Computer Vision and Image Understanding 117 (2013) 922–933

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Particle filters and occlusion handling for rigid 2D–3D pose tracking ☆

Jehoon Lee [a,*], Romeil Sandhu [b], Allen Tannenbaum [a]

[a] Department of Electrical and Computer Engineering, UAB, Birmingham, AL 35294, USA
[b] Harper Laboratories, LLC, Atlanta, GA 30318, USA

ARTICLE INFO

Article history:
Received 7 August 2012
Accepted 17 April 2013
Available online 28 April 2013

Keywords:
2D–3D pose estimation
Object tracking
Occlusion handling
Particle filters

ABSTRACT

In this paper, we address the problem of 2D–3D pose estimation. Specifically, we propose an approach to jointly track a rigid object in a 2D image sequence and to estimate its pose (position and orientation) in 3D space. We revisit a joint 2D segmentation/3D pose estimation technique, and then extend the framework by incorporating a particle filter to robustly track the object in a challenging environment, and by developing an occlusion detection and handling scheme to continuously track the object in the presence of occlusions. In particular, we focus on partial occlusions that prevent the tracker from extracting an exact region properties of the object, which plays a pivotal role for region-based tracking methods in maintaining the track. To this end, a dynamical choice of how to invoke the objective functional is performed online based on the degree of dependencies between predictions and measurements of the system in accordance with the degree of occlusion and the variation of the object's pose. This scheme provides the robustness to deal with occlusions of an obstacle with different statistical properties from that of the object of interest. Experimental results demonstrate the practical applicability and robustness of the proposed method in several challenging scenarios.

1. Introduction

Visual tracking has been a significant topic of research in the field of computer vision; see [1–4] and references therein. The ultimate goal of visual tracking is to continuously identify the 3D location of an object of interest from an image sequence. This amounts to what is known as the 2D–3D pose tracking problem [5,6]. However, due to the difficulty of developing a tractable solution for estimating the 3D position from a 2D scene, many researchers have tacitly restricted the tracking problem to be concerned with only the relative 2D location of the object in which segmentation is often employed in conjunction with Kalman or particle filters [7,8]. Recent techniques attempt to revisit the 2D–3D pose tracking problem for challenging scenarios by leveraging on 2D image segmentation to estimate the 3D location [9–12]. While impressive results have been obtained that rival both pose tracking and segmentation based algorithms, these schemes did not fully exploit the underlying system dynamics that is inherent in any visual tracking task. Thus, in order to effectively treat the temporal nature of the observed 2D scene, we propose to extend a similar framework proposed by [13] in which we now incorporate a particle filter to perform 3D pose estimation of a rigid object. However, before doing so, let us revisit several contributions related to the proposed method.

Several algorithms have been introduced to solve the 2D–3D pose tracking task. In general, they are based on local or region attributes for feature matching. For example, such features include points [14], lines [15], polyhedral shape [16], complete contours [17,18], or surfaces [12]. Specifically, in [16], the authors perform a 2D global affine transformation as an initialization to 3D pose estimation, and then the 3D object pose is computed by an energy minimization process with respect to an approximate polyhedral model of the object. The authors in [18] present a 3D pose estimation algorithm by using visible edges. That is, they use binary space partition trees for finding and determining the visible line to track the edges of the model. However, since these methods rely on local features, the resulting solutions may yield unsatisfactory results in the presence of noise or cluttered environments. To overcome this, an early attempt to couple segmentation and pose estimation is given in [10]. In their work, the authors propose a region-based active contour that employs a unique shape prior, which is represented by a generalized cone based on a single reference view of an object. More recently, authors in [13] as well as Schmaltz et al. [12] propose a region-based model scheme for 2D–3D pose tracking by projecting a 3D object onto a 2D image plane such that the optimal 3D pose transformation coincides with the correct 2D segmentation. In a similar fashion, Kohli et al. [9] proposed a joint segmentation and pose estimation scheme using the graph cut methodology. Although these methods perform exceptionally well

☆ This paper has been recommended for acceptance by Y. Aloimonos.

* Corresponding author.
E-mail addresses: jehoon.lee25@gmail.com (J. Lee), romeil.sandhu@harperlabs.com (R. Sandhu), tannenba@uab.edu (A. Tannenbaum).

1077-3142/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2013.04.002

for many cases, they do not exploit the underlying dynamics inherent in a typical visual tracking task. We should note, in the context of the proposed work, the incorporation of system dynamics can be viewed as an extension to these baseline algorithms.

In addition to the aforementioned approaches for 2D–3D pose tracking, many works may be found in the literature which purely focus on restricting the visual tracking problem to the 2D domain. Because a complete overview of existing methods is beyond the scope of this paper, we just consider those methods that employ various filtering schemes such as the Kalman filter [19], unscented Kalman filter [7,20], and particle filter [8,21] as well as explicit algorithms for occlusion handling [22,23]. Specifically, the authors in [24,25] employ a finite dimensional parameterization of curves, namely B-splines, in conjunction with the unscented Kalman filter for rigid object tracking. Generalizing the Kalman filter approach, the work in [26] presents an object tracking algorithm based on particle filtering with quasi-random sampling. Since these approaches only track the finite dimensional group parameters, they cannot handle local deformations of the object. As a result, several tracking schemes have been developed to account for deformation of the object via the level set technique [27,28]. In relation to our work, some early attempts for 2D visual tracking using level set methods can be found in [29,30]. In particular, authors in [30] propose a definition of motion for a deformable object. This is done by decoupling an object's motion into a finite group motion known as *deformotion* with that of deformation, which is any departure from rigidity. Building on this, authors in [31] introduce a deformable tracking algorithm that utilizes the particle filtering framework in conjunction with geometric active contours. Other approaches closely related to these frameworks are given in [19,32,33]. Here the authors use a Kalman filter for predicting possible movements of the object, while the active contours are employed only for tracking deformations of the corresponding object.

In addition to employing filtering schemes to increase the robustness of tracking, many algorithms invoke a systematic approach to handle occlusions. We should note that although the main contribution of our work focuses on employing particle filtering to estimate the 3D pose during 2D visual tracking, a neat feature of the resulting methodology is its ability to handle occlusions effectively. Thus, we briefly revisit several attempts to specifically handle occlusions in the context of visual tracking [22,34,35]. Such occlusions can occur when another object lies between the target and a camera, or the target occludes parts of itself. In general, in the case of contour tracking, most methods incorporate shape information of an object of interest into a tracking framework online [35] or offline [23] to make up for poor distinguishable statistics between the object and background or missing parts of the object. To this end, a shape prior can be obtained or learnt from linear principal component analysis (PCA) if the assumption of small variations in shape holds [36]. Otherwise, for highly deformable objects, locally linear embedding (LLE) [37] or non-linear PCA [22] may be employed. In the case of template tracking, in which a complete region occupied by the object is not tracked, recently, the L1 tracker [38] is modified to accurately represent occlusions using the trivial templates in [39]. In [39], the energy of the trivial coefficients is adaptively controlled to capture the effect of occlusions.

Contribution: The algorithms proposed in this paper are closely related to the works in [13] and [40]. In the work of [13], the authors derive a variational approach to jointly carry out tasks of 2D region based segmentation and 3D pose estimation. This method shows robust performance for segmenting a 2D image and estimating the 3D pose of an object over image sequences even in cluttered environments. However, since this method ignores the temporal nature of the observed images, it cannot handle erratic movements or challenging occlusions. That is, the variational technique relies only on image information to drive the corresponding 3D pose estimate, which may cause unsatisfactory results in the presence of occlusions that are statistically different from that of the object of interest; see the experiments in Section 4.2. In the work of [40], the set of 3D transformation matrices is randomly constructed to find the optimal pose of an object of interest by Monte Carlo sampling on the special Euclidean group, and then a region based statistical energy is applied to evaluate the optimality of 2D projection of each transformed 3D model. This approach shows promising results in estimating 2D–3D pose of the object of interest under a cluttered environment. However, it easily fails to maintain track in the case of its dynamic movement under occlusions; see the experiments in Section 4.2. In addition this framework suffers from high computational complexity due to the nature of a sampling based method.

In this paper, not only to utilize both frameworks above but also to overcome their disadvantages, we extend them by incorporating a particle filter to exploit the underlying dynamics of the system, and by developing an occlusion handling scheme to continuously track the object of interest in a more general and challenging environment. Compared with the approaches described in [13,40], we employ a more natural particle filtering scheme to generate and propagate the translation and rotation parameters in a decoupled manner in order to find the optimal pose of an object of interest. In addition, we focus more on occlusion detection and handling to maintain track in the presence of occlusions. In particular, we deal with partial occlusions that distort the region properties of the tracked object, which is a key feature for region-based tracking methods. To this end, we improve the l-iteration scheme introduced in [31,41] by controlling dependencies between predictions and measurements of the system according to the degree of occlusion and variation of the object's 3D pose from previous accumulated results; see Section 3.4. This improvement provides the robust method to deal with occlusions of an obstacle with different statistical properties in a tracking framework that relies heavily on region properties of an image. To the best of our knowledge, this is the first attempt to exploit the degree of dependencies between predictions and measurements of the system for these particular case of occlusions taken place in region-based frameworks involving iterative and derivative formulation.

