Action-Reaction: Forecasting the Dynamics of Human Interaction⋆

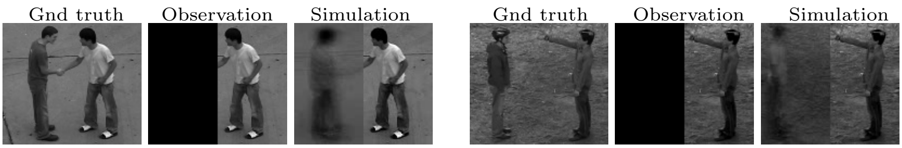
De-An Huang and Kris M. Kitani

Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract. Forecasting human activities from visual evidence is an emerging area of research which aims to allow computational systems to make predictions about unseen human actions. We explore the task of activity forecasting in the context of dual-agent interactions to understand how the actions of one person can be used to predict the actions of another. We model dual-agent interactions as an optimal control problem, where the actions of the initiating agent induce a cost topology over the space of reactive poses – a space in which the reactive agent plans an optimal pose trajectory. The technique developed in this work employs a kernel-based reinforcement learning approximation of the soft maximum value function to deal with the high-dimensional nature of human motion and applies a mean-shift procedure over a continuous cost function to infer a smooth reaction sequence. Experimental results show that our proposed method is able to properly model human interactions in a high dimensional space of human poses. When compared to several baseline models, results show that our method is able to generate highly plausible simulations of human interaction.

1 Introduction

It is our aim to expand the boundaries of human activity analysis by building intelligent systems that are not only able to classify human activities but are also capable of mentally simulating and extrapolating human behavior. The idea of predicting unseen human actions has been studied in several contexts, such as early detection [17], activity prediction [22], video gap-filling [2] and activity forecasting [9]. The ability to predict human activity based on visual



UT-Interaction 1 **UT-Interaction 2**

Fig. 1. Examples of ground truth, observation, and our simulation result

⋆ Electronic supplementary material -Supplementary material is available in the online version of this chapter at http://dx.doi.org/10.1007/978-3-319-10584-0_32. Videos can also be accessed at http://www.springerimages.com/videos/978-3-319-10583-3

D. Fleet et al. (Eds.): ECCV 2014, Part VII, LNCS 8695, pp. 489–504, 2014.

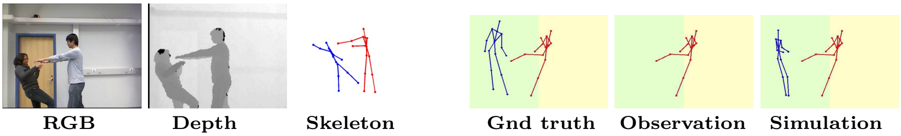


Fig. 2. Left three are the RGB, depth, and tracked skeleton images of SBU dataset. Right three images show our ground truth, observation, and simulation.

observations of the world is essential for advances in domains such as assistive robotics [10], human-robot interaction [7], robust surveillance [2], and smart coaching systems. For example, in the context of video surveillance, it is often the case that human activities are not fully visible to the camera due to occlusion, and in extreme cases parts of the activity may fall *outside* of the field of view (e.g., two people fighting at the periphery of the screen). A human observer however, can extrapolate what is happening despite large amounts of missing data. By observing a single person punching something outside of the field of view, we can visualize with high accuracy how the opponent has been hit. The important point being that humans have the ability to leverage contextual information to make very accurate predictions despite large amounts of visual occlusion. In this work, we aim to build a system that is able to predict and more importantly *simulate* human behavior in both space and time from partial observations.

We simplify our target domain by focusing on understanding and simulating dual-agent interaction. Traditionally dual-agent interactions (e.g.., hugging, pushing) have been represented as a joint phenomenon, where observations from both people are used as features to recognize human interactions from video [22,2,21,8,26]. Alternatively, human interactions can also be modeled as a dependent process, where one person is reacting to the actions of an initiating agent. In this work we model dual-agent interaction as a reactive control system, where the actions of the initiating agent induces a cost topology over the space of reactive poses – a space in which the reactive agent plans an optimal pose trajectory. This alternative representation of human interaction is a fundamentally new way of modeling human interactions for vision-based activity analysis.

The use of a decision-theoretic model for vision-based activity analysis has been proposed previously by Kitani *et al.* [9], where a cost function was learned over a low-dimensional 2D floor plane (with only 4 possible actions) for a single agent. While their work highlighted the importance of decision-theoretic modeling, the framework was defined over a low-dimensional state space (and action space) and was limited by the assumption of a single agent acting in a static world. In reality, the world is not static and people interact with each other. Additionally, if we desire to model human pose, the state space through which a person moves is extremely high-dimensional. To give an example, the pose space used in this work is a 819 dimensional HOG feature space, where both the state and action space are extremely large. In this scenario, it is no longer feasible to use the discrete state inference procedure used in [9].

In this work, we aim to go beyond a two dimensional state space and forecast dual-agent activities in a high-dimensional pose space. In particular, we introduce kernel-based reinforcement learning [18] to handle the high-dimensionality of human pose. Furthermore, we introduce an efficient mean-shift inference procedure [4] to find an optimal pose trajectory in the continuous cost function space. In comparative experiments, the results verify that our inference method is able to effectively represent human interactions. Furthermore, we show how this procedure proposed for 2D dual-agent interaction forecasting can also be applied to 3D skeleton pose data. Our final qualitative experiment also shows how the proposed model can be used for human pose analysis and anomaly detection.

