

Activity-informed Dynamic Data Driven Simulation

Xiaolin Hu^{1,a}, Sanish Rai¹, and Xiaoming Wang²

¹Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

²Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

Abstract. In previous work we developed dynamic data driven simulation (DDDS) that assimilates real time sensor data using Sequential Monte Carlo (SMC) methods. This paper builds on previous work and presents a framework that adds a real time behavior pattern detection layer on top of data assimilation for dynamic data driven simulation. The real time behavior pattern detection layer uses Hidden Markov Model (HMM) to detect the behavior patterns of a system in real time and uses the detected behavior pattern to inform the simulation model for more accurate simulation. We apply the proposed framework to a smart environment application and discuss how to recognize behavior pattern from spatial-temporal sensor data using Coupled HMM (CHMM).

1 Introduction

Incorporating real time data into a running simulation model has the potential to greatly improve simulation results. Unfortunately, until recently this line of work did not receive significant research attention in the modeling and simulation community. While sophisticated simulation models have been developed, traditional simulations are largely decoupled from real systems by making little usage of real time data from the systems under study. With recent advances in sensor and network technologies, the availability and fidelity of such real time data have greatly increased. To utilize real time sensor data for improving simulation results, we carried out research on dynamic data driven simulation (DDDS), where a simulation system is continually influenced by the real time data streams for better analysis and prediction of a system under study [1, 2].

^aCorresponding author: xhu@gsu.edu

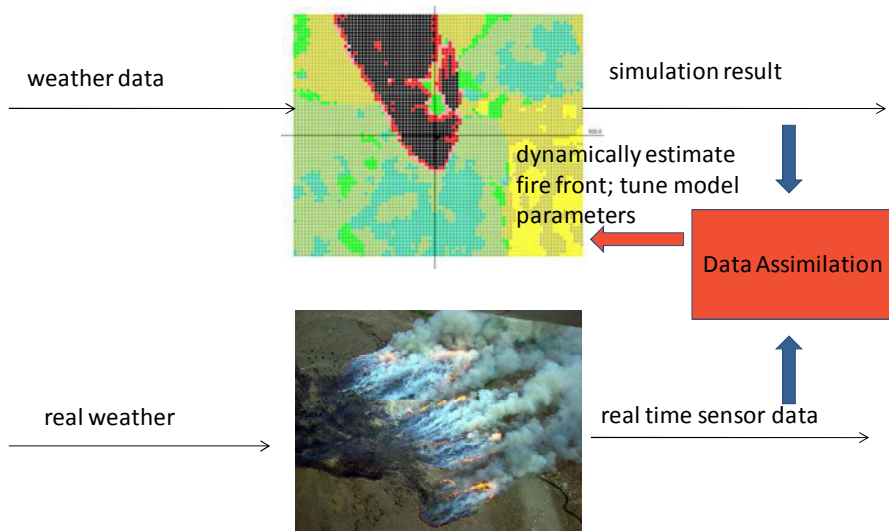


Figure 1. Dynamic Data Driven Simulation for Wildfire Spread Prediction.

Figure 1 illustrates the idea of dynamic data driven simulation based on the application of wildfire spread simulation. In the figure, the top part represents the wildfire simulation model; the bottom part represents the real wildfire. As the wildfire spreads, streams of real time data collected from fire sensors deployed in the fire area are transmitted to and assimilated by the data assimilation component. This allows the simulation system to continuously adjust itself in order to achieve more accurate predictions of wildfire spread. By coupling a simulation system with real time data, dynamic data driven simulation can greatly increase the power of simulation-based study. In the wildfire example, based on the real time sensor data of a wildfire, a wildfire simulation can better estimate the current fire front and fire intensity and thus providing more accurate fire spread predictions. The sensor data also carries “feedback” information for a simulation to calibrate its model parameters to reduce discrepancies between its simulation results and the observations. Dynamic data driven simulation finds applications in many different areas. Besides the wildfire application described above, other potential applications include traffic simulation where real time traffic sensor data are assimilated to improve traffic predictions and crowd behavior simulation where real time motion sensor data are used to infer the crowd density and crowd state for better simulation of crowd movement.

