

A Novel Sensing and Data Fusion System for 3-D Arm Motion Tracking in Telerehabilitation

Yaqin Tao and Huosheng Hu, *Senior Member, IEEE*

Abstract—In this paper, we present a novel sensing and data fusion system to track 3-D arm motion in a telerehabilitation program. A particle filter (PF) algorithm is adopted in the system to fuse data from inertial and visual sensors in a probabilistic manner. It is able to propagate multimodal distributions of system states based on an “importance sampling” technique by using sets of weighted particles. To avoid the problem of conventional PF algorithms that suffer from particle degeneracy and perform poorly in a narrow distribution situation, we adopt two strategies in our system, namely state space pruning and an arm physical geometry constraint. Experimental results show that the proposed PF framework outperforms other fusion methods and provides accurate results in comparison to the ground truth.

Index Terms—Biomedical measurements, particle filter (PF), sensor fusion, telerehabilitation, upper limb pose estimation, 3-D arm motion tracking.

I. INTRODUCTION

RECENTLY, vision-based tracking and analysis of human motion has become an active research area because of its wide real-world application such as surveillance monitoring [1], medical analysis [2], and human–computer interfaces [3], [4]. Existing visual tracking methods can roughly be divided into two groups as follows: 1) single camera tracking and 2) multiple camera tracking. Single camera tracking is cheap and desirable and is normally based on shape models. It is, however, affected by many constraints and requires prior information with regard to a subject’s motion and appearance [5]–[7]. In contrast, multiple camera tracking experiences fewer constraints but is expensive and difficult to construct. Due to the characteristics of visual sensors, both tracking methods have motion blur and occlusion problems, which hamper many real-world applications.

Therefore, it is necessary to integrate visual sensors with other types of sensors to improve tracking performance. Many hybrid approaches have been proposed to achieve this, such as visual–audio [8], visual–radar [9], and visual–inertial approaches [10]–[15]. In this paper, we employ visual and inertial sensors to track human arm motion in a telerehabilitation program [15], [16]. This hybrid tracking scheme is based on visual and inertial sensors to offer not only accurate motion tracking and stability but also robust performance over occlusions and

fast motion. It is intended to allow stroke patients to recover their motor ability in their home environments so that the burden on hospitals and physiotherapists can be relieved. However, its successful implementation relies on the proper use of fusion algorithms.

Traditionally, a Kalman filter (KF) is used to fuse data from different sensors; however, this is only suitable for linear systems. In nonlinear systems, an extended KF (EKF) is normally employed [10]–[15], which linearizes a nonlinear system. In both KF and EKF, the probability distribution of system state and noise are assumed as Gaussian distributions. EKFs work well when systems can be approximated using linear systems, and distributions are modeled as Gaussian. However, when systems are highly nonlinear, or a system’s state and noise are multimodal distributions, an EKF will produce noisy results or may fail to estimate the system’s state.

Recently, the particle filter (PF), which is also known as a sequential Monte Carlo (MC) method, condensation [17], or bootstrap filter, has become popular in sensor fusion [8], motion tracking [18]–[20], and object recognition [21] (in nonlinear systems with multimodal distributions). Its implementation is realized by successfully modeling system posterior distributions using sets of weighted particles and by propagating the sets of particles in terms of Bayes’ rule. The number of particles N determines how well the set of particles approximates the posterior distribution. In theory, when the number of particles N approaches infinite, the set of particles will represent the true posterior probability density function (pdf). Therefore, increasing the number of particles should improve fusion performance. However, computational cost exponentially increases with an increase in the number of particles. It is therefore necessary to find a tradeoff between PF performance and computational cost. On the other hand, a PF may suffer from the particle degeneracy problem, i.e., the number of effective particles decreases as time passes, and particle sets do not reflect the true posterior density. Effective particles are those with large weights.

There are many versions of PFs [22], whose essential difference is the way in which they tackle degeneracy problems. There are four major factors that affect the performance of a PF, i.e., the number of particles it uses, the choice of proposal distributions [25], the selection of resampling algorithms [22], and the dynamic model of a system. These factors are not exclusive, but they are related. A change in one factor will affect the others. Resampling is an effective method that can be employed to reduce the degeneracy problem. The idea is to eliminate particles that have small weights and to concentrate on particles with large weights by replicating particles in

Manuscript received February 2, 2007; revised August 24, 2007.

The authors are with the Department of Computing and Electronic Systems, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: ytao@essex.ac.uk; hhu@essex.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2007.913828

proportion to their weight. A number of resampling algorithms are available, namely residual, systematic, multiple-nominal, and stratified. Each of these has advantages and disadvantages. Douc *et al.* [23] compared various resampling approaches in detail. We chose a residual resampling algorithm, as it performed well in our application.

The performance of a PF also highly depends on the quality of the proposal distribution, i.e., a suitable distribution depending on the old state and new measurements. Doucet *et al.* [25] demonstrated that the optimal proposal distribution is the system posterior distribution. However, it is difficult to directly sample from the optimal proposal distribution, as they are rarely available in a closed form. Suboptimal proposal distributions are sought by employing the importance sampling approach, as introduced in [25], which describes the auxiliary knowledge of which areas of state space contain most information about the posterior given current measurements and old states. Particles are independently drawn from the proposal distribution. Isard and Blake [18] combined PF with an importance sampling function to fuse color and contour features. It greatly reduced the number of samples required by employing persistent color features. The sampling procedure is first guided by using color probability to roughly locate a target object and then refined by using contour tracking. Prez *et al.* fused different image features or information from audio to track motion [8]. Color features were employed to serve as primary features, whereas sound or motion features were used as auxiliary features to assist tracking. This method worked well when primary features were consistently reliable.

A simple version of a proposal distribution is the use of a dynamic system model [25], [27], which can be used to predict a system's state. The dynamic model of a system can be a constant velocity or an acceleration model. It can also be learned from sampling sequences [7]. The problem of using a dynamic model as a proposal distribution is that it poorly performs for narrow likelihood functions, particularly in higher dimension spaces like whole body human tracking [7]. To effectively sample particles from the posterior, Deutscher *et al.* [24] proposed an annealed PF. The posterior pdf was implemented from coarse to fine by resampling particles around the peaks of a previous layer of posterior pdf. No particles were sampled from the area of the state space, where the probability of the system state was low; this reduced the number of particles required to model the posterior pdf. A similar method called covariance scaled sampling was proposed by Sminchisescu and Triggs [6], which did not use layered posterior pdf. Only the exact posterior pdf was employed, and the method first found the "X" best local extremes in the posterior pdf and later sampled about extremes. Therefore, the particles focused on the most likely regions and avoided low probability regions. The drawback was that they could fail to track sudden motions.

Another method for improving PF performance is Rao-Blackwellization [25]. It divides a state space into linear and nonlinear subspaces and then uses PF and KF to separately propagate each substate space. The main principle is to use as many linear properties as possible. Experimental results show that it outperforms a conventional PF and uses

fewer particles [26]. The basic idea behind this method is to reduce the size of state vectors to reduce the number of particles. Inspired by the Rao-Blackwellization method, we propose a state space pruning method to reduce the number of particles in this paper. Although a good choice of proposal distribution and resampling can reduce the degeneracy effect, they cannot completely eliminate it. Therefore, an arm physical geometry constraint is employed in this research to deal with the degeneracy problem. In addition, a dynamic system model of constant acceleration is used as a proposal distribution in this paper for the following three reasons: 1) It can be used with an inertial sensor with high frequency data sampling properties, which makes a constant acceleration motion model realistic; 2) the state space of the system can be reduced by pruning; and 3) the introduced physical arm constraint greatly improves the system's performance by reducing the search space.

The rest of this paper is organized as follows. Section II describes our arm motion tracking system configuration and how the system configuration properties are used to improve PF performance by using a state space pruning strategy. In Section III, the inertial visual tracker is introduced, and a human arm geometry constraint is employed as a second strategy to improve PF performance. Some experimental results for the performance of a PF are provided in Section IV. Finally, conclusion and future work are presented in Section V.

II. SYSTEM OVERVIEW AND STATE SPACE PRUNING

A. System Configuration

This paper aims at tracking human arm motion. The arm is modeled as a skeleton structure, which consists of two segments linked by a revolute joint (see Fig. 1). A video camera and an inertial sensor (MT9 from XSENS) are employed in our sensing and data fusion system to capture the arm motion of a subject. The video camera is fixed and used to capture the motion of an arm. The inertial sensor is attached to the wrist of the subject to measure his/her acceleration and orientation changes. The arm motion tracking is achieved by integrating the visual and inertial information.

In total, there are three coordinate systems involved in our arm motion tracking: 1) a camera frame (C); 2) an inertial sensor frame (B); and 3) a reference system (W). Note that frame B is the body attached to the frame of the inertial sensor (please see Fig. 2), which is attached to the wrist joint of the human arm. The origin of frame B coincides with the position of the wrist joint, and its y -axis is aligned with the forearm. The benefit of this configuration is demonstrated in Section II-B.

