

Tracking Human Motion With Multichannel Interacting Multiple Model

Suk Jin Lee, *Student Member, IEEE*, Yuichi Motai, *Member, IEEE*, and Hongsik Choi

Abstract—Tracking human motion with multiple body sensors has the potential to promote a large number of applications such as detecting patient motion, and monitoring for home-based applications. With multiple sensors, the tracking system architecture and data processing cannot perform the expected outcomes because of the limitations of data association. For the collaborative and intelligent applications of motion tracking (Polhemus Liberty AC magnetic tracker), we propose a human motion tracking system with multichannel interacting multiple model estimator (MC-IMME). To figure out interactive relationships among distributed sensors, we used a Gaussian mixture model (GMM) for clustering. With a collaborative grouping method based on GMM and expectation-maximization algorithm for distributed sensors, we can estimate the interactive relationship with multiple body sensors and achieve the efficient target estimation to employ a tracking relationship within a cluster. Using multiple models with filter divergence, the proposed MC-IMME can achieve the efficient estimation of the measurement and the velocity from measured datasets of human sensory data. We have newly developed MC-IMME to improve overall performance with a Markov switch probability and a proper grouping method. The experiment results shows that the prediction overshoot error can be improved on average by 19.31% by employing a tracking relationship.

Index Terms—Cluster number selection, expectation-maximization (EM), Gaussian mixture model (GMM), interacting multiple model, motion tracking.

I. INTRODUCTION

TRACKING human motion with multiple body sensors has the potential to improve the quality of human life and to promote a large number of application areas such as health care, medical monitoring, and sports medicine [1]–[3]. Tracking human motion with multiple body sensors has the potential to promote a large number of applications of detecting patient motion and monitoring human behavior for home-based applications, e.g., smart television and gesture-based gaming. The information provided by multiple body sensors are expected to be more accurate than the information provided by a single sensor [2]–[4]. In multiple-sensory systems, however,

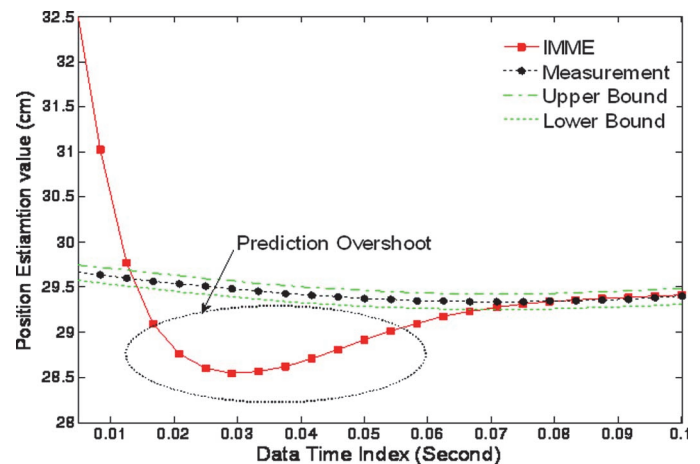


Fig. 1. Prediction overshoot of IMME. This figure shows the position estimation of benign motion for the human chest. The upper and lower bounds can be derived from adding the marginal value to the measurement and subtracting the marginal value from the measurement, respectively [26].

the tracking system architecture and data processing cannot perform the expected outcomes because of the limitations of data association [5]–[14]. As shown in Fig. 1, an individual sensory system using interacting multiple model estimator (IMME) shows the position estimates of benign motion for the human chest. The typical problem showed in this figure is that the prediction overshoots at the beginning of tracking estimation can result in a significant prediction error. This fact has motivated us to develop an appropriate method that may reduce the initial prediction estimate error. We propose a new method to reduce the initial prediction estimate error by employing a tracking relationship of data association [15]–[25].

As a solution to prevent significant prediction overshoots from initial estimate errors, we adopt multiple sensory systems with a grouping method based on a Gaussian mixture model (GMM) for clustering. Clustering is a method that divides a set of distributed sensors into subsets so that distributed sensors in the subset are executed in a similar way. A variety of studies have been investigated on clustering methods based on k -means, spectral clustering, or expectation-maximization (EM) algorithm [1], [27]–[38]. However, a known limitation of these clustering methods is that the number of clusters must be predetermined and fixed. Recently, Bayesian nonparametric methods with Dirichlet process mixture have become popular to model the unknown density of the state and measurement noise [39], [40]. However, because of the relatively small set of samples, it will not adequately reflect the characteristics of the

Manuscript received August 11, 2012; revised February 20, 2013; accepted April 01, 2013. Date of publication April 12, 2013; date of current version August 16, 2013. This work was supported in part by the Dean's Office of the School of Engineering, Virginia Commonwealth University (VCU), VCU Presidential Research Incentive Program Award, and by the National Science Foundation CAREER Award 1054333. Paper no. TII-12-0626.

S. J. Lee and Y. Motai are with the Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: leesj9@mymail.vcu.edu; ymotai@vcu.edu).

H. Choi is with the Department of Information Technology, Georgia Gwinnett College, Lawrenceville, GA 30043 USA (e-mail: hchoi1@ggc.edu).

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/TII.2013.2257804

cluster structure [30]. For the time sequential datasets of distributed body sensors, we would like to add a *prior* distribution on the cluster association probability [41], [42]. We refer to this *prior* distribution as hyper-parameters [41]. Therefore, we proposed a new collaborative grouping method for distributed body sensors.

Multiple model (MM) may have multiple possible dynamic models for multisensor systems with Markov model switches. In such a hybrid system, the possible models make multiple sensors supply the information about the interested variable and, thus, are collaborative and complementary. The basic idea of all MM approaches is that complicated target movements are made up of random variations originating from basic (straight-line) target motion. Due to the difficulty in representing this motion with a single model, MMs including potentially dynamic models operate in a parallel way with Markov switch probability [15]. The proposed solution employs a tracking relationship among distributed sensors by adding switching probability for multiple models and clustering method to figure out the interactive relation among the sensors. IMME algorithm can be used to combine different filter models with improved control of filter divergence. As a suboptimal hybrid filter [15]–[17], IMME makes the overall filter recursive by modifying the initial state vector and covariance of each filter through a probability weighted mixing of all the model states and probabilities [18]–[25].

The overall contribution of this paper is to minimize the prediction overshoot originating from the initialization process by the newly proposed multichannel IMME (MC-IMME) algorithm with the interactive tracking estimation. MC-IMME can estimate the object location as well as the velocity from measured datasets using multiple sensory channels. For this MC-IMME, we have extended the IMME to improve overall performance by adding switching probability to represent the conditional transition probability and a collaborative grouping method to select a proper group number based on the given dataset. The technical contributions of this paper are twofold. First, we propose a group number selection method for distributed body sensors based on Dirichlet hyper-prior on the cluster assignment probabilities. Second, we present a new prediction method to reduce the initial estimate error by employing a tracking relationship among distributed body sensors. For the performance improvement, we added switching probability to represent the conditional transition probability from a previous channel state to a current channel state and a collaborative transition probability to select a proper group number based on the given datasets.

This paper is organized as follows. In Section II, the theoretical background for the proposed algorithm is briefly discussed. In Sections III and IV, the proposed grouping criteria with distributed sensors placement based on EM algorithm and the proposed estimate system design for distributed body sensors are presented in detail, respectively. Section V presents and discusses experimental results of proposed methods—grouping methods and adaptive filter design. A summary of the performance of the proposed method is presented in Section VI.

II. BACKGROUND

A. Kalman Filter (KF)

The KF provides a general solution to the recursive minimized mean-square estimation problem within the class of linear estimators [43], [44]. Use of the KF will minimize the mean-squared error as long as the target dynamics and the measurement noise are accurately modeled. Consider a discrete-time linear dynamic system with additive white Gaussian noise (AWGN) that models unpredictable disturbances. The problem formulation of dynamic and the measurement equation are

$$\begin{aligned} x(k+1) &= F_i(k)x(k) + G_i(k)u(k) + v_i(k) \\ z(k) &= H_i(k)x(k) + w_i(k) \end{aligned} \quad (1)$$

where $x(k)$ is the n -dimensional state vector and $u(k)$ is an n -dimensional known vector (which is not used in our application), the subscript i denotes quantities attributed to model M_i , and $v(k)$ and $w(k)$ are process noise and measurement noise with the property of the zero-mean white Gaussian noise with covariance, $E[v(k)v(k)^T] = Q(k)$ and $E[w(k)w(k)^T] = R(k)$, respectively. The matrices F , G , H , Q , and R are assumed to be known and possibly time-varying, which means that the system can be time-varying and the noise nonstationary.