A major task of DDDS is data assimilation that assimilates sensor data collected from the real system into the simulation model. Typically, the real system’s states, which change over time, cannot be directly observed and is unknown to the simulation model. This makes the simulation start from a state different from the state of the real system, leading to inaccurate simulation results. Thus there is a need to dynamically estimate the “current” state of the real system and then feed the estimated states to the simulation model. This is achieved through data assimilation that assimilates real time sensor data for inference of the “current” system state. Our previous work on DDDS for wildfire spread simulation used Sequential Monte Carlo Methods [3], also called Particle Filters, for data assimilation and archived improved results. Data assimilation combines the observations (i.e., sensor data) of the current state of a system with the results from a prediction model (i.e., the simulation model) to produce an analysis. The results of data assimilation thus depend not only on the observation data, but also on the simulation model that “predicts” the evolution of system state. In DDDS where the simulation model runs in parallel with the real system, it is important for the simulation model to be engaged in the same behavior pattern as that of the real system so that it can simulate the evolution of system state in high accuracy.

It is observed that complex spatial-temporal systems often exhibit distinct behavior patterns in both space and time. Considering the traffic system as an example, “traffic congestion” refers to the

behavior pattern where vehicles move in extremely slow speed; while “no traffic congestion” refers to the behavior pattern where vehicles move freely. Similarly, in a smart office environment, “holding a meeting” refers to the behavior pattern where people stay in the conference room for the meeting; while “taking a break” refers to the behavior pattern where people moves out of the conference room and take a break. Note that each of these behavior patterns characterizes a certain way of how the system behaves and generates different “activity signatures” in the real time data collected from sensors. Recognizing the behaviour pattern of a system from sensor activities in real time can inform the simulation model for more accurate simulations of the system under study. Motivated by these observations, in this paper we focus on behavior pattern recognition from spatial-temporal sensor activities. We place this work within an activity-informed dynamic data driven simulation framework, and propose a method to utilize spatial-temporal sensor activities for informing dynamic data driven simulation.

2 A FRAMEWORK FOR ACTIVITY-INFORMED DDDS

The activity-informed DDDS framework adds a new layer of *behavior pattern recognition from sensor activities* on top of *data assimilation* and uses the recognized behavior pattern to inform the simulation model. Figure 2 shows this activity-informed DDDS framework. As can be seen, at the bottom layer is data assimilation into the simulation model. The data assimilation uses real time sensor data and the simulation model to infer the state of the system and/or to tune the model parameters in real time. We developed data assimilation based on the Sequential Monte Carlo Methods in previous work (see [1, 2] for more details).

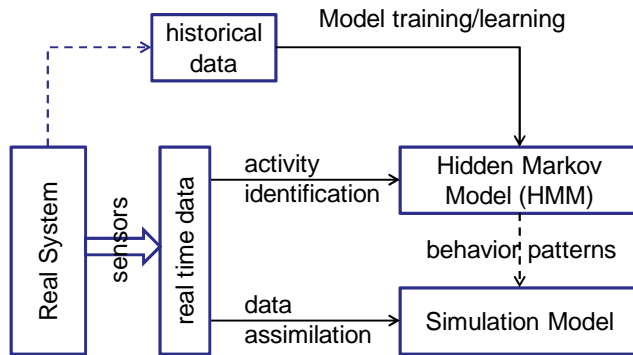


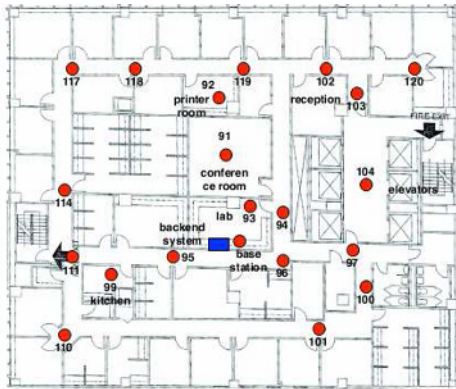
Figure 2. Activity-informed DDDS framework.

On top of data assimilation is the activity identification and behavior pattern recognition layer. In this paper, we propose to use Hidden Markov Model (HMM) [4] and extensions of HMM for the task of behavior pattern recognition from spatial-temporal sensor activities. HMM is defined as quintuple (S, E, P, A, B) , where $S = \{S_1 \dots S_N\}$ are the values for the hidden states, $E = \{O_1 \dots O_T\}$ are the values for the observations, P is probability distribution of the initial state, A is the transition probability matrix and B is the emission probability matrix. HMM is used to recognize events/patterns in many different applications (see, e.g., [5, 6, 7]). In our framework, HMM supports behavior pattern recognition from real time sensor data and outputs the recognized behavior patterns to the simulation model to inform the latter for better simulation. Figure 2 also shows that the HMM needs to be trained from historical data before it can be used in real time for behavior pattern recognition.