To simplify the tracking problem, one constraint is used in our method, i.e., the shoulder joint is assumed fixed during motion, and its position known *a priori*. This is a realistic constraint because the shoulder joint normally moves much less as part of an arm motion. Such a constraint has been widely used in many arm motion tracking systems [28], [29]. In addition, a hand is assumed to be a stiff extension of the lower arm, and the length of both forearm L_2 and the upper arm L_1 are known *a priori*. The tracking task is now reduced to that of tracking elbow and wrist joints in 3-D.

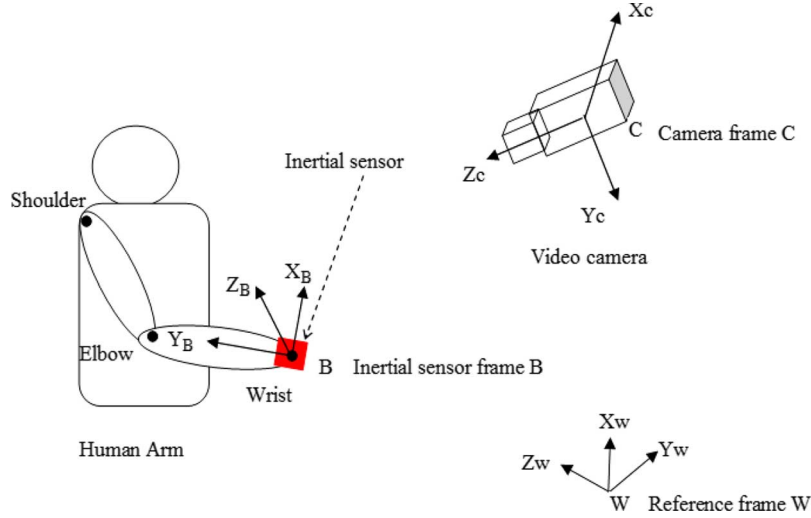


Fig. 1. Arm model and system configuration.

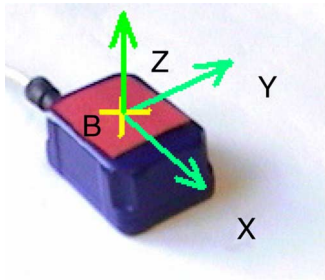


Fig. 2. Inertial sensor (MT9 from XSENS [38]).

B. State Space and Pruning Methods

Based on the above analysis, the posture of the arm can be determined using six variables in Cartesian coordinates at any time k , namely the three coordinates of the elbow position $\mathbf{P}_{e,k} = (x_{e,k}, y_{e,k}, z_{e,k})^T$ and the wrist position $\mathbf{P}_{w,k} = (x_{w,k}, y_{w,k}, z_{w,k})^T$, where subscripts e and w represent the elbow and wrist joints, respectively, and k is a time index. A bold symbol represents a vector or matrix in this paper. Using a state space representation, the system state vector becomes $\mathbf{x}_k = (x_{w,k}, \dot{x}_{w,k}, y_{w,k}, \dot{y}_{w,k}, z_{w,k}, \dot{z}_{w,k}, x_{e,k}, \dot{x}_{e,k}, y_{e,k}, \dot{y}_{e,k}, z_{e,k}, \dot{z}_{e,k})^T$, where $\dot{x}_{w,k}, \dot{y}_{w,k}, \dot{z}_{w,k}$ is the first derivative of $x_{w,k}, y_{w,k}, z_{w,k}$ and represents the velocity of the wrist joint in the x -, y -, and z -directions. The velocity of the elbow joint in the x -, y -, and z -directions is represented as $\dot{x}_{e,k}, \dot{y}_{e,k}, \dot{z}_{e,k}$.

It is always desirable that the size of a state vector should be as small as possible, while capturing all of the essential properties of the system. The computation time for a PF exponentially increases with the number of particles, whereas the number of particles grows with the size of the system state. Therefore, by reducing the size of the state vector, it is possible to reduce the computational cost of the filter. This can also improve the rate at which a search method converges to a global minimum or maximum in a small state space.

Considering the properties of the system configuration discussed in Section II-A and the properties of an MT9 inertial sensor that measures not only acceleration but angular velocity

information as well, we could reduce the size of the state vector in Cartesian coordinates. The main idea is to convert the elbow and wrist joint tracking problem to that of tracking the pose of the wrist joint. The elbow joint is then inferred from wrist joint tracking and arm geometry information. The detailed procedure is as follows.

- 1) Since the origin of an inertial sensor, where a body is attached to frame B , is defined at the wrist joint, and the y -axis is aligned with the forearm, given the length of the forearm L_2 , the elbow joint in the inertial sensor frame B is constant and can be represented as

$$\mathbf{P}_{e,k}^B = (0, L_2, 0)^T \quad (1)$$

where superscript B represents the inertial sensor frame B .

- 2) The position and orientation (pose) of a body attached to the inertial sensor frame B , which is represented in a reference frame W at any time, can be characterized by a 3×3 rotation matrix \mathbf{R}_B^W and a 3×1 translation vector \mathbf{T}_B^W . Given the pose $\mathbf{R}_B^W, \mathbf{T}_B^W$ of the inertial sensor frame B with respect to reference frame W , an elbow joint position can be represented in reference frame W using

$$\mathbf{P}_{e,k} = \mathbf{R}_B^W * \mathbf{P}_{e,k}^B + \mathbf{T}_B^W = \mathbf{R}_B^W * \mathbf{P}_{e,k}^B + \mathbf{P}_{w,k}. \quad (2)$$

It is clear that once the orientation \mathbf{R}_B^W and position \mathbf{T}_B^W of the wrist joint is calculated, it is straightforward to determine an elbow position using (2). Since the origin of the inertial sensor frame B coincides with the wrist joint, \mathbf{T}_B^W has the same meaning as $\mathbf{P}_{w,k}$.

The MT9 inertial sensor contains a proprietary algorithm that can accurately calculate the absolute orientation \mathbf{R}_B^W of a moving sensor B with respect to a reference frame W in a 3-D space (Fig. 3). Our previous work in [15] shows that the orientation information from MT9 is quite accurate and has the potential for use in a rehabilitation application to capture upper limb motion. We will directly use orientation data from the inertial sensor in this paper and use the unit quaternion $\mathbf{Q} = (q_w, q_1, q_2, q_3)$ to represent \mathbf{R}_B^W . Quaternion representation

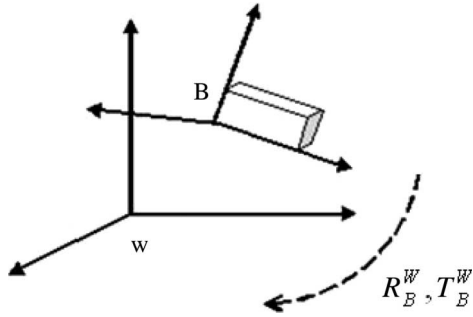


Fig. 3. Coordinate transformation between frame B and frame W .

does not have a Gimbal lock problem and facilitates transformation and calculation in a simple form.

According to the above analysis, the system's state vector can be reduced to track the wrist joint position and is given as follows:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{P}_{w,k} \\ \mathbf{V}_{w,k} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_k \\ \mathbf{V}_k \end{pmatrix} = \begin{pmatrix} x_{w,k} \\ y_{w,k} \\ z_{w,k} \\ \dot{x}_{w,k} \\ \dot{y}_{w,k} \\ \dot{z}_{w,k} \end{pmatrix} \quad (3)$$

where $\mathbf{V}_{w,k} = (\dot{x}_{w,k}, \dot{y}_{w,k}, \dot{z}_{w,k})^T$ is the velocity of the wrist joint. Note that \mathbf{P}_k and \mathbf{V}_k are used in the rest of this paper to represent $\mathbf{P}_{w,k}$ and $\mathbf{V}_{w,k}$, respectively, for simplicity.

In our arm motion tracking method, the inertial sensor is used to obtain the orientation and position of the wrist joint with respect to reference frame W . The visual sensor is used to track a 2-D image projection of the wrist joint. Both tracking results are then input in a fusion estimator PF to recursively estimate the state vector of (3). Elbow position calculation is a straightforward task, which is based on the wrist joint calculation and the geometry information of a human arm, according to (2). Detailed fusion information to estimate a wrist position is introduced in the following sections.

III. DATA FUSION INERTIAL VISUAL TRACKER

A. Dynamical Motion Model—Inertial Data

An inertial sensor MT9 was attached to the wrist joint of a subject. It comprises three accelerometers and three gyros. Therefore, the accelerations \mathbf{a}_k^B and rate of turns $\boldsymbol{\omega}_k^B$ of the wrist joint in the inertial sensor frame B are measurable. However, useful acceleration data are the wrist joint acceleration in reference frame W , which are represented as \mathbf{a}_k . It is calculated from the inertial measurements using

$$\mathbf{a}_k = \mathbf{Q}_k * \mathbf{a}_k^B * \mathbf{Q}_k' - \mathbf{g} \quad (4)$$

where $*$ is the quaternion multiplication. $\mathbf{g} = (0, 0, g)^T$ represents gravity. \mathbf{Q} is a quaternion representation of the rotation transformation between the inertial sensor frame B and reference frame W . The acceleration measured by an inertial sensor \mathbf{a}_k^B includes the acceleration caused by a specific force and the gravity. After transforming the acceleration into frame W

and subtracting the gravity, the \mathbf{a}_k represents the acceleration caused by the specific force only imposed on the target.

