Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 1958–1970
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Real-time hand tracking using a mean shift embedded particle filter

Caifeng Shan[a,*], Tieniu Tan[b], Yucheng Wei[b]

[a]Department of Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK
[b]National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing 100080, China

Received 4 May 2006; received in revised form 6 December 2006; accepted 10 December 2006

Abstract

Particle filtering and mean shift (MS) are two successful approaches to visual tracking. Both have their respective strengths and weaknesses. In this paper, we propose to integrate advantages of the two approaches for improved tracking. By incorporating the MS optimization into particle filtering to move particles to local peaks in the likelihood, the proposed mean shift embedded particle filter (MSEPF) improves the sampling efficiency considerably. Our work is conducted in the context of developing a hand control interface for a robotic wheelchair. We realize real-time hand tracking in dynamic environments of the wheelchair using MSEPF. Extensive experimental results demonstrate that MSEPF outperforms the MS tracker and the conventional particle filter in hand tracking. Our approach produces reliable tracking while effectively handling rapid motion and distraction with roughly 85% fewer particles. We also present a simple method for dynamic gesture recognition. The hand control interface based on the proposed algorithms works well in dynamic environments of the wheelchair.

Keywords: Hand tracking; Particle filter; Mean shift; Hand gesture recognition; Human–computer interaction

1. Introduction

As computers become pervasive in our daily life, people are in great need of natural and efficient human–computer interaction (HCI). Currently the most popular and dominant mode of HCI is still based on devices such as keyboard and mouse, which is inconvenient and unnatural for human beings, and also limits the speed of interaction. The more intuitive and friendly interaction can be achieved if computers could capture and understand human motion. Visual analysis of human motion would bring a revolution to HCI [1].

Hand gesture is the meaningful or intentional movements of human hands and arms. As a universal body language of mankind, hand gesture is one of the most natural and effective means for humans to communicate non-verbally with others. The ability to recognize hand gestures is indispensable and important for successful interpersonal social interaction. Vision-based hand gesture recognition [2], enabling computers to understand hand gestures as humans do, is an important technology for intelligent HCI. Therefore, visual analysis of hand gestures has attracted much attention in the last decade, and much progress has been made [2].

Our research goal is to design and implement a realistic real-time control interface for a robotic wheelchair based on hand gesture recognition. The block diagram of our work is shown in Fig. 1. As an important step, this paper mainly focuses on real-time hand tracking in dynamic environments of the moving wheelchair.

1.1. Related work

Hand tracking, aiming to estimate continuous hand motion in image sequences, is a challenging but essential step for hand gesture recognition. Due to its critical role in vision problems involving hand motion, hand tracking has gained wide interest in recent years [3–17]. Here we briefly review some previous work on hand tracking in order to put our work in context.

Hand motion is performed in the 3D space, and captured in 2D image sequences. So hand tracking can be carried out in the 3D space or in the 2D image plane. The existing hand tracking approaches can be mainly classified into 2D methods and 3D methods. In 2D methods, a hand is represented

* Corresponding author. Tel.: +44 20 7882 8018; fax: +44 20 8980 6533.
E-mail addresses: cfshan@dcs.qmul.ac.uk (C. Shan), tnt@nlpr.ia.ac.cn (T. Tan), weiyucheng@gmail.com (Y. Wei).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.12.012

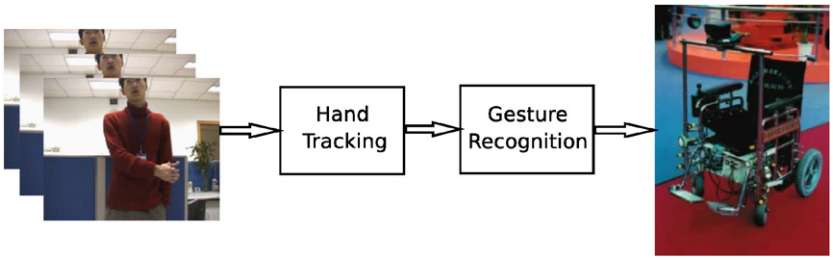

Fig. 1. Block diagram of the developed hand control interface for the wheelchair.

by its geometric features such as contours [3,18] and fingertips [4], or its non-geometric features such as color [5] and texture. Isard and Blake [18] adopted parameterized B-spline curves to model hand contours, and tracked hands by tracking the deformed curves. However, since hand contours are view-dependent and vary dramatically in natural hand motion, the contour-based trackers usually constrain the viewpoint and assume that hands keep several predefined shapes. For example, McAllister et al. [3] developed a contour-based bimanual tracking system in which the viewpoint is constrained. Fingertips are another effective geometric features for hand tracking, for instance, Oka et al. [4] proposed to track multiple fingertips for hand gesture recognition. Skin color is a distinctive feature of hands if no other exposed skin body parts or skin-colored objects in background, so many color-based trackers were utilized to track hand motion [5,19]. Originating from the idea that different types of image features are visible depending on the scale of observation, Laptev and Lindeberg [5] built a hierarchical hand model based on multi-scale image features for hand tracking.

Every single type of features has its limitations, for example, skin color is not reliable due to varying illumination. In order to overcome weaknesses of each individual feature, many algorithms were proposed to integrate multiple cues for robust hand tracking [6,12,13,20,21]. Isard and Blake [20] described an approach to augment the contour tracker by skin-colored blob tracking for hand tracking. In Ref. [6], Huang and Reid developed a joint Bayes filter to combine particle filter (PF) and hidden Markov model (HMM) for tracking and recognition of the articulated hand motion. Color and shape were utilized as hand representation, and particle filtering was adopted for color-region tracking to assist HMM in analyzing hand shape variations. Kolsch and Turk [12] recently proposed to integrate optical flow and color cues for fast hand tracking with flocks of features.

The 2D hand tracking is computationally efficient for real-time applications. Hence many existing applications are based on 2D approaches [22]. However, 2D tracking usually can only track global hand motion, and cannot determine articulated motion of fingers. In contrast, 3D hand tracking can locate hands in 3D space and provide 3D position and orientation information explicitly.

Recently 3D model-based tracking has been an active and growing research area [7,11,15,17,23]. It has the ability to cope with occlusion and self-occlusion, and can obtain detailed and accurate motion data that can be used in many applications. Model-based tracking usually estimates the state of a hand by projecting the pre-stored 3D hand model to the image plane and comparing it with image features. Lu et al. [7] presented a model-based approach to integrate multiple cues such as edges, optical flow and shading information, for articulated hand motion tracking. A forward recursive dynamic model was adopted to track the articulated motion in response to data derived 3D forces. The generalized force was derived to adjust the model shape at each step. Sudderth et al. [11] adopted non-parametric belief propagation for visual tracking of a geometric hand model. Chang et al. [15] recently proposed a model-based tracking method, which integrates both sequential motion transition information and appearance information. In Ref. [8], the concept of eigen-dynamics was proposed to model the dynamics of natural hand motion. Hand motion was modeled as a high-order stochastic linear dynamic system (LDS) consisting of five low-order LDSs, each of which corresponds to one eigen-dynamics. The 3D tracking can also base on 3D data obtained by stereo cameras or scanners [9]. More recently Inaguma et al. [14] proposed to set a 3D search volume for efficient palm tracking using two cameras. Although providing more accurate results than 2D tracking, 3D methods usually suffer from high computational cost due to the articulated hand motion and the large number of degrees of freedom. Thus, 3D tracking is seldom feasible in real-time applications.

1.2. Overview of our work

In this paper, we present an effective approach to real-time hand tracking in dynamic environments of the robotic wheelchair. Considering simplicity and practical feasibility, we perform hand tracking in the 2D image plane. Although many methods have been developed, most of them are designed for a specific problem, and are based on some assumptions (e.g., constrained viewpoint, static background, clutter-free environments). If applied directly to natural hand gesture tracking in the moving wheelchair, they faces two major difficulties: (1) the background is dynamically changing and cluttered due to the wheelchair's free movements in indoor or outdoor environments, and illumination also varies considerably. So hand tracking is required to be robust enough in the realistic situation. (2) The control interface is required to work in real time. Moreover, the wheelchair performs other tasks such as self-localization and obstacle avoidance when interacting with users, so hand tracking must be efficient and low cost.