Moreover, in the present work, the variational approach is embedded into the proposed framework in designing a measurement model to reduce computational complexity and to facilitate the effective search of a local optimum in a particle filtering framework. This method allows the samples to move further into modes of a filtering distribution so that only a few samples are necessary; see Section 3.3. Variational methods, such as Mean-shift [42], are typically gradient based optimization methods minimizing a cost functional in order to find the local mode of a probability distribution. To effectively reduce the sample size of the particle filtering framework, variational approaches are embedded into particle filters in a number of works. The authors in [43] present a mean shift embedded particle filter, in which a smaller number of samples is used to estimate the posterior distribution than conventional particle filters by shifting samples to their neighboring modes of the observation so that samples are moved to have large weights. In [44], the underlying probability density function (pdf) is represented with a semi-parametric method using a mean-shift based mode seeking algorithm to solve a tracking problem for high dimensional color imagery. The authors of [45] fuse a deterministic search based on gradient descent and random search guided by a stochastic motion model, then, in object tracking, they effectively switch two search methods according to a momentum using the inter-frame motion difference.

The remainder of this paper is organized as follows. In the next section, we briefly explain an overview of the two fundamental concepts used in the proposed method, particle filtering and the

gradient descent flow presented in [13], respectively. Section 3 describes the overall particle filtering approach for 2D object tracking and 3D pose estimation. Specifically, we derive a prediction model, a measurement model, as well as an occlusion handling scheme for the task of object tracking. In Section 4, we provide experiments on both synthetic and real life imagery in hopes of highlighting the viability (and limitations) of the proposed algorithm in the context of visual tracking. Lastly, we conclude and discuss possible future research directions in Section 5.

2. Preliminaries

2.1. Particle filtering

Sequential Bayesian filtering estimation with Monte Carlo simulation, called *particle filtering*, was first introduced by Gordon et al. [8]. In recent years, it has proven to be a powerful scheme for nonlinear and non-Gaussian estimation problems due to its simplicity and versatility.

The overall objective is to estimate a (hidden) variable of interest governing a particular underlying system with respect to what is being observed or measured. With this goal, we let s_t be a state vector and z_t be a set of observations. We then model the state transition equation and the observation equation by

$$\begin{aligned} s_{t+1} &= f_t(s_t, u_t) \\ z_t &= h_t(s_t, v_t) \end{aligned} \tag{1}$$

where u_t and v_t are independent and identically distributed (iid) random variables representing noise whose probability density functions (pdfs) are known. Furthermore, we assume that the initial state distribution $p(s_0)$ is known.

Drawing $N \gg 1$ samples $\{s_t^i\}_{i=1,\dots,N}$ from $p(s_t|z_{1:t})$, one can properly construct an empirical estimate of the true hidden state s_t. However, generating samples from the posterior distribution $p(s_t|z_{1:t})$ is usually not possible (e.g., f and h are non-linear). Thus, if one can only generate samples from a similar density $q(s_t|z_{1:t}) \approx p(s_t|z_{1:t})$, the problem becomes one of *importance sampling*. That is, one can form a Monte Carlo estimate of s_t by generating N samples according to $q(s_t|z_{1:t})$ with associated importance weights $\{w_t^i\}_{i=1,\dots,N}$ at each time t. More importantly, as the algorithm progresses and if N is chosen sufficiently large, the proposal distribution can be shown to *evolve* toward the correct posterior distribution, i.e., $q(s_t|z_{1:t}) = p(s_t|z_{1:t})$.

Thus, the generic algorithm begins by first sampling N times from initial state distribution, $p(s_0)$. Following this, the algorithm can be decomposed in two steps: the prediction step and the update step. Using importance sampling [21], the **prediction step** is the act of drawing N samples from the alternative proposal distribution $q(s_t|z_{1:t})$. As new information arrives online at time t from the observation z_t, one needs to evaluate the *fitness* of the predicted samples or particles. In other words, as z_t becomes available, the measurement or **update step** in particle filtering is incorporated through the importance weights by the following equation

$$\tilde{w}_t^i = \tilde{w}_{t-1}^i \frac{p(z_t|s_t^i)p(s_t^i|s_{t-1}^i)}{q(s_t^i|s_{t-1}^i, z_{1:t})} \tag{2}$$

where $p(z_t|s_t^i)$ is the likelihood of the arrived observation at time t. From the above approach, the filtering distribution is represented by a set of samples s_t^i and its associated weights w_t^i as

$$p(s_t|z_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(s_t - s_t^i) \tag{3}$$

where δ denotes the Dirac function, and w_t^i is the normalized weight of ith particle: $\sum_{i=1}^{N} w_t^i = 1$. Moreover, one can now obtain an empirical estimate of the state s_t via maximum likelihood or through a different statistical measure. We should note that in all particle filtering methods, a resampling operation is generally performed to eliminate particles with low weights (sample degeneracy). On the other hand, if one resamples inefficiently, there may be a loss of diversity for a set of particles (sample impoverishment). We refer the reader to Section 3 for more information on resampling.

2.2. Energy model

Like [13,40], we assume we have prior knowledge of the 3D shape of interest for which we would like to estimate the corresponding 3D pose from the 2D scene. Let $\mathbf{X} = [X, Y, Z]^T$ be the coordinates of a point in $\mathbb{R}^3$ with respect to the referential of the camera. Here, it is assumed that the calibrated camera is already given and is modeled as a pinhole camera: $\boldsymbol{\pi}: \mathbb{R}^3 \mapsto \Omega;\ \mathbf{X} \mapsto \mathbf{x}$ where $\Omega \subset \mathbb{R}^2$ is the domain of an image $I(\mathbf{x})$ and $\mathbf{x} = [x, y]^T = [X/Z, Y/Z]^T$ denotes coordinates in Ω. $S \subset \mathbb{R}^3$ is a smooth surface representing the shape of interest and $\mathbf{N} = [N_1, N_2, N_3]^T$ denotes the outward unit normal to S at each point $\mathbf{X} \in S$. Let $R = \boldsymbol{\pi}(S) \subset \Omega$ be the region on which the surface S is projected and $R^c = \Omega \backslash R$ be the complementary region of R. Similarly, the curve $\hat{c} = \boldsymbol{\pi}(C) \subset \Omega$ is the projection of the curve $C \subset S$ and $\hat{c}$ also denotes a boundary of R, $\hat{c} = \partial R$. Note, the curve $\hat{c}$ in 2D and the curve C in 3D are referred to as the *silhouette* and the *occluding curve*, respectively.

For our particular segmentation problem, we seek to find a boundary that optimally partitions the object of interest or foreground from the corresponding background in a given 2D image. Inspired by region-based active contour models [46–48], the authors in [13] define an objective energy functional based on the global statistics of an image so that the curve $\hat{c}$ (and 3D pose) is evolved to maximize the image statistical measure of discrepancy between its interior and exterior regions. This is given as follows:

$$E = \int_R r_o(I(\mathbf{x}), \hat{c}) d\Omega + \int_{R^c} r_b(I(\mathbf{x}), \hat{c}) d\Omega \tag{4}$$

where $r_o : \chi,\ \Omega \mapsto \mathbb{R}$ and $r_b : \chi,\ \Omega \mapsto \mathbb{R}$ are functions measuring the visual consistency of the image pixels with a statistical model over the regions R and R^c, respectively. Here, χ is the space that corresponds to photometric variable of interest. In this work, r_o and r_b are given by:

$$r_o = -\log(\Sigma_o) - \frac{(I(\mathbf{x}) - \mu_o)^2}{\Sigma_o}, \quad r_b = -\log(\Sigma_b) - \frac{(I(\mathbf{x}) - \mu_b)^2}{\Sigma_b} \tag{5}$$

Here Σ_o and Σ_b are variances inside and outside the curve $\hat{c}$, and are given by

$$\Sigma_o = \frac{\int_R (I(\mathbf{x}) - \mu_o)^2 d\Omega}{\int_R d\Omega}, \quad \Sigma_b = \frac{\int_{R^c} (I(\mathbf{x}) - \mu_b)^2 d\Omega}{\int_{R^c} d\Omega} \tag{6}$$

where

$$\mu_o = \frac{\int_R I(\mathbf{x}) d\Omega}{\int_R d\Omega}, \quad \mu_o = \frac{\int_{R^c} I(\mathbf{x}) d\Omega}{\int_{R^c} d\Omega} \tag{7}$$

For gray-scale images, $\mu_{o/b}$ and $\Sigma_{o/b}$ are scalars and for color images, $\mu_{o/b} \in \mathbb{R}^3$ and $\Sigma_{o/b} \in \mathbb{R}^{3\times 3}$ are vectors and matrices. Note that r_o and r_b can be chosen as various forms describing the region properties of the pixels located inside and outside the curve (e.g., mean intensities [46], distinct Gaussian densities [47], and generalized histograms [48]). As seen above, r_o and r_b are chosen to be the region based functional of [47].

