320</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>2D-m</td>
<td>2D</td>
<td></td>
<td></td>
<td>Misdetection</td>
</tr>
<tr>
<td>2D-c</td>
<td>2D</td>
<td></td>
<td></td>
<td>Clutter</td>
</tr>
</tbody>
</table>

or not available. Similarly, for the HumanEva dataset we use 39 markers.

The comparison of TEC with JPDAF is done by using three-dimensional marker points, while the comparison with other state-of-the-art trackers (including JPDAF) is done on two-dimensional marker points as the trackers implementation is for two-dimensional targets. The two-dimensional markers are obtained by removing the less variable coordinate component followed by removing markers indices that overlap and are very close to each other in their two-dimensional view. The labels obtained from the dataset are used as a ground-truth for evaluating tracking results. Moreover, sources of failures are added to the original data (Table 2). 3D is the original dataset obtained from CMU and HumanEva, while 3D-m and 3D-c represent the data with imposed misdetections and clutter, respectively. 2D-m and 2D-c are two dimensions of the original 3D dataset with misdetections and clutter, respectively. The added clutter appears only in the rectangular volume occupied by the object for the 3D dataset and in the image plane view of the object for the 2D dataset. We test the experimentation for a variable amount of clutter, misdetection durations and probability of misdetection (see Table 3).

We use the same tracking parameters for each type of motion sequences
Table 3: Parameters and values used to generate $DS-m$ and $DS-c$ from Dataset $DS$.
Legend- $Ψ$: uniformly-distributed pseudo-random number generator function, $∆$: difference between maximum and minimum values of the marker states, $x$: position component of the object state in the 3D space.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Parameters</th>
<th>Symbol</th>
<th>Value/Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-m</td>
<td>duration of misdetections</td>
<td>$L_{md}$</td>
<td>$1 \leq L_{md} \leq 15$ frames</td>
</tr>
<tr>
<td></td>
<td>misdetected marker number</td>
<td>$N_{md}$</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>misdetected marker index</td>
<td>$I_{md}$</td>
<td>$Ψ$</td>
</tr>
<tr>
<td>3D-c</td>
<td>amount of clutter</td>
<td>$N_{c}$</td>
<td>$100 \leq N_{c} \leq 600$</td>
</tr>
<tr>
<td></td>
<td>volume occupied by the object</td>
<td>$V$</td>
<td>$Δx_1 \times Δx_2 \times Δx_3$</td>
</tr>
<tr>
<td></td>
<td>state of clutter</td>
<td>$x_c$</td>
<td>$V \times Ψ$</td>
</tr>
<tr>
<td>2D-m</td>
<td>probability of misdetection</td>
<td>%MD</td>
<td>0%, 1%, 5%, 10%</td>
</tr>
<tr>
<td></td>
<td>fixed amount of clutter</td>
<td>%$N_{c}$</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>misdetected marker index</td>
<td>$I_{md}$</td>
<td>$Ψ$</td>
</tr>
<tr>
<td>2D-c</td>
<td>percentage of clutter</td>
<td>%$N_{c}$</td>
<td>0%, 50%, 100%, 200%</td>
</tr>
<tr>
<td></td>
<td>state of clutter</td>
<td>$x_c$</td>
<td>$Ψ$</td>
</tr>
<tr>
<td></td>
<td>fixed probability of misdetections</td>
<td>%MD</td>
<td>5%</td>
</tr>
</tbody>
</table>

and optimal parameters are selected by examining the results. The matrices $Q_t$ and $R_t$ in the Kalman filters of JPDAF are set according to the formulation in [57] (variances of 20cm and 5cm, respectively). The gate probability for the Gaussian distribution is set to 0.99. As the quality measure for covariance grow, we use $Σ_T = (2 - 4)Σ_{int}$ ($Σ_{int}$ is the initial covariance matrix in the Kalman filter estimated from $Q_t$). A trajectory length of $m = 12$ is used to train the correlation model. Long trajectories are expensive in terms of computation and do not model appropriately fast changing correlations between states of model points. We use $γ = 4$ for selecting the number of best predictor trackers and zero weights are assigned to the rest of the trackers (Eq. 24). Kalman filters in HDAF and NNDF are set similarly to JPDAF. The minimum association threshold for HDAF is set to 10cm. For SA tracker, we use a singular threshold value of 0.3, a time window of 80 frames and ADMM method. For PT, we use the default parameters as provided in the code and tracklets shorter than 30 frames are removed.

