Human Trajectory Forecasting In Indoor Environments Using Geometric Context

Pranav Mantini
University of Houston
4800 Calhoun Road
Houston, TX 77004
pmantini@cs.uh.edu

Shishir K. Shah
University of Houston
4800 Calhoun Road
Houston, TX 77004
shah@cs.uh.edu

ABSTRACT

A human trajectory is the likely path a human subject would take to get to a destination. Human trajectory forecasting algorithms try to estimate or predict this path. Such algorithms have wide applications in robotics, computer vision and video surveillance. Understanding the human behavior can provide useful information towards the design of these algorithms. Human trajectory forecasting algorithm is an interesting problem because the outcome is influenced by many factors, of which we believe that the geometry of the environment plays a significant role. In addressing this problem, we have built a model to estimate the occupancy behavior of humans based on the geometry and behavioral norms. We also develop a trajectory forecasting algorithm that understands this occupancy and leverages it for trajectory forecasting in previously unseen geometries. We perform experiments to quantify the error between our prediction model and the trajectories obtained from real world human subjects and compare them to state of the art models. Results obtained suggests a significant enhancement in the accuracy of trajectory forecasting by incorporating the occupancy behavior model.

Keywords

Trajectory forecasting, human occupancy behavior, 3D geometric context

1. INTRODUCTION

Given a human subject and their destination, trajectory forecasting deals with predicting or estimating the likely path a subject will take to reach the destination. Trajectory forecasting has a variety of applications. In robotics, it can be used for robot motion planning, in surveillance it can be used for predicting the future location of subjects and could also be used to improve the accuracy of computer vision algorithms for tracking, re-acquisition, etc.

ICVGIP '14, December 14-18, 2014, Bangalore, India
Copyright 2014 ACM 978-1-4503-3061-9/14/12 ...$15.00
http://dx.doi.org/10.1145/2683483.2683547 .

Networked cameras are widely used for monitoring human activity in public areas. Camera networks spanning from hundreds to thousands of cameras per network is a common occurrence in busy public locations like airports. Most of these cameras might have non-overlapping fields of view. A holistic automated surveillance system cannot infer a semantic understanding of the scenario without a model for linking the observed actions from individual cameras. The surveillance system should have an understanding of the 3D geometry of the environment it is present in, along with an understanding of the relation between the cameras. Considerable effort is focused on automatic generation of 3D models for outdoor and indoor environments [5, 6, 17, 28, 7]. Furthermore, reasonable attempts have been made in understanding the camera topography [8, 22, 11, 31] for applications like tracking [16, 27] and re-identification [24, 21, 20]. In these cases, it is very essential to predict the trajectories of humans based on the geometry of the environment. For example consider a re-identification problem, where a human is observed in two different cameras in the same network. Estimate of the trajectory starting from the observation in the first camera to the destination in the second can impart an approximate spatial and temporal knowledge of the human's actions. This can assist in designing robust re-identification algorithms. Similarly, human motion and estimation of trajectories is critical in urban planning where the design of new public spaces and their geometries will be influenced by simulations of expected human occupancy and their movements [3, 14].

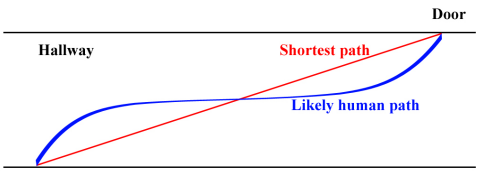


Figure 1: Shortest path Vs. likely path

Human motion is influenced by a multitude of factors, many of which are driven by perception. It is well understood that 3D geometry and the physical world imposes specific constraints on human motion. In many cases, interaction of humans with the surrounding geometry is not explicitly modeled. In general, we can agree that the trajectory that a human subject would take is motivated by the target destination they are trying to reach, but it is not

necessary that they would always take the shortest path defined by the geometry. Though the human subject's main motivation is to reach the destination in the shortest time, they may still subconsciously follow some behavioral norms. For example if we are trying to reach a door that is on the left hand side at the far end of a hallway, we would walk in the center or close to the center of the hallway almost all the way until we get close to the door as shown in Figure 1. In this case, the shortest distance is to stay as close to the left wall of the hallway as it is physically possible, but we rarely see such behavior. So we believe that this behavior of ours, though at a subconscious level, is being influenced by the surrounding 3D geometry and behavioral norms. Our continuous interaction with different geometries in various environments over time may have led to the evolution of this behavior. In this paper we mainly deal with indoor environments whose 3D geometry is available and static. If we consider a scenario with a starting point and a destination that are located at either ends of the hallway, and we observe the trajectories followed by a large number of human subjects, we can assume that certain points on the floor in the hallway are accessed more often then other points on the floor when the human subjects are traveling the hallway. So there might be a certain distribution or a occupancy map to the floor. This occupancy map can be very helpful in forecasting our trajectory.

