Activity Forecasting

Kris M. Kitani, Brian D. Ziebart, J. Andrew Bagnell, and Martial Hebert

Carnegie Mellon University, Pittsburgh, PA 15213 USA
{kkitani,bziebart}@cs.cmu.edu, {dbagnell,hebert}@ri.cmu.edu

Abstract. We address the task of inferring the future actions of people from noisy visual input. We denote this task *activity forecasting*. To achieve accurate activity forecasting, our approach models the effect of the physical environment on the choice of human actions. This is accomplished by the use of state-of-the-art semantic scene understanding combined with ideas from optimal control theory. Our unified model also integrates several other key elements of activity analysis, namely, destination forecasting, sequence smoothing and transfer learning. As proof-of-concept, we focus on the domain of trajectory-based activity analysis from visual input. Experimental results demonstrate that our model accurately predicts distributions over future actions of individuals. We show how the same techniques can improve the results of tracking algorithms by leveraging information about likely goals and trajectories.

Keywords: activity forecasting, inverse optimal control

1 Introduction

We propose to expand the current scope of vision-based activity analysis by exploring models of human activity that reason about the *future*. Although reasoning about future actions often requires a large amount of contextual prior knowledge, let us consider the information that can be gleaned from *physical scene features* and prior knowledge of *goals*. For example, in observing pedestrians navigating through an urban environment, we can predict with high confidence that a person will *prefer* to walk on sidewalks more than streets, and will

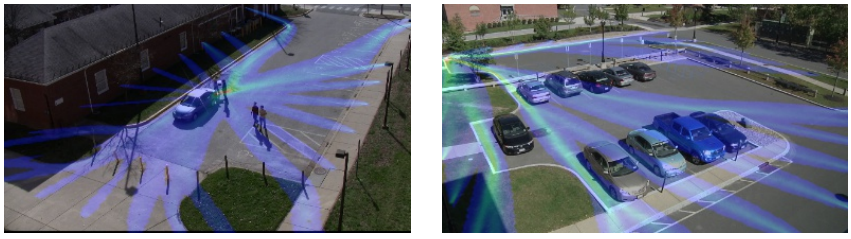


Fig. 1. Given a single pedestrian detection, our proposed approach forecasts plausible paths and destinations from noisy vision-input

most certainly avoid walking into obstacles like cars and walls. Understanding the concept of human *preference* with respect to physical scene features enables us to perform higher levels of reasoning about future human actions. Likewise, our knowledge of a *goal* also gives us information about what a person might do. For example, if an individual desires to approach his car parked across the street, we know that he will prefer to walk straight to the car as long as the street is walkable and safe. To integrate these two aspects of prior knowledge into modeling human activity, we leverage recent progress in two key areas of research: (1) semantic scene labeling and (2) inverse optimal control.

Semantic scene labeling. Recent semantic scene labeling approaches now provide a robust and reliable way of recognizing physical scene features such as pavement, grass, tree, building and car [1], [2]. We will show how the robust detection of such features plays a critical role in advancing the representational power of human activity models.

Inverse optimal control. Work in optimal control theory has shown that human behavior can be modeled successfully as a sequential decision-making process [3]. The problem of recovering a set of agent preferences (the reward or cost function) consistent with demonstrated activities, can be solved via Inverse Optimal Control (IOC) – also called Inverse Reinforcement Learning (IRL) [4] or inverse planning [5]. What is especially intriguing about the IOC framework is that it incorporates concepts, such as *immediate rewards* (what do I gain by taking this action?), *expected future rewards* (what will be the consequence of my actions in the future?) and *goals* (what do I intend to accomplish?), which have close analogies to the formation of human activity. We will show how the IOC framework expands the horizon of vision-based human activity analysis by integrating the impact of the environment and goals on future actions.

In this work, we extend the work of Ziebart *et al.* [6] by incorporating vision-based physical scene features and noisy tracker observations, to forecast activities and destinations. This work is different from traditional IOC problems because we do not assume that the state of the actor is fully observable (e.g., video games [7] and locations in road networks [6]). Our work is also different from Partially Observable Markov Decision Process (POMDP) models because we assume that the *observer* has noisy observations of an actor, where the actor is fully aware of his own state. In a POMDP, the actor is uncertain about his own state and the observer is not modeled. To the best of our knowledge, this is the first work to incorporate the uncertainty of vision-based observations within a robust IOC framework in the context of *activity forecasting*. To this end, we propose a Hidden variable Markov Decision Process (hMDP) model which incorporates uncertainty (e.g., probabilistic physical scene features) and noisy observations (e.g., imperfect tracker) into the activity model. We summarize our contributions as follows: (1) we introduce the concept of inverse optimal control to the field of vision-based activity analysis, (2) we propose the hMDP model and a hidden variable inverse optimal control (HIOC) inference procedure to deal with uncertainty in observations and (3) we demonstrate the performance of forecasting, smoothing,