In this work, we address these difficulties in the context of developing the hand control interface for the wheelchair. The presence of background clutter, complex dynamics of hand motion, and varying illumination result in that hand tracking is a typical non-linear and non-Gaussian problem. Particle filtering [24] (also known as the CONDENSATION algorithm [18]) provides a robust Bayesian framework for this problem, thanks to its ability to handle multi-modal problems by maintaining multiple hypotheses. So particle filtering is adopted as the tracking framework in this paper. However, for PFs, to guarantee their robustness, sampling must be sufficient to capture the variations in the state space. Dense sampling of PFs brings high computation load, so conflicts with the low cost demand of the hand control interface. One has to devise tracking schemes that succeed with fewer particles. Some techniques have been presented to improve the sampling efficiency of particle filtering [20,25–31]. One can reduce the number of particles by choosing a better proposal distribution [25]. Importance sampling [20] was introduced to obtain better proposal by combining prediction based on previous configuration with additional knowledge from auxiliary measurements. For tracking more than one object, partitioned sampling [26] was presented to reduce the number of particles needed; however, this approach is based on an assumption that the configuration space can be sliced in partitions. Another way to obtain better sampling is to introduce optimization procedures that move particles to peaks of the density [27,28]. However, during optimization the particles do not follow the posterior distribution anymore, so these methods do not produce an approximation to the desired posterior [29,30].

By incorporating an efficient non-parametric optimization into particle filtering, we present a novel method to improve the sampling efficiency of PFs. Mean shift (MS) is a gradient-based optimization method, and has been successfully used for real-time tracking [32]. However, MS trackers cannot guarantee global optimality, and easily fall into local maxima. We propose to embed the MS optimization into particle filtering for better sampling, to combine advantages of the two tracking approaches. The mean shift embedded particle filter (MSEPF) allows to use much fewer particles to maintain multi-modes. We realize real-time reliable hand tracking in the wheelchair using the proposed MSEPF. The proposed tracking method in fact is a general approach, and can be applied to tracking problems. Following real-time hand tracking, we present a simple method for dynamic gesture recognition. Online gesture recognition works well in the wheelchair, which illustrates the superior performance of the presented algorithms in dynamic environments.

This paper is an extended version of our previous work described in Ref. [33]. The major modification lies in the formulation and justification of the improved tracking scheme, more experiments, and hand gesture recognition. The main contributions of this paper are summarized as follows:

- By incorporating the MS optimization into particle filtering to move particles towards local modes, we develop the MSEPF algorithm for efficient visual tracking, which decreases the required particles greatly.
- We realize real-time hand tracking in the wheelchair using MSEPF. By fusing color and motion cues, an effective observation model is presented for the MS iteration and likelihood function.
- We introduce a simple method for dynamic gesture recognition. The hand control interface based on the proposed algorithms works well in dynamic environments of the wheelchair.

The remainder of this paper is organized as follows. Section 2 describes the proposed MSEPF algorithm. Section 3 discusses real-time hand tracking using MSEPF, respectively, detailing dynamical model, observation model, MS analysis, and likelihood function. Hand gesture recognition is then presented in Section 4. Experimental results are discussed in Section 5, and Section 6 concludes the paper.

2. The MSEPF

2.1. Particle filtering

Tracking objects efficiently and robustly in complex environments is a challenging issue in computer vision. Visual tracking can be regarded as the estimation of the system state that changes over time using a sequence of noisy measurements made on the system. The goal of Bayesian tracking is to recursively compute the posterior density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ of the current object state $\mathbf{x}_t$ conditioned on all observations $\mathbf{z}_{1:t} = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$, up to time t. The probability density function (pdf) $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ can be obtained recursively in two stages: prediction and update. If the time-varying state $\mathbf{x}_t$ is modeled as a first-order Markov process, the pdf $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is derived as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \kappa p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1}), \tag{1}$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\, \mathrm{d}x_{t-1}, \tag{2}$$

where κ is a normalizing constant that is independent of $\mathbf{x}_t$, $p(\mathbf{z}_t|\mathbf{x}_t)$ is the likelihood function, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the dynamic model, and $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ is the temporal prior over $\mathbf{x}_t$ given past observation. $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{z}_t|\mathbf{x}_t)$ are not required to be Gaussian here.

Particle filtering [18,24] is a technique for implementing a recursive Bayesian filter by Monte Carlo simulations. In particle filtering, the required posterior density is approximated by a weighted particle set $\{(s_t^{(n)}, \pi_t^{(n)})\}_{n=1}^N$ at each time t. Each particle $s_t^{(n)}$ represents one hypothetical state of the object, and is weighted by a discrete sampling probability $\pi_t^{(n)} = p(\mathbf{z}_t|\mathbf{x}_t = s_t^{(n)})$, the probability of that the current observations were generated by the hypothetical state. The state at each time t can finally be estimated based on these particles and weights. The particle set is propagated according to the system dynamic model over time. We summarize the particle filtering algorithm in Fig. 2. PFs can cope with non-linear dynamics and non-linear observations, by maintaining multiple hypotheses. Managing multi-modal density allows PFs to handle clutter and recover from failures in visual tracking. Due to its robustness,

Given the particle set at $t-1$, perform the following steps:
(1) **Resample** N particles from the set $\{(s_{t-1}^{(n)}, \pi_{t-1}^{(n)})\}_{n=1}^{N}$ to give $\{(s_{t-1}^{'(n)}, \frac{1}{N})\}_{n=1}^{N}$.
(2) **Propagate** each particle by the dynamic model $s_t^{(n)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1} = s_{t-1}^{'(n)})$ to give $\{(s_t^{(n)}, \frac{1}{N})\}_{n=1}^{N}$.
(3) **Weight** the particles as $\pi_t^{(n)} \propto p(\mathbf{z}_t|\mathbf{x}_t = s_t^{(n)})$ to give $\{(s_t^{(n)}, \pi_t^{(n)})\}_{n=1}^{N}$, where $\pi_t^{(n)}$ are normalized so that $\sum_{n=1}^{N} \pi_t^{(n)} = 1$.
(4) **Estimate** the tracking result $E[\mathbf{x}_t] = \sum_{n=1}^{N} \pi_t^{(n)} s_t^{(n)}$.

Fig. 2. The particle filtering algorithm.

particle filtering has attracted significant attention [20,34,35] in computer vision community. PFs were originally proposed to utilize edge-based image features [18,20]; recently color-based PFs were introduced [34,35].

The success of particle filtering depends on its ability to maintain a good approximation to the posterior. To capture the variations in state-space, a certain number of particles are required to guarantee sufficient sampling, and this number increases exponentially with the state dimension. High computation cost caused by a large number of particles usually makes PFs infeasible for real-time applications.

2.2. Mean shift

MS tracking algorithms are another type of successful approaches taken in visual tracking, and have recently become popular due to their simplicity and effectiveness [32,36–38]. In MS trackers, a color histogram is usually used to describe the target region [32]. The Bhattacharyya coefficient or other similarity measures is employed to measure the similarity between the template (or model) region and the current target region. Tracking is accomplished by iteratively finding the local minima of the distance measure functions.

As an efficient gradient-based optimization method, MS has fast convergence speed and low computation cost. Moreover, MS is a non-parametric method, and provides a general optimization solution independently from target features. However, MS cannot guarantee global optimality, and is susceptible to fall into local maxima, in case of clutter or occlusion. As demonstrated in Ref. [35], MS trackers easily fail in tracking rapid moving objects, and cannot recover from failures. Therefore, their efficiency is traded off against robustness. Kalman filter and its extensions can be used to improve their robustness [37]. However, as they are based on a deterministic algorithm, MS trackers cannot deal with multi-modal problems.

2.3. The MSEPF

Particle filtering and MS both have their respective strengths and weaknesses. In this work, we propose to combine advantages of both approaches for improved tracking. By incorporating the MS optimization into particle filtering, we present a simple and efficient method to improve the sampling efficiency of PFs. Compared to the conventional particle filtering, the number of particles required is reduced greatly. The proposed method is similar to Ref. [23] in the sense that particles are moved to the local maxima in the likelihood by a local optimization. However, we adopt a computationally more efficient non-parametric approach by introducing the MS iteration.

