Stream Segmentation — A Data Fusion Approach for Sensor Networks

WU Jian-Kang[1,2] DONG Liang[1] BAO Xiao-Ming[1]

[1](*Institute for Infocomm Research, Singapore* 119613, *Singapore*)
[2](*Graduate School, Chinese Academy of Sciences, Beijing* 100049)
(E-mail: jkwu@gucas.ac.cn, ldong@i2r.a-star.edu.sg, baoxm@i2r.a-star.edu.sg)

Abstract Sensor networks provide means to link people with real world by processing data in real time collected from real-world and routing the query results to the right people. Application examples include continuous monitoring of environment, building infrastructures and human health. Many researchers view the sensor networks as databases, and the monitoring tasks are performed as subscriptions, queries, and alert. However, this point is not precise. First, databases can only deal with well-formed data types, with well-defined schema for their interpretation, while the raw data collected by the sensor networks, in most cases, do not fit to this requirement. Second, sensor networks have to deal with very dynamic targets, environment and resources, while databases are more static. In order to fill this gap between sensor networks and databases, we propose a novel approach, referred to as "spatiotemporal data stream segmentation", or "stream segmentation" for short, to address the dynamic nature and deal with "raw" data of sensor networks. Stream segmentation is defined using Bayesian Networks in the context of sensor networks, and two application examples are given to demonstrate the usefulness of the approach.

Key words Sensor networks, spatiotemporal data processing, databases, Bayesian networks

1 Introduction

Information fusion has been studied for decades, mainly by research forces for military applications, robotics and remote sensing[1,2]. Information fusion deals with issues such as gathering data from various sources, and developing algorithms and methods to integrate data items at various levels in order to reach right decision. There are several data fusion frameworks. The most popular one is JDL (Joint Directors of Laboratories)[3]. In JDL, the whole information fusion processing consists of 5 levels.

Level 0 — Sub-object data association and estimation: signal level data association and characterization. We call it data level fusion. For example, in remote sensing application[2], two satellites take images of the same city at different time and from different attitude. In order to fuse the two images, image registration operation is to be performed to transform these two images into the same coordinates so that the data describing the same objects from these two images are matched with one another.

Level 1 — Object refinement. We call it object level fusion. At this level, detection and recognition of the object is achieved, the states of the detected objects are estimated. Take remote sensing as an example, we can classify the land type being either residential area, forest, or crop and further estimate the stage of the crop growth in case land type being crop.

Level 2 — Situation refinement, in which object clustering and relation analysis are carried out by incorporating force structure and cross-force relations, communications, physical context, *etc.* We call it event level fusion. At this level, object states are analyzed to form behavior, and relationship between objects are studied to infer what is happening in a broader context.

Level 3 — Significance estimation, which includes risk estimation, event prediction, susceptibility and vulnerability assessment. We call it impact level fusion. At this level, the impact of the event is assessed and the risks of its consequences are evaluated.

Level 4 — Process refinement. We call it adaptation level. At this level, the system adapts and responds to the situation to fine-tune its functionalities. Such adaptation is especially important for sensor networks. For example, when a vehicle is detected to stop near a building, the monitoring system may perform its routine inspection such as instructing the video camera to have a close look of the vehicle. In case that the vehicle is suspicious, precautionary security check is activated by the system such as sending a robot to inspect if there are explosives. Such precautionary measure is a kind of process refinement to be adapted to specific situation.

Sensor networks are meant for distributed and collaborative monitoring, command and control. These pose a lot of challenges which have never been addressed. In the JDL model[3], Level 2 and Level

Received February 1, 2005; in revised form April 3, 2006

3 fusion are very application oriented, and Level 4 fusion is more on sensor network system adaptation. These topics are beyond the scope covered here. In this article, we will focus on Level 0 and Level 1 data fusion issues in the sensor network context.

There is little work done on data fusion in sensor networks. However, similar issues have already been addressed by several pioneers in the sensor network field from the database point of view. University of California at Berkeley developed TinyDB, a query processing system for extracting information from a network of TinyOS sensors[3]. Cornell University database group′s project "Courgar" is to build a new distributed data management layer that scales with the growth of sensor interconnectivity and computational power on the sensors[4]. These two works have a vision, "sensor network is a database". They either explicitly or implicitly assume that the sensor networks consist of a massively distributed sensor nodes like UCB motes, and each sensor is viewed as a tuple in the database table. Because the collected data are of simple type or self-explainable (such as temperature and noise level), simple data aggregation operation would be adequate and there is no need for complicated data fusion.

It is common understanding that massively distributed sensors keep collecting data for timely monitoring tasks. The data collection in sensor networks forms a spatial and temporal data stream. In most cases, the collected raw data do not have explicit meaning, and need to be interpreted before going to the database for high level decision making. As such, Stanford is working on data stream server[5,6] for centralized stream processing and management. However, this does not seem to match the distributed and collaborative nature of the sensor networks.