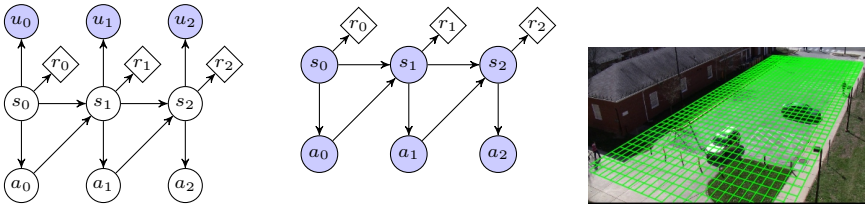
Homotopy classes **Our approach**

Fig. 2. Qualitative comparison to homotopy classes. Trajectories generated by distinct homotopy classes and trajectories generated by physical attributes of the scene. Physical attributes are able to encode agent preferences like using the sidewalk

destination forecasting and knowledge transfer operations in a single framework on real image data.

As a proof-of-concept, we focus on trajectory-based human activity analysis [8]. We take a departure from traditional motion-based approaches [9], [10] and explore the interplay between features of the environment and pedestrian trajectories. Previous work [11], [12], has shown that modeling the impact of the social environment, like actions of nearby pedestrians, can improve priors over pedestrian trajectories. Our work is complementary in that, our learned model explains the effect of the *static environment*, instead of the dynamic environment like moving people, on future actions. Other work uses trajectories to infer the functional features of the environment such as road, sidewalk and entrance [13]. Our work addresses the inverse task of inferring trajectories from physical scene features. Work exploring the impact of destinations, such as entrances and exits, of the environment on trajectories has shown that knowledge of goals yields better recognition of human activity [14], [15]. Gong *et al.* [16] used potential goals and motion planning from homotopy classes to provide a prior for tracking under occlusion. Our work expands the expressiveness of homotopy classes in two significant ways, by generating a distribution over all trajectories including homotopy classes, and incorporating observations about *physical scene features* to make better inference about paths. Figure 2 depicts the qualitative difference between shortest distance paths of 'hard' homotopy classes and 'soft' probability distributions generated by our proposed approach. Notice how the distribution over potential trajectories captures subtle agent preferences such as walking on the sidewalk versus the parking lot, and keeping a safe distance from cars.

There is also an area of emerging research termed *early recognition*, where the task is to classify an incoming temporal sequence as early as possible while maintaining a level of detection accuracy [17], [18], [19]. Our task of *activity forecasting* differs in that we are recovering a *distribution over a sequence of future actions* as opposed to classifying a partial observation sequence as a discrete activity category. In fact, our approach can forecast possible trajectories before any pedestrian observations are available.



(a) Proposed hMDP (b) MDP (c) state representation

Fig. 3. Underlying graphical model and state representation for IOC. (a) Proposed hMDP: agent knows own state s, action a and reward (or cost) r but only noisy measurements of the state u are observed, (b) MDP: agent state and actions are fully observed and (c) ground plane is discretized into cells which represent states

2 Preliminaries

Markov Decision Processes and Optimal Control. The Markov decision process (MDP) [20] is used to express the dynamics of a decision-making process (Figure 3b). The MDP is defined by an initial state distribution $p(s_0)$, a transition model $p(s'|s,a)$ (shorthand $p^{s'}_{s,a}$) and a cost function $r(s)$. Given these parameters, we can solve the *optimal control* problem by learning the optimal policy $\pi(a|s)$, which encodes the distribution of action a to take when in state s. To be concrete, Figure 3c depicts the state and action space defined in this work. The state s represents a physical location in world coordinates $s = [x, y]$ and the action a is the velocity $a = [v_x, v_y]$ of the actor. The policy $\pi(a|s)$ maps states to actions, describing which direction to move (action) when an actor is located at some position (state). The policy can be deterministic or stochastic.

Inverse Optimal Control. In the inverse optimal control problem, the cost function is not given and must be discovered from demonstrated examples. Various approaches using structured maximum margin prediction [21], feature matching [4] and maximum entropy IRL [3] have been proposed for recovering the cost function. We build on the maximum entropy IOC approach in [6] and extend the model to deal with noisy observations. We make an important assumption about the form of the cost function $r(s)$, which enables us to translate from observed physical scene features to a single cost value. The cost function:

$$r(s;\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{f}(s), \tag{1}$$

is assumed to be a weighted combination of feature responses $\boldsymbol{f}(s) = [f_1(s) \cdots f_K(s)]^\top$, where each $f_k(s)$ is the response of a physical scene feature, such as the soft output of a grass classifier, and $\boldsymbol{\theta}$ is a vector of weights. By learning the parameters of the cost function, we are learning how much a physical scene feature affects a person's actions. For example, a feature such as car and building, will have large weights because they are high cost and should be avoided. This explicit modeling of the effect of physical scene features on actions via the cost function sets this approach apart from traditional motion-based models of pedestrian dynamics.