A generic motion model—constant acceleration—is assumed in wrist joint position tracking. That is

$$\ddot{\mathbf{x}} = \mathbf{a} + \tilde{\mathbf{f}} \quad (5)$$

where $\tilde{\mathbf{f}}$ is the noise and is modeled as white noise with $E\{\tilde{\mathbf{f}}\} = 0$, $E\{\tilde{\mathbf{f}}\tilde{\mathbf{f}}^T\} = \sigma^2 \mathbf{I}$, i.e., $\tilde{\mathbf{f}}$ is a zero mean random vector, and its autocorrelation matrix is a multiple of the identity matrix \mathbf{I} with the square of deviation σ . The noise is assumed to be constant and the same for each coordinate. After double integration of the constant acceleration motion equation, we can obtain a discrete

$$\mathbf{x}_{k+1} = \begin{pmatrix} \mathbf{P}_{k+1} \\ \mathbf{V}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{I} * T_S \\ 0 & \mathbf{I} \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} T_S^2/2 * \mathbf{I} \\ T_S * \mathbf{I} \end{pmatrix} \mathbf{a}_k + \begin{pmatrix} T_S^3/6 * \mathbf{I} \\ T_S^2 * \mathbf{I} \end{pmatrix} \mathbf{f}_k \quad (6)$$

where T_S is the sensor sampling time interval, and \mathbf{I} is a 3×3 identity matrix. \mathbf{f}_k is the discrete representation of $\tilde{\mathbf{f}}$.

B. Measurement Model—Visual Data

The measurement equation is derived from the output of the visual sensor. A color patch is attached to the inertial sensor (see Fig. 2) and tracked by the camera using the color tracking method Camshift [30]. Tracking results are the 2-D image position $\mathbf{z} = (I_x, I_y, 1)^T$ of the color patch, which also represents the position of the wrist joint. Visual data are related to the state vector as follows:

$$\mathbf{z}_k = \begin{pmatrix} I_{x,k} \\ I_{y,k} \\ 1 \end{pmatrix} \propto \mathbf{K}\mathbf{I} * (\mathbf{R}_W^C * \mathbf{P}_k + \mathbf{T}_W^C) + \mathbf{e}_k \quad (7)$$

where $\mathbf{K}\mathbf{I}$ is a camera intrinsic matrix, \mathbf{e}_k is the measurement noise, and \mathbf{R}_W^C , \mathbf{T}_W^C is the pose of the reference frame W that is represented in the camera frame C . They are both calibrated offline and known *a priori*. Homogeneous coordinates are used in (7) to express an affine transformation. The measurement noise \mathbf{e}_k is a vector of zero mean noise, which is taken to be isotropic with $\sigma = 4$ pixels in this paper. The deviation is calculated by comparing the visual tracking results with ground truth.

C. Data Fusion

Equations (6) and (7), as described, are consistent with a framework proposed by Gustafsson *et al.* [26], for positioning, navigation, and tracking the following:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}_u \mathbf{u}_k + \mathbf{B}_f \mathbf{f}_k \quad (8)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{e}_k \quad (9)$$

where \mathbf{x}_k is a state vector, \mathbf{u}_k is an input, \mathbf{f}_k is a process noise, \mathbf{z}_k is a measurement, and \mathbf{e}_k is the measurement noise. h is a nonlinear function of the state vector \mathbf{x}_k .

Equation (8) is the dynamical motion model of a system, which shows how system state evolves given the previous state and input information. The motion model is formatted to obtain a dynamical pdf $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as follows:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p_{\mathbf{f}_k} \left(B_{\mathbf{f}}^+ (\mathbf{x}_k - A\mathbf{x}_{k-1} - B_{\mathbf{u}}\mathbf{u}_k) \right) \quad (10)$$

where $B_{\mathbf{f}}^+$ is the Moore–Penrose pseudoinverse, and $p_{\mathbf{f}_k}$ is the process noise density. Equation (9) is the measurement model, which represents the relationship between system state \mathbf{x}_k and measurement \mathbf{z}_k . The measurement model is formatted in a probabilistic form as the observation density $p(\mathbf{z}_k|\mathbf{x}_k)$ in the following equation:

$$p(\mathbf{z}_k|\mathbf{x}_k) = p_{\mathbf{e}_k} (\mathbf{z}_k - h(\mathbf{x}_k)) \quad (11)$$

where $p_{\mathbf{e}_k}$ is the measurement noise density.

The task in this section is to estimate the system state vector in (3) based on the dynamical model and measurement model. A Bayesian filter is a general probabilistic framework that has been widely used to estimate the state of a system $\mathbf{x}_k \in R^{N \times}$, by calculating the maximum *a posteriori* $p(\mathbf{x}_k|\mathbf{Z}_k)$ estimate given a sequence of noisy measurements $\mathbf{Z}_k = (\mathbf{z}_0, \dots, \mathbf{z}_k)$ or the minimum mean square error (MMSE) estimate. Assuming $p(\mathbf{x}_0|\mathbf{z}_0) \equiv p(\mathbf{x}_0)$, the posterior pdf $p(\mathbf{x}_k|\mathbf{Z}_k)$ is then recursively obtained using the following two steps: 1) prediction and 2) update.

If we assume that the pdf $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ at time $k-1$ is known, then the prediction step is to use the dynamical model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ to obtain the predicted prior distribution $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ at time k , using the following equation:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}). \quad (12)$$

The likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ is used to update the predicted prior pdf $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ and obtain the posterior pdf $p(\mathbf{x}_k|\mathbf{Z}_k)$ via a direct application of Bayes' rules. Thus, we have

$$p(\mathbf{x}_k|\mathbf{Z}_k) = k_k p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1}) \quad (13)$$

where k_k is a normalization constant.

A PF is a sequential MC simulation implementation of a Bayesian filter. The posterior distribution $p(\mathbf{x}_k|\mathbf{Z}_k)$ is approximated using a set of N weighted particles $\{(\mathbf{x}_k^i, \pi_k^i)\}_{i=1}^N$ and propagated in a prediction-update manner. Starting with a set of particles $\{(\mathbf{x}_{k-1}^i, \pi_{k-1}^i)\}_{i=1}^N$ at time $k-1$, new particles are generated from a proposal distribution, which is the system dynamical model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ in this paper. The sampling stage here, i.e., $\mathbf{x}_k^i \sim p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$, $i = 1 \dots N$, corresponds to the prediction step. The particle's weight is then updated according to

$$\pi_k^i \propto \pi_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)} = \pi_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i) \quad (14)$$

where \propto means the equation is subject to a normalization constant. Updated weights are normalized to sum to 1. A

new particle set $\{(\mathbf{x}_k^i, \pi_k^i)\}_{i=1}^N$ can be obtained after applying resampling on the normalized particle set, and it is a discrete weighted approximation to the true posterior $p(\mathbf{x}_k|\mathbf{Z}_k)$.

D. Layered Sampling PF

In theory, after obtaining dynamical and measurement models, a PF can be used to fuse data and recursively estimate a state vector. However, it was found that the fusion results from the experiments were not accurate in comparison to the ground truth. The PF suffers from a severe particle degeneracy problem. The reason for this is that only one visual feature point from the visual sensor is employed to update particle weights; this is not sufficient. It imposes fewer constraints on the subject's motion and environment but requires a more robust fusion algorithm. The system is only dynamically observable when using 2-D information to deduce 3-D data.