7.2. Evaluation measure

Tracking results are compared to the labeled ground-truth data based on OSPA [58]. Let $O = \{O_t\}_{t=1}^{τ}$, $O_t = \{o_{t}^{i}\}_{i=1}^{G_t}$, $t \in \mathbb{N}^+$ be the labeled
ground-truth data, where \( G_t \) is the number of targets at each time. The OSPA distance \( D_t \) is estimated according to

\[
D_t(X_t, O_t) = \left[ \frac{1}{\max(N_t, G_t)} \left( \min_{\pi \in \Pi_{G_t}} \sum_{t=1}^{N_t} d_c(x_t^i, o_t^j)^b + |G_t - N_t| c^b \right) \right]^{1/b},
\]

where \( \Pi_{G_t} \) is a set of permutations of length \( N_t \) taken from the \( \{1, 2, \ldots, G_t\} \), \( d_c(x_t^i, o_t^j) = \min(c, d(x_t^i, o_t^j)) \) is a cut of distance with \( c > 0 \), \( d(x_t^i, o_t^j) \) is the base distance (i.e. Euclidean) and \( b \) is the order of OSPA metric. We use the MATLAB code provided in [58] for estimating \( D_t \), with \( c = 10 \text{cm} \) by considering the spatial distance between markers, and \( b=1 \).

To characterize the overall tracker performance, we also count the number of False Positive FP for local track output components. For a frame length of \( \lambda \), the FP level on a local tracker \( l \) is

\[
FP = \begin{cases} 
1 & \text{if } d_t^l \geq d_{Th}, t \in \lambda \\
0 & \text{otherwise},
\end{cases}
\]

where \( q_{Th} \) is the thresholds of error value. In the reported results, we use \( q_{Th} = 10 \text{ cm} \) and \( \lambda > 10 \). We calculate FP for online trackers only, as the association of tracks and ground-truth data is not known for offline trackers.

7.3. Discussion

The first set of experiments examines how accurately the PLS regression model can estimate the state of one marker from the states of other markers. Fig. 11 shows the results of the state estimation accuracy with the corresponding regression model covariance (Eq. 23). The covariance of the regression model is estimated from the training error according to Eq. 19, which measures estimation accuracy. The results indicate that different local point trackers have different state estimation accuracy to a specific local point tracker and the accuracy is variable in time. This property is noticeable for deformable objects and articulated structures, such as human motion which is considered here. Table 4 gives the mean and standard deviation errors for estimating the state of one marker from the state of other markers.

Figure 12 shows the result of state estimation of marker index RTOE (see Fig. 10) by weighting adaptively the estimation of other markers. The
Figure 11: PLS regression model (training) errors and estimation errors for marker RTOE (see Fig. 10) using the states of the other marker indices. (a) Using LTOE; (b) using RWRA.

Table 4: Mean and standard deviation of errors (cm) for the PLS regression model over the walking sequence for different markers selected (see Fig. 10). Legend: $\mu$: mean, $\sigma$: standard deviation, P: predictors variables (row space), R: response variables (column space).