The KF estimates a process by using a form of feedback control, and the equations for the KF can be represented with two groups: time update equations and measurement update equations. The estimation algorithm starts with the initial estimate $\hat{x}(0)$ of $x(0)$ and associated initial covariance $P(0)$. The problem formulation of the predicted state and the state prediction covariance can be written as

$$\begin{aligned} \hat{x}(k+1) &= F(k)\hat{x}(k) + G(k)u(k) \\ P(k+1) &= F(k)P(k)F(k)^T + Q(k). \end{aligned} \quad (2)$$

For the proposed MC-IMME, we use (1) and (2) with a different model of filters, i.e., a constant velocity model and a constant acceleration model.

B. Interacting MM Framework

MM algorithms can be divided into three generations: autonomous multiple models (AMMs), cooperating multiple models (CMMs), and variable structure multiple models (VSMMs) [14], [15]. The AMM algorithm uses a fixed number of motion models operating autonomously. The AMM output estimate is typically computed as a weighted average of the filter estimates. The CMM algorithm improves on AMM by allowing the individual filters to cooperate. The VSMM algorithm has a variable group of models cooperating with each other. The VSMM algorithm can add or delete models based on performance, eliminating poorly performing ones and adding candidates for improved estimation [53], [54]. The well-known IMME algorithm is part of the CMM generation [15].

The main feature of the interacting multiple model (IMM) is the ability to estimate the state of a dynamic system with several behavior models. For the IMM algorithm, we have implemented two different models based on the KF: 1) a constant velocity (CV) filter in which we use the direct discrete-time kinematic

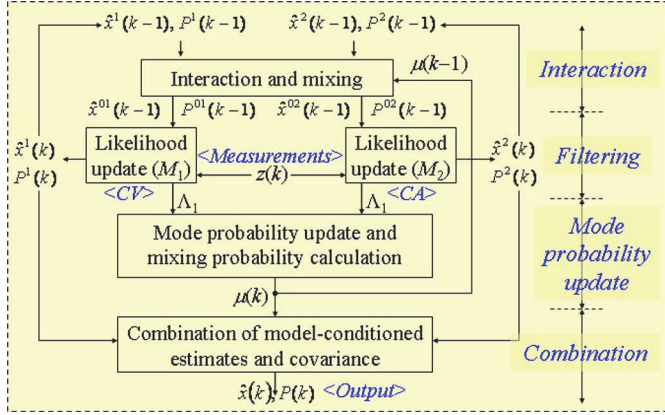


Fig. 2. IMME has a four-step process in a way that different state models are combined into a single estimator to improve performance.

models and 2) a constant acceleration (CA) filter in which the third-order state equation is used [23]–[25], [43], [45]–[47]. The IMME is separated into four distinct steps: interaction, filtering, mode probability update, and combination [43]. Fig. 2 depicts a two-filter IMM estimator, where $\hat{x}$ is the system state, P is the filter estimate probability, z is the measurement data, and μ are mixing probabilities. Note that the previous state of each model is reinitialized by the interaction stage each time the filter iterates. In IMME, at time k the state estimate is computed under each possible current model using CV or CA.

In Fig. 2, the mixing probability (μ_{ij}) represents the conditional transition probability from state i to state j . With an initial state of each model ($\hat{x}^i(k-1)$), a new filter state is computed to estimate the mixed initial condition ($\hat{x}^{0i}(k-1)$) and the associated covariance ($P^{0i}(k-1)$) according to the mixing probability. The above estimates and the covariance are used as input to the likelihood update matched to $M_j(k)$, which uses the measurement data ($z(k)$) to yield $\hat{x}^i(k)$ and $P^i(k)$. The likelihood function (Λ_i) corresponding to each model i is derived from the mixed initial condition ($\hat{x}^{0i}(k-1)$) and the associated covariance ($P^{0i}(k-1)$). After mode probability update based on a likelihood function (Λ_i), combination of the model-conditioned estimates and covariance is computed for output purposes with the mixing probability. For our distributed sensory system of target estimation, each filter state of IMM is dedicated for each sensor, and distributed target estimations independently progress according to each IMME.

C. Cluster Number Selection Using GMM and EM Algorithm

For industrial applications of motion tracking, distributed body sensors placed on a target surface with different positions and angles can have specific correlation with others. This means that distributed body sensors can cooperate with each other within a group. Recently, several clustering algorithms have been developed to partition the observations (L) into several subsets (G) [27]–[38]. The most notable approaches are a mean square error (MSE) clustering and a model-based approach. The MSE clustering typically is performed by the well-known k -means clustering. In general, k -means clustering problem is NP -hard [27], so a number of heuristic algorithms are generally used [33], [35], [36].

A model-based approach to deal with the clustering problem consists of certain models, e.g., a Gaussian or a Poisson model

for clusters and attempting to optimize the fit between the data and the model. The most widely used clustering method of this kind is a GMM [30]–[32]. In GMM, the joint probability density that consists of the mixture of Gaussians $\phi(z; m_y, \Sigma_y)$, where $y = 1 \dots G$, should be solved [29], [30]. We assume a training set of independent and identically distributed points sampled from the mixture, and our task is to estimate the parameters, i.e., prior probability (α_y), mean (m_y), and covariance (Σ_y) of the clustering components (G) that maximize the log-likelihood function $\delta(\cdot)$ based on the EM algorithm [30], [32]. Given an initial estimation (α_0, m_0, Σ_0), EM algorithm calculates the posterior probability $p(y|z_j)$ in E-step. Based on the estimated result, we can calculate the prior probability (α_y), mean (m_y), and covariance (Σ_y) for the next iteration, respectively, in the following M-step [48]–[50].

The log-likelihood for the incomplete datasets in the EM algorithm can never be decreased [51, Ch. 1.6], because the EM algorithm iterates the computations E-step and M-step until the convergence to a local maximum of the likelihood function. This means that the consecutive log-likelihood functions monotonically increase and could be very similar but not identical. We define the discrepancy of consecutive values as the difference (Δ). Now, we can define the difference (Δ) as follows:

$$\Delta(G) \cong \delta(\Theta, G) - \delta(\Theta, G-1). \quad (3)$$

Once we estimate the parameter $\Theta \equiv \{\alpha_y, m_y, \Sigma_y\}_{y=1}^G$, we can find the optimal cluster number (G^*) with the conventional brute-force search algorithm by introducing $\Delta(G)$ that is a log-likelihood function after parameter learning with the following equation: $G^* = \text{argmin}_G \Delta(G)$. In practice, we can set a threshold (Δ_{th}) that is almost closed to zero to save the redundant iteration step. We can start with $G = 2$ for a first candidate solution for the cluster number selection, estimate the set of finite mixture model parameter $\Theta^* \equiv \{\alpha_y^*, m_y^*, \Sigma_y^*\}_{y=1}^G$ using EM algorithms based on the sample data, and calculate $\Delta(G)$. After checking whether a candidate G is an appropriate cluster number for L , we can use the cluster number G as an appropriate cluster. The search algorithm based on the log-likelihood function can only work in the static data model but cannot guarantee to work in the time sequential data because of the lack of adaptive parameter for the time sequential data. Thus, it has the following two limitations: 1) it can only work within limitation of the initial dataset and 2) it cannot guarantee the global optimal based on the time sequential data because of the lack of adaptive parameter for the time sequential data. To overcome such static grouping limitations, we introduce distributed grouping using the multichannel (MC) selection for the time sequence data of distributed body sensors in Section III-A.

III. PROPOSED GROUPING CRITERIA WITH DISTRIBUTED SENSORS

The motivation of this section is to prepare interactive relationships among distributed sensory data for clustering, i.e., how to collaborate with distributed measurements to achieve better performances compared to the single measurement. In Section III-A, we will show how to initialize the hyper-parameter presenting a hypothetical prior probability for background knowledge and can select the collaborative cluster number using

EM iteration. In Section III-B, we will calculate switching probability representing the conditional transition probability from channel a to channel b within a cluster number.

