

GROUND MOBILE TARGET TRACKING BY HIDDEN MARKOV MODEL

Huynh Quan Hieu, Vu Dinh Thanh
University of Technology, VNU-HCM

1. INTRODUCTION

The mobile target tracking is one of new researches which appear in the next generation of mobile communication, where all the mobile system requires higher performances in operation. In the real mobile system, the positions of object are not known exactly and directly but we can detect them through a time sequence of measurements. This shows that there are two processes in parallel: the first process involves the real movement of the target which has to recognize and the other is the accumulated observation sequences which are provided by the first one. Such problems are the same for speech recognition or video tracking. This paper uses the results of works based on image sequences obtained by fixed surveillance cameras, or by oriented signals from BTS or by sensor array to find the series of positions of the target versus time.

While the target moves, the eventual motions and the object position are updated and so the data-base is changed. For the motion on a plane surface, e.g. mobile target moving on roads, the 2D_tracking is sufficiently used. Since the positioning data are directly updated, one smoothing path process is activated and then applied the HMM. These data are now used as the input parameters of HMM system.

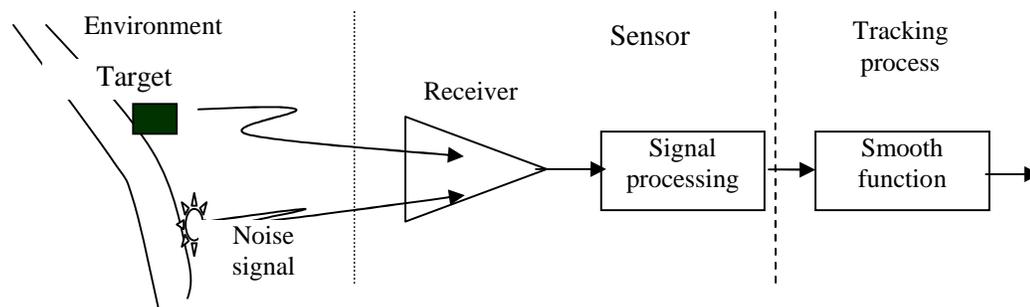
In this paper, we only focus on a Hidden Markov Model with discrete hidden states and discrete observations from the states; the signal processing is not involved here. The simulations of the model, implementing on Matlab, shows the results of tracking paths and the respective accuracies according to the learning or non-learning modes.

2. SYSTEM OVERVIEW

The input system is presented in Fig. 1[1]. Its inputs are the signals transmitted by target. A sensor receives these signals and then conveys them into database as the input of a smooth function. The output of a smooth function is the crude position measurement which will be considered as parameters of HMM system.

In the smooth functions, the parameters whose happening probabilities, when they are extracted from signal processing, is highest are regarded as the primary location of the target.

Besides, the medium noise plays an important effect on the input signals which can influence much on the accuracy of the model. The HMM is also able to decrease this effect due to its training capacity.



The block scheme of the HMM system is shown in Fig.2 [3], where the smoothed input signals are used to characterize the object motion and to generate the observations. The HMM provides the estimated locations of the target and then produces its tracking path.

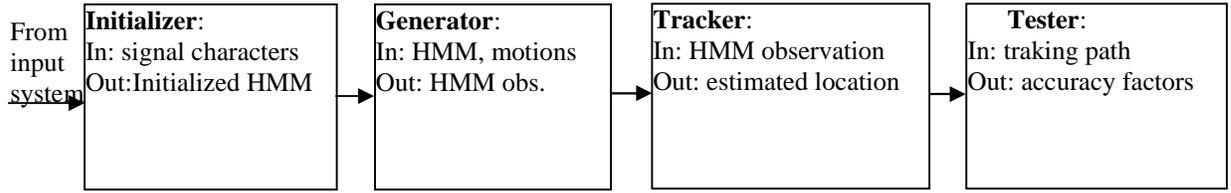


Figure 2. HMM system

In tracking time, while the target is moving from state to another state, it will send its changed location signals (azimuth angle, elevation angle, TOA, range error,) to BST which, in turn, processes signals and then transmits its data to the “**Server**”. The server does calculate the coarse positions of the target before giving the HMM system. [1][5].