Today, networked databases and web link all computing machines in the world together, provide excellent platform for people to share information, and speed up their business. Sensor networks provide a powerful extension of this platform to link the real world (people, animals, infrastructure, environment and even space) together. Databases and web deal with well-formatted data, which have inter-operable data models, and can be shared worldwide for various applications. It is true that database concepts and tools provide very useful means for application development on top of sensor networks. On other hand, because data streams are not "well-formatted" in most cases, they cannot be directly incorporated into the database, and data aggregation functions provided by the database are incompetent to perform rather complicated data fusion tasks. As such, a bridge must be built between the sensor networks and databases before we can freely use existing database concepts and tools to monitor situations and track the changes of the real world. This bridge is mainly Level 0 and Level 1 data fusion, following that, Level 2 and 3 fusion can then be done on top of databases.

In this article, a new concept, "Stream segmentation", is described in more details. This idea is developed by modeling the sensor networks using random field, defining dynamic active sensor regions for collaborative processing, employing dynamic Bayesian network models to perform temporal segmentation, and finally, converting the data stream into well-formed database records. The rest of the paper is organized as follows. Fundamental discussions on stream segmentation are presented in Section 2. This is followed by application examples that are given in Section 3 and 4, respectively. Concluding remarks and future work are presented in Section 5.

2 Stream segmentation

2.1 Modeling sensor networks

Networked sensors are embedded into the real world to collect data which reflect the nature of the world. Let us consider the characteristics of the sensor networks and nature of the collected data.

1) In sensor networks, data are collected from the real world where the sensors are embedded. It reflects certain properties of the world, but may not be directly understandable by users. For example, acoustic and infrared sensor signals from moving vehicles are meaningless to database users. Instead, type, location and trajectories of those vehicles are important.

2) Data stream is continuous. Real world produces continuous time sequence of multi-modality spatiotemporal data stream. To realize our goal of monitoring the dynamics, sensor networks must sample the data stream without much loss of information.

3) Multi-modality data are collected using multiple types of sensors, which represent various aspects of the targets.

4) There may be uncertainty due to reliability of sensors and communication networks, and dynamic nature of the environment

Clearly, multi-modality spatiotemporal data stream collected by sensor networks are not well-formatted. It reflects dynamics of the targets but possesses uncertainties. We choose random field and

graph model to model sensor network data.

1) Sensor network can be best represented by a graph $G = (V, E)$, with V denoting vertex set and E the edge set.

2) Let v denote a sensor node at a spatial location, having its coordinates in the geographic space. We define $Z_v = \{z_v^1, \cdots, z_v^M\}$ the random value measurements indexed by vertex v, where z_v^j represents the measurement for the jth sensor mounted on the sensor node v at its location.

3) The edge between two vertices represents the relationships and dependences between the two data vectors.

2.2 Neighborhood system and collaboration region

All sensor networks are meant to accomplish certain tasks. A group of sensors collaboratively working together to fulfill a task is one of the major characteristics of sensor networks. A task can be location-invariant, such as monitoring a junction of a bridge, or dynamic, such as tracking moving target. Here, we define a neighborhood system as follows.

Vertices are related to each other *via* a neighborhood system. A neighborhood consists of one or a group of sensor nodes, performing a common task or forming a homogeneous region.

A neighborhood system is defined as

$$S = \{S_i | \forall i \in V\} \tag{1}$$

where S_i is the set of vertices neighboring vertex i, and vertex i is referred to as the site of a neighborhood S_i.

A vertex is not neighboring to itself, and neighboring relationship is mutual. Mathematically,

$$i \notin S_i, \text{ and } i \in S_{i'} \Leftrightarrow i' \in S_i \tag{2}$$

A neighborhood can be dynamic. The site of the neighborhood can change as the task location changes. In this case, vertices in the neighborhood change accordingly.

Fig. 1 shows a dynamic neighborhood system for tracking moving targets. At time k, there are two active neighborhoods sensing and tracking vehicle 1 and 2, respectively. As vehicle moves, the active neighborhood for tracking vehicle 1 moves forward as depicted in Fig. 1. Fig. 2 is a "Sensor network activity monitoring system, (SAMS)". There are three accelerometer sensors on waist and two legs forming neighborhood 1 for the task of activity status monitoring, while environment sensors and sensors in neighborhood 1 form neighborhood 2. Its task is to monitor the relationship between activity and environment conditions.