<table>
<thead>
<tr>
<th>P/R</th>
<th>RTOE $\mu$</th>
<th>RTOE $\sigma$</th>
<th>LTOE $\mu$</th>
<th>LTOE $\sigma$</th>
<th>RWRA $\mu$</th>
<th>RWRA $\sigma$</th>
<th>LKNE $\mu$</th>
<th>LKNE $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTOE</td>
<td>1.61</td>
<td>2.94</td>
<td>1.65</td>
<td>2.17</td>
<td>1.43</td>
<td>1.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTOE</td>
<td>2.61</td>
<td>3.78</td>
<td>-</td>
<td>-</td>
<td>1.14</td>
<td>1.59</td>
<td>1.09</td>
<td>1.61</td>
</tr>
<tr>
<td>RWRA</td>
<td>0.57</td>
<td>0.83</td>
<td>0.27</td>
<td>0.32</td>
<td>-</td>
<td>-</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>LKNE</td>
<td>0.86</td>
<td>1.08</td>
<td>0.34</td>
<td>0.48</td>
<td>0.29</td>
<td>0.24</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>LSHO</td>
<td>0.51</td>
<td>0.66</td>
<td>0.38</td>
<td>0.65</td>
<td>0.24</td>
<td>0.23</td>
<td>0.19</td>
<td>0.16</td>
</tr>
<tr>
<td>RFHD</td>
<td>0.51</td>
<td>0.67</td>
<td>0.36</td>
<td>0.50</td>
<td>0.24</td>
<td>0.26</td>
<td>0.19</td>
<td>0.14</td>
</tr>
<tr>
<td>LWRA</td>
<td>0.41</td>
<td>0.43</td>
<td>0.71</td>
<td>1.31</td>
<td>0.43</td>
<td>0.71</td>
<td>0.34</td>
<td>0.53</td>
</tr>
</tbody>
</table>

weights assigned to each predictor trackers (i.e. strong trackers) is indicated in the heat map (Fig. 12(a)). Due to the dynamic changing nature of human motion, the importance of adaptive modeling of the correlation is shown from the scattered weights assigned to the predictor marker indices in the heat map. From the map it is possible to see that for RTOE the marker index 32, just above it, is more correlated for large frame number durations. Fig. 12(b) shows the state estimation error of RTOE from the weighted prediction of all the other markers. The weighted estimate shows smaller error than the estimate errors by individual indices (Fig. 11).

Tracking on 3D data involves creating labels (data association) for each
Figure 12: Selection of best predictor indices. (a) Heat map of weights calculated for predictor marker indices for estimating the state of marker RTOE. (b) Estimation error of marker RTOE from the weighted estimate of all the other markers.

Figure 13: Sample three-dimensional track results on the CMU 3D dataset for walking (Table 3) using JPDAF. The ellipses represent the validation regions of each local tracker for the markers. The number next to each ellipse is the label (index) of the marker.

marker index at each frame in the sequence. For this dataset, the performance of JPDAF, without TEC, produces good tracking results. For the Dancing sequence $FP = 1$, while for the Walking, Running, WalkingEva and JoggingEva $FP = 0$. Fig. 13 shows sample tracking results for the walking sequence of 3D. The 3D ellipses represent the validation region of the marker trackers. Video results for the sequences described in Table 2 are
Comparisons of JPDAF and TEC, using the CMU dataset, are shown in Fig. 14. The first row of the figure shows the result of performance for JPDAF on dataset 3D- \( m \) compared to TEC using the distance metric given in Eq. 26. Here, misdetections duration \( L_m = 5 \) is considered. The results presented are averaged over 50 runs. The proposed approach shows a significant improvement of the error distance \( D \) compared to JPDAF in all types of sequences considered. The obtained error distances are small, since only a small percentage of the overall markers undergoes low-quality tracking performance or tracking failure. The increase in the distance error with frame numbers is due to an increase in the number of failed local tracks in the motion capture sequence. The second row of Fig. 14 shows the performance of JPDAF and TEC using the number of FP for local tracks on 3D- \( m \). The results are shown for variable misdetection length \( L_m \) (Table 3). The percentage of failures for markers are 19\% and 5\% for JPDAF and TEC, respectively, when averaged over all types of sequences in 3D- \( m \). The result using TEC for the running dataset has not shown much improvement compared to the other sequences, particularly for longer misdetection durations. This particular case is a limitation to our correlation-based correction approach. The reason behind the limitation is due to rapid changes in correlation between marker trackers, which is caused by the rapid change in the dynamics of the markers. Additionally, the proposed correction schema is likely to fail with model points getting spatially close to each other for long time intervals.

