The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Position Prediction in Crossing Behaviors

A. Castro-González, Masahiro Shiomi, Takayuki Kanda, *Member,IEEE,* M. A. Salichs, Hiroshi Ishiguro, *Member,IEEE,* Norihiro Hagita, *Senior Member, IEEE*

***Abstract*—Due to the anticipated future, extensive use of robots, human beings will probably share common spaces with them. The relationships between robots and humans will be conducted at close distances. Predicting people's future positions helps robots understand human behavior and react safely and naturally. In this paper, we propose a method for predicting people's positions in crossing behaviors, i.e. different trajectories people follow when they are crossing each other. We conducted a field experiment to gather various crossing behaviors of pedestrians in a shopping mall environment and analyzed them by focusing on "hot areas" spaces where people modify their trajectories for crossing. We clustered typical crossing behaviors in hot areas and modeled them using Hidden Markov Models for predictions. Our algorithm more accurately predicts the future positions of pedestrians by considering moving direction and speed.**

I. Introduction

ROBOTICS researchers have started to explore the daily applications of mobile robots based on advances in robotics technologies. Past robotics researches have revealed that mobile robots can be used in such daily environments as museum guides [1,2], peer-tutors in schools [3], delivery staff in hospitals [4], and greeters in shopping malls [5,6].

Such mobile robots must be able to predict and track people's future positions to work more safely and efficiently in daily environments. Past related works represented the importance of predicting people's future positions. Kanda et al., who anticipated the behaviors of individuals by a few seconds, demonstrated that their system enabled robots to efficiently serve people [5]. Foka et al. predicted people's motions and exploited the predictions so that their robot could avoid people [7]. Some works also focused on robots that avoid people by predicting people's future positions [8-10]. Another related work predicted people's future position with a statistical approach and extended the method to car-like robots [11-13].

Moreover, [22] presents an approach for determining robot movements efficiently accomplish the robot's tasks while no hindering the movements of people within the environment. So they predict pedestrians' trajectories in order to plan the robot movements without disturbing them.

Nevertheless, this is not always possible: in crowded environments or narrow corridors persons walk very close and their trajectories are affected each other.

However, none of these past researches focused on crossings that affect future positions, although such behaviors often occur in daily environments. People choose a distance from others by safely changing their movement direction before they meet (Fig. 1). After crossing, they return to their original direction. We believe this is the key for mobile robots to behave safely during human robot interaction, especially in crowded or tight environments in which humans and robots usually move and avoiding crossings is not feasible. Therefore, we focus on people's crossing behavior to more accurately predict their future positions. Crossings behaviors are defined as the significant different trajectories a person follows when he is crossing with other and it depends on different aspects as direction, shape of the space or even social factors.

In this paper, we propose a method to predict people's future positions by considering their crossing behaviors. We first conducted an experiment to gather pedestrian trajectories in a shopping mall for one week. The proposed method clusters these trajectories with the k-means method [23] and makes models for each cluster using Hidden Markov Models. These models are used for predicting people's future positions in a crossing.

Fig. 1. People creating distance when crossing

II. Data Collection

A. Position Estimation with Laser Range Finders

Data were collected during an experiment in a shopping mall. The motion of people through the environment area was monitored using a ubiquitous sensor network consisting of six HOKUYO UTM-30LX laser range finders mounted around

Manuscript received March 10, 2010. The authors gratefully acknowledge the funds provided by the Spanish Government through the project called "A new approach to social robotics" (AROS), of MICINN (Ministry of Science and Innovation).

A. Castro-González and M. A. Salichs are with the RoboticsLab, Univ. Carlos III, Leganés 28911, Madrid, Spain (e-mail: {acgonzal, Salichs} @ ing.uc3m.es.

Masahiro Shiomi, Takayuki Kanda, Hiroshi Ishiguro and Norihiro Hagita are with ATR-IRC Laboratories, Kyoto 619-0288, Japan (e-mail: {m-shiomi, kanda, ishiguro, hagita}@atr.jp).

the trial area's perimeter at 110 cm height (Fig. 2).

A particle filtering technique tracked the people's trajectories through this space. Each person's location in the scan area was calculated based on the combined torso-level scan data from the laser range finders. In our tracking algorithm, a background model was first computed for each sensor by analyzing hundreds of scan frames to filter out noise and moving objects. Points detected in front of this background scan are grouped into segments, and segments within a certain size range that persist over several scans are registered as human detections.

Each person is then tracked with a particle filter using a linear motion model with random perturbations. Likelihood is evaluated based on the potential occupancy of each particle's position (i.e., humans cannot occupy spaces that have been observed to be empty). By computing a weighted average across all particles, the x-y position is calculated at a frequency of approximately 37 Hz. This tracking technique provides quite stable and reliable position data with position accuracy measured to be +/- 6 cm for our environment. Further details on this algorithm are presented in [14, 15].