3 Hidden Variable Inverse Optimal Control (HIOC)

In a vision-based system, we do not have access to the true state, such as the location of the actor, or the true action, such as the velocity of the actor. Instead, we only have access to the output of a noisy tracking algorithm. Therefore, we deal with observation uncertainty via a hidden state variable (Figure 3a). Using this hidden model, HIOC determines the *reliability* of observed states, in our case tracker detections, by adjusting its associated cost weight. For example, if the tracker output has low precision, the corresponding weight parameter will be decreased during training to minimize the reliance on the tracker output.

In the maximum entropy framework, the distribution over a state sequence $\boldsymbol{s}$ is defined as:

$$p(\boldsymbol{s};\boldsymbol{\theta}) = \frac{\prod_t e^{r(s_t)}}{Z(\boldsymbol{\theta})} = \frac{e^{\sum_t \boldsymbol{\theta}^\top \boldsymbol{f}(s_t)}}{Z(\boldsymbol{\theta})}, \tag{2}$$

where $\boldsymbol{\theta}$ are the parameters of the cost function, $\boldsymbol{f}(s_t)$ is the vector of feature responses at state s_t and $Z(\theta)$ is the normalization function. In other words, the probability of generating a trajectory $\boldsymbol{s}$ is defined to be proportional to the exponentiated sum of features encountered over the trajectory.

In our hMDP model (Figure 3a), we add state observations $\boldsymbol{u}$ to represent the uncertainty of being in a state. This implies a joint distribution over states and observations as:

$$p(\boldsymbol{s},\boldsymbol{u};\boldsymbol{\theta}) = \frac{\prod_t p(u_t|s_t) e^{\boldsymbol{\theta}^\top \boldsymbol{f}(s_t)}}{Z(\boldsymbol{\theta})} = \frac{e^{\sum_t \left\{\boldsymbol{\theta}^\top \boldsymbol{f}(s_t) + \theta_o \log p(u_t|s_t)\right\}}}{Z(\boldsymbol{\theta})}, \tag{3}$$

where the observation model $p(u_t|s_t)$ is a Gaussian distribution. Notice that by pushing the observation model into the exponent as $\log p(u_t|s_t)$ it can also be interpreted as an auxiliary 'observation feature' with an implicit weight of one, $\theta_o = 1$. However, we increase the expressiveness of the model by allowing the weight parameter θ_o of observations to be adjusted at training.

3.1 Training and inference

In the training step, we recover the optimal cost function parameters $\boldsymbol{\theta}$ and consequentially an optimal policy $\pi(a|s)$, by maximizing the entropy of the conditional distribution or equivalently the likelihood maximization of the observations under the maximum entropy distribution,

$$p(\boldsymbol{s}|\boldsymbol{u};\boldsymbol{\theta}) = \frac{e^{\left\{\sum_t \boldsymbol{\theta}^\top \boldsymbol{f}'(s_t)\right\}}}{Z(\boldsymbol{\theta})}, \tag{4}$$

where the feature vector $\boldsymbol{f}'(s_t)$ now includes the tracker observation features.

To maximize the entropy of (4), we use exponentiated gradient descent to iteratively minimize the gradient of the log-likelihood $\mathcal{L} \triangleq \log p(\boldsymbol{s}|\boldsymbol{u};\boldsymbol{\theta})$. The gradient can be shown to be the difference between the *empirical* mean feature count

Algorithm 1 – Backwards pass

$V(s) \leftarrow -\infty$
for $n = N, \ldots, 2, 1$ **do**
  $V^{(n)}(s_{goal}) \leftarrow 0$
  $Q^{(n)}(s,a) = r(s;\boldsymbol{\theta}) + E_{P^{s'}_{s,a}}[V^{(n)}(s')]$
  $V^{(n-1)}(s) = \text{soft}\max_a Q^{(n)}(s,a)$
end for
$\pi_\theta(a|s) = e^{Q(s,a)-V(s)}$

Algorithm 2 – Forward pass

$D(s_{initial}) \leftarrow 1$
for $n = 1, 2, \ldots, N$ **do**
  $D^{(n)}(s_{goal}) \leftarrow 0$
  $D^{(n+1)}(s) = \sum_{s',a} P^s_{s',a} \pi_\theta(a|s') D^{(n)}(s')$
end for
$D(s) = \sum_n D^{(n)}(s)$
$\hat{\mathbf{f}}_\theta = \sum_s \boldsymbol{f}(s) D(s)$