We introduce an arm physical geometry constraint into the fusion system and allow it to serve as one of the measurement sources. This information is consistently available and does not place extra stress on image processing from the visual camera or subject motion. Experimental results show that it improves the performance of the PF. The arm geometry we use is upper-arm length L_1 and forearm length L_2 , which are assumed known *a priori* and constant during motion. This constraint is formulated as (15), where the terms on both sides of the equation are the square of the upper arm length. The following equation serves as a measurement equation to assist color-based measurement:

$$\begin{aligned} L_1^2 &= (\mathbf{P}_{e,k} - \mathbf{P}_{s,k})^2 + w_k \\ &= (\mathbf{R}_B^W * \mathbf{P}_{e,k}^B + \mathbf{P}_k - \mathbf{P}_{s,k})^2 + w_k \end{aligned} \quad (15)$$

where $\mathbf{P}_{s,k}$ is the position of the shoulder joint, which is assumed known and constant. $\mathbf{P}_{e,k}$ is the elbow joint, and \mathbf{P}_k the wrist joint position at time k ; w_k is a noise variable. The noise w_k is assumed as a Gaussian noise with $\sigma = 0.01$ ms, which means the deviation of upper arm length is 1 cm. The following equation is the pdf representation of the noise w_k :

$$p(\mathbf{z}_k|\mathbf{x}_k) = p_{w_k} \left(L_1^2 - (\mathbf{R}_B^W * \mathbf{P}_{e,k}^B + \mathbf{P}_k - \mathbf{P}_{s,k})^2 \right). \quad (16)$$

Since a PF implements in a prediction-update way by fusing a measurement model and a dynamic model at a time, a random walk model was employed as a dynamical model to make up for the prediction-correction form with the measurement equation (15). The dynamical model is represented as

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{r}_k \quad (17)$$

where \mathbf{r}_k is the process noise and is assumed to be a Gaussian vector with isotropic $\sigma = 0.01$ m/s². Equation (17) is the pdf representation of the noise \mathbf{r}_k , i.e.,

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p_{\mathbf{r}_k}(\mathbf{x}_{k+1} - \mathbf{x}_k). \quad (18)$$

The pdf representation of dynamic models (10) and (17) is denoted by $p^1(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p^2(\mathbf{x}_k|\mathbf{x}_{k-1})$, respectively,

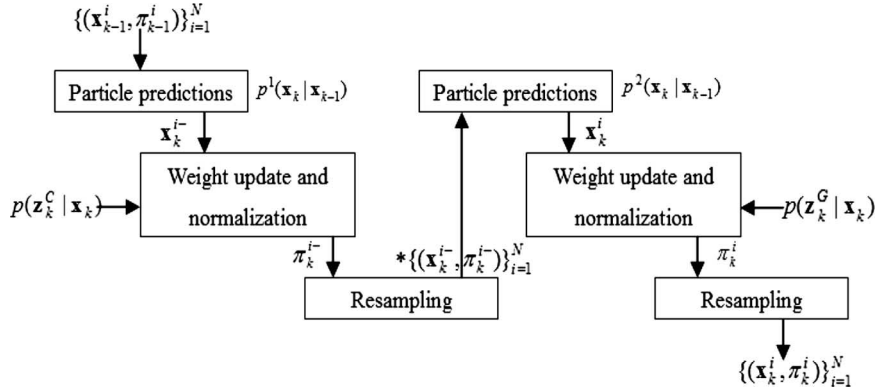


Fig. 4. Block diagram of one cycle operation of a layered sampling particle filter.

whereas $p(\mathbf{z}_k^C | \mathbf{x}_k)$ and $p(\mathbf{z}_k^G | \mathbf{x}_k)$ are the pdf of measurement models (11) and (16). We fuse information using a layered PF, as described in the next section.

E. Implementation

A layered sampling PF implementation is described in this section. The particle set at a previous time $k - 1$ is represented as $\{(\mathbf{x}_{k-1}^i, \pi_{k-1}^i)\}_{i=1}^N$. Particles are propagated at time k as follows.

- 1) *Particle prediction.* Generate new particles $\{\mathbf{x}_k^{i-}\}$ from $p^1(\mathbf{x}_k | \mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i)$, $i = 1 \dots, N$.
- 2) *Weight update.* Compute the weights for each particle via the likelihood of color tracking $\pi_k^{i-} = p(\mathbf{z}_k^C | \mathbf{x}_k = \mathbf{x}_k^{i-})$, and normalize $\tilde{\pi}_k^{i-} = \pi_k^{i-} / \sum_{j=1}^N \pi_k^{j-}$, $i = 1 \dots, N$.
- 3) *Resampling.* Generate a new set $*\{(\mathbf{x}_k^{i-}, \pi_k^{i-})\}_{i=1}^N$ by resampling $*\{(\mathbf{x}_k^{i-}, \pi_k^{i-})\}_{i=1}^N = \text{resampling}\{(\mathbf{x}_k^{i-}, \pi_k^{i-})\}_{i=1}^N$.
- 4) *Particle prediction.* Generate new particles $\{\mathbf{x}_k^i\}$ from $p^2(\mathbf{x}_k | \mathbf{x}_{k-1} = \mathbf{x}_k^{i-})$, $i = 1 \dots, N$.
- 5) *Weight update.* Compute the weights for each particle using the likelihood of arm geometry $\pi_k^i = p(\mathbf{z}_k^G | \mathbf{x}_k = \mathbf{x}_k^i)$, and normalize $\tilde{\pi}_k^{(i)} = \pi_k^i / \sum_{j=1}^N \pi_k^j$, $i = 1 \dots, N$.
- 6) *Resampling.* Generate a new set $*\{(\mathbf{x}_k^i, \pi_k^i)\}_{i=1}^N$ by resampling $*\{(\mathbf{x}_k^i, \pi_k^i)\}_{i=1}^N = \text{resampling}\{(\mathbf{x}_k^i, \pi_k^i)\}_{i=1}^N$.
- 7) Set $k := k + 1$, and iterate to Step 1.

The “resampling” function in Steps 3 and 6 represents the standard residual resampling algorithm [25]. A functional block diagram of the layered sampling PF is shown in Fig. 4. It shows one iteration cycle; data flow is also marked.

It is pointed out in [37] that asymptotically, i.e., as the number of particles goes to infinity, the convergence of a PF can be ensured under very weak assumptions. An efficient sampling method that sample particles in regions of high probability mass can reduce the required number of particles and make the PF implemental. The two strategies introduced in this paper efficiently contribute to sample particles and make the proposed PF convergent using small number particles (500 particles in the experiment). It is a great improvement because many systems use thousands of particles [7], [26].

In our previous research on an EKF [36], an arm geometry constraint was also employed to improve estimation accuracy. However, it was used after EKF estimation and formulated into



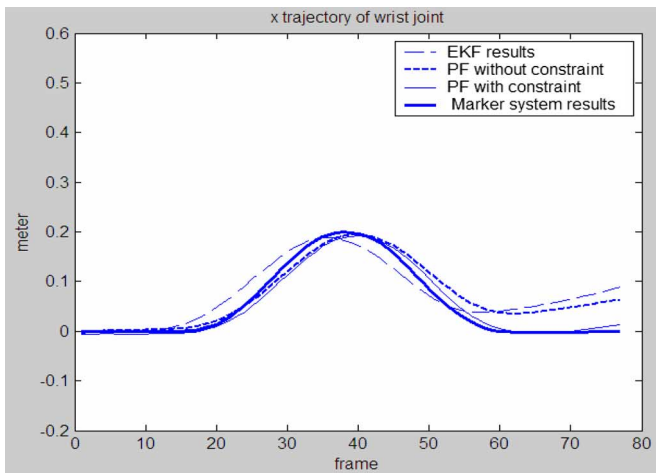
Fig. 5. Subject wearing both an inertial sensor and three ball markers, performing a circle motion.

an optimization framework in which the previous arm motion tracking method consisted of two separate frameworks—EKF and optimization. Here, the arm geometry constraint was used as one measurement and consistently fused into the PF.

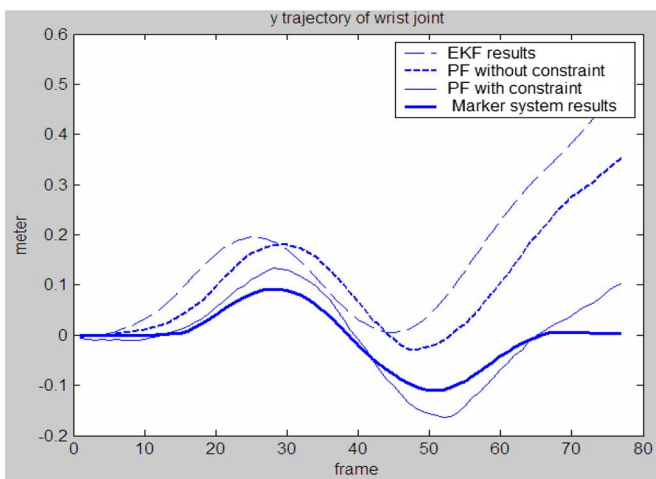
IV. EXPERIMENTAL RESULTS

A. Circle Motion Demonstration

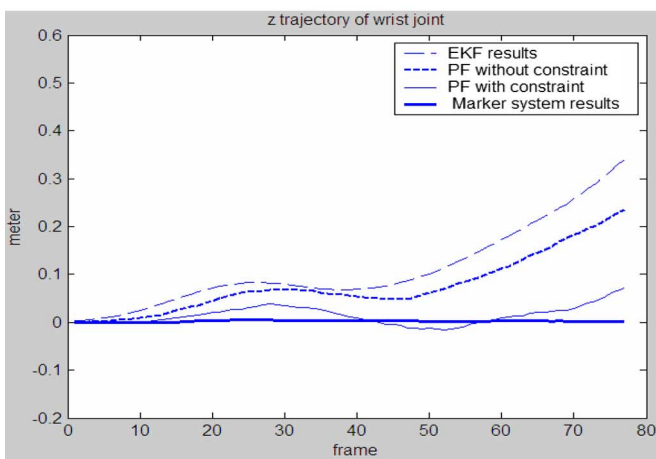
The performance of the proposed method is demonstrated by tracking a circle trajectory on a table (see Fig. 5). In stroke rehabilitation, patients are expected to perform motion patterns such as a circle, rectangle, or target reaching, using their upper limbs in a rehabilitation program [31], [32]. Our experiment setup involved the following two tracking systems: 1) the proposed system and 2) a commercial marker-based system Qualisys. Fig. 5 shows a subject wearing both an inertial sensor and three passive ball markers. The marker-based tracking system with three special cameras was used to track three passive ball markers on the shoulder, elbow, and wrist joints to measure arm motion and provide the ground truth. Our proposed method ran on a personal computer with a Pentium (R) 4/2.8 GHz central processing unit. The number of particles was 500. The processing time was approximately 0.17 s/frame. If the processing ability of computers continues to rapidly increase, a real-time implementation may not be difficult to achieve in the future.