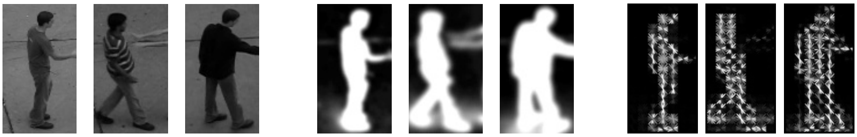
Interestingly, the idea of generating a reactive pose trajectory has been explored largely in computer graphics; a problem known as interactive control. The goal of interactive control is to create avatar animations in response to user input [12,15]. Motion graphs [11] created from human motion data are commonly used, and the motion synthesis problem is transformed into selecting proper sequences of nodes. However, these graphs are discrete and obscure the continuous properties of motion. In response, a number of approaches have been proposed to alleviate this weakness and perform continuous control of character [24,13,14]. It should be noted that all of the interactive control approaches focus on synthesizing animations in response to a clearly defined mapping [11,15] from the user input to pose. In contrast, we aim to simulate human reaction based only on visual observations, where the proper reaction is non-obvious and must be learned from the data.

2 Dual Agent Forecasting

Our goal is to build a system that can simulate human reaction based only on visual observations. As shown in Figure 1, the ground truth consists of both the true reaction $\boldsymbol{g} = [g^1 \cdots g^T]$ on the left hand side (LHS) and the observation $\boldsymbol{o} = [o^1 \cdots o^T]$ of the initiating agent on the right hand side (RHS). In training time, M demonstrated interaction pairs $\boldsymbol{g}_m$ and $\boldsymbol{o}_m$ are provided for us to learn the cost topology of human interaction. At test time, only the actions of the initiating agent $\boldsymbol{o}$ (observation) on the RHS is given. We perform inference over the learned cost function to obtain an optimal reaction sequence $\boldsymbol{x}$.

2.1 Markov Decision Processes

In this work, we model dual-agent interaction as a Markov decision process (MDP) [1]. At each time step, the process is in some state c, and the agent may choose any action a that is available in state c. The process responds by moving to a new state c' at the next time step. The MDP is defined by an initial state distribution $p_0(c)$ a transition model $p(c'|c,a)$ and a reward function $r(c,a)$, which is equivalent to the negative cost function. Given these parameters, the goal of *optimal control* is to learn the optimal policy $\pi(a|c)$, which encodes the distribution of action a to take when in state c that can maximize the expected



(a) Input image (b) Foreground map (c) HOG

Fig. 3. HOG features in (c) are our 819 dimensional states, which are the HOG responses of the input images weighted by the probability of foreground maps in (b)

reward (minimize the expected cost). In this work, the actions are *deterministic* because we assume humans have perfect control over their body where one action will deterministically bring the pose to the next state. Therefore, $p(c'|c, a)$ concentrates on a single state $c' = c_a$ and is zero for other states.

2.2 States and Actions

States. We use a HOG [6] feature of the whole image as a compact state representation, which does not contain the redundant textural information in the raw images. Some visualizations are shown in Figure 3. Note that only the poses on the left hand side (LHS) are referred as states, while poses on the right hand side (RHS) are our observations. We further make two changes to adapt HOG feature to our current application, pose representation. First, the HOG is weighted by probability of foreground (PFG) of the corresponding image because we are only interested in the human in the foreground. The PFG is computed by median filtering followed by soft thresholding. Second, we average the gradient in the 2×2 overlapping cells in HOG to reduce its dimension. This results in a continuous high-dimensional vector of 819 dimensions (64×112 bounding box).

Actions. Even with a continuous state space, a discrete set of actions is still more efficient to solve the MDP when possible [14]. Furthermore, there are actually many redundant actions for similar states that can be removed [23]. To alleviate redundancy, we perform k-means clustering on all the training frames on the LHS to quantize the continuous state space into K discrete states. For each cluster c ($c = 1$ to K), we will refer to the cluster center X_c as the HOG feature of quantized state c. The kth action is defined as going from a quantized state c to the kth nearest state, which gives us a total K actions. In the rest of the paper, we will fix this quantization. Given a new pose vector (HOG feature) x on the LHS, it is quantized to state c if X_c is the closest HOG feature to x.

2.3 Inverse Optimal Control over Quantized State Space

In this work, we model dual-agent interaction as an optimal control problem, where the actions of the initiating agent induce a cost topology over the space of reactive poses. Given M demonstrated interaction pair $\boldsymbol{o}_m$ (Observation) and $\boldsymbol{g}_m$ (true reaction), we leverage recent progress in inverse optimal control (IOC) [27] to recover a discretized approximation of the underlying cost topology.

In contrast to optimal control in Section 2.1, the cost function is not given in IOC and has to be derived from demonstrated examples [16]. We make an important assumption about the form of the cost function, which enables us to translate from visual observations to a single cost for reactive poses. The reward (negative cost) of a state c and an action a:

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

is assumed to be a weighted combination of feature responses $\boldsymbol{f}(c, a) = [f_1(c, a) \cdots f_J(c, a)]^\top$, where each $f_j(c, a)$ is the response of a type of feature extracted from the video, such as the velocity of the agent's center of mass, and $\boldsymbol{\theta}$ is a vector of weights for each feature. By learning these parameters $\boldsymbol{\theta}$, we are learning how the actions of the initiating agent affect the reaction of the partner. For example, a feature such as moving forward will have a high cost for punching interactions because moving forward increases the possibility of being hit by the punch. In this case, the punching activity induces a high cost on moving forward and implies that this feature should have a high weight in the cost function. This explicit modeling of human interaction dynamics via the cost function sets our approach apart from traditional human interaction recognition models.