$\bar{\mathbf{f}} = \frac{1}{M}\sum_m^M \boldsymbol{f}(\boldsymbol{s}_m)$, the average features accumulated over M demonstrated trajectories, and the *expected* mean feature count $\hat{\mathbf{f}}_\theta$, the average features accumulated by trajectories generated by the parameters, $\nabla\mathcal{L}_\theta = \bar{\mathbf{f}} - \hat{\mathbf{f}}_\theta$. We update $\boldsymbol{\theta}$ according to the exponentiated gradient, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} e^{\lambda \nabla \mathcal{L}_\theta}$, where λ is the step size and the gradient is computed using a two-step algorithm described next. At test time, the learned weights are held constant and the same two-step algorithm is used to compute the forecasted distribution over future actions, the smoothing distribution or the destination posterior.

Backward pass. In the first step (Algorithm 1), we use the current weight parameters $\boldsymbol{\theta}$ and compute the expected cost of a path ending in s_g and starting in $s_i \neq s_g$. Essentially, we are computing the expected cost to the goal from every possible starting location. The algorithm revolves around the repeated computation of the *state log partition function* $V(s)$ and the *state-action log partition function* $Q(s,a)$ defined in Algorithm 1. Intuitively, $V(s)$ is a soft estimate of the expected cost of reaching the goal from state s and $Q(s,a)$ is the soft expected cost of reaching the goal after taking action a from the current state s. Upon convergence, the maximum entropy policy is $\pi_\theta(a|s) = e^{Q(s,a)-V(s)}$.

Forward pass. In the second step (Algorithm 2), we propagate an initial distribution $p(s_0)$ according to the learned policy $\pi_\theta(a|s)$. Let $D^{(n)}(s)$ be defined as the *expected state visitation count* which is a quantity that expresses the probability of being in a certain state s at time step n. Initially, when n is small, $D^{(n)}(s)$ is a distribution that sums to one. However, as the probability mass is absorbed by the goal state, the sum of the state visitation counts quickly converges to zero. By computing the total number of times each state was visited $D(s) = \sum_n D^{(n)}(s)$, we are computing the unnormalized marginal state visitation distribution. We can compute the *expected* mean feature count as a weighted sum of feature counts $\hat{\mathbf{f}}_\theta = \sum_s \boldsymbol{f}(s) D(s)$.

3.2 Destination forecasting from noisy observations

In novel scenes, the destination of an actor is unknown and must be inferred. For each activity, a prior on potential destinations $p(s_g)$, may be generated (e.g., points along the perimeter of a car for the activity 'approach car') and, in principle, a brute force application of Bayes' rule enables computing the posterior over both destinations and intermediate states. A naive application, however, is

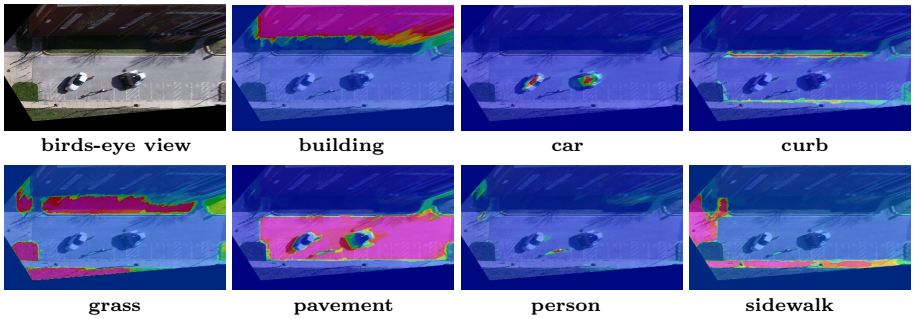


Fig. 4. Classifier feature response maps. Top left is the original image.

quite expensive as we may wish to consider a large number of possible goals – potentially every state.

Fortunately, the structure of the proposed maximum entropy model enables efficient inference. Following Ziebart *et al.* [6], we approximate the posterior over goals using a ratio of partition functions, one with and one without observations:

$$p(s_g|s_0, u_{1:t}) \propto p(u_{1:t}|s_0, s_g) \cdot p(s_g) \tag{5}$$

$$\propto e^{V_{u_{1:t}}(s_g) - V(s_g)} \cdot p(s_g), \tag{6}$$

where $V_{u_{1:t}}(s_g)$ is the state log partition of s_g given the initial state is s_0 and the observations $u_{1:t}$ and $V(s_g)$ is the state log partition of s_g without any observations. The ratio of log partition functions measure the 'progress' made toward a goal by adding observations. In deterministic MDPs, where the action decisions may be randomized but the state transitions follow deterministically from a state-action pair, we can invert the role of goal and start locations for an agent. Doing so enables computing the partition functions required in time *independent* of the number of goals. Using this inversion property, the state partition values for each goal can be computed efficiently by inverting the destination and start states and running Algorithm 1.