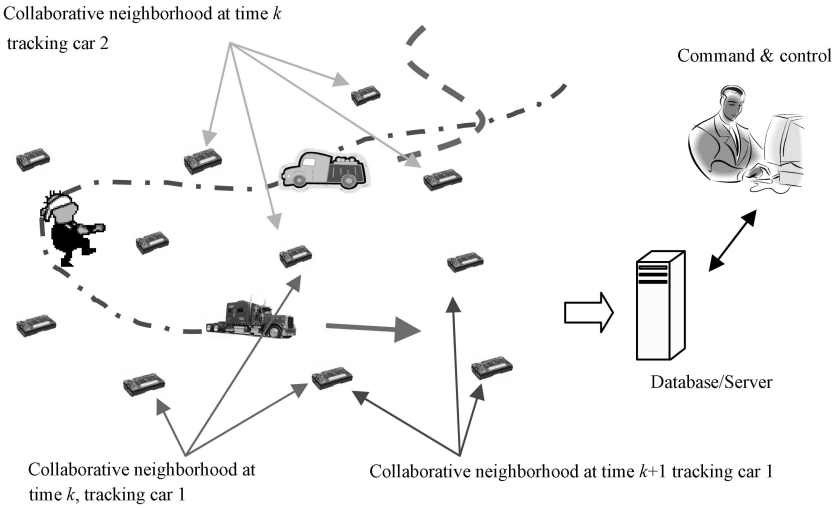

Fig. 1 Sensor network system for tracking moving vehicle and people

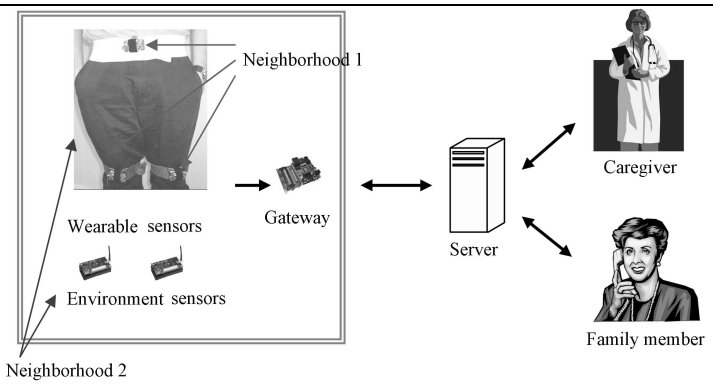

Fig. 2 Block diagram of SAMS and its neighborhood system

2.3 Detection and classification

Sensors are scheduled to detect any objects or events, which may occur anytime and anywhere. As soon as an object or event is detected, an active region is formed. Sensors inside the active region will work together to classify the object. Technically speaking, this classification is done by using labels and models. Definitions of labeling and labels are given below.

1) A labeling is an event that may happen to a site.

2) Labeling is to assign a label from the label set L to a set of objects at sites in S.

Model is defined as follows.

3) A model λ_L is constructed for each L. For example, a surveillance system may have a label set $L = \{car, lorry, people, dog\}$. A model refers to a classifier which assigns labels to the detected object with certain probabilities using the signals collected by all the active sensors.

After the type of the objects is determined, it is possible to track and monitor the status of the objects and further analyze their behavior.

2.4 Monitoring and tracking with dynamic system model

The real world that the sensor networks are monitoring is a spatiotemporal dynamic system. We have modeled the spatial properties of sensor networks, and now we would like to model the whole dynamic system by Bayesian network as depicted in Fig. 3. In the figure, Y_k^s and Z_k^s denote the state and measurements at time k of the neighborhood at site s, respectively. It should note that the measurement here is a composite one for the whole neighborhood, produced by all types of sensors mounted on all those nodes in the neighborhood. In other words, the measurement is generated by multiple sensors of multiple types located at different locations.

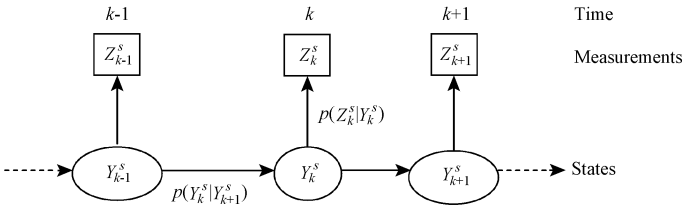

Fig. 3 Bayesian network representation of a spatiotemporal dynamic system

We assume that the state sequence is a first order Markov process, and we have the state transition equation:

$$Y_k^s = f_k(Y_{k-1}^s, u_{k-1}^s, ny_{k-1}^s) \tag{3}$$

We also assume that the measurement only depends on the state, then the measurement equation is

$$Z_k^s = h_k(Y_k^s, u_k^s, nz_k^s) \tag{4}$$

where u is the known input, ny is state noise (or system noise) and nz is measurement noise. In order to evaluate the state in the model to solve the dynamic system problem, we usually evaluate the maximum of the posterior distribution for a given sequence of measurement with the following two steps.