In this paper, we choose to model trajectory forecasting as a Markov model. So the trajectory is defined as a transition from one point/state on the floor to another. In order to sample points to form a trajectory, we will need a probability transition matrix. This transition matrix should have a higher probability of choosing points that are closer to the destination, and also have a high frequency in the occupancy map. If we have an energy function that calculates and assigns higher energy to such points, compared to those that do not, we can create our transition matrix using an energy maximization framework. We can construct a distance map for every point on the floor that the energy function can use to determine how close the point is to the destination. We can also estimate an occupancy map that the energy function can use to determine how accessible the point is with respect to the geometry. These two maps enable us to build a transition matrix for sampling points to form trajectories.

To construct the occupancy map for a novel geometry, first we observe the occupancy map of the floor for a known geometry. For every point on the floor, we can calculate some geometric features to capture the spatial structure of the environment surrounding it. Further if we model a relationship between the geometric features and the observed occupancy map, we can use this model to estimate the occupancy map of the points on any floor based on its geometric features. In this paper we propose a set of geometric features and a linear model to solve this problem. The main contributions of this paper are:

- We propose a set of novel geometric features that describe a point on the floor with respect to the geometry of the 3D environment around it considering the perception of the geometry within the context of behavioral norms.
- Given the geometry of an environment, we propose a method to estimate the human occupancy map.
- Given the geometry of an environment, the location of start and destination of a subject, we propose a model to forecast the trajectory of the human subject by leveraging this developed human occupancy map.

2. RELATED WORK

Trajectory forecasting is a widely researched field. A complete survey was done by Morris and Trivedi [25]. Traditional models for forecasting have followed a two stage approach, a data-driven learning and then a prediction stage. In the learning stage, the trajectory patterns are observed for the scenario and a model is learned. In the prediction stage, the initial information about the trajectory is used along with the learned model to predict future actions. Junejo *et al.* [18] used minimum graph cuts with edges weighted by Hausdorff distance for training and a combination of spatial, velocity and curvature features for trajectory prediction of real world outdoor pedestrians. Vasquez and Fraichard [32] used pairwise clustering to learn trajectory models of human subjects within indoor scenarios, and the prediction is done using the mean and the variance of the clusters. Weiming *et al.* [15] learnt the trajectory model of real world pedestrians and toy cars in a model scene using fuzzy self organizing neural networks. Markov model was used to model the piecewise trajectories of vehicles by Paki and Martial [9]. A bank of previously observed switched dynamic models were used for predicting the trajectories of humans in indoor scenarios by Nascimento *et al.* [26]. Vasquez and Fraichard [33] used a hidden Markov model based on growing neural gas algorithm for trajectory forecasting within indoor scenarios. Saleemi *et al.* [29] modeled the trajectory patterns of real world pedestrians using a kernel density estimator and a unified Monte Carlo Markov Chain framework was used for predicting the likely trajectories. All of the above models work at a pixel level on the 2D images and does not explicitly model the effect of the environment on the humans that may influence the shaping of their trajectory. Moreover, the models are scene dependent and cannot be transfered to a new geometry. So, even a small change in the environment like introducing a new object in the scene would require a complete new set of training data.

Recently proposed prediction models employ a model-driven approach that accounts for the environment. Bhattacharya *et al.* hypothesized the trajectories around obstacles for robot motion planning in an environment by forming homotopy classes [12, 4]. Ziebart *et al.* [35] used maximum entropy inverse optimal control for prediction of human trajectories in outdoor scenarios, and also took the environment into consideration. Kitani *et al.* [19] model the environment using semantic scene labeling and perform prediction using inverse optimal control. More recently in [34] a visual prediction of motion was also generated along with trajectory forecasting. While [19] is closely related to our approach, the effect of the environment modeled on the formation of the human trajectory is limited to a small set of scene labels. The scene understanding is at an image level. In our method, we build and use the actual 3D model of the entire geometry, and the amount of training required is minimal and only done once. This is because, we are trying to learn the human behavior around the 3D geometry in general, rather than trying to learn the human behavior for a specific scene. Once we understand this behavior, the human behavior for any new geometry can be estimated without training as long as the new geometry is available. To the

best of our knowledge, this is the first work that accounts for the 3D geometry for trajectory forecasting. Our method is not scene or geometry dependent and explicitly models the effect of the 3D geometry on humans to predict their likely trajectory.