3. OVERVIEW HIDDEN MARKOV MODEL

A Hidden Markov Model (HMM) consists of a set of N states, each of which is associated with a set of M possible observations. The parameters of the HMM include:

- An initial matrix of state probabilities:

$$\mathbf{p} = [p_1, p_2, \dots, p_N]^T \quad (1)$$

whose elements $p_i, i \in [1, N]$ describe the position distribution probabilities of the target over the initial state set at the beginning $t = 1$.

- A transition matrix \mathbf{A} and an observation matrix \mathbf{B} :

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \mathbf{K} & a_{1N} \\ a_{21} & a_{22} & \mathbf{K} & a_{2N} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ a_{N1} & a_{N2} & \mathbf{K} & a_{NN} \end{pmatrix}; \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \mathbf{K} & b_{1M} \\ b_{21} & b_{22} & \mathbf{K} & b_{2M} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ b_{N1} & b_{N2} & \mathbf{K} & b_{NM} \end{pmatrix} \quad (2)$$

whose elements $a_{ij}; i, j \in [1, N]$ are the transition probabilities from state i to state j , and elements b_{im} are the probabilities of observing symbol $m \in [1, M]$ given that the system is at the state $i \in [1, N]$.

- Finally, the HMM parameter set is denoted by $I = (\mathbf{A}, \mathbf{B}, \mathbf{p})$.

As usual, the HMM have three problems [4]:

- *Evaluating problem*: what is the probability of the observation O , given the model I , i.e. $P(O|I)$? \Rightarrow Solution: Forward or Backward algorithm.
- *Decoding problem*: What is the most likely state sequence given the observation O , i.e. $\arg_s [\max P(S, O|I)]$? \Rightarrow Solution: Viterbi algorithm.
- *Estimating problem*: How can we estimate parameters given the training observation sequences, $I^* = \arg_I [\max P(O|I)]$? \Rightarrow Solution: Baum-Welch algorithm.

4. APPLICATION OF HMM IN TRACKING TARGET

It finds out that the process of tracking target is likely to solve these three problems of HMM. Therefore, this research proposes three steps in tracking a target as follows:

4.1. Initiating the model

Firstly, we assume that the target is moving in a known surveillance area. The discrete model makes the area be divided into N cells corresponding to the states of the object. For instance, when the target is in cell i at time t , its model is in the state $q_i(t)$.

The movement of the target from the state i to the state j is described by transition probabilities a_{ij} in matrix \mathbf{A} . Their values often depend on the speed distribution of the target, on the geographical feature of the area and on the allowed transitions.

The state q_i of the target will emit an observation symbol $o(t)$. The observation symbol probability distributions $\mathbf{B}=[b_{ij}]$ are estimated from the observed signals from sensors. In this paper, we suppose these signals are available. The number of symbols of each state M is set equal to N .

Finally, we establish the initial state distribution \mathbf{p} , which is the probability that the model is in states at beginning. For easy tracking, we assume that the initial state is known.

4.2. Training the model

Once the model has been established, we wish that the observations received from model are of the highest probability. That is why we change the HMM's parameters for their matching to reality as much as possible. Now, we use forward-backward procedure to calculate $P(O|I)$ [5].

Forward procedure:

$$\text{- Initial forward variable: } a_1(i) = p_i \cdot b_i(O_1), \quad 1 \leq i \leq N \quad (3)$$

$$\text{- Forward steps: } a_{t+1}(j) = \left[\sum_{i=1}^N a_t(i) a_{ij} \right] \cdot b_j(O_{t+1}), \text{ with } 1 \leq t \leq T-1, 1 \leq j \leq N \quad (4)$$

$$\text{- And the result: } P(O|I) = \sum_{i=1}^N a_T(i) \quad (5)$$

Backward procedure:

- Initial backward variable: $b_T(j) = 1, \quad 1 \leq j \leq N$ (6)

- Backward steps: $b_{t-1}(i) = \left[\sum_{j=1}^N b_t(j) a_{ij} \right] b_t(O_{t-1}),$ with $2 \leq t \leq T, 1 \leq i \leq N$ (7)

- And the result: $P(O|I) = \sum_{i=1}^N b_1(i)$ (8)

The effectiveness in forward and backward procedures is almost identical. The result $P(O|I)$ is mainly used for criterion of training model.