For dataset 3D- \( c \), the third row of Fig. 14 shows the performance of JPDAF and TEC using \( D \) at each frame of the sequence. The performance of tracking markers is improved by using TEC over JPDAF. The result reported is also averaged on 50 runs. The fourth row of Fig. 14 shows the performance of tracking by counting the number of FP for different amounts of added clutter. The amount of clutter is increased in such a way that it will lead the JPDAF to a likely failure. Averaging over all types of sequences in 3D- \( c \), the percentage of failure for markers is smaller than 1\% for the TEC framework, while for the tracker alone it is 6\%. The result shows that the TEC framework improves the tracking result of JPDAF in clutter. Trajectory output plots for JPDAF and TEC framework on the dataset 3D are shown in Fig. 15. In this figure, results from selected indices are shown to compare the improvements made by TEC over JPDAF. Tracks from JPDAF show failures due to drift caused by the presence of clutter and wrong association with other local trackers.
Figure 14: Comparisons of JPDAF and Track-Evaluate-Correct (TEC) on the 3D dataset from CMU. The results are averaged on 50 independent runs. First row: OSPA (Eq. 26) plots for misdetections duration of 5 frames. Second row: Number of FP local tracks with different misdetection durations. Third row: OSPA plots for 500 points of added clutter. Fourth row: Number of false positive (FP) local tracks with variable amount of clutter. First column: Walking. Second column: Running. Third column: Dancing.
Similarly using the HumanEva dataset, comparisons of JPDAF and TEC are shown in Fig. 16. The improvement made by TEC on the JoggingEva dataset is not as good as that of the WalkingEva dataset due to the rapid changes in correlation between markers (second row, second column). Averaging over the sequences in the HumanEva dataset, TEC has 12%, while JPDAF has 20.4% of failures. The subjects in the HumanEva dataset make circular motion, unlike Walking and Running in the CMU dataset which are on a straight direction and cause rapid changes in the correlation model. Failures in the HumanEva dataset are more numerous than those in the CMU dataset.

Figure 17 shows the performance of TEC compared to other state-of-the-art trackers using the datasets $2D-m$ and $2D-c$. HDAF performs better if the measurements contain either clutter or misdetections. However, when both challenges exist at the same time the performance degrades. NNDAF performs well when there are no misdetections. Unlike JPDAF, HDAF and NNDAF make hard decisions in the association, therefore the presence of clutter and misdetections strongly influence their tracking performance.

As for the offline trackers: PT and SA perform poorly, since misdetection and clutter largely affect the overall optimization accuracy. In addition to this, offline trackers produce two or more tracklets for a single marker and result in a higher $D$ as compared with online trackers that produce tracks equal to the number of markers. JPDAF performs well with a large amount of clutter and presence of higher misdetections compared to the other trackers.
However, using the proposed TEC framework, we have further improved the tracking results from JPDAF. For the dataset 2D-m (2D-m), TEC improves D and FP of JPDAF by 0.64 (1.14) and 1.5 (2.2), respectively. Comparison
Table 5 compares the results obtained by all the trackers under consideration. The statistics given are averaged over the different types of sequences, the frame lengths and the separate multiple iteration runs. The number of failures and tracking errors in the 2D dataset are larger compared to the 3D dataset. The trajectories of markers in the 2D dataset crossover multiple times, thus increasing the data association problem compared to the 3D dataset. On average over all the experiments, TEC improves the tracking performance of JPDAF by reducing $D$ and $FP$ by 0.66 and 2.87, respectively.