A. Collaborative Grouping With Distributed Body Sensors

The cluster number selection using GMM works well in the distributed means model as well as in the static data model. But it only works within limitation of the initial dataset. The tracking estimate system with distributed body sensors has time sequential data. That means the measured information from each sensor can be changed depending on the applications from time to time. To make the collaborative grouping system, we introduce some background knowledge that can be presented as a hypothetical *prior* probability (β_y) that we call a hyper-parameter [41], [42]. Suppose that $\alpha_y(k)$ is an initial *prior* probability at time k . The initial hyper-parameter β_y can be found as follows:

$$\beta_y(0) = \lim_{k \rightarrow \infty} \alpha_y(k). \quad (4)$$

In practice, we can get the hyper-parameter (β_y) using sample training data instead of the infinite training data with respect to time. After calculating the hyper-parameter with sample training data using (4), the hyper-parameter (β_y) should be adaptive with respect to time k . Please note that the hyper-parameter can be selected based on the global information of sample data. This parameter is selected for corresponding to the steady state. It can be accomplished using the switching probability that will be explained in detail in Section IV-C. The adaptive hyper-parameter can be increased or decreased based on the current switching probability comparing to the previous switching probability and can be calculated as follows:

$$\beta_y(k) = \beta_y(k-1) + \Delta\mu^y \quad (5)$$

where $\Delta\mu^y$ is the difference between the current switching probability and the previous one. $\Delta\mu^y$ can be calculated using a switching probability at time k , i.e., $\mu^y(k)$ indicating the switching weight of group y . We will describe how to select the difference ($\Delta\mu^y$) in detail in Section IV-D. After calculating the adaptive hyper-parameter, the adaptive (ADT) *posterior* probability $p_{ADT}(y|z_j)$ is calculated at time k in E-step as follows:

$$p_{ADT}(y|z_j)(k) = \frac{\alpha_y^{(t)} \phi(z_j; m_y^{(t)}, \Sigma_y^{(t)}) + \beta_y(k)}{\sum_{y=1}^G \alpha_y^{(t)} \phi(z_j; m_y^{(t)}, \Sigma_y^{(t)}) + \sum_{y=1}^G \beta_y(k)}. \quad (6)$$

Using the modified one, we can proceed to the M-step at time k as follows:

$$\begin{aligned} \alpha_y^{(t+1)}(k) &= \frac{1}{L} \sum_{j=1}^L p_{ADT}(y|z_j)(k) \\ m_y^{(t+1)}(k) &= \frac{\sum_{j=1}^L p_{ADT}(y|z_j) z_j}{\sum_{j=1}^L p_{ADT}(y|z_j)} = \frac{1}{\alpha_y L} \sum_{j=1}^L p_{ADT}(y|z_j) z_j \end{aligned}$$

$$\begin{aligned} \sum_y^{(t+1)}(k) &= \frac{1}{\alpha_y L} \sum_{j=1}^L p_{ADT}(y|z_j) \\ &\times \left[(z_j - m_y^{(t+1)})(z_j - m_y^{(t+1)})^T \right]. \quad (7) \end{aligned}$$

We can estimate the t th iteration result of the adaptive *posterior* probability $p_{ADT}(y|z_j)$ at time k from (6). Based on the modified result we can calculate the *prior* probability (α_y), the mean (m_y), and the covariance (Σ_y) in the $(t+1)$ th iteration for the collaborative grouping for time sequential data, respectively, using (7). A local maximum at time k can be selected by iterating the above two steps. We can select the collaborative cluster number (G_{ADT}^*) by introducing $\Delta_{ADT}(G)$ that is a log-likelihood function after parameter learning with the following equations:

$$\begin{aligned} G_{ADT}^* &= \arg \min_G \Delta_{ADT}(G) \\ \Delta_{ADT}(G) &= \delta_{ADT}(\Theta^*, G) - \delta_{ADT}(\Theta^*, G-1) \quad (8) \end{aligned}$$

where δ_{ADT} is a log-likelihood function with the adaptive *posterior* probability. Note that (3) is extended into (8) with the hyper-parameter (β_y). Comparing with the previous algorithm, the collaborative grouping with time sequential data can select local maxima at time k by iterating two steps: E-step and M-step. We can select the global optimal from a series of local maxima of time k .

B. Estimated Parameters Used for IMME

Collaborative grouping with time sequence data can select local maxima at time k using the difference (Δ) of the consecutive log-likelihood functions. We can set the difference (Δ) of the consecutive log-likelihood functions as $\Delta(G^*, k)$ with respect to time k . To reduce the notation complexity, $\Delta(k)$ is simply used for $\Delta(G^*, k)$. Now we can use this $\Delta(k)$ for IMME to estimate the MC estimates and covariance. As mentioned, the log-likelihood function in each EM step cannot decrease [51]. This means that we can minimize the difference ($\Delta(k)$) of the consecutive log-likelihood functions with respect to time k because $\Delta(k)$ converges to zero over a period of time. Therefore, we can find out the following relationship: $\Delta(k-1) \geq \Delta(k)$. In the standard IMME, it is assumed that the mixing probability density is a normal distribution all the time. Now we derive the switching probability (μ_{ab}) for the estimated parameter from mixing probability (μ_{ij}). Since it is hard to get $\mu_{ab}(k-1)$ directly, we used a tractable estimation $\mu_{ab}(k-1) + \Delta(k-1)$, as follows:

$$\mu_{ab}(k-1) = \mu_{ij}(k-1) + \Delta(k-1) \quad (9)$$

where μ_{ij} is the mixing probability that represents the conditional transition probability from state i to state j , and μ_{ab} is the switching probability that represents the conditional transition probability from channel a to channel b . Note that we define mixing probability (μ_{ij}) as switching probability (μ_{ab}). This means that our assumption is still valid in the switching probability, i.e., switching probability density follows a normal distribution. The equality of (9) is true because the value of $\Delta(k-1)$ can be zero as k goes to infinity. We can use the right-hand side of (9) to dynamically select the switching probability (μ_{ab}) with

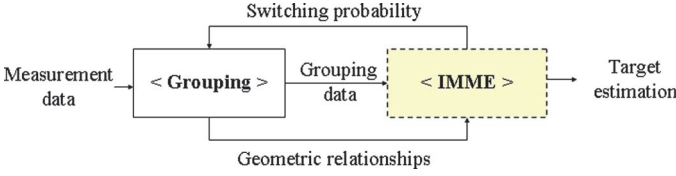


Fig. 3. General block diagram for the proposed MC-IMME.

$\Delta(k)$. Equation (9) above provides us with a method to design the filter for distributed sensors at the second stage, because the switching probability can be adjusted more dynamically based on $\Delta(k)$ in the second stage filter. We will explain how to estimate the MC estimates and covariance using (9) in Section IV.

IV. MC-IMME: PROPOSED SYSTEM DESIGN

The proposed method with collaborative grouping for distributed sensory data can achieve the efficient target estimation by using geometric relationships of target information emerging from distributed measurements. Fig. 3 shows a general block diagram to represent the relationship between the proposed method (MC-IMME) and IMME.

In MC-IMME, grouping data can be used for target-tracking estimation with IMME. Geometric information of distributed measurements is used for the switching probability update in the target estimation. Even though the proposed method needs the initialization process that is the same as in the IMME prediction, the interactive relationship with distributed sensors can compensate for the prediction estimate error. For the interactive tracking estimate, the proposed system design herein can be extended from Figs. 2–4.

A. MC Mixed Initial Condition and the Associated Covariance

Starting with an initial $\bar{x}_a^y(k-1)$ for each channel a in a group y , new filter state is computed to estimate the mixed initial condition and Kalman filter covariance matrices (10) according to the relationships

$$\begin{aligned} \hat{x}_a^{0y}(k-1) &= \sum_{a=1}^r \bar{x}_a^y(k-1) [\mu_{ab}^y(k-1)], \\ P_a^{0y}(k-1) &= \sum_{a=1}^r [\mu_{ab}^y(k-1)] \\ &\quad \times [P_a^y(k-1) + DP_{ab}^y(k-1)] \end{aligned} \quad (10)$$

where μ_{ab}^y is a switching probability presenting the relationship between channel a and channel b within the same group y . As shown in Fig. 4, we have added the blue line indicating how the difference ($\Delta(k)$) in Stage 1 would be used for Stage 2. We denote r as the channel number of the group and $DP_{ab}^y(k-1)$ as an increment to the covariance matrix to account for the difference in the state estimates from channels a and b , expressed by $[\bar{x}_a^y(k-1) - \bar{x}_b^y(k-1)] \cdot [\bar{x}_a^y(k-1) - \bar{x}_b^y(k-1)]^T$.

Note that the initial states of IMME are extended into (10) incorporating with the switching probability and $\Delta(k-1)$. We have adopted the results of Section III on grouping criteria $\Delta(k)$ of (9). The difference ($\Delta(k)$) can be minimized with respect to

time k , so we can adjust to estimate the filter state from a coarse probability to a dense probability.

B. MC Likelihood Update

The above estimates and covariance are used as input to the filter matched to $M_a^y(k)$, which uses $z_a^y(k)$ to yield $\hat{x}_a^y(k)$ and $P_a^y(k)$. The likelihood functions corresponding to each channel are computed using the mixed initial condition and the associated covariance matrix (10) as follows: $\Lambda_a^y(k) = p[z(k) | M_a^y(k), \hat{x}_a^{0y}(k-1), P_a^{0y}(k-1)]$, where y is a group number and $r(y)$ is the number of sensors for each group y . To reduce the notation complexity, r is simply used for $r(y)$.