After propagating particles via the dynamic model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, the MS optimization is run for each of the particles. Particles are moved in the gradient ascent direction in the likelihood until they converge to their neighboring local peaks. The resulting particle set maintains fair representation of the modes of the distribution, and provides good local characterization of the likelihood. As the local maxima are always represented well by the particle set, much fewer particles are needed to maintain the multi-mode distribution. However, we cannot use the new particles directly for particle filtering, as these particles are not sampled from the prediction density (temporal prior) $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$, otherwise the whole concept of a Bayesian approach would be lost. We can use the new particle set without destroying the original distribution by adopting the technique called importance sampling [20].

Importance sampling applies when auxiliary knowledge is available in the form of an importance function $g_t(\mathbf{x}_t)$ describing which areas of state-space contain most information about the posterior. The idea is to concentrate particles in those areas of state-space by generating $s_t^{(n)}$ from $g_t(\mathbf{x}_t)$ rather than sampling from the temporal prior $f_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$. Indeed, the MS optimization tends to move particles to such promising areas upon convergence. So the particles after running the MS optimization can be regarded as sampling from an importance function $g_t(\mathbf{x}_t)$. A correction term f/g must be added to the weights of the particles as

$$\pi_t^{(n)} = \frac{f_t(\mathbf{s}_t^{(n)})}{g_t(\mathbf{s}_t^{(n)})} p(\mathbf{z}_t|\mathbf{x}_t = \mathbf{s}_t^{(n)}) \quad \text{where}$$
$$f_t(\mathbf{s}_t^{(n)}) = p(\mathbf{x}_t = \mathbf{s}_t^{(n)}|\mathbf{z}_{1:t-1}) \tag{3}$$

to compensate for the uneven distribution of particles. The prediction density $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ is a mixture of continuous density kernels shaped by the dynamic model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, so the effect of the correction ratio is also to preserve the information about motion coherence which is present in the dynamic model. Although the particles are generated according to g_t, the distribution approximated by $\{(\mathbf{s}_t^{(n)}, \pi_t^{(n)})\}_{n=1}^{N}$ still generates $p(\mathbf{x}_t|\mathbf{z}_{1:t})$. In this way, the integration of the MS optimization does not change the probabilistic model, while improving the sampling efficiency.

In practice, the importance function g_t may omit some likely peaks, so we combine the particles generated by standard sampling from $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ and particles generated by importance sampling using g_t as the particle set for particle filtering. Therefore, as long as $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ and g_t do not simultaneously fail to predict the object state, tracking will succeed. The functions $f_t(\mathbf{x}_t)$ and $g_t(\mathbf{x}_t)$ are not available in closed form, but can be

Given the particle set at $t-1$, perform the following steps:
(1) **Resample** N particles from the set $\{(s_{t-1}^{(m)}, \pi_{t-1}^{(m)})\}_{m=1}^{2N}$ to give $\{(s_{t-1}^{'(n)}, \frac{1}{N})\}_{n=1}^{N}$.
(2) **Propagate** each particle by the dynamic model $s_t^{''(n)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1} = s_{t-1}^{'(n)})$ to give $\{(s_t^{''(n)}, \frac{1}{N})\}_{n=1}^{N}$.
(3) **Optimize** each particle with the mean shift optimization: $s_t^{'''(n)} \sim \text{Mean\_Shift}(s_t^{''(n)})$.
(4) **Combine** the particles generated by the standard sampling and the mean shift optimization, $\{(s_t^{(m)}, \frac{1}{2N})\}_{m=1}^{2N} = \{(s_t^{''(n)}, \frac{1}{2N})\}_{n=1}^{N} \cup \{(s_t^{'''(n)}, \frac{1}{2N})\}_{n=1}^{N}$, and calculate the correction ratio $\lambda_t^{(m)} = f_t(\mathbf{s}_t^{(m)})/g_t(\mathbf{s}_t^{(m)}), 1 \le m \le 2N$.
(5) **Weight** the particles as $\pi_t^{(m)} \propto \lambda_t^{(m)} p(\mathbf{z}_t|\mathbf{x}_t = s_t^{(m)})$ to give $\{(s_t^{(m)}, \pi_t^{(m)})\}_{m=1}^{2N}$, where $\pi_t^{(m)}$ are normalized so that $\sum_{m=1}^{2N} \pi_t^{(m)} = 1$.
(6) **Estimate** the tracking result $E[\mathbf{x}_t] = \sum_{m=1}^{2N} \pi_t^{(m)} s_t^{(m)}$.

Fig. 3. The mean shift embedded particle filter.

approximated as a mixture of Gaussian G:

$$f_t(\mathbf{x}_t) = \frac{1}{N}\sum_{n=1}^{N} G(s_t^{(n)}, \Sigma)(\mathbf{x}_t), \tag{4}$$

$$g_t(\mathbf{x}_t) = \frac{1}{2}\left(\frac{1}{N}\sum_{n=1}^{N} G(s_t^{(n)}, \Sigma)(\mathbf{x}_t) + \frac{1}{N}\sum_{n=1}^{N} G(s_t^{*(n)}, \Sigma)(\mathbf{x}_t)\right), \tag{5}$$

where $s_t^{*(n)}$ denotes the new particles after running the MS optimization. We summarize the MSEPF algorithm in Fig. 3.

3. Real-time hand tracking using MSEPF

The MSEPF algorithm applies to any visual tracking problems. In this section, we present a specific implementation of MSEPF for real-time hand tracking in the wheelchair.

3.1. Dynamic model

Following the methodology in many hand tracking methods [10], we represent hands by rectangles bounding themselves, which gives model independence from hand shapes. Suppose the size of the rectangle is constant for simplicity, the hand state is defined as

$$\mathbf{x} = \{x, y\},$$

where $\mathbf{x}$ is the state variable, (x, y) is the coordinate of the rectangle center. The dynamics of hand motion is modeled as

$$\mathbf{x}_t - \mathbf{x}_{t-1} = \mathbf{x}_{t-1} - \mathbf{x}_{t-2} + w_{t-1}, \tag{6}$$

where w_{t-1} is a zero-mean Gaussian stochastic component. The particle set is propagated by this simple constant velocity model (Step (2) in Fig. 3).

As described in Ref. [39], the dynamic model can be learned from a set of pre-labeled training sequences. Although it can make strong prediction for particle propagation, the learned dynamic model is not universal. For example, it cannot make effective prediction in sequences that are not included in the training set. In our work, it is difficult to learn a strong dynamic model for all kinds of natural hand motion. Therefore, we adopt the above weak constant velocity model, which is natural and universal for hand motion though its ability of prediction is limited.

For PFs, when adopting a weak dynamic model, a large number of particles will be required for reliable tracking. However, in MSEPF, particles are moved to local modes actively after propagated by the dynamic model; this can be regarded as compensating the dynamic model with current observation. So MSEPF can cope with weak dynamic models to a certain extent, and allows reliable tracking with much fewer particles.

3.2. Observation model

To perform the MS optimization and particle weighting, it is necessary to derive a reliable observation model. When users perform natural hand gestures in front of the wheelchair, hand motion may be rapid and the shape of hands in image sequences varies greatly, so it is difficult to track hand reliably based on geometric features such as hand contours. Here we utilize non-geometric features such as color and motion to describe hands.

Skin color is a distinctive feature of hands if no other exposed skin body parts or skin-colored objects in background exist, so skin color cues have been widely utilized in hand tracking [5,6,12,19–21]. Color cues have many merits such as invariance to rotation and scale changes, and robustness to partial occlusions [34]. Following Bradski [36], we first learn a histogram-based skin color model, then convert the input image to the color probability distribution image via the skin color model. Fig. 4(b) is one frame from an example sequence, and Fig. 4(c) is the corresponding color probability distribution image. As it is well known, the distribution of skin color may change due to varying illuminations. In the HCI situation of the wheelchair, the illumination changes much due to the wheelchair's movement. So we make the skin color model self-adaptive online [40], which is updated frame by frame with the tracked skin color cue. In this way, our approach can handle changing illuminations to a certain extent.