Fig. 2. Laser range finder

B. Data collection in a shopping mall

Human motion data were collected for seven consecutive days in a shopping mall environment from 11 am to 8 pm each day (Fig. 3). We chose this time schedule because the shops open at 11 a.m., and the number of visitors drops after 8 p.m. In this environment, the major flow consisted of visitors crossing the space from the left to the right or vice versa, generally taking about 50 seconds to go through it. The users were mainly commuters, couples, and families. We gathered about 71,425 visitor trajectories[1].

III. Analysis of Crossing Behavior

Using the laser range finder system described in the previous section, a vast amount of data was stored in a database. We analyzed how people cross using the stored data with a graphical interface tool on the experimental data stored in the database [16].

[1] In this study, we obtained approval from shopping mall administrators for this recording under the condition that the information collected would be carefully managed and only used for research purposes. The experimental protocol was reviewed and approved by our institutional review board.

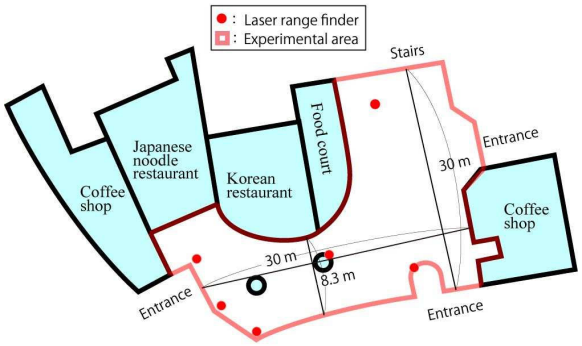

Fig. 3. Shopping mall environment

A. Hot area

We first focused on a *hot area*, which is a space where people modify their trajectories when they are crossing. In most cases, people modified their movement directions at a distance to another person of less than five meters. Therefore, we defined the size of the *hot area* as a circular area within a five-meter radius around a person. In other words, our system focuses on all movements in that area and predicts the future positions of people within it. At Fig.4 hot area is presented and a character is the center of this area where other person is walking in.

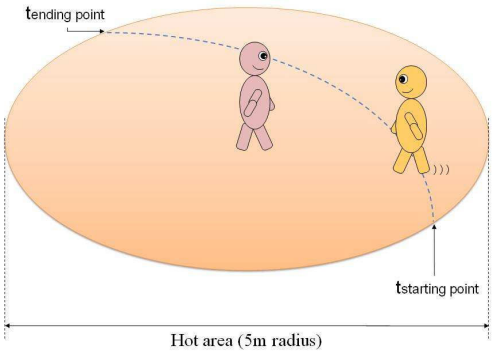

Fig. 4. *Hot area* and *crossing time* concepts. Dashed line represents the trajectory a person follows inside other's hot area.

B. Crossing time

Defining crossing time between two persons as the time a person remains in the other's hot area, it is one crucial metric to distinguish among crossing behaviors and others. For example, when two people are walking together, their trajectories are always observed in a hot area because their speeds and directions are almost identical. In such behaviors, the crossing time will be much longer than in crossing behaviors. For example, if two people are walking together, it takes around 50 seconds from entrance to exit. Therefore, we defined the maximum crossing time as ten seconds, calculated based on the hot area's dimension, the average walking speed in the environment (around four kilometers per hour), and observations from stored trajectories. This idea can be observed at Fig.4 where a person is walking in other's hot area at $t_{starting\ time}$ and leving it at $t_{ending\ time}$ instant. Consequently, crossing time is $t_{ending\ time}$ minus $t_{starting\ time}$.

C. *Changes of speed in crossing behavior*

We also analyzed the speed during crossings because it directly affects the future positions in those crossing. In most cases, no significant speed variation was found during a crossing, even when direction of movement has been changed. At Fig.5 we show the typical relation between walking speeds (blue and red lines) and distance (black line) of two people crossing. Distance decreases until a minimum is reached (around two meters) and then it increases again; at the same time, both walking speeds are not affected. Therefore, we concluded that average speed is a reliable value for predicting people's future positions in crossing behaviors.

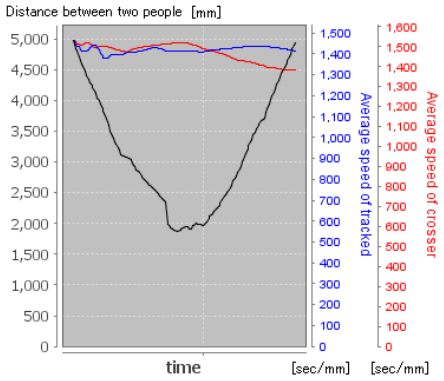

Fig. 5. Variation of speed during a 2-person crossing. Concave line means distance between two people. Upper two lines mean average speeds of two people crossing.

