PiGraphs: Learning Interaction Snapshots from Observations

Manolis Savva* Angel X. Chang* Pat Hanrahan* Matthew Fisher* Matthias Nießner*

Stanford University

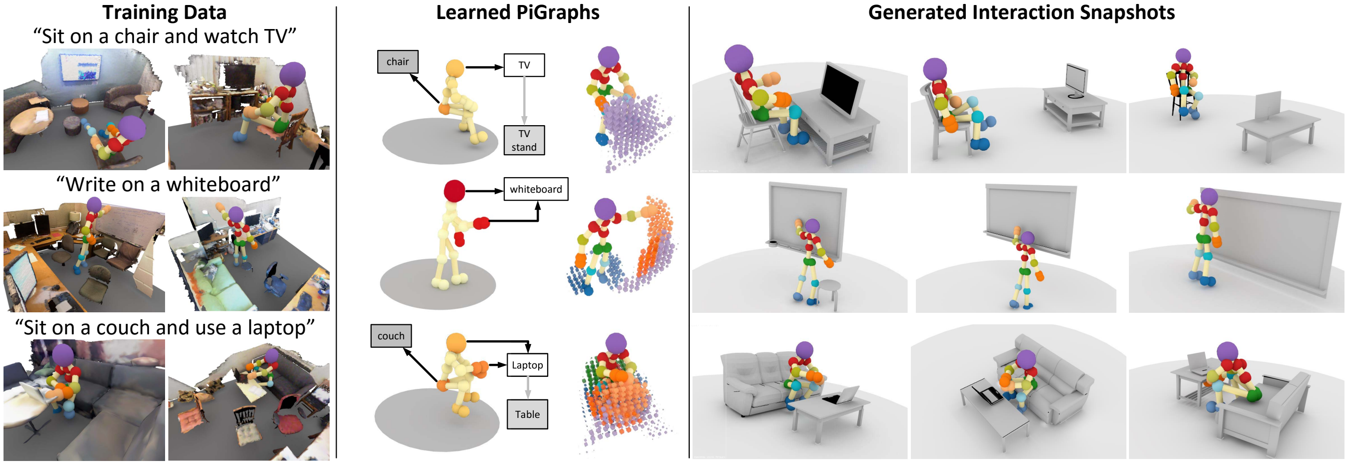
Training Data
"Sit on a chair and watch TV"
"Write on a whiteboard"
"Sit on a couch and use a laptop"
Learned PiGraphs
chair
TV
TV stand
whiteboard
couch
Laptop
Table
Generated Interaction Snapshots

Figure 1: *We use commodity RGB-D sensors to capture people in common everyday actions within reconstructed real-world indoor environments (left). Using this pose and 3D scene data we learn a set of prototypical interaction graphs (PiGraphs) containing priors on the human pose and object geometry during interaction (middle). We then generate likely poses and arrangements of objects given the action (right).*

Abstract

We learn a probabilistic model connecting human poses and arrangements of object geometry from real-world observations of interactions collected with commodity RGB-D sensors. This model is encoded as a set of *prototypical interaction graphs* (PiGraphs), a human-centric representation capturing physical contact and visual attention linkages between 3D geometry and human body parts. We use this encoding of the joint probability distribution over pose and geometry during everyday interactions to generate *interaction snapshots*, which are static depictions of human poses and relevant objects during human-object interactions. We demonstrate that our model enables a novel human-centric understanding of 3D content and allows for jointly generating 3D scenes and interaction poses given terse high-level specifications, natural language, or reconstructed real-world scene constraints.

Keywords: object semantics, human pose modeling, person–object interactions, 3D content generation

Concepts: •**Computing methodologies** → **Computer graphics; Spatial and physical reasoning;**

1 Introduction

Computer graphics has made great progress in enabling people to create visual content. However, we still face a big content creation bottleneck. In particular, designing 3D scenes and virtual character interactions within them is still a time–consuming task requiring expertise and much manual effort. A common theme in addressing the content creation challenge in various subfields of graphics has been to leverage data in order to build statistical methods for automated content generation.

In character animation, motion capture technology and an organized effort to collect human motion data—such as in the widely used CMU Motion Capture Dataset—led to much progress in data-driven animation methods. Models of human motion and learned character controllers have made it easier to design virtual character animation. However, prior work in animation typically focuses on the character motion, meaning that the surrounding environment is not modeled directly. In some cases motion capture datasets use simple props (e.g., folding chairs [Ofli et al. 2013]) or specialized setups (e.g., instrumented kitchens [De la Torre et al. 2009; Tenorth et al. 2009]). Predominantly, the configuration of the 3D scene the person is moving in is not explicitly correlated with the observed human poses, nor is it used to condition the poses.

Work in geometric analysis has followed a similar trajectory. The increasing availability of 3D object mesh data led to work in data-driven shape synthesis [Kalogerakis et al. 2012; Huang et al. 2015; Yumer et al. 2015]. The domain of scenes has also seen its share of data-driven methods in the line of work addressing scene synthesis [Yu et al. 2011; Fisher et al. 2012; Xu et al. 2014]. However, the data and models used in this work implicitly ignore the presence of people and represent scenes and objects independently of human interaction. A notable exception is some recent work for predicting human pose given 3D objects [Kim et al. 2014] and work showing that an explicit model of human presence and object affordances can improve scene synthesis [Fisher et al. 2015].