2.3. Gradient descent flow

Let $X_0 \in \mathbb{R}^3$ be the coordinates of points on S_0 where S_0 is the identical reference surface in 3D. By the rigid transformation $g \in SE(3)$, one can locate S in the camera referential by $S = g(S_0)$. Written in a point wise fashion yields $\mathbf{X} = g(\mathbf{X}_0) = \mathbf{R}\mathbf{X}_0 + \mathbf{T}$ with $\mathbf{R} \in SO(3)$ denoting the rotational group and $\mathbf{T} \in \mathbb{R}^3$ representing translations. Here, 3D pose motions are represented by a set of six parameters. The parameters of the rigid motion g will be denoted by $\lambda = [\lambda_1, \ldots, \lambda_6]^T = [t_x, t_y, t_z, \omega_x, \omega_y, \omega_z]^T$. Rotations are represented in exponential coordinates, which is a more compact form than using quaternion (4 entries) or basic rotation matrices in three dimensions (12 entries); see [5]. Now, since we assume that the 3D shape of the rigid object is known, our objective is to minimize energy E in Eq. (4) by exploring only the regions R and R^c that result from projecting the surface S onto the image plane. For a calibrated camera, these regions are functions of the transformation g only. Solving for the transformation that minimizes E can be undertaken via gradient descent over the parameters λ. This is described next.

The partial differentials of E with respect to the pose parameters λ_i's can be computed using the chain-rule:

$$\frac{\partial E}{\partial \lambda_i} = \int_{\hat{c}} (r_o(I(\mathbf{x})) - r_b(I(\mathbf{x})))\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} + \int_R \left\langle \frac{\partial r_o}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega + \int_{R^c} \left\langle \frac{\partial r_b}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega \tag{8}$$

where $\hat{s}$ is the arc-length parameterization of the silhouette $\hat{c}$ and $\hat{\mathbf{n}}$ is the (outward) normal to the curve at $\mathbf{x}$.

Using the arc-length s of C and $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, one has

$$\begin{aligned} \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, J\frac{\partial \hat{c}}{\partial \hat{s}} \right\rangle d\hat{s} = \left\langle \frac{\partial \boldsymbol{\pi}(C)}{\partial \lambda_i}, J\frac{\partial \boldsymbol{\pi}(C)}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix} \frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds \\ &= \frac{\|\mathbf{X}\|}{Z^3}\sqrt{\frac{\kappa_{\mathbf{X}}\kappa_{\mathbf{t}}}{K}}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds \end{aligned} \tag{9}$$

where K denotes the Gaussian curvature, and $\kappa_{\mathbf{X}}$ and $\kappa_{\mathbf{t}}$ denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$, respectively, where $\mathbf{t}$ is the vector tangent to the curve C at the point $\mathbf{X}$, i.e. $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$. Note that the last two terms in Eq. (8) collapse due to the choice of the r_o and r_b in Eq. (5). Now we have the following gradient descent flow (see [13] for detailed computation):

$$\frac{\partial E}{\partial \lambda_i} = \int_C (r_o(I(\boldsymbol{\pi}(\mathbf{X}))) - r_b(I(\boldsymbol{\pi}(\mathbf{X})))) \cdot \frac{\|\mathbf{X}\|}{Z^3}\sqrt{\frac{\kappa_{\mathbf{X}}\kappa_{\mathbf{t}}}{K}}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds \tag{10}$$

where the term $\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle$ can be computed for the evolution of the pose parameter λ_i which is a translation parameter ($i = 1, 2, 3$) or a rotation parameter ($i = 4, 5, 6$):

- For a translation parameter,

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0 + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = N_i \tag{11}$$

where the Kronecker symbol $\delta_{i,j}$ was used ($\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ otherwise) and $\mathbf{T} = [t_x, t_y, t_z]^T = [\lambda_1, \lambda_2, \lambda_3]^T$.

- For a rotation parameter,

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X}_0}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \exp\left(\begin{bmatrix} 0 & -\lambda_6 & \lambda_5 \\ \lambda_6 & 0 & -\lambda_4 \\ -\lambda_5 & \lambda_4 & 0 \end{bmatrix} \right) \begin{bmatrix} 0 & -\delta_{6,i} & \delta_{5,i} \\ \delta_{6,i} & 0 & -\delta_{4,i} \\ -\delta_{5,i} & \delta_{4,i} & 0 \end{bmatrix} \mathbf{X}_0, \mathbf{N} \right\rangle \tag{12}$$

3. Proposed method

3.1. State space model

In what follows, the location of the object can be characterized by the translation and rotation parameters of a rigid transformation. Thus, the (hidden) variable or state s_t that we want to estimate are the pose parameters at time t and is given by

$$s_t = \begin{pmatrix} \mathbf{T} \\ \mathbf{W} \end{pmatrix}_t \tag{13}$$

Here, the translation and rotation vectors are $\mathbf{T} = [t_x, t_y, t_z]^T$ and $\mathbf{W} = [\omega_x, \omega_y, \omega_z]^T$, respectively. We should mention that many 2D visual tracking schemes involving elastic deformations of the target have a theoretically infinite dimensional state space [49,50]. In contrast, the state variable describing the (perceived) deformation in the 2D domain can now be succinctly represented via a finite set in the 3D space. Note that this only holds for the particular but general case of tracking rigid objects. Lastly, when a new image I_t arrives at time t, we obtain an observation z_t.

3.2. Prediction model

As mentioned previously, it is important to carefully utilize system dynamics in order for a given tracking algorithm to converge to the correct optimum. One may choose a random walk model for the state transition equation. However, since this model is usually only practical with a sufficiently large number of samples in a particle filtering framework, we chose instead to employ a first-order autoregressive (AR) model [1] for the state dynamics. We perform the propagation of translation and rotation parameters in a decoupled manner. Consequently, the system dynamics for predicting pose parameters, $\hat{s}_t = [\widehat{\mathbf{T}}_t, \widehat{\mathbf{W}}_t]^T$, is given by:

$$\begin{aligned} \widehat{\mathbf{T}}_t^i &= \mathbf{T}_{t-1}^i + A(\widehat{\mathbf{T}}_{t-1}^i - \mathbf{T}_{t-1}^i) + u_t^i \\ \widehat{\mathbf{W}}_t^i &= \mathbf{W}_{t-1}^i + u_t^i \end{aligned} \tag{14}$$

$\mathbf{T}_{t-1}$ and $\mathbf{W}_{t-1}$ are the translation and rotation vectors at time $t-1$, respectively. The rotation matrix R for each particle $\{s_{t-1}^i\}_{i=1,\ldots,N}$ can be computed by $\widehat{R}_t^i = \exp(\widehat{W}_t^i) \cdot R_{t-1}$ where $\exp(\cdot)$ denotes the matrix exponential [5] and A is the AR process parameter. The noise model u_t^i is defined as:

$$u_t^i \sim \mathcal{N}(0, \rho \cdot e_{t-1}^i (e_{t-1}^i)^T) \tag{15}$$

where $\mathcal{N}(\cdot)$ represents the normal distribution and ρ is a user-defined diffusion weight. Moreover, a motion alignment error e_{t-1}^i for each particle $\{s_{t-1}^i\}_{i=1,\ldots,N}$ is obtained from the predicted and measured states at time $t-1$, i.e.,

$$e_{t-1}^i = \tilde{s}_{t-1}^i - \hat{s}_{t-1}^i \tag{16}$$

where $\tilde{s}_{t-1} = [\widetilde{\mathbf{T}}_{t-1}, \widetilde{\mathbf{W}}_{t-1}]$ is the measured state vector at time $t-1$. Here, it is noted that the proposed decoupled methodology not only provides flexibility in designing the state dynamics of the system, but it also allows us to differently deal with the translational and rotational equations in (14) by controlling the diffusion weight. For example, since an orientation space is wrapped on itself and angles behave linearly within small angle approximation [51], we apply a small perturbation to only the rotational dynamics without affecting the translation equation because of the decoupling.

Now, inspired by [41], we define the bandwidth b_{t-1}^i for each particle as the combination of the diffusion weight ρ and the motion alignment error e_{t-1}^i. That is, $b_{t-1}^i = \rho \cdot e_{t-1}^i (e_{t-1}^i)^T$. Then the process noise can be represented by a multivariate Gaussian distribution based on the bandwidth term, b_{t-1}^i:

$$u_t \sim \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi)^{N/2}|b_{t-1}^i|^{1/2}} e^{-\frac{1}{2}\left(\tilde{s}_{t-1}^i - \hat{s}_{t-1}^i\right)^T \left(b_{t-1}^i\right)^{-1}\left(\tilde{s}_{t-1}^i - \hat{s}_{t-1}^i\right)} \tag{17}$$

From the Eq. (17) above, one can see that the particles are driven by the bandwidth term in an online fashion and diffuse in the direction of motion of the object. Next, we discuss the measurement model.