D. *Selecting reasonable trajectories*

Based on the previous analysis, we extract crossing behaviors from the stored trajectories. We have defined crossing behaviors as the significant different trajectories persons follow when they are crossing each other inside the hot area. Therefore, in our approach, trajectories are the key element. Trajectory data have to be filtered to avoid noise from false detections in the position tracking system. Then filtering data are required to maintain reasonable trajectories. So we conducted four filtering processes.

First, since a minimum lifespan was established for every trajectory, any people found to exist less than five seconds were not considered.

Second, as indicated above, the crossing time is limited to ten seconds so any trajectory over this time will be removed from the data set.

In daily human environments, people from entry and exit points, i.e. doors, suddenly appear and disappear; then, no general trajectories are observed. These situations are not covered in this work. According to it, in the third filtering, only crossing trajectories starting and ending farther than four meters are taken into account

Fourth, we removed trajectories whose missing duration is more than three seconds. It means that trajectories which total lost data sums more than three seconds are not considered.

After the filtering process was completed and clarified, about 11,000 crossing trajectories were obtained.

IV. Extracting Crossing Behaviors

Because of the big amount of trajectories obtained from the previous steps, clustering is needed in order to define the crossing behaviors. Otherwise, real time applications using the proposed method are not feasible. Due to the nature of our behaviors, trajectories must be compared for clustering them and establishing different patterns. To compare trajectories inside the hot area, every two subjects that move within a distance less than five meters away are studied. One of the subjects is considered the reference point because it is in the center of the hot area, and the other one is moving inside it. The relative trajectory between both characters is computed and called the crossing trajectory, which is defined as the relative trajectory between two persons when one of them is inside the other's hot area. These trajectories are considered as a sequence of points, so a crossing trajectory t consist of a sequence of N positions $t = \{p^0,..,p^{N-1}\}$ where each position has x and y coordinates.

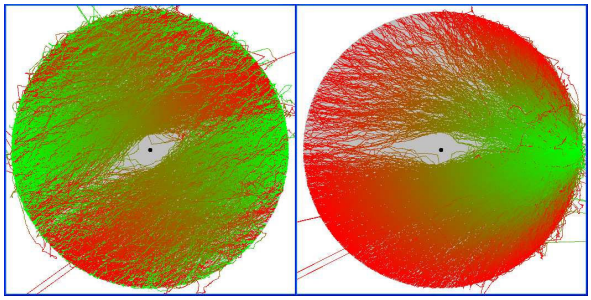

Fig. 6. (a) raw trajectory data and (b) normalized trajectory data. Green (white gray in monochrome printing) means starting point and red (dark gray in monochrome printing) ending point.

A. *Normalization and clustering of crossing behaviors*

After gathering the crossing trajectory data and before clustering, we need a method to fairly compare them to acquire different groups of trajectories with similar characteristics. In our approach, normalization was accomplished by rotating all trajectories to the same starting angle; hence similar trajectories can be recognized despite different entry and exit angles, life spans, speed, or other variables. After normalization, all crossing trajectories have the same starting zone, and different courses can be observed. At Fig. 6(a) raw data from all trajectories are presented. Trajectories go from starting point (green color) to ending point (red color). Fig. 6(b) represents all trajectories after normalization and reader can observe how all starting points are in the same area.

K-means algorithm [17, 18] was applied to the crossing trajectories to create typical crossing behaviors. This algorithm's basic form is used for classifying discrete objects ignoring the number of dimensions. The algorithm was adapted to our problem for classifying trajectories, which are considered a sorted sequence of points in the space.

The procedure follows a simple and easy way to classify a given trajectories set through a certain number of clusters

(assume k clusters). The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to associate each trajectory to the nearest centroid. When no trajectory is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids as the average trajectory of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same trajectories sets and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done, it is when centroids do not move any more.

As described in [19], when k-means is used to cluster the trajectories, we need to compare two trajectories by measuring the distance between them. In this case, the average Euclidean distance was implemented between every pair of points in the trajectories. This implies that both trajectories must have the same number of points. Therefore the trajectory with the fewest points is linearly interpolated until the needed number of points is obtained. Then, the distance between two trajectories is calculated based on this expression:

$$dist(t1,t2) = \frac{\sum_{i=0}^{N} dist(p1_i, p2_i)}{N},$$

where *t1* and *t2* are trajectories with *N1* and *N2* points, respectively. *N* is calculated as $\max(N1, N2)$ and $p1_i$ and $p2_i$ are the *i-th* points of *t1* and *t2* respectively.