3. APPROACH

Given the 3D geometry of the environment like the floors, walls, hallways, etc. along with the starting point and the destination of a trajectory, we propose a trajectory forecasting algorithm as a Markov chain model. Let $P = \{p_1, p_2, p_3...\}$ be set of all points on the floor like the centroids of triangles in a triangle mesh as shown in Figure 2. The motion from a starting point to a destination point is depicted as a trajectory T formed by transitions from one point to another. $T = \{S_1, S_2, S_3...\}$ where S_1, S_2, S_3 are the states at times $\{t_1, t_2, t_3...\}$, and $S_i \in P$. As in a Markov chain model, the decision of which point to transition next depends only on the current state of the subject and can be denoted by:

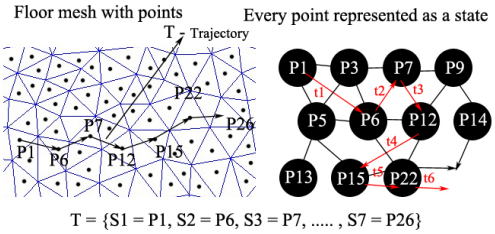


Figure 2: Trajectory modeled as a Markov chain model.

$$P(S_n|S_{n-1}, S_{n-1}|S_{n-2}...) = P(S_n|S_{n-1})P(S_{n-1}|S_{n-2})... \quad (1)$$

The Scenario in Figure 2 can be model as a Morkov Chain model:

$$\begin{aligned} & P(S_7 = P_{26}|S_6 = P_{22}, ..., S_1 = P_1) \\ & = P(S_7 = P_{26}|S_6 = P_{22})...P(S_2 = P_6|S_1 = P_1) \\ & = P(S_7 = P_{26}|S_6 = P_{22}) \end{aligned}$$

The problem is to create a transition matrix $P(S_n|S_{n-1})$ for our Markov model that we can use to sample points to form trajectories that travel from the start to destination and also conform with behavioral norms.

The steps involved in our trajectory forecasting model are (1) estimate the occupancy map of the new geometry (2) create the distance map based on the destination (3) combine the occupancy map and the distance map to create the energy function (4) define a transition matrix based on an energy maximization framework (5) and then sample points using the transition matrix to form a trajectory. Given the geometry, the occupancy map is estimated first. Later given the destination, the distance map, the energy function and consequently the transition matrix are estimated. The methodology is described in two steps (a) Section 3.1 describes estimating the occupancy map and (b) Section 3.2 describes trajectory forecasting. The flowchart in Figure 3 showcases the entire framework.

3.1 Occupancy Map Estimation

To estimate the occupancy map for our geometry, we begin by observing the occupancy map for a known geometry. Then we compute the geometric features for the points on the floor with respect to the surrounding geometry. Later we propose a model to establish a relationship between the occupancy of the point and its computed features using linear regression. We leverage this relationship to estimate the occupancy map for any new geometry.

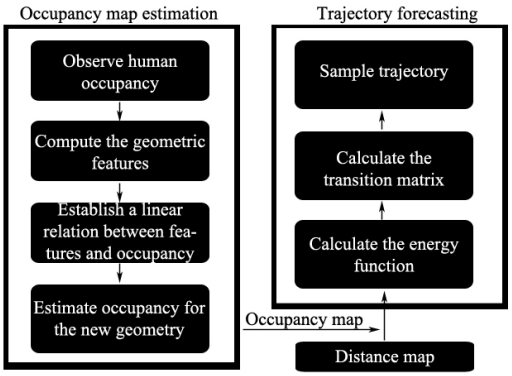


Figure 3: Flowchart illustrating the trajectory forecasting framework.

Observing the human occupancy map: In our geometry, we model the floor as a uniform triangle mesh. Let the centroids of the triangles on the floor mesh be represented by a set of points P. We take the video from a calibrated camera for a prolonged period of time, and perform human detection [10], that outputs the bounding box for each detected human. For every detected human, we re-project the bottom of the bounding box onto the floor in our 3D model. We find the triangle in which this re-projected point falls and increase its occupancy accordingly. The occupancy map observed in a hallway over a period of 5 days is shown in Figure 4.

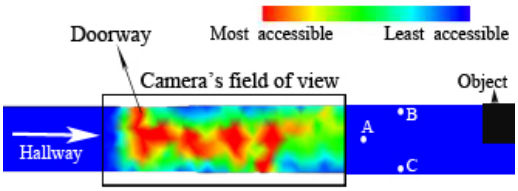


Figure 4: Observed occupancy map of a hallway in a building from a video observed over 5 days.

