A Two-Stage Dynamic Model for Visual Tracking

Matej Kristan, Stanislav Kovačič, Aleš Leonardis, *Member, IEEE*, and Janez Perš

***Abstract*—We propose a new dynamic model which can be used within blob trackers to track the target's center of gravity. A strong point of the model is that it is designed to track a variety of motions which are usually encountered in applications such as pedestrian tracking, hand tracking and sports. We call the dynamic model a two-stage dynamic model due to its particular structure, which is a composition of two models: a liberal model and a conservative model. The liberal model allows larger perturbations in the target's dynamics and is able to account for motions in between the random-walk dynamics and the nearly-constant-velocity dynamics. On the other hand, the conservative model assumes smaller perturbations and is used to further constrain the liberal model to the target's current dynamics. We implement the two-stage dynamic model in a two-stage probabilistic tracker based on the particle filter and apply it to two separate examples of blob tracking: (i) tracking entire persons and (ii) tracking of a person's hands. Experiments show that, in comparison to the widely used models, the proposed two-stage dynamic model allows tracking with smaller number of particles in the particle filter (e.g., 25 particles), while achieving smaller errors in the state estimation and a smaller failure rate. The results suggest that the improved performance comes from the model's ability to actively adapt to the target's motion during tracking.**

***Index Terms*—Dynamic Models, Two-Stage Models, Blob Tracking, Probabilistic Tracking, Particle Filters**

I. Introduction

Tracking in video data is a part of a broad domain of computer vision that has received a great deal of attention from researchers over the last twenty years. One of the reasons is that the computer-vision-based visual tracking has found its way into many real-world applications such as visual surveillance, video editing, analysis of sport events, kinematic analysis in medical applications, and human-computer interfaces. This gave rise to a body of literature, of which surveys can be found in the work of Aggarval and Cai [1], Gavrila [2], Gabriel et al. [3], Hu et al. [4] and Moeslund et al. [5], [6]. In many applications, the tracked object (the target) is approximated by a single region or a blob. The blob's interior is used to extract the target's appearance model, while the blob's centre of gravity is used to encode the target's position in the image. These trackers are called the blob trackers and are convenient for tracking a variety of objects, since they typically make only weak assumptions about the object's shape. Some examples of modelling a human body and only a part of a human body by a blob are shown in Figure 1.

(a) (b) (c)

Fig. 1. Examples of using blobs to model entire persons (a,b) and a person's hand (d). The blobs are parameterized by ellipses.

A major difficulty in visual tracking is the uncertainty associated with the visual data as well as the uncertainty associated with the dynamics of the tracked target. One way to deal with these uncertainties is to apply a recursive Bayes filter (see, e.g., [7], page 638) to continuously estimate a posterior probability density function (pdf) over the parameter space of the target's model. The mode, or the mean, of the posterior can then be taken as an estimate of the target's state (e.g., position). An early analytical solution to the recursive Bayes, which approximates the posterior by a single Gaussian distribution, was presented in the sixties. Although the original derivation was not presented strictly in a Bayesian form it is the first solution to the recursive Bayes and is now commonly known as the Kalman filter [8]. The assumptions which are used to derive the Kalman filter are often too restrictive for visual tracking and pose a major drawback for its application in real-life scenarios. The reason is that the measurement process and the target's dynamic model are assumed to be linear and Gaussian; these assumptions are often severely violated in the visual tracking. In the past decade, Monte-Carlo-based numeric approximations of the recursive Bayes filter, called the particle filters [9], have become a widely-used approach to tracking, due to their ability to account for much more general processes than the Kalman filter. A central point of the particle filters is that they approximate the posterior at one time-step by a number of weighted particles and can thus deal with a large variety of shapes of the posterior. The posterior for the next time-step is obtained by sampling and propagating the particles through the target's dynamic model, and re-evaluating their weights against the visual data. The variance of the estimated state will largely depend on the number of particles used and the strategy by which they are allocated at hypothesized states. For example, in many applications it is difficult to derive a good dynamic model for the target's dynamics and usually a simple dynamic model with a large process noise is used instead. In those cases, after the simulation step, the particles have to be well spread in the state-space so they can cover the alternative hypotheses in the target's motion. This, however, can result in many particles having low weights and contributing very little to the final

M. Kristan is with the Faculty of Computer and Information Science, and with the Faculty of Electrical Engineering, University of Ljubljana, Slovenia.

A. Leonardis is with the Faculty of Computer and Information Science, University of Ljubljana, Slovenia.

S. Kovačič and J. Perš are with the Faculty of Electrical Engineering, University of Ljubljana, Slovenia.

estimate of the target's state. As a result, the variance of the estimated state value increases and can even lead to a loss of track. Two solutions are commonly used in practice: (i) increase the number of particles and/or (ii) use clever strategies to allocate the particles close to the modes of the posterior. Such strategies may be applications of an auxiliary-variable particle filter [10], local-likelihood sampling [11], a boosted particle filter [12] or hybrid applications with hill-climbing routines [13], to name just a few.

An alternative, by which the above problems of an increased variance of the state estimator can be addressed, is to use better models of the target's dynamics to more efficiently allocate the particles in the first place. Good allocation of particles also improves tracking by reducing the uncertainty which arises from the visual data, thus achieving a more reliable track. In this paper we focus on such an approach. We propose a new dynamic model for tracking the target's centre of gravity in blob trackers. The model is implemented within the framework of particle filters and is designed to track human motions. The dynamic model allows an improved tracking in comparison to the widely-used dynamic models while at the same time requires only a low number of particles in the particle filter. We provide simple rules to selecting the model's parameters and demonstrate its effectiveness quantitatively as well as qualitatively with examples of tracking person's body and with examples of tracking person's hands.

A. Related work

When the dynamics of the tracked object are known, the search space of the parameters to be estimated during tracking can be constrained considerably. In this respect, most of the work on human dynamic models has been focused on deriving *detailed* kinematic models for human pose estimation, e.g., [14], [15], [16], [17]. However, in blob trackers, when tracking entire persons or only a part of a person (see Figure 1 for examples), the motion cannot be constrained as much in practice, and simpler dynamic models are used instead. The common choices are a random-walk (RW) model or a nearly constant velocity (NCV) dynamic model; see [18] for a good treatment of these. The RW model assumes that the target's velocity is a white-noise sequence and is thus temporally completely non-correlated. On the other hand, the NCV model assumes that velocity is temporally strongly correlated, since it assumes that the changes in velocity arise only due to the white noise of the acceleration. The RW model thus describes the target's dynamics best when the target performs radical accelerations in random directions, e.g. when undergoing abrupt movements. However, when the target moves in a certain direction (which is often the case in, e.g., surveillance), the RW model performs poorly and the motion is better described by the NCV model.

In practice, the target will undergo various different types of motion and therefore, to cover a range of possible dynamics of the tracked object, some authors have proposed an interacting multiple model (IMM) approach. In this approach multiple trackers, each with a different dynamic model, are used in parallel for tracking the target. A special scheme is used to determine how well each model describes the target's current motion and the estimates from different trackers are then combined accordingly. A detailed treatment of different combination schemes is given in [19]. The interacting multiple model approaches based on Kalman filters have received considerable attention in the work on aircraft tracking with radars [20], [21], and an application to camera gaze control can be found in [22]. A particle-filter-based implementation of IMM can be found in [23], [24], [25]. A drawback of IMM approaches is that the complexity of tracking increases dramatically, since now the probability distributions have to be estimated (jointly) over each of the interacting models. In particle filters, the likelihood function of observations has to be evaluated for each hypothesis (particle). In visual tracking, calculating the likelihoods is usually time-consuming since the visual model has to be calculated for each particle and compared to the reference model. Thus computational efforts of visual tracking with particle filters is considerably increased when using IMM approaches.

For many applications, such as tracking in sports, gesture-based human-computer interfaces and surveillance, it is difficult to find a compact set of rules that govern the target's dynamics. Because of this, and the computational complexity associated with the IMM methods, researchers usually model the target's motion using a single model. In practice, to cover a range of different motions, a common solution is to choose either a RW [26], [27], [13] or a NCV [28], [12], [29], [30], [31], [32], [33] model, and increase the process noise to account for the unmodelled dynamics. An obvious drawback of this approach is that poorly modelling dynamics can significantly deteriorate the tracker's performance. Another drawback, in absence of additional solutions, is that the increase of the process noise requires an increase of the number of the particles to maintain a good track, which can in turn slow down the tracking and rendering the trackers less appropriate for realtime applications.

To alleviate the increasing variance of the estimation when using a random walk model with a large process noise, Okuma et al. [12] integrated a particle filter with an AdaBoost object detector. Another approach to variance reduction was proposed by Needham [27], who applied a Kalman filter to further filter the estimates from the particle filter. An ad-hoc scheme was then used to *move* the particles in the particle filter closer to the Kalman-filtered estimate. In a multiple-interacting-targets application [34], a similar approach was applied with a collision-avoidance algorithm: Each time a target would collide with another target, the dynamic model for one target would be modified to move particles in the particle filter away from the other target. Both approaches have in common the concept that a single dynamic model is used within a particle filter and another model is used on top of a particle filter to improve the final estimation of the target's state; we adopt this concept in our approach.

B. Our approach

We propose a two-stage dynamic model for tracking the target's centre of gravity in blob trackers. The dynamic model is designed to account for motions which we usually observe

in tracking persons. We call the model a *two-stage* dynamic model due to its particular structure, which is composition of two models: a liberal and a conservative model. The liberal model allows larger perturbations in the target's dynamics and is able to account for motions in between the RW dynamics and the NCV dynamics. This is achieved by explicitly modelling the target's velocity as a non-zero-mean Gauss-Markov process. The conservative model assumes smaller perturbations in the velocity and is used in to further constrain the liberal model to the target's current dynamics. We implement the proposed dynamic model in a two-stage probabilistic tracker which is based on the particle filter. We apply the proposed dynamic model to examples of tracking entire persons and to examples of tracking person's hands.

The outline of the remainder of the paper is as follows. In Section II we first briefly overview the bootstrap particle filter. In Section III-A we develop the liberal dynamic model and analyze how the parameters of the model influence the model's structure. The conservative model is proposed in Section III-B, in Section III-C we propose the two-stage dynamic model and its application to blobtracking in Section III-D. In Section IV results of the experiments are reported, and conclusions are drawn in Section V.

II. Bootstrap particle filter

We give here only the basic concept of the particle filters and notations, and refer the reader to [35] for more details. Let $\mathbf{x}_{k-1}$ denote the state (e.g., position and size) of a tracked object at time-step $k-1$, let $\mathbf{y}_{k-1}$ be an observation at $k-1$, and let $\mathbf{y}_{1:k-1}$ denote a set of all observations up to $k-1$. From a Bayesian point of view, all of the interesting information about the target's state $\mathbf{x}_{k-1}$ is encompassed by its posterior $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$. During tracking, this posterior is recursively estimated as the new observations $\mathbf{y}_k$ arrive, which is realized in two steps: prediction (1) and update (2),

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}, \quad (1)$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}). \quad (2)$$

The recursion (1,2) for the posterior, in its simplest form, thus requires a specification of a dynamical model describing the state evolution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and a model that evaluates the likelihood of any state given the observation $p(\mathbf{y}_k|\mathbf{x}_k)$.