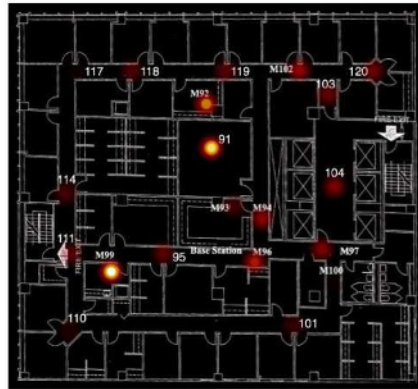
This framework separates the two layers that have different concerns: the simulation model captures the low-level dynamics of the system behavior; the HMM recognize the high level “behavior pattern” to inform the simulation model. In the reminder of this paper we focus on the behavior pattern recognition layer and use a smart office environment as the application context to show how to recognize behavior patterns form spatial-temporal sensor data using HMM.

3 A SMART OFFICE ENVIRONMENT

We consider a smart office environment and use DDDS to estimate the system state (e.g., density and locations of people in the environment) from sensor data to support emergency response planning for scenarios such as fire alarm and evacuation. The Department of Computer Science at Georgia State University has set up a smart office environment on the 14th floor of the building that consists of 20 TelosW static wireless sensor nodes (Figure 3(a)) deployed throughout the 30 meter \times 30 meter floor workplace environment [8]. The sensors are fixed on the ceiling and are triggered when a person walks under it, but it does not detect a person sitting motionless under the sensor. The sensor nodes are deployed mainly in the hallways, key positions like entry, exit, positions with high motion activities in workday like conference room, printer room, kitchen, busy lab. The nodes are equipped with Panasonic AMN-31111 PIR (passive infrared) motion sensor. The detected motion data are sampled at 10 Hz when an event triggered by motion. The data are collected by the base station through multi-hop communication (formed a 5-hop network) and stored in a back-end database.



(a) Environment Map and sensors



(b) Heat map based on sensor count in a week

Figure 3. A Smart Office Environment Example.

The data can be visualized in the form of a two dimensional matrix, where the columns are sensors and the rows are the time intervals. The time interval of row is 100ms and duration of 2 seconds is typically observed for detecting a person crossing under a sensor. Figure 3(b) uses a heat map showing the sensor counts counting the activities on each sensor, based on data collected for a consecutive week from morning 8 am to evening 20 pm. The data for each node is counted which provides the activity occurring at that node. Brighter colors indicate higher counts. From the figure, clearly node M99 (Kitchen) and 91(Conference) have the highest count followed by node M92 (Printer room).

By looking at the data from the sensor network, we can identify regions with various densities of activities. For example a region where a meeting is going to take place will record a high number of activities for that duration. The hallways leading to that region of high activity will also record a high density corresponding for the entering and exit of the people. When the meeting is over, the people will leave the meeting room and may go to a cafe or their respective rooms. This behavior pattern will lead to reduced activity in the conference room and increased activity in the cafe and other office rooms. So we will have activities depending on time and space. Understanding this kind of spatial and temporal relationships we can develop a model to represent human behaviors in space and time, and then recognize the “current” behavior patterns from real time sensor data.

4 BEHAVIOR PATTERN RECOGNITION USING COUPLED HMM

Within the application context of smart office environment, this section uses a simple example to show how to recognize behavior pattern from spatial-temporal sensor data using Coupled HMM

(CHMM) [9]. We consider a simplified smart office environment that has a conference room and a cafeteria room connected by a hallway, as shown in Figure 4. People attend the conference in the conference room and go to cafeteria during breaks. Suppose the three places have binary sensors that capture the motions of the people. We can say that during the breaks, there will be more counts in cafeteria and increased counts across the hallway during start and end of the break. At other times more counts will be in the conference room. Also at each location, the sensor activities will depend on various events like start of the conference, short breaks, change of speakers, end of conference and others.

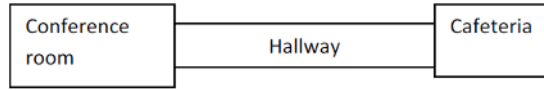


Figure 4. A simple example of a smart office environment consisting of a conference room and cafeteria connected by a hallway.