In this work, we apply the maximum entropy IOC approach [27] on the quantized states to learn a discretized approximation of the cost function. In this case, for a pose sequence $\boldsymbol{x} = [x^1 \cdots x^T]$ on the LHS, we quantize it into sequence $\boldsymbol{c} = [c^1 \cdots c^T]$ of quantized states defined in Section 2.2. In the maximum entropy framework [27], the distribution over a sequence $\boldsymbol{c}$ of quantized states and the corresponding sequence $\boldsymbol{a}$ of actions is defined as:

$$P(\boldsymbol{c}, \boldsymbol{a}; \boldsymbol{\theta}) = \frac{\prod_t e^{r(c^t, a^t)}}{Z(\boldsymbol{\theta})} = \frac{e^{\sum_t \boldsymbol{\theta}^\top \boldsymbol{f}(c^t, a^t)}}{Z(\boldsymbol{\theta})}, \tag{2}$$

where $\boldsymbol{\theta}$ are the weights of the cost function, $\boldsymbol{f}(c^t, a^t)$ is the corresponding vector of features of state c^t and action a^t, and $Z(\boldsymbol{\theta})$ is the partition function.

In the training step, we quantize M training pose sequences $\boldsymbol{g}_1 \cdots \boldsymbol{g}_M$ on the LHS to get the corresponding sequences $\boldsymbol{c}_1 \cdots \boldsymbol{c}_M$ of quantized states. We then recover the reward function parameters $\boldsymbol{\theta}$ by maximizing the likelihood of these sequences under the maximum entropy distribution (2). We use exponentiated gradient descent to iteratively maximize the likelihood. The gradient can be shown to be the difference between the *empirical* mean feature count $\bar{\mathbf{f}} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{f}(\boldsymbol{c}_m, \boldsymbol{a}_m)$, the average feature counts over the demonstrated training sequences, and the *expected* mean feature count $\hat{\mathbf{f}}_\theta$, the average feature counts over the sequences generated by the parameter $\boldsymbol{\theta}$. With step size η, we update $\boldsymbol{\theta}$ by $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t e^{\eta(\bar{\mathbf{f}} - \hat{\mathbf{f}}_\theta)}$. In order to compute the expected feature count $\hat{\mathbf{f}}_\theta$, we use a two-step algorithm similar to that described in [9] and [27].

Backward Pass. In the first step, current weight parameters $\boldsymbol{\theta}$ is used to compute the expected reward $V^{(t)}(c)$ to the goal from any possible state c at any time step t. The expected reward function $V^{(t)}(c)$ is also called the *value*

Algorithm 1. Backwards pass

```
$V^{(T)}(c) \leftarrow 0$
for $t = T-1, \dots, 2, 1$ do
    $V^{(t)}(c) = \text{soft}\max_a r(c, a; \boldsymbol{\theta}) + V^{(t+1)}(c_a)$
    $\pi_\theta^{(t)}(a|c) \propto e^{V^{(t)}(c_a) - V^{(t)}(c)}$
end for
```

Algorithm 2. Forward pass

```
$D^{(1)}(c) \leftarrow \frac{1}{K}$
for $t = 1, 2, \dots, T-1$ do
    $D^{(t+1)}(c_a)$ += $\pi_\theta^{(t)}(a|c) D^{(t)}(c)$
end for
$\hat{\mathbf{f}}_\theta = \sum_t \sum_c \sum_a \boldsymbol{f}^{(t)}(c, a) D^{(t)}(c)$
```

function in reinforcement learning. The maximum entropy policy is $\pi_\theta^{(t)}(a|c) \propto e^{V^{(t)}(c_a) - V^{(t)}(c)}$, where c is the current state, a is an action, and c_a is the state we will get by performing action a at state c. In other words, the probability of going to a state c_a from c is exponentially proportional to the increase of expected reward or value. The algorithm is summarized in Algorithm 1.

Forward Pass. In the second step, we propagate an uniform initial distribution $p_0(c) = \frac{1}{K}$ according to the learned policy $\pi_\theta^{(t)}(a|c)$, where K is the number of states (clusters). We do not assume c^1 is known as in [9] and [27]. In this case, we can compute the *expected state visitation count* $D^{(t)}(c)$ of state c at time step t. Therefore, the expected mean feature count can be computed by $\hat{\mathbf{f}}_\theta = \sum_t \sum_c \sum_a \boldsymbol{f}^{(t)}(c, a) D^{(t)}(c)$. The algorithm is summarized in Algorithm 2.

2.4 Features for Human Interaction

According to (1), the features define the expressiveness of our cost function and are crucial to our method in modeling dynamics of human interaction. Now we describe the features we use in our method. In this work, we assume that the pose sequence $\boldsymbol{o} = [o^1 \cdots o^T]$ of the initiating agent is observable on the RHS. For each frame t, we compute different features $\boldsymbol{f}^{(t)}(c, a)$ from the sequence $\boldsymbol{o}$.