In a realistic HCI environment, there are many skin-colored objects in background such as faces. To deal with the distraction caused by skin-colored objects, we further include motion cue in the observation, as hands correspond to motion regions in image sequences. We assume that other skin-colored objects like faces in background move more slowly than hands. As background changes all the time due to the wheelchair's movement, not background substraction but temporal image differencing is adopted here to detect motion. We use the temporal differencing method described in Ref. [41], which computes the absolute value of differences in the neighborhood surrounding each pixel, and then derive the accumulated difference by summing the difference of all neighboring pixels. When the accumulated difference is above a predetermined threshold, the pixel is assigned to the moving region. Applying the

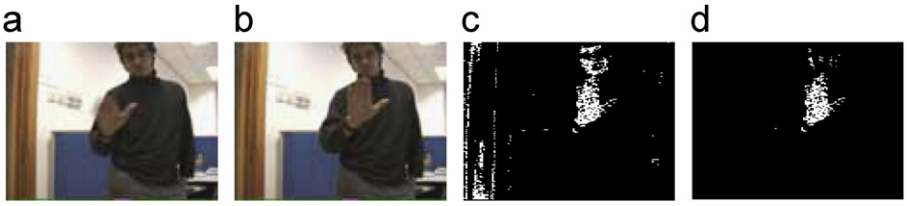

Fig. 4. (a) Frame 89; (b) frame 90; (c) the color probability distribution image of frame 90; (d) the motion–color probability distribution image of frame 90.

threshold to the accumulated difference rather than the individual pixel difference has an effect similar to morphological dilation, which can fill small gaps in motion regions. As only the motion of skin-colored regions is useful for particle weighting, we further apply the logical AND operator between the color probability distribution image and the difference image to obtain the motion–color probability distribution image, which displays the moving skin-colored regions. Fig. 4(a) and (b) are two consecutive images from a sequence, and Fig. 4(d) is the corresponding motion-color probability distribution image.

The color cue and motion cue are calculated as

$$\mathbf{M}_c = \sum_x \sum_y I_c(x, y), \quad \mathbf{M}_m = \sum_x \sum_y I_m(x, y), \tag{7}$$

where $\mathbf{M}_c$ is the zeroth moment of the rectangle in the color probability distribution image, $\mathbf{M}_m$ is the zeroth moment of the rectangle in the motion–color probability distribution image, $I_c(x, y)$ and $I_m(x, y)$ are the pixel values at (x, y) in the color probability distribution image and the motion–color probability distribution image, respectively, and x and y range over the rectangle. It is observed that the motion–color probability distribution image contains reliable motion and color cues when hand moves rapidly. On the other hand, when a hand moves slowly, particles can be mainly weighted by the color cue in the color probability distribution image. So we take a linear combination of color cue and motion cue:

$$\begin{aligned}\mathbf{M} &= (1-\alpha)\cdot\mathbf{M}_c + \alpha\cdot\mathbf{M}_m \\ &= \sum_x \sum_y ((1-\alpha)I_c(x, y) + \alpha I_m(x, y)),\end{aligned} \tag{8}$$

where $\mathbf{M}$ is the integrated zeroth moment, the coefficient α is proportional to the hand velocity when the hand velocity is less than a threshold, otherwise $\alpha = 1$, i.e., $\alpha = \min\{k\sqrt{\dot{x}^2 + \dot{y}^2}, 1\}$. Here $\dot{x}$ and $\dot{y}$ are the motion of x and y, $\sqrt{\dot{x}^2 + \dot{y}^2}$ is the hand velocity, and k is an empirical constant we set 0.04 typically.

3.3. MS optimization

Based on the above observation model, we present a simple and efficient MS iteration to drive particles to local peaks in the likelihood (Step (3) in Fig. 3).

Given a particle $s^{(i)}$ corresponds to the position $\mathbf{C}_0(x_0, y_0)$, we choose $\mathbf{C}_0(x_0, y_0)$ as the initial location, and initialize the iteration number as $itn = 0$.

Step 1: Compute the observed color and motion cues in the rectangle:

$$\begin{aligned}\mathbf{M}_{00} &= \mathbf{M} = \sum_x \sum_y ((1-\alpha)I_c(x, y) + \alpha I_m(x, y)), \\ \mathbf{M}_{10} &= \sum_x \sum_y x\cdot((1-\alpha)I_c(x, y) + \alpha I_m(x, y)), \\ \mathbf{M}_{01} &= \sum_x \sum_y y\cdot((1-\alpha)I_c(x, y) + \alpha I_m(x, y)),\end{aligned} \tag{9}$$

where $\mathbf{M}_{00}$ is the zeroth moment, $\mathbf{M}_{10}$ is the first moment for x, and $\mathbf{M}_{01}$ is the first moment for y. α is the coefficient described in Eq. (8).

Step 2: Compute the mean location (the centroid) of the rectangle based on the zeroth and the first moments:

$$x_1 = \frac{\mathbf{M}_{10}}{\mathbf{M}_{00}}; \quad y_1 = \frac{\mathbf{M}_{01}}{\mathbf{M}_{00}} \tag{10}$$

and then center the rectangle at the mean location $\mathbf{C}(x_1, y_1)$. Also increase the iteration number: $itn = itn + 1$.

Step 3: If $\|\mathbf{C}_1 - \mathbf{C}_0\| < \varepsilon$ or $itn > itn_0$, stop the iteration and update the particle $s^{(i)}$ with the position $\mathbf{C}_1$. Otherwise, set $\mathbf{C}_0 = \mathbf{C}_1$ and go to Step 1.

The first stop condition is that the mean position moves less than a predefined threshold ε, and the other is that the iteration number is larger than a predefined threshold itn_0, which is set to avoid high computational cost due to the excessive iteration.

3.4. Likelihood function

This section introduces the likelihood function used in particle weighting (Step (5) in Fig. 3). The likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ relates the observation $\mathbf{z}_t$ in the image to the state $\mathbf{x}_t$. For a particle $s^{(i)}$, its observation is the color and motion cues in the rectangle it corresponds, as defined in Eq. (8). To calculate the likelihood, we use a similarity function which defines the distance between the target and the candidate defined as

$$Dis = \sqrt{1 - \mathbf{M}/\mathbf{M}_0}, \tag{11}$$

where $\mathbf{M}_0 = \sum_x \sum_y 1$ is the number of pixels in the rectangle. Then the likelihood of the particle $s^{(i)}$

$$\pi^{(i)} = \frac{-1}{\sqrt{2\pi\sigma}} \mathrm{e}^{-Dis^2/2\sigma^2} \tag{12}$$

is specified by a Gaussian with variance $\sigma.\sigma$ is an empirical constant selected by observing the tracking performance in pilot

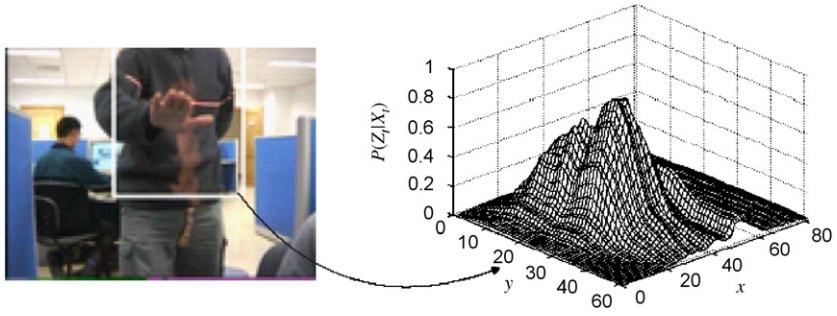

Fig. 5. The observation curve of the multi-mode distribution in a real-world image. The right side is surface plot of the observation in the white rectangle area of the left image.

studies. To illustrate the effectiveness of the likelihood function, Fig. 5 shows the observation distribution in one frame: the right side plots the observation curve of the area around the hand in the left image. Due to skin-colored sunlight regions in the background, the observation distribution is multi-mode.