In our implementation we use a simple bootstrap particle filter [36], [37]. The posterior at time-step $k-1$ is estimated by a finite Monte Carlo set of states $\mathbf{x}_{k-1}^{(i)}$ and their respective weights $w_{k-1}^{(i)}$, $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \approx \{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N}$, such that all weights in the particle set sum to one. At time-step k the particles are first resampled according to their weights, in order to obtain an unweighted representation of the posterior $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \approx \{\tilde{\mathbf{x}}_{k-1}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$. Then they are propagated according to the dynamical model $p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k-1}^{(i)})$, to obtain a representation of the prediction $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \{\mathbf{x}_{k-1}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$. Finally, a weight is assigned to each particle according to the likelihood function $w_k^{(i)} \propto p(\mathbf{y}_k|\mathbf{x}_k^{(i)})$, all weights are normalized to sum to one, and the posterior at the time-step k is approximated by a new weighted particle set $p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}$. The current state of the target $\hat{\mathbf{x}}_k$ can then be estimated as the minimum mean-square error (MMSE) estimate over the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{N} \mathbf{x}_k^{(i)} w_k^{(i)}. \quad (3)$$

III. The two-stage dynamic model

A. The liberal model

As noted in the introduction, the conceptual difference between RW and NCV models is that they assume two extremal views on the temporal correlation of the velocity. With this rationale we can obtain a more general model by simply treating the velocity as a correlated (colored) noise, but without deciding on *the extent* to which it is correlated. A convenient way to model the correlated noise is to use a Gauss-Markov process (GMP). The GMP has been previously used with some success in modelling the acceleration of an airplane in flight (see, e.g., [38], [39], [40]), which allowed an improved tracking of air maneuvers. In this section we show that by modelling the velocity with a Gauss-Markov process, we obtain a model of which RW and NCV are only special cases and which is able to account for more general dynamics; we will call this model the liberal model. In all our subsequent experiments we have used separate dynamic models to model the target's horizontal and vertical motion independently. We therefore require derivation of the model for one-dimensional problems only. After derivation of the liberal model we also provide an analysis of its parameters.

We start by noting that changes in the position $x(t)$ arise due to a non-zero velocity $v(t)$ of the target, i.e., $\dot{x}(t) = v(t)$. The velocity $v(t)$ is modelled as a non-zero-mean correlated noise

$$v(t) = \tilde{v}(t) + \hat{v}(t), \quad (4)$$

where $\tilde{v}(t)$ denotes a zero-mean correlated noise and $\hat{v}(t)$ is the current mean of the noise; we will call $\hat{v}(t)$ the *input velocity*. We model the correlated noise $\tilde{v}(t)$ as a Gauss-Markov process with an autocorrelation function $R_{\tilde{v}}(\tau) = \sigma e^{-\beta|\tau|}$, where σ^2 is the variance of the process noise, and β is the correlation time constant. To derive the dynamic model of the process (4) in a form which we can use for tracking, we have to first find a stochastic differential equation (s.d.e.) of the process (4), governed by a white-noise process, and then find its discretized counterpart.

Applying a shaping filter (see, e.g., [41], page 137) to the correlated noise $\tilde{v}(t)$ gives the following s.d.e.

$$\dot{\tilde{v}}(t) = -\beta\tilde{v}(t) + \sqrt{q_c}u(t), \quad (5)$$

where $q_c = 2\beta\sigma^2$ is the spectral density of the equivalent white-noise process acting on $\dot{\tilde{v}}(t)$ and where, $u(t)$ denotes a unit-variance white-noise process. The continuous-time s.d.e. of (4) can now be derived by expressing $\tilde{v}(t)$ in (4) and plugging it into (5),

$$\dot{\tilde{v}}(t) = -\beta v(t) + \beta\hat{v}(t) + \sqrt{q_c}u(t). \quad (6)$$

In order to arrive at a discretized form of the above model, we first note from (4) that $\dot{\tilde{v}}(t) = \frac{\partial}{\partial t}(v(t) - \hat{v}(t))$ and assume

that the input velocity $\hat{v}(t)$ remains constant over a sampling interval[1]. Thus we obtain

$$\dot{v}(t) = -\beta v(t) + \beta\hat{v}(t) + \sqrt{q_c}u(t). \tag{7}$$

Since $\dot{x}(t) = v(t)$, we can write the complete system s.d.e. in the matrix form

$$\dot{\mathbf{X}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -\beta \end{bmatrix} \mathbf{X}(t) + \begin{bmatrix} 0 \\ \beta \end{bmatrix} \hat{v}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sqrt{q_c}u(t), \tag{8}$$

where we have defined $\mathbf{X}(t) = [x(t), v(t)]^{\mathrm{T}}$. Applying a standard discretization (see, e.g.,[42] Appendix B) to (8) we obtain the continuous-time liberal model (4) with discretized states $\mathbf{X}_k = [x_k, v_k]^T$:

$$\mathbf{X}_k = \Phi\mathbf{X}_{k-1} + \Gamma\hat{v}_{k-1} + W_k, \tag{9}$$
$$\Phi = \begin{bmatrix} 1 & \frac{1-e^{-\Delta t\beta}}{\beta} \\ 0 & e^{-\Delta t\beta} \end{bmatrix}, \Gamma = \begin{bmatrix} \frac{\Delta t\beta - 1 + e^{-\Delta t\beta}}{\beta} \\ 1 - e^{-\Delta t\beta} \end{bmatrix}.$$

From the theory of estimation and control (e.g., [19] page 186) the analytic from of the equation (9) is known as the *discrete-time linear stochastic dynamic system*, in which Φ corresponds to the *state transition matrix* and Γ is the *discrete-time gain* (assumed constant over the sampling interval) through which a deterministic sequence $\hat{v}_{k-1}$ enters the system. In our treatment, $\hat{v}_{k-1}$ in (9) is the input velocity for the current time-step k, Δt is the time-step length, and W_k is a white-noise sequence with a covariance matrix

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} q_c, \tag{10}$$
$$q_{11} = \frac{1}{2\beta^3}(2\Delta t\beta - 1 + 4e^{-\Delta t\beta} - e^{-2\Delta t\beta}),$$
$$q_{12} = \frac{1}{2\beta^2}(1 + e^{-2\Delta t\beta} - 2e^{-\Delta t\beta}),$$
$$q_{22} = \frac{1}{2\beta}(1 - 2e^{-2\Delta t\beta}).$$

Note that there are two parameters which can be set in the liberal model (9, 10): one is the correlation-time parameter β and the other is the spectral density q_c of the noise. In the following we first give an analysis of how the parameter β influences the structure of the proposed liberal model. Then we propose a method for selecting the spectral density q_c for a given class of objects.

1) Parameter β: In terms of the parameter β, the dynamic properties of the liberal model (9) can be considered as being in between a random-walk and a nearly-constant-velocity model; this can be seen by limiting β to zero, or to infinity. In the case of $\beta \to 0$, the model takes the form of a pure NCV model with a state transition matrix $\Phi_{\beta\to 0}$ and the input matrix $\Gamma_{\beta\to 0}$

$$\Phi_{\beta\to 0} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \Gamma_{\beta\to 0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{11}$$

[1]Note that this assumption may be restrictive if the target significantly changed its motion during a time step. In our application, however, we assume a camera frame-rate of 20 to 30 frames per second, which results in 33-50ms time steps. We can therefore reasonably assume a constant mean of the GMP and absorb the unmodelled changes into the white noise sequence.

On the other hand, at $\beta \to \infty$, the model takes the form of a RW model with the state transition matrix $\Phi_{\beta\to\infty}$ and the input matrix $\Gamma_{\beta\to\infty}$

$$\Phi_{\beta\to\infty} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \Gamma_{\beta\to\infty} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{12}$$

Note that the values of $\Gamma_{\beta\to\infty}$ are nonzero, thus the input velocity has to be set to zero, $\hat{v}_{k-1} = 0$, to obtain the pure random-walk model.

We have seen thus far that the liberal dynamic model takes the structure of RW and NCV models at the limiting values of β. But what happens when β is set to somewhere in between zero and infinity? To get a better understanding of that, it is beneficial to rewrite the model in the following way. Let $\mathbf{x}_k = [x_k, v_k]^T$ denote the state at the time-step k with position x_k and velocity v_k, and, similarly, let $\mathbf{x}_{k-1} = [x_{k-1}, v_{k-1}]^T$ denote the state at the previous time-step $k-1$. We also rewrite the elements of the system transition matrix Φ and the input matrix Γ (9) in the following abbreviated form

$$\Phi = \begin{bmatrix} 1 & \phi_{1,2} \\ 0 & \phi_{2,2} \end{bmatrix}, \Gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}. \tag{13}$$

Note from (9) that Φ and Γ depend on the size of the time-step Δt, which is the time between one and the next measurement. Without loss of generality we can set the time-step to unity, i.e., $\Delta t = 1$. For completeness, let us also define the values of the noise terms, at time-step k, acting on the position and velocity by $W_k = [w_{xk}, w_{vk}]^T$. Now we can rewrite the liberal model (9) in terms of the state's components as

$$\begin{aligned} x_k &= x_{k-1} + \phi_{1,2}v_{k-1} + \gamma_1\hat{v}_{k-1} + w_{xk} \\ v_k &= \phi_{2,2}v_{k-1} + \gamma_2\hat{v}_{k-1} + w_{vk}. \end{aligned} \tag{14}$$

Since we have set $\Delta t = 1$, we have from (9) and (14)

$$\phi_{1,2} + \gamma_1 \equiv 1 \text{ and } \phi_{2,2} + \gamma_2 \equiv 1.$$

This means that $\phi_{1,2}$ and γ_1 are the proportions in which the internal velocity v_{k-1} and the *input* velocity $\hat{v}_{k-1}$ will be combined into the deterministic part of the velocity acting on the current *position* x_k.[2] Similarly, $\phi_{2,2}$ and γ_2 are the proportions in which the internal velocity v_{k-1} and the *input* velocity $\hat{v}_{k-1}$ will be combined into the deterministic part of the velocity acting on the current *velocity* v_k. With Δt fixed, the values of the mixing factors $\phi_{1,2}$, $\phi_{2,2}$, γ_1 and γ_2 depend solely on β. We show this dependence in Figure 2.

From the Figure 2 we see that by increasing β, the influence of the input velocity $\hat{v}_{k-1}$ increases in (14), and for a very large β, the internal velocity v_{k-1} is completely disregarded by the dynamic model as $\phi_{1,2}$ and $\phi_{2,2}$ of (14) tend to zero. On the other hand, γ_1 and γ_2 tend to zero for small values of β. This means that we can consider β as a parameter that specifies an a-priori confidence of the input $\hat{v}_{k-1}$ and internal velocity v_{k-1}. If, for example, we know that $\hat{v}_{k-1}$ is very accurate, then β should be set to a very large value. Otherwise, smaller β should be used.

[2]The *nondeterministic* part of the velocity acting on the current position x_k is the white noise w_{xk}.

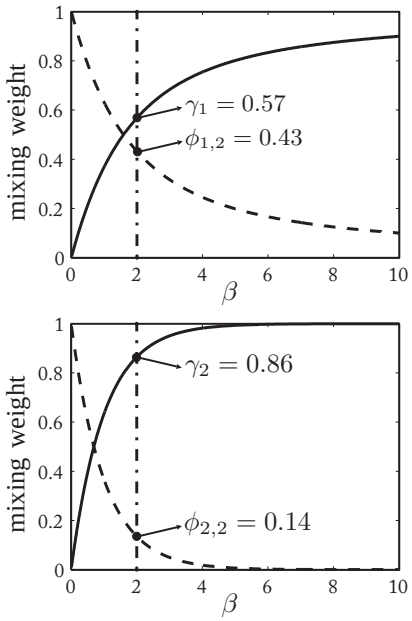


Fig. 2. The values of the components of Φ and Γ at $\Delta t = 1$ w.r.t. different values of β. The left graphs (upper row) show $\phi_{1,2}$ and γ_1 which are used for mixing v_{k-1} and $\hat{v}_{k-1}$, respectively, in estimating the current position x_k. The right graphs (lower row) show the values of $\phi_{2,2}$ and γ_2 which are used for mixing v_{k-1} and $\hat{v}_{k-1}$, respectively, in estimating the current velocity v_k. In (a), the values of $\phi_{1,2}$ are depicted by the dashed line, while the values of γ_1 are depicted by the full line. Similarly, in (b), the values of $\phi_{2,2}$ are depicted by the dashed line, while the values of γ_2 depicted by the full line. In both images, the upright dash-dotted line depicts the values of $\phi_{1,2}$, $\phi_{2,2}$, γ_1 and γ_2 at $\beta = 2$. For convenience, these values are written out at the marked locations.