Though modeling 3D scenes and modeling human motion have been addressed extensively decoupled from one another, much of

*email: {msavva,angelx,hanrahan,mdfisher,niessner@cs.stanford.edu}
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 ACM.
SIGGRAPH '16 Technical Paper, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07 $15.00
DOI: http://dx.doi.org/10.1145/2897824.2925867

the challenge in 3D content creation lies where the two come together; i.e., when characters interact with 3D environments. There is work on specific tasks such as path planning and grasping in both graphics and robotics. One approach to looking at a more general set of interactions is to try and describe the interaction with natural language. But generating even static interaction scenarios from high–level natural language specifications such as "person reading a book while sitting on a couch" is very challenging. Many implicit constraints involving physical contact, static support and attention need to be inferred—this is true of most everyday interactions with common objects such as furniture and electronics. There has been some seminal early work in the WordsEye system of Coyne and Sproat [2001] and the definition of "Smart Objects" by Kallmann and Thalmann [1999]. A data-driven revolution such as the one seen in animation and scene synthesis research has yet to happen for generating characters interacting with 3D environments.

In this paper, we bridge the gap between human pose modeling and 3D scene synthesis. We present a data-driven approach for generating *interaction snapshots*; i.e., depictions of how people interact with arrangements of objects. We collect a dataset with observations of people performing everyday actions in reconstructed 3D rooms. From this data, we learn *prototypical interaction graphs* (*PiGraphs*), human-centric representations of interactions that link attributes of the human pose with the geometry and layout of the objects near the person. We show how PiGraphs can be used to generate interaction snapshots given high-level specifications. We evaluate the generated snapshots with ablative comparisons and a user study judging plausibility against baseline approaches.

2 Related Work

The seminal work on affordances by Gibson [1977] has provided inspiration for leveraging human-object interaction data in a variety of tasks, including the improvement of pose estimation, object recognition, action classification, and many other problems [Stark et al. 2008; Bohg et al. 2013; Koppula and Saxena 2013; Zheng et al. 2014]. We aim to connect 3D environments and human poses through a model of interactions with priors for both obtained from RGB-D observation data.

Work in computer vision and robotics has jointly modeled common human activities and interactions with objects observed in RGB-D data [Koppula et al. 2013; Wei et al. 2013a; Wei et al. 2013b]. This line of work focuses on using a model of human activities to classify objects and actions, or to predict likely sequences of actions given current observations. In contrast, we focus on generative graphics applications. Another line of work encodes human interactions from RGB-D data to hallucinate plausible human poses for labeling objects or for learning priors on the spatial distribution of objects with respect to people [Jiang et al. 2012; Jiang et al. 2013; Jiang and Saxena 2013]. Grabner et al. [2011] focus specifically on the action of "sitting" and sample scenes with posed 3D human models to infer sittable objects. Other approaches in computer vision aim to learn pose predictions from RGB video [Delaitre et al. 2012], or to determine affordances in new images based on inferred poses [Gupta et al. 2011; Fouhey et al. 2012].

Recent work uses RGB-D interaction observations to predict the likelihood of actions in 3D scenes [Savva et al. 2014]. Our goal is similar in the sense that we aim to extract human action priors from RGB-D observation data. However, we focus on jointly modeling both the human pose and arrangement of objects with which it is interacting. In contrast, Savva et al. [2014] do not model pose parameters or object arrangement. They address a discriminative action labeling problem rather than a generative task.

Fisher et al. [2015] present an activity model for improved 3D scene synthesis. The activity model is based on agent annotations containing position, orientation and action information. Along with a set of manually annotated object interactions their method learns the parameters of agent–object interaction scoring functions to evaluate how well a given 3D scene supports specified activities. Most importantly, they do not model or generate human poses.

An alternative approach for modeling 3D scene functionality is to encode object–object features as presented by ICON [Hu et al. 2015]. Though object relations are critical for functional scenes, we focus on joint analysis of human pose and object arrangements. Combining priors from these two viewpoints is an interesting direction for future work.

Interaction snapshot generation is complementary to work in animation synthesis—a broad field with much prior work. Guo et al. [2014] provide a comprehensive survey. Data-driven methods for modeling and synthesizing human motion are prevalent, with the emphasis usually being in the temporal axis. Much prior work has addressed modeling of human motion style to allow for high-level control and adaptation of motion data [Grochow et al. 2004; Shapiro 2011; Min and Chai 2012]. Some work has addressed motion for concurrent object manipulation tasks [Bai et al. 2012], though again the focus is in the temporal domain and not on complex high-level constraints on object arrangements.

Prior work in animation captured interaction sequences on "patches" representing common environment types and then synthesized these sequences into longer behaviors [Lee et al. 2006]. This work shares our goal of correlating interactions with the geometry of the environment. However, we take a complementary view, as we predict and generate interaction keyframes satisfying high-level constraints, instead of full animation sequences. More recent work computes plausible, statically supported human poses given the geometry of an environment as input [Kang and Lee 2014]. Both of these methods treat the environment as input and the poses as output. In contrast, we jointly generate poses and environments. We introduce a unified model for human-object interactions and learn both pose and object interaction priors from RGB-D data.

3 Overview

Our goal is to automatically generate 3D depictions of interactions by modeling how people interact with objects. We will encode observations gathered from the real world into probability distributions describing the human pose, nearby object categories, and their contact, support or attention linkages to the pose, conditioned on specific actions. To compactly aggregate these priors, we propose the PiGraph representation: a human-centric graph-based representation that encodes objects and body parts as nodes, and interactions between nodes as edges (see Section 5). In this section, we summarize our approach and formalize the task that we will address.