Since the activity consists of people moving across the rooms in various time intervals, we can assume that each room exhibits the Markov property in the temporal domain. We can create a single HMM for each room based on the observation of the sensor data. Observing each spatial location separately, a localized model can be achieved by creating a single HMM for each room separately. Figure 5 illustrates a HMM for the conference room.

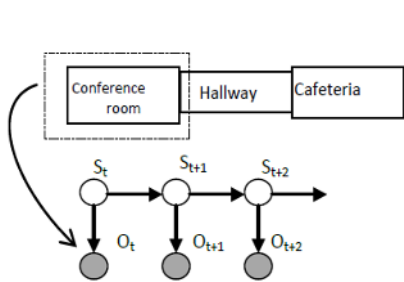


Figure 5. HMM for each location.

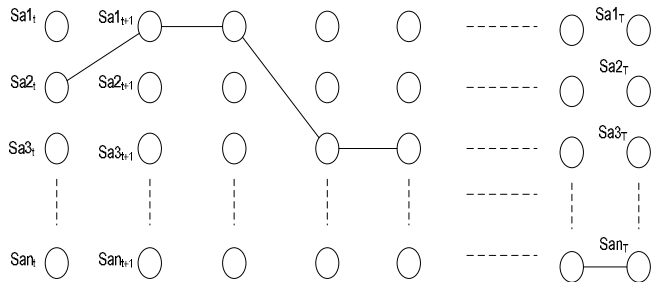


Figure 6. A single path through a n -state HMM.

In Figure 5, the circles represent the states. The shaded circles are the observed states and others are hidden one. The hidden states correspond to the activity of each location. Figure 6 shows a detail representation of a conventional HMM for a location with n states [9]. The straight line shows the transitional probability and represents the hidden states selected for each time step. For example the states of a conference room may be *begin_conference*, *having_conference*, *short_break*, *end_conference*, etc. Similarly the cafeteria may have states like *empty*, *break_time*, etc. These states will differ according to the time period and also will have relation with the neighboring locations. Using the HMM for each spatial location we can capture the temporal behavior pattern relationships within each locations. Incorporating the temporal information will enable the detection of behavior pattern with varying temporal transitions.

As discussed before, there is also dependency between different places. When a person moves from one place to another, the neighboring places of the source exhibit certain activity pattern. For example, in Figure 4, during break when people go to cafeteria from conference room, they go through the hallway. Thus the flow of motion will show a certain pattern from the conference room to the hallway and then to the cafeteria. This means the sensor activity will have a strong correlation within the spatially neighboring regions. The sensor activity in a place will have relationships with the activities in close proximity regions. We can model this strong relationship between spatially neighboring regions to account for the activity relationship over space. In this paper, we choose to use coupled Hidden Markov Model [9] to capture this correlation behavior of the neighboring regions.

Figure 7 shows a coupled Hidden Markov Model (CHMM) that captures the relationship between the activity occurring in the conference room, hallway and cafeteria. In the figure, S_a , S_b and S_c represents the states of conference room, hallway and cafeteria respectively. The solid straight lines represent the transition probabilities for states corresponding to one location and the dashed lines represent the coupling probabilities affecting the states between different locations. Figure 8 shows the detailed model and displays a single path for two locations where location a has 3 states and location b has 4 states. The coupled HMM captures the spatial-temporal correlation between the neighboring locations. It can be trained to help to recognize the behavior patterns from sensor data.

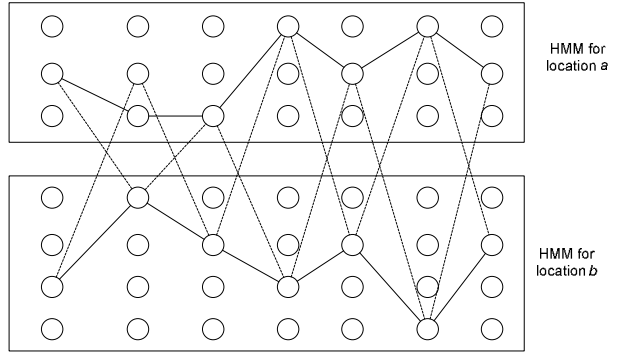
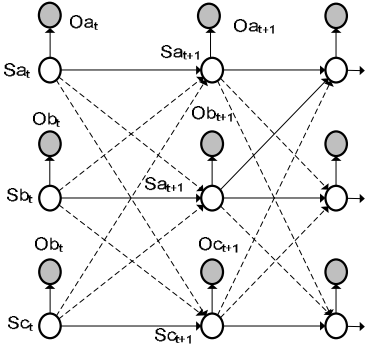


Figure 7. CHMM for 3 locations.