Cooccurrence. Given a pose o^t on the RHS, we want to know how often a state c occurs on the LHS. This provides a strong clue for simulating human interaction. For example, when the hand of the pose o^t is reaching out, there is a high chance that the hand of the reacting person is also reaching out in response. This can be captured by the cooccurrence of reaching out poses on both LHS and RHS. Therefore, the cooccurrence feature $f_1^{(t)}(c, a) = P(c|o^t)$ is the posterior state distribution given the observation on the RHS. We estimate this distribution by discrete approximation. We quantize the observed pose o^t to observable quantized state c_o^t by k-means clustering as in Section 2.2, but now the quantization is on the RHS rather than the LHS. We approximate $P(c|o^t)$ by $P(c|c_o^t)$, which can be estimated by histogram density estimation.

Transition Probability. We want to know what actions will occur at a state c, which model the probable transitions between consecutive states. For example, at a state c that the agent is moving forward, transition to a jumping back state is less likely. Therefore, the second feature is the transition probability $f_2^{(t)}(c, a)$ $= P(c_a|c)$, where c_a is the state we will get to by performing action a at state

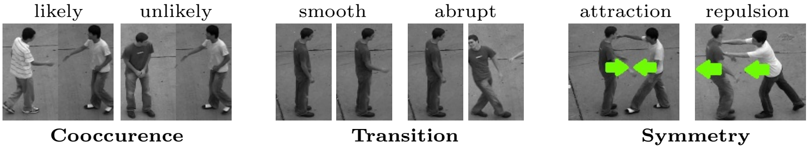


Fig. 4. We use statistics of human interaction as our features for the cost function

c. We accumulate the transition statistics from the M training sequences $\boldsymbol{c}_m$ of quantized states on the LHS. This feature is independent of time step t.

Centroid Velocity. We use centroid velocities to capture the movements of people when they are interacting. For example, it is unlikely that the centroid position of human will move drastically across frames and actions that induce high centroid velocity should be penalized. Therefore, we define the feature *smoothness* as $f_3^{(t)}(c,a) = 1 - \sigma(|v(c,a)|)$, where $\sigma(\cdot)$ is the sigmoid function, and $v(c,a)$ is the centroid velocity of action a at state c. Only the velocity along the x-axis is used. In addition, the relative velocity of the interacting agents gives us information about the current interaction. For example, in the hugging activity, the interacting agents are approaching each other and will have centroid velocities of opposite directions. Therefore, we define the feature *attraction* as $f_4^{(t)}(c,a) = \mathbb{1}(v_o^t \times v(c,a) < 0)$, where $\mathbb{1}(\cdot)$ is the indicator function, and v_o^t is the centroid velocity of the initiating agent at time t. This feature will be one if the interacting agents are moving in a symmetric way. We also define the complementary feature *repulsion* as $f_5^{(t)}(c,a) = \mathbb{1}(v_o^t \times v(c,a) > 0)$ to capture the opposite case when the agents are repulsive to each other.

2.5 Quantized State Inference

Given a set of demonstrated interactions, we can learn a discretized approximation of the underlying cost function by the IOC algorithm presented in Section 2.3. At test time, only the pose sequence of the initiating agent $\boldsymbol{o}_{test}$ on the RHS is observable. We first compute the features $\boldsymbol{f}^{(t)}(c,a)$ under the observation $\boldsymbol{o}_{test}$, and weight the features by the learned weight parameters $\boldsymbol{\theta}$ to get the reward function (negative cost) $r(c,a;\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{f}(c,a)$. This gives us the approximated cost topology induced by $\boldsymbol{o}_{test}$, the pose sequence of the initiating agent.

In discrete Markov decision process, inferring the most probable sequence is straightforward: First, we fix the induced $r(c,a;\boldsymbol{\theta})$ and perform one round of backwards pass (Algorithm 1) to get the discrete value function $V^{(t)}(c)$. At each time step t, the most probable state is the state with the highest value. However, the result depends highly on the selection of K in this case. If we choose K too large, the $O(K^2T)$ Algorithm 1 becomes computational prohibited. Furthermore, the histogram based estimations become unreliable. On the other hand, if we choose K too small, the quantization error can be large.

Algorithm 3. Extended Mean Shift Inference

```
Compute $V^{(t)}(c)$ by Algorithm 1
$x^1 = X_{c^*}$, where $c^* = \arg\max_c V^{(1)}(c)$
for $t = 2, \ldots, T$ do
    $x_0 = x^{t-1}$, $w_c = V^{(t)}(c)$
    while not converged do
        $x_{i+1} = \frac{1}{C_h} \sum_{c=1}^{K} X_c w_c K_h(x_i, X_c)$, where $C_h = \sum_{c=1}^{K} w_c K_h(x_i, X_c)$
    end while
    $x^t = x_{converged}$
end for
```

2.6 Kernel-Based Reinforcement Learning

In order to address the problems of discretizing the state space, we introduce kernel-based reinforcement learning (KRL) [18] to our problem. Based on KRL, the value function $V_h^{(t)}(x)$ for any pose x in the *continuous* state space is assumed to be a weighted combination of value functions $V^{(t)}(c)$ of the quantized states. This translate our inference from discrete to continuous state space. At each time step t, the value function of a continuous state x is:

$$V_h^{(t)}(x) = \frac{\sum_{c=1}^{K} K_h(x, X_c) V^{(t)}(c)}{\sum_{c=1}^{K} K_h(x, X_c)}, \tag{3}$$

where X_c is the HOG feature of the quantized state c, and $K_h(\cdot, \cdot)$ is a kernel function with bandwidth h. In this work, we use the normal kernel.