C. Switching Probability Update

Given the likelihood function ($\Lambda_a^y(k)$) corresponding to each channel, the switching probability update is done as follows:

$$\begin{aligned} \mu^y(k) &= \frac{1}{c_y} \Lambda^y(k) \bar{c}_y, \quad y = 1, \dots, G \\ \Lambda^y &= \frac{1}{r} \sum_{a=1}^r \Lambda_a^y, \quad c_y = \sum_{y=1}^G \Lambda^y \bar{c}_y \end{aligned} \quad (11)$$

where Λ^y is the likelihood function for a group y , c_y is the summarized normalization constant, r is the channel number of a group y , and G is the number of group. Equation (11) above provides the probability matrices used for combination of MC-conditioned estimates and covariance in the next step. It can also show us how to use these parameter results for collaborative grouping criteria with multiple sensors.

D. Feedback From Switching Probability Update to Stage 1 for Grouping Criteria With Distributed Sensors

For the collaborative grouping, we introduced the adaptive hyper-parameter ($\beta_y(k)$) in Section III-A. The adaptive hyper-parameter $\beta_y(k)$ can be dynamically increased or decreased depending on the weight of the channel. The weight of channel can be represented as the switching probability. That means we can use the switching probability ($\mu^y(k)$) as a reference to adjust the adaptive hyper-parameter as follows:

$$\begin{aligned} \beta_y(k) &> \beta_y(k-1) \quad \text{if} \quad \mu^y(k) > \mu^y(k-1) \\ \beta_y(k) &= \beta_y(k-1) \quad \text{if} \quad \mu^y(k) = \mu^y(k-1) \\ \beta_y(k) &< \beta_y(k-1) \quad \text{if} \quad \mu^y(k) < \mu^y(k-1). \end{aligned} \quad (12)$$

If there is no change of the switching probability, $\beta_y(k)$ is the same as $\beta_y(k-1)$. If the current switching probability is greater than the previous one, $\beta_y(k)$ could be increased; otherwise, $\beta_y(k)$ could be decreased as shown in (12). Therefore, we can calculate the difference ($\Delta\mu^y$) between the current switching probability and the previous one as follows:

$$\Delta\mu^y = \mu^y(k) - \mu^y(k-1). \quad (13)$$

That means the adaptive hyper-parameter $\beta_y(k)$ can be increased or decreased based on the current switching probability compared to the previous one. In Fig. 4, we have added the red line indicating how the difference of the switching probability in Stage 2 would be used for Stage 1.

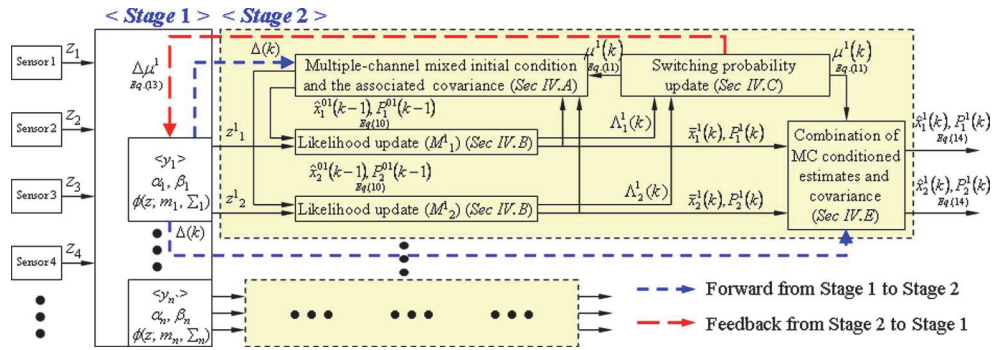


Fig. 4. System design for distributed body sensors has two stages. At the first stage, all of the distributed sensors are partitioned into the groups that have a tracking relationship with each other. At the second stage, the interactive tracking estimate is performed for distributed groups.

TABLE I
COMPARISON OF THE COMPUTATIONAL COMPLEXITY

Methods	KF	IMME	MC-IMME
Complexity	$O(L \times k \times N^3)$	$O(L \times k \times T(N))$	$O(L \times k \times T(N))$

E. Combination of MC Conditioned Estimates and Covariance

Combination of the MC conditioned estimates and covariance is done according to the mixture equations

$$\begin{aligned} \hat{x}_a^y(k) &= \sum_{b=1}^r \bar{x}_b^y(k) [\mu_{ab}^y(k)] \\ P_a^y(k) &= \sum_{b=1}^r [\mu_{ab}^y(k) [P_b^y(k) + DP_b^y(k)]] \end{aligned} \quad (14)$$

where μ_{ab}^y is a switching probability presenting the relationship between channel a and channel b within the same group y , and $DP_b^y(k)$ as an increment to the covariance matrix to account for the difference between the intermediate state and the state estimate from model b , expressed by $[\bar{x}_b^y(k) - \hat{x}_b^y(k)] \cdot [\bar{x}_b^y(k) - \hat{x}_b^y(k)]^T$.

Note that the combination of the model-conditioned estimates and covariance matrices in Fig. 2 is extended into (14) incorporating with the switching probability and $\Delta(k)$. As can be seen in Section IV-A, we also have adopted the results of Section III on grouping criteria $\Delta(k)$ of (9). In Fig. 4, the entire flow chart illustrates the idea of MC-IMME proposed in this study. We have added the blue line indicating how the difference ($\Delta(k)$) in Stage 1 would be used for the IMME outcomes of Stage 2, corresponding to (14). This combination is only for output purposes.

F. Computational Time

We have evaluated how much additional computational time is required when we implement the proposed method by comparing it to the standard KF and the IMME method in Table I.

The computational complexity of the standard KF for the upper bound is orders of growth N^3 , where N represents the states estimated using KF derived by Karlsson *et al.* [44]. The computational complexity can be increased as a linear function of the sensor number (L) and time k . Accordingly, the asymptotic upper bound of KF is orders of growth $L \times k \times N^3$.

IMME extends the complexity by defining $T(N)$ as the asymptotic upper bound of recursive computation based on the states estimated using IMME. In the IMME the computational complexity is increased as a linear function of the independent sensor number (L). In addition, IMME needs recursive computation based on time k . Therefore, the asymptotic upper bound for the worst-case running time of IMME is orders of growth $L \times k \times T(N)$ [52].

Let us define $T(L)$ as a upper bound of iteration execution time for k -means clustering based on L points. Har-Peled *et al.* showed that the k -means heuristic needs orders of growth L iterations for L points in the worst case [35]. In addition, the adaptive grouping method needs to calculate the difference ($\Delta_{ADT}(G)$) of the consecutive log-likelihood functions based on time sequential data (k) for the appropriate group number selection. Therefore, the upper bound for the worst-case running time of the adaptive grouping method is orders of growth $L \times k \times T(L)$.

MC-IMME uses the same recursive computation as IMME with respect to the estimate states. This means that the running time of stage 2 is the same as that of IMME. and MC-IMME needs additional computation for the first stage to make grouping. Suppose that the asymptotic upper bound of recursive computation ($T(N)$) is equal to the upper bound of iteration execution time for k -means clustering ($T(L)$). Then, the asymptotic upper bound for the computational complexity of MC is orders of growth $L \times k \times T(N)$, because both stages 1 and 2 have the same orders of growth $L \times k$ [52]. Please note that IMME and MC-IMME have the identical computational complexity since distributed sensory systems both have the same channel number L and the same data length k .

V. EXPERIMENTAL RESULTS

The motivation of this section is to validate the proposed MC-IMME with comprehensive experimental results. In Section V-A, we will describe the target motions for the experimental tests (chest, head, and upper body). For each target motion, the optimal cluster number based on the proposed grouping method is selected follows in Section V-B, and this selection number is further investigated in comparison to grouping number methods using other clustering techniques in Section V-C. The prediction accuracy of the proposed MC-IMME is evaluated with the normalized root mean-squared

TABLE II
CHARACTERISTICS OF MOTION DATA WITH
SHORT LENGTH AND LONG LENGTH

Motion Data	Motion Type	Speed (cm/sec)	Recording Time (sec)
Chest_S	Slow motion	0.64 – 0.87	20.72
Chest_M	Moderate motion	6.70 – 7.91	20.40
Chest_V	Violent motion	24.84 – 32.63	22.18
Chest_C	Complex motion	1.46 – 32.54	105.8
Head_S	Slow motion	0.63 – 1.08	20.36
Head_M	Moderate motion	6.62 – 8.37	20.40
Head_V	Violent motion	16.07 – 67.70	21.43
Head_C	Complex motion	1.08 – 66.16	104.05
Upper Body_S	Slow motion	0.68 – 1.48	20.83
Upper Body_M	Moderate motion	3.64 – 28.08	20.64
Upper Body_V	Violent motion	38.48 – 118.18	21.03
Upper Body_C	Complex motion	1.36 – 117.98	104.15

error (NRMSE) and the prediction overshoots, in Sections V-D and V-E, respectively. We also show CPU time used for the computational time in Section V-F.