3.1 Approach

We collect real-world observations to capture gaze and body part interactions for learning the PiGraph representation. Manual specification of pose priors, object arrangements, and constraints between poses and objects is impractical: it requires an inordinate amount of effort to cover the rich set of object configurations and poses that human actions can exhibit. Furthermore, a data-driven approach can be tailored to specific domains and is straightforward to scale.

We start by range scanning real-world environments using a dense fusion approach from prior work [Nießner et al. 2013]. We then perform skeletal tracking of people as they interact with the scanned environments, using a stationary RGB-D sensor [Shotton et al.

2013]. We ask a volunteer to annotate each recorded interaction video with all time ranges where specific actions are taking place. We encode these annotations as sets of verb-noun pairs (e.g., “sit-chair”, “type-keyboard”—see Section 4). The 3D skeletal tracks provide joint positions for the pose and are registered to the coordinates of the environments, allowing us to create individual *interaction graphs* (*iGraphs*, cf. PiGraphs which are prototypes aggregating a set of iGraphs). An iGraph’s nodes correspond to human body joints or segments of geometry within the scene, and the edges are specific observed contact or gaze events (see Section 5.1).

During learning, we aggregate iGraphs with the same action annotations to generate a PiGraph (see Section 6). These PiGraphs encode the correlation between features of the geometry and the observed human pose during the given action.

We then use the learned PiGraphs to generate interaction snapshots. At a high level, we iteratively sample the probability distributions of pose and object configuration, seeking to maximize an overall interaction score. This score includes an object arrangement likelihood, a pose likelihood, and pose–object interaction likelihoods. To do this, we define a similarity metric between pairs of iGraphs, and between a PiGraph and an iGraph which we use to compute interaction scores (see Section 7.4.3). We demonstrate that using these metrics and the PiGraphs we can generate plausible interaction snapshots from terse specifications.

We also demonstrate that the priors encoded in PiGraphs have various other applications. We use a standard NLP pipeline to convert natural language input into action specifications thus creating an end-to-end “text2interaction” system. We also show that PiGraphs can be used to analyze 3D reconstructions by predicting object labels and interaction regions. With these predictions, we generate interaction snapshots constrained by the observed geometry in the reconstruction, demonstrating a novel 3D scene modeling pipeline.

3.2 Task Definition

For interaction snapshot generation, we create a *snapshot* consisting of a pose and a set of objects given an interaction as input.

More formally, our input is: (i) a corpus of 3D models with categorical label c_i for each model m_i, and (ii) an interaction A given as a set of verb-noun pairs. The output is a snapshot $IS = (J, M)$ consisting of a posed figure J, and a set of positioned objects M. We represent M as a set of model instances, with $M = \{(m_j, T_j)\}$, consisting of a model m_j and its associated transform matrix T_j.

In Section 7, we generate interaction snapshots given an interaction A as input. In Section 9, we show how interaction snapshots can be generated directly from natural language text, or constrained to match given 3D scene geometry.

In both cases, we must answer several questions:

1. How should the person be posed? We sample likely poses from a pose distribution associated with each PiGraph.
2. Where can this action occur? This is given as input or predicted with the PiGraph for an input 3D scene.
3. What objects are nearby? Once we have posed and positioned the person at a location, we use the PiGraph to predict likely object categories.
4. Which body parts are interacting with objects? How are they positioned relative to the object?
5. What is the placement and orientation of the objects? We retrieve and arrange 3D models to maximize an overall interaction snapshot score $L_A(J, M)$.

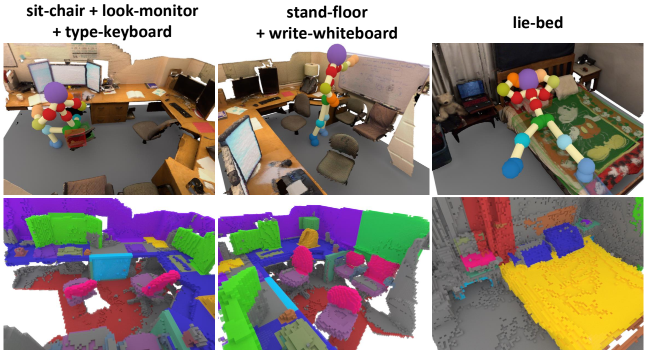

Figure 2: *Interaction observation frames from our dataset. The pose of the observed person is tracked and projected into the reconstructed scene. The scene is annotated with object labels at the voxel level (indicated by colors). Proximity of geometry to body parts is used to create iGraphs used for learning the PiGraph of each unique set of verb-noun annotations (labels at top).*

We will first describe the data collection and annotation effort, and then address the details of our representation and approach.

4 Dataset

The procedure used to collect our dataset is similar to prior work on action map prediction [Savva et al. 2014]. We first obtain a 3D reconstruction of an environment using a volumetric fusion framework [Nießner et al. 2013]. We then set up a static Kinect.v2 RGB-D camera to observe people as they interact within these environments. In these interaction sequences, we track a person’s skeleton at 30Hz using the Kinect SDK v2 framework [Shotton et al. 2013]. The reconstructions are obtained using a Structure sensor[1]. Once skeleton data was obtained, we projected all 3D joint positions to the reconstructed scene coordinates, and asked a volunteer to annotate each recorded video with all time ranges during which specific actions were taking place (a set of appropriate verb and noun labels was provided). Note that often multiple interactions are annotated over a single time period (e.g., “sit-chair + read-book”). Finally, we asked another student volunteer to annotate all objects in the 3D scene reconstructions at the part level using a predefined set of part labels (e.g., “chair:seat”, “chair:back”, “table:top”). The segmentations were manually labeled by grouping sets of segments obtained from an unsupervised normal-based segmentation [Felzenszwalb and Huttenlocher 2004], and assigning an object and part label to each group.