The advantage of KRL is two-fold. First, it guarantees the smoothness of our value function. Second, we have the value function on the continuous space. Therefore, even with smaller K, we can still perform continuous inference. Furthermore, this formulation allows us to perform efficient optimization for x with maximal $V_h^{(t)}(x)$ as we will show in the next section.

2.7 Extended Mean Shift Inference

Now that we have the value function $V_h^{(t)}(x)$ on the continuous state space, we want to find the pose x^* with the highest value. In contrast to optimization in the discretized space, it is infeasible to enumerate the values of all the states in continuous space. We leverage the property of human motion to simplify the optimization problem. Since human motion is smooth and will not change drastically across frames, the optimal next pose should appear in a local neighborhood of the current pose. This restricts our search space of optimal pose to a local neighborhood. In addition, we leverage the resemblance of our formulation in (3) to the well-studied *kernel density estimation* (KDE) in statistics, which has also achieved considerable success in the area of object tracking [3]. To optimize the density in KDE, the standard approach is to apply the mean shift procedure, which will converge robustly to the local maximum of the density function.

Our formulation allows us to leverage the similarity of our problem to KDE and apply the extended mean shift framework proposed in [3] to perform efficient inference. As shown in [3], the maximization of a function of the form

$$\frac{\sum_{c=1}^{K} w_c K_h(x, X_c)}{\sum_{c=1}^{K} K_h(x, X_c)} \tag{4}$$

can be done efficiently by the extended mean shift iterations

$$x_{i+1} = \frac{\sum_{c=1}^{K} X_c w_c G_h(x_i, X_c)}{\sum_{c=1}^{K} w_c G_h(x_i, X_c)} \tag{5}$$

until convergence, where G_h is the negative gradient of K_h. In normal kernel, G_h and K_h has the same form [4]. Therefore, we can replace G_h in (5) by K_h.

Our goal is to find the pose x that maximize the value function $V_h^{(t)}(x)$ locally. In this case, the expression in (3) will have the exact same form to optimize in (4) if we define $w_c = V^{(t)}(c)$. Therefore, we derive our final extended mean shift update rule by scaling $V^{(t)}(c)$ linearly to $[0, 1]$ as w_c. The algorithm is summarized in Algorithm 3. The mean shift iterations is performed at each time step, where the update is initialize by the pose of the last frame x^{t-1}, and the converged result is taken as x^t. In our experiments, the first frame x^1 is initialized by the quantized state with the highest value.

3 Experiments

Our goal is to build intelligent systems that are capable of mentally simulating human behavior. Given two people interacting, we observe only the actions of the initiator on the right hand side (RHS) and attempt to forecast the reaction on the left hand side (LHS). For video in which the initiator is on the LHS, we flip the video to put the initiator on the RHS. Since we do not have access to the ground truth distribution over all possible reaction trajectories, we measure how well the learned policy is able to describe the single ground truth trajectory. For interaction videos, we use videos from three datasets, *UT-Interaction 1*, *UT-Interaction 2* [20], and *SBU Kinect Interaction Dataset* [25] where the UTI datasets consist of only RGB videos, and SBU dataset consists of RGB-D (color plus depth) human interaction videos. In each interaction video, we occlude the ground truth reaction $\boldsymbol{g} = [g^1 \cdots g^T]$ on the LHS, observe $\boldsymbol{o} = [o^1 \cdots o^T]$ the action of the initiating agent on the RHS, and attempt to forecast $\boldsymbol{g}$. For experiments, we will first evaluate which model provides the best representation for human reaction. Then we will evaluate the features of the cost function.

3.1 Metrics

We compare the ground truth sequence with the learned policy using two metrics. The first one is probabilistic, which measures the probability of performing the ground truth reaction under the learned policy. A higher probability means the learned policy is more consistent with the ground truth reaction sequence. We use the Negative Log-Likelihood (NLL):

$$-\log P(\boldsymbol{g}|\boldsymbol{o}) = -\sum_t \log P(g^t|g^{t-1}, \boldsymbol{o}), \tag{6}$$

as our probabilistic metric. For discrete models, the ground truth reaction sequence is quantized into a sequence $\boldsymbol{c}$ of quantized states. The probability is evaluated by $P(g^t|g^{t-1}, \boldsymbol{o}) = P(c^t|c^{t-1}, \boldsymbol{o})$. For our continuous model, $P(g^t|g^{t-1}, \boldsymbol{o})$ are interpolated according to (3). The second metric is deterministic, which directly measure the physical HOG distance (or joint distance for skeleton video) of the ground truth reaction $\boldsymbol{g}$ and the reaction simulated by the learned policy. The deterministic metric is the average frame distance:

$$\frac{1}{T-1}\sum_t ||g^t - x^t||^2 \tag{7}$$

where x^t is the resulting reaction pose at frame t. The distance is not computed for the last frame because the reward function $r(c, a)$ is not defined.