Prediction:

$$p(Y_k^s|Z_{1:k-1}^s) = \int p(Y_k^s|Y_{k-1}^s)p(Y_{k-1}^s|Z_{1:k-1}^s)dY_{k-1}^s \tag{5}$$

Update:

$$p(Y_k^s|Z_{1:k}^s) = \frac{p(Z_k^s|Y_k^s)p(Y_k^s|Z_{1:k-1}^s)}{p(Z_k^s|Z_{1:k-1}^s)} \tag{6}$$

In the prediction step, the state transition probability and the posterior at time $k-1$ are used to predict the new state at time k. In the update step, the likelihood factor is added to obtain the estimation of the posterior at time k. Principally, we can evaluate states at any time instance k for a given initial state and measurement sequences using the model mentioned above. In real applications, the challenges are: First, we do not have prior knowledge about the distribution, neither posterior nor likelihood. Second, the dynamics of the system are usually not linear. To address the first challenge, we may assume Gaussian or mixture Gaussian distribution, then use Kalman filter for tracking. One alternative method is using particle filter to approximate the real distribution. We will demonstrate that in the following sections.

3 Cross sensor and cross modality data fusion for tracking

We have developed a sensor network tracking system as shown in Fig. 1. The lead sensor in each active sensor region reports to the server the object type and its location. The report is saved in the database, forming the trajectory of an object. Further data fusion for the object is carried out at the server, with intervention by command and control. Here, we will focus on Level 0 and Level 1 data fusion, which correspond to cross sensor and cross modality fusion in this application.

Each sensor node is mounted with acoustic and seismic sensors. Moving vehicles are considered as acoustic and seismic sources. The sensors placed at various locations on the way of vehicles will detect acoustic and seismic signals emitted by vehicles. Vehicles are classified by their acoustic and seismic signal signatures. Their locations are determined by the signal strengths received by different sensors at different locations.

3.1 Cross-sensor and cross-modality data fusion

As we have defined, the measurement at time k, Z_k^S, in above equations is a composite measurement for the whole neighborhood. There will be cross-sensor and cross-modality data fusion to derive the composite measurement and its corresponding likelihood function.

Assume that there are N_S sensor nodes in the neighborhood S, and each sensor node is mounted with M types of sensors. As shown in Fig. 4, the signals collected by all sensors in the neighborhood using the same type of sensor, m, denoted as $z^m = (z_1^m, z_2^m, \cdots, z_{Ns}^m)$, are of the same modality. Because data elements in z^m are collected by different sensors at different locations, they will contribute collectively in the determination of source location. In that sense, they are "cooperative" in source localization. The way of "cooperation" among sensors is usually application dependent. As such, cross-sensor data fusion is usually embedded in the measurement model, and it is incorporated into the derivation of likelihood function.

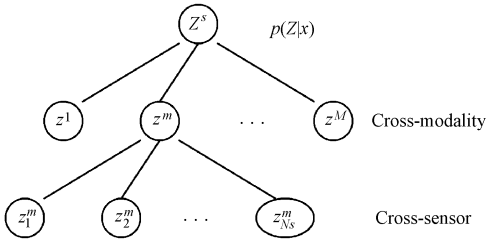

Fig. 4 Multiple modalities and multiple sensors in sensor networks

The multi-modality data, $Z = (z^1, z^2, \cdots, z^M)$, are complementary. It means that they measure different aspects of the targets, and complementarily contribute to the estimation. Formally, we refer this type of data fusion as cross-modality data fusion. In many cases, people assume that different types of sensors work independently, and the data collected by different types of sensors (referred to as data of different modalities) are independent as well. In those cases, the data fusion is very simple,

which can be formulated as multiplication of their likelihood function:

$$p(Z|x) = p(z^1|x)p(z^2|x)\cdots p(z^M|x) \tag{7}$$

3.2 Approximate posterior distribution using particle filter

Because posterior distribution is not always Gaussian, we use Monte Carlo importance sampling (or particle filter) to implement target tracking in sensor networks.

The idea of Monte Carlo simulation is to draw an identical independently distributed set of samples from a target PDF. These L samples can be used to approximate the target density with the empirical point-mass function.

The simulation starts with a weighted set of samples, $\{x^i_{k-1}, w^i_{k-1}\}^L_{i=1}$, which are approximately distributed according to $p(x_{k-1}|z_{k-1})$. New samples are then generated according to a suitably chosen proposal distribution, which may depend on the old state and the new measurements

$$x^i_k \to q(x_k|x^i_{k-1}, z_k), \quad i = 1, \cdots, L \tag{8}$$

The expectation of the state can then be approximated by those samples,

$$\overline{E(x_{0:k})} = \frac{1}{L}\sum_{i=1}^{L} w^i_{0:k} x^i_{0:k} \tag{9}$$

The new importance weights are updated in each time slot with likelihood function, so that the posterior distribution can be represented using a set of samples or particles.

$$w^i_k = w^i_{k-1}\frac{p(z_k|x^i_k)p(x^i_k|x^i_{k-1})}{q(x^i_k|x^i_{k-1}, z_k)}, \quad \sum_i w^i_k = 1 \tag{10}$$

In most cases, re-sampling is necessary to prevent de-generation.