The two-stage dynamic model which is presented in this paper usually yields reasonable estimates of the input velocity $\hat{v}_{k-1}$ for a large class of targets. In practice we have observed that it is thus beneficial to let the input velocity $\hat{v}_{k-1}$ have a dominant effect over v_{k-1} in estimating the current *velocity* v_k. However, if we want the liberal model to be able to account for a greater agility of the target, it is also beneficial to let the internal velocity v_{k-1} to have a greater effect on predicting the current *position* x_k. We have found that these requirements are sufficiently well met at $\beta \approx 2$ which is the value we use in all subsequent experiments. The values of $\phi_{1,2}$, γ_1, $\phi_{2,2}$ and γ_2 at $\beta = 2$ are shown in Figure 2.

2) Selecting the spectral density: Another important parameter of the liberal model (9) is the spectral density q_c of the process noise (10). Note that in many cases it is possible to obtain some general characteristics of the dynamics of the class of objects which we want to track. Specifically, the expected squared distance σ_m^2 that objects of certain class travel between two time-steps is often available. Assuming that we have some estimate of σ_m^2, and that the time-step size Δt and the parameter β are known, we now derive a rule-of-thumb rule for selecting the spectral density q_c.

To derive the rule-of-thumb, let us consider the following example. Assume that at time-step $k-1$ a target is located at the origin of the coordinate system, i.e., $x_{k-1} = 0$, and begins moving with a velocity $v_{k-1} \sim q_{22} q_c$, i.e., $\mathbf{X}_{k-1} = [0, v_{k-1}]^{\mathrm{T}}$. Assuming that the input velocity $\hat{v}_{k-1}$ in (9) is zero, the target's state after a single time-step is

$$\mathbf{X}_k = \Phi \mathbf{X}_{k-1} + W_k. \tag{15}$$

The covariance of the position at time-step k is

$$\begin{aligned} \mathbf{P} &= \langle \mathbf{X}_k \mathbf{X}_k^T \rangle \\ &= \langle \Phi \mathbf{X}_{k-1} \mathbf{X}_{k-1}^T \Phi^T \rangle + \langle \Phi \mathbf{X}_{k-1} W_k^T \rangle \\ &\quad + \langle W_k \mathbf{X}_{k-1}^T \Phi^T \rangle + \langle W_k W_k^T \rangle, \end{aligned} \tag{16}$$

where $\langle \cdot \rangle$ denotes the expectation operator. Since the state $\mathbf{X}_{k-1}$ is not correlated with the noise W_k and since $Q \triangleq \langle W_k W_k^T \rangle$, the equation (16) simplifies into

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \Phi \langle \mathbf{X}_{k-1} \mathbf{X}_{k-1}^{\mathrm{T}} \rangle \Phi^{\mathrm{T}} + \mathrm{Q}. \tag{17}$$

Since p_{11} in (17) is just the expected squared change of target's position in consecutive time steps, i.e., $p_{11} = \sigma_m^2$, we have

$$\begin{aligned} \sigma_m^2 &= p_{11} \\ &= (\frac{1 - e^{-\Delta t \beta}}{\beta})^2 \langle v_{k-1} v_{k-1} \rangle + q_{11} q_c. \end{aligned} \tag{18}$$

Since we have defined earlier $v_{k-1} \sim q_{22} q_c$, we know that $\langle v_{k-1} v_{k-1} \rangle = q_{22} q_c$, and (18) is rewritten into

$$\sigma_m^2 = ((\frac{1 - e^{-\Delta t \beta}}{\beta})^2 q_{22} + q_{11}) q_c. \tag{19}$$

Inverting (19) finally gives the rule-of-thumb rule for selecting the spectral density

$$q_c = \sigma_m^2 ((\frac{1 - e^{-\Delta t \beta}}{\beta})^2 q_{22} + q_{11})^{-1}. \tag{20}$$

B. The conservative model

In contrast to the liberal model, the conservative model assumes that the target's velocity does not change abruptly and approximates the local dynamics by fitting a linear model to the past filtered states. This model is used in the two-stage dynamic model to regularize the estimated target positions from the liberal model.

Let $\hat{o}_{k-K:k-1} = \{\hat{o}_i\}_{i=k-K}^{k-1}$ denote a sequence of the K past regularized (e.g., horizontal) positions $\hat{o}_i$ of the target, and let $\pi_{k-K:k-1} = \{\pi_i\}_{i=k-K}^{k-1}$ denote the set of their weights. These weights indicate how well the corresponding positions have been estimated. The conservative model aims to locally approximate the sequence $\hat{o}_{k-K:k-1}$ by the following linear model

$$\tilde{x}_i = \hat{v}_{\hat{\mathrm{o}}k-1} i + \hat{a}_{\hat{\mathrm{o}}k-1}, \tag{21}$$

where $\tilde{x}_i$ is the target's linearly approximated position at time-step i. The subscript $(\cdot)_{\hat{\mathrm{o}}k-1}$ in (21) is used to indicate that the parameters have been estimated using a sequence of filtered positions up to position $\hat{o}_{k-1}$. Since all positions are usually not estimated equally well, and since the recent positions are more relevant for estimating the target's current dynamics, the parameters $\hat{v}_{\hat{\mathrm{o}}k-1}$ and $\hat{a}_{\hat{\mathrm{o}}k-1}$ of the linear model (21) are estimated such that they minimize the following weighted sum of squared differences

$$C_{k-1} = \sum_{i=k-K}^{k-1} G_{k-1}^{(i)} (\hat{o}_i - \tilde{x}_i)^2 \tag{22}$$

in which the weights $G^{(i)}_{(\cdot)}$ are defined as

$$G^{(i)}_j = \pi_i e^{-\frac{1}{2}\frac{(i-j)^2}{\sigma_o^2}}. \quad (23)$$

While the first term in (23) reflects the likelihood of the position o_i, the second term is a Gaussian which assigns higher a-priori weights to the more recent states. In practice this means that we only consider $K = 3\sigma_o$ past positions in (22), since the a-priori weights of all the older positions are negligible. Note that the Gaussian form was used for the last term exclusively to attenuate the importance of the older positions. In general, however, other forms that exhibit similar behavior (e.g., an exponential function) could have been used.

From (21) and (22) we can now find $\hat{v}_{\hat{o}k-1}$ and $\hat{a}_{\hat{o}k-1}$ simply by setting the corresponding partial derivatives to zero

$$\frac{\partial C_{k-1}}{\partial \hat{v}_{\hat{o}k-1}} \equiv 0 \quad , \quad \frac{\partial C_{k-1}}{\partial \hat{a}_{\hat{o}k-1}} \equiv 0, \quad (24)$$

which gives

$$\begin{aligned} \hat{v}_{\mathrm{o}k-1} &= \frac{\sum_{i=k-K}^{k-1} iG^{(i)}_{k-1}\hat{o}_i + A_{k-1}B_{k-1}(\sum_{i=k-K}^{k-1} iG^{(i)}_{k-1})}{\sum_{i=k-K}^{k-1} i^2 G^{(i)}_{k-1} - A_{k-1}(\sum_{i=k-K}^{k-1} iG^{(i)}_{k-1})^2}, \\ \hat{a}_{\hat{o}k-1} &= A_{k-1}(B_{k-1} - \hat{v}_{\hat{o}k-1}\sum_{i=k-K}^{k-1} iG^{(i)}_{k-1}), \end{aligned} \quad (25)$$

where we have defined

$$A_{k-1} = (\sum_{i=k-K}^{k-1} G^{(i)}_{k-1})^{-1} \; ; \; B_{k-1} = \sum_{i=k-K}^{k-1} G^{(i)}_{k-1}\hat{o}_i \quad (26)$$

The conservative model is completely defined with parameters $\hat{v}_{\hat{o}k-1}$ and $\hat{a}_{\hat{o}k-1}$. A conservative prediction of the target's position $\tilde{x}_k$ at time-step k is calculated as

$$\tilde{x}_k = \hat{v}_{\hat{o}k-1}k + \hat{a}_{\hat{o}k-1}. \quad (27)$$

Note that the parameter $\hat{v}_{\hat{o}k-1}$ can be interpreted as a conservative approximation of the target's current velocity calculated from a sequence of $\hat{o}_{k-K:k-1}$ filtered positions.

C. A two-stage dynamic model

The liberal model in section III-A was derived from a continuous-time non-zero-mean Gauss-Markov process and is capable of accounting for various types of dynamics, ranging from a random walk to the nearly-constant-velocity behavior. This model can be readily used within a particle filter to estimate the posterior over the target's state recursively in time. A mean value calculated on this posterior can be taken as a minimum-mean-squared-error estimate of the target's current state (e.g., position). While the liberal model can potentially well explore the target's state-space, it requires estimation of the mean value of the Gauss-Markov process (the input velocity) and the quality of the state estimation will quickly deteriorate with decreasing the number of particles in the particle filter. In cases when the target's velocity is locally-linear, the conservative model from Section III-B may provide a better approximation of the target's dynamics, however, it lacks the ability of the liberal model's exploration of the state-space.

We therefore propose a two-stage dynamic model, which combines the liberal model with the conservative model from section III-B into a two-stage probabilistic tracker as follows. Assume we have a sequence of the past K filtered target positions $\hat{o}_{k-K:k-1}$ and their weights $\pi_{k-K:k-1}$, and that we have fitted a linear model (27) to this sequence. Recall that the parameter $\hat{v}_{\hat{o}k-1}$ in (27) is a conservative estimate of the target's velocity from the target's positions up to time-step $k-1$. At time-step k, when a new image arrives, we can use this conservative estimate $\hat{v}_{\hat{o}k-1}$ to approximate the input velocity $\hat{v}_{k-1}$ for the liberal model (9). With the input velocity approximated, the liberal model, which accounts for the non-constant velocity, can be used within a particle filter to approximate the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ over the target's current state. The mean value of the posterior (3) is the liberal approximation of the target's state $\hat{\mathbf{x}}_k = [\hat{x}_k, \hat{v}_k]^T$. The variance of the liberal estimate of the target's position can be reduced by taking into account the conservative estimate as well. The conservative model is used to generate the current conservative prediction (27) of the target's position $\tilde{x}_k$. The liberal estimate can be fused with the conservative estimate by using the visual data as follows. We measure the likelihood[3] $w_{\hat{x}_k} = p(\mathbf{y}_k|\hat{x}_k)$ that the target is located at the liberal estimate of it's position $\hat{x}_k$ and we can do similarly for the conservative estimate $\tilde{x}_k$, $w_{x(k)} = p(\mathbf{y}_k|\tilde{x}_k)$. The conservative and the liberal estimates are then fused as

$$\hat{o}_k = \frac{\tilde{x}_k \cdot w_{\tilde{x}_k} + \hat{x}_k \cdot w_{\hat{x}_k}}{w_{\tilde{x}_k} + w_{\hat{x}_k}}, \quad (28)$$

the corresponding weight π_k of the new regularized position $\hat{o}_k$ is evaluated using the visual likelihood function, $\pi_k = p(\mathbf{y}_k|\hat{o}_k)$, and the new parameters ($\hat{v}_{\hat{o}k}$ and $\hat{a}_{\hat{o}k}$) of the conservative model (27) are recalculated using (25). The new regularized state from the two-stage dynamic model is then constructed from the fused position and conservative velocity, $\mathbf{o}_k = [\hat{o}_k, \hat{v}_{\mathrm{o}k}]^T$. The prediction of the regularized state $\tilde{\mathbf{o}}_{k+1}$ from the two-stage dynamic model for the next time-step can be calculated under the assumption of a locally-constant velocity as

$$\tilde{\mathbf{o}}_{k+1} = F\mathbf{o}_k \quad ; \quad F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}. \quad (29)$$

Note that the proposed two-stage algorithm essentially implements a well-known concept of information fusion. In particular, we can summarize an iteration of the algorithm in the following three steps: (i) generating a "conservative prediction" of the target's position from the conservative model, (ii) generating the "liberal measurement" from the liberal model in the particle filter (estimated mean value of the posterior) and (iii) fusing the "prediction" and the "measurement" into a single estimate using the visual data (likelihood). The concept of fusing prediction and measurement by linear (weighted) combination is, for example, a

[3]The visual likelihoods $p(\mathbf{y}_k|\hat{x}_k)$ and $p(\mathbf{y}_k|\tilde{x}_k)$ refer to the likelihood function, which we also use in the particle filter and which evaluates the likelihood that a target is located at some position or a state, e.g., the color-based likelihood function.

central approach of the Kalman filter [8]. There, the weights for fusing prediction and measurement by linear interpolation are computed from the covariance of the prediction's and measurement noise. In the two-stage dynamic model, the weights are provided directly from the visual model which tells us how well each hypothesis (*conservative prediction* and the *liberal measurement*) is supported in the visual data. The linear interpolation between various estimates is also central to the particle filter: each particle produces a state and a belief that the target is actually in that state. While each particle is in itself an estimator of the target's position, the variance of this estimator is usually very large. However, the linear averaging of particles gives a mean value of the posterior, which is an estimator with a reduced (but potentially still large) variance. By enforcing regularization from our conservative prediction, the variance can be further reduced. In particular, the regularization (implemented here as linear interpolation) can be viewed as adding additional (conservative) samples to the particles from the particle filter, appropriately weighting them (using the visual data) and calculating their mean value. Effectively, this combination of two alternative hypotheses allows the two-stage dynamic model to handle constant as well as nonconstant motions. The reduction in the variance of the estimator and improved tracking of the various motions are validated in our experiments in the section IV .