Our dataset is composed of 30 scenes and 63 observations. These 63 observations are video recordings of five subjects (4 male, 1 female) with skeletal tracking provided by the Kinect.v2 devices. Tracking occurs within 3D scenes that were reconstructed using the more mobile Structure sensors. The total recording duration is about two hours (100k frames at 15 Hz) with a per-recording average length of 2 minutes and an average of 4.9 action annotations. In total, there are 298 actions, and the average action duration is 8.4 s. There are 43 observed combinations of verb-noun pairs with 13 common action verbs such as look, sit, stand, lie, grasp, and read. 19 object categories are associated with these verbs (e.g., couch, bed, keyboard, monitor). As Figure 2 illustrates, the observations in our dataset are corresponded to volumetric reconstructions of each 3D scene with object annotations of all occupied voxels.

[1] http://structure.io/

Symbol	Interpretation	Type
j	Body part joint	Person
$J = \{j_i\}$	Body pose	
s	Geometric segment	Geometry
s_J	Active segment given pose J	
$S_J = \{s_J\}$	Active region given pose J	
m	3D model mesh representing an object	
$a = (v, n)$	Action tuple (verb v, applied on noun n)	Concept
$A = \{a_i\}$	Performed activity as set of actions	
$I_A = (V_A, E_A)$	Interaction graph (iGraph) for observed A	
$\widetilde{I}_A$	Prototypical interaction graph (PiGraph) of A	

Table 1: *Symbols used in our formalization.*

5 Representation

Each observed interaction is represented as an interaction graph (iGraph): a graph-based representation encoding human-object interactions (see Section 5.1), which we then aggregate into prototypical interaction graphs we refer to as PiGraphs (see Section 5.2). The PiGraph edges represent probability distributions of the spatial relationships between an interacting joint and an object, as well as the probability of the presence or absence of the interaction. The pose of the person is represented using a hierarchical joint angle encoding at each joint node (see Section 5.3).

Our goal is to have a representational model that is compositional, interpretable, and generative. See Table 1 for a summary of symbols in our formalization.

5.1 Interaction Graphs

We define an iGraph $I = (V, E)$ consisting of the node and edge sets V and E. Nodes represent either a joint of the human body j_i or a geometric segment s_i. The set of joints is the set used by the Kinect.v2, visualized in Figure 3, and also includes an abstract center of mass representing the body center, and an abstract gaze joint representing the eyes. Body joints are connected through edges representing the skeletal structure of the human body ($j - j$), and joints are associated with specific segments through *contact* or *gaze* linkage events ($j - s$).

Nodes in the graph represent regions of geometry a person is interacting with, and specific body parts of the person. Nodes are attributed with a continuous feature space representation of the properties of geometric regions or body parts. The edges are also attributed with a feature vector that takes into account relative 3D space positions and orientations of the linked nodes. We describe how to extract iGraphs from observations in Section 6.1.

This representation allows us to encode the correlation between human pose and geometry while actions are performed. The presence of an edge associating a body part with an object indicates a coupling between the two during an interaction. By looking at the frequency statistics of such couplings we can construct priors on what interactions are likely. We aggregate observations to create a PiGraph representing each action type (see Section 5.2). Figure 3 shows an example of a observed interaction and how it is aggregated to form a PiGraph.

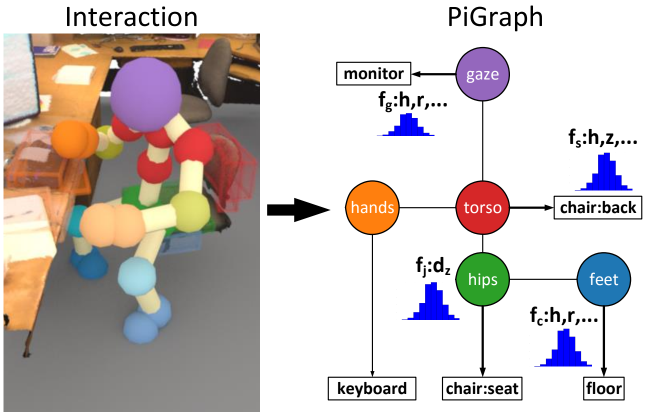

Figure 3: *Aggregation of iGraphs. Left: activated segments highlighted in boxes colored corresponding to body part. Right: features of the segments and their linkage to a body part are computed and aggregated into the nodes and edges of the PiGraph.*

Note that our goal is not to propose a new set of geometric or human pose features. Instead, we use simple features that are easily interpretable but capture important properties that are robust to noise in the reconstructions and the tracked poses. Current passive sensor technology has difficulty resolving joint positions more accurately than $\sim$ 10cm. Furthermore, we do not formulate any features based on appearance (i.e., RGB color) as our data does not sufficiently capture the large variability over visual appearance (e.g., chairs can have many different materials).