(a)



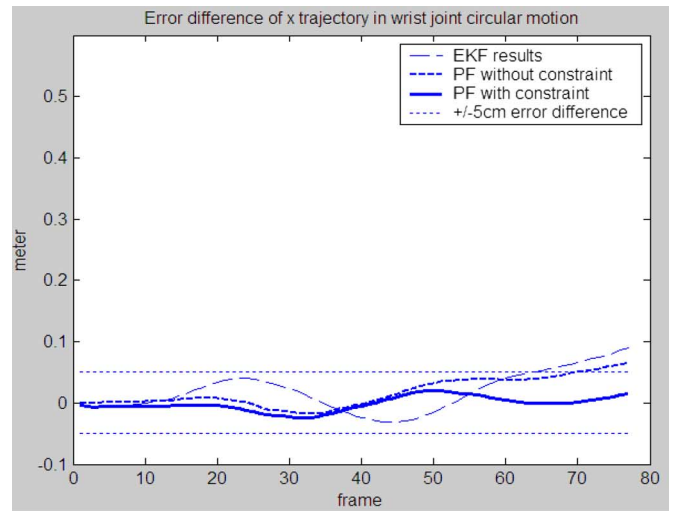
(b)



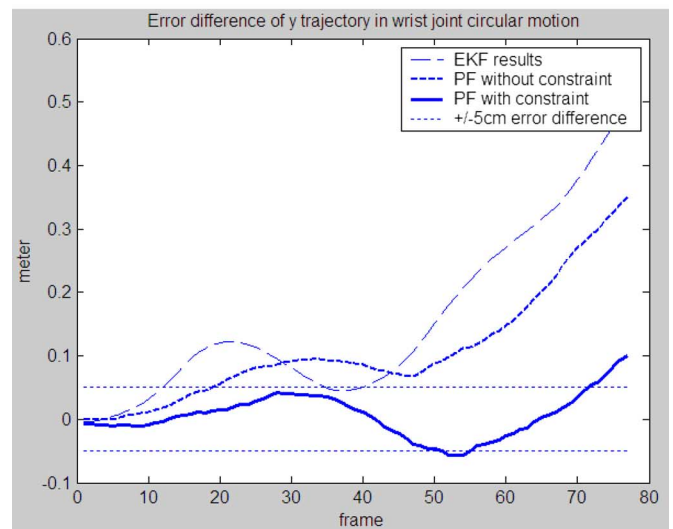
(c)

Fig. 6. Fusion results for a circle motion from an EKF, a PF, a PF with constraint, and a marker system. (a) X-coordinate trajectory of a wrist circle motion. (b) Y-coordinate trajectory of a wrist circle motion. (c) Z-coordinate trajectory of a wrist circle motion.

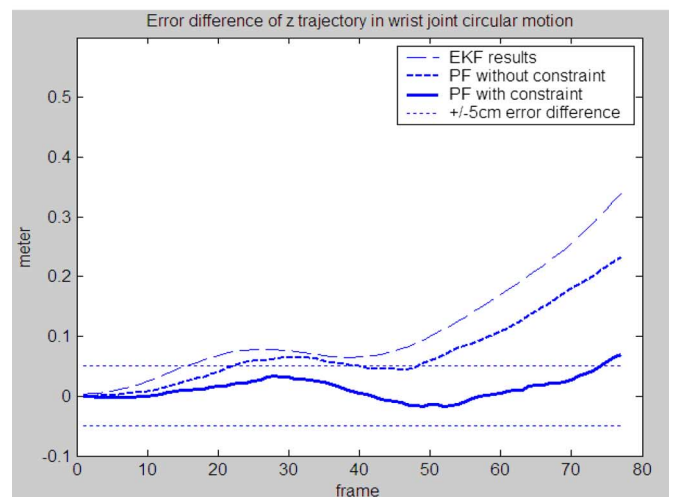
The fusion results from the proposed PF with constraint were compared to the results from PF, EKF, and the ground truth in our experiments to highlight the different performance characteristics of individual fusion algorithms. Fig. 6 illustrates the circle motion trajectories of the wrist joint in X-, Y-, and



(a)



(b)



(c)

Fig. 7. Error difference of a circle motion from different fusion methods, compared to the ground truth. (a) Error difference of circle motion in the X-axis. (b) Error difference of circle motion in the Y-axis. (c) Error difference of circle motion in the Z-axis.

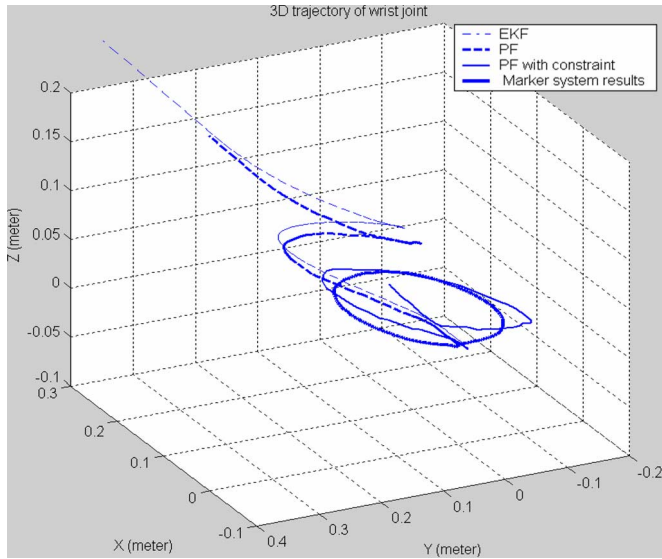


Fig. 8. Three-dimensional reconstruction of a circle motion.

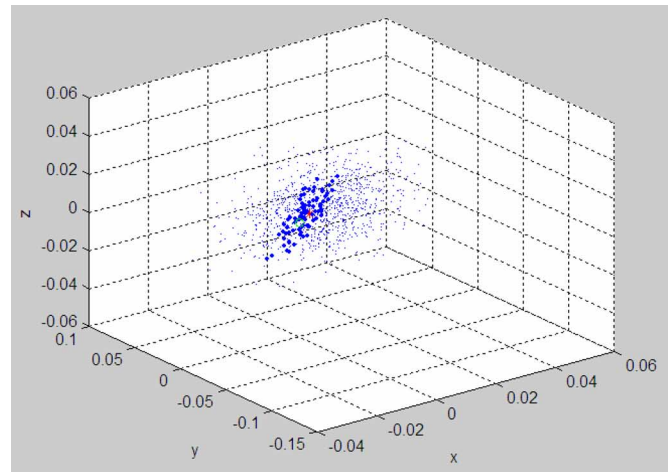
Z-coordinates from each algorithm, respectively. The dashed lines represent results from the EKF, the dotted lines are results from the PF without constraint, the thin solid lines show results from the proposed PF with constraint, and the bold solid lines show the ground truth from the marker-based system. It is clear that the PF with constraint produces better results than the EKF fusion algorithm, particularly in the y - and z -directions.

Fig. 7 shows the error difference between the ground truth and individual results from an EKF, a traditional PF, and the proposed PF. In Fig. 7, the two horizontal lines represent a ± 0.05 -m error range. It is very obvious that the EKF algorithm performs poorly and produces the biggest error difference in all directions (among all the methods). The error differences are unbounded and reach 0.1 m in the x -axis, 0.6 m in the y -axis, and 0.4 m in the z -axis at 80 frames. This is obviously not suitable for a rehabilitation application. The normal PF provides better results than the EKF algorithm; however, the error is still too large, particularly in the y - and z -directions, which have an error difference of up to 0.4 and 0.2 m, respectively. In contrast, the error difference of the proposed PF accuracy is bounded to 0.05 m over 70 frames, as shown by the bold solid lines in Fig. 7(a)–(c).