D. Application to blobtracking

Here we present an implementation of the two-stage dynamic model on an example of particle-filter-based blob tracker. This tracker is used in our experiments to compare the performance of the proposed model with common dynamic models used for blob tracking. In a probabilistic blob tracker, the target is commonly modelled by an elliptical or rectangular region and its appearance is encoded by a color histogram (e.g., [43], [44], [26], [45], [30]). The color-based likelihood function for the particle filter $p(\mathbf{y}_k|\mathbf{x}_k)$ is thus calculated through comparison of reference color histograms to the histograms extracted at the target's state $\mathbf{x}_k$ (see, e.g., [45]). We use two one-dimensional two-stage models for modelling the motion of target's horizontal and vertical position (i.e., ellipse's center) and two one-dimensional random-walk models for the target's width and height. The target's state is thus defined as $\mathbf{x}_k = [x_k, v_{xk}, y_k, v_{yk}, H_{xk}, H_{yk}]^T$, where $[x_k, y_k]$, $[v_{xk}, v_{yk}]$, $[H_{xk}, H_{yk}]$ are the target's position, velocity and size, respectively. The liberal model (9), which is also used for the state transition model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ in the particle filter, is therefore defined as

$$\begin{aligned} \mathbf{x}_k &= \Phi_L \mathbf{x}_{k-1} + \Gamma_L \hat{\mathbf{v}}_{k-1} + W_{Lk}, \qquad (30)\\ \Phi_L &= diag[\Phi, \Phi, 1, 1], \quad \Gamma_L = [diag[\Gamma, \Gamma], 0_{2\times 1}]^T, \end{aligned}$$

where Φ and Γ are defined in (9)[4], $0_{2\times 1} = [0, 0]^T$, $\hat{\mathbf{v}}_{k-1} = [\hat{\mathbf{v}}_{xk-1}, \hat{\mathbf{v}}_{yk-1}]^T$ are the horizontal and vertical input velocities, and W_{Lk} is a discrete-time white noise sequence defined by a zero-mean normal distribution, $W_{Lk} \sim \mathcal{N}(0, Q_L)$, with covariance matrix

$$Q_L = diag[Q, Q, I_{2\times 2}\sigma_H^2], \qquad (31)$$

[4]Without loss of generality the time-step size in Φ and Γ is set to $\Delta t = 1$

with Q defined in (10) and $I_{2\times 2} = diag[1, 1]$. The parameter σ_H corresponds to the noise in the random-walk models on the target's size. As in [46], we fix this parameter in all our subsequent experiments such that the target's size does not change between two time-steps by more than 15%. The conservative model uses sequences of the past K regularized states $\hat{\mathbf{o}}_{k-K:k-1}$ and weights $\pi_{k-K:k-1}$ to fit a linear model to the regularized positions (Section III-B) and estimates $\hat{\mathbf{v}}_{\mathbf{o}k-1}$ and $\hat{\mathbf{a}}_{\hat{\mathbf{o}}k-1}$. A conservative prediction of the target's position $\tilde{\mathbf{x}}_k$ at time-step k is then defined as

$$\tilde{\mathbf{x}}_k = \hat{\mathbf{v}}_{\hat{\mathbf{o}}k-1} k + \hat{\mathbf{a}}_{\hat{\mathbf{o}}k-1}. \qquad (32)$$

The liberal model provides a liberal estimate of the target's position $\hat{\mathbf{x}}_k$, which is fused with the conservative prediction $\tilde{\mathbf{x}}_k$ into a regularized position. The conservative estimate of the velocity $\hat{\mathbf{v}}_{\hat{\mathbf{o}}k-1}$ is recalculated and combined with the regularized position into a new regularized state

$$\mathbf{o}_k = [o_{xk}, o_{vxk}, o_{yk}, o_{vyk}, o_{Hxk}, o_{Hyk}]^T. \qquad (33)$$

This is the output of the two-stage dynamic model. The prediction of the two-stage dynamic model is made under the assumption of constant velocity and is defined as

$$\tilde{\mathbf{o}}_{k+1} = F_T \mathbf{o}_k \quad ; \quad F_T = diag[F, F, F], \qquad (34)$$

with F is defined in (34). Following the above description, we summarize the particle-filter-based blob tracker with a two-stage dynamic model in Algorithm 1.

1) The parameters: Since the two-stage dynamic model in Algorithm 1 is composed of the liberal and the conservative model, there are a few parameters that have to be set. Two parameters have to be set for the liberal model (9): the parameter β and the spectral density q_c of the process noise. A detailed discussion of how the parameter β influences the structure of the liberal model was provided in Section III-A1. There we have concluded, that the required dynamic properties of the liberal model are met at $\beta = 2$. The remaining parameter of the liberal model, the spectral density q_c, has to be specified for the problem at hand and we have proposed a principled way to selecting q_c in Section III-A2. The conservative model requires setting a single parameter σ_o, which effectively determines the number of the recent regularized states which are considered in the linearization. We set this parameter using the following rationale. We can assume that the objects which are considered in our applications do not usually change their velocity drastically within a half of the second. Since most of our recordings used in the experiments are recorded at 25 frames per second, this means that we consider only $K = \frac{1}{2}25 \approx 13$ recent regularized states. We have noted in Section III-B that $K = 3\sigma_o$, which means that $\sigma_o = 4.3$. For convenience, we summarize the parameters in Table I.

TABLE I
PARAMETERS OF THE TWO-STAGE DYNAMIC MODEL.

The liberal dynamic model (Section III-A)
• Parameter $\beta = 2$.
• Spectral density q_c selected by the rule from Section III-A2
The conservative dynamic model (Section III-B)
• Parameter $\sigma_o = 4.3$.

Algorithm 1 A probabilistic blob-tracker with a two-stage dynamic model.

Input:
- $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\ldots$ estimate of the posterior pdf from the previous time-step
- $k-1$.
- $\hat{\mathbf{o}}_{k-K:k-1}, \pi_{k-K:k-1}\ldots$ sequence of the previous K regularized positions and weights.
- $\hat{\mathbf{v}}_{\hat{\mathbf{o}}k-1}, \hat{\mathbf{a}}_{\hat{\mathbf{o}}k-1}\ldots$ parameters of the conservative model from time-step $k-1$.
- The current image.

Output:
- $\mathbf{o}_k\ldots$ the new regularized state.
- $p(\mathbf{x}_k|\mathbf{y}_{1:k})\ldots$ the new estimate of the posterior pdf.
- $\hat{\mathbf{o}}_{k-K+1:k}, \pi_{k-K+1:k}\ldots$ augmented sequence of regularized positions and weights.
- $\hat{\mathbf{v}}_{\hat{\mathbf{o}}k}, \hat{\mathbf{a}}_{\hat{\mathbf{o}}k}\ldots$ new parameters of the conservative model.

Iteration:

1: Approximate the input velocity $\hat{\mathbf{v}}_{k-1}$ of the liberal model (30) by a conservative estimate $\hat{\mathbf{v}}_{\hat{\mathbf{o}}k-1}$.
2: Execute an iteration of the particle filter using a color-based likelihood function $p(\mathbf{y}_k|\mathbf{x}_k)$ and the liberal model (30) for the state transition model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The result is the approximation of the new posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.
3: Calculate the liberal estimate of the state $\hat{\mathbf{x}}_k = \langle\mathbf{x}_k\rangle_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}$ (3).
4: Calculate the conservative prediction $\tilde{\mathbf{x}}_k$ (32).
5: Fuse the liberal and conservative estimates into a regularized position $\hat{\mathbf{o}}_k$ according to Section III-C, calculate the weight $\pi_k = p(\mathbf{y}_k|\hat{\mathbf{o}}_k)$ and augment the weighted sequence of regularized positions into $\hat{\mathbf{o}}_{k-K+1:k}$ and $\pi_{k-K+1:k}$.
6: Recalculate parameters $\hat{\mathbf{v}}_{\hat{\mathbf{o}}k}$ and $\hat{\mathbf{a}}_{\hat{\mathbf{o}}k}$ from $\hat{\mathbf{o}}_{k-K+1:k}$ and $\hat{\pi}_{k-K+1:k}$.
7: Construct a regularized state $\mathbf{o}_k$ (33) as a concatenation of the regularized position $\hat{\mathbf{o}}_k$ and conservative velocity $\hat{\mathbf{v}}_{\hat{\mathbf{o}}k}$.

IV. Experimental Study

We carried out two sets of experiments to evaluate the performance of the proposed two-stage dynamic model from Section III-C. In the first set of experiments (Section IV-A) we have tracked persons moving on a predefined path on the ground. This experiment was designed for quantitative and qualitative comparison of the estimation accuracy of the proposed two-stage dynamic model and the two commonly-used dynamic models. The second experiment was designed to demonstrate the generality of the proposed dynamic model and to demonstrate how it can help to reduce the visual ambiguity which occurs when the target is moving close to another visually similar object (Section IV-B). In that experiment, we have applied the two-stage dynamic model to tracking person's palms and to tracking a person in a squash match. In all the experiments, the target was described by an elliptical region and its visual properties were encoded by a color histogram (see, e.g., [42], page 39). For the videos demonstrating the results presented in this paper and additional examples of the tracker's performance, please see http://vicos.fri.uni-lj.si/data/matejk/tracking/DynamicModel/Sub/index.html.

A. Experiment 1: accuracy of estimation

Seven players of handball were instructed to run on a predefined path drawn on the court (Figure 3). The path was designed such that the observed motion involved accelerations, decelerations and rapid changes in the direction of motion. The scene was recorded with a camera mounted on the ceiling of the sport's hall, such that the size of each player was approximately 10×10 pixels. The video was recorded at the frame rate of 25 frames per second. Each player was manually tracked thirty times through each frame and the average of the thirty trajectories obtained for each player was taken as the ground truth. In this way approximately 273 ground-truth positions $p_k = (x_k, y_k)$ per player were obtained.

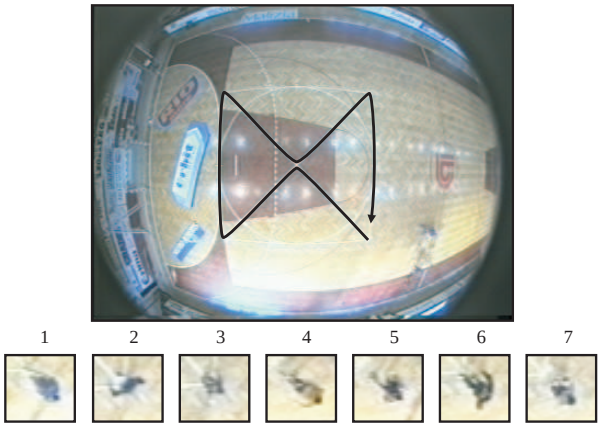


Fig. 3. Seven players and the path used in the first experiment.