At this point, the k-means algorithm is applied (Fig. 7). First, we define the number of clusters and the initial centroids. Firstly, the centroids were chosen randomly, but the resulting clusters were not satisfactory. Thereby the initial centroids were evenly assigned by pre-classification after normalization: crossing trajectories were split based on the exit angles and groups were made every five degrees creating 72 groups (360º/5º). For every group, its average trajectory is calculated and its initial centroid is chosen as the closest trajectory to it.

A trajectories group C_k consists of a set of trajectories $C_k = \{t_0, \cdots, t_M\}$ where each trajectory t_i consists of a sequence $t_i = \{p_i^0, \ldots, p_i^{Ni}\}$ of positions. With the aim of obtain its average trajectory $\overline{t_k}$, trajectories in the same cluster are modified to have the same number of points ($N_i = N, \forall i$). Therefore $\overline{t_k}$ is determined by sequence $\overline{t_k} = \{\overline{p^0}, \ldots, \overline{p^N}\}$ of the average points for every position. So $\overline{p^i}$ is defined as:

$$\overline{p^i} = \frac{\sum_{j=0}^{M} p_j^i}{M};\ 0 \le i \le N$$

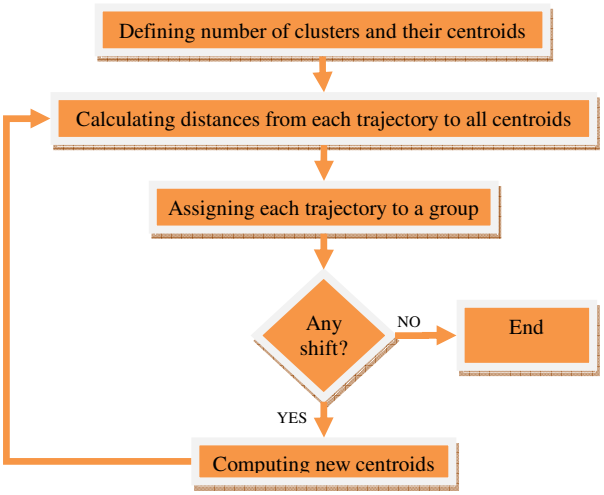

Fig.7. Adapted k-means algorithm

Now, using these initial centroids as the characteristics for all clusters, the distances from each trajectory to every centroid are calculated. At this, every trajectory is shifted to the group represented by its closest centroid. New centroids are computed using the trajectories from every group. Then the algorithm starts calculating the distances to centroids again and continues to run until convergence is reached, it is no shifts of trajectories occur among groups.

The final centroids represent the different behavior patterns related to each cluster of trajectories.

V. Estimating Future Positions

For estimating a person's future positions in a crossing, the person's crossing behavior has to be determined by his trajectory. In the previous section, the trajectories representing different behaviors were calculated and they will be compared with the person's trajectory to assess his behavior. As the estimation tool, Hidden Markov Models (HMM) [20, 21] have been chosen. Markov Models are very rich in mathematical structures, so they can be used in a wide range of applications. Previous works, such as [7] [24], from different fields have proved that HMM is a reliable stimation tool.

Deriving HMM from clustering

Since every behavior will have an associated HMM, an HMM is created for every group obtained in the clustering step. We used the mean trajectory in each cluster as the base for each HMM.

HMM λ is defined by the next five elements: $\lambda = (Q, O, \pi, A, B)$

- Q (hidden states): $Q = \{q_1, \ldots, q_N\}$.

The hidden states are related to different points of the average trajectory. The distance between two consecutive hidden states, defined as *ds,* is used for calculating all the hidden states in succession. This parameter affects the system accuracy. In the experiments, *ds* was set to 500 mm. All our HMM will have at least two states: starting and ending states. Therefore, a 1300 mm length trajectory will have four hidden

states which represent positions at 0 mm, 500 mm, 1000 mm and 1300 mm from the starting point.

- O (observations): $O = \{o_1, \ldots, o_M\}$

  This is the set of observations per state. In our system, the coordinates of every position were used as observations.

- π (initial state probability): $\pi = \{\pi_1, \ldots, \pi_N\}$

  For each state, this determines the probability of its being the first state. Due to the normalization of the crossing trajectories, the initial state of the HMM is well known: $\pi_i = \begin{cases} 1, i = 1 \\ 0, 1 < i \leq N \end{cases}$

- A (state transition probability):

  $A = \{a_{ij}\} = P(q_{t+1} = q_j | q_t = q_i);\ 1 \leq i, j \leq N$

a_{ij} is the probability of staying at q_j in time $t+1$ after q_i on t. Because of the idiosyncrasy of the crossing trajectories, the HMMs are left-to-right, meaning that no backward transitions can be observed (Fig. 8). These transition probabilities are set to zero ($a_{ij} = 0, j < i$), and the rest are evenly distributed.