Geometric features: The features f_i of any point p_i on the floor in the 3D model are represented as a set of numbers $\{d_{i1}, d_{i2}, d_{i3}...\}$, which are its distances from the walls and objects surrounding the point p_i. The richest description is obtained by taking distances from the point on the floor to every other point in the geometry surrounding it. However, such amounts of data is redundant, computationally infeasible and will likely result in over training. We would like to use a feature set that gives us sufficient information to estimate the occupancy of that point. When traversing indoors, our immediate decision of movement is influenced by the objects in our path in the hallway and the surrounding walls. For example the way we navigate around tables and chairs when moving from one corner of a classroom to the opposite corner. So, to build our features we measure distances to

walls or objects in the hallway along vectors pointing at a certain inclination from the floor. In this paper, we use 30 - 60 degrees, considering this is sufficient to capture objects present in the hallway. Pointing vectors at regular interval spanning an entire circle with its tail fixed at the point p_i as shown in Figure 5. There are two issues concerning these features. First, if the closest wall or an object in a certain direction is very far away like in the case of an object at the far end of a hallway like the point A in Figure 4 with respect to the object, the local motion or occupancy decisions of a human subject is indifferent to an object at such great distance. Second, if we take two points on the floor that are close to the walls in a hallway but on either end of the hallway like the points B and C in Figure 4, these points in essence are the same and are likely to have the same occupancy, yet the features representing these points are different. If for example we want to take the distances surrounding the point in clockwise direction starting from the first direction being upwards, the features of B would start with a small number and increase before decreasing. However for C, the features would start with a much larger value and then increase before decreasing to a smaller value following the convention for computing distances. So the features would require some preprocessing as we would like to make the features scale and rotational invariant.

Scale invariance can be achieved by thresholding the distances to a hemisphere with its center at the location of the human subject's feet as shown in Figure 5. The radius r of this hemisphere is inferred from the theory of Proxemics [13]. This is a theory based on observation that defines how human beings unintentionally make use of physical space around them. Proxemics classifies the space close to a human subject into four broad regions, Intimate, Personal, Social and Public distance. For our purpose, we consider the interaction between human subjects in closed hallways to take place within the social distance, which is 7-12 ft. (80-140 in.). The radius of the hemisphere is defined by this distance ($f_i = \{d_{i1}, d_{i2}, d_{i3}...\}, d_{ij} = r \forall d_{ij} \geq r, 80 \leq r \leq 140$). To make the features rotationally invariant, we measure the features always starting with the smallest distance ($f_i = \{d_{i1}, d_{i2}, d_{i3}...\}, d_{i1} \leq d_{ij}, 2 \leq j \leq n$) and take measurements following a certain convention (either clockwise or anti-clockwise direction), in which case, the points B and C in Figure 4 will have similar features. The features are not arranged in ascending order, but are only measured starting from the smallest value keeping the order unchanged.

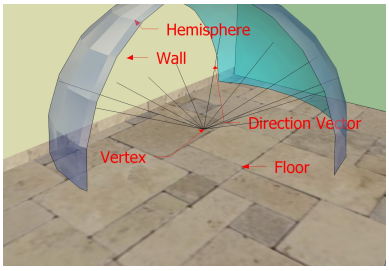


Figure 5: Geometric features.

Modeling relationship between occupancy map and geometric features: Now we know the geometry and know how humans occupy this specific geometry. Let $f_i = \{d_{i1}, d_{i2}, d_{i3}...\}$ be the features of the points p_i with occupancy o_i. Given the dataset $\{o_i, d_{i1}, d_{i2}, ...d_{in}\}$, we choose to model the relation between the dependent variable o_i and the independent variables, vectors of f_i, using a linear relationship. Let ϵ_i be the error term. Then we have

$$\begin{aligned} o_i &= \beta_1 d_{i1} + \beta_2 d_{i2} + ... + \beta_n d_{in} + \epsilon_i = F^T{}_i \beta + \epsilon_i \\ o &= F\beta + \epsilon \end{aligned} \tag{2}$$

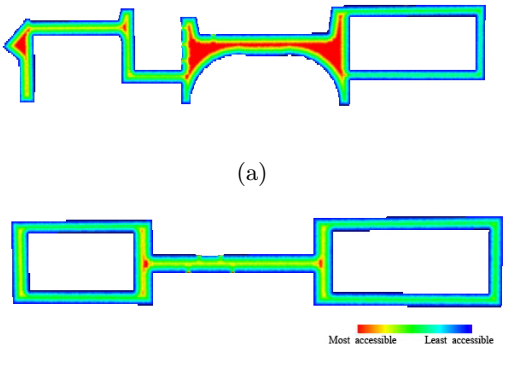
$$o = \begin{pmatrix} o_1 \\ o_2 \\ . \\ . \\ . \end{pmatrix}; \quad F = \begin{pmatrix} f_1 \\ f_2 \\ . \\ . \\ . \end{pmatrix} = \begin{pmatrix} d_{11} & ... & d_{1n} \\ d_{21} & ... & d_{2n} \\ . & ... & . \\ . & ... & . \\ . & ... & . \end{pmatrix};$$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \end{pmatrix}; \qquad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ . \\ . \\ . \end{pmatrix}$$

To estimate the values of β we minimize the sum of squares of the error term ϵ, which would give us $\beta = (F^T F)^{-1} F^T o$. To determine the occupancy of any point on the floor in any new geometry, all we need is to compute the geometric features of that point. We can use the estimated β to calculate the occupancy at that point. Figure 6 depicts the occupancy for two different geometries in a building. We can see how the occupancy of the points in the center of the hallway are higher than those along the edges. The rotational invariance of the features allow for the expected estimation of the occupancy even along curved hallways as seen in Figure 6 (a).