A. Motion Data

We have used three kinds of motion data, i.e., chest motion, head motion, and upper body motion. Motion data were collected using a Polhemus Liberty ac magnetic, operating at 240 Hz for approximately 20 and 100 s.¹ Eight sensors were attached on the target motion surface with the magnetic source rigidly mounted approximately 25.4 cm from the sensors. Each motion data was randomly selected based on the motion speed for Monte Carlo analysis with four sets of motion data—the first datasets for slow motion, the second datasets for moderate, the third for the violent motion, and the fourth for the complex motion including slow, moderate, and violent motions. For the target estimation, the experimental tests have been conducted based on repeated random sampling to compute their results for Monte Carlo analysis. Each of the datasets was taken with great care to limit target movement to the type based on Table II.

B. Collaborative Grouping Initialization

Before the efficient target tracking, the proposed collaborative method needs to make the grouping for distributed sensory data. The objective of this section is to find out the optimal group number with an adaptive hyper-parameter. First, we need to find out the initial hyper-parameter (β_y) in Section V-B1 and then calculate the group number (G) based on the adaptive (ADT) posterior probability $p_{ADT}(y|z_j)$. Sections V-B2 and V-B3 compared the difference (Δ) of the consecutive log-likelihood functions between noncollaborative grouping method described in Section II-B and collaborative grouping method described in Section III-A.

1) *Calculation of Hyper-Parameter (β_y):* The objective of this section is to calculate the initial hyper-parameter (β_y) with potential group numbers. To find out the hyper-parameter, we

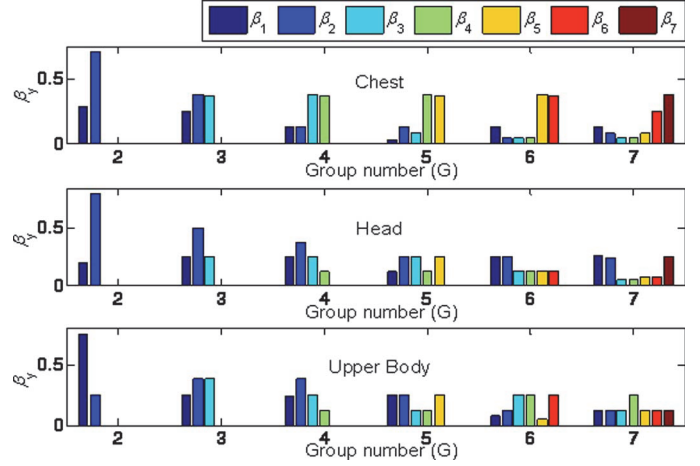


Fig. 5. Hyper-parameter values based on the motion data and group number.

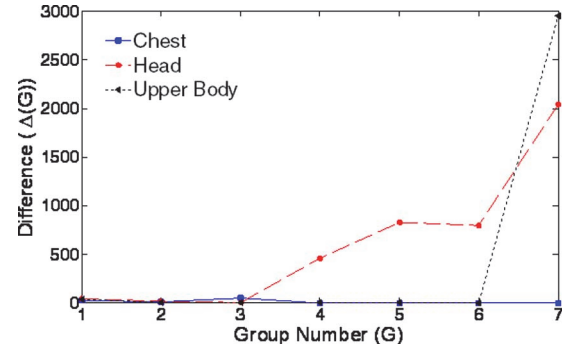


Fig. 6. The difference ($\Delta(G)$) of the consecutive log-likelihood with non-collaborative grouping.

iterate expectation and maximization steps with sample training data (approximately a 2400-sample dataset) for each motion. We increased the group number (G) of the EM process from two to seven to show all of the potential hyper-parameters. Please note that we have eight sensory channels, so that we can show all available group numbers.

Fig. 5 shows the hyper-parameters (β_y , where y is a group number) described in (4), based on the target motion data and group number (G). Given in Fig. 5, we can notice that the higher the group number, the bigger the iteration number; and the more even the group distribution probabilities of a sample training data, the smaller the iteration number.

2) *Calculation of the Difference (Δ) of the Consecutive Log-Likelihood With Noncollaborative Grouping:* The objective of this section is to find an optimal cluster number (G^*) with the consecutive log-likelihood functions (3) based on EM process. Fig. 6 below shows the difference ($\Delta(G)$) of the consecutive log-likelihood functions, described in (3). For example, when $G = 2$, we calculate all of the log-likelihood functions of EM operations and then select the minimum as a representing value in Fig. 6. We iterate the same procedure with different group number ($G = 2, \dots, 7$) in the three kinds of the motion data. We expect to find out, as described in Section II-B, the minimum of $\Delta(G)$.

Given the results in Fig. 6, we may select the group number G^* for the three datasets: 2, 4, or 6 for Chest; 2 or 3 for Head;

¹[Online]. Available: <http://www.polhemus.com/>

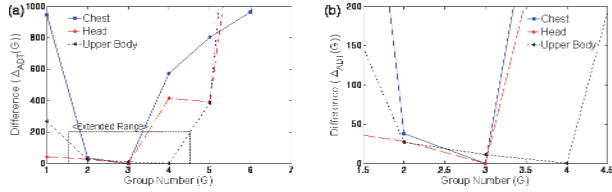


Fig. 7. Difference ($\Delta_{\text{ADT}}(G)$) with collaborative grouping: (a) whole range and (b) extended range.

and 2, 4, 5, or 6 for Upper Body. As the group numbers are increased, the differences start to become drastically greater.

However, we cannot identify the least minimum number; for example, it is hard to choose among 2, 4, 5, or 6 for Upper Body. Therefore, in the next experiment, we will recalculate the difference ($\Delta_{\text{ADT}}(G)$) of the consecutive log-likelihood with collaborative grouping.

3) *Calculation of the Difference (Δ) of the Consecutive Log-Likelihood With Collaborative Grouping:* The objective of this section is to find an optimal cluster number (G^*) using log-likelihood function with the adaptive *posterior* probability (8). Based on the initial hyper-parameter (β_y) of Section V-B1), we can calculate the adaptive (ADT) *posterior* probability $p_{\text{ADT}}(y|z_j)$ and iterate E-step (6) and M-step (7) with a specific group number (G).

Now we can show the difference ($\Delta_{\text{ADT}}(G)$) of the consecutive log-likelihood functions, described in (8) of Section III-A, with the adaptive *posterior* probability in the three kinds of the motion data. We applied (8) for the minimum value of $\Delta_{\text{ADT}}(G)$. We iterate the same procedure with a different group number ($G = 2, \dots, 7$), as shown in Fig. 7.

In Fig. 7, we can select the group number G^* for the three datasets: 3 for Chest; 3 for Head; and 4 for Upper Body. Compared with Figs. 6 and 7, it is clear that the collaborative grouping provides more distinct difference $\Delta_{\text{ADT}}(G)$ of grouping numbers; for example, while Fig. 6 had the candidates of the group numbers 2, 4, 5, or 6 for Upper Body, Fig. 7 now identifies the minimum number 4 for Upper Body by introducing the adaptive *posterior* probability.

C. Comparison of Grouping Number Methods Using Other Clustering Techniques

To find out the best grouping numbers, we have evaluated several clustering algorithms: *k*-means [33], spectral clustering [37], [38], nonparametric Bayesian inference [55], and EM algorithm [31]. To determine the quality of group number hypothesis, we would like to show established metrics, i.e., Akaike's Information Criterion (AIC) that provides a measure of model quality by simulating a statistical model for model selection [56]. For this selection, we assume that the model errors are normally and independently distributed and that the variance of the model errors is unknown but equal for them all.

Let n be the number of training observations. The formula AIC can be expressed as a simple function of the residual sum of squares (RSS), i.e., $\text{AIC} = 2k + n[\ln(\text{RSS}/n)]$, where k and RSS are the number of parameters in the statistical model and

TABLE III
COMPARISON OF GROUPING NUMBER METHODS WITH AIC VALUES

	<i>k</i> -means	Spectral clustering	Nonparametric Bayesian	EM algorithm
Chest	G=2 7444	7411	7346	7404
	G=3 7393	6328	6942	7379
	G=4 7608	6356	7523	7603
	G=5 7824	6977	7383	7550
	G=6 7674	7365	7662	7680
Head	G=7 7761	7177	7514	7497
	G=2 6272	6272	6284	6256
	G=3 6222	6314	5847	6220
	G=4 6783	6509	6500	6770
	G=5 6677	6455	6337	6305
Upper Body	G=6 6427	6512	6325	6529
	G=7 6711	6471	6402	6530
	G=2 10874	10885	10760	10827
	G=3 11043	10967	10645	10780
	G=4 10809	10874	10617	10448
	G=5 10962	10928	10757	10928
	G=6 10941	10987	10938	10987
	G=7 11127	10901	10876	10861

the residual sum of squares ($\sum_{i=1}^n \epsilon_i^2$, ϵ_i : estimated residuals for a candidate model), respectively [56, Ch. 2.2]. Given any estimated models, the model with the minimum value of AIC is the one to be preferred.