3.3. Measurement model

Now we specify the measurement function $h(\cdot)$ for the observed image I_t at time t as follows: First, we carry out l-iterations of gradient descent flow in Eq. (10) for each particle $\{\hat{s}_t^i\}_{i=1,\ldots,N}$. Here, it should be noted that the choice of l is carefully considered to avoid *sample degeneracy* and *sample impoverishment* (see [31] for details). If the tracker reaches a local minimum of the objective functional in Eq. (4) with too large l, the state at time t would lose the temporal coherency with the state at time $t-1$ (*sample degeneracy*). On the other hand, if l is selected to be too small, most particles would not move to the region of high likelihood (*sample impoverishment*). While the authors in [31,41] use a l-iteration scheme for their resampling algorithm, l is experimentally chosen based on the imagery and the type of local optimizer used. In the present work, contrary to previous works, the number of l is dynamically chosen online according to the degree of occlusion and object's movement. This will be described in Section 3.4.

Next, we assign and update the importance weights associated for each particle. This is done by employing a transitional prior density as the proposal distribution [52]. In doing so, the weight update equation is $w_t^i = w_{t-1}^i p(z_t|s_t)$ where the likelihood of the observation is defined as:

$$p(z_t|s_t) = e^{-E(\mathbf{T}_t, \mathbf{W}_t, I_t)} \tag{18}$$

Lastly, we obtain the measurement for time t. That is, the pose parameters with the minimum energy are taken as the measurement pose. Thus, the projected silhouette is the best fitting curve which makes its interior and exterior statistical properties be maximally different in a 2D image domain (i.e., it minimizes the region-based energy in Eq. (4)).

3.4. Occlusion handling

The presence of occlusions generally hinders tracking algorithms from continuously tracking an object of interest. Occlusion detection is a necessary task before performing occlusion handling. In [35], the authors detect an occlusion using the relative change of the object's size compared to the average object's size as well as the distance between the object and an obstacle. However, this method may give ambiguous results if an object's size changes due to camera zooming. Here, we propose a histogram based occlusion detection technique, which is performed by checking the variation of the histogram of the object during tracking. To do this, we define an appearance model as a normalized histogram h. For color-based imagery, a histogram is calculated by the mean of color intensities (in RGB spaces) for pixels inside $\hat{c}$. Here, the RGB color space is normalized to remove the effect of intensity variations. Note that better performance could be expected if one uses other color spaces or image feature descriptors such as HSV, $m_1m_2m_3$ spaces [53], spatiograms [54], and HOG (histograms of oriented gradients) [55]. However, we chose the normalized histogram in the normalized RGB space to construct the appearance model due to their simplicity and usability.

The evaluation of the histogram change is achieved by computing the Bhattacharyya coefficient between two appearance models of the current silhouette curve and of the template model. The template model can be obtained from the initial curve of the first segmentation of the given sequence. The Bhattacharyya distance [56] between two probability density functions (pdfs), p_1 and p_2, is defined by:

$$D_B = -log\left(\int_{\mathbb{R}^2} \sqrt{p_1(\mathbf{x})p_2(\mathbf{x})}d\mathbf{x}\right) \tag{19}$$

Considering discrete densities, such as $h_t(b)$ and $h_{template}(b)$, the Bhattacharyya coefficient is defined as:

$$\beta = \sum_{b\in\mathbb{R}} \sqrt{h_t(b)h_{template}(b)} \tag{20}$$

where $h_t(b)$ and $h_{template}(b)$, with $b \in \mathbb{R}$, are appearance models of the curve $\hat{c}$ at time t and of a template model, respectively. The Bhattacharyya coefficient β varies between 0 and 1 (0 indicates complete mismatch and 1 is a perfect match). Small β indicates that another object has occluded the target being tracked because statistical information inside the tracked object is changed. Thus, in this work, the Bhattacharyya coefficient β is used as the appearance similarity measure between two histograms. Unfortunately, the template model can be influenced by illumination changes and geometric variations as well as through camera angle differences even though no occlusion occurs. To cover the undesired appearance changes, we should periodically update the template model according to the degree of histogram variations between frames. More specifically, if the histogram of the target is changing slowly over time, the template model is updated as the new histogram of the segmented object at a current frame and is preserved otherwise. The update condition for the template model is defined as

$$\beta(h_{template}, h_{(t-1)t_d}) > \beta_{th}, \quad (t > 1) \tag{21}$$

where β_{th} is a positive threshold between 0 and 1. The value t_d is the user-defined checking interval. Note that the interval for checking should be set large enough because the histogram variance between consecutive frames is generally small even though the occlusion occurs. This approach allows the tracker to keep the histogram of the template model until the sequence finishes. In this work, β_{th} and t_d are experimentally selected as [0.7, 0.8] and [10, 20], respectively.

Now, we elucidate the proposed occlusion handling scheme. Recently, region-based pose tracking of [57] deal with occlusions by introducing the visibility function and background manipulation. These methods are used to exclude the occluded points from the interior region of the tracked object and to simplify background region, respectively, in estimating probability density functions (pdfs) for interior and exterior regions of the tracked object and an obstacle. Contrary to the work of [57], in this paper, we propose a method for occlusion handling without manipulating computation of each probability density function (pdf) inside and outside the object. Instead, we follow the underlying concept of Bayesian estimation in the context of visual tracking. The basic idea of occlusion handling for this paper is regularization between the measurement and prediction model of the system. To do this, l-iterations of gradient descent of the objective functional in Eq. (10) is dynamically adjusted in an online manner. Not only is this l-iteration scheme used to resample the particles in Section 3.3, but also it can be interpreted as a function of the *Kalman gain* in a Kalman filtering like framework. In other words, one can view this parameter as the amount of confidence in the system model with respect to the current measurements [37]. For example, if the object being tracked is not observed appropriately during occlusion, then the degree of trust of the measurement model is reduced. Thus, we should only employ a few iterations within the measurement model so that the method depends more on the prediction model to maintain the track. On the other hand, if one can completely trust the obtained measurement (e.g., the observed image

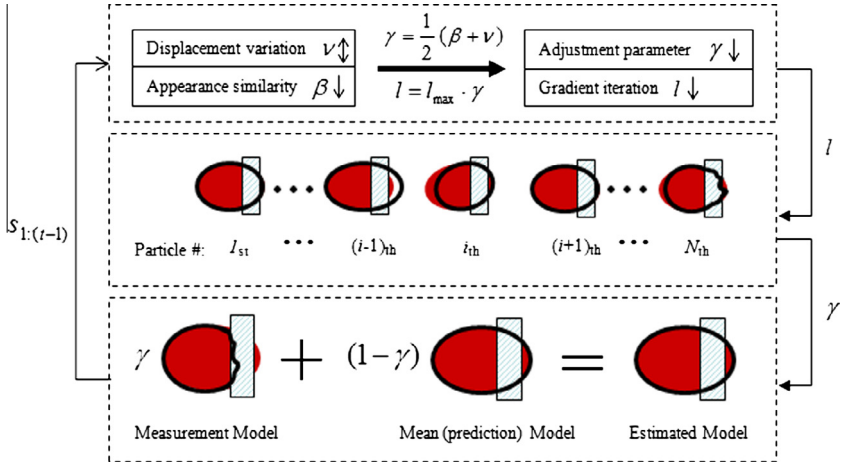

Fig. 1. Schema summarizing the proposed occlusion handling scheme.

Fig. 2. Different views of 3D point models used in this paper: an elephant, a helicopter, and a car.

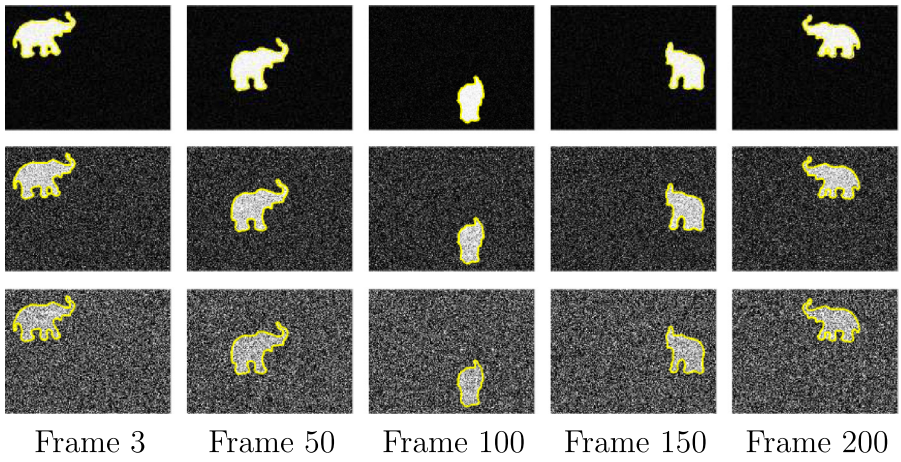

Fig. 3. Quantitative tracking results for robustness test to noise over 200 frames of the sequences. Gaussian noises with $\sigma_n^2 = 1\%$ (upper row), $\sigma_n^2 = 25\%$ (middle row), and $\sigma_n^2 = 100\%$ (bottom row) were added, respectively.