3.2 Trajectory Forecasting

Now that we have the occupancy map, given the destination we create a distance map and then combine the occupancy and distance maps to construct the energy function. Then finally we use a energy maximization framework to create the transition matrix for our trajectory forecasting.



(a)

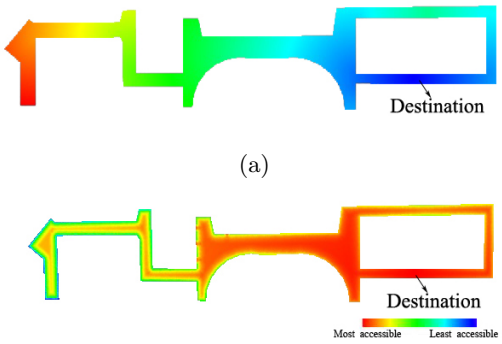
(b)

Figure 6: Estimated occupancy maps: Red being most accessible and blue being the least. (a) geometry A; (b) geometry B.

Destination map: Given a destination, we can create a distance map from every other point on the floor, to the destination. In general, hallways are complex polygons with areas that are inaccessible. Using Euclidean distance can potentially be erroneous. So we make use of Geodesic distance [23] instead. Euclidean distance between two points is not

altered by the presence of inaccessible areas, but geodesic distance is measured around the inaccessible areas along the hallway and gives a more accurate sense of distance for human navigation. The geodesic distances from all the points is calculated to the destination ahead of time. This needs to be done only once for the entire geometry. A rendering of the distance map for geometry A with a given destination is shown in Figure 7 (a).

The energy function: We use a combination of the distance map and the occupancy map to create our energy function. Let O be the occupancy map function and let D be the distance map function. If $p_i \in P$ is any point on the floor. Then the energy of that point is defined by the function $E = -D(p_i)/O(p_i)$. The energy function for geometry A is shown in Figure 7 (b). We can see how the energy is higher in the center of the hallway than along the edges, and the energy keeps increasing as we move towards the destination.



(a)

(b)

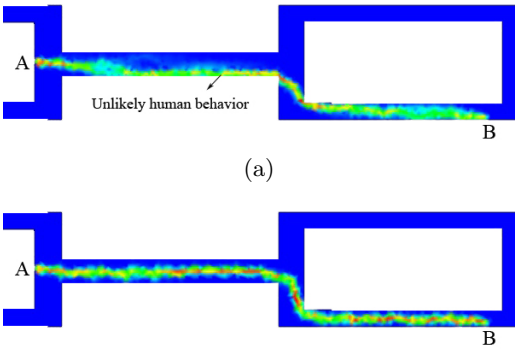
Figure 7: Distance map and energy function:(a) Distance map for geometry A with a given destination; (b) Energy function for geometry A with a given destination.

Trajectory sampling: We build our transition probability matrix by choosing states that maximize the energy with higher probability. For every state the subject is present in, the only possible states of transition are states representing its neighboring points. Let the current state be S_t, and this point has m neighbors $S_{t1}, S_{t2}, ...S_{tm}$. The probability of transition to these m neighboring states is proportional to the difference in energy. So $P(S_{tm}|S_t)$ is.

$$\propto \left\{ \begin{array}{cc} E(s_{tm}) - E(s_t) & \text{if} \quad D(s_{tm}) - D(s_t) \leq 0 \\ 0 & otherwise \end{array} \right\} \quad (3)$$

We are only choosing states that are closer to the destination (i.e. $D(s_{tm}) - D(s_t) \leq 0$), to ensure that the propagation does not get stuck in a local maximum. The neighboring states are sampled with a probability that is proportional to the difference in their energies.

Algorithm 1 summarizes our complete trajectory forecasting methodology. Figure 8 (a) shows the distribution of simulating the trajectory prediction algorithm 5,000 times without the use of the occupancy map, but only using distance minimization. Figure 8 (b) simulates with the help of the occupancy map in geometry B. We can see how the estimated occupancy map complements the geodesic distance minimization and forms a more desirable trajectory, that conforms to expected human behavior.



(a)

(b)

Figure 8: A is the starting location and B is the destination (a) Distribution created by simulating trajectory prediction without using occupancy map; (b) Distribution created by simulating trajectory prediction using occupancy map.