4 Experiments

We evaluate the four tasks of activity analysis, namely, (1) forecasting, (2) smoothing, (3) destination prediction and (4) knowledge transfer, using our proposed unified framework. For our evaluation we use videos from the VIRAT ground dataset [22]. Our dataset consists 92 videos from two scenes, shown in Figure 1. Scene A consists of 56 videos and scene B consists of 36 videos. Each scene dataset consists of three activities categories: *approach car*, *depart car* and *walk through*. In all experiments, 80% of the data was used for training and the remaining 20% used for testing using 3-fold cross validation.

The physical attributes were extracted using the scene segmentation labeling algorithm proposed by Munoz *et al.* [1]. In total 9 semantics labels were used,

including grass, pavement, sidewalk, curb, person, building, fence, gravel, and car. For each semantic label, four features were generated, including the raw probability and three types of 'distance-to-object' features. The distance feature is computed by thresholding the probability maps and computing the exponentiated distance function (with different variance). A visualization of the probability maps used as features is shown in Figure 4. For the smoothing task, the pedestrian tracker output is blurred with three different Gaussian filters which contribute three additional features. By adding a constant feature to model travel time, the total number of features used is 40.

Our state space is the 3D floor plane and as such, 2D image features, observations and potential goals are projected to the floor plane (camera parameters are assumed to be known) for all computations. For the activities *depart car* and *walk through* potential goals are set densely around the outer perimeter of the floor plane projection. For the activity *approach car*, connected components analysis is used to extract polygonal shape contours of detected cars, whose vertices are used to define a set of potential goals.

4.1 Metrics and baselines

In each of the experiments, we have one demonstrated path, a sequence of states s_t and actions a_t, generated by a pedestrian for a specific configuration of a scene. We compare the demonstrated path with the probabilistic distribution over paths generated by our algorithm using two different metrics: first is probabilistic and evaluates the likelihood of the demonstrated path under the predicted distribution, the second performs a more deterministic evaluation by estimating the physical distances between a demonstrated path and paths sampled from our distribution. We use the negative log-loss (NLL) of a trajectories, as in [6] as our probabilistic comparison metric. The negative log-loss:

$$\mathrm{NLL}(\boldsymbol{s}) = E_{\pi(a|s)}\left[-\log \prod_t \pi(a_t|s_t) \right], \tag{7}$$

is the expectation of the log-likelihood of a trajectory $\boldsymbol{s}$ under a policy $\pi(a|s)$. In our example, this metric measures the probability of drawing the demonstrated trajectory from the learned distribution over all possible trajectories. We also compute the modified Hausdorff distance (MHD) as a physical measure of the distance between two trajectories. The MHD allows for local time warping by finding the best local point correspondence over a small temporal window (± 15 steps in our experiments). When the temporal window is zero, the MHD is exactly the Euclidean distance. We compute the mean MHD, by taking the average MHD between the demonstrated trajectory and 5000 trajectories randomly sampled from our distribution. The units of the MHD are in pixels in the 3D floor plane, not the 2D image plane. We always divide our metrics by the trajectory length so that we can compare metrics across different models and trajectories of different lengths.

We compare against a maximum entropy Markov model (MEMM) that estimates the policy based on environmental attribute features and tracker observation features. The policy is computed by:

$$\pi(a|s) \propto \exp\{\boldsymbol{w}_a^\top \boldsymbol{F}(s)\}. \tag{8}$$

where the weight vector $\boldsymbol{w}_a$ is estimated using linear regression and $\boldsymbol{F}(s)$ is a vector of features for all neighboring states of s. This model only takes into the account the features of the potential next states when choosing an action and has no concept of the future beyond a one-step prediction model.

We also compare against a location-based Markov motion model, which learns a policy from observed statistics of states and actions in the training set:

$$\pi(a|s) \propto c(a, s) + \alpha, \tag{9}$$

where $c(a, s)$ is the number of times the action a was observed in state s and α is a pseudo-count used to smooth the distribution via Laplace smoothing.

4.2 Forecasting evaluation

Evaluating the true accuracy of a *forecasting distribution* over all future trajectories is difficult because we do not have access to such 'ground truth' from the future. As a proxy, we measure how well a learned policy is able to describe a single annotated test trajectory. We begin experiments in a constrained setting, were we fix the start and goal states to evaluate forecasting performance in isolation. Unconstrained experiments are performed in section 4.4. We compare our proposed model against the MEMM and the Markov motion model. Figure 5a and Table 1a show how our proposed model outperforms the baseline models. Note that tracker observations are not used in this experiment since we are only evaluating the performance of *forecasting* and not *smoothing*.