3.3 Dynamic model

In the target tracking problem, for simplicity, the location of a target is treated as the state. Therefore, the state transition model can be simplified as follows.

$$x_k = x_{k-1} + disp_{k-1} + nx \tag{11}$$

where $disp_{k-1}$ is the displacement of the target at time $k-1$. $disp_{k-1}$ will be approximated by the actual displacement at last time instance. The noise term, nx, is assumed to be constant for all time instances, which is simulated by Gaussian noise.

Particle filters have evolved into many different varieties over the past few years[7]. The key issue of particle filter is the choice of proposal distribution, which can best approximate the target posterior distribution. Here we choose sampling importance resampling (SIR) approach for its simplicity and effectiveness. With SIR, the proposal distribution and the update formula are reduced to the following forms.

$$q(x_k|x^i_{k-1}, z_k) \to p(x_k|x^i_{k-1}) \tag{12}$$

$$w^i_k = p(z_k|x^i_k), \quad \sum_i w^i_k = 1 \tag{13}$$

With this proposal distribution, the samples are drawn around the predicted location.

3.4 Measurement model and likelihood

Let us look into the acoustic sensors. When the sound propagates in the free and homogenous space, the acoustic energy decay function can be modeled by the following equation.

$$z^{aco}_n = g_n\frac{S}{\|y - r_n\|^2} + nz_n = g_n d^{aco}_n S + nz_n \tag{14}$$

where z^{aco}_n is the acoustic signal received by the nth sensor, nz_n is the noise term that summarizes the net effects of background additive noise and the parameter modeling error. g_n and r_n are the gain factor and location of the nth sensor, respectively. S and y are the respective energy emitted by the source and its location. Here we only consider one target, and we assume that the distribution of sensor nodes is

dense enough so that there is only one target enters into the neighborhood. d_n^{aco} represent the distance factor for acoustic signal. In this case, it is $1/d^2$. The acoustic measurement in the neighborhood can be represented by a vector as follows.

$$z^{aco} = (z_1^{aco}, z_2^{aco}, \cdots, z_N^{aco})^t = GDS + nz \tag{15}$$

where we omit superscript "aco" with the understanding that this measurement equation is for the neighborhood of Ns number of acoustic sensors and the acoustic source at location y.

Assume that the noise term in the measurement function is Gaussian white noise with standard deviation σ, the likelihood function of having measurement z at state x^i is

$$p(z^{aco}|x^i) = \frac{1}{(2\pi\sigma)^{1/2}} \exp(-\frac{(GDS - z^{aco})^{\mathrm{T}}(GDS - z^{aco})}{2\sigma^2}) \tag{16}$$

In the case of seismic, the attenuation model is $z^{sei} = S^{sei}e^{-\kappa d}$, where d is the distance, and κ is attenuation factor, being $40.0013 - 0.003$ in [6]. The likelihood of seismic model can be computed in the similar way as in (16). Let $e = (GDS - z)^{\mathrm{T}}(GDS - z)$, the weight update function for fusion of multiple modalities of measurements can be written as

$$w_k^i = p(z^{aco}|x_k^i)p(z^{sei}|x_k^i) = \exp(-\frac{c^{aco}e^{aco} + c^{sei}e^{sei}}{2\sigma^2}) \tag{17}$$

where we have omitted the constant in front of exponential function because there will be a normalization process for weights. The coefficients c^{aco} and c^{sei} reflect the quality of measurements and the contribution of the measurements to the overall likelihood.

3.5 Particle filter algorithm

We summarize the particle filter implementation of Bayesian filter for target tracking in sensor networks using acoustic and seismic sensors, for cross-sensor and cross-modality (CSCM) data fusion as follows.

CSCM Algorithm

1) Initialize $x^i, i = 1, 2, \cdots, L$ (number of samples)

2) Prediction:

For each particle $i = 1 : L$,

Draw $x_k^i = x_{k-1}^i + disp_{k-1}$

3) Determine active neighborhood by threshold of received sensor signals to generate vector z^{aco} and z^{sei}.

4) Weight update:

$$w_k^i = \exp(-\frac{c^{aco}e^{aco} + c^{sei}e^{sei}}{2\sigma^2})$$

Normalization is then carried out for the weights.

5) Compute expectation of the target state

$$\overline{E(x_k)} = \frac{1}{L}\sum_{i=1}^{L} w_k^i x_k^i$$

6) Resample - drop those samples with too small weights, and split those samples with largest weights so that the number of samples remain to be L.