3.2 Evaluating the Interaction Model

For model evaluation, we select three baselines to compare with the proposed method. The first baseline is the per frame nearest neighbor (NN) [5], which only uses the *cooccurrence* feature at each frame *independently* and does not take into account the effect of consecutive states. For each observation o^t, we find the LHS quantized state with the highest cooccurrence. That is $c^t_{NN} = \arg\max_c P(c|c^t_o) \approx P(c|o^t)$, where c^t_o is the observable quantized state of o^t.

The second baseline is the hidden Markov model (HMM) [19], which has been widely used to recover hidden time sequences. HMM is defined by the transition probabilities $P(c^t|c^{t-1})$ and emission probabilities $P(o^t|c^t)$, which are equivalent to our *transition* and *cooccurrence* features. However, the weights for these two features are always the same in HMM, while our algorithm learns the optimal feature weights $\boldsymbol{\theta}$. The likelihood is computed by the forward algorithm and the resulting state sequence $\boldsymbol{c}_{HMM}$ is computed by the Viterbi algorithm.

Table 1. Average frame distance (AFD) and NLL per activity category for UTI

(a)AFD	NN[5]	HMM[19]	MDP[9]	Proposed
shake	5.35	5.21	4.68	**3.14**
hug	4.00	4.06	3.74	**2.88**
kick	6.17	6.16	5.33	**3.96**
point	3.62	3.62	3.31	**2.45**
punch	5.10	4.99	4.23	**3.03**
push	4.90	4.91	4.01	**3.24**

(b)NLL	NN[5]	HMM[19]	MDP[9]	Proposed
shake	651.04	**473.91**	862.81	476.10
hug	751.46	608.81	958.49	**487.21**
kick	382.62	**263.08**	550.52	282.36
point	577.22	426.40	750.11	**374.73**
punch	353.85	260.72	483.01	**257.06**
push	479.33	357.00	561.01	**320.92**

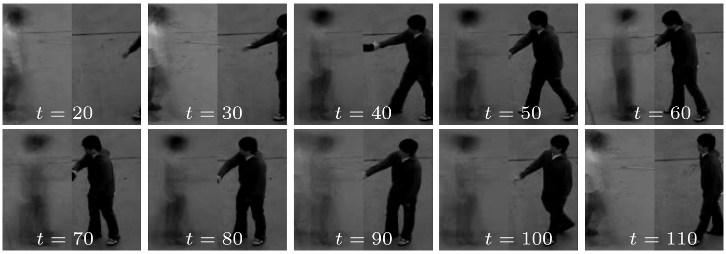


Fig. 5. Forecasting result of UTI dataset 1. The RHS is the observed initiator, and the LHS is the simulated reaction of the proposed method. The activity is shaking hands.

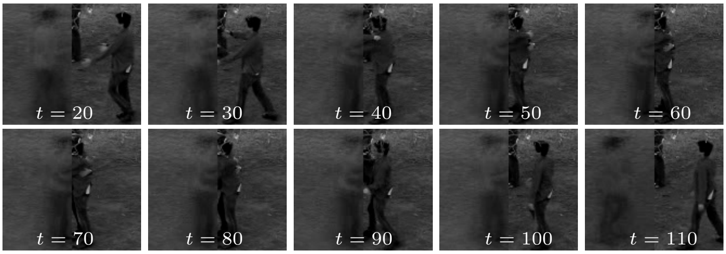


Fig. 6. Forecasting result of UTI dataset 2. The RHS is the observed initiator, and the LHS is the simulated reaction of the proposed method. The activity is hugging.

The third baseline is the discrete state inference in Section 2.5. This can be seen as applying the discrete Markov decision process (MDP) inference used in [9] to a quantized state space. We will refer to this baseline as MDP. The likelihood for MDP is computed by $\prod_t \pi_\theta^{(t)}(a^t|c^t)$, the stepwise product of the policy executions. We follow [9] and produce the probabilistic-weighted output.

We first evaluate our method on *UT-Interaction 1*, and *UT-Interaction 2* [20] datasets, which consist of RGB videos only, and some examples have been shown in Figure 1. The UTI datsets consist of 6 actions: hand shaking, hugging, kicking, pointing, punching, pushing. Each action has a total of 10 sequences for both datasets. We use 10-fold evaluation as in [2]. We use $K = 100$ in the experiments. We now evaluate which method can best simulate human reaction.

The average NLL and frame distance per activity for each baseline is shown in Table 1. It can be seen that, optimal control based methods (MDP and proposed) outperform the other two baselines in terms of frame distance. In addition, the proposed mean shift inference achieves the lowest frame distance for all activities and significantly outperforms other baselines because we use kernel-based reinforcement learning to alleviate quantization error and the mean shift inference ensures the smoothness of the resulting reaction trajectory. It should be noted that although the MDP is able to achieve lower frame distance than NN and HMM, the NLL is higher. This is because the performance of discretized inference can be affected significantly by unseen data. For example, if a transition is

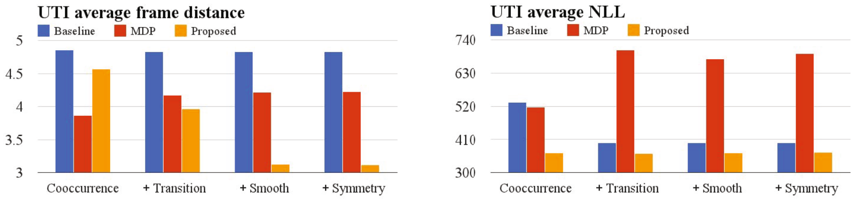


Fig. 7. Ablative analysis shows that the proposed method continually outperforms the baselines and verifies the effectiveness of our features

not observed in the training data, it will generate a low transition probability feature and induce a high cost in the IOC framework. This will make the overall likelihood of the ground truth significantly lower (a high NLL). On the other hand, our kernel-based reinforcement learning framework interpolates a smooth value function over the continuous state space and alleviates this phenomenon. The effectiveness of our approach is verified by the NLL shown in Table 1. Some visualization of the results are shown in Figure 5 and Figure 6.