Qualitative results of activity forecasting are depicted in Figure 6. Our proposed model is able to leverage the physical scene features and generate a distribution that preserves actor preferences learned during training. Since many pedestrians used the sidewalk in the training examples, our model has learned that sidewalk areas have greater rewards or lower cost than paved parking lot areas. Notice that although it would be faster and shorter to walk diagonally across the parking lot, in terms of actor preferences it is more optimal to use the sidewalk. Without the use of informative physical scene features, we would need to learn motion dynamics with a Markov motion model from a large amount of demonstrated trajectories. Unfortunately, the Markov motion model degenerates to a random walk when there are not enough training trajectories for this particular configuration of the scene.

4.3 Smoothing evaluation

In our smoothing evaluation, we measure how the computed smoothing distribution accounts for noisy observations and generates an improved distribution over

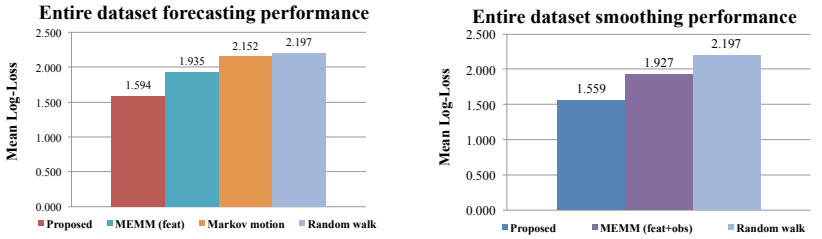


Fig. 5. Mean NLL of forecasting and smoothing performance

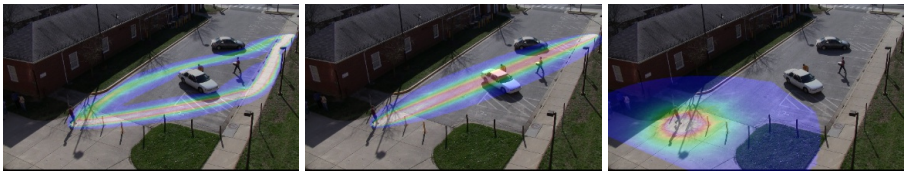


Fig. 6. Comparing forecasting distributions. The travel time only MDP ignores physical attributes of the scene. The Markov motion model degenerates to a random walk when train data is limited

trajectories. We run our experiments with a state-of-the-art super-pixel tracker (SPT) [23] and an in-house template-based tracker to show how the smoothing distribution improves the quality of estimated pedestrian trajectories. Again, we fix the start and goal states to isolate the performance of smoothing. Our in-house tracker is conservative and only keeps strong detections of pedestrians, which results in many missing detections. Many gaps in detection causes the MHD between the observed trajectory and true trajectory to be large without smoothing. In contrast, the trajectories of the SPT have no missing observations due to temporal filtering but have a tendency to drift away from the pedestrian. As such, the SPT has much better performance compared to our in-house tracker before smoothing. Figure 7 shows a significant improvement for both trackers after smoothing. Despite that fact that our in-house tracker is not as robust as

Table 1. Average NLL per activity category and dataset (A and B) for (a) forecasting and (b) smoothing performance

(a) Forecasting	Proposed	MEMM	MarkovMot
approach (A)	**1.657**	1.962	2.157
depart (A)	**1.618**	1.940	2.103
walk (A)	**1.544**	2.027	2.174
approach (B)	**1.519**	1.780	2.180
depart (B)	**1.519**	1.903	2.115
walk (B)	**1.707**	1.997	2.182

(b) Smoothing	Proposed	MEMM
approach (A)	**1.602**	1.942
depart (A)	**1.594**	1.923
walk (A)	**1.483**	2.022
approach (B)	**1.465**	1.792
depart (B)	**1.513**	1.882
walk (B)	**1.695**	2.001