We set the number of training observations to $n = 1000$ for all of the datasets. Table III shows the comparison of grouping number methods with AIC values. We can notice that all of the methods except the spectral clustering method have selected the identical grouping numbers: $G = 3$ for Chest datasets, $G = 3$ for Head datasets, and $G = 4$ for Upper Body. Please note that all of the grouping number methods have the minimum AIC values for Chest ($G = 3$) and Upper Body ($G = 4$) datasets. In Head datasets, there exists inconsistency among the methods. That means head motion can be classified into different groups in the given datasets. For our tracking estimation, we use grouping number $G = 3$ for Head datasets because of the minimum AIC value in the given results.

D. MC-IMME

Based on the group number (G^*) chosen in the experiment in Section V-B of the first stage, we can perform the target estimation using MC-IMME of the second stage with respect to each group.

1) *Position Estimation:* The comparison with the standard KF is very hard to justify since many KF variations had been proposed. Accordingly, we compare the performance of motion tracking estimation among particle filter [57], Ensemble KF [58], IMME, and MC-IMME. Fig. 8 shows that MC-IMME can estimate the target motion more accurately than other tracking methods, particle filter, Ensemble KF, and IMME at the initial stage.

In addition, we compare the accumulated position errors for each channel across the entire measurement period among particle filter, Ensemble KF, IMME, and MC-IMME. Fig. 9 shows that the accumulated position errors of particle filter, Ensemble KF, and IMME are greater than those of MC-IMME for Head_S

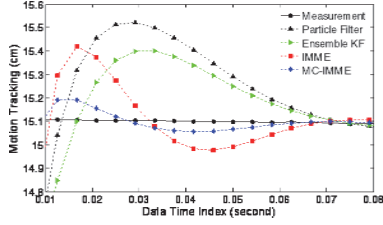


Fig. 8. Comparison of motion tracking estimation for Head_S dataset.

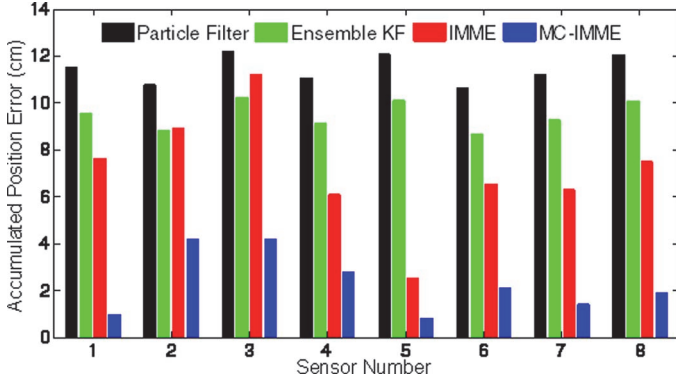


Fig. 9. Comparison of accumulated position error of each channel for Head_S.

dataset for each sensor channel. We can notice that MC-IMME outperforms IMME by 28.82% in the complex Head motion.

Table IV shows the overall performance of accumulated error among the datasets listed in Table II. As shown in Table IV, MC-IMME can show the improvement with comparison to particle filter and Ensemble KF. In addition, the proposed method outperforms IMME around by 25.61%–40.47% in the chest motion, 8.86%–52.87% in the head motion, and 42.94%–50.94% in the aggressive motion. Please note that the proposed method can achieve 50.94% improvement over IMME in the upper body motion.

2) *Prediction Time Horizon*: For the prediction accuracy, we changed the prediction time horizon. Here, prediction time horizon is the term to represent the time interval window to predict the future sensory signal. We would like to compare the error performance with the various prediction time horizons among particle filter, Ensemble KF, IMME, and MC-IMME in Table V. For the comparison, we used a normalization that is the normalized root mean squared error (NRMSE) between the predicted and actual signal over all the samples in the test datasets.

In Table V, the prediction accuracy of Chest_S dataset in the proposed MC-IMME was improved by 49.67% for particle filter, 40.16% for Ensemble KF, and 36.10% for IMME of the average prediction time horizon. We can notice that the proposed method outperforms particle filter, Ensemble KF and IMME in the other Chest motion datasets as well, even though the average improvements were less than 17% with comparison to IMME. The average improvements were 28.14% for particle file, 18.55% for Ensemble KF, and 14.46% for IMME.

In the Head_S dataset, the prediction accuracy was significantly improved by 76.73% for particle filter, 74.39% for Ensemble KF, and 73.50% for IMME of the average prediction

time horizon. Notice that the improvement of prediction accuracy for the proposed method maintained around 15% across the prediction time horizons. In the other Head motion datasets, the proposed method can improve other methods, even though the average improvements were less than 5% in comparison to IMME. The average improvements were 33.93% for particle filter, 26.61% for Ensemble KF, and 23.54% for IMME.

In the Upper_Body_S dataset, the prediction accuracy of the proposed method was improved by 61.12% for particle filter, 55.30% for Ensemble KF and 52.95% for IMME of the average prediction time horizon. We can notice that the improvement of MC-IMME maintained around 40% across the prediction time horizons. We can also notice that the proposed method outperforms particle filter for 28.72%, Ensemble KF for 20.54%, and IMME for 17.33% of the average prediction time horizon over all the datasets, even though the average improvements were less than 6.39% for Upper_Body_M, 3.90% for Upper_Body_V, and 6.09 for Upper_Body_C in comparison to IMME.

3) *Effect of the Feedback/Forward Method*: We would like to show the advantage of the proposed feedback/forward method by comparing the performance of velocity estimation of MC-IMME with no feedback/forward versus feedback/forward. We have evaluated the tracking performance of the average velocity for the Chest_V dataset in Fig. 10. We have observed that the feedback/forward method slightly increases the prediction accuracy of MC-IMME by 14%.

We show all nine datasets to compare the overall performance of velocity error averaged among eight channels between no feedback/forward versus feedback/forward. Table VI shows the overall performance of velocity error among the datasets listed in Table II. Given in Table VI, feedback/forward method outperforms no feedback/forward method around 15%–37% for Chest dataset, 20%–26% for Head dataset, and 14%–22% for Upper Body dataset.

E. Prediction Overshoot

We define overshoot for cases in which the predicted output exceeds a certain marginal value with confidence levels corresponding to the tolerances [26]. The initialization process is an essential step of KF-based target tracking. Unfortunately, this process produces an unexpected prediction estimate error. To compensate for the prediction estimate error, we used a marginal value to generate a 95% prediction interval for the measurement prediction, so that we can define the upper bound and the lower bound by adding the marginal value to the measurement and subtracting the marginal value from the measurement, respectively [26].

Fig. 11 shows the prediction overshoot comparison between IMME and MC-IMME. We can notice that the prediction overshoot error with distributed sensory data was improved in the average of 10.02% with slow motion, 12.47% with moderate motion, 34.67% with violent motion, and 15.48% with complex motion. Moreover, the total error of MC-IMME was decreased by 18.16% in comparison with that of IMME.

Table VII shows the comparison of overshoot dataset sample numbers between IMME and MC-IMME, where the second column represents the average number of overshoot dataset

TABLE IV
OVERALL PERFORMANCE OF ACCUMULATED ERROR AMONG THE DATASETS LISTED IN TABLE II

		Chest_S	Chest_M	Chest_V	Chest_C	Head_S	Head_M	Head_V	Head_C	Body_S	Body_M	Body_V	Body_C
PF	Min.	4.79	14.96	56.92	148.67	8.20	9.28	13.97	55.06	10.92	11.94	120.45	279.50
	Avg.	5.19	16.20	46.25	130.28	9.00	10.51	49.81	130.01	13.38	39.42	200.26	498.58
	Max.	6.12	17.47	38.53	118.12	9.76	11.43	130.14	294.95	16.31	125.48	636.21	1549.62
EnKF	Min.	3.84	12.51	35.87	101.86	4.78	5.86	11.92	42.53	3.42	6.06	40.87	53.12
	Avg.	4.24	13.75	43.59	119.02	5.58	7.09	47.76	116.95	5.88	31.98	192.69	459.23
	Max.	5.17	15.03	54.26	144.19	6.34	8.00	128.08	283.61	8.81	118.05	543.79	1339.41
IMME	Min.	2.92	6.53	15.97	45.89	2.49	2.32	9.28	25.83	2.63	3.67	21.64	51.26
	Avg.	3.67	11.54	18.19	63.72	6.98	7.32	23.25	86.34	6.59	50.45	166.13	467.86
	Max.	5.75	14.29	21.58	78.91	10.12	9.78	92.51	248.99	9.75	120.24	261.89	847.67
MC-IMME	Min.	2.14	3.26	8.43	25.51	1.87	1.84	5.54	12.69	1.13	1.18	9.17	25.55
	Avg.	2.73	6.87	10.95	40.44	3.29	4.38	21.19	61.46	3.79	24.75	108.87	352.93
	Max.	3.87	10.31	12.39	48.44	4.19	6.12	51.16	134.33	7.89	87.95	181.78	623.37