3.6 Experimental results

We have applied our algorithm to the SensIt dataset[8]. Among all the data, we selected 6 collections so that there are enough number of sensor nodes, and the quality of the data are reasonable. For all 6 data collections, three runs of experiments were conducted. Fig. 5 shows the experimental results for data collection of "aav6". In the figure, "*" denotes sensor nodes. The true track of vehicle is plotted using dark dots, while the lighter dots indicate the estimated track. The lightest symbols are samples used at the last step in the tracking.

The acoustic data collected by sensors indicate that there are large differences of gain of sensors: sensors located farther may output larger signal than those located nearer. In the experiments, we did not spend much effort on estimating the sensor gain. As such, we can observe from the Fig. 5(a) that the tracking behaves well for those locations where there are sensors on both sides of the track. In

contrary, there are obvious tracking errors for those locations where there are sensors on only one side of the track.

The seismic signals from all sensors are comparatively uniform. The tracking results are quite satisfactory as shown in Fig. 5(b) with mean square error of 2.44. By fusion of seismic and acoustic signal, we are able to obtain much better tracking results as shown in Fig. 5(c). The mean square error is 1.397, which is much smaller than tracking using seismic only. Here in this experiments, we set $c^{aco} = 0.1$, and $c^{sei} = 1$, because of the poor quality of the acoustic data.

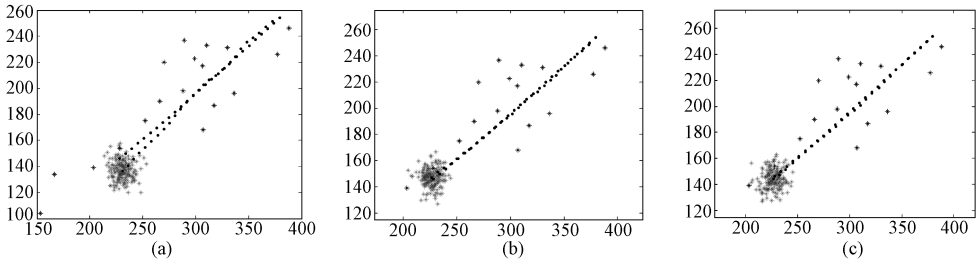

Fig. 5 (a) Tracking using acoustic sensor only (the mean square tracking error is 4.7). (b) Tracking using seismic sensor only (the mean square tracking error is 2.44). (c) Tracking by fusion of seismic and acoustic signal (the mean square tracking error is 1.397)

4 Activity detection, classification, and tracking

4.1 Extraction of body features for activity classification

We have developed a sensor network activity monitoring systems (SAMS) as depicted in Fig. 2. Here, the spatiotemporal data stream collected from 3 accelerometers is (here we use t rather than k to indicate time index)

$$\{(a_{x,t}^{waist}, a_{y,t}^{waist}), (a_{x,t}^{left}, a_{y,t}^{left}), (a_{x,t}^{right}, a_{y,t}^{right})\} \tag{18}$$

In order to detect and classify activities, we extracted following feature measures from the data streams at each time interval.

$$\begin{aligned}
&z_{1,t}^s \to Energy \quad en_t^s = \|a_t - a_{t-1}\| \\
&z_{2,3,4,t}^s \to Vertical_Accelaration \quad va_t^v \\
&z_{5,t}^s \to Dominance_Frequncy \quad df_t^s \\
&z_{6,t}^s \to HarmonicRatio \quad hr_t^s = \sum even_harmonic / \sum odd_harmonic \\
&z_{7,t}^s \to Stance/swing_ratio \quad sr_t^s = time_{stance}/time_{swing} \\
&z_{8,t}^s \to BipedSymmetry \quad bs_t^s = ac^{left} - ac^{right}
\end{aligned} \tag{19}$$

Using those feature measures, based on the learned activity model $\lambda_{activity}$, we have designed a tree classifier to detect and classify the activities, and then convert the data stream into database entries as follows.

$$\begin{aligned}
&\{WearerID, (a_{1,t}^{waist}, a_{2,t}^{waist}), (a_{1,t}^{left}, a_{2,t}^{left}), (a_{1,t}^{right}, a_{2,t}^{right})\}_{\lambda_{activity}} \Rightarrow \\
&\{wearerID, (l^1, t_{start}^1, duration^1, a_{sample}^1), \cdots, (l^n, t_{start}^n, duration^n, a_{sample}^n)\}
\end{aligned} \tag{20}$$

Here in the SAMS system, $L = \{standing, lying, sitting, normalwalking, abnormalwalking, falling\}$. After that, we have used database functions to implement activity status query, warning of abnormal walking and falling, and providing statistics of activities of one person or a group of people.

4.2 Body movement tracking using extended Kalman filter

Statistics of human daily activities indicate that most (over 90%) body postures are in proximities of three typical body postures, sitting, lying and standing. In another word, it is reasonable to assume that a body posture is likely to be a deviation from one of these three body postures, and these deviations (such as detailed walking patterns and sleeping patterns) may provide very important health related information.