Figure 17: Comparison of trackers performance using OSPA (Eq. 26) averaged over the different human motions. (a) Dataset 2D-$m$; (b) Dataset 2D-$c$.

Figure 18: Comparison of trackers performance using number of false positive ($FP$) averaged over the different human motions. (a) Dataset 2D-$m$ with misdetections $MD$; (b) Dataset 2D-$c$ with amount of clutter $N_c$.

of online trackers in terms of number of $FP$ is shown in Fig. 18. The number of $FP$ for offline trackers is not given as the track association with ground-truth is not known a priori.
Table 5: Comparisons of the results. For the best performing tracker values are indicated in bold and values not calculated are indicated by “-”.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>D(cm)</th>
<th>Number of FP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average</td>
<td>Min</td>
</tr>
<tr>
<td>3D-m</td>
<td>JPDAF</td>
<td>1.01</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>TEC</td>
<td>0.08</td>
<td>0.03</td>
</tr>
<tr>
<td>3D-c</td>
<td>JPDAF</td>
<td>0.46</td>
<td>0.19</td>
</tr>
<tr>
<td></td>
<td>TEC</td>
<td>0.22</td>
<td>0.11</td>
</tr>
<tr>
<td>2D-m</td>
<td>PT</td>
<td>4.43</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>SA</td>
<td>6.20</td>
<td>3.75</td>
</tr>
<tr>
<td></td>
<td>NNDAF</td>
<td>6.14</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>HDAF</td>
<td>3.80</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>JPDAF</td>
<td>1.01</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>TEC</td>
<td>0.37</td>
<td>0.00</td>
</tr>
<tr>
<td>2D-c</td>
<td>PT</td>
<td>6.34</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>SA</td>
<td>6.58</td>
<td>3.72</td>
</tr>
<tr>
<td></td>
<td>NNDAF</td>
<td>8.53</td>
<td>8.01</td>
</tr>
<tr>
<td></td>
<td>HDAF</td>
<td>5.46</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>JPDAF</td>
<td>1.46</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>TEC</td>
<td>0.32</td>
<td>0.00</td>
</tr>
</tbody>
</table>

8. Conclusions

We presented a Track-Evaluate-Correct framework for improving the robustness in extended-object tracking by a set of model points using a Bayesian tracker. The framework uses an on-line performance evaluation based on predefined failure models to decide whether the results obtained by each local point tracker are weak or strong. A weak tracker corrects its state based on the assistance from the available strong trackers. Inferring the corrected state of the weak tracker from that of strong trackers is done by Partial Least Square regression using the short windowed trajectories of the trackers. For an accurate estimation of a weak tracker state, a weight is assigned to the estimations from strong trackers based on the observed correlation level, which is obtained from the regression model. Experimental results on tracking markers with challenges such as misdetections and clutter have shown improved tracking performance by using the proposed approach.

As a future work, the proposed framework will be extended to other existing model points trackers and multi-target tracking scenarios. The correlation information between trackers, which has been used for correction,
can be exploited to assist hard decision in data association strategies.

9. Acknowledgment

T. Biressaw is supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the Education, Audio-visual & Culture Executive Agency under the FPA no. 2010-0012.

References


[46] J. Hartikainen, S. Särkkä, RBMCDAbox-Matlab Toolbox of Rao-Blackwellized Data Association Particle Filters, Department of Biomedical Engineering and Computational Science, Helsinki University of Technology (February 2008).


URL http://www.mathworks.co.uk/matlabcentral/fileexchange/34146

URL http://studentdavestutorials.weebly.com/multi-bugobject-tracking.html

URL http://www.mathworks.co.uk/matlabcentral/fileexchange/34040-simple-tracker


URL http://mocap.cs.cmu.edu

[56] Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion, last accessed: March 2014.
URL http://vision.cs.brown.edu/humaneva/index.html