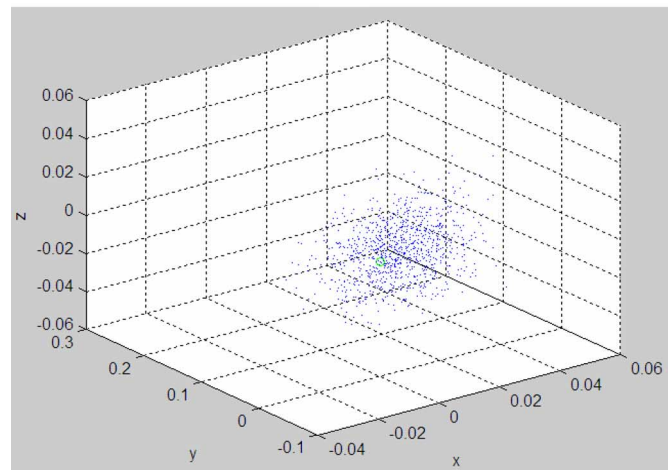
The tracking results for an arm circle motion are represented in three axes in Fig. 6 to more intuitively see the different tracking performance from an EKF, a PF without constraint, and a PF with constraint. Fig. 8 shows the fusion results for a circle motion from different methods, which are reconstructed in 3-D. It is clear that the proposed method provides the best results and is very close to the ground truth. Traditional PFs perform slightly better than the EKF method, but their accuracy is not acceptable.

B. Discussion

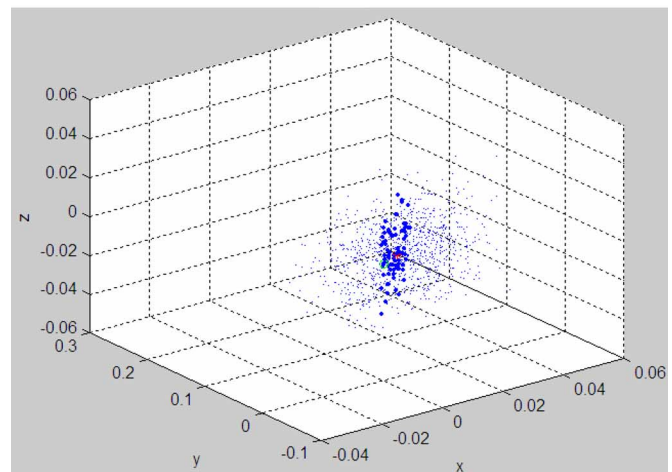
Fig. 9 shows how the particles propagate within one iteration cycle of a PF algorithm for state estimation. The state space is in three dimensions. Each point in each figure represents a particle and illustrates one possible position of the target and its size,



(a)



(b)



(c)

Fig. 9. One iteration cycle of the PF propagation. (a) Particles represent the prior distribution. (b) Particles after prediction. (c) Particles after weight update, representing the posterior distribution.

which is proportional to its weight (the probability). The set of particles in each figure represents a probability distribution. Fig. 9(a) shows the prior distribution. The sets of particles are resampled from this distribution and used to predict new particles, i.e., after applying a system dynamics motion model. All the particles are assigned the same weights in Fig. 9(b). The

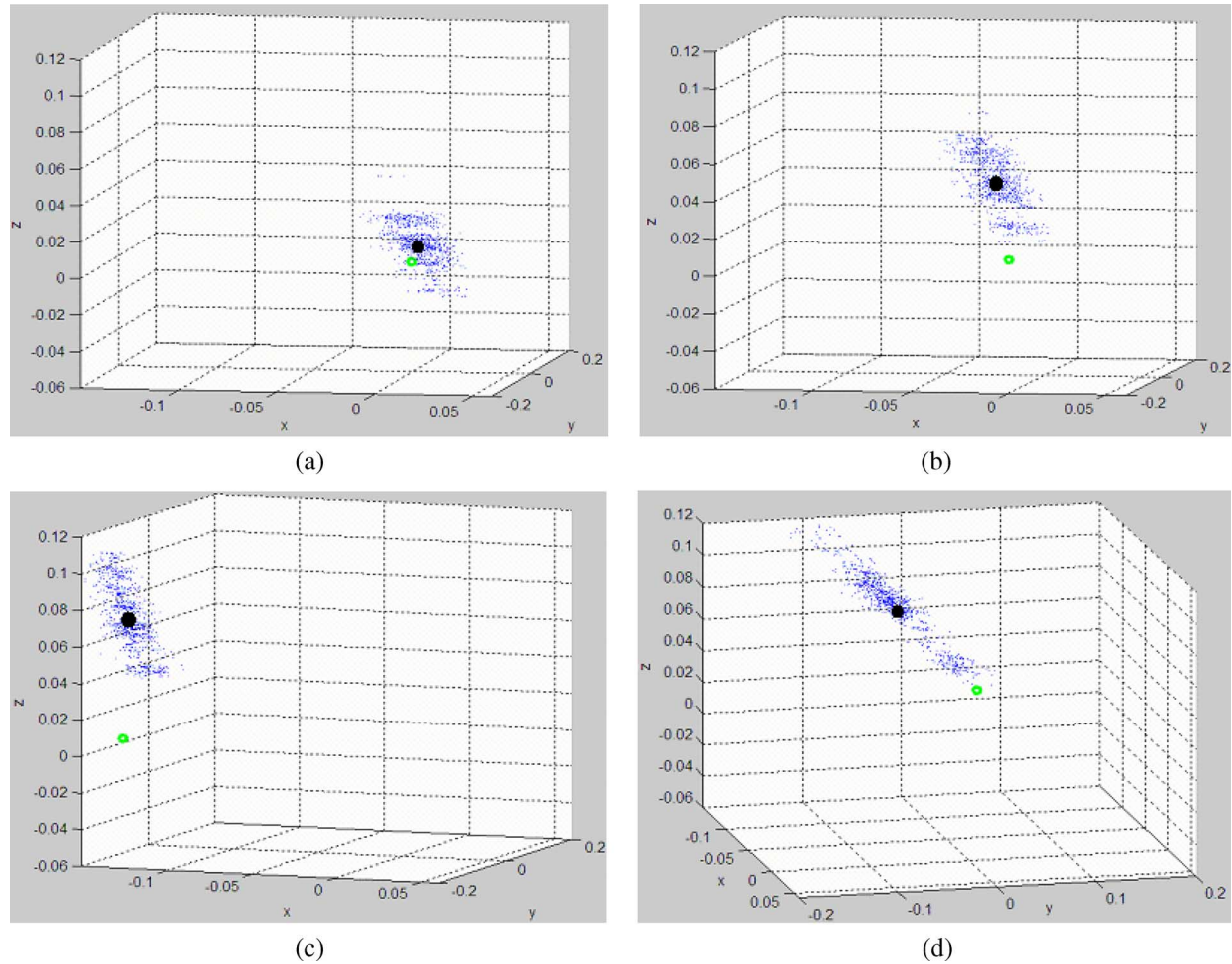


Fig. 10. Fusion results for an arm circle motion using PF without an arm constraint. (a) Frame 10. (b) Frame 20. (c) Frame 30. (d) Frame 40.

weights of predicted particles are updated using the observation model in Fig. 9(c).

The true position of a target is marked as a big size black point, and the estimated position based on the set of particles is marked as a circle in Fig. 9(c). Ideally, the cross should superpose the circle if the estimated state is perfectly accurate. However, in real applications, there is always noise that affects estimation accuracy. The closer the estimated state to the circle, the more accurate the PF. We took some snapshots from an arm circle motion sequence to illustrate the different fusion performance of the PFs (see Fig. 10), with and without an arm geometry constraint (see Fig. 11). Only the posteriors at each time instant are demonstrated. It is clear that the performance of the proposed PF with an arm constraint is better than the conventional PF in the following ways.

- 1) The estimated position of a target from our proposed PF method is more approximate to the true position than that from a conventional PF. As shown in Fig. 10(a), the estimated target position from a conventional PF is very close to the true position at the beginning; however, it continuously diverges in Fig. 10(b)–(d), and the particle sets drift from the ground truth. In contrast, particle sets correctly follow the ground truth when the proposed PF is employed, as shown in Fig. 11; fusion performance slightly degenerates from (a) to (b); however, the tracker

gradually comes back to track the target in Fig. 11(c) and finally coincides with the ground truth in Fig. 11(d). This proves that the proposed particle fusion method has the ability to recover.

- 2) The effective number of particles in the proposed method is larger than a conventional PF. Effective particles are those with large weights, which are shown as large points in the figures. This means that the proposed method suffers less particle degeneracy than a conventional PF.
- 3) The distribution of the proposed PF is more consistent and wide, whereas a conventional PF has patches of particles.