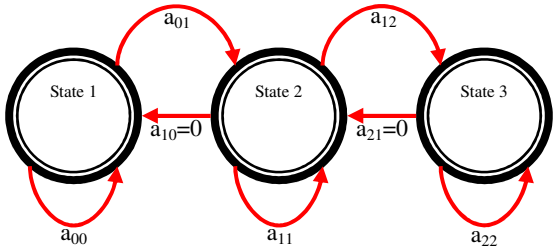

Fig. 8. State transition probabilities in a left-to-right HMM with N=3

- B (observation probability):

  $b_j(k) = P(o_k \text{ at } t | q_t = q_j),\ 1 \leq j \leq N, 1 \leq k \leq M$

$b_j(k)$ is the probability of observation o_k at state q_j. In this research, each hidden state has a bivariate Gaussian distribution utilized as the observation probability distribution. The coordinates of positions represented by the hidden states are used as the distribution's mean values. The standard deviation and covariance matrix are calculated when the hidden states are computed. All of these parameters are required to define the bivariate Gaussian distribution for each state.

The bivariate Gaussian function refers to two distributions, a and b, and can be represented as $X = \begin{pmatrix} a \\ b \end{pmatrix}, X \sim \mathcal{N}(\mu, \Sigma)$, where mean matrix $\mu = \begin{pmatrix} \sigma_a \\ \sigma_b \end{pmatrix}$ and covariance matrix $\Sigma = \begin{pmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{pmatrix}$ are calculated. The probability is calculated based on the next equation:

$$P(X) = \frac{1}{2\pi \|\Sigma\|^{1/2}} exp\left(-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)\right).$$

Because both distributions are considered independents, σ_{ab} is set to zero. Variances are calculated, but the minimum value is set to 1.

In Fig. 9, several behaviors are represented. In each one, the plotted trajectories defined the behavior. The black line represents the average trajectory for the corresponding cluster, and the black dots correspond to the states in the HMM that represent the behavior.

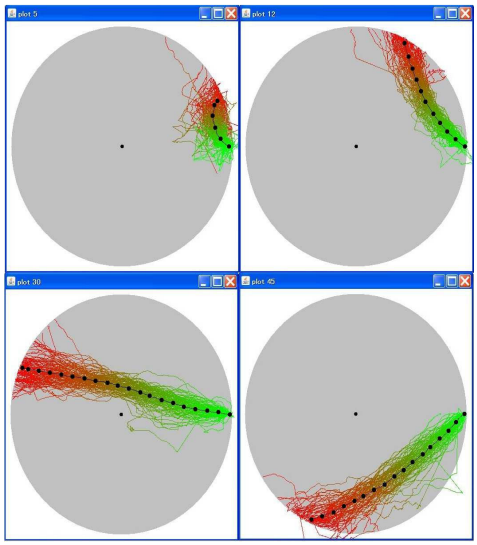

Fig. 9. Examples of clustered crossing trajectories

After the HMM models have been created, the least significant behaviors are rejected. If the number of trajectories associated with a behavior is less than 10, this behavior has to be removed because it is not adequately representative. Finally, after calculating, we obtained 64 behaviors, utilizing 64 HMMs. Fig. 10 displays all average trajectories and their corresponding states in HMMs. After filtering the noisy data and eliminating the strange trajectories, 10,335 trajectories were used to create all the HMMs.

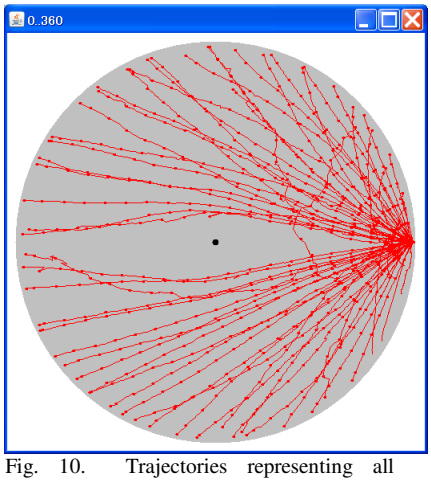

Fig. 10. Trajectories representing all behaviors used in experiments

Estimating person´s behavior

Once all HMMs have been calculated, the estimation step follows. Observations of a person's positions are required to appraise behaviour. Every new observation is added to a sequence of observations computed in all HMMs. The HMM responding with the highest probability of the observation sequence will reveal the current behaviour. Given the observation sequence and the HMM, the observation probability is computed by the Forward-Backward algorithm [21].