(Unit: Accumulated error (cm))

TABLE V
ERROR PERFORMANCE AMONG PREDICTION TIME HORIZON

	Window	Chest_S	Chest_M	Chest_V	Chest_C	Head_S	Head_M	Head_V	Head_C	Body_S	Body_M	Body_V	Body_C
PF	100ms	0.102	0.229	0.449	0.302	0.129	0.215	0.649	0.387	0.057	0.196	0.958	0.489
	200ms	0.190	0.521	0.508	0.469	0.189	0.491	0.688	0.521	0.092	0.300	1.000	0.557
	300ms	0.274	0.590	0.498	0.504	0.239	0.551	0.763	0.586	0.123	0.386	1.022	0.598
	400ms	0.338	0.623	0.529	0.548	0.285	0.582	0.695	0.592	0.149	0.504	0.963	0.631
EnKF	100ms	0.086	0.153	0.412	0.251	0.119	0.143	0.596	0.334	0.051	0.171	0.898	0.451
	200ms	0.151	0.472	0.467	0.417	0.173	0.446	0.631	0.476	0.080	0.263	0.932	0.514
	300ms	0.231	0.544	0.459	0.461	0.217	0.505	0.708	0.535	0.107	0.327	0.944	0.553
	400ms	0.294	0.567	0.479	0.491	0.255	0.533	0.646	0.578	0.128	0.431	0.892	0.575
IMME	100ms	0.081	0.127	0.400	0.243	0.116	0.119	0.579	0.342	0.049	0.163	0.878	0.442
	200ms	0.138	0.455	0.454	0.405	0.168	0.431	0.612	0.485	0.076	0.250	0.909	0.497
	300ms	0.216	0.528	0.446	0.451	0.209	0.489	0.690	0.531	0.102	0.307	0.918	0.521
	400ms	0.280	0.549	0.463	0.473	0.245	0.516	0.629	0.570	0.121	0.407	0.869	0.553
MC-IMME	100ms	0.028	0.102	0.335	0.218	0.025	0.092	0.511	0.264	0.014	0.146	0.796	0.398
	200ms	0.085	0.440	0.452	0.397	0.043	0.425	0.607	0.462	0.033	0.235	0.898	0.476
	300ms	0.167	0.505	0.443	0.435	0.059	0.481	0.684	0.504	0.057	0.291	0.900	0.504
	400ms	0.230	0.517	0.457	0.451	0.075	0.504	0.623	0.546	0.073	0.391	0.842	0.538

(Unit: NRMSE value)

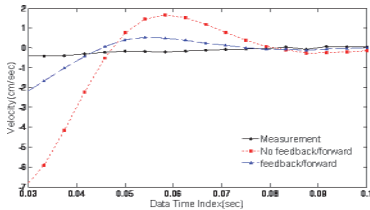


Fig. 10. Comparison of the velocity estimations of MC-IMME with no feedback/forward versus feedback/forward.

samples listed in Table II. The overall improvement in the benign motion is around 10%, whereas the overall improvement in the aggressive motion is over 20%. This means that distributed sensory data can reduce the prediction estimate error at the beginning of target tracking. We may expect this prediction accuracy to decrease for different datasets (e.g., including head and chest motions) due to the lack of interactive relationships. For our experimental tests, however, we have focused on the human body motion including head and chest. That means our experimental results can be generalized to the upper body.

F. Computational Time

Regarding CPU experimental time, we have evaluated the overall performance of average CPU time used for the datasets listed in Table II. We have collected the motion data using a Polhemus Liberty ac magnetic tracker with eight sensors and

TABLE VI
COMPARISON OF OVERALL VELOCITY ERROR AVERAGED AMONG 8 CHANNELS BETWEEN NO FEEDBACK/FORWARD VERSUS. FEEDBACK/FORWARD

Datasets	No feedback/forward	Feedback/forward
Chest_S	0.415	0.292
Chest_M	0.335	0.211
Chest_V	0.605	0.514
Chest_C	0.485	0.384
Head_S	1.527	1.168
Head_M	1.386	1.014
Head_V	1.517	1.201
Head_C	1.497	1.212
Upper Body_S	2.012	1.550
Upper Body_M	3.162	2.572
Upper Body_V	3.999	3.404
Upper Body_C	3.627	3.270

(Unit: cm/sec)

then conducted the experimental test for the computational complexity with offline. We have implemented the proposed method with MATLAB language using a PC of Pentium core 2.4 GHz with RAM 3.25 GB.

In Table VIII, we evaluated the individual dataset to compare particle filter, Ensemble KF, and IMME methods with MC-IMME. Table VIII shows the overall performance of CPU time used among the datasets. For the comparison of the different target-tracking methods, we evaluated the computational

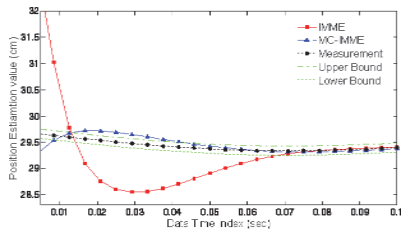


Fig. 11. Prediction overshoot comparison between IMME and MC-IMME. The prediction overshoot error can be improved with MC-IMME.

TABLE VII
PREDICTION OVERSHOOT COMPARISON LISTED IN TABLE II

Datasets	Average number of overshoot dataset (IMME/MC-IMME)	Improvement (%)
Chest_S	15.00/13.37	10.87
Chest_M	15.25/13.12	13.97
Chest_V	26.00/12.75	50.96
Chest_C	29.20/23.92	18.08
Head_S	15.00/13.75	8.33
Head_M	14.75/13.62	7.66
Head_V	62.12/42.50	31.58
Head_C	44.06/36.33	17.55
Upper Body_S	15.00/13.37	10.87
Upper Body_M	27.75/23.37	15.78
Upper Body_V	99.62/78.25	21.45
Upper Body_C	79.10/70.53	10.83

(Unit: overshoot dataset #)

TABLE VIII
CPU TIME USED AMONG THE DATASETS

Datasets	PF	EnKF	IMME	MC-IMME
Chest	0.4135	0.563	0.957	0.802
Head	0.4055	0.565	0.966	0.804
Upper Body	0.4155	0.582	0.974	0.829

(Unit: ms/sample numbers)

time calculating target-tracking estimate filters. This means that we only counted the calculation time for particle filter, Ensemble KF, and IMME operations with all the methods. Note that MC-IMME can improve approximately 16% of the average computational time in comparison with IMME, even though it requires more than twice the computational time of the particle filter and Ensemble KF, as shown in Table VIII. An interesting result is that the proposed method can improve the computational time over IMME. We think that the actual difference for CPU time used in Table VIII mainly comes from the simultaneous calculation of distributed sensory data in MC-IMME. In IMME, it needs to calculate target-tracking estimation individually, whereas MC-IMME can evaluate a couple sets of target estimation simultaneously.

VI. CONCLUSION

In this paper, we have presented a new MC-IMME and grouping criteria with distributed sensors placement. Our new method has two main contributions to improve the traditional

IMME-based target tracking. The first contribution is to comprehensively organize the distributed channel sensory process by providing a collaborative grouping number with the given datasets to achieve the efficient target estimation. The second contribution is to add feedback/forward modules to import the results from the first multiple channels grouping for interactive tracking estimation to employ a tracking relationship with each other.

The experiment results validated that we can identify a proper group number with the collaborative grouping method using hyper-parameter and the collaborative grouping method can outperform the conventional target-tracking methods, e.g., particle filter, Ensemble KF, and IMME, by comparing the prediction overshoot and the prediction accuracy of target tracking with respect to the accumulated position error. We have also evaluated that MC-IMME with feedback/forward method can increase the prediction accuracy of MC-IMME throughout the experiment results. The prediction overshoot ratio at the beginning of target tracking can be improved in the average of 19.31% with employing a tracking relationship in this specific datasets. For the generalized extent of motion tracking, more complicated motions and different sensory positions are required. As a future research, this research work may be extended with human pattern detection for markerless tracking. For the real environment without any special markers the proposed method should be integrated with human pattern detection based on feature analysis.