The Baum – Welch algorithm:

- We define : $x_k(i, j) = \frac{a_k(i) a_{ij} b_j(O_{k+1}) b_{k+1}(j)}{P(O|I)}$ (9)

In which $b_{k+1}(i) = \sum b_{k+2}(j) a_{ij} b_j(O_{k+1}); \quad b_K(i) = 1$ (10)

- And we have: $g_k(i) = \sum_{j=1}^N x_k(i, j) = \frac{a_k(i) b_{k+1}(i)}{P(O|I)}$ (11)

The result :
$$\left\{ \begin{array}{l} a'_{ij} = \sum_{k=1}^{K-1} x_k(i, j) / \sum_{k=1}^{K-1} g_k(j) \\ b'_j(O_k = i) = \sum_{\substack{k=1 \\ O_k=i}}^K g_k(j) / \sum_{k=1}^K g_k(j) \\ p'_i(k=1) = g_1(i) \end{array} \right.$$
 (12)

Equation (12) produces a new set of training parameters of HMM system. The trained model $I' = \{A', B', p'\}$ has a property $P(O|I') \geq P(O|I)$. This means that the trained model parameters are more suitable to observations than the former model. Furthermore, we can learn model parameters from K observation sequences in [4]. It is proven that the model I' is becoming the real one when a range of K observation sequences is used.

4.3. Extracting the target's track

Commonly, HMM is only a model of tracking which is based on observations with an optimal algorithm. Hence, the more accurate observations, the better target's tracking is. In real system, HMM, therefore, is used after the primary estimator (for extracting the observations).

We can obtain the target's trajectory through two algorithms:

- The former is deriving the most likely state q_t at each time step individually:

$$q_t = \arg \max_{1 \leq i \leq N} [g_t(i)], \quad 1 \leq t \leq T$$
 (13)

If the matrix \mathbf{A} has the transition state probability $a_{ij} = 0$, the best state sequence may not be valid. Therefore, it is required to find an algorithm estimating the single best state sequence over the entire observation time: the Viterbi algorithm.

- Viterbi algorithm: We define the quantity:

$$d_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 \dots q_{t-1}, q_t = S_i, o_1 o_2 \dots o_t | I)$$

$d_t(i)$ is the highest probability along a single path at time t , which accounts for the first t observations and ends in state S_i .

By induction, we have

$$d_{t+1}(j) = [\max_i d_t(i) a_{ij}] b_j(o_{t+1}) \quad (14)$$

The whole Viterbi algorithm is described as follows [5]:

- o Initialization:

$$\begin{aligned} d_1(i) &= p_i \cdot b_i(o_1), \quad 1 \leq i \leq N \\ j_1(i) &= 0 \end{aligned} \quad (15)$$

- o Recursion:

$$\begin{aligned} d_t(j) &= \max_{1 \leq i \leq N} [d_{t-1}(i) a_{ij}] \cdot b_j(o_t) \\ j_t(j) &= \arg \max_{1 \leq i \leq N} [d_{t-1}(i) a_{ij}] \end{aligned} \quad (16)$$

with $2 \leq t \leq T$; $1 \leq j \leq N$

- o Termination:

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} [d_T(i)] \\ q_T^* &= \arg \max_{1 \leq i \leq N} [d_T(i)] \end{aligned} \quad (17)$$

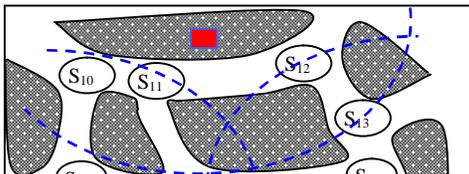
- o State sequence backtracking:

$$q_t^* = j_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1 \quad (18)$$

The study shows that Viterbi algorithm creates a better trajectory than the traditional algorithm because Viterbi algorithm decides the real states depended on all states. We suppose a parallel program, in which many state sequences are tracked concurrently. The final one is the most likelihood states.