Table 1
Noise level table: table displaying % – *absolute* error statistics over 200 frames of the sequences as given in Fig. 3. T.avg, T.std, R.avg, and R.std denote average and standard deviation of translation error and rotation error, respectively.

Noise level (%)	% – *absolute* error (in %)			
	T.avg	T.std	R.avg	R.std
1	3.01	0.74	3.56	3.29
25	3.00	0.74	3.68	3.41
50	3.05	0.86	3.86	3.48
75	3.25	1.58	5.17	4.42
100	3.52	1.68	5.09	4.65

shows a smooth object movement without occlusions), the number of l should be assigned to be relatively large.

While the method proposed is similar to that of [37], a key difference is that we dynamically choose the number of l-iterations online based on both the degree of occlusion (or severity) and the degree of the object's motion displacement as opposed to an experimental fixed choice of l. In addition, the degree of object's pose variation is obtained from using the accumulated history of the object's location; translation and rotation vectors

$$v = 1 - \exp\{-\mathbf{var}(s_{(t-t_d):t})\} \tag{22}$$

where $\mathbf{var}(\cdot)$ is variance of the given variable and the checking interval, t_d, indicates how many previous states are used. If the variation of the location over the previous frames is large, we infer that the object is moving during the sequence and l is maintained so that the tracker is able to follow the object's movement as much as possible.

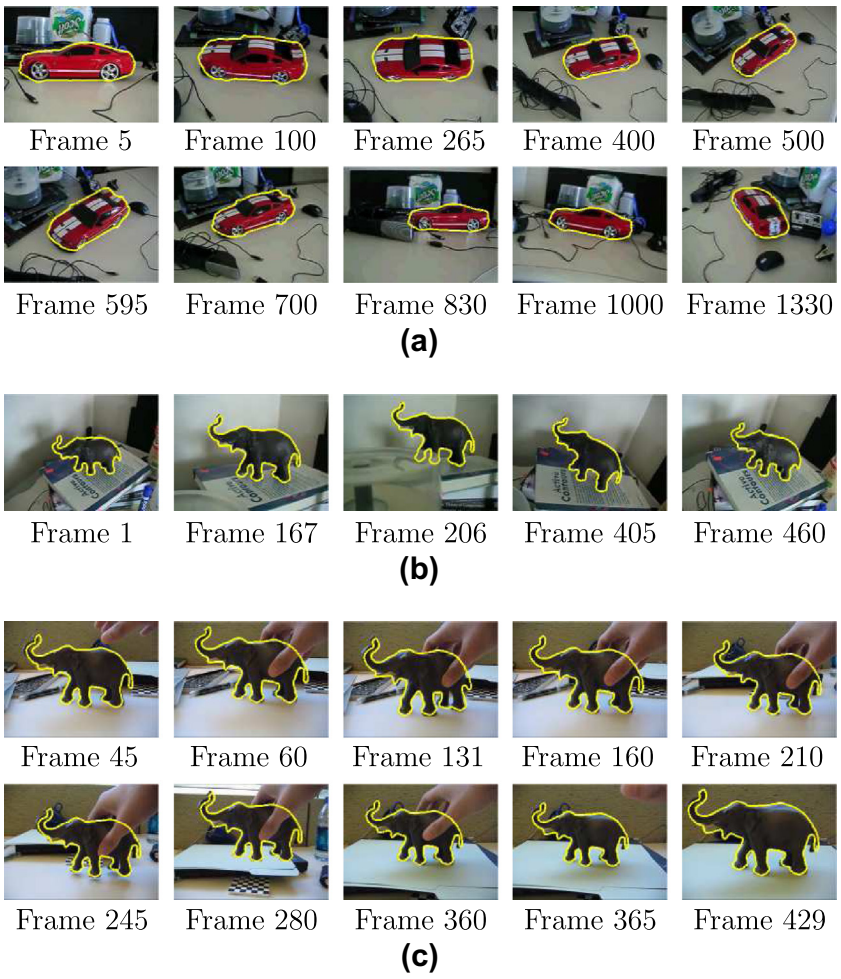

Fig. 4. Tracking in noisy and cluttered environments.

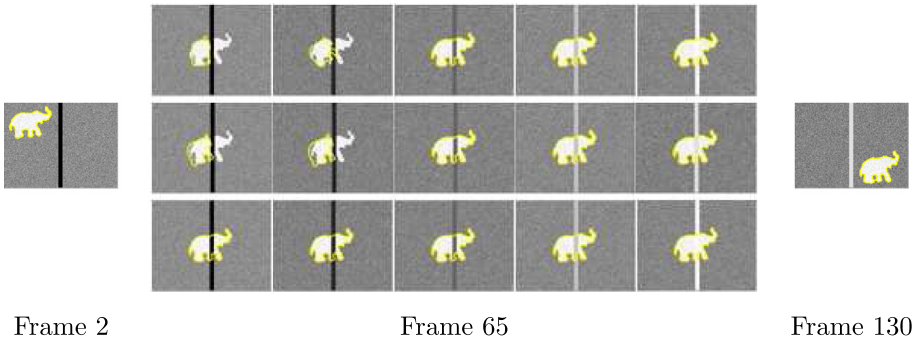

Fig. 5. Quantitative tracking results for robustness test to different brightness levels of the occlusion bar over 130 frames of the sequences. Gaussian noise with $\sigma_n^2 = 1\%$ was added. From left to right in frame 65, the brightness levels of the bar were assigned as 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. Upper row in frame 65: results using the method in [13]. Middle row in frame 65: results using the method in [40]. Bottom row in frame 65: results using the proposed method. In frame 2 and frame 130, results using the proposed method are only displayed when the brightness levels of the bar were assigned as 0.1 and 0.9, respectively.

Now the adjustment parameter γ is computed and the number of l-iterations is assigned online as follows:

$$\gamma = \frac{1}{2}(\beta + v), \quad l = l_{max} \cdot \gamma \tag{23}$$

where l_{max} is a maximum allowance iteration number of l. This can also be viewed as a boundary and initial condition of l.

The overview of the proposed occlusion handling scheme is illustrated in Fig. 1. Here, if the occlusion is detected, the adjustment parameter γ decreases according to the displacement variation v and appearance similarity parameter β. Consequently, it reduces the number of l-iterations of the gradient descent flow. This approach eventually overcomes occlusions in the course of tracking due to the fact that it makes the algorithm depend more on the prediction model in challenging situations.

Table 2
Brightness level table: table displaying % − *absolute* error statistics over 130 frames of the sequences as given in Fig. 5. The indicators, *, $^\diamond$ and $^\#$, denote the results using the proposed method, using the method in [40], and using the method in [13], respectively. $T^{(\cdot)}$ and $R^{(\cdot)}$ denote the average values of translation error and rotation error, respectively. Note that no % − *absolute* errors are obtained in case of the loss of track.

Brightness level	% − *absolute* error (in %)					
	T^*	$T^\diamond$	$T^\#$	R^*	$R^\diamond$	$R^\#$
0.1	2.2	n/a	n/a	3.77	n/a	n/a
0.3	2.2	n/a	n/a	3.77	n/a	n/a
0.5	2.4	1.82	2.47	3.8	4.9	4.4
0.7	1.7	2.14	2.55	3.2	4.11	4.19
0.9	2.1	2.44	2.8	2.9	4.41	4.5

3.5. Tracking framework

An overview of the proposed system for 2D–3D object tracking using the particle filtering framework is now described.

1. Initialization Step:
   (a) Initialize state, s_t, at $t = 0$ by using 2D segmentation and 3D pose estimation method introduced in [13] in the first frame of the given sequence.
   (b) Obtain the template appearance model, $h_{template}(\mathbf{x})$, at $t = 0$ from the initial segmented curve in a 2D image plane.
2. Prediction Step:
   (a) Generate N transformation parameters, $\{\hat{s}_t^i\}_{i=1,\ldots,N}$, around s_{t-1}^i by Eq. (14):

$$\widehat{\mathbf{T}}_t^i = \mathbf{T}_{t-1}^i + A(\widehat{\mathbf{T}}_{t-1}^i - \mathbf{T}_{t-1}^i) + u_t^i$$
$$\widehat{\mathbf{W}}_t^i = \mathbf{W}_{t-1}^i + u_t^i$$

3. Update Step:
   (a) Perform l-iterations of the gradient descent flow in Eq. (10) on each generated parameter, $\{\hat{s}_t^i\}_{i=1,\ldots,N}$.
   (b) Calculate the importance weights from Eq. (18):

$$\tilde{w}_t^i = \tilde{w}_{t-1}^i e^{-E\left(\widehat{\mathbf{T}}_t^i, \widehat{\mathbf{W}}_t^i, I_t\right)}$$

and normalize:

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{i=1}^N \tilde{w}_t^i}$$

   (c) Represent the posterior distribution of the system by a set of weighted particles: $p(s_t|z_{1:t}) = \sum_{i=1}^N w_t^i \delta(s_t - s_t^i)$.
   (d) Resample N particles according to $p(s_t|z_{1:t})$ by using the generic re-sampling scheme introduced in [52]. Note that $\{\bar{s}_t^i\}_{i=1\ldots N}$ denote resampled particles.
4. Adjustment Step:
   (a) Estimate state, s_t, using the mean state of the set $\bar{s}_t^i$ and the measurement state s_t^m as follows:

$$s_t = \gamma s_t^m + (1-\gamma)\left(\frac{1}{N}\sum_{i=1}^N \bar{s}_t^i\right) \tag{24}$$

   (b) Update the template appearance model, $h_{template}$, by the condition in Eq. (21) and compute the appearance similarity measure, $\beta(h_t, h_{template})$.
   (c) Compute the adjustment parameter γ and assign the number of l by Eq. (23).