Tracking Results From a Conventional PF Drift: From an analysis on how the PF propagates, it is clear how the proposed PF with an arm physical constraint contributes to a better performance in a circular arm motion tracking (in comparison to a conventional PF). In a conventional PF, the positions of particles are fixed at the prediction stage, as shown in Fig. 9(b), using the previous particle sets $\{(\mathbf{x}_k^i, \pi_k^i)\}_{i=1}^N$ and the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The motion model is built based on an inertial sensor; due to the characteristics of the inertial sensor and the random nature of the motion model, the predicted particle positions are not accurate and drift considerably. Ideally, prediction error can be reduced when measurement information is used to update particle weights in the correction step. As

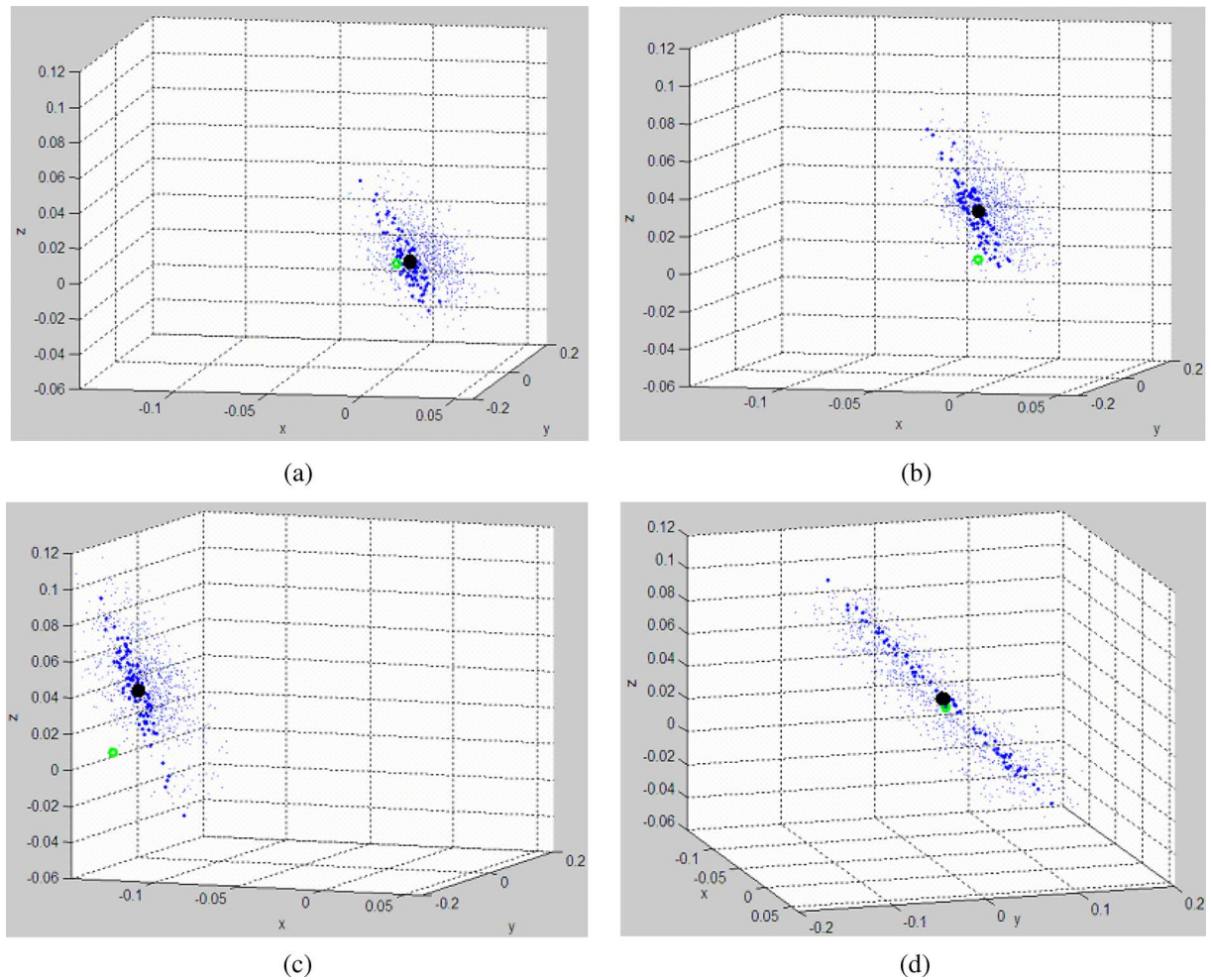


Fig. 11. Fusion results for an arm circle motion using PF with an arm constraint. (a) Frame 10. (b) Frame 20. (c) Frame 30. (d) Frame 40.

particles close to a truth position, they obtain heavy weights, and new particles will be sampled from those that have heavy weights.

However, the measurement information in this application is the 2-D position of the target. It is an underconstrained task to derive the 3-D position of a target from the 2-D image measurements. No unique position can be found. Any point on the backprojected line could be a candidate for a real position. In theory, particle sets will drift along a backprojected line. However, in reality, it can drift in any direction, as error accumulates and becomes unbounded. When particles widely scatter away from truth positions in a space, only a few particles obtain large weights. The number of efficient particles drops sharply, and a conventional PF suffers from severe sample impoverishment. The essential problem is that the information provided by the visual measurements is not sufficient.

Benefit of the Physical Arm Constraint: Therefore, it is necessary to introduce an arm geometry constraint into a PF and fuse information in a layered manner. An arm geometry constraint is consistent during a whole motion sequence. It can correct drift error by supporting visual measurements to locate target positions on a backprojected line. How the arm constraint and visual measurements contribute to the fusion performance can be described in simple terms as a sphere surface and line

intersection. Particle sets are first propagated according to the dynamics of a random walk model, and weights are updated using arm geometry constraint (15), where all particles are distributed on a sphere surface. The sets of particles are then distributed using the dynamic model (12), and weights are then upgraded using visual measurements (14), where particles that lie on the backprojected line of the color object have large weights (high probability). This activity is conducted to find the intersections between a sphere surface and a backprojected line. A heavy weight is assigned to the particles that lie near the intersection of the backprojected line and sphere surface.

The other advantage of using an arm geometry constraint is that particles are widely and evenly distributed in the state space (see Fig. 11), allowing sudden motion to be covered. It is very obvious in Fig. 10 (using the same process noise parameters) that particle distributions are sometimes patchy when an arm constraint is not employed. To cover a large state space, process noise in the motion model should be artificially high.

C. Statistical Results

To further evaluate the proposed fusion method, we calculated the statistical performance of the algorithm. The statistical properties of the mean and standard deviation for each method

TABLE I
MEAN ERROR AND STANDARD DEVIATION OF EACH AXIS FOR A
CIRCLE MOTION WHEN USING DIFFERENT ALGORITHMS

Circle motion	X coordinate error (cm)	Y coordinate error (cm)	Z coordinate error (cm)
Inertial only	21.31/37.70	-12.41/16	6.74/5.23
EKF	0.4/3.30	10.56/9.39	7.83/5.72
PF	1.8/2.67	10.08/6.55	6.84/4.15
PF with constraint	-0.68/2.0	1.14/5.47	1.77/3.27

were calculated for our experiments. A number of identical motion patterns were captured and compared with the marker-based tracking results. We calculated the mean error and standard deviation of each motion axis for a circle. Table I shows the statistical properties of the different algorithms. It is clear that the “inertial only” tracking method suffers severe drift error. Error mean and deviation is as great as 21.31/37.70 cm in the X -coordinate. The EKF greatly improved the tracking results by fusing inertial data with visual data; however, its accuracy was not good enough. The PF provided slightly better performance than the EKF method. The PF with an arm constraint gave the best performance of all algorithms. Tracking errors were mainly in a range of ± 5 cm when compared to the ground truth.

V. CONCLUSION AND FUTURE WORK

This paper presents a novel arm motion tracking system for telerehabilitation applications, which is based on the integration of both vision and inertial sensors. Data fusion is implemented using PFs. We tackle the PF degeneracy problem by proposing a state-space pruning strategy and by introducing an arm physical geometry constraint. This results in a great reduction of computational cost. The efficiency of the proposed method is analyzed and illustrated graphically. Experimental results show that it outperforms other fusion methods such as EKF and PF under the same circumstances. Although inertial and visual sensors have been widely used to track rigid objects, our proposed system is able to track nonrigid human arm motion in 3-D.

To make the system more applicable and robust, our future work will be focussed in three directions. First, a more efficient noise compression method will be developed to reduce the noise caused by the deformable human body in which inertial sensors are attached. Second, a wireless sensor will be adopted to replace the wired inertial sensor in order for a subject to move freely. Third, the fixed shoulder constraint will be released, and the tracking method will be extended to track the upper body or whole body. More image features will be used, including edge, contour, and suitable subject geometry and kinematics models.

ACKNOWLEDGMENT

The authors would like to thank Dr. M. H. Sellens of the Biological Science Department, University of Essex, for

allowing them to use their Qualisys motion tracking system and Dr. H. Zhou of Queen Mary College, University of London, for useful discussions.