Equations (18)∼(20) have laid theoretical foundation for classifying body postures. Here, we present a more practical solution of activity tracking by using extended Kalman filter(EKF)[9], which

assumes Gaussian posterior distribution function. This approach finds optimal solution by using classical mean square estimation method. Currently, we are in the process of studying the feasibility of monitoring elderly in hospital to prevent possible falls by detecting the patterns from lying to sitting and getting out of bed, and at the meanwhile, presenting nurses with animated cartoons for their attention.

For the task of activity tracking, wearable sensors are attached at the waist and both legs as depicted in Fig. 6. The sensors are "vertically" posed because the x-o-y plane of the sensor is vertical to the ground. The proposed EKF takes the measurements of the vertically posed sensor as its inputs. From Fig. 6, it is seen that b and d are y-axis and x-axis acceleration readings of the sensor while a and c are actual vertical acceleration and horizontal acceleration, respectively. θ is the angle between the y-axis of the sensor and the gravity. The state transition equation here becomes

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + \boldsymbol{W} \tag{21}$$

where $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\boldsymbol{x}(k) = \begin{pmatrix} \theta_k \\ a_k \\ c_k \end{pmatrix}$ and $\boldsymbol{W} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$ are state transition matrix, state variables, and system noise, respectively. Here, system noise is assumed independent and zero mean Gaussian, *i.e.* $w_1 = G(0, q_1)$, $w_2 = G(0, q_2)$, $w_3 = G(0, q_3)$ where q_1, q_2 and q_3 are variances of the distributions.

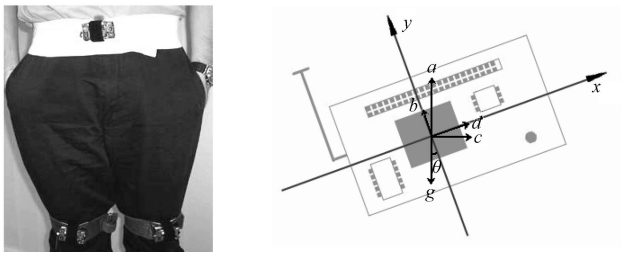

Fig. 6 Placement of the wearable sensors and the measurements of the sensors

Measurement equation is written as follows in the context of this sensor setting.

$$z(k) = F[x(k)] + V \tag{22}$$

where $z(k) = \begin{pmatrix} b_k \\ d_k \end{pmatrix}$, $V = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ is measurement noise and F is the transition function between the state variables $x(k)$ and measurements $z(k)$. Measurement equation can be re-written in a simpler form as follows.

$$\begin{cases} b_k = (a_k - g)\cos\theta_k + v_1 \\ d_k = c_k\cos\theta_k - g\sin\theta_k + v_2 \end{cases} \tag{23}$$

After implementing local linearization, we have

$$z(k) = \tilde{z}(k) + F'[x(k) - \tilde{x}(k)] + V \tag{24}$$

where F' is the Jacobian matrix of partial derivatives of F with respect to x, $\tilde{x}(k)$ and $\tilde{z}(k)$ are the approximate state and measurement vectors of $x(k)$ and $z(k)$, respectively. After some manipulation, we have

$$F' = \begin{bmatrix} -(a_k - g)\sin\theta_k & \cos\theta_k & 0 \\ -c_k\sin\theta_k - g\cos\theta_k & 0 & \cos\theta_k \end{bmatrix} \tag{25}$$

For the EKF, the initial values of $x(k)$ can be set as constants, *e.g.* $x(0) = \begin{pmatrix} \theta_0 \\ a_0 \\ c_0 \end{pmatrix}$, such that the EKF is motivated from a fixed starting point while treating with different body postures. However, it is better to set $x(0)$ based on the values of the readings at time $t = 0$. For example, let $a_0 = 0$, *i.e.* no vertical acceleration at $t = 0$ in addition to gravity, we have $\theta_0 = -\arccos(\frac{b_0}{g})$ and $c_0 = \frac{d_0 + g\sin\theta_0}{\cos\theta_0}$, where b_0 and d_0 are sensor readings at $t = 0$.

Another initial factor for the EKF is the a posteriori estimate covariance matrix.

$$P_k = E([\boldsymbol{x}(k) - \bar{\boldsymbol{x}}(k)][\boldsymbol{x}(k) - \bar{\boldsymbol{x}}(k)]^{\mathrm{T}}) \tag{26}$$

where $\bar{\boldsymbol{x}}(k)$ is the a posteriori estimate of $\boldsymbol{x}(k)$. Because the system noise is assumed independent, matrix P_k is diagonal. The state variables in $\boldsymbol{x}(k)$ are hidden parameters of the system and their estimation error is initialized empirically. In this experiment, the initial values of P_k are set as

$$P_0 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{27}$$

With the state transition equation, measurement equation and initial settings mentioned above, a posteriori estimates of the state parameters are obtained for each time slot by using the standard Kalman recursion[10].