4. Hand gesture recognition

Following real-time hand tracking, we can perform hand gesture recognition to control the wheelchair. Hand gesture recognition is usually categorized as image-based (or static) and sequence-based (or dynamic), based on whether the temporal information is used in recognition. In this section, we present a simple method to perform online dynamic hand gesture recognition in the wheelchair.

Dynamic hand gestures are characterized by the spatio-temporal structure of their motion patterns. So they are usually recognized by analyzing gesture trajectories, which are obtained by hand tracking in image sequences. Gesture trajectories can be statistically modeled in state space using algorithms such as HMM [42] and TDNN [21]. In Refs. [43,44], gestures were represented as template trajectories, and the input trajectory was matched with the stored templates. Davis and Bobick [45] presented a non-trajectory approach, temporal templates, for action recognition. Specifically they used motion-energy image (MEI) and motion-history image (MHI) to collapse space–time action into static images. MEI represents where motion occurs in an image sequence, and the pixel intensity in MHI is a function of the temporal history of motion at that point. By converting spatio-temporal motion into static images, the approach avoids explicit temporal analysis and sequence matching. Moreover, this method is computationally efficient, and does not need complex training. However, there are two limitations in this non-trajectory method. First, the action recognition is only based on global shape analysis of the motion region, so spatio-temporal characteristics of motion patterns are lost. Second, MHI and MEI are generated by image differencing which is not suitable for dynamic background.

To address these problems, by introducing trajectory into temporal templates, we present a simple method for dynamic gesture recognition. As MEI can be generated by thresholding MHI above zero, we only adopt MHI to represent hand motion.

Fig. 6. Examples of gesture WV, CW and HR (from top to bottom).

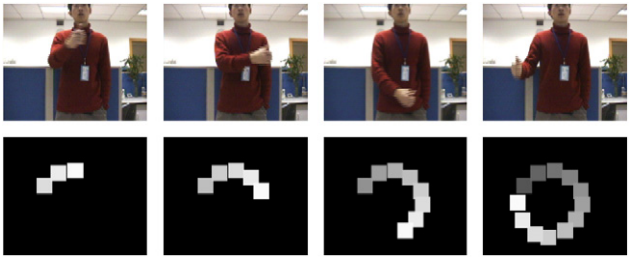

Fig. 7. Generation of TTBT for gesture ACW.

For the gesture image sequence $I(x, y, t)$, we first obtain motion regions in each image by hand tracking. Let $D(x, y, t)$ be the binary image sequence indicating regions of hand motion, the special MHI $H_\tau(x, y, t)$ for hand gesture is defined as

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1, \\ \max(0, H_\tau(x, y, t-1) - 1) & \text{otherwise,} \end{cases}$$

where τ is the duration of the template [45]. In this way, the spatial–temporal trajectories of hand gestures are retained in the static image. The trajectories are called temporal template based trajectories (TTBT). Compared to the original temporal templates, TTBT contains explicit spatio-temporal information of hand motion, thus can better distinguish hand gestures. Moreover, as it is generated from hand tracking results, TTBT

Fig. 8. TTBT of five predefined gestures (from left to right: WV, ACW, CW, HR, VT).

does not require static background and can work in dynamic environments.

In our work, we defined five simple dynamic gestures: "wave" (WV), "move horizontally" (HR), "move vertically" (VT), "move clockwise" (CW), and "move anti-clockwise" (ACW). Examples of gesture WV, CW and HR are shown in Fig. 6. The generation of TTBT for gesture ACW is demonstrated in Fig. 7. TTBT for the five gestures are shown in Fig. 8.

We perform hand gesture recognition based on the statistical analysis of TTBTs shape and orientation. Following the methodology in Ref. [45], we adopt seven Hu moments to describe the shape of TTBT. The description is invariant to translation, rotation and scale. As these moments are of different orders, we use the Mahalanobis distance metric for classification. The TTBT of HR and VT have similar shape, so do those of CW and ACW. As shown in Fig. 8, these gestures have different motion orientation. So we further use the motion orientation information of TTBT to distinguish gestures whose TTBT have similar shape.

As hand gestures may be performed at varying speeds, we adopt the backward looking algorithm in Ref. [45] to efficiently search over a wide range of τ. In the training phase, we measure the maximum and minimum duration that hand gestures may take, $\tau_{\max}$ and $\tau_{\min}$. In recognition phase, we choose $\Delta\tau$ to be $((\tau_{\max} - \tau_{\min})/(n-1))$, where n is the number of temporal windows to be considered; then at each time step, we can obtain n different τ by

$$\tau_i = \tau_{\max} - \Delta\tau \cdot (i-1), \quad i = 1, \ldots, n.$$

Therefore, n different TTBT are derived to perform gesture recognition.

5. Experiments

In this section, we first carry out extensive experiments to evaluate the developed hand tracking algorithm. Then we conduct online gesture recognition in dynamic environments of the moving wheelchair, to validate the performance of the proposed algorithms in real-world applications.

5.1. Hand tracking

We performed hand tracking experiments on more than 30 video sequences of hand gestures. The video sequences were captured at rate 12fps using a Sony camera equipped on the wheelchair in common office environments. Each sequence contains around 400 frames, and the image resolution is 240 × 180 pixels. No constraints about illumination and background were set in capturing, so varying illumination and cluttered background exist in some sequences. The proposed algorithms were implemented on a HP Notebook (Pentium 2.4G, Window 2000) with MS Visual C++.

Table 1
Tracking performance of MSEPF (50 particle), PF (200 particles) and MS

	MSEPF	PF	MS
Correct tracking rates	100%	100%	73.3%

5.1.1. Comparative study

We first access the proposed MSEPF algorithm's performance in comparison to the MS tracker and the conventional PF. For the sake of fair comparison, PF adopted the same dynamic model and observation model (likelihood function) as MSEPF, and MS was implemented as the CAMShift algorithm [36], which uses the same color information as MSEPF. With regards to the number of particles, 200 particles were used for PF and 50 particles for MSEPF.

We define tracking to be lost when the center of hand region (ground truth) is not in the rectangle anymore. The tracking for the sequence was stopped then, even though the hand might later have coincidentally caught the tracker again due to the hand's path intersecting the erroneously tracked location. The ground truth was manually labeled. Table 1 shows the overall tracking performance of the three trackers. We can observe that MSEPF (50 particles) and PF (200 particles) provide superior tracking performance over MS. Some sample tracking results in one test sequence are shown in Fig. 9. In this sequence, the subject has continuous body movements, and there is skin-colored sunlight on the upper body; so the observation distribution is multi-modes in many frames (as shown in Fig. 5). The tracking accuracy of the three algorithms for part of this sequence is plotted in Fig. 10. As can be seen in Figs. 9 and 10, MSEPF and PF successfully track the hand, whereas MS occasionally loses the hand, for example, in frame 63, due to the confusion caused by the skin-colored sunlight. As only a single hypothesis is carried, MS cannot deal with multi-modal distribution, so easily converges to a local maximum, and it cannot recover from the lost track. In contrast, PF and MSEPF can robustly track the hand through clutter, as they maintain multiple modes simultaneously.

To investigate how much the embedded MS iteration can improve the sampling efficiency of PF for hand tracking, we also performed comparative study on the number of particles between MSEPF and PF. Fig. 11 presents the tracking results when varying the number of particles. It is observed that, as hand motion varies much and a weak dynamic model is adopted,

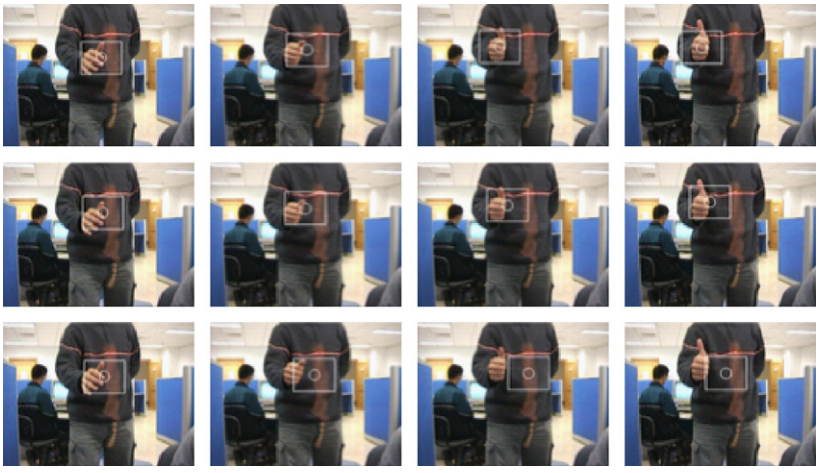

Fig. 9. Some tracking results of MS, PF and MSEPF for one test sequence (frame 61 to frame 64). Top row: MSEPF. Middle row: PF. Bottom row: MS.