3.3 Evaluating the Effect of Features

As noted in Section 2.4, the features define the expressiveness of our cost function, and are essential for us to model the dynamics of human interaction. In the previous section, we have shown that the proposed method is the best interaction model. We now evaluate the effects of different features for our model.

The average NLL and frame distance for the entire UTI dataset (1 and 2) using different features are shown in Figure 7. The performances of baselines and MDP are also shown for reference. It should be note that because centroid-based features (smooth, attraction, repulsion) cannot be easily integrated into baselines NN and HMM, the performances of HMM still only use the first two features in the *+Smooth* and *+Symmetry* columns. It can be seen that adding more features help our method to learn a policy that is more consistent with the ground truth, and significantly outperforms other baselines because our kernel-based reinforcment learning and mean-shift framework provides an efficient way for inference over a continuous space and ensures the smoothness of the result.

3.4 Extension to 3D Pose Space

To show that our method can also work in 3D pose space (not just 2D), we evaluate our method on *SBU Kinect Interaction Dataset* [25], in which interactions performed by two people are captured by a RGB-D sensor and tracked skeleton positions at each frame are provided. In this case, the state space becomes a 15×3 (joint number times x, y, z) dimensional continuous vector. We use $K = 50$ for

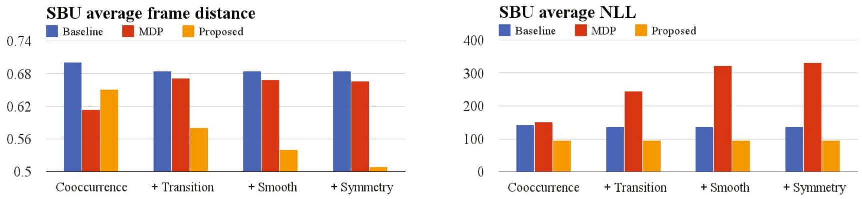


Fig. 8. Ablative analysis shows the effectiveness of our features and verifies that our 2D interaction forecasting framework can also be applied to 3D skeleton data

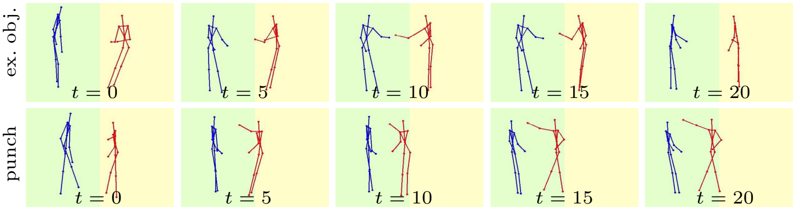


Fig. 9. Forecasting result of SBU dataset. The red skeleton is the initiating agent (observation), and the blue skeleton is our simulation result. Top row shows a result of activity exchaging object, and the bottom row shows a result of acitivity punching.

the actions because the dataset contains less frames per video compared to the UT-Interaction datasets. The SBU dataset consists of 8 actions: approaching, departing, kicking,pushing, shaking hands, hugging, exchanging object, punching. The first two actions (approaching & departing) are excluded from our experiments because the initiating agent has no action and provides no information for forecasting. 7 participants performed activities in the dataset and results in 21 video sets, where each set contains videos of a pair of different people performing all interactions. We use 7-fold evaluation, in which videos of one participants are held out for one fold. The average NLL and frame distance per activity are shown in Table 2. Again, the proposed model achieves the best performance on both frame distance and NLL. The feature evaluation results are shown in Figure 8. It can be seen that adding more features is beneficial for modelling the dynamics of human interaction. A visualization of the results are shown in Figure 9. The top row of the figure shows the result of activity 'exchanging object'. It can be seen that, the forecasting result (blue skeleton) raises his hand to catch the object

Table 2. Average frame distance (AFD) and NLL per activity category for SBU dataset

(a)AFD	NN[5]	HMM[19]	MDP[9]	Proposed
kick	0.855	0.824	0.875	**0.660**
push	0.575	0.559	0.573	**0.413**
shake	0.551	0.537	0.503	**0.389**
hug	0.768	0.751	0.690	**0.504**
exchange	0.755	0.742	0.724	**0.574**
punch	0.700	0.692	0.633	**0.510**

(b)NLL	NN[5]	HMM[19]	MDP[9]	Proposed
kick	107.17	92.89	308.85	**74.65**
push	143.46	139.84	399.56	**99.06**
shake	187.11	183.14	381.97	**120.89**
hug	166.06	169.62	284.24	**112.21**
exchange	133.87	124.95	309.43	**87.87**
punch	111.81	111.05	306.11	**78.98**

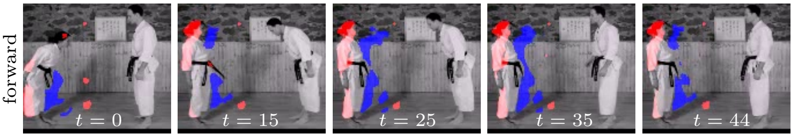


Fig. 10. Anomaly detection results. The standing part of the reacting agent is detected as anomalous (red) and the blue pixels form a proper bowing reaction.

provided by the initiating agent (red skeleton). The bottom row of the figure shows the result of activity 'punching'. Our result forecasts correctly that the opponent will avoid the punch by moving back.