Features. We attach real-valued features on the nodes and edges of iGraphs depending on the type of node or edge:

- Activation features f_a: frequency of activation of node pairs (stored at edges) and frequency of co-activation of body part or object (at the nodes).
- Joint features f_j: height above ground h.
- Segment features f_s: centroid height above ground h_c, segment bounding box height h_s, horizontal diagonal length d_{xy}, horizontal area A_{xy}, and the dominant normal z vector (i.e., min-PCA axis)'s dot product with upwards vector.
- Contact features f_c: absolute height of contact point h, radial distance from skeletal center of mass to contact point on segment r, vertical displacement from center of mass to contact point z, angle of vector from center of mass to contact point in xy plane θ_{xy}, and the contact segment's dominant normal vector z dot product with direction of contact.
- Gaze features f_g: same as contact features, except reference point is head location instead of center of mass.

5.2 Prototypical Interaction Graphs

After encoding observed interactions as iGraphs, we aggregate them into a *prototypical interaction graph* (PiGraph) for each action (Section 6.2). Note that the set of action verbs associated with each observation defines each class of PiGraph (e.g., sit prototype, stand prototype, sit+type prototype, stand+type prototype).

PiGraphs aggregate the active geometry, the connectivity of geometry with pose, and the assigned noun references for each action. The nodes contain distributions over the segment features and nouns. The edges contain a connection probability (i.e., activation probability) and PDFs over the attributes of the connection.

5.3 Pose Representation

We use a hierarchical joint angle encoding to represent poses. The input skeletal tracking data consists of the positions and orientations of 25 joints which we encode as quaternions and positions relative to parent joints in the kinematic chain. In addition, we encode the global vertical orientation of the skeleton with respect to the up vector. This is the same approach as used by Grochow et al. [2004].We then convert the joint orientation quaternions into latitude, longitude, and roll angles for which we fit von Mises distributions [Mardia and Jupp 2009]. We also fit a normal distribution to each bone length, normalized by total bone length for stability across individuals. The von Mises distributions (for orientation angles) and Gaussian distribution (for bone length ratio) at each joint form a total pose distribution under which we evaluate the likelihood of a given pose. We use this distribution to sample for likely poses during snapshot generation. This representation allows us to use a sampling scheme with one degree of freedom per joint (roll angle around mean rotation axis) as we discuss in Section 7.3.

6 Learning from Observations

In order to learn a set of priors connecting features of the human pose and geometry, we first extract iGraphs for each observed frame of an interaction recording with an action annotation. Then, we aggregate these iGraphs into a PiGraph for each unique set of verb-noun pairs. Finally, we learn joint weights that indicate the importance of each joint for a particular action.

6.1 Extraction of Interaction Graphs

Given an action observation A with a person in a pose J, we extract from the scene a set of active geometry segments s_J. These segments are activated either by contact (a joint of the pose is in close proximity), or by gaze (the geometry is within the view cone).

Contact Activation. For each joint $j \in J$, we perform a nearest neighbor lookup within a radius r_{act} to find vertices of the reconstructed scene mesh (we use $r_{act} = 10\,\mathrm{cm}$ for all results in this paper). The segment s_i with the closest vertex within this threshold is taken as an active segment for the specific joint j_i.

Gaze Activation. We estimate the gaze direction from a skeletal pose J by a least squares plane fit to the body joints within the torso (shoulders, spine, hips). We then take the two possible normal vectors and vote on the "front" orientation by counting the number of mobile joints (hands, elbows, knees, feet) that have a positive dot product with each direction. The direction which gives more positive dot products is chosen as front. We then randomly cast $N = 200$ gaze rays from the center of the head in a 45° view cone. For segments intersected within a maximum distance of $2\,\mathrm{m}$, we accumulate a distance-weighted intersection count. This is the product of ray intersection ratio (intersection count normalized by total intersections), and the average distance to intersection (normalized by maximum intersection distance). We sort segments by this weight, and mark the top three as activated by gaze. This approach can have false positives when big vertical surfaces such as walls are behind smaller surfaces of attention such as monitors. A more robust approach could assign weights to the rays corresponding to the salience of the given gaze direction, potentially conditioned on the parameters of the observed pose.

Graph Construction. Once we have the set of active segments $\{s_i\}$, we can create an iGraph for the observed action A. Each joint j_i in the pose J is represented by a node populated with joint features. Joint nodes are connected by edges representing the structure of the human pose (i.e., bones between joint pairs) which contain the relative pose features (such as vertical distance between the joints in the pair). The active segments $\{s_i\}$ are connected to joints with which they maintain contact or gaze associations. Segment nodes are populated with segment features, and the contact and gaze edges contain contact and gaze features respectively. Figure 3 shows an example observation and corresponding PiGraph.

6.2 Aggregating iGraphs into PiGraphs

A PiGraph $\widetilde{I}_A$ captures the connectivity (activation) frequencies and distributions over the features observed at the nodes and edges of all iGraphs I_A in a given action set A. In other words, $\widetilde{I}_A$ forms a joint probability distribution summarizing the observed I_A and their features. The process of constructing the PiGraph leverages the fixed structure of the joints and bones across all I_A in order to correspond the observed active segments and their connections.

Initialization. We instantiate a new $\widetilde{I}_A$ by constructing a graph containing just the joint nodes j_i and corresponding bone edges. All edges are initialized with a frequency count which aggregates the number of observations I_A for which that edge holds. Both edges and nodes are initialized with an empty set of histograms over the features that they contain as members of iGraph I_A. Nodes and edges that are members of the pose (i.e., joint nodes and bone edges) contain a single histogram. Nodes and edges that represent events and geometry are linked to a set of histograms conditioned on the noun n of the active geometry segment which is observed. This set of histograms conditioned on n will represent the conditional distribution over the segment features and contact or gaze features for each type of segment n. For generality we used histograms as a non-parametric representation, though other choices such as kernel density estimation or fitting parametric models can better capture specific distributions.