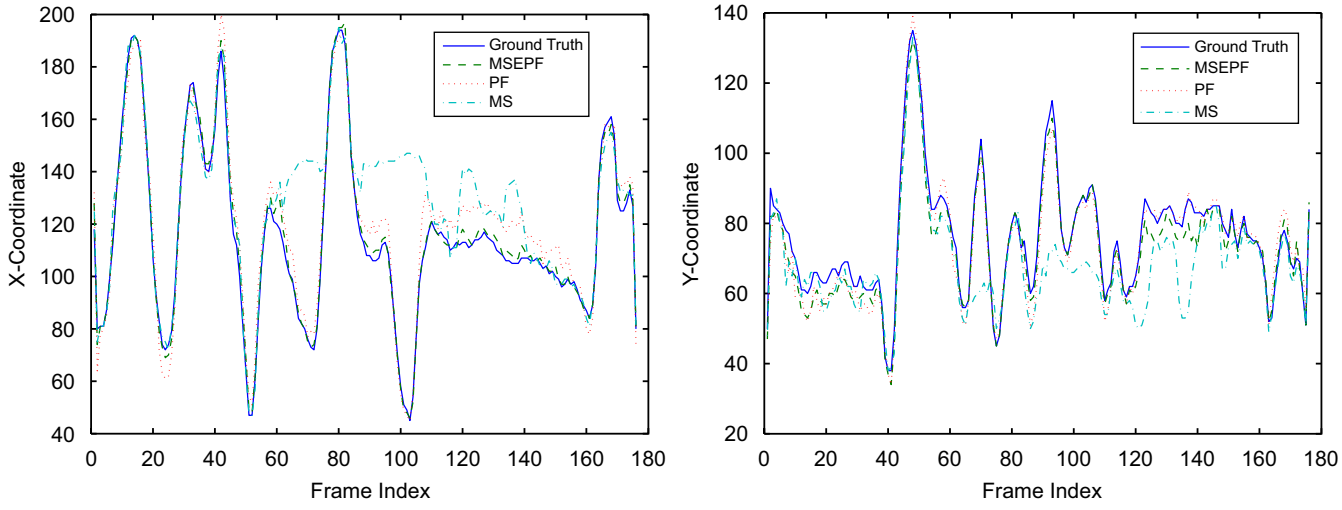

Fig. 10. Comparison of the tracking accuracy for one test sequence. Left: x coordinate. Right: y coordinate.

PF requires at least 150 particles for stable tracking in all test sequences, while MSEPF requires only 20 particles.[1] With regard to the computation time, PF (150 particles) spends 63 ms on average for each frame, whereas for MSEPF (20 particles), though MS iterations added, each frame only costs an average time of 28 ms as the number of particles is decreased greatly. Therefore, in the application of hand tracking, MSEPF decrease 85% particles than PF, and thus is much faster than PF. Table 2 summaries the optimized tracking performance of the three algorithms on all test sequences. We observe that MSEPF achieves better tracking results than both PF and MS. Although it is faster than MSEPF, MS cannot achieve stable tracking and sometimes loses the track. Obviously, the efficiency improvement of MSEPF over PF depends on how good particle positions update that the MS iterations can provide, which is variable across different tracking problems at hand. Thus, it is difficult to provide a general conclusion on how many particles can be saved in MSEPF. What we can conclude is embedding the MS optimization into PF improves the efficiency of the conventional PF.

Another question is how to design the number of particles. In our study, we decided the optimal number of particles via a few pilot studies as said above. The existing applications usually select the number of particles using ad hoc criteria, or statistical methods such as Monte Carlo simulator or some standard statistical bound [46]. For example, Boers [46] presented a method to relate the required number of particles to the accuracy by which the conditional pdf $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is being approximated by the discrete distribution, and derived statistical bounds for the number of particles.

Most of the existing PFs select a fixed number of particles in advance. However, this is not necessary, and can be highly inefficient, as the dynamics of most systems usually produces great variation in the complexity of the posterior distribution

[1] It is worth pointing out that, as shown in Fig. 3, for MSEPF with N particles, $2N$ particles are used in particle weighting. So the optimized MSEPF actually uses 40 particles in particle weighting for successful tracking.

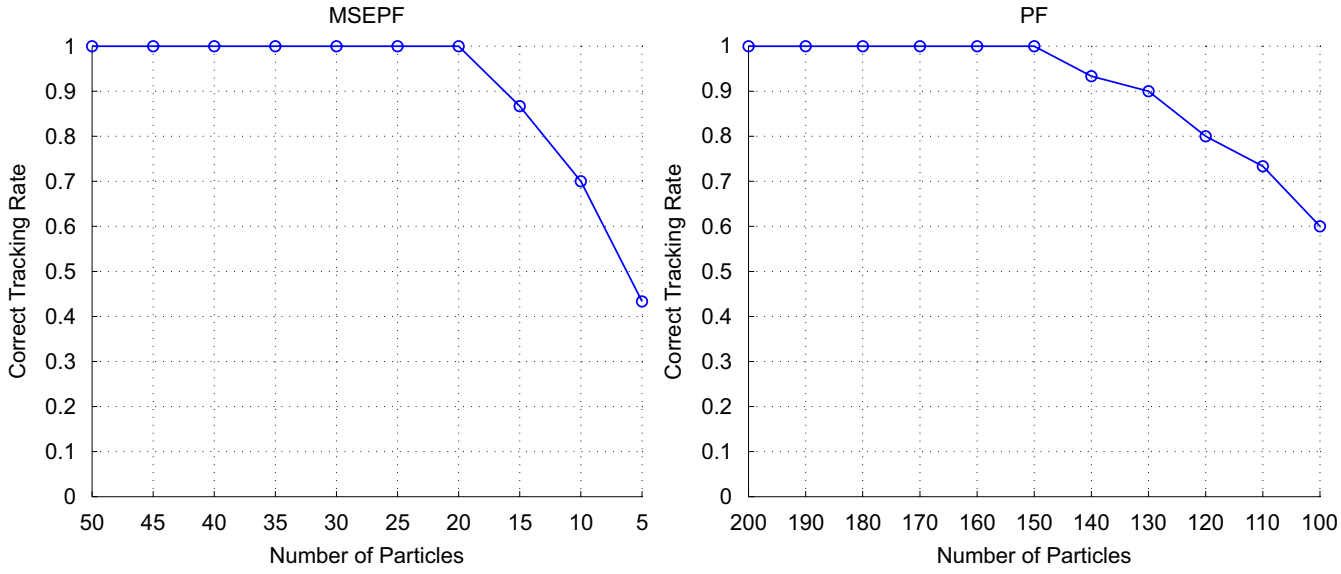

Fig. 11. Tracking performance vs. the number of particles. Left: MSEPF. Right: PF.

Table 2
Optimized tracking performance of MSEPF, PF and MS

Algorithm	Particles required	Average time/frame (ms)	Average position error (pixels)
MSEPF	20	28	5.8
PF	150	63	8.7
MS	—	20	Track lost sometimes

over time. For example, in the beginning, a large number of particles is needed as the variance is high due to uncertainty in the initial state, while as time evolves the variance reduces, only a small number of particles suffices to perform good estimation. Recently, several approaches have been introduced to adapt the number of particles over time [47–50]. The adaptive approach in Refs. [47,48] adjusts the number of particles based on the likelihood of observations, i.e., generating particles until the sum of the non-normalized likelihoods exceeds a pre-specified threshold. Fox [49] proposed to adapt the size of particle sets based on the approximation errors introduced by the sample-based representation. Soto [50] more recently presented a self-adaptive PF that uses statistical methods to adapt the number of particles and the propagation function at each iteration. Therefore, in order to further optimize the number of particles, we will study how to improve MSEPF by adapting the number of particles over time.