Then, during a defined period called observation time t_{obs}, the system stores observations that will be evaluated later with the HMMs. During this observation time, the average speed is also computed. Finally, we get the most probable HMM for the observation sequence and consequently the most probable behavior. The average trajectory for the selected behavior is the guideline for estimating the future position: the closest point in the average trajectory to the last observation is the start point (p_{st}) for the estimation. Using the average speed previously calculated and estimation time t_{est}, displacement (d) is calculated. Then the estimated point after t_{obs} seconds observation becomes the point in the most probable average trajectory after covering d from p_{st}. As explained previously, since the person's walking speed is not significantly affected when she crosses another person, speed is considered constant, and the results will not be heavily affected.

VI. Experiments

We experimentally investigated the performance of our proposed system using trajectories randomly selected from the data collection. These trajectories were not used for setting up the system, so no relationship exists between them and our models.

The trials simulated real-time data acquisition by defining observation time step Δot as the interval between two observations in a row. This means that the chosen trajectory will be covered by "jumping" Δot ms before a new observation (the corresponding coordinates) is extracted from it. Δot is constant.

In this experiment, the future trajectory positions were predicted and compared with the actual succeeding positions. Future positions are considered the coordinates of the trajectories several seconds after the last observation was acquired. In our experiments, we measured the accuracies of predicting future positions after one/two seconds of observation (t_{obs}). Future positions are appraised after one, two, three, four, and five seconds (t_{est}). This is explained at Fig. 11.

Our method was compared with the baseline method, which uses average speed and straight direction calculated by the most recent observations. In both cases, the same 300 trajectories constitute the testing data set. The trajectories selected for the trials have to be long enough in time, i.e., life spans of at least $t_{obs} + t_{est}$. The prediction process is executed until the rest of the lifespan for each trajectory becomes less than t_{est}.

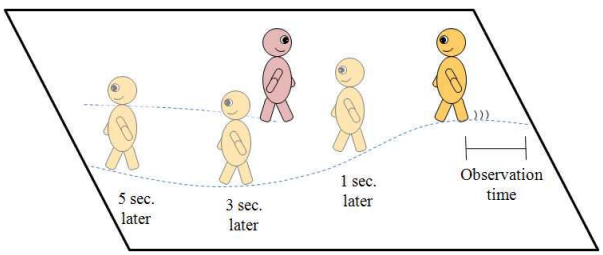

Fig. 11 Configuration of the experiments

Table I Average errors values from our proposed method and the baseline [mm].

Estimation time	Observation time			
	1 second		2 seconds	
	Proposed method	Baseline	Proposed method	Baseline
1 sec. later	829.44	1966.57	904.75	6582.94
2 sec. later	1927.93	1994.49	1239.94	2166.08
3 sec. later	1915.37	2535.85	1741.80	3029.60
4 sec. later	2381.59	2827.30	2165.54	3749.49
5 sec. later	2505.91	3549.93	2477.14	3482.29

A. Experimental results

Table I presents the average error values from the experiments. These data is used in the next plots for a better comprehension.

Figure 12 shows the average error when predicting the future positions after one second of observations. The proposed method outperforms the baseline for all situations and achieved less than 1-m prediction position error one second later. As expected, errors increased when the estimation continues in time, but our method still achieved 2.5-m prediction error five seconds later.

Figure 13 shows the results after two-second estimations. The rest of the configuration in the experiment is identical as the former one. The proposed method achieved less than 1-m prediction position error one second later and 2.5-m error when predicting the future position five seconds later. Our method gets better results here as well, and the difference with the baseline is bigger now. This means that the longer the observation, the more our method outperforms the baseline.