All seven players from Figure 3 were then tracked with three trackers: Two reference trackers and the proposed tracker. The only difference between these trackers was in the dynamic models they used for modelling the dynamics of the player's position. The proposed tracker, we denote it by $\mathbf{T}_{\mathrm{TS}}$, was the two-stage probabilistic tracker from Section III-D. The reference trackers were essentially the color-based particle filters from [42](chapter 3), which employed two widely-used dynamic models on the player's position. The first reference tracker, $\mathbf{T}_{\mathrm{RW}}$, used the random-walk model, while the second reference tracker used the nearly-constant-velocity model; we denote this tracker by $\mathbf{T}_{\mathrm{NCV}}$. All three trackers used random-walk models to model the dynamics of the player's size.

The parameters of the RW and NCV dynamic models in $\mathbf{T}_{\mathrm{NCV}}$ and $\mathbf{T}_{\mathrm{RW}}$ were learned from the ground truth. In particular, the only parameter of the RW and NCV model that has to be specified is the spectral density of the process noise. The spectral densities were estimated using a linear-dynamic-system learning method (see, e.g., [7] pages 635-644) from $7 \times 30 = 210$ ground truth trajectories. The method yielded the spectral density $q_{\mathrm{RW}} = 4.6$ for the RW model and the spectral density $q_{\mathrm{NCV}} = 0.4$ for the NCV model. We have

observed in experiments that the estimated spectral density for RW was too small and, in practice, the tracker was failing frequently for some of the players. For that reason, the spectral density in the RW model was increased to $q_{\mathrm{RW}} = 6$ in the experiments.

The spectral density q_c of the liberal model (9) in $\mathbf{T}_{\mathrm{TS}}$ was determined using the rule-of-thumb rule, which we have proposed in Section III-A2. Recall that the rule requires us to provide an estimate of the squared distance σ_m^2 that the objects under consideration are expected to travel between two time steps. Since we track sports players in our experiment, we can find σ_m^2 as follows. Based on the findings of Bon et al. [47], who refer to Kotzamanidis [48], Erdmann [49] and Bangsbo [50] regarding the dynamics of handball/soccer players, we can estimate the highest velocity of a player as $v_{max} = 8.0$m/s. At a frame rate of 25frames/s we can say that $v_{max} = 0.32$m/frame. During tracking, the player is usually determined by an ellipse that is approximately the size of his/hers shoulders. We estimate this size to be $H_t \approx 0.4$m. Assuming a Gaussian form of the velocity distribution, the highest velocity can be approximated with three standard deviations of the Gaussian. This gives $v_{max} = 3\sigma_{xy}$/frame and the parameter $\sigma_m = H_t \frac{0.32}{3 \cdot 0.4} \doteq H_t \frac{1}{4}$. Using the rule-of-thumb rule (20) the spectral density of the liberal model is thus estimated as

$$q_c = (H_t \tfrac{1}{4})^2 (q_{11} + q_{22} (\tfrac{1-e^{-\beta}}{\beta})^2)^{-1}, \tag{35}$$

where q_{11} and q_{22} are defined in (10).

1) Quantitative evaluation: Using the parameters given above, all seven players from Figure 3 were tracked thirty times with the trackers $\mathbf{T}_{\mathrm{RW}}$, $\mathbf{T}_{\mathrm{NCV}}$ and $\mathbf{T}_{\mathrm{TS}}$. Thus $K = 30$ trajectories per player were recorded for each tracker. Note that $\mathbf{T}_{\mathrm{RW}}$ and $\mathbf{T}_{\mathrm{NCV}}$ have failed during tracking on a few occasions by losing the player. In those situations, tracking was repeated and only the trajectories where tracking did not fail were considered for evaluation. In all experiments $\mathbf{T}_{\mathrm{TS}}$ never failed.

A standard one-sided hypothesis testing [19] was applied to determine whether the accuracy of estimation by $\mathbf{T}_{\mathrm{TS}}$ was greater than the accuracy of the reference trackers $\mathbf{T}_{\mathrm{RW}}$ and $\mathbf{T}_{\mathrm{NCV}}$. In the following, when not referring to a specific tracker, we will abbreviate the reference trackers by $\mathbf{T}_{\mathrm{REF}}$. The performance of the trackers in the r-th repetition was defined in terms of the root-mean-square (RMS) error as

$$C^{(r)} \triangleq \frac{1}{7} \sum_{i=1}^{7} \left(\frac{1}{K} \sum_{k=1}^{K} \| {}^{(i)}\mathbf{p}_k - {}^{(i)}\hat{\mathbf{p}}_k^{(r)} \|^2 \right)^{\frac{1}{2}}. \tag{36}$$

In (36) ${}^{(i)}\mathbf{p}_k$ denotes the ground-truth position at time-step k for the i-th player, ${}^{(i)}\hat{\mathbf{p}}_k^{(r)}$ is the corresponding estimated position and $\| \cdot \|$ is the l_2 norm. At each repetition, a *sample-performance-difference*

$$\Delta^{(r)} = C_{REF}^{(r)} - C_{TS}^{(r)} \tag{37}$$

was calculated. The term $C_{TS}^{(r)}$ was the cost value (36) of $\mathbf{T}_{TS}$, while $C_{REF}^{(r)}$ presented the cost value of the reference tracker $\mathbf{T}_{REF}$.

In our case the null hypothesis H_0 was that $\mathbf{T}_{\mathrm{TS}}$ is *not* superior to $\mathbf{T}_{\mathrm{REF}}$. For each tracker we calculated the sample-performance-difference mean

$$\bar{\Delta} = \frac{1}{R} \sum_{r=1}^{R} \Delta^{(r)} \tag{38}$$

and its standard error

$$\sigma_{\bar{\Delta}} = \sqrt{\frac{1}{R^2} \sum_{r=1}^{R} (\Delta^{(r)} - \bar{\Delta})^2}. \tag{39}$$

The null hypothesis was then tested against an alternative hypothesis H_1, that $\mathbf{T}_{TS}$ *is* superior to the reference tracker $\mathbf{T}_{REF}$, using the statistic $\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$. Usually, the alternative hypothesis is accepted at a significance level of α if $\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}} > \mu_\alpha$, where μ_α represents a point on the standard Gaussian distribution corresponding to the upper-tail probability of α. As is standard practice in hypothesis testing, we set the significance level to $\alpha = 0.05$.

The results of the hypothesis testing on position and prediction with respect to a different number of particles in the particle filter are shown in Table II and Table III. Table II shows the results for testing the hypothesis that $\mathbf{T}_{\mathrm{TS}}$ is superior to $\mathbf{T}_{\mathrm{RW}}$, while Table III shows the results for testing the hypothesis that $\mathbf{T}_{\mathrm{TS}}$ is superior to $\mathbf{T}_{\mathrm{NCV}}$. The second and third column in Table II and Table III show the test statistic $\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$. In all cases the test statistic is greater than $\mu_{0.05} = 1.645$. From Table II we can thus accept the hypothesis that $\mathbf{T}_{TS}$ is superior to $\mathbf{T}_{\mathrm{RW}}$ in estimating the position and the prediction at the $\alpha = 0.05$ level. Similarly, from Table III we can also accept the hypothesis that the tracker $\mathbf{T}_{TS}$ is superior to $\mathbf{T}_{\mathrm{NCV}}$ in estimating the position and the prediction at the $\alpha = 0.05$ level. Note that these hypotheses could have been accepted even at levels lower than $\alpha = 0.01$ ($\mu_{0.01} = 3.090$). Since the only difference between the $\mathbf{T}_{TS}$, $\mathbf{T}_{RW}$ and $\mathbf{T}_{NCV}$ was in the dynamic model of the player's position, we can conclude that the two-stage dynamic model is superior to both, the random-walk, as well as the nearly-constant-velocity model.

TABLE II
THE TEST STATISTIC $\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$ FOR THE ALTERNATIVE HYPOTHESIS THAT "$\mathbf{T}_{\mathrm{TS}}$ *is superior to* $\mathbf{T}_{\mathrm{RW}}$", CALCULATED FROM 30 RUNS. THE HYPOTHESIS MAY BE ACCEPTED AT SIGNIFICANCE LEVELS $\alpha = 0.05$ AND $\alpha = 0.01$.

no. particles	Position ($\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$)	Prediction ($\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$)
25	19.2	32.8
50	24.5	54.9
75	71.0	148.6
100	62.9	149.2

2) Qualitative evaluation: To further illustrate the performance of the trackers, the RMS errors (36) were averaged over all thirty repetitions for each tracker and are shown in Figure 4(a,b) and Figure 5(a,b). To visualize how the smoothness of the obtained trajectories changes with the number of particles, we have also calculated the mean-absolute-differences (MAD) on positions for different numbers of particles in the

TABLE III
THE TEST STATISTIC $\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$ FOR THE ALTERNATIVE HYPOTHESIS THAT "$\mathbf{T}_{\mathrm{TS}}$ *is superior to* $\mathbf{T}_{\mathrm{NCV}}$", CALCULATED FROM 30 RUNS. THE HYPOTHESIS MAY BE ACCEPTED AT SIGNIFICANCE LEVELS $\alpha = 0.05$ AND $\alpha = 0.01$.

no. particles	Position ($\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$)	Prediction ($\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$)
25	14.4	14.7
50	7.5	7.7
75	8.6	7.7
100	6.0	4.8

particle filter,

$$\mathrm{MAD} \triangleq \frac{1}{30}\sum_{r=1}^{30}\frac{1}{7}\sum_{i=1}^{7}\frac{1}{K}\sum_{k=1}^{K}|{}^{(i)}\bar{\mathbf{p}}_k - {}^{(i)}\hat{\mathbf{p}}_k^{(r)}|, \tag{40}$$

where ${}^{(i)}\bar{\mathbf{p}}_k = \frac{1}{30}\sum_{r=1}^{30}{}^{(i)}\hat{\mathbf{p}}_k^{(r)}$ was the position of the i-th player at k-th time-step, averaged over thirty tracking repetitions; the MADs are shown in Figure 4c and Figure 5c.

Figure 4 thus shows the results for the average RMS errors of position and prediction and MAD values of position when the number of particles used in the particle filter is varied. Using only 25 particles the proposed dynamic model in $\mathbf{T}_{\mathrm{TS}}$ achieved smaller RMS errors for position (Figure 4a) and prediction (Figure 4b) than the $\mathbf{T}_{\mathrm{RW}}$, even when four times as many particles were used in the $\mathbf{T}_{\mathrm{RW}}$. $\mathbf{T}_{\mathrm{TS}}$ also consistently produced smaller MAD values than $\mathbf{T}_{\mathrm{RW}}$ for all numbers of particles (Figure 4c).

In Figure 5, we can compare the average RMS errors and MAD values between $\mathbf{T}_{\mathrm{TS}}$ and $\mathbf{T}_{\mathrm{NCV}}$. Using only 25 particles, the $\mathbf{T}_{\mathrm{TS}}$ achieved equal average RMS errors for position (Fig. 5a) and prediction (Fig. 5b) as the $\mathbf{T}_{\mathrm{NCV}}$ with 100 particles. $\mathbf{T}_{\mathrm{TS}}$ also consistently produced smaller MAD values than $\mathbf{T}_{\mathrm{NCV}}$ for all numbers of particles (Figure 5c) and, again, using only 25 particles $\mathbf{T}_{\mathrm{TS}}$ achieved approximately equal MAD value as NCV at 100 particles. An important point to note here is that the $\mathbf{T}_{\mathrm{TS}}$ outperformed the $\mathbf{T}_{\mathrm{RW}}$ and $\mathbf{T}_{\mathrm{NCV}}$ even though the spectral densities for the $\mathbf{T}_{\mathrm{RW}}$ and $\mathbf{T}_{\mathrm{NCV}}$ were estimated from the test data. In fact, since $\hat{\mathbf{v}}_{k-1}$ was not taken into account in the derivation of the rule-of-thumb rule (20), the obtained spectral density for $\mathbf{T}_{\mathrm{TS}}$ was overestimated, and presents an upper bound on the actual density. Nevertheless, the two-stage model outperformed both, the RW and the NCV model. This implies powerful generalization capabilities of the proposed two-stage dynamic model.