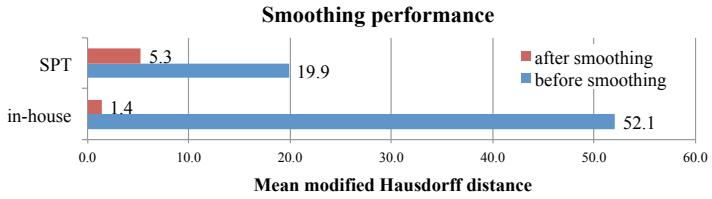


Fig. 7. Improvement in tracking accuracy with the smoothing distribution

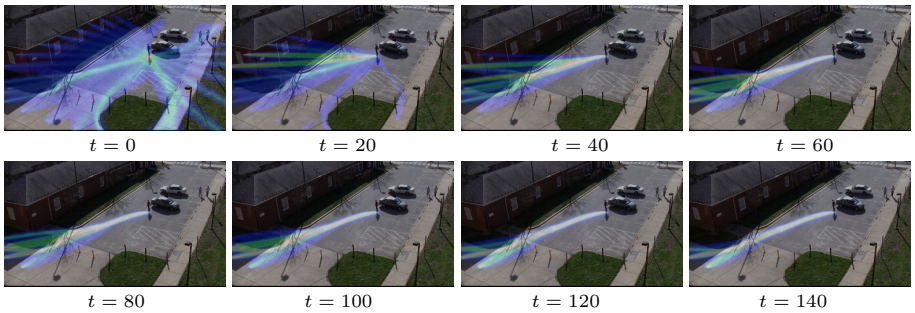


Fig. 8. Destination forecasting and path smoothing. Our proposed approach infers a pedestrians likely destinations as more noisy observations become available. Concurrently, the *smoothing distribution* (likely paths up to the current time step t) and the *forecasting distribution* (likely paths from t until the future) are modified as observations are updated

SPT, the MHD after smoothing is actually better than the SPT post-smoothing. This is due to the fact that our tracker only generates confident, albeit sparse, detections. The distributions generated by our approach also outperforms the MEMM, as shown in Table 1b.

4.4 Destination forecasting evaluation

In the most general case, the final destination of a pedestrian is not know in advance so we must reason about probable destinations as tracker observations become available. In this experiment we hold the start state and allow the destination state to be inferred by Equation (6). Figure 8 shows a visualization of destination forecasting, and consequentially, the successive updates of the forecasting and smoothing distributions. As noisy pedestrian tracker observations are acquired, the posterior distribution over destinations, the forecasting and smoothing distributions are updated. Quantitative results shown in Figure 9 show that the MHD quickly approaches a minimum for most activity categories, after about 30% of the noisy tracker trajectory has been observed. This indicates

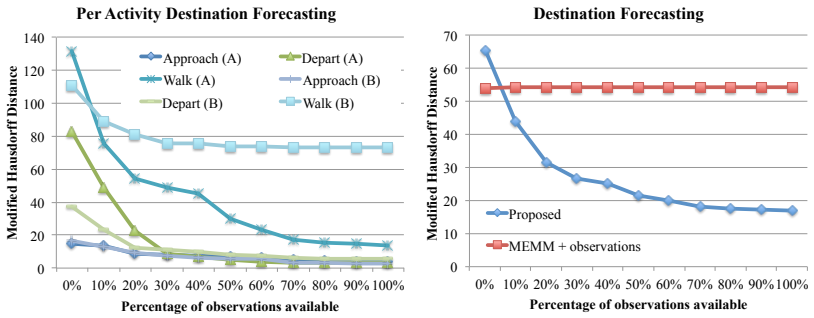


Fig. 9. Destination forecasting performance. Modified Hausdorff distance is the average distance between the ground truth trajectory and sampled trajectories from the inferred distribution. (a) per activity category performance over datasets, (b) average performance over the entire dataset

that we can forecast a person's likely path to a final destination after observing only a third of the trajectory.

4.5 Knowledge transfer

Since our proposed method encapsulates activities in terms of physical scene features and not physical location, we are also able to generalize to novel scenes. This is a major advantage of our approach over other methods that use scene-specific motion dynamics. In this experiment we use two locations: scene A and scene B, and show that learned parameters can be transferred in both directions with similar performance. Table 2 shows that the transferred parameters perform on par with scene specific parameters. With respect to forecasting performance, the average MHD between a point of the ground truth and a point of a trajectory sampled from the forecasting distribution, is degraded by 0.584 pixels. It is interesting to note that in the case of training on scene A and transferring to scene B, the transferred model actually performs slightly better. We believe that this is caused by the fact that we have more training trajectories from scene A. In Figure 10 we also show several qualitative results of trajectory forecasting and destination forecasting on novel scenes. Even without observing a single trajectory from the scene, our approach is able to generate plausible forecasting distributions for activities such as walking through the scene or departing from a car.

5 Conclusion

We have demonstrated that tools from inverse optimal control can be used for computer vision tasks in activity understanding and forecasting. Specifically, we

Table 2. MHD for knowledge transfer performance. (a) forecasting and (b) smoothing. Proposed approach can be applied to novel scenes with comparable performance

(a) Forecasting	TEST	
TRAIN	Scene A	Scene B
Scene A	**9.8520**	**7.4925**
Scene B	10.4358	8.9774
$\|\Delta\|$	0.584	1.485

(b) Smoothing	TEST	
TRAIN	Scene A	Scene B
Scene A	**3.2582**	**6.4705**
Scene B	4.9194	7.2837
$\|\Delta\|$	1.661	0.813