Aggregation. For each iGraph I_A within an action set A, we aggregate the observed features into the PiGraph $\widetilde{I}_A$ as follows. The features of each joint node in I_A are added into a feature histogram at the corresponding joint node in $\widetilde{I}_A$. Likewise, the features over each bone edge in I_A are added to a histogram in the corresponding edge in $\widetilde{I}_A$. In the case of contact or gaze linkage nodes and edges in I_A, we aggregate the observed features in the appropriate histogram under the segment label n.

Joint Weights. We define a per-joint weight by taking the conditional probability of a joint being linked to geometry segments corresponding to a verb target (see Figure 4). These weights reflect strong correlations between different verb-joint pairs such as "stand" and feet, "sit" and hips, and "look" and gaze. Figure 5 shows the differences in joint weights for different sit poses. For instance, the model learns that "sit chair" and "sit couch" will activate the back, while "sit-stool" and "sit-bed" typically do not.

While this method works for well-represented actions, it can fail due to insufficient observations. For instance, we were unable to learn any weights for rare actions such as "grasp" and "switch". The poor sensor resolution at the scales of these interactions made it difficult to detect the correct interacting object.

6.3 Encoding Human Pose Distributions

As described in Section 5, we represent poses as von Mises and Gaussian distributions at each joint. Each joint orientation is defined relative to its kinematic parent joint thus implicitly encoding relations along the kinematic chain. Because the dependency between child and parent joint is captured by the parameterization, we

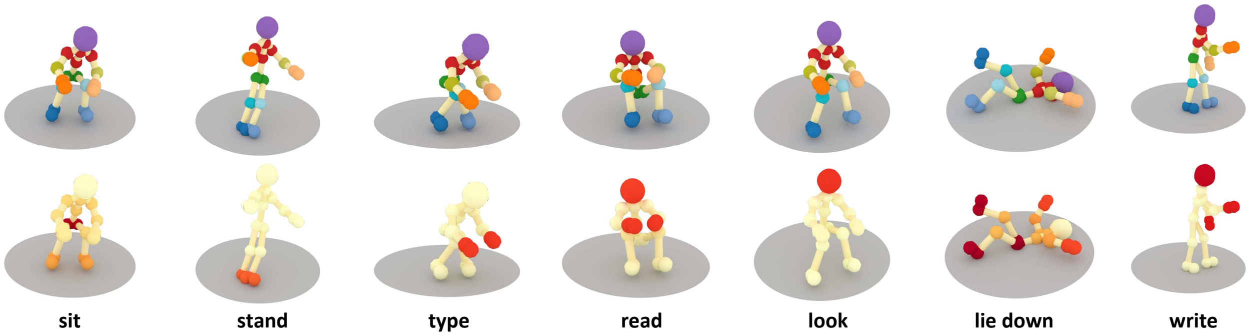

Figure 4: *Top: maximum likelihood poses for aggregated skeleton distributions of some action verbs. Bottom: conditional probabilities of body part interaction with objects during each action (indicated as red saturation). Parts critical to each action have high interaction probability. The somewhat atypical "write" pose is due to all our observations of writing being "writing on whiteboard" interactions.*

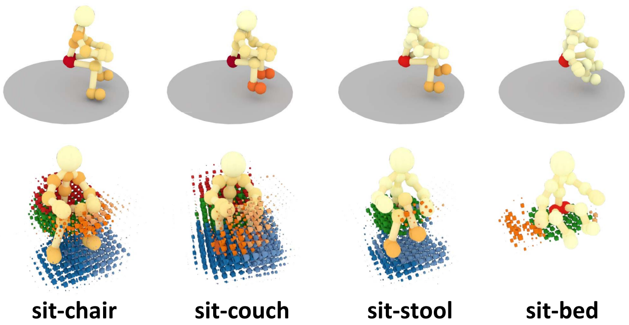

Figure 5: *Comparison of body part interaction weights and interaction volume priors for sitting on different types of objects. Top: torso interaction weight decreases from left to right while hip weight is high overall. Bottom: height and density of the hand interaction (orange voxels) shifts due to the different object categories.*

assume that joints are independent of one another during learning. This allows us to learn the distribution of each joint independently.

Parameter estimation, evaluation, and sampling for von Mises and Gaussian distributions are straightforward to do analytically. The exception is estimating the concentration κ for von Mises distributions, which we approximate using the approach of Sra [2012].

Though this model may not capture the full richness of general human motion, it is appropriate for our data of common, largely static activities exhibiting little motion around a mean pose. Much prior work has looked into more advanced models that can better capture a broader range of motions.

After learning a set of PiGraphs from our dataset, we can address the interaction snapshot generation algorithm.

7 Generating Interaction Snapshots

Figure 6 provides a summary of the process used to generate interaction snapshots given a set of action specifications. First, the input specifications are used to retrieve the corresponding PiGraph. The set of objects is expanded using the PiGraph and then their contact and gaze linkage with the pose and their support relation with each other are inferred. Then, a set of 3D models with category labels matching the objects is retrieved, and a pose is sampled from the distribution of the PiGraph. The interaction volume priors of the action verbs composing the active PiGraph are now used along with the pose distribution to iteratively score object placements and poses, and optimize for the joint likelihood of the pose and object arrangement under the PiGraph.