REFERENCES

- [1] L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis,” *J. Pattern Recognit.*, vol. 36, no. 3, pp. 585–601, Mar. 2003.
- [2] T. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *J. Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 231–268, Mar. 2001.
- [3] D. M. Gavrilu, “The visual analysis of human movement: A survey,” *J. Comput. Vis. Image Underst.*, vol. 73, no. 1, pp. 82–98, Jan. 1999.
- [4] J. K. Aggarwal and Q. Cai, “Human motion analysis: A review,” *J. Comput. Vis. Image Underst.*, vol. 73, no. 3, pp. 428–440, Mar. 1999.
- [5] C. Sminchisescu, “Estimation algorithms for ambiguous visual models 3D human modelling and motion reconstruction in monocular video sequences,” Ph.D. dissertation, Inst. Nat. Politechnique de Grenoble (INRIA), Grenoble, France, Jul. 2002.
- [6] C. Sminchisescu and B. Triggs, “Covariance scaled sampling for monocular 3D body tracking,” in *Proc. Int. Conf. Comput. Vis. Pattern Recog.*, Dec. 9–14, 2001, pp. 447–454.
- [7] H. Sidenbladh, M. Black, and D. Fleet, “Stochastic tracking of 3D human figures using 2D image motion,” in *Proc. Eur. Conf. Comput. Vis.*, D. Vernon, Ed. Dublin, Ireland: Springer-Verlag, Jun. 2000, vol. 1843, pp. 702–718.
- [8] P. Prez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particles,” *Proc. IEEE*, vol. 92, no. 3, pp. 495–513, Mar. 2004.
- [9] S. C. Thomopoulos and L. Nilsson, “3-D motion tracking using stereo camera and range radar,” in *Proc. SPIE—Sensing Reconstruction Three-Dimensional Objects Scenes*, B. Girod, Ed., Jan. 1990, vol. 1260, pp. 21–35.
- [10] D. Strelow and S. Singh, “Motion estimation from image and inertial measurements,” *Int. J. Robot. Res.*, vol. 23, no. 12, pp. 1157–1195, Dec. 1, 2004.
- [11] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington, “FlightTracker: A novel optical/inertial tracker for cockpit enhanced vision,” in *Proc. IEEE/ACM ISMAR*, Washington, DC, Nov. 2–5, 2004, pp. 212–221.
- [12] L. Chai, W. A. Hoff, and T. Vincent, “Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality,” *Presence: Teleoperators Virtual Environ.*, vol. 11, no. 5, pp. 474–492, Oct. 2002.
- [13] P. Lang, M. Ribo, and A. Pinz, “A new combination of vision-based and inertial tracking for fully mobile, wearable and real-time operation,” in *Proc. 26th Workshop GM/APR*, Graz, Austria, Sep. 2002, vol. 160, pp. 141–148.
- [14] S. You and U. Neumann, “Fusion of vision and gyro tracking for robust augmented reality registration,” in *Proc. IEEE Virtual Reality*, Yokohama, Japan, Mar. 2001, pp. 71–78.
- [15] Y. Tao, H. Hu, and H. Zhou, “Integration of vision and inertial sensors for home-based rehabilitation,” in *Proc. IEEE 2nd Workshop InerVis ICRA*, Apr. 18, 2005.
- [16] H. Zhou and H. Hu, “Inertial motion tracking of human arm movements in stroke rehabilitation,” in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Niagara Falls, ON, Canada, Jul. 29–Aug. 1, 2005, pp. 1306–1311.
- [17] M. Isard and A. Blake, “CONDENSATION—Conditional density propagation for visual tracking,” *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, Aug. 1998.
- [18] M. Isard and A. Blake, “CONDENSATION: Unifying low-level and high-level tracking in a stochastic framework,” in *Proc. Eur. Conf. Comput. Vis.*, 1998, pp. 893–908.
- [19] T. B. Moeslund and E. Granum, “Bootstrapping sequential Monte Carlo tracking,” in *Proc. Scandinavian Conf. Image Anal.*, Goteborg, Sweden, Jun. 2003, pp. 1030–1037.
- [20] T. B. Moeslund and E. Granum, “Sequential Monte Carlo tracking of body parameters in a sub-space,” in *Proc. Int. Workshop Anal. Model. Faces Gestures*, Nice, France, Oct. 2003, pp. 84–91.
- [21] S. Zhou, V. Krueger, and R. Chellappa, “Face recognition from video: A CONDENSATION approach,” in *Proc. Int. Conf. Autom. Face Gesture Recog.*, Washington, DC, May 20–21, 2002, pp. 212–217.
- [22] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [23] R. Douc, O. Cappé, and E. Moulines, “Comparison of resampling schemes for particle filtering,” in *Proc. 4th Int. Symp. Image Signal Process. Anal.*, 2005, pp. 64–69.

- [24] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. Int. Conf. Comput. Vis. Pattern Recogn.*, Hilton Head Island, SC, Jun. 13–15, 2000, pp. 126–133.
- [25] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," Dept. Eng., Cambridge Univ., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR. 310, 1998.
- [26] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.
- [27] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Electr. Eng. F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [28] L. Goncalves, E. Di Bernardo, E. Ursella, and P. Perona, "Monocular tracking of the human arm in 3D," in *Proc. ICCV*, 1995, pp. 764–770.
- [29] T. B. Moeslund and E. Granum, "3D human pose estimation using 2D-data and an alternative phase space representation," in *Proc. Workshop Human Model.*, *Anal. Synthesis CVPR*, 2000, pp. 26–33.
- [30] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technol. J.*, vol. 2, 1998.
- [31] *GENTLE/S: Robotic assistance in neuro and motor rehabilitation*, Mar. 2006. [Online]. Available: <http://www.gentle.rdg.ac.uk/>
- [32] Rehabworks, Jan. 2007. [Online]. Available: <http://rehabworks.ksc.nasa.gov/education/protocols/basicshoulder.php>
- [33] P. D. Moral and L. Miclo, "Branching and interacting particle systems approximations of Feynman–Kac formulae with applications to non-linear filtering," in *Séminaire de Probabilités XXXIV*, vol. 1729, J. Azéma, M. Emery, M. Ledoux, and M. Yor, Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 1–145.
- [34] P. D. Moral and P. M. Ledoux, "On the convergence and the applications of empirical processes for interacting particle systems and nonlinear filtering," *J. Theor. Probab.*, vol. 13, no. 1, pp. 225–257, 2000.
- [35] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, Mar. 2002.
- [36] Y. Tao and H. Hu, "3D arm motion tracking for home-based rehabilitation," in *Proc. 3rd CWUAAT*, Cambridge, U.K., Apr. 10–12, 2006.
- [37] Sequential Monte Carlo Methods Homepage, Jul. 2007. [Online]. Available: <http://www.sigproc.eng.cam.ac.uk/smc/>
- [38] XSENS. [Online]. Available: <http://www.xsens.com/>



Huosheng Hu (M'94–SM'01) received the M.Sc. degree in industrial automation from the Central South University, Changsha, China, in 1982 and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor with the Department of Computing and Electronic Systems, University of Essex, Colchester, U.K., where he leads the Human Centred Robotics Group. He has published more than 200 papers in journals, books, and conferences and has received some best paper awards. From 1997

to 2000, he was a member of the Editorial Advisory Board for the *International Journal of Industrial Robots*. He is currently the Editor-in-Chief for the *International Journal of Automation and Computing*. Since 2000, he has been a Visiting Professor at six universities in China, namely Central South University; Shanghai University, Shanghai; Wuhan University of Science and Engineering, Wuhan; Kunming University of Science and Technology, Kunming; Chongqing University of Post and Telecommunication, Chongqing; and Northeast Normal University, Changchun. He is a reviewer for a number of international journals such as *Automatic Control*, *Neural Networks*, and the *International Journal of Robotics Research*. His research interests include autonomous mobile robots, human–robot interaction, evolutionary robotics, multirobot collaboration, embedded systems, pervasive computing, sensor integration, RoboCup, intelligent control, and networked robotics.

Dr. Hu is a founding member of Networked Robots of the IEEE Society of Robotics and Automation Technical Committee. He was a member of the International Association of Science and Technology for Development (IASTED) Technical Committee on "Robotics" for 2001–2004. He was a General Cochair of the 2007 IEEE International Conference on Mechatronics and Automation (ICMA07), Harbin, China; a Publication Chair of the 2007 IEEE International Conference on Networking, Sensing, and Control (ICNSC07), London, U.K.; a Cochair of Special and Organized Sessions of the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO 2007), Sanya, China; and a Cochair of the International Program Committee, First International Conference on Human Body Simulation and Modelling, Shanghai University, October 28–30, 2004. He is a Chartered Engineer and a member of the Institution of Engineering and Technology, the Association for the Advancement of Artificial Intelligence, the Association for Computing Machinery, IASTED, and IAS.



Yaqin Tao received the B.A. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2002. She is currently working toward the Ph.D. degree in robotics with the Department of Computing and Electronic Systems, University of Essex, Colchester, U.K.

Her research interests include image processing, visual tracking of human motion, and data fusion. She has published a number of journal articles and conference papers in these areas.