4. EXPERIMENTS

4.1 Implementation

This section explains the implementation of our methodology. First, we present how we build our 3D models. Then we describe a method to embed virtual cameras in the 3D model that represent the cameras in real world. Finally we will see how a human subject detected in the image of a real world camera can be projected onto a point in our 3D model.

Modeling 3D geometry: We model the 3D geometry of the environment like floors, walls, hallways, etc. using Google Sketchup, a 3D modeling tool. Figure 9 depicts the 3D model of a building constructed using existing floor plans to obtain the measurements and dimensions. We then export the 3D model using a *common digital asset exchange format* [2] called COLLADA file format which we later use for rendering and understanding the 3D environment. COLLADA Document Object Model (DOM) library is used to load and save this 3D model into an application, and then we use OpenGL to interact with this 3D data in the application.

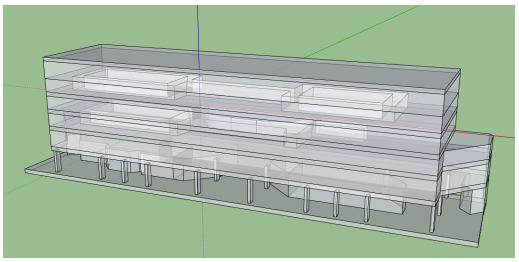


Figure 9: Model of a building using Google Sketchup.

Embedding virtual cameras and calibration: An initial step is to create virtual cameras in our 3D model

which represent the cameras in real world. In order to do this we first determine the internal camera parameters of the existing real world camera by using a general calibration approach using a checkerboard. Once the camera's internal parameters are obtained, we can use OpenGL to create virtual cameras in our model which render perspective projections of the 3D model that are conceptually equivalent to the real world cameras. Now in order to determine the location and orientation of the camera in our 3D model, we take an image from the real world camera and try to manually register it with the corresponding camera's perspective projection in our graphics model, by manually changing the parameters in the transformation matrix using OpenGL. When the images register as shown in Figure 10, we extract the transformation matrix of the camera which gives us the approximate location and orientation of the camera in the 3D model [30].

Delaunay triangulation of the floor mesh: We choose to represent the floor using a triangular mesh though other representation are possible. For our purpose we would like a rich description of the triangular mesh representing the floor where human subjects walk. Triangles in the mesh should have adequate height and base with respect to the normal human motion characteristics. Assuming that the humans walk at an average pace of 3 ft/sec and the camera in use has a frame rate of 30 fps, if we plan to take a sample every 10 seconds or for every feet the human moves, it would be convenient if the triangles have a height and base that are at least 1 ft. long. We use Delaunay triangulation to obtain a mesh that is uniformly spaced as shown in Figure 10. An implementation of the Delaunay triangulation is available in the Computational Geometric Algorithms Library (CGAL) [1].

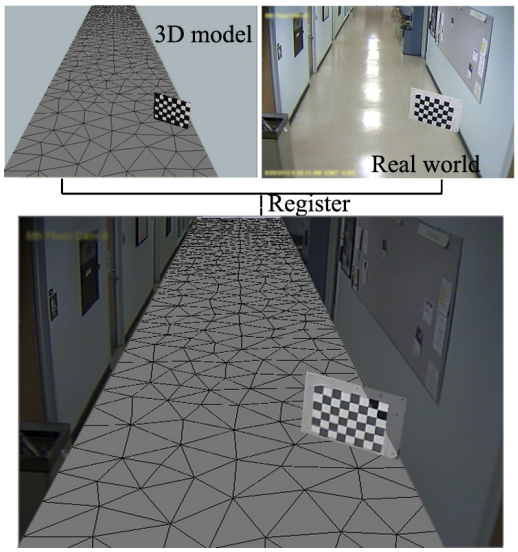


Figure 10: Manual registration of an image from a camera with the perspective rendering of the 3D model to extract the transformation matrix. The floor is represented by a uniform triangle mesh obtained by Delauney triangulation.