To perform Viterbi algorithm, we must obtain observation sequences. In order to get these data, we use an array of sensors; each sensor gives an observation at each time instant. A data fusion approach is then used for linking all observations. The final observation has the highest accurate state where the target occupies. The detection of a target on each sensor depends on the correlation between the target and the sensor; such as the range and Doppler, the TOA (Time of Arrival), the AOA (Angle of Arrival), [6] [7]. HMM is only a mathematical model so that many kinds of crude observations can be employed.

5. THE MODEL SIMULATION AND RESULT



A real model as in Fig. 3 is considered:
In this model:

: the state i of the target



Figure 3: A real model area

For simulating the model, we initial its parameter as follows:

- The transition probability is determined by the target’s kinematics constraints and the surveillance area constraints [8]. For example, a_{ij} approximates to 1 if the state j is near the state i and becomes less in case of farther states or a_{ij} equal to 0, on condition that the state i can not reach the state j at each time step. For saving the number of states, states should be defined at crossroads.
- The observation probability distribution gets its value by the best quantity at the diagonal of matrix \mathbf{B} which means that the target in the state i will produce the most observation i and b_{ij} ($i \neq j$) has lower distribution.
- The initial matrix can be simplified, because the initial state is supposed to be known. For example, $p = [0,0,1,0,\mathbf{K},0]$ means the first state is $i = 3$.

The final diagram is depicted as the following Fig.5. Clearly, the estimated path without learning can not track the true path, while the estimated path with learning can. The learning method also depends on the number of recursive trainings. The program has applied statistics and probability theory into a practical situation which has a surprising result if some observations are distinct from others, especially the initial state observation. The research concludes that the sequence of estimated states depends strongly on the initial matrix, so it has to be decided carefully.

In this research, much more observations from sensors surrounding the true initial state are employed in trained p . Moreover, matrix \mathbf{B} can be attained by averaging the probabilities that sensors produce observations.

Another factor that also influences the accuracy of observations is the surveillance area of a sensor. If the surveillance area is small, a sensor can detect the target’s appearance correctly; but this requires more sensors in the same area. Therefore, the observation probability distributions \mathbf{B} depend on the distance between the sensor and the target.

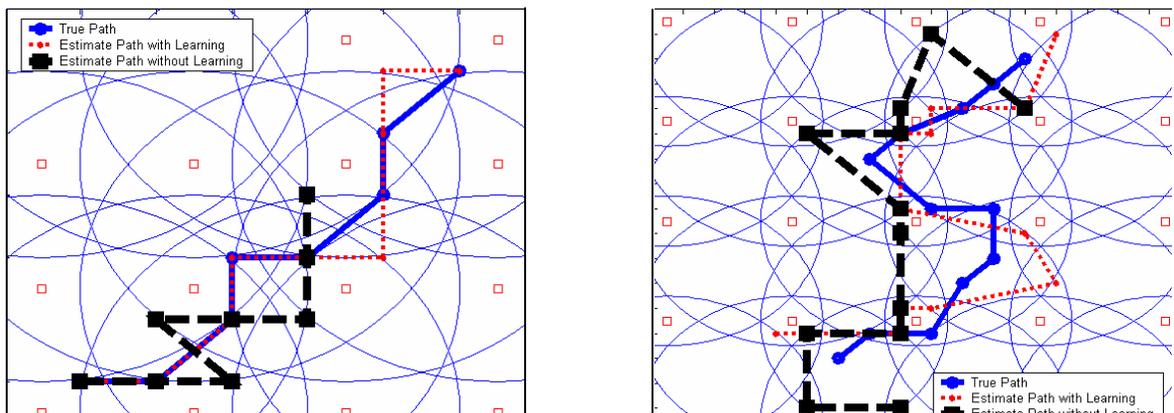


Figure 4: The mobile target's trajectory in the surveillance area with learning and non-learning methods

Figure 5: The estimated path in case of increasing the number of states.