5.1.2. More tracking results

To verify the effectiveness of the proposed observation model, we run the algorithm on some sequences containing distraction caused by skin-colored objects in the background. For comparison, we also implemented the tracking algorithm using the observation model based on color cue only. Fig. 12 shows some tracking results of one test sequence with two different observation models. (*note*: all video demonstrations $V1$–$V7$ are available at http://www.dcs.qmul.ac.uk/~cfshan/hand_gesture.html). It is observed that the tracker with the observation model based on color cue only loses the hand due to the confusion caused by the face, whereas the tracker with the proposed observation model successfully handle the distraction. So the measurement based on fusion of color and motion cues is much reliable than the one using the individual cue. More tracking results about handling skin-colored distraction are shown in Fig. 13.

Some tracking results of one sequence containing varying illumination are shown in Fig. 14. As the skin color model is adapted during tracking, the algorithm can handle skin colors change due to varying illumination to a certain extent. Fig. 15 shows that the algorithm successfully tracks rapid hand motion, even if the search regions do not overlap in consecutive frames. In contrast, MS loses the target in this situation.

The above experiments demonstrate that the developed hand tracking algorithm achieves reliable tracking in an efficient way, i.e., the average tracking rate of 35 Hz, which builds foundation for late hand gesture recognition.

5.2. Online dynamic gesture recognition

We first performed off-line experiments to evaluate the performance of the proposed dynamic gesture recognition method. We had four people perform each gesture eight times. Out of the 160 gesture sequences, 80 were randomly chosen for training, 16 from each class. The remaining gesture sequences were used to test. The overall recognition rate is 97.3%. To compare with the temporal templates in Ref. [45], we conducted same experiments using temporal templates. As temporal templates are generated by image differencing, many motion in the background are included in them, which make gesture recognition much difficult. Furthermore, the shape description of temporal templates is not distinguished enough for

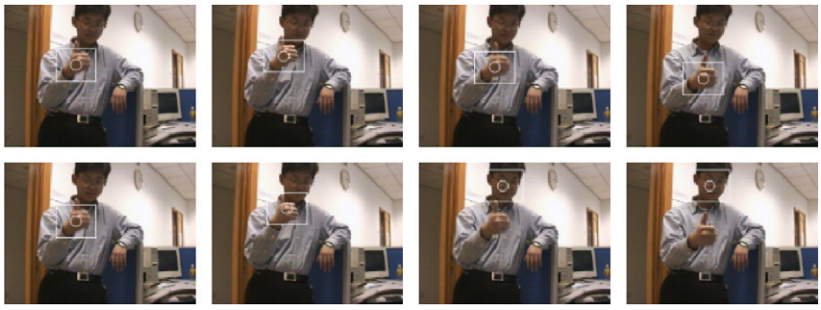

Fig. 12. Comparison between two different observation models for one test sequence (frames 5 to 8). Top row: Observation model based on color and motion cues. Bottom row: Observation model based on color cue only. The video demo $V1$ is available online.

Fig. 13. Tracking results of two sequences with skin-colored distraction. Top row: Frames 45, 47, 49 and 50. Bottom row: Frames 19, 21, 22 and 24. The video demos $V1$ and $V2$ are available online.

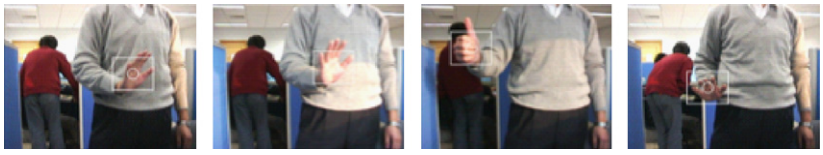

Fig. 14. Tracking results of one sequence with illumination changes (frames 77, 103, 176 and 229). The video demo $V3$ is available online.

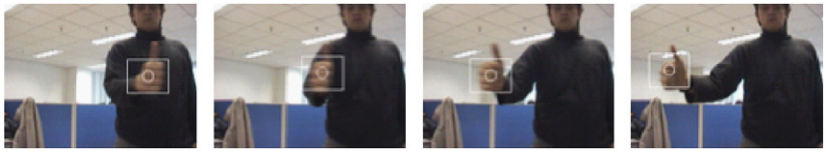

Fig. 15. Tracking results of one sequence with rapid hand motion (frames 71 to 74). The video demo $V4$ is available online.

Table 3
Recognition performance of TTBT and temporal templates

Methods	Recognition rates
TTBT	97.3%
Temporal templates [45]	49.1%

hand gesture recognition. Comparison results summarized in Table 3 demonstrate that the proposed TTBT is more effective than temporal templates for dynamic hand gesture recognition.

Following off-line experiments, we implemented a hand control interface based on dynamic gesture recognition in the wheelchair. The control interface worked well in dynamic environments. The video demo $V5$ of controlling the wheelchair via dynamic gesture recognition is available online.

6. Conclusions and discussions

Vision-based hand gesture recognition is an important technology for intelligent HCI. In this paper, we address real-time hand tracking and gesture recognition in the context of developing a hand control interface for a robotic wheelchair. By integrating the MS optimization into particle filtering, we propose an improved PF MSEPF for efficient and robust tracking. Real-time hand tracking in the wheelchair is implemented

using MSEPF. Extensive experiments demonstrate that MSEPF is superior to the MS tracker and the conventional PF. Our approach produces reliable tracking while effectively handling rapid motion and distraction with roughly 85% fewer particles. We also present a simple but effective approach for dynamic gesture recognition. The hand control interface based on the proposed algorithms works well in dynamic environments of the wheelchair.

As only color and motion cues are used, our hand tracking algorithm in fact is a general method and could be used in many tracking problems. We are considering to include other features, e.g., texture of hands, for more robust hand tracking. Although the proposed MSEPF is effective for hand tracking, further investigation should be conducted to verify its effectiveness in other tracking problems, especially the higher dimensional problems such as 3D articulated object tracking, as the number of particles required in high dimensional space is more prohibitive. As far as dynamic gesture recognition is concerned, the TTBT models are learned from only one view at the moment. We will combine multiple views in training, which will improve the recognition performance.

Acknowledgments

The authors would like to thank Frederic Ojardias and Xianchao Qiu for their help in gesture recognition experiments. This research is funded by the National Hi-Tech R&D Program (no. 2001AA422430), NSFC (no. 60121302), LIAMA Projects (no. 01-03), and the National 973 Program (Grant 2004CB318100).

References

[1] Y. Wu, Vision and learning for intelligent human–computer interaction, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 2001.

[2] V. Pavlovic, R. Sharma, T. Huang, Visual interpretation of hand gestures for human–computer interaction: a review, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 677–695.

[3] G. McAllister, S. McKenna, I. Ricketts, Hand tracking for behaviour understanding, Image Vision Comput. 20 (12) (2002) 827–840.

[4] K. Oka, Y. Sato, H. Koike, Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems, in: Proceedings of IEEE International Conference on Automated Face and Gesture Recognition (FG'02), 2002, pp. 411–416.

[5] I. Laptev, T. Lindeberg, Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features, in: Proceedings of IEEE Workshop on Scale-Space and Morphology, 2001.

[6] H. Fei, I. Reid, Probabilistic tracking and recognition of non-rigid hand motion, in: Proceedings of IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG'03), 2003, pp. 60–67.

[7] S. Lu, D. Metaxas, D. Samaras, J. Oliensis, Using multiple cues for hand tracking and model refinement, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03), 2003, pp. II: 443–450.

[8] H. Zhou, T. Huang, Tracking articulated hand motion with eigen dynamics analysis in: Proceedings of IEEE International Conference on Computer Vision (ICCV'03), 2003, pp. 1102–1109.

[9] L. Tsap, Gesture-tracking in real time with dynamic regional range computation, Real Time Imaging 8 (2) (2002) 115–126.

[10] A. Shamaie, A. Sutherland, A dynamic model for real-time tracking of hands in bimanual movements, Gesture-Based Communication in Human–Computer Interaction (International Gesture Workshop, GW03).