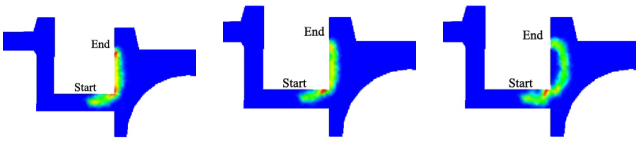
4.2 Experiments

First the training data for estimating the occupancy map was obtained from the hallway shown in Figure 10. Five days worth of video was used to create the training data and the observed occupancy map is shown in Figure 4. This data was used to estimate the β values through linear regression as describe in Section 3.1. We take real world trajectories from three different scenarios, one from geometry A and two from geometry B as shown in Figure 11. A sample size of 14 videos from scenario 1, 12 videos from scenario 2 and 11 videos from scenario 3 (37 different human subjects) were used to evaluate our trajectory forecasting model. None of these scenario's geometries were used in estimating the β values during linear regression in Section 3.1. The human test subjects were given information regarding the destination only. An object was placed at the destination and all they were instructed was to walk to the destination, pick up the object and come back. The test subjects were not made aware that the purpose of the experiment was to observe their trajectories. To form a trajectory, we take a video of the test subject, run it through the human detection algorithm [10], and the detection are then projected back into the 3D Model. The consecutive detections are then connected to form the subject's trajectory. We evaluate our model using two different metrics. First, we calculate the modified Hausdorff distance of the real world trajectory to predicted trajectories. Second, we determine the negative log likelihood of the real world trajectory from the distribution created by the trajectories from the proposed model. In order to quantify the performance of our model, we compare the distribution of trajectories predicted by our model with a state-of-the-art approach (activity forecasting [19]) and a baseline algorithm. The work of Kitani *et al.* [19] is more extensive as the destination was also forecasted before the trajectory prediction was performed unlike the proposed method where the destination is assumed to be known. From a trajectory forecasting stand point, both the models require the destination to be known before trajectory forecasting and are defined manually during the experiments. In the baseline algorithm, all the points are assumed to have equal occupancy hence allowing us to evaluate the impact of our model of human behavior. In activity forecasting [19], for each scenario the images are given semantic labels manually. To evaluate this approach on our dataset, the walls are labeled as building and the floor as sidewalk. The weights for the features/labels are learned from a different geometry and are then transfered and used for forecasting the trajectory distribution in the new geometry. Figure 12 compares the distribution of trajectories generated for scenario 1 using baseline, activity forecasting and proposed method.

(a) (b) (c)

Figure 11: Experimental scenarios with a sample trajectory, red - actual trajectory, green - predicted trajectory; (a) scenario 1 (geometry A); (b) scenario 2 (geometry B); (c) scenario 3 (geometry B);.

Modified Hausdorff distance: Let $T_o = \{S_{o1}, S_{o2}, S_{o3}..\}$ be the observed trajectory and $T_q = \{S_{q1}, S_{q2}, S_{q3}..\}$ predicted trajectory, where $S_i \in P$ are points on the floor. The Hausdorff distance $D_H(T_o, T_p)$ between the two trajectories



(a) (b) (c)

Figure 12: Trajectory distribution around the corner for scenario 1 in geometry A; (a) Baseline; (b) Activity Forecasting; (c) Proposed Method;.

Scenario	Baseline	Activity Forecasting [19]	Proposed
1	23.802	22.993	9.259
2	15.677	26.713	8.473
3	18.658	35.116	9.449

Table 1: Hausdorff distance of real world trajectories compared with simulated trajectories. The distances are measured in inches.

is defined as $\max\{D(T_o, T_q), D(T_q, T_o)\}$, where

$$D(T_o, T_q) = \frac{1}{N_o} \sum_{a \in T_o} d(a, T_q)$$
$$d(a, T_q) = \min_{b \in T_q} d(a, b) = \min_{b \in T_q} |a - b| \quad (4)$$

$d(a, b)$ is the Euclidean distance between the points a and b, and N_o is the number of points in the trajectory T_o. This essentially is a metric for quantifying the difference between the trajectories T_o and T_q. Each trajectory is compared with 500 simulations of the predicted trajectory from our model to calculate the average modified Hausdorff distance. The average modified Hausdorff distance over the real world trajectories for the three geometries are shown in the Table 1. We can see in all the three geometries, the error in predictions is decreased by using the distribution from our model.

Negative log likelihood: To directly compare our approach to the method in [19], given a starting point and a destination point, we simulate our model multiple times and try to build a transition probability matrix. If $T_q = \{S_{q1}, S_{q2}, S_{q3}..\}$ be the the simulated trajectory, using multiple simulations, we build a NXN transition matrix, where N is the total number of states or points on the floor. Let $T_o = \{S_{o1}, S_{o2}, S_{o3}...\}$ be the observed trajectory. We estimate the probability of sampling the observed trajectory from the distribution created by the predicted trajectories as described in Activity Forecasting [19]. We use 2500 predictions to create the transition probability matrix for the prediction model. For an observed trajectory T_o, the error is estimated as

$$L(T_o) \propto -\ln \prod_i P(S_{(i)o} | S_{(i-1)o}), \quad (5)$$

where $S_{(i-1)o}, S_{(i)o} \in T_o$ and $P(S_{(i)o}|S_{(i-1)o})$ is the probability of transition from state $S_{(i-1)o}$ (current triangle) to $S_{(i)o}$ (next triangle). This measure is normalized by dividing it with the length of the trajectory. The results in the Table 2 show the average negative log likelihood for each geometry. The results demonstrate how the energy function decreases the error in forecasting the trajectory.