Figure 8. Detail CHMM for location a having 3 states and location b having 4 states

The posterior of a state sequence through the coupled HMM of two locations a and b can be obtained by using equation by Brand [9] as follows :

$$P(S|O) = \frac{\pi_{a_1} P(O_1^a | a_1) \pi_{b_1} P(O_1^b | b_1)}{P(O)} \prod_{t=2}^T P_{a_t|a_{t-1}} P_{b_t|b_{t-1}} P_{a_t|b_{t-1}} P_{b_t|a_{t-1}} P(O_t^a | a_t) P(O_t^b | b_t) \quad (1)$$

where π_{a_1} and π_{b_1} are the initial probabilities of states, $P_{a_t|a_{t-1}}$ and $P_{b_t|b_{t-1}}$ are the inner-state transition probabilities, $P_{a_t|b_{t-1}}$ and $P_{b_t|a_{t-1}}$ are the coupling probabilities modeling the interactions between two HMMs, $P(O_t^a | a_t)P(O_t^b | b_t)$ are the output probabilities of the states. From the observed data, we need to find a state sequence S which maximizes $P(S|O)$. For each state we need to compute both the inner-chain transition and coupling probabilities and using in equation (1) outputs the best state sequence S which involves recognized activity state sequences S_a for location a and S_b for location b . We can expand equation (1) for more than two locations thus modeling the spatial-temporal relationship for activities in the environment.

Specifically, we follow the CHMM model used by [10] to carry out behavior pattern recognition. We develop a HMM for each of conference room, hallway and cafeteria, and establish couplings between the three HMMs. We consider our model as a special case of dynamic Bayesian networks and consider each stream of a HMM as a continuous mixture. For the learning process of CHMM, the parameters can be defined as follows :

$$\pi_o^c(i) = P(q_t^c = i) \quad (2)$$

$$b_i^c(i) = P(O_t^c | q_t^c = i) \quad (3)$$

$$a_{i|j,k,l}^c(i) = P(q_t^c | q_{t-1}^c = j / k / l) \quad (4)$$

where q_t^c denotes the state of the coupled node for the location c at time t , $\pi_0^c(i)$ is the initial state probability, $b_t^c(i)$ is the observation probability and $a_{i|j,k,l}^c(i)$ is the transition probability for c^{th} location at time t for transition from state j , k and l state of other locations to state i .

The observation probability for Gaussian mixture components [10] is given by

$$b_t^c(i) = \sum_{m=1}^{M_i^c} (w_{i,m}^c N(O_t^c, \mu_{i,m}^c, U_{i,m}^c)) \quad (5)$$

where $\mu_{i,m}^c$ and $U_{i,m}^c$ represents the mean and covariance matrix of the i^{th} state of the coupled node with m^{th} component of the associated mixture node in the c^{th} channel, M_i^c is the number of mixtures and $w_{i,m}^c$ represents the conditional probability $P(s_i^c = m | q_i^c = i)$ when s_i^c is the component of the mixture node in the c^{th} stream at time t . In the training phase, for each training observation sequence r , the data in the stream is managed according to the number of states of coupled node. A K-means algorithm [11] with M_i^c clusters can be used to determine a sequence of mixture components for each stream.

The new parameters of the model can be estimated as:

$$\mu_{i,m}^c = \frac{\sum_{r,t} \gamma_{r,t}^c(i,m) O_{r,t}^c}{\sum_{r,t} \gamma_{r,t}^c(i,m)} \quad (6)$$

$$U_{i,m}^c = \frac{\sum_{r,t} \gamma_{r,t}^c(i,m) (O_{r,t}^c - \mu_{i,m}^c)(O_{r,t}^c - \mu_{i,m}^c)^T}{\sum_{r,t} \gamma_{r,t}^c(i,m)} \quad (7)$$

$$w_{i,m}^c = \frac{\sum_{r,t} \gamma_{r,t}^c(i,m)}{\sum_{r,t} \sum_{m'} \gamma_{r,t}^c(i,m')} \quad (8)$$

$$a_{i|j,k,l}^c = \frac{\sum_{r,t} \varepsilon_{r,t}^c(i,j,k,l)}{\sum_{r,t} \sum_j \sum_k \sum_l \varepsilon_{r,t}^c(i,j,k,l)} \quad (9)$$

where,

$$\gamma_{r,t}^c(i,m) = \begin{cases} 1, & \text{if } q_{r,t}^c = i, s_{r,t}^c = m \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$\varepsilon_{r,t}^c(i,j,k,l) = \begin{cases} 1, & \text{if } q_{r,t}^c = i, q_{r,t-1}^c = j/k/l \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$s_{r,t}^c = \max_{m=1, \dots, M_i^c} P(O_t^c | q_t^c = i, m) \quad (12)$$