B. Experiment 2: robustness of tracking

As mentioned in the introduction of this paper, a good dynamic model can not only provide a better accuracy of estimation but can also improve tracking in situations of high visual ambiguity. These situations arise, for example, when the target is moving close to another visually similar object, or when the target is occluded by such an object. Even though the tracking can be improved by applying a better visual model to reduce the visual ambiguity by itself, the dynamic model can *further improve* the tracking by preferring the target which corresponds to the model's dynamics, thus alleviating the effect of the visual ambiguity. To demonstrate the performance of the two-stage dynamic model in those situations, we have first applied it to tracking hands of a person (Figure 6). The person was facing the camera, waving his hands, and the hands occluded each other 17 times with majority of occlusions occurring in front of the persons face. Both hands were approximately 20×20 pixels large, and were tracked with the two-stage tracker from the previous experiment. All parameters of the tracker remained the same as in the previous experiment, except for the spectral density q_c. The spectral density was estimated using the rule-of-thumb rule from section III-A2 and assuming that the expected distance that the hand travels between two time-steps is approximately $\sigma_m = 5$ pixels. The number of particles in the particle filter was set to only $N = 25$ particles. We denote this tracker by $\mathbf{T}_{\mathrm{TS}}$. For reference, the hands were also tracked using a tracker which applied a nearly-constant-velocity (NCV)[5] model instead of the two-stage dynamic model and which used $N = 100$ particles in the particle filter; we denote this tracker by $\mathbf{T}_{\mathrm{NCV}}$.

The hands were tracked separately five times with $\mathbf{T}_{\mathrm{TS}}$ and $\mathbf{T}_{\mathrm{NCV}}$, and an average times that the tracker lost a hand was recorded. The results of tracking are shown in the second and third row of the Table IV. There we see that $\mathbf{T}_{\mathrm{NCV}}$ lost a hand on average 27 times, while the two-stage dynamic model in $\mathbf{T}_{\mathrm{TS}}$ reduced the number of failures approximately by 10 failures. All the failures occurred when the tracked hand was moving in front of a person's face, or was moving close to the other hand. An example of such situation is shown in Figure 6. In those situations, the visual ambiguity was highest, since the hands and the face were of similar color, which caused spurious modes in the visual likelihood function. Since the tracker which used the proposed two-stage dynamic model reduced the failure rate in comparison to a NCV model, this means that the two-stage dynamic model helped to reduce the visual ambiguities simply by better modelling the motion of the hand. However, there were still 15 hand overlaps, where the visual ambiguity was too high and could not be resolved merely by the motion model. We have therefore repeated the experiments, but instead of using a simple color-based visual model, we used the recently proposed local-motion [51], which uses optical flow to resolve the color ambiguities. The results are shown in the last two rows of the Table IV. We see that while the visual model by itself reduced the number of failures, its combination with the proposed two-stage dynamic model even further decreased the failure rate. Note, that not only did the two-stage dynamic model reduce the number of failures in comparison to the NCV model, but was able to do so requiring a quarter as many particles in the particle filter as the NCV model.

To demonstrate how the two-stage dynamic model performs when tracking an object which rapidly changes its motion, we have applied it to tracking a player of squash (Figure 7). Due to frequent occlusions between the players, we have used the local-motion visual model [51] in this experiment. The player

[5]The NCV model was used in preference to the RW model, since the hand motion was closer to a nearly-constant-velocity motion than the random-walk motion.

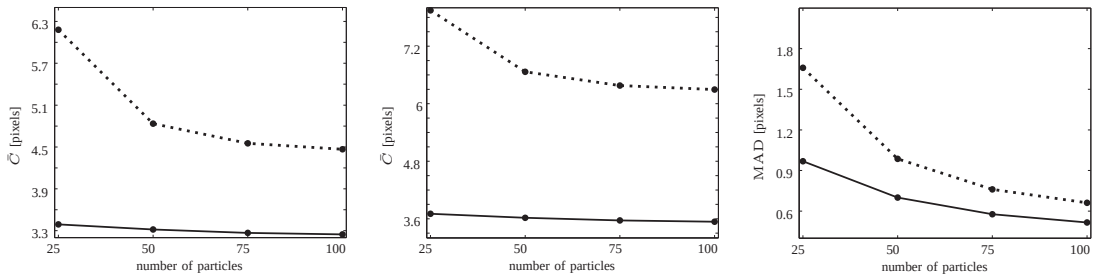


Fig. 4. Graphs on (a) and (b) show the average RMS errors (denoted by $\bar{C}$) of position (a) and prediction (b), respectively, as a function of the number of particles. Graphs in (c) show the mean-absolute-differences (denoted by MAD) values of position estimates. The results for $\mathbf{T}_{\mathrm{RW}}$ are depicted by the *dotted* lines, while solid lines depict the results for $\mathbf{T}_{\mathrm{TS}}$.

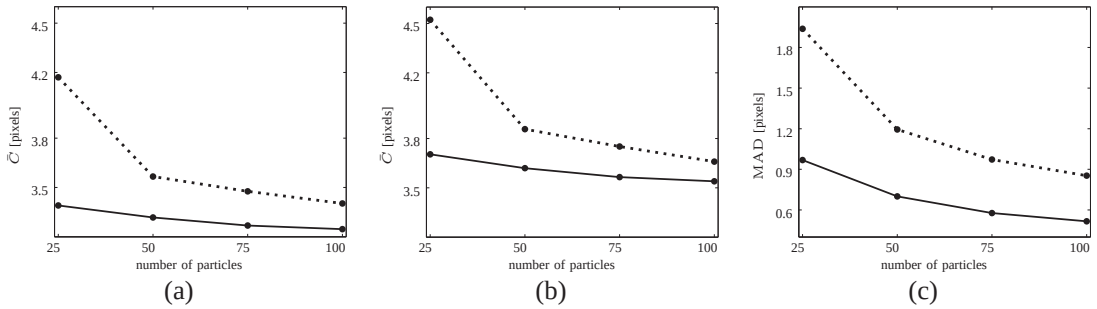


Fig. 5. Graphs on (a) and (b) show the average RMS errors (denoted by $\bar{C}$) of position (a) and prediction (b), respectively, as a function of the number of particles. Graphs in (c) show the mean-absolute-differences (denoted by MAD) values of position estimates. The results for $\mathbf{T}_{\mathrm{NCV}}$ are depicted by the *dotted* lines, while solid lines depict the results for $\mathbf{T}_{\mathrm{TS}}$.

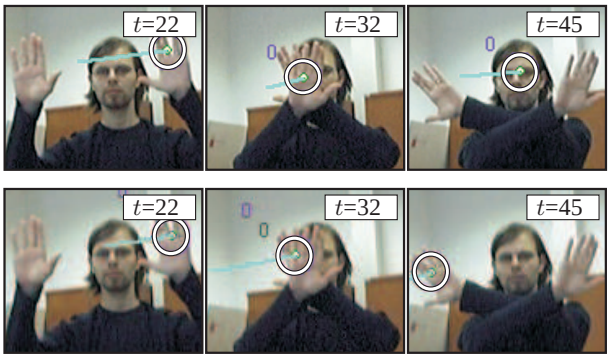


Fig. 6. Frames from the experiments with hand tracking using a NCV dynamic model with 100 particles (upper row) and using the two-stage dynamic model with only 25 particles (lower row). The white ellipse depicts the tracked region.

TABLE IV
RESULTS OF TRACKING HANDS USING THE NCV AND THE TWO-STAGE DYNAMIC MODEL.

tracker	visual model	dynamic model	number of particles	number of failures
$\mathbf{T}_{\mathrm{TS}}$	color-based	two-stage	25	15
$\mathbf{T}_{\mathrm{NCV}}$	color-based	NCV	100	27
$\mathbf{T}_{\mathrm{TS}}$	combined	two-stage	25	2
$\mathbf{T}_{\mathrm{NCV}}$	combined	NCV	100	4

was approximately 25×45 pixels large and was occluded 14 times by another visually similar player. The sequence was especially difficult to track due to frequent occlusions and rapid changes in the player's motion. All parameters of the tracker $\mathbf{T}_{\mathrm{TS}}$ remained the same as in the previous experiment. The spectral density q_c was again estimated using the rule-of-thumb rule from section III-A2 and assuming that the expected distance that the player travels between two time-steps is approximately $\sigma_m = 5$ pixels. For reference, the player was also tracked using the recently proposed state-of-the-art tracker [51] which applied a nearly-constant-velocity (NCV) and the local-motion visual model. We denote this tracker by $\mathbf{T}_{\mathrm{NCV}}$. The player was tracked five times with each tracker and the number of times the tracker failed was recorded. All the failures occurred when the player was occluded by the other visually similar player.

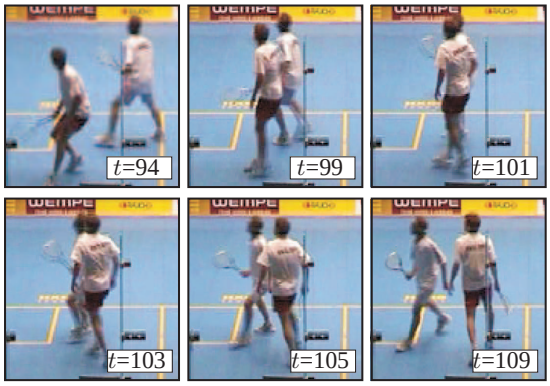


Fig. 7. An example from tracking the squash player in which the player gets occluded.

Table V shows the average number of times each tracker failed with respect to the number of particles used in the

particle filter. Using 25 particles, the NCV model failed on average four times, while the two-stage model failed only three times. When the number of particles was increased to 50, the NCV model improved in performance by reducing the failure rate to three failures, while the two-stage model reduced the failure rate to two failures. However, when the number of particles was increased to 100, the failure rate of the NCV model remained at three failures, while the two-stage model further reduced the failure rate to only a single failure. The two-stage dynamic model consistently outperformed the NCV model by producing a smaller failure rate for all the selected number of particles. Note also that, using only 25 particles, the two-stage dynamic model achieved an equal failure rate as the NCV model with 100 particles. As the number of the particles was increased, the two-stage model further decreased the failure rate. Note that the improvements in tracking come from two sources. One source is that the TS model reduces the ambiguity by better modelling the target's motion despite the rapid changes in motion. The second source is that the local-motion visual model [51] relies on updating its model by using the target velocity estimated from the tracker. Since the TS produces better estimates of the velocity than NCV, the result is an improved visual model and improved tracking.

TABLE V
THE AVERAGE NUMBER OF TIMES THE TRACKER FAILED TO CORRECTLY TRACK THE PLAYER OF SQUASH WHEN USING A NCV ($\mathbf{T}_{\mathrm{NCV}}$) AND THE TWO-STAGE DYNAMIC MODEL ($\mathbf{T}_{\mathrm{TS}}$).

tracker	dynamic model	number of particles	number of failures	execution times [ms]
$\mathbf{T}_{\mathrm{TS}}$	two-stage	25	3	21
$\mathbf{T}_{\mathrm{TS}}$	two-stage	50	2	27
$\mathbf{T}_{\mathrm{TS}}$	two-stage	100	1	37
$\mathbf{T}_{\mathrm{NCV}}$	NCV	25	4	19
$\mathbf{T}_{\mathrm{NCV}}$	NCV	50	3	26
$\mathbf{T}_{\mathrm{NCV}}$	NCV	100	3	35

As noted in the introduction, a very important aspect of every tracker is its processing speed. In particular, the particle filters are Monte Carlo methods, which rely on estimating distributions using simulations of particles and evaluations of the likelihood function. While simulation from the dynamic model is a fast operation, evaluating the likelihood function for each particle presents a bottle-neck in the processing speed – the processing time of each iteration increases with increasing the number of particles. We have therefore recorded average execution times per time-step for the experiment in Table V. Given the same number of particles in the filter, the processing times of $\mathbf{T}_{\mathrm{TS}}$ are practically equal to those of $\mathbf{T}_{\mathrm{NCV}}$. Note, however, that $\mathbf{T}_{\mathrm{TS}}$ required only 25 particles to achieve performance of $\mathbf{T}_{\mathrm{NCV}}$ at 100 particles. This means that $\mathbf{T}_{\mathrm{TS}}$ achieved an equal performance to $\mathbf{T}_{\mathrm{NCV}}$ but with 40% reduction in the processing time.