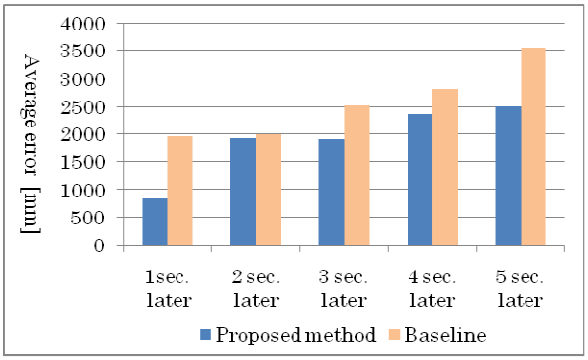

Fig. 12 Average error after 1-sec observations

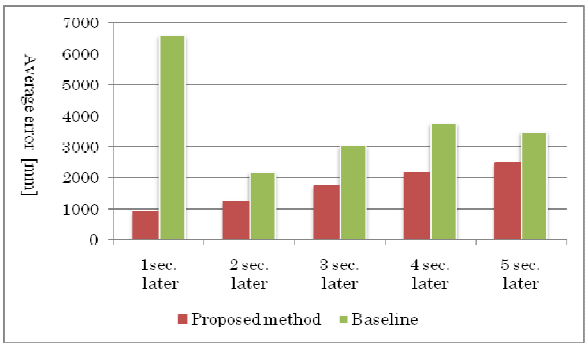

Fig. 13 Average error after 2-sec observations

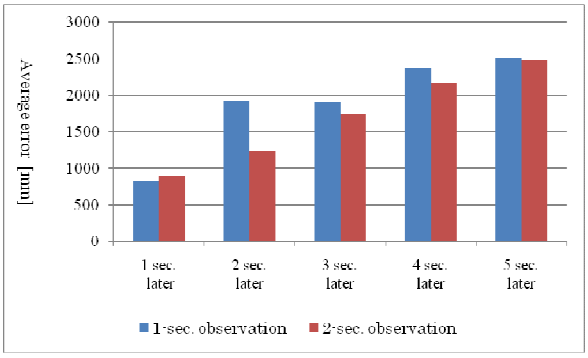

Fig. 14 Average errors of proposed method

Figure 14 compares the results of our method with one- and two-second observations. After two seconds, the error is smaller in all cases, except for the first situation. This is logical; after longer observations, the estimating error decreases because more data have been analyzed. At the same time, as estimations continue, both methods get very similar results. Then the results of estimations farther in the future will be more or less equal despite observation times.

We conclude that our method works better than the baseline because of the nature of the considered behaviors in each one. The former is based on natural behaviors that include avoiding movements, detours, and human reactions. The last one just supposes completely unnatural linear behaviors, and therefore its outputs are not relevant.

VII. Discussion

A. Implementation for real-time prediction

In our experiments, we conducted offline processing to predict the future positions of people. To achieve real-time prediction, the processing time is crucial. In our method, the maximum time for evaluating a sequence of observations with all HMMs is around 30 ms in a Core2Duo (3.00 GHz) processor with 3.25GB of RAM, so applying it to such real-time systems as robotics is feasible. Heavy calculation cost is needed to model crossing behaviors, but that is an offline process. After models are calculated, they can be used for real-time estimation. Therefore, we do not think that processing time is a problem for real-time prediction.

Another problem is dealing with strange trajectories in real-time prediction; however, this problem can be overcome because such trajectories are very rare. During experiments, the percentage of weird trajectories was minor. We tried to understand these strange trajectories that apparently reflected persons who were aimlessly wandering. Even the person herself will have difficulty precisely estimating such behavior or such a trajectory.

Very short trajectories also present estimation problems. HMM is used as an estimation algorithm, so achieving good results requires a sequence of observations. If the trajectory is too short, it means that there will not be enough observations of the trajectory for a good appraisal.

B. Feasibility in other environments

We expect to easily apply our proposed method to other environments because it can model crossing behaviors by gathering crossing trajectories in any environment. Once models are computed for a new environment, a calculated relative trajectory can be used to estimate future positions of crossing people. Such a wide range of applications is one of our method's strengths.

One possible future work is to predict the future position of people who are crossing a mobile robot. We expect that their crossing behavior involving robots will be different from their crossing behaviors with people. Our method can model the crossing behavior between a robot and people as well as the crossing behavior of people. By applying our method for this purpose, robots can move more safe and natural by considering the future positions of the crossing people.

VIII. Conclusion

Predicting people's future positions is one of important ability as well as tracking their positions for mobile robots that act in daily environments. Because robots will share common rooms with persons, avoiding crossings is not possible. Hence, to predict future positions more accurately, we focused on the crossing behavior of people and proposed a method to predict the future positions of people who are crossing each other. This method will be applied later to robot-person crossings on an actual scene.

We first conducted a field experiment to gather various crossing behaviors of pedestrians in a shopping mall environment to analyze them. We focused on a "hot area," a space where people modify their trajectories when they are crossing. We clustered typical crossing behaviors in hot areas and modeled them using Hidden Markov Models for predictions.

We conducted experiments to investigate the performance of our proposed method. Using the gathered trajectories from a shopping mall, we measured the accuracies of predicting future positions after one/two seconds of observations. Future positions were appraised after one, two, three, four, and five seconds. The proposed method achieved less than 1-m prediction position error one second later and 2.5-m prediction error of future positions five seconds later.

The presented method can work on real time applications because heavy calculations are processed off-line. Estimation process is in the order of 30 ms.

In future experiments, the performance of our method will be tested on real environments.

Acknowledgments

We wish to thank the staff of the Asia and Pacific Trade Center Co., Ltd. for their kind cooperation.