4. Experiments

Various synthetic and real sequences of different rigid objects were used to demonstrate the robustness of the proposed method to noise, cluttered environments, as well as the algorithm's ability to cope with partial occlusions and imperfect information. 3D models[1] used in this paper consist of an elephant, a car, and a helicopter as shown in Fig. 2. In this section, we provide qualitative and quantitative results of various tracking scenarios including a comparison to the algorithms presented in [13,40]. In particular, in the quantitative experiments, two quantitative results regarding the robustness to noise and occlusion of the proposed method are provided on synthetic data. We also should note that because code of other joint 2D–3D pose estimation/segmentation algorithms were not readily available, our experiments are focused on highlighting the advantages and limitations of exploiting dynamics in visual tracking. However, before doing so, we briefly mention some numerical details associated with the experiments performed.

Implementation details: In these experiments, the parameters used were held fixed across all sequences. Specifically, the value of maximum l-iteration, l_{max}, was selected within [20, 30]. Its range is chosen by depending on the objective functional used and its step-size (e.g. here, the gradient descent flow in Eq. (10)). The number of particles $N = 40$ was empirically chosen to provide good performance without significant computational burden. Note that the setting of minimal amount of samples ($N = 40$) can be realized by embedding the gradient descent flow into measurement function proposed in Section 3.3. Using this scheme allows the particle filtering framework to rely on a small number of particles as opposed to the conventional CONDENSATION filter [60]. Note, this setup is also used for exploration of shape deformation [31] and point set registration [41]. Overall, using un-optimized code for the proposed method shows acceptable performance with computation time of approximately 10 s per frame on a 3.6 GHz Windows machine with 2 GB of RAM. For reference, the running times for the methods of [13,40] were about 0.4 and about 20 s per frame on the same system, respectively.

4.1. Tracking in noisy and cluttered environments

In this subsection, first of all, we show quantitative results regarding the robustness of the proposed method to noise on synthetic data. In generating the synthetic data, we first construct a basic elephant sequence, and then add several diverse noise levels of Gaussian noise whose variance ranges from $\sigma_n^2 = 1\%$ to $\sigma_n^2 = 100\%$. The sequences (and results) can be seen in Fig. 3. The translation and rotation parameters linearly increase and decrease throughout the sequences of 200 frames to produce a large variation for the aspect of the object. The size of the sequence images is 242×322. To quantitatively evaluate the tracking results, percent (%)-*absolute* errors are computed for both translation and rotation over each level of noise sequences:

$$\%-\textit{absolute}\ \textbf{error} = \frac{\|\mathbf{v}_{\text{measured}} - \mathbf{v}_{\text{truth}}\|}{\|\mathbf{v}_{\text{truth}}\|} \times 100 \tag{25}$$

where $\mathbf{v}_{\text{measured}}$ and $\mathbf{v}_{\text{truth}}$ are measured and ground-truth of translation and rotation vectors, respectively. Table 1 displays average and standard deviation errors of position and rotation of the results from the sequences in Fig. 3. As the noise level increases, the average and standard deviation errors also increase. This is due to the probability of encountering unexpected local minima will also increase given that it is now harder to distinguish the object of inter-

[1] All 3D models for rigid objects were acquired from the method presented by Yezzi and Soatto [58,59].

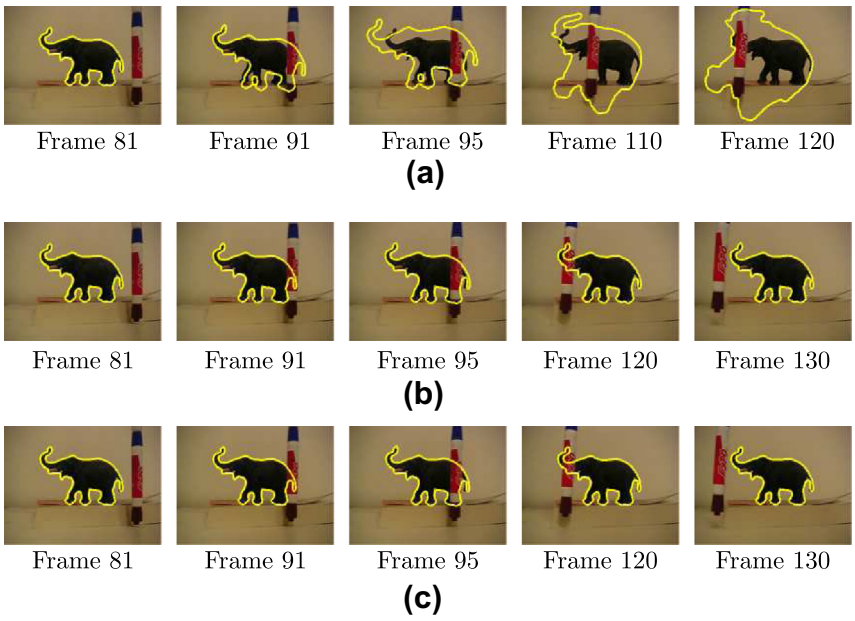

Fig. 6. Elephant sequence I with occlusion in a cluttered environment. Tracking results: (a) using the method in [13], (b) using the method in [40], and (c) using the proposed method.

est from the background. However, tracking is still maintained throughout the entirety of each sequence and the tracking errors did not exceed above 3% and 5% for translation and rotation, respectively. Note that the elephant and backgrounds are barely visibly distinguishable in the sequence of noise level $\sigma_n^2 = 100\%$.

Fig. 4 shows several sequences capturing the drastic changes of an object's pose in a cluttered background. Despite the cluttered background and significant changes of pose, successful tracking results were obtained. The sequence in Fig. 4a is comprised of 1350 frames, while Fig. 4b contains 470 frames. For the sequence shown in Fig. 4c, there are 450 frames available to track. In Figs. 4a and 4b, the red car and the gray elephant are observed under a dynamically moving camera. The elephant is manually moved by a hand as one can see in Fig. 4c. Thus, the corresponding pose of all the objects tested is altered significantly over the various sequences. Nevertheless, the proposed algorithm yields satisfactory tracking results. In particular, the hand holding the elephant partially occludes the elephant in Fig. 4c. However, in this case, since the hand has the similar color to the elephant, it does not have much effect on the statistical property of the elephant, and the track of the elephant is successfully maintained. In the next subsection, we demonstrate the ability of the proposed occlusion handling scheme to

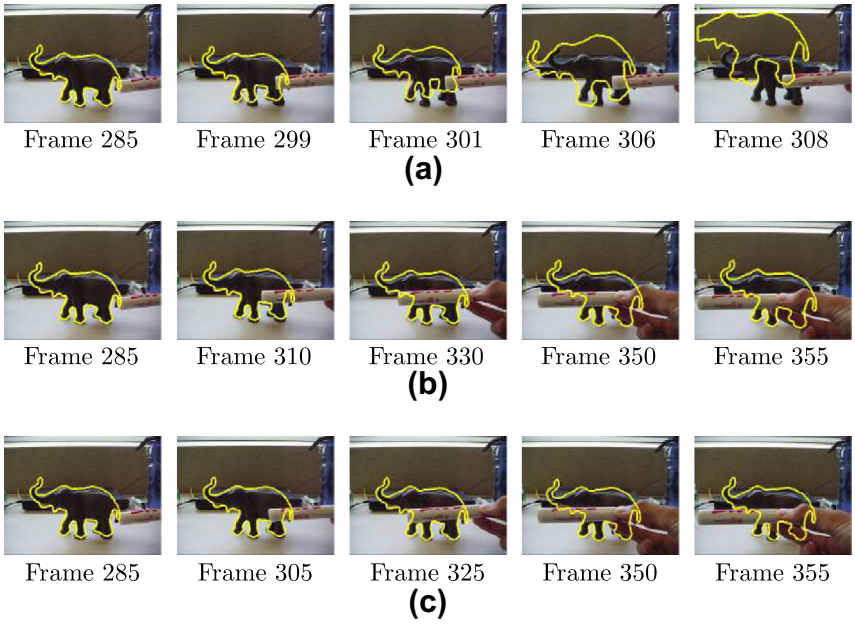

Fig. 7. Elephant sequence II with occlusion in a cluttered environment. Tracking results: (a) using the method in [13], (b) using the method in [40], and (c) using the proposed method.