From the results in Table IV and Table V we see that the two-stage dynamic model can improve tracking by reducing the number of failures by reducing the visual ambiguity, while at the same time requiring only a small number of particles in the particle filter, which effectively reduces the processing time. We can also conclude that the two-stage dynamic model is general enough to improve tracking not only when tracking rapidly moving persons but also parts of persons, such as hands. For further examples of tracking with the two-stage dynamic model please see the online videos at http://vicos.fri.uni-lj.si/data/matejk/tracking/DynamicModel/Sub/index.html.

C. Sensitivity to parameters

There are two main parameters in the two-stage dynamic model. The first parameter is the parameter β in the liberal model (9). We have studied this parameter in some detail in section III-A and chose its value $\beta = 2$. Note that this value has been fixed for *all* our experiments reported here, which also practically justifies the value we have chosen in the section III-A. The other very important parameter is the spectral density of the liberal model. This parameter, however, very much depends on the given application and the setup and is in standard dynamic models related directly or indirectly to the variance of the state estimates. In particular, in a standard particle-filter-based tracker, using a large spectral density necessarily requires increasing the number of particles to maintain a low variance of the final estimate (e.g., accuracy of position). To make setting the spectral density in the two-stage dynamic model a more intuitive matter, we have related it to the distance a target is expected to travel in consequent time-steps, σ_m, and derived the corresponding rule-of-thumb rule in the section III-A2.

To gain a further insight of how different values of the parameter σ_m affect the performance of the two-stage dynamic model, we have revisited the experiment from section IV-A. In that experiment the spectral density was set using the rule-of-thumb rule with the average-distance parameter σ_m estimated from the sports literature. To see how the results vary with this parameter the experiment was repeated for the $\mathbf{T}_{\mathrm{TS}}$ with 25 particles in the particle filter. The parameter σ_m was decreased by some factor α to a point where the tracker started to fail and then increased to a point where it started to fail.

When lowering the α, the tracker started to fail at $\alpha = 0.5$ and then, when this parameter was increased, the tracker started to fail again at $\alpha = 1.4$. The results for the values $\alpha = \{0.5, 0.7, 1.0, 1.2, 1.4\}$ are shown in Figures 8. That figure shows that the optimum is reached at $\alpha = 0.9$ which means that the optimum parameter σ_m is 90% of that estimated by the rule-of-thumb rule. We also see that parameter values around the value estimated by the rule-of-thumb rule do not significantly deteriorate the tracker's performance. This means that despite of increased variance of the noise in the liberal dynamic model, the variance of the tracker's estimate (e.g., position and prediction) remains low when using the two-stage dynamic model.

To further demonstrate the accuracy of tracking with an overestimated noise and with low number of particles in the particle filter, we have have considered an example of tracking in cluttered environment as shown in Figure 9. The tracked person was performing rapid movements and was occluded many times by other persons. The person was first tracked with a color-based tracker that used the two-stage dynamic model ($\mathbf{T}_{\mathrm{TS}}$) with 25 particles in the particle filter. The noise parameter σ_m was estimated as in (35) and was

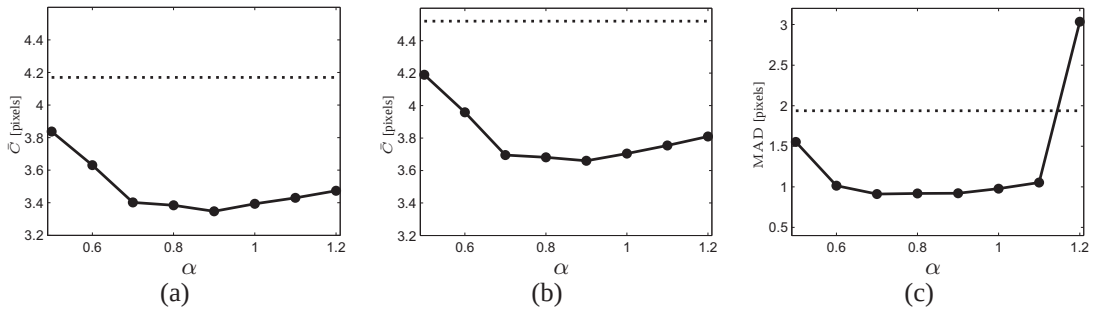


Fig. 8. Graphs on (a) and (b) show the average RMS errors (denoted by $\bar{C}$) of position (a) and prediction (b), respectively, as a function of parameter α. Graphs in (c) show the mean-absolute-differences (denoted by MAD) values of position estimates. The solid lines depict the results for $\mathbf{T}_{\mathrm{TS}}$ while dotted line depicts performance of $\mathbf{T}_{\mathrm{NCV}}$.

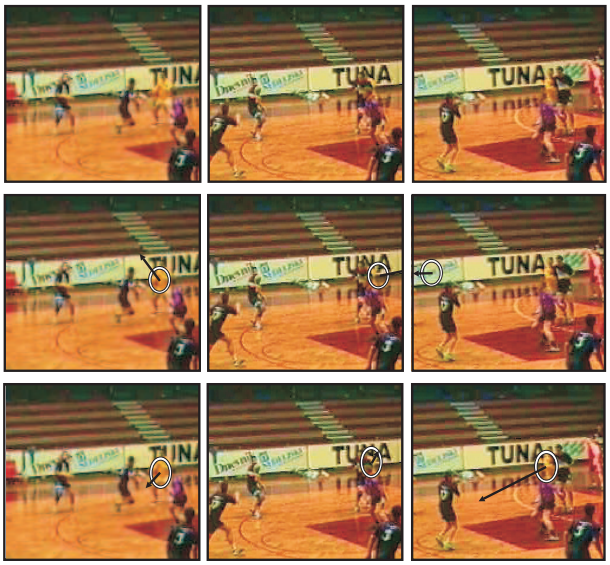


Fig. 9. An example of tracking with highly overestimated noise in the dynamic model. The upper row shows the tracked (yellow) player in sequence of images with indexes 202, 212 and 225. The middle row shows an example in which the $\mathbf{T}_{\mathrm{NCV}}$ fails due to a large variance of the estimate during the occlusion. The lower row shows the same sequence in which the the $\mathbf{T}_{\mathrm{TS}}$ does not fail. The ellipses depict the tracked region and the arrow shows the estimated velocity.

multiplied by a factor $\alpha = 2$ to grossly overestimate it. The same person was then tracked using a NCV dynamic model ($\mathbf{T}_{\mathrm{NCV}}$) with comparably overestimated noise. We have observed that the variance of the estimates (position as well as prediction) provided by the $\mathbf{T}_{\mathrm{NCV}}$ was significantly higher than those of $\mathbf{T}_{\mathrm{TS}}$ tracker. Despite the overestimated noise, the $\mathbf{T}_{\mathrm{TS}}$ tracker was able to track the person throughout the sequence, whereas the $\mathbf{T}_{\mathrm{NCV}}$ failed when the person was completely occluded by other persons. This is shown in Figure 9 and for the full video demonstrating this performance, see the paper's homepage at http://vicos.fri.uni-lj.si/data/matejk/tracking/DynamicModel/Sub/index.html.

V. Conclusion

We have proposed a two-stage dynamic model, and a corresponding two-stage probabilistic tracker, that can account for various types of motions which we usually encounter when tracking persons. The proposed model is composed from two separate dynamic models. The first dynamic model is called the liberal dynamic model which was derived in Section III-A from a non-zero-mean Gauss-Markov process. An analysis of the parameters of the liberal model in Section III-A1 has shown that two widely-used models, the random-walk (RW) model and the nearly-constant-velocity (NCV) model, are obtained at the limiting values of the model's parameters. We have also noted that the liberal model can explain even motions which are in between the RW and the NCV model. An important parameter of the liberal model is the spectral density of the Gauss-Markov process, which depends on the dynamics of the class of objects to be tracked. In Section III-A2 we have therefore derived a rule-of-thumb rule to selecting this density, which requires only a vague estimate of the target dynamics. Furthermore, by controlling the mean value of the Gauss-Markov process, the liberal model can even further adjust to the dynamics of the tracked target. To efficiently estimate this mean value in the liberal model, another dynamic model, which we call the conservative model, was proposed in Section III-B. In contrast to the liberal model which allows greater perturbations in target's motion, the conservative model assumes stronger constraints on the target velocity. In Section III-C we have proposed a two-stage probabilistic tracker which uses the liberal dynamic model within a particle filter to efficiently explore the state space of the tracked target. On the other hand, the conservative model is used to estimate the mean value of the Gauss-Markov process in the liberal model as well as for regularizing the estimations from the particle filter.

Two sets of experiments were designed to evaluate the performance of the proposed two-stage dynamic model. The first set of experiments involved tracking persons running on the path which was drawn on the floor. The path was designed such that the observed motion included accelerations, decelerations, short runs in a certain direction and sudden changes in the direction of motion. All persons were tracked with the proposed dynamic model as well as with two reference trackers which employed one of the two widely-used dynamic models – the RW model and the NCV model. The results have shown that the proposed dynamic model performed significantly better than the RW as well as NCV model. In particular, the two-stage dynamic model yielded a

better accuracy of tracking in comparison to the RW and NCV models, and at the same time required significantly smaller number of particles in the particle filter. In the second set of experiments we have tracked person's hands and a player in squash using the proposed dynamic model and a NCV model. These experiments were designed to demonstrate the performance of the two-stage dynamic model when the target is moving in a close proximity of a visually similar object. In the experiment of tracking a person's hands, the proposed dynamic model was able to use half as many particles in the particle filter as the NCV model while still reducing the number of times that tracking failed in comparison to the NCV model. This shows the ability of the two-stage dynamic model to reduce the visual ambiguity in the target's position by better modelling the target's dynamics. To demonstrate the performance of the two-stage dynamic model when tracking a person who rapidly changes its motion, we have applied it to tracking a squash player. The results again showed that the two-stage model allows smaller number of particles in the particle filter to achieve a comparable of better performance than the NCV model achieves with a large number of particles. The results of the two sets of the experiments imply a superiority of the two-stage model over the RW and NCV in accounting for various dynamics of moving persons as well as parts of persons such as hands.

We have seen in the experiment of tracking hands and a squash player that the two-stage dynamic model can help to resolve some of the visual ambiguities which occur when the target is moving close to another visually similar object. However, there were situations in which the dynamic model could not resolve the ambiguity by it self. Since the dynamic model was implemented within a standard particle filter, the visual model which was used in the experiment can be easily replaced or augmented by more powerful existing visual models, e.g., [52], [53], [54], [55], which may better handle some of the visual ambiguities. The performance of the proposed two-stage dynamic model strongly depends on the selected noise parameter of the dynamic system (the spectral density). Improper values of this parameter might lead to failed tracking in certain situations. If the spectral density in the liberal model is set too low, then the dynamic model will not be able to account for the abrupt motions and will act as having a great inertial properties. Consider an example in which the target is quickly moving to the left for a while and then abruptly changes its direction and starts moving to the right. A very low spectral density will likely result in tracker not being able to keep up with the target even from the start and it will lose the target. By slightly increasing the spectral density (but still keeping it low), the tracker will exhibit strong inertial properties and initially keep up with the target, but then, when the target changes its motion, it will continue approximately in the direction in which it was initially moving, and again lose the target. Note that these are pathological situations. Indeed, we have observed that the two-stage model is robust to variations of the spectral density around the one which is calculated by the rule-of-thumb. Therefore, if a designer of a tracking algorithm wishes to fine tune the spectral density for a given application, a good starting point is the equation (35). Another pathological case is when the spectral density is severely overestimated and the particles in the particle filter spread far beyond the target. Having a small number of particles this inevitably increases the variance of the estimator and the motion model becomes weaker. As a result, the target's motion is poorly modelled and if a visually-similar object is somewhere in the target's surrounding, chances are that the tracker will fail to keep a lock on the correct target.

A convenient property of the two-stage dynamic model is that, since it typically requires a smaller number of particles in the particle filter, it allows faster tacking with more complex visual models in comparison to other dynamic models which require more particles. Note also that the implementation of the two-stage model allows adopting existing solutions for improved particle filtering, like the ones mentioned in the introduction [10], [11], [12], [13], [32]. These can be used to even further improve the tracker's performance, both in terms of improved estimation accuracy as well as in reduction of the failure rate. These topics are the focus of ongoing research.