For finding the hidden states, we can use the Viterbi algorithm [10] for the CHMM as below:

- Initialization:

$$\begin{aligned} \delta_0(i, j, k) &= \pi_0^A(i) \pi_0^B(j) \pi_0^C(k) b_t^A(i) b_t^B(j) b_t^C(k) \\ \psi_0(i, j, k) &= 0 \end{aligned} \quad (13)$$

- Recursion:

$$\begin{aligned}\delta_t(i, j, k) &= \max_{l, m, n} \{ \delta_{t-1}(l, m, n) a_{i|l, m, n} a_{j|m, n} a_{k|l, m, n} \} b_t^A(l) b_t^B(m) b_t^C(n) \\ \psi_t(i, j, k) &= \arg \max_{l, m, n} \{ \delta_{t-1}(l, m, n) a_{i|l, m, n} a_{j|m, n} a_{k|l, m, n} \}\end{aligned}\quad (14)$$

- Termination:

$$\begin{aligned}P &= \max_{i, j, k} \{ \delta_T(i, j, k) \\ \{q_T^A, q_T^B, q_T^C\} &= \arg \max_{i, j, k} \{ \delta_T(i, j, k)\end{aligned}\quad (15)$$

- Backtracking:

$$\{q_T^A, q_T^B, q_T^C\} = \psi_{t+1}(q_{t+1}^A, q_{t+1}^B, q_{t+1}^C) \quad (16)$$

where q_T^A, q_T^B, q_T^C represents the hidden states for each of the HMMs recognized during backtracking step of the Viterbi algorithm.

Experiments of using this coupled HMM to recognize behavior patterns for the conference room example are still undertaken and will be reported in future work.

5 Conclusions

Recognizing the behavior patterns of a system from sensor data in real time can inform a simulation model for more accurate simulations of the system under study. In this paper we propose an activity-informed dynamic data driven simulation framework and focus on behavior pattern recognition from sensor data that reflect the spatial-temporal activities of the system. We use coupled HMM to carry out behavior pattern recognition and apply it to a smart office environment example. Experimental results of this work will be reported in future work.

References

1. X. Hu, Dynamic Data Driven Simulation, SCS M&S Magazine, (2011)
2. H. Xue, F. Gu, X. Hu, Data Assimilation Using Sequential Monte Carlo Methods in Wildfire Spread Simulation, TOMACS, (2012).
3. A. Doucet, N. D. Freitas, N. Gordon (eds.), Sequential Monte Carlo methods in practice, Springer, (2001)
4. L. Rabiner, B. Juang, An introduction to hidden Markov models, ASSP Magazine, IEEE, vol.3, no.1, pp.4-16, (1986)
5. W. Hoff, J. Howard, Recognition in a dense sensor network, SNA, (2009)
6. L. Kratz, K. Nishino. Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern Models, CVPR, pages 1446–1453, (2009)
7. L. Wang, T. Gu, X. Tao, J. Lu, Sensor-based human activity recognition in a multi-user scenario, in Ambient Intelligence, volume 5859, chapter 10, pages 78±87. Springer Berlin Heidelberg, Berlin, Heidelberg, (2009)
8. D. De, W. Z. Song, M. Xu, D. Cook, X. Huo, FindingHuMo: Real-Time Tracking of Motion Trajectories from Anonymous Binary Sensing in Smart Environments, ICDCS, (2012).
9. M. Brand, Coupled Hidden Markov Models for Modeling Interacting Processes, Technical report, MIT, (1996)
10. A. Nefian, L. Liang, X. Pi, X. Liu, C. Mao, K. Murphy, A coupled HMM for audio-visual speech recognition. In Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, pp. 2013–2016, Orlando, Fla, USA, (2002)
11. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, John Wiley & Sons, New York, NY, USA, 2nd edition, (2000)