REFERENCES

- [1] H. Ghasemzadeh and R. Jafari, "Physical movement monitoring using body sensor networks: A phonological approach to construct spatial decision trees," *IEEE Trans. Ind. Inf.*, vol. 7, no. 1, pp. 66–77, Feb. 2011.
- [2] S. J. Lee, Y. Motai, and M. Murphy, "Respiratory motion estimation with hybrid implementation of extended Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4421–4432, Nov. 2012.
- [3] H. Chen and Y. Li, "Enhanced particles with pseudolikelihoods for three-dimensional tracking," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 2992–2997, Aug. 2009.
- [4] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1696–1710, Dec. 2006.
- [5] D. Naso, B. Turchiano, and P. Pantaleo, "A fuzzy-logic based optical sensor for online weld defect-detection," *IEEE Trans. Ind. Inf.*, vol. 1, no. 4, pp. 259–273, Nov. 2005.
- [6] T. Mukai and M. Ishikawa, "An active sensing method using estimated errors for multisensor fusion systems," *IEEE Trans. Ind. Electron.*, vol. 43, no. 3, pp. 380–386, Mar. 1996.
- [7] J. Liu, J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed state representation for tracking problems in sensor networks," in *Proc. Inf. Process. Sensor Networks*, 2004, pp. 234–242.
- [8] K. Zhou and S. I. Roumeliotis, "Optimal motion strategies for range-only constrained multisensor target tracking," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1168–1185, Oct. 2008.
- [9] J. G. Garcia, J. G. Ortega, A. S. Garcia, and S. S. Martinez, "Robotic software architecture for multisensor fusion system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 766–777, Mar. 2009.
- [10] N. Bellotto and H. Hu, "Multisensor-Based human detection and tracking for mobile service robots," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 39, no. 1, pp. 167–181, Feb. 2009.
- [11] T. Kirubarajan, H. Wang, Y. Bar-Shalom, and K. R. Pattipati, "Efficient multisensor fusion using multidimensional data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 386–400, Apr. 2001.
- [12] Z. Khan, T. Balch, and F. Dellaert, "MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1960–1972, Dec. 2006.

- [13] L. Hong, S. Cong, and D. Wicker, "Distributed multirate interacting multiple model fusion (DMRIMMF) with application to out-of-sequence GMTI data," *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 102–107, Jan.–Feb. 2004.
- [14] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA, USA: Artech House, 1999.
- [15] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking—Part V: Multiple model methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.
- [16] A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.
- [17] A. K. Jana, "A hybrid FLC-EKF scheme for temperature control of a refinery debutanizer column," *IEEE Trans. Ind. Inf.*, vol. 6, no. 1, pp. 25–35, Feb. 2010.
- [18] L. C. Yang, J. H. Yang, and E. M. Feron, "Multiple model estimation for improving conflict detection algorithms," in *Proc. IEEE Conf. Syst., Man Cybern.*, 2004, vol. 1, pp. 242–249.
- [19] H. Bom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.
- [20] L. Campo, P. Mookerjee, and Y. Bar-Shalom, "State estimation for systems with sojourn-time-dependent Markov model switching," *IEEE Trans. Autom. Control*, vol. 36, no. 2, pp. 238–243, Feb. 1991.
- [21] X. R. Li and Y. Zhang, "Numerically robust implementation of multiple-model algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 266–278, Jan. 2000.
- [22] X. R. Li, Z. Zhao, and X. Li, "General model-set design methods for multiple-model approach," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1269–1276, Sep. 2005.
- [23] L. Hong, "Multirate interacting multiple model filtering for target tracking using multirate models," *IEEE Trans. Autom. Control*, vol. 44, no. 7, pp. 1326–1340, Jul. 1999.
- [24] W. Farrell, "Interacting multiple model filter for tactical ballistic missile tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 418–426, Apr. 2008.
- [25] X. R. Li and Y. Bar-Shalom, "Performance prediction of the interacting multiple model algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 3, pp. 755–771, Jul. 1993.
- [26] J. G. Ramírez, *Statistical Intervals: Confidence, Prediction, Enclosure*. Cary, NC, USA: SAS Institute Inc., 2009.
- [27] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Mach. Learning*, vol. 56, pp. 9–33, 2004.
- [28] L. Xu, "How many clusters?: A YING-YANG machine based theory for a classical open problem in pattern recognition," in *Proc. IEEE Int. Conf. Neural Networks*, 1996, vol. 3, pp. 1546–1551.
- [29] L. Xu, "Bayesian ying-yang machine, clustering and number of clusters," *Pattern Recognit. Lett.*, vol. 18, pp. 1167–1178, 1997.
- [30] P. Guo, C. L. P. Chen, and M. R. Lyu, "Cluster number selection for a small set of samples using the Bayesian Ying-Yang model," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 757–763, May 2002.
- [31] N. Vlassis and A. Likas, "A greedy EM algorithm for Gaussian mixture learning," *Neural Process. Lett.*, vol. 15, no. 1, pp. 77–87, 2002.
- [32] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1344–1348, Aug. 2005.
- [33] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-Means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [34] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou, "Robust sequential data modeling using an outlier tolerant hidden Markov model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1657–1669, Sep. 2009.
- [35] S. Har-Peled and B. Sadri, "How fast is the k-means method," *Algorithmica*, vol. 41, no. 3, pp. 185–202, 2005.
- [36] R. Nock and F. Nielsen, "On weighting clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1223–1235, 2006.
- [37] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process.*, vol. 14, pp. 849–856, 2002.
- [38] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [39] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian inference for dynamic models with Dirichlet process mixtures," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 71–84, Jan. 2008.
- [40] S. Kim, P. Smyth, and H. Stern, "A Bayesian mixture approach to modeling spatial activation patterns in multisite fMRI data," *IEEE Trans. Med. Imag.*, vol. 29, no. 6, pp. 1260–1274, Jun. 2010.
- [41] M. Ramoni, P. Sebastiani, and P. Cohen, "Bayesian clustering by dynamics," *Mach. Learning*, vol. 47, no. 1, pp. 91–121, 2001.
- [42] R. L. Streit and P. K. Willett, "Detection of random transient signals via hyperparameter estimation," *IEEE Trans. Signal Process.*, vol. 47, no. 7, pp. 1823–1834, Jul. 1999.
- [43] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.
- [44] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, Nov. 2005.
- [45] H. Himberg and Y. Motai, "Head orientation prediction: Delta quaternions versus quaternions," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1382–1392, Jun. 2009.
- [46] M. H. Kim, S. Lee, and K. C. Lee, "Kalman predictive redundancy system for fault tolerance of safety-critical systems," *IEEE Trans. Ind. Inf.*, vol. 6, no. 1, pp. 46–53, Feb. 2010.
- [47] V. P. Jilkov and X. R. Li, "Bayesian estimation of transition probabilities for Markovian jump systems by stochastic simulation," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1620–1630, Jun. 2004.
- [48] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.
- [49] G. McLachlan and D. Peel, *Finite Mixture Models*. New York, NY, USA: Wiley, 2000.
- [50] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc. B*, vol. 39, no. 1–38, pp. 1–1, 1977.
- [51] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. New York, NY, USA: Marcel Dekker, 1988.
- [52] J. Kleinberg and É. Tardos, *Algorithm Design*. Boston, MA, USA: Pearson Education, 2006, ch. 2.
- [53] X.-R. Li and B.-S. Yaakov, "Multiple-Model estimation with variable structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493, Apr. 1996.
- [54] X. R. Li, X. R. Zhi, and Y. M. Zhang, "Multiple-model estimation with variable structure-Part III: Model-group switching algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 1, pp. 225–241, Jan. 1999.
- [55] P. Orbanz and Y. W. Teh, "Bayesian nonparametric models," in *Encyclopedia of Machine Learning*. Berlin, Germany: Springer, 2010.
- [56] K. P. Burnham and D. R. Anderson, *Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach*. New York, NY, USA: Springer-Verlag, 2002.
- [57] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, pp. 5–28, 1998.
- [58] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dynam.*, vol. 53, pp. 343–367, 2003.



network, target estimate, and classification



intelligence, particularly in medical imaging, pattern recognition, computer vision, and sensory-based robotics.

Suk Jin Lee (S'11) received the B.Eng. degree in electronic engineering and M.Eng. degree in telematics engineering from Pukyong National University, Busan, Korea, in 2003 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from Virginia Commonwealth University, Richmond, VA, USA, in 2012.

In 2007, he was a Visiting Research Scientist with GW Center for Networks Research, George Washington University, Washington, DC, USA. His research interests include network protocols, neural

Yuichi Motai (M'01) received the B.Eng. degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991, the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA., in 2002.

He is currently an Assistant Professor of electrical and computer engineering at Virginia Commonwealth University, Richmond, VA, USA. His research interests include the broad area of sensory intelligence, particularly in medical imaging, pattern recognition, computer vision, and sensory-based robotics.



Hongsik Choi received the Ph.D. degree in computer science from George Washington University, Washington, DC, USA, in 1996.

He was a Faculty Member of several universities, including Hallym University, George Washington University, and Virginia Commonwealth University. He has been a member of Information Technology Discipline Faculty at the Georgia Gwinnet College, Lawrenceville, GA, USA, where he is currently an Associate Professor.