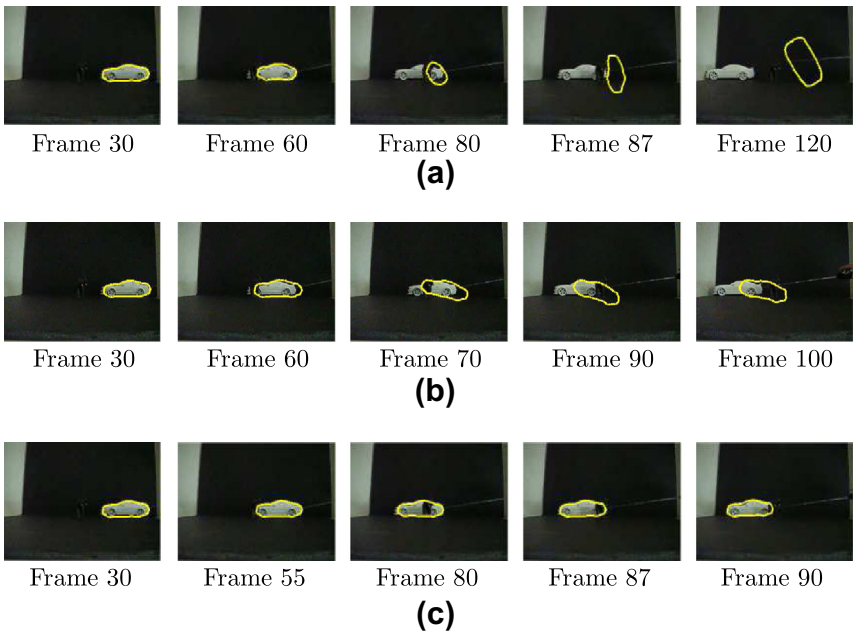

Fig. 8. Car sequence with occlusion in a cluttered environment. Tracking results: (a) using the method in [13], (b) using the method in [40], and (c) using the proposed method.

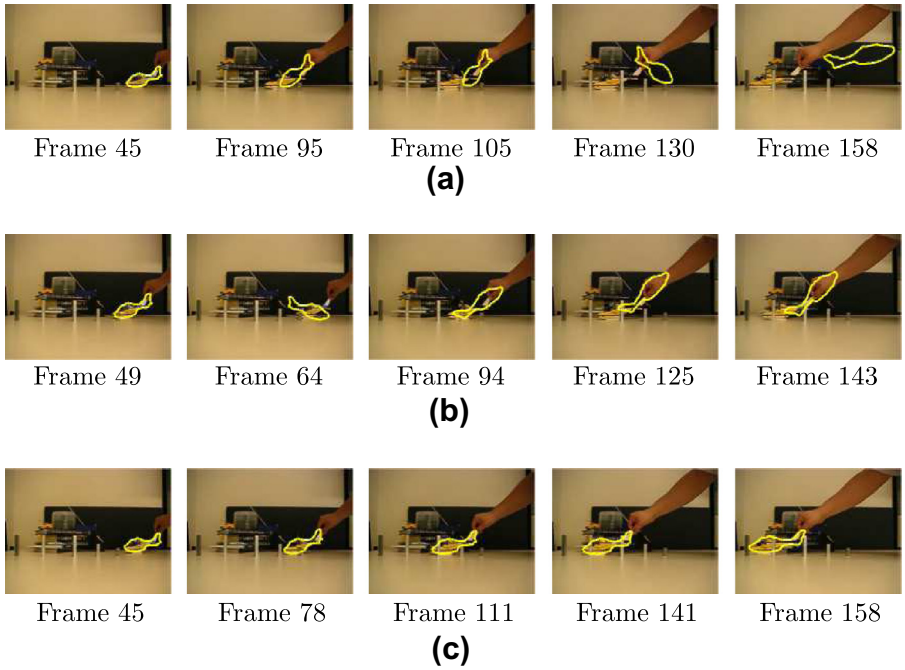

Fig. 9. Helicopter sequence with occlusion in a cluttered environment. Tracking results: (a) using the method in [13], (b) using the method in [40], and (c) using the proposed method.

deal with partial occlusions and imperfect information in the scenarios including the occlusions that contain a statistically different intensity from that of the object of interest.

4.2. Tracking in the presence of occlusion

In contrast to the previous sequences, the scenarios in this subsection not only include dynamic changes of an object's pose, but other objects (such as markers, staples, and a dark elephant), which occlude the object of interest, and provide an added difficulty to the overall tracking problem. Moreover, this subsection demonstrates how the proposed method outperformed the approaches of [13,40] in dealing with occlusion handling.

First, to provide quantitative results regarding the robustness of the proposed method in handling occlusions, we generate set of synthetic sequences, in which an obstacle bar exhibiting different levels of gray-scale intensity is added. In these sequences, Gaussian noise with zero mean and variance of $\sigma_n^2 = 1\%$ was added to a binary image of the toy elephant. From a tracking viewpoint, the bar prevents many algorithms, which rely only on image and shape information, from successfully maintaining track of the object. Specifically, we vary the brightness level of the bar from 0.1 to 0.9

where 0 denotes black and 1 denotes white. The results are shown in Fig. 5. In Fig. 5, when the brightness level of the bar is less than 0.5, the methods in [13,40] lose the track, but the proposed tracking algorithm maintains track over the generated sequences. The average values of translation and rotation errors are measured by Eq. (25) and are displayed in Table 2. The table shows the improvement of the proposed method for occlusion handling compared to the methods in [13,40]. More detailed comparison of the proposed algorithms with the methods of [13,40] is discussed in the next experiments on real sequences.

Figs. 6 and 7 show a red marker (vertically held by a hand) and a white marker (horizontally held by a hand) that pass by the gray elephant from right to left in a cluttered background, respectively. We should mention that the algorithms in [13] works well with the occlusions that are similar in nature to that of the interested object; see Fig. 5. However, due to the fact that the occlusion contains a statistically different intensity from that of the object of interest, the methods in [13] is not able to maintain track as shown in Figs. 6 and 7. Specifically, utilizing only the gradient descent flow presented in Eq. (10), the movement of the marker acts as if it *pushes* or *blows* the silhouette curve off of the elephant. This is simply due to the statistical difference between the object and the occlusion. That is, if one were to look at Eq. (10) from a robust statistics point of view, the flow regarding the integration about the occluding curve excludes the possible points on the white marker because they are viewed as outliers. In turn, one cannot properly estimate the 3D pose or maintain track. However, if one exploits the underlying dynamics as done in the proposed algorithm, one achieves a more robust result.

Compared to the sequences of Figs. 6 and 7, Figs. 8 and 9 show sequences with moving objects and static occlusions. Fig. 8 shows a white car is moved by a narrow stick and it is occluded by a black elephant while moving. In this sequence, the black board was used as a means for a clutter-free background such that tracking would be not affected from other possible clutter. As shown in Fig. 8a, the tracker in [13] yields unsatisfactory results in tracking the car while the proposed tracker successfully maintains track. In Fig. 9, the helicopter is manually moved throughout several set of staples that are positioned at different heights. While the tracker in [13] fails after the helicopter passes by the second set of staples, the proposed tracker overcomes the continual occlusions and maintains track of the pose of the helicopter as shown in Fig. 9a.

It is interesting to note that the tracker of [40] eventually lost track of the targets (the white car and the yellow helicopter) and got trapped in a local minimum as shown in Figs. 8b and 9b, respectively, even though it continuously estimated the pose of the elephant in the presence of occlusions (i.e., the red or white markers) in Figs. 6 and 7. These results show that the approach of [40] is more vulnerable to occlusions when it tracks a moving object than a stationary object. This is because the work of [40] disregards control of predictions and measurements of the system, which was taken into account in the present occlusion handling scheme. In addition, in the proposed method, in contrast to the work of [40], the separately distributed samples of pose parameters and the embedded variational technique aid the tracker in finding the optimum in a filtering distribution.

5. Conclusion

In this paper, we proposed a robust algorithm for 2D visual tracking and 3D pose estimation using particle filters. In particular, the degree of *trust* between predictions and measurements of the system is dynamically controlled in an online fashion to provide the reliable occlusion handling regardless of occlusions of an obstacle whose statistical properties are significantly different from those of the object of interest. The resulting methodology was shown to improve tracking performance in continuously locating the target even in the presence of noise, clutter, and occlusions during tracking in several challenging experiments.

The proposed method has some limitations that we intend to investigate in our future work. First of all, the overall algorithm is computationally expensive despite the benefits described in Section 3.3. In addition, our approach could lose the track for non-rigid objects. One possible solution is to incorporate knowledge of multiple 3D shapes as shown in [61]. In other words, for a successful non-rigid tracking framework one should include not only filtering of the pose parameters, but also the shape parameters. This approach would hopefully allow one to track a non-rigid objects of interest in some important scenarios. Moreover, for a number of practical applications, since it is impossible to always know the prior knowledge of the object's shape before using the proposed algorithms, automatic construction of a 3D shape prior is an important task. This may be accomplished, for example, by collecting information about the object from available 3D shape dictionaries or repositories.