The algorithm is still valid when the number of states increases. However, if the number of states is overwhelming, the time spent for running the algorithm is not ensure for real-time applications. The tester calculates the distant errors versus time steps as in Fig.6.

Nearly, the estimated path without learning produces more distant errors than the learning one all time steps. The results indicate that the learning model, in other words a model with its parameters matching the surveillance area, is better than the non-learning algorithm.

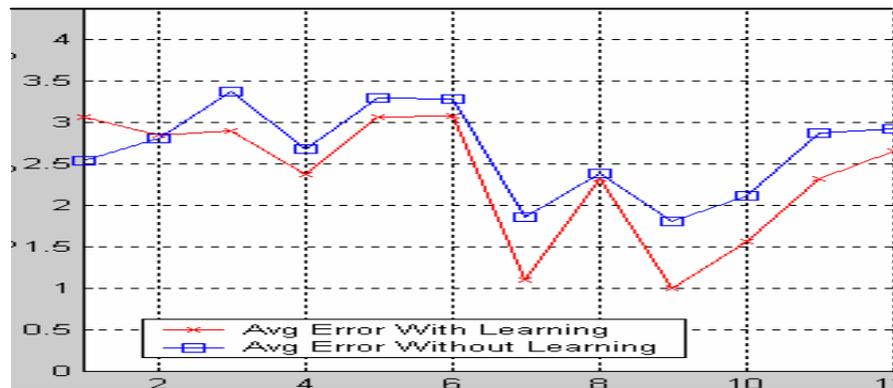


Figure 6: The distant errors versus time steps

7. CONCLUSION

This paper realizes successfully a mobile tracking simulation based on full detailed analysis of HMM. All of three steps in searching target's trajectory are done correctly by solving three problems of HMM. Furthermore, the programming of HMM is flexible allowing to widen the supervised area, i.e. examining many cases of the number of states and sensors. That the program is also recursive in order to obtain an acceptable error produces more accurate trajectories than the previous programs.

From the error estimations, it asserts that the appropriate results can be achieved whenever accurate observation sequences and good model's parameters. In addition, for the better accuracy of target's positions, observations from sensors are associated. When an ergodic HMM (i.e. $a_{ij} \geq 0$, and every transition is possible) is used, it takes much time to run the algorithm. However, time can be reduced by decreasing the number of states and/or preparing a sparse transition matrix that could have the target tracked in real time.

Finally, this paper has set up a fundamental framework that can be continuously enhanced by later algorithms.

REFERENCES

- [1]. Ashley W.Stroupe, Martin C.Martin, and Tucker Balch, *Distributed Sensor Fusion for Object Position Estimation by Multi- Robot Systems*, the Robotics Institute, Carnegie Mellon University.
- [2]. Trond Nypan, Oddvar Hallingstad, *Cellular positioning by database comparision and hidden Markov model*, Unik – Uni. Graduate center.
- [3]. Andrew Adam, Saleema Amershi, *Identifying humans by their walk and generating new motions using hidden Markov models*, CPSC 532A Topics in AI: Graphical models and CPSC 526 computer animation, December 15, 2004.
- [4]. Lawrence Rabiner and Biing- Hwang Juang, *Fundamentals of speech recognition*, Prentice – Hall International Editions.
- [5]. Chih-Chung Ke, *Literature survey on ground target tracking problem*, Center for Multisource Information Fusion State University of New York, No. CMIF 3-99.
- [6]. Richard I.A. Davis, Chrisrian J. Walder and Brian C. Lowell, *Improved classification using hidden Markov averaging from multiple observation sequences*, The Uni. Of Queensland, Australia.
- [7]. H.Yang and B. Sikdar, *A Protocol for Tracking Mobile Targets using Sensor Networks*, Research Polytechnie Institute, Troy, NY 12180.
- [8]. R. Fraile and S.J. Maybank, *Vehicle trajectory approximation and classification*, the University of reading Whiteknights, UK.