We follow a simple sampling approach at each step, alternating between sampling a pose with the object arrangement fixed, and sampling new object placements with the pose fixed. At each step the pose and object arrangement likelihood are scored using the pose distribution and object priors of the PiGraph correspondingly. Models are retrieved and placed in order of support hierarchy from supporting objects to supported objects, and from largest to smallest. We then sample parameters for object support surfaces, positions, orientations, and pose joint angles. Each sample is scored according to the distributions encoded in the PiGraph nodes and edges.

Since our interaction snapshots consist of a few key objects, a simple sampling approach can give reasonable results. For more complex scenes, advanced sampling techniques such as Hamiltonian Monte Carlo or other varieties of Monte Carlo sampling would be helpful. With such approaches we can handle more complex joint distributions, which is by itself a challenging research problem.

We describe the snapshot generation steps in the following sections.

7.1 Interaction Graph and Support Prediction

To infer an iGraph, we take the set of verb-noun tuples and determine the set of objects and their interacting joints. In addition, we infer a support hierarchy for the objects, and a supporting object for the person. The iGraph provides the basic constraints on the objects that need to be present, and the positioning of objects with respect to the person, while the support hierarchy provides constraints on the placement of objects with respect to each other. The support hierarchy also provides a natural ordering for placing objects (supporting parents are placed before supported children objects).

Interaction Links. Given the PiGraph $\tilde{I}_A$, we predict likely interacting object categories by looking at the probability $P_{obs}(c|j)$ that an object of category c is interacting with joint j. We estimate $P_{obs}(c|j) = w_c$, where w_c is the fraction of time a segment with label c is observed interacting with joint j for the given action A.

We make the following simplifying assumptions: 1) each joint is interacting with at most one object, and 2) there is at most one object of each category in the interaction. The first assumption allows us to select just one object per joint and we choose the most likely category. The second assumption allows us to avoid issues of coreference (i.e., person's arm and hip both interacting with a chair

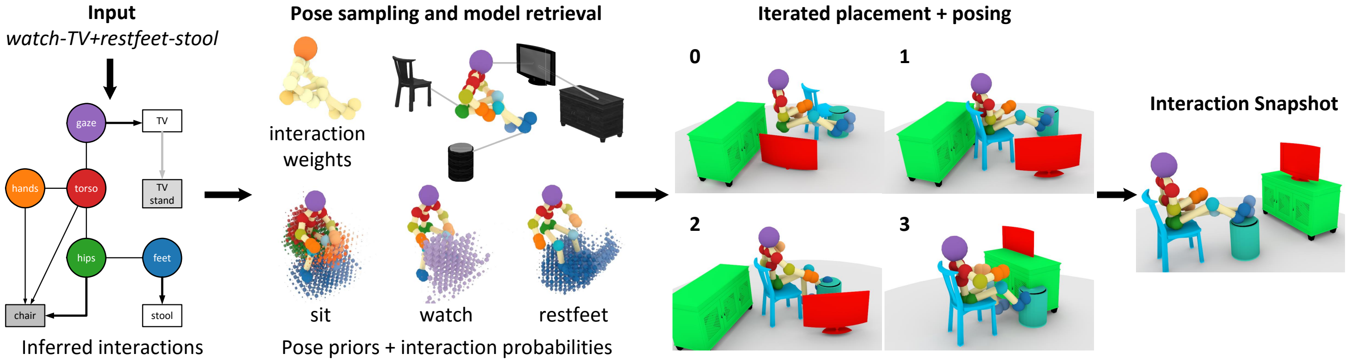

Figure 6: *Overview of pipeline for generating interaction snapshots. Left: inference on the input specifications expands the set of objects and relations. Middle: pose, 3D models and interaction priors are retrieved from a matching PiGraph. Right: iterative re-posing and object arrangement to optimize likelihood under PiGraph and generate final interaction snapshot.*

could mean two chairs or one chair). We also filter out categories with probability less than $\tau = 0.05$ to avoid spurious interactions.

Support Hierarchy. We use a static support hierarchy for placing objects. Following Fisher et al. [2012], we obtain a set of support priors $P(c_p|c_c)$ between object categories from the synthetic 3D scene data in that work. Using these support priors, we create a support hierarchy by identifying the most likely support parent category c_p^*. We also identify the support object for the pose and objects supported by the pose. We classify the pose as "standing" or "not standing" by assuming that a vertical displacement between knees and hips greater than half the average calf length is an indicator for standing. If the person is "not standing", we use the object interacting with the hip as the support object.

While building the support hierarchy, we may encounter object categories, c_p^*, not originally identified as part of the interaction. For each such c_p^*, we add it to the set of interacting object categories and retrieve appropriate models. If we cannot retrieve a parent support object, but the object is normally supported, we assume that it is supported by the interacting joint (e.g., books held for reading).

7.2 Model Retrieval and Object Placement

Once we identify the set of objects that are present, we select appropriate 3D models to represent the objects. For an object with category c, we retrieve a random model matching the category c. We use the categorized 3D model database of Fisher et al. [2012]. We directly retrieve models by category. A more sophisticated approach could use the learned interaction priors to help retrieve more relevant models through size and interaction surface area heuristics.

Placement order. The object supporting the skeleton is placed first at the same position as the skeleton, and oriented so it faces the same direction as the skeleton. Other objects are placed by traversing the support hierarchy upwards.

Sampling. To identify good placement candidates for an object, we sample the upwards oriented horizontal support surfaces on the support parent and maximize the probability given the distance of the point to the interacting joints. These objects are oriented by having them face towards the person.