[11] E.B. Sudderth, M.I. Mandel, W.T. Freeman, A.S. Willsky, Visual hand tracking using nonparametric belief propagation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04), 2004.

[12] M. Kolsch, M. Turk, Fast 2d hand tracking with flocks of features and multi-cue integration, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04), 2004.

[13] Q. Yuan, S. Sclaroff, V. Athitsos, Automatic 2d hand tracking in video sequences in: Proceedings of IEEE Workshop on Applications of Computer Vision (WACV'05), 2005.

[14] T. Inaguma, H. Saji, H. Nakatani, Hand motion tracking based on a constraint of three-dimensional continuity, J. Electron. Imaging 14 (1) (2005) 013021.

[15] W.Y. Chang, C.S. Chen, Y.P. Hung, Appearance-guided particle filtering for articulated hand tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[16] S. Wu, L. Hong, Hand tracking in a natural conversational environment by the interacting multiple model and probabilistic data association (imm-pda) algorithm, Pattern Recognition 38 (2005) 2143–2158.

[17] B. Stenger, A. Thayananthan, P.H.S. Torr, R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter, IEEE Trans. Pattern Anal. Mach. Intell. 28 (9) (2006) 1372–1384.

[18] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, Int. J. Comput. Vision 29 (1) (1998) 5–28.

[19] C. Ng, S. Ranganath, Real-time gesture recognition system and application, Image Vision Comput. 20 (13–14) (2002) 993–1007.

[20] M. Isard, A. Blake, ICONDENSATION: unifying low-level tracking in a stochastic framework, in: Proceedings of European Conference on Computer Vision (ECCV'98), vol. 1, Freiburg, Germany, 1998, pp. 893–908.

[21] M. Yang, N. Ahuja, M. Tabb, Extraction of 2d motion trajectories and its application to hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1061–1074.

[22] Vision based hand gesture recognition systems. URL ⟨http://ls7-www.cs.uni-dortmund.de/research/gesture/vbgr-table.html ⟩.

[23] M. Bray, E. Koller-Meier, L. Van Gool, Smart particle filtering for 3d hand tracking, in: Proceedings of IEEE International Conference on Automated Face and Gesture Recognition (FG'04), 2004, pp. 675–680.

[24] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–189.

[25] Y. Rui, Y. Chen, Better proposal distributions: object tracking using the unscented particle filter, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), vol. 2, Hawaii, 2001, pp. 786–793.

[26] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: Proceedings of European Conference on Computer Vision (ECCV'00), 2000, pp. 3–19.

[27] T. Cham, J.M. Rehg, A multiple hypothesis approach to figure tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99), 1999, pp. 239–245.

[28] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealled particle filtering, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), 2000, pp. 126–133.

[29] K. Choo, D.J. Fleet, People tracking using hybrid Monte Carlo filtering, in: Proceedings of IEEE International Conference on Computer Vision (ICCV'01), 2001, pp. 1068–1075.

[30] C. Chang, R. Ansari, Kernel particle filter: iterative sampling for efficient visual tracking, in: Proceedings of IEEE International Conference on Image Processing (ICIP'03), 2003.

[31] B. Han, D. Comaniciu, Y. Zhu, L. Davis, Incremental density approximation and kernel-based Bayesian filtering for object tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04), Washington, DC, 2004.

[32] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), South Carolina, 2000, pp. 142–149.

[33] C. Shan, Y. Wei, T. Tan, F. Ojardias, Real time hand tracking by combining particle filtering and mean shift, in: Proceedings of IEEE International Conference on Automated Face and Gesture Recognition (FG'04), Seoul, Korea, 2004, pp. 669–674.
[34] P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: Proceedings of European Conference on Computer Vision (ECCV'02), 2002, pp. I: 661–675.
[35] K. Nummiaro, E. Koller-Meier, L. Van Gool, An adaptive color-based particle filter, Image Vision Comput. 21 (1) (2003) 99–110.
[36] G. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technol. J. Q2.
[37] D. Comaniciu, V. Ramesh, Mean shift and optimal prediction for efficient object tracking, in: Proceedings of IEEE Conference on Image Processing (ICIP'00), Vancouver, Canada, 2000, pp. 70–73.
[38] C. Yang, R. Duraiswami, L. Davis, Efficient mean-shift tracking via a new similarity measure, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.
[39] A. Blake, M. Isard, Active Contours, Springer, Berlin, 1998.
[40] M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, Skin detection in video under changing illumination conditions, in: Proceedings of International Conference on Pattern Recognition (ICPR'00), 2000, pp. 839–842.
[41] H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, E. Petajan, Multi-modal system for locating heads and faces, in: Proceedings of International Conference on Automatic Face and Gesture Recognition (FG'96), 1996, pp. 88–93.
[42] T. Starner, J. Weaver, A. Pentland, Real-time american sign language recognition using desk and wearable computer based video, IEEE Trans. Pattern Anal. Mach. Intell. 20 (12) (1998) 1371–1375.
[43] M.J. Black, A.D. Jepson, Recognition temporal trajectories using the CONDENSATION algorithm, in: Proceedings of IEEE International Conference on Automated Face and Gesture Recognition (FG'98), Japan, 1998, pp. 16–21.
[44] A. Psarrou, S. Gong, M. Walter, Recognition of human gestures and behaviour based on motion trajectories, Image Vision Comput. 20 (5–6) (2002) 349–358.
[45] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 257–267.
[46] Y. Boers, On the number of samples to be drawn in particle filtering, in: IEE Colloquium on Target Tracking: Algorithms and Applications, 1999, pp. 5/1–5/6.
[47] D. Koeller, R. Fratkina, Using learning for approximation in stochastic processes, in: Proceedings of International Conference on Machine Learning (ICML'98), 1998.
[48] D. Fox, W. Burgard, F. Dellaert, S. Thrun, Monte Carlo localization: efficient position estimation for mobile robots, in: Proceedings of National Conference on Artificial Intelligence (AAAI), 1999.
[49] D. Fox, Kld-sampling: adaptive particle filters, in: Advances in Neural Information Processing Systems (NIPS), 2001.
[50] A. Soto, Self adaptive particle filter, in: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'05), 2005.

About the Author—CAIFENG SHAN received the B.Eng. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2001, the M.Eng. degree in Pattern Recognition and Intelligent System from National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China, in 2004. Currently he is a Ph.D. student in Queen Mary, University of London, London, UK. He was awarded the Queen Mary Research Studentship (2004–2007). His research interests include computer vision, pattern recognition, and image/video processing. He is a student member of IEEE.

About the Author—TIENIU TAN received the B.Sc. degree in electronic engineering from Xian Jiaotong University, China, in 1984 and the M.Sc., DIC, and Ph.D. degrees in electronic engineering from the Imperial College of Science, Technology, and Medicine, London, UK, in 1986 and 1989, respectively. He joined the Computational Vision Group, The University of Reading, Reading, UK, in October 1989, where he was a Research Fellow, Senior Research Fellow, and Lecturer. Currently, he is a Professor and the Director of the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing. He is an Associate Editor of Pattern Recognition. His current research interests include image processing, computer vision, pattern recognition, multimedia, and robotics. Dr. Tan is an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence. He was an Elected Member of the Executive Committee of the British Machine Vision Association and Society for Pattern Recognition (1996–1997) and is a Founding Co-Chair of the IEEE International Workshop on Visual Surveillance.

About the Author—YUCHENG WEI received the B.S. and M.S. degree in Automation from Hunan University in 1997 and 2000, and the Ph.D. degree in Pattern Recognition and Intelligent System from the Institute of Automation, Chinese Academy of Sciences in 2004. From April 2004 to December 2005, he was an associate researcher in NEC research lab, China. From January 2006 to April 2006, he was CTO of AssureDigit Tech. Ltd, and from May 2006 to September 2006, he was CTO of Ibayway Tech. Ltd. Since October 2006, he has been with Beijing Irisking Tech. Ltd, where he presently holds the position of CEO. His research interests include vision-based human-robot interaction and navigation, multimedia watermarking, iris recognition and information security. He is a member of IEEE.