Scenario	Baseline	Activity Forecasting [19]	Proposed
1	2.209	2.529	1.927
2	2.218	1.907	1.550
3	2.259	1.943	1.739

Table 2: Negative log likelihood of real world trajectories compared to simulated trajectories.

5. CONCLUSION

We have modeled a set of geometric features that describe a point on the floor with respect to the structure of the surrounding geometry. We have proposed a method to estimate the occupancy map using the geometric features for any new geometry without the need for training data. We have developed an algorithm to forecast human motion trajectories using this estimated human behavior model. It is observed that incorporating the estimated occupancy map in the trajectory prediction can improve the accuracy of prediction significantly. The decrease in the log likelihood and the modified Hausdorff distance with the incorporation of the energy function supports the accuracy of this method.

Acknowledgment

This work was supported in part by the US Department of Justice 2009-MU-MU-K004. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of our sponsors.

6. REFERENCES

[1] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[2] R. Arnaud and M. C. Barnes. *Collada: Sailing the Gulf of 3d Digital Content Creation.* AK Peters Ltd, 2006.

[3] G. Aschwanden, J. Halatsch, and G. Schmitt. Crowd simulation for urban planning. *Proceedings of eCAADe 2008*, 2008.

[4] S. Bhattacharya, V. Kumar, and M. Likhachev. Search-based path planning with homotopy class constraints. In *In Proc. National Conference on Artificial Intelligence.*

[5] P. Biber, H. Andreasson, T. Duckett, and A. Schilling. 3d modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004.

[6] P. Biber, S. Fleck, and T. Duckett. 3d modeling of indoor environments for a robotic security guard. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, 2005.

[7] P. Biber, S. Fleck, W. Strasser, S. Fleck, and W. Strasser. A mobile platform for acquisition of 3d-models of large environments : The wagele. In *In 3D-ARCH 2005: 3D Virtual Reconstruction and Visualization of Complex Architectures*, 2005.

[8] X. Chen, K. Huang, and T. Tan. Learning the three factors of a non-overlapping multi-camera network

topology. In C.-L. Liu, C. Zhang, and L. Wang, editors, *Pattern Recognition*, Communications in Computer and Information Science. Springer Berlin Heidelberg, 2012.

[9] P. Y. Choi and M. Hebert. Learning and predicting moving object trajectory: a piecewise trajectory segment approach. Technical report, Robotics Institute, 2006.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *ICCV*, June 2005.

[11] T. J. Ellis, D. Makris, and J. K. Black. Learning a multi-camera topology. In *in Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[12] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *ICCV*. IEEE, 2011.

[13] E. T. Hall. *The Hidden Dimension*. Anchor Books. ISBN 0-385-08476-5, 1966.

[14] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science*, 2005.

[15] W. Hu, D. Xie, T. Tan, and S. Maybank. Learning activity patterns using fuzzy self-organizing neural network. *Sys., Man, and Cybernetics, IEEE Trans. on*, 2004.

[16] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera spaceâĂŞtime and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 2008.

[17] E. Jeon and S. Jo. Real-time building of a 3d model of an indoor environment with a mobile robot. In *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, 2011.

[18] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *ICPR 2004*, 2004.

[19] K. Kitani, B. D. Ziebart, J. A. D. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, Oct. 2012.

[20] C. Loy, T. Xiang, and S. Gong. Time-delayed correlation analysis for multi-camera activity understanding. *International Journal of Computer Vision*, 2010.

[21] C. Loy, T. Xiang, and S. Gong. Incremental activity modeling in multiple disjoint cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2012.

[22] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.

[23] D. Martínez, L. Velho, and P. C. Carvalho. Computing geodesics on triangular meshes. *Comp. Graph.*, Oct. 2005.

[24] R. Mazzon, S. F. Tahir, and A. Cavallaro. Person re-identification in crowd. *Pattern Recogn. Lett.*, 2012.

[25] B. Morris and M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *CSVT*, 2008.

[26] J. Nascimento, M. Figueiredo, and J. Marques. On-line classification of human activities. In *Pattern Rec. and Image Analysis*, volume 4478 of *Lec. Notes in Comp Sci.* 2007.

[27] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.

[28] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 2008.

[29] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *TPAMI*, 2009.

[30] P. Shirley and M. Ashikhmin. *Fundamentals of Computer Graphics, Second Edition*. Ak Peters Series. Peters, 2005.

[31] K. Tieu, G. Dalley, and W. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005.

[32] D. Vasquez and T. Fraichard. Motion prediction for moving objects: a statistical approach. In *ICRA '04*, 2004.

[33] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier. Intentional motion on-line learning and prediction. *Mach. Vision Appl.*, Sept. 2008.

[34] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. *Proc. Computer Vision and Pattern Recognition, 2014.*, March 2014.

[35] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. D. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. IROS 2009*.