7.3 Generating Poses

We sample likely poses from the pose distribution associated with a given PiGraph. To reduce the parameter space we always take the mean bone length and mean latitude–longitude angles. The former has the effect of generating a person of average height, while the latter fixes the rotation axis of each joint orientation to the mean orientation axis of that joint in spherical coordinates. The remaining roll angle can be sampled to determine the full joint orientation. This simplification fixes all joints to one degree of freedom which is a strong restriction. However, this is a simple approach for exploring a high-likelihood region of the pose distribution and sufficient for our purposes. Estimating and sampling of more powerful models such as the Scaled Gaussian Process Latent Variable Model presented by Grochow et al. [2004] is a natural extension to the simple approach we have taken.

Note that we do not use any inverse kinematics or static support reasoning to refine the pose. Sampling pose parameters from the PiGraphs directly gives the results. This direct sampling captures representative poses and variations that obey key contact and gaze constraints for each action. Combining this sampling approach with pose stability and IK methods can improve the quality of the results. Our focus in this work is to show that combining priors on object arrangement and pose is a surprisingly effective yet simple approach for encoding common human poses.

7.4 Scoring

The overall score $L_A(J, M)$ of an interaction snapshot is the combination of the pose, object, and interaction scores. We compute $L_A(J, M)$ as the weighted sum of pose score $L_{p_A}(J)$, object placement score $L_o(M)$ and an interaction score $L_{i_A}(J, M)$:

$$L_A(J, M) = w_p L_{p_A}(J) + w_o L_o(M) + w_i L_{i_A}(J, M)$$

7.4.1 Pose Score

We compute the pose score $L_{p_A}(J)$ as the sum of log probabilities for each joint orientation and a self collision avoidance energy term:

$$L_{p_A}(J) = \sum_i V_i(j_i) - C(J) \ ,$$

where V_i is the log likelihood function for the von Mises distribution at joint j_i and $C(J)$ is a $[0, 1]$ normalized measure of self collision evaluated by point sampling the oriented bounding box of each bone and checking how many points are contained by other bones (a cross section radius of 10 cm is used for all bones except the torso which has a radius of 15 cm).

7.4.2 Object Placement Score

Given the constraints of the joint interactions and the simplicity of the interaction snapshots, we use a simple object placement energy term that penalizes collisions and rewards support:

$$L_o(M) = \sum_{m_i, m_j, j \neq i} (1 - C(m_i, m_j))$$

It is possible to use a more advanced object placement score that takes into account the likely position of objects which may be useful for more complex interactions with many objects.

7.4.3 Interaction Score

We define the interaction score L_{iA} to be the similarity of a given iGraph I corresponding to the pose J and object configuration M, with the PiGraph $\tilde{I}_A$:

$$L_{iA}(J, M) = sim(I, \tilde{I}_A)$$

To define $sim(I, \tilde{I}_A)$, we first define the similarity between two iGraphs I and I' as a weighted combination of a gaze similarity sim_{gaze} and an overall contact similarity sim_{acon}:

$$sim(I, I') = w_{gaze} sim_{gaze} + w_{acon} sim_{acon}$$

The gaze similarity includes the similarity between the gazed segment features, and the gaze edge features: $sim_{gaze} = \max_{(s,s')}(sim(f_s, f'_s) \times sim(f_e, f'_e))$ where s is a gazed segment in I and e is the gaze edge for s. For node and edge feature similarity we use the $[0, 1]$ angular similarity of the feature vectors.

Similarly, we define for each joint j a per-joint contact similarity $sim_{con_j} = \max_{(s,s')}(sim(f_s, f'_s) \times sim(f_e, f'_e))$ where s is now a contacted segment and e is the contact edge. We aggregate across all joints with contacts to obtain the aggregated $sim_{acon} = \sum_{j \in J} sim_{con_j}$.

We set the contact and gaze weights to be equal, though tuning them for specific prediction or classification tasks is an interesting direction to explore in future work.

Similarity of PiGraphs and iGraphs. In order to compute the similarity of an iGraph I_{J,S_J} to a PiGraph $\tilde{I}_A$, we use the probability that a feature vector f from I_{J,S_J} is drawn from an aggregated histogram $hist$ of $\tilde{I}_A$: $sim_{hist}(f) = P_{hist}(f)$.

8 Results

We start by visualizing several aspects of the priors encoded in the PiGraph representation. This allows us to verify empirically that common sense facts are captured by the representation and is also a good sanity check decoupled from the quality of interaction snapshots. We then quantitatively evaluate the results of interaction snapshot generation with a human judgment study.

8.1 Visualizing Interactions

The learned PiGraphs can be used to evaluate the likelihood of new interaction graphs under the observed distributions for a given action A. The priors on the expected geometry of interacting objects, and their linkage to the human pose are encoded in normalized histograms such as the examples plotted in Figure 7.

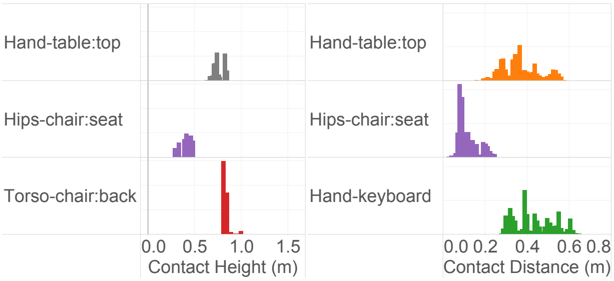

Figure 7: *Normalized histograms of contact height and contact radial distance from pose center of mass for several joint-object pairs. Note the natural ordering of chair seat, table top, chair back in ascending contact height, and the ordering of chair seat, table top, keyboard in increasing radial distance.*