Acknowledgments

This work was supported in part by grants from NIH, AFOSR, ARO, ONR, and MDA. This work is part of the National Alliance for Medical Image Computing (NA-MIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from http://nihroadmap.nih.gov/bioinformatics. Finally, this project was supported by grants from the National Center for Research Resources (P41-RR-013218) and the National Institute of Biomedical Imaging and Bioengineering (P41-EB-015902) of the National Institutes of Health.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cviu.2013.04.002.

References

[1] A. Blake, M. Isard, Active Contours, Springer, 1998.
[2] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34 (3) (2004) 334–352.
[3] J. Shi, C. Tomasi, Good features to track, in: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.
[4] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys (CSUR) 38 (4) (2006) 1–45.
[5] Y. Ma, S. Soatto, J. Kosecka, S. Sastry, An Invitation to 3D Vision, Spinger, 2003.
[6] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.
[7] S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Proceedings of the IEEE 92 (3) (2004) 401–420.
[8] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, in: IEE Proceedings F on Radar and Signal Processing, vol. 140, 1993, pp. 107–113.
[9] P. Kohli, J. Rihan, M. Bray, P. Torr, Simultaneous segmentation and pose estimation of humans using dynamic graph cuts, International Journal of Computer Vision 79 (3) (2008) 285–298.
[10] T. Riklin-Raviv, N. Kiryati, N. Sochen, Prior-based segmentation by projective registration and level sets, in: IEEE International Conference on Computer Vision, vol. 1, 2005, pp. 204–211.
[11] B. Rosenhahn, T. Brox, J. Weickert, Three-dimensional shape knowledge for joint image segmentation and pose tracking, International Journal of Computer Vision 73 (3) (2007) 243–262.
[12] C. Schmaltz, B. Rosenhahn, T. Brox, D. Cremers, J. Weickert, L. Wietzke, G. Sommer, Region-based pose tracking, Pattern Recognition and Image Analysis (2007) 56–63.
[13] S. Dambreville, R. Sandhu, A. Yezzi, A. Tannenbaum, A geometric approach to joint 2D region-based segmentation and 3D pose estimation using a 3D shape prior, SIAM Journal on Imaging Sciences 3 (1) (2010) 110–132.

[14] M. Abidi, T. Chandra, Pose estimation for camera calibration and landmark tracking, in: IEEE International Conference on Robotics and Automation, 1990, pp. 420–426.
[15] M. Dhome, M. Richetin, J. Lapreste, G. Rives, Determination of the attitude of 3D objects from a single perspective view, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (12) (1989) 1265–1278.
[16] É. Marchand, P. Bouthemy, F. Chaumette, A 2D–3D model-based approach to real-time visual tracking, Image and Vision Computing 19 (13) (2001) 941–955.
[17] B. Rosenhahn, C. Perwass, G. Sommer, Pose estimation of 3D free-form contours, International Journal of Computer Vision 62 (3) (2005) 267–289.
[18] T. Drummond, R. Cipolla, Real-time visual tracking of complex structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 932–946.
[19] D. Terzopoulos, R. Szeliski, Tracking with Kalman snakes, in: A. Blake, A. Yuille (Eds.), Active Vision, The MIT Press, 1993, pp. 3–20.
[20] E. Wan, R. Van Der Merwe, The unscented Kalman filter for nonlinear estimation, in: Adaptive Systems for Signal Processing, Communications, and Control Symposium, 2000, pp. 153–158.
[21] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing 10 (3) (2000) 197–208.
[22] D. Cremers, T. Kohlberger, C. Schnörr, Nonlinear shape statistics in Mumford-Shah based segmentation, in: European Conference on Computer Vision, 2002, pp. 516–518.
[23] T. Zhang, D. Freedman, Tracking objects using density matching and shape priors, in: IEEE International Conference on Computer Vision, 2003, pp. 1056–1062.
[24] P. Li, T. Zhang, B. Ma, Unscented Kalman filter for visual curve tracking, Image and Vision Computing 22 (2) (2004) 157–164.
[25] Y. Chen, T. Huang, Y. Rui, Parametric contour tracking using unscented Kalman filter, in: IEEE International Conference on Image Processing, vol. 3, 2002, pp. 613–616.
[26] C. Yang, R. Duraiswami, L. Davis, Fast multiple object tracking via a hierarchical particle filter, in: IEEE International Conference on Computer Vision, vol. 1, 2005, pp. 212–219.
[27] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49.
[28] J. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, 1999.
[29] N. Paragios, R. Deriche, Geodesic active contours and level sets for the detection and tracking of moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (3) (2000) 266–280.
[30] A. Yezzi, S. Soatto, Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images, International Journal of Computer Vision 53 (2) (2003) 153–167.
[31] Y. Rathi, N. Vaswani, A. Tannenbaum, A. Yezzi, Tracking deforming objects using particle filtering for geometric active contours, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (8) (2007) 1470–1475.
[32] N. Peterfreund, Robust tracking of position and velocity with Kalman snakes, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (6) (1999) 564–569.
[33] N. Peterfreund, The velocity snake: deformable contour for tracking in spatio-velocity space, Computer Vision and Image Understanding 73 (3) (1999) 346–356.
[34] H. Nguyen, A. Smeulders, Fast occluded object tracking by a robust appearance filter, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (8) (2004) 1099–1104.
[35] A. Yilmaz, X. Li, M. Shah, Contour-based object tracking with occlusion handling in video acquired using mobile cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1531–1536.
[36] M. Leventon, W. Grimson, O. Faugeras, Statistical shape influence in geodesic active contours, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2000, pp. 316–323.
[37] Y. Rathi, N. Vaswani, A. Tannenbaum, A generic framework for tracking using particle filter with dynamic shape prior, IEEE Transactions on Image Processing 16 (5) (2007) 1370–1382.
[38] X. Mei, H. Ling, Robust visual tracking using l-1 minimization, in: IEEE International Conference on Computer Vision, 2009, pp. 1436–1443.
[39] C. Bao, Y. Wu, H. Ling, H. Ji, Real time robust l1 tracker using accelerated proximal gradient approach, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1830–1837.
[40] J. Lee, R. Sandhu, A. Tannenbaum, Monte Carlo sampling for visual pose tracking, in: IEEE International Conference on Image Processing, 2011, pp. 509–512.
[41] R. Sandhu, S. Dambreville, A. Tannenbaum, Point set registration via particle filtering and stochastic dynamics, IEEE Transactions on Pattern Analysis and Machine Intelligence (2009) 1459–1473.
[42] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence (2003) 564–575.
[43] C. Shan, T. Tan, Y. Wei, Real-time hand tracking using a mean shift embedded particle filter, Pattern Recognition 40 (7) (2007) 1958–1970.
[44] B. Han, D. Comaniciu, Y. Zhu, L. Davis, Incremental density approximation and kernel-based Bayesian filtering for object tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 638–644.
[45] J. Sullivan, J. Rittscher, Guiding random particles by deterministic search, in: IEEE International Conference on Computer Vision, vol. 1, 2001, pp. 323–330.
[46] T. Chan, L. Vese, Active contours without edges, IEEE Transactions on Image Processing 10 (2) (2001) 266–277.
[47] N. Paragios, R. Deriche, Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision, Journal of Visual Communication and Image Representation 13 (1/2) (2002) 249–268.
[48] O. Michailovich, Y. Rathi, A. Tannenbaum, Image segmentation using active contours driven by the Bhattacharyya gradient flow, IEEE Transactions on Image Processing 16 (11) (2007) 2787–2801.
[49] S. Dambreville, Y. Rathi, A. Tannenbaum, Tracking deformable objects with unscented Kalman filtering and geometric active contours, in: American Control Conference, vol. 6, 2006, pp. 2856–2861.
[50] Y. Rathi, N. Vaswani, A. Tannenbaum, A. Yezzi, Particle filtering for geometric active contours with application to tracking moving and deforming objects, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 2–9.
[51] G. Bishop, G. Welch, B. Allen, Tracking: beyond 15 min of thought, in: SIGGRAPH Course 11, 2001.
[52] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, 2004.
[53] T. Gevers, W. Smeulders, Color based object recognition, Pattern Recognition 32 (3) (1999) 453–464.
[54] S. Birchfield, S. Rangarajan, Spatiograms versus histograms for region-based tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 1158–1163.
[55] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[56] T. Kailath, The divergence and Bhattacharyya distance measures in signal selection, IEEE Transactions on Communication Technology 15 (1) (1967) 52–60.
[57] C. Schmaltz, B. Rosenhahn, T. Brox, J. Weickert, Region-based pose tracking with occlusions using 3d models, Machine Vision and Applications (2011) 1–21.
[58] A. Yezzi, S. Soatto, Structure from motion for scenes without features, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2003, pp. 525–532.
[59] A. Yezzi, S. Soatto, Stereoscopic segmentation, International Journal of Computer Vision 53 (1) (2003) 31–43.
[60] M. Isard, A. Blake, Condensation-conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.
[61] R. Sandhu, S. Dambreville, A. Yezzi, A. Tannenbaum, Non-rigid 2D–3D pose estimation and 2D image segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 786–793.