References

[1] W. Burgard, A. B. Cremers, D. Fox, D. Hänel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, The interactive museum tour-guide robot, Proc. of National Conference on Artificial Intelligence, pp. 11-18, 1998.

[2] R. Siegwart et al., Robox at Expo. 02: A Large Scale Installation of Personal Robots, Robotics and Autonomous Systems, 42(3), pp. 203-222, 2003.

[3] T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro, Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial, Human Computer Interaction, 19(1-2), pp. 61-84, 2004.

[4] Bilge Mutlu and Jodi Forlizzi, Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction, Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, March 12-15, 2008, Amsterdam, The Netherlands.

[5] Kanda, T. et al., Who will be the customer?: A social robot that anticipates people's behavior from their trajectories, *UbiComp2008*, 2008.

[6] M. Shiomi, T. Kanda, D. F. Glas, S. Satake, H. Ishiguro, and N. Hagita, "Field Trial of Networked Social Robots in a Shopping Mall," Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems (IROS2009), 2009. (to appear)

[7] A. Foka and P. Trahanias, "Real-time hierarchical POMDPs for autonomous robot navigation," Robot. Auton. Syst., vol. 55, no. 7, pp. 561–571, 2007.

[8] Rongxin Jiang, Xiang Tian, Li Xie, and Yaowu Chen, "A Robot Collision Avoidance Scheme Based on the Moving Obstacle Motion Prediction," Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management.

[9] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Probabilistic Rapidly-exploring Random Trees for autonomous navigation among moving pedestrians," Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems (IROS2009), 2009.

[10] E. Owen and L. Montano, Motion planning in dynamic environments using the velocity space, RSJ International Conference on Intelligent Robots and Systems (IROS'2005), Edmonton, Alberta, Canada, 997-1002, August 2-6, 2005.

[11] Montesano, J. Minguez, and L. Montano, Modeling the Static and the Dynamic Parts of the Environment to Improve Sensor-Based Navigation, In Proceedings of the International Conference on Robotics and Automation (ICRA), pages 4567–4573, Barcelona, Spain, 2005.

[12] Dizan Vasquez and Thierry Fraichard, 2004, "Motion Prediction for Moving Objects: a Statistical Approach," In: Proceedings of the 2004 IEEE International Conference on Robotics & Automation, 2004

[13] Large, F., Vasquez, D., Fraichard, T., and Laugier, C., "Avoiding cars and pedestrians using velocity obstacles and motion prediction," IEEE Intelligent Vehicles Symposium, 2004.

[14] Glas, D. F., Miyashita, T., Ishiguro, and H., Hagita, N. Laser Tracking of Human Body Motion Using Adaptive Shape Modeling, In Proc. IROS2007, (2007), pp. 602-608.

[15] Glas, D. F., Miyashita, T., Ishiguro, and H., Hagita, N. Laser-Based Tracking of Human Position and Orientation Using Parametric Shape Modeling, Advanced Robotics 23 (2009), pp. 405-428.

[16] Tijn Kooijmans, Takayuki Kanda, Christoph Bartneck, Hiroshi Ishiguro and Norihiro Hagita, Accelerating Robot Development through Integral Analysis of Human-Robot Interaction, IEEE Transactions on Robotics (Special Issue on Human-Robot Interaction), 23(5), pp. 1001-1012, 2007.

[17] Loyd, S. P. Least squares quantization in PCM. IEEE Transactions on Information Theory 28 (2), March 1982, pp. 129-137.

[18] MacKay, David. Chapter 20. An Example Inference Task: Clustering. Information Theory, Inference and Learning Algorithms. Cambridge University Press. pp. 284-292. ISBN 0-521-64298-1.

[19] Stuart Russell and Peter Norvig. Artificial Intelligence A Modern Approach 2nd Edition. Pearson Education. 2003. ISBN 0-13-080302-2.

[20] L.R. Rabiner and B.H. Juang. An Introduction to Hidden Markov Models. IEEE ASSP Magazine. January 1986.

[21] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77 (2). February 1989.

[22] Brian D. Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew Bagnell,, Martial Hebert, Anind K. Dey and Siddhartha Srinivasa. Planning-based Prediction for Pedestrians. IEEE/RSJ International Conference on Intelligent Robots and Systems. October 2009. St. Louis, USA.

[23] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297. 1967.

[24] Robert B. Scharpf, Giovanni Parmigiani, Jonathan Pevnser, and Ingo Ruczinski, "A HIDDEN MARKOV MODEL FOR JOINT ESTIMATION OF GENOTYPE AND COPY NUMBER IN HIGH-THROUGHPUT SNP CHIPS" (February 2007). Johns Hopkins University, Dept. of Biostatistics Working Papers. Working Paper 136.