ACKNOWLEDGMENT

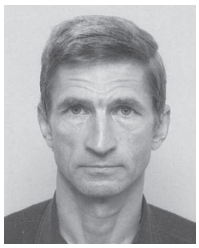
This research has been supported in part by: Research program P2-0214 (RS), research program P2-0095 (RS), M3-0233 project PDR sponsored by the ministry of defense of republic of Slovenia, and EU FP7-ICT215181-IP project CogX. We would also like to thank the editor and the anonymous reviewers for their constructive comments, which helped us to improve our paper.

REFERENCES

[1] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Comp. Vis. Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999. 1
[2] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comp. Vis. Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999. 1
[3] P. Gabriel, J. Verly, J. Piater, and A. Genon, "The state of the art in multiple object tracking under occlusion in video sequences," in *Proc. Advanced Concepts for Intelligent Vision Systems*, 2003, p. 166173. 1
[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Systems, Man and Cybernetics, C*, vol. 34, no. 30, pp. 334–352, 2004. 1
[5] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comp. Vis. Image Understanding*, vol. 81, no. 3, pp. 231–268, March 2001. 1
[6] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Comp. Vis. Image Understanding*, vol. 103, no. 2-3, pp. 90–126, November 2006. 1
[7] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer Science+Business Media, LCC, 2006. 1, 8
[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Engineering*, vol. 82, pp. 34–45, 1960. 1, 7
[9] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, January 2001. 1
[10] M. K. Pitt and N. Sheppard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999. 2, 14
[11] P. Torma and C. Szepesvari, "On using likelihood-adjusted proposals in particle filtering: Local importance sampling," in *Proc. Int. Symp. Image and Signal Processing and Analysis*, September 2005. 2, 14
[12] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. European Conf. Computer Vision*, vol. 1, 2004, pp. 28–39. 2, 14
[13] A. Naeem, T. Pridmore, and S. Mills, "Managing particle spread via hybrid particle filter/kernel mean shift tracking," in *Proc. British Machine Vision Conference*, 2007. 2, 14

[14] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *Proc. European Conf. Computer Vision*, 2002, pp. 784–800. 2
[15] A. Agarwal and B. Triggs, "Tracking articulated motion with piecewise learned dynamical models," in *Proc. European Conf. Computer Vision*, vol. 3, 2004, pp. 54–65. 2
[16] R. Urtasun, D. Fleet, and P. Fua, "3d people tracking with gaussian process dynamic models," in *Proc. Conf. Comp. Vis. Pattern Recognition*, vol. 1, 2006, pp. 238–245. 2
[17] B. Li, Q. Meng, and H. Holstein, "Articulated pose identification with sparse point features," *IEEE Trans. Systems, Man and Cybernetics, B*, vol. 34, no. 3, pp. 1412–1422, 2004. 2
[18] X. Rong Li and V. Jilkov P., "Survey of maneuvering target tracking: Dynamic models," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1363, October 2003. 2
[19] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001, ch. 11, pp. 438–440. 2, 4, 9
[20] W. R. Li and Y. Bar-Shalom, "Performance prediction of the interacting multiple model algorithm," *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 755–771, 1993. 2
[21] Y. Bar-Shalom, Ed., *Multitarget/Multisensor Tracking: Applications and Advances*. YBS Publishing, 1998, vol. 2. 2
[22] K. J. Bradshaw, I. D. Reid, and D. W. Murray, "The active recovery of 3d motion trajectories and their use in prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 219–234, 1997. 2
[23] S. McGinnity and G. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 1006–1012, 2000. 2
[24] H. A. P. Blom and E. A. Bloem, "Exact bayesian and particle filtering of stochastic hybrid systems," *IEEE Trans. Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 55–70, 2006. 2
[25] J. Xue, N. Zheng, J. Geng, and X. Zhong, "Tracking multiple visual targets via particle-based belief propagation," *IEEE Trans. Systems, Man and Cybernetics, B*, vol. 38, no. 1, pp. 196–209, 2008. 2
[26] P. Pérez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proc. of the IEEE*, vol. 92, no. 3, pp. 495–513, 2004. 2, 7
[27] C. J. Needham, "Tracking and modelling of team game interactions," Ph.D. dissertation, School of Computing, The University of Leeds, October 2003. 2
[28] A. Senior, "Tracking people with probabilistic appearance models," in *Perf. Eval. Track. and Surveillance in conjunction with ECCV02*, 2002, pp. 48–55. 2
[29] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Patt. Recogn.*, vol. 40, no. 7, pp. 1958–1970, 2007. 2
[30] F. Pernkopf, "Tracking of multiple targets using online learning for reference model adaptation," *IEEE Trans. Systems, Man and Cybernetics, B*, vol. 38, no. 6, pp. 1465–1475, 2008. 2, 7
[31] F. Talantzis, A. Pnevmatikakis, and A. G. Constantinides, "Audiovisual active speaker tracking in cluttered indoors environments," *IEEE Trans. Systems, Man and Cybernetics, B*, vol. 38, no. 3, pp. 799–807, 2008. 2
[32] J. Wang and Y. Yagi, "Adaptive mean-shift tracking with auxiliary particles," *IEEE Trans. Systems, Man and Cybernetics, B*, vol. 39, no. 6, pp. 1578–1589, 2009. 2, 14
[33] N. H. H. Bellotto, "Multisensor-based human detection and tracking for mobile service robots," *IEEE Trans. Systems, Man and Cybernetics, B*, vol. 39, no. 1, pp. 167–181, 2009. 2
[34] M. Perse, J. Pers, M. Kristan, G. Vuckovic, and S. Kovacic, "Physics-based modelling of human motion using kalman filter and collision avoidance algorithm," in *International Symposium on Image and Signal Processing and Analysis*, September 2005, pp. 328–333. 2
[35] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Trans. Signal Proc.*, vol. 50, no. 2, pp. 174–188, February 2002. 3
[36] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," in *IEE Proc. Radar and Signal Processing*, vol. 40, no. 2, 1993, pp. 107–113. 3
[37] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998. 3
[38] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-6, no. 4, pp. 473–483, 1970. 3
[39] R. A. Singer and K. W. Benhke, "Real-time tracking filter evaluation and selection for tactical applications," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-7, no. 1, pp. 100 – 110, 1971. 3
[40] H. Zhou and K. S. P. Kumar, "A current statistical model and adaptive algorithm for estimating maneuvering targets." *AIAA J. of Guidance*, vol. 7, no. 5, p. 596602, 1984. 3
[41] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997. 3
[42] M. Kristan, "Tracking people in video data using probabilistic models," Ph.D. dissertation, Faculty of Electrical Engineering, University of Ljubljana, 2008. 4, 8
[43] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. European Conf. Computer Vision*, vol. 1, 2002, pp. 661–675. 7
[44] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Color features for tracking non-rigid objects," *Chinese J. Automation*, vol. 29, no. 3, pp. 345–355, May 2003. 7
[45] M. Kristan, J. Perš, M. Perše, M. Bon, and S. Kovačič, "Multiple interacting targets tracking with application to team sports," in *International Symposium on Image and Signal Processing and Analysis*, September 2005, pp. 322–327. 7
[46] M. Kristan, J. Perš, M. Perše, and S. Kovačič, "Closed-world tracking of multiple interacting targets for indoor-sports applications," *Comput. Vision Image Understanding*, vol. 113, no. 5, pp. 598–611, May 2009. 7
[47] M. Bon, J. Perš, M. Šibila, and S. Kovačič, *Analiza gibanja igralca med tekmo*, V. Vuleta and A. Leonardis, Eds. Faculty of Sport, University of Ljubljana, 2001. 9
[48] C. Kotzamanidis, K. Chatyikoloutas, and A. Gianakos, "Optimization of the training plan of the handball game," *Handball: periodical for coaches, referees and lecturers*, vol. 2, pp. 65–71, 1999. 9
[49] W. S. Erdmann, "Gathering of kinematic data of sport event by televising the whole pitch and track," in *Proc. Int. Soc. Biomech. Sports Sympossium*, 1992, pp. 159–162. 9
[50] J. Bangsbo, "The physiology of soccer: With special reference to intense intermittent exercise," *Acta Physiologica Scandinavica*, vol. 619, pp. 1–155, 1994. 9
[51] M. Kristan, J. Perš, S. Kovačič, and A. Leonardis, "A local-motion-based probabilistic model for visual tracking," *Pattern Recognition*, vol. 42, no. 9, pp. 2160–2168, 2009. 10, 11, 12
[52] W. Du and J. Piater, "A probabilistic approach to integrating multiple cues in visual tracking," in *10th European Conference on Computer Vision*, 2008. 14
[53] W. L. Lu and J. J. Little, "Tracking and recognizing actions at a distance," in *Proc. Workshop on Compter Vision Based Analisys in Sport Environments In conjunction with ECCV06*, May 2006, pp. 49–60. 14
[54] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1372– 1384, 2006. 14
[55] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequences," *Image and Vision Computing*, vol. 25, no. 8, pp. 1217–1227, August 2007. 14

Matej Kristan received a BSc, MSc and PhD degree in 2003, 2005 and 2008, respectively, in Electrical engineering at the Faculty of Electrical Engineering, University of Ljubljana. He is currently a researcher at the Visual Cognitive Systems Laboratory at the Faculty of Computer and Information Science, University of Ljubljana, and a researcher at the Machine Vision Laboratory at the Faculty of Electrical Engineering, University of Ljubljana. His research interests include probabilistic methods for computer vision and pattern recognition with focus on tracking, probabilistic dynamic models and online learning. He has received several awards for his research in the field of computer vision and pattern recognition.



Stanislav Kovačič is a full professor and the head of the Laboratory for Machine Vision at the Faculty of Electrical Engineering, University of Ljubljana. From 1985 to 1988 he was a visiting researcher in the General Robotics and Active Sensory perception Laboratory at the University of Pennsylvania. He was also a visiting researcher and a visiting professor at the Technische Fakultät der Friedrich-Alexander-Universität in Erlangen, and at the Faculty of Electrical Engineering and Computing, University of Zagreb. His research is focused on various aspects of image and video analysis, image registration, with applications in medicine, industry and sports. He has authored or coauthored more than 120 publications in journals, conferences, and book chapters.

Aleš Leonardis is a full professor and the head of the Visual Cognitive Systems Laboratory with the Faculty of Computer and Information Science, University of Ljubljana. He is also an adjunct professor at the Faculty of Computer Science, Graz University of Technology. From 1988 to 1991, he was a visiting researcher in the General Robotics and Active Sensory Perception Laboratory at the University of Pennsylvania. From 1995 to 1997, he was a postdoctoral associate at the PRIP, Vienna University of Technology. He was also a visiting researcher and a visiting professor at the Swiss Federal Institute of Technology ETH in Zurich and at the Technische Fakultaet der Friedrich-Alexander-Universitaet in Erlangen, respectively. His research interests include robust and adaptive methods for computer vision, object and scene recognition and categorization, statistical visual learning, 3D object modeling, and biologically motivated vision. He is an author or coauthor of more than 160 papers published in journals and conferences and he coauthored the book Segmentation and Recovery of Superquadrics (Kluwer, 2000). He is an Editorial Board Member of Pattern Recognition, an Editor of the Springer Book Series Computational Imaging and Vision, and an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence. He has served on the program committees of major computer vision and pattern recognition conferences. He was also a program co-chair of the European Conference on Computer Vision, ECCV 2006. He has received several awards. In 2002, he coauthored a paper, Multiple Eigenspaces, which won the 29th Annual Pattern Recognition Society award. In 2004, he was awarded a prestigious national Award for scientific achievements. He is a fellow of the IAPR and a member of the IEEE and the IEEE Computer Society.

Janez Perš received a BSc, MSc and PhD degrees in Electrical engineering at the Faculty of Electrical Engineering, University of Ljubljana, in 1998, 2001 and 2004, respectively. He is currently assistant at the Machine Vision Laboratory at the Faculty of Electrical Engineering, University of Ljubljana. His research interests lie in image sequence processing, object tracking, human motion analysis, dynamic motion based biometry, and in autonomous and distributed systems.