4.3 Experimental results

The proposed EKF is applied to estimate inclination angle of the trunk and flexion angles of both legs. Based on these values, physical movement of the wearer is reconstructed and is animated with a small cartoon man as depicted in Fig. 7. It is seen that the details of the body status can be accurately estimated no matter the person is standing, sitting or lying down.

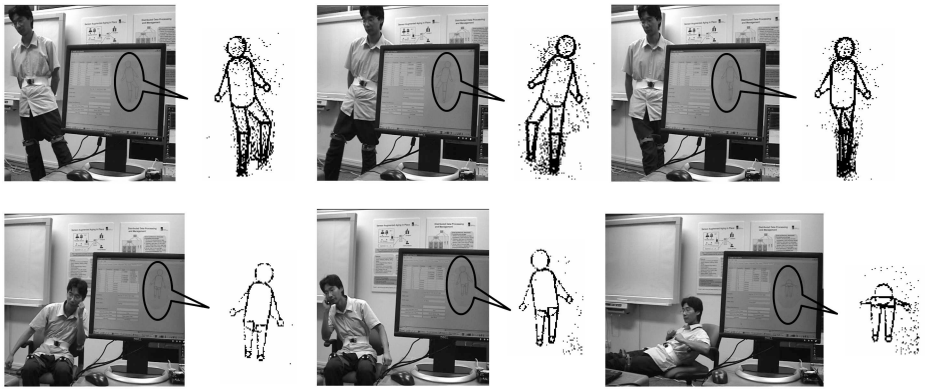

Fig. 7 Tracking results of body movement using extended Kalman filter

5 Conclusion and future work

We have proposed a data fusion approach in sensor networks. We questioned about the statement of "sensor networks is a database", and argued that in most applications, the data stream is much more complicated than database entries, and therefore, Level 0 and Level 1 data fusion must be performed in order to segment the data stream and convert the data stream into well-formatted data before entering into the database. We further described the idea of "stream segmentation" by analyzing sensor network models, sensor data, neighborhood and dynamic systems. Two application examples are given to illustrate the theoretical foundation and algorithms about stream segmentation.

Our future work is to further extend the work to cover nonlinear dynamics, and real-time in-network implementation in an energy constraint sensor network.

References

1 Proceedings of the 7th International Conference on Information Fusion. Stockholm, Sweden: International Society for Information Fusion, 2004
2 Legg J A. Tracking and sensor fusion issues in the tactical land environment. Technical Report: TN.0605, DSTO Information Sciences Laboratory, Australia, 2005
3 TinyDB – a declarative database for sensor networks. [Online], available: http://telegraph.cs.berkeley.edu/tinydb/
4 Yao Y, Gehrke J E. Query processing in sensor networks. In: Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR 2003). Asilomar, CA, USA: 2003

5 Arasu A. *et al*. STREAM: The Stanford data stream management system, [Online], available: http://dbpubs.stanford.edu/pub/2004-20, 2004
6 Hammad M, Aref W, Franklin M, Mokbel M, Elmagarmid A K. Efficient execution of sliding window queries over data streams. Technical Report: CSD TR 03-035, Purdue University, 2003
7 Arulampalam M S, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 2002, **50**(2): 174∼188
8 Sheng X, Hu Y H. Energy based acoustic source localization. In: Proceedings of the International Conference on Information Processing in Sensor Networks. Heidelberg, Barlin: Spring-Verlag, 2003. 285∼300
9 Welch G, Bishop G. An introduction to the Kalman filter. Technical Report: TR95-041, University of North Carolina at Chapel Hill, Apr. 2004

WU Jian-Kang Has been the principal scientist of the Institute for Infocomm Research, the national research institute of Singapore. He is now professor in the Graduate School, Chinese Academy of Sciences, and the director of the Sensor Networks and Applications Research Center, a joint research organization between the Institute of Automation and the Graduate School, Chinese Academy of Sciences. His research interests include sensor networks, digital media processing, communication and right management, and imaging systems.

DONG Liang Received his Ph. D. degree from the National University of Singapore in 2005. He is now a research fellow at the Institute for Infocomm Research, Singapore. His research interests include speech recognition, signal processing, and sensor networks.

BAO Xiao-Ming Received his bachelor, master, and Ph. D. degrees in Department of Automatic Control of Shanghai Jiaotong University in 1984, 1987, and 1991, respectively. He is currently a research staff at Institute for Infocomm Research in Singapore. His research interest includes real-time embedded system design and implementation, wireless body sensor network, and multi-modal sensor data fusion for personalized healthcare applications.