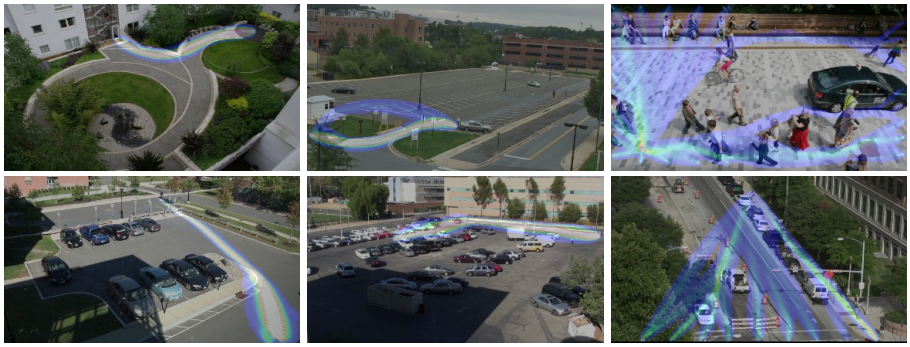


Fig. 10. Knowledge transfer examples of forecasting in novel scenes

have modeled the interaction between moving agents and semantic perception of the environment. We have also made proper modifications to accommodate the uncertainty inherent to tracking and detection algorithms. Further, the resulting formulation, based on a hidden variable MDP, provides a unified framework to support a range of operations in activity analysis: smoothing, path and destination forecasting, and transfer, which we validated both qualitatively and quantitatively. Our initial work focused on paths in order to generate an initial validation of the approach for computer vision. Moving forward, however, our proposed framework is general enough to handle non-motion representations such as sequences of discrete action-states. Similarly, we limited our evaluation to physical attributes of the environments, but an exciting possibility would be to extend the approach to activity features, similar to those used in crowd analysis, or other semantic attributes of the environment.

Acknowledgement

This research was supported in part by NSF QoLT ERC EEEC-0540865, U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016 and Cooperative Agreement W911NF-10-2-0061. We especially thank Daniel Munoz for sharing and preparing the semantic scene labeling code.

References

1. Munoz, D., Bagnell, J.A., Hebert, M.: Stacked hierarchical labeling. In: ECCV. (2010)
2. Munoz, D., Bagnell, J.A., Hebert, M.: Co-inference machines for multi-modal scene analysis. In: ECCV. (2012)
3. Ziebart, B., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J., Hebert, M., Dey, A., Srinivasa, S.: Planning-based prediction for pedestrians. In: IROS. (2009)
4. Abbeel, P., Ng, A.: Apprenticeship learning via inverse reinforcement learning. In: ICML. (2004)
5. Baker, C., Saxe, R., Tenenbaum, J.: Action understanding as inverse planning. Cognition **113**(3) (2009) 329–349
6. Ziebart, B., Maas, A., Bagnell, J., Dey, A.: Maximum entropy inverse reinforcement learning. In: AAAI. (2008)
7. Levine, S., Popovic, Z., Koltun, V.: Nonlinear inverse reinforcement learning with Gaussian processes. In: NIPS. (2011)
8. Morris, B., Trivedi, M.: A survey of vision-based trajectory learning and analysis for surveillance. Transactions on Circuits and Systems for Video Technology **18**(8) (2008) 1114–1127
9. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: ECCV. (2008)
10. Zen, G., Ricci, E.: Earth mover's prototypes: A convex learning approach for discovering activity patterns in dynamic scenes. In: CVPR. (2011)
11. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: CVPR. (2009)
12. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV. (2009)
13. Turek, M., Hoogs, A., Collins, R.: Unsupervised learning of functional categories in video scenes. In: ECCV. (2010)
14. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: ECCV. (2008)
15. Kaucic, R., Amitha Perera, A., Brooksby, G., Kaufhold, J., Hoogs, A.: A unified framework for tracking through occlusions and across sensor gaps. In: CVPR. (2005)
16. Gong, H., Sim, J., Likhachev, M., Shi., J.: Multi-hypothesis motion planning for visual object tracking. In: ICCV. (2011)
17. Xing, Z., Pei, J., Dong, G., Yu, P.: Mining sequence classifiers for early prediction. In: SIAM international conference on data mining. (2008)
18. Ryoo, M.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV. (2011)
19. Hoai, M., De la Torre, F.: Max-margin early event detectors. In: CVPR. (2012)
20. Bellman, R.: A Markovian decision process. Journal of Mathematics and Mechanics, **6**(5) (1957) 679–684
21. Ratliff, N., Bagnell, J., Zinkevich, M.: Maximum margin planning. In: ICML. (2006)
22. Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C., Lee, J., Mukherjee, S., Aggarwal, J., Lee, H., Davis, L., et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: CVPR. (2011)
23. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: ICCV. (2011)