3.5 Extension to Per-pixel Anomaly Detection

While we have shown that our model is able to extrapolate human behavior from a partially observed video, the application of the learned reaction policy is not limited to this scenario. We extend the proposed method to *anomaly detection*. We address this problem by comparing the simulated probability of foreground (PFG) map and the PFG map of the testing sequence. We downloaded four Karate bowing videos from YouTube. We train our model on three of the videos and test on the remaining one. We synthesize a anomalous reaction by shifting the LHS of the testing video 20 frames forward temporally. The visual anomaly detection result is shown in Figure 10. The anomalous part of the body pose is labeled as red and the normal parts of the pose are shown in blue. This visual feedback can be used in training scenarios for social interaction or sports.

4 Conclusions

We have presented a fundamentally new way of modeling human interactions for vision-based activity analysis. While interactions have traditionally been modeled as a joint phenomenon for recognition, we treat human interactions as a dependent process and explicitly model the interactive dynamics of human interaction. We have pushed beyond previous optimal control approaches for low-dimensional spaces and have introduced kernel-based reinforcement learning and mean-shift procedure to tackle the high-dimensional and continuous nature of human poses. Experimental results verified that our proposed method is able to generate highly plausible simulations of human reaction and outperforms several baseline models. Furthermore, we have shown successful extensions to 3D skeleton pose data and an application to the task of pose-based anomaly detection.

Acknowledgement. This research was sponsored in part by the Army Research Laboratory (W911NF-10-2-0061) and by the National Science Foundation (Purposeful Prediction: Co-robot Interaction via Understanding Intent and Goals).

References

1. Bellman, R.: A Markovian decision process. Journal of Mathematics and Mechanics 6(5), 679–684 (1957)
2. Cao, Y., Barrett, D.P., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S.J., Siskind, J.M., Wang, S.: Recognize human activities from partially observed videos. In: CVPR (2013)
3. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR (2000)
4. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24(5), 603–619 (2002)
5. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions in Information Theory IT-13(1), 21–27 (1967)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
7. Dragan, A.D., Lee, K.C.T., Srinivasa, S.S.: Legibility and predictability of robot motion. In: ACM/IEEE International Conference on Human-Robot Interaction (2013)
8. Gaur, U., Zhu, Y., Song, B., Chowdhury, A.K.R.: A "string of feature graphs" model for recognition of complex activities in natural videos. In: ICCV (2011)
9. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 201–214. Springer, Heidelberg (2012)
10. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. In: RSS (2013)
11. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: SIGGRAPH 2002 Conference Proceedings. Annual Conference Series, pp. 473–482. ACM Press/ACM SIGGRAPH (2002)
12. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. ACM Trans. Graph. 21(3), 491–500 (2002)
13. Lee, Y., Wampler, K., Bernstein, G., Popovic, J., Popovic, Z.: Motion fields for interactive character locomotion. ACM Trans. Graph. 29(6), 138 (2010)
14. Levine, S., Wang, J.M., Haraux, A., Popovic, Z., Koltun, V.: Continuous character control with low-dimensional embeddings. ACM Trans. Graph. 31(4), 28 (2012)
15. McCann, J., Pollard, N.S.: Responsive characters from motion fragments. ACM Trans. Graph. 26(3), 6 (2007)
16. Ng, A.Y., Russell, S.: Algorithms for inverse reinforcement learning. In: ICML (2000)
17. Nguyen, M.H., la Torre, F.D.: Max-margin early event detectors. In: CVPR (2012)
18. Ormoneit, D., Sen, S.: Kernel based reinforcement learning. Machine Learning 49(2-3), 161–178 (2002)
19. Rabiner, L.R., Juang, B.H.: An introduction to hidden Markov models. ASSP Magazine (1986)
20. Ryoo, M.S., Aggarwal, J.K.: UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA) (2010), `http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html`
21. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV (2009)

22. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV (2011)
23. Safonova, A., Hodgins, J.K.: Construction and optimal search of interpolated motion graphs. ACM Trans. Graph. 26(3), 106 (2007)
24. Treuille, A., Lee, Y., Popovic, Z.: Near-optimal character animation with continuous control. ACM Trans. Graph. 26(3), 7 (2007)
25. Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D.: Two-person interaction detection using body-pose features and multiple instance learning. In: CVPRW (2012)
26. Zhang, Y., Liu, X., Chang, M.-C., Ge, W., Chen, T.: Spatio-temporal phrases for activity recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 707–721. Springer, Heidelberg (2012)
27. Ziebart, B., Maas, A., Bagnell, J., Dey, A.: Maximum entropy inverse reinforcement learning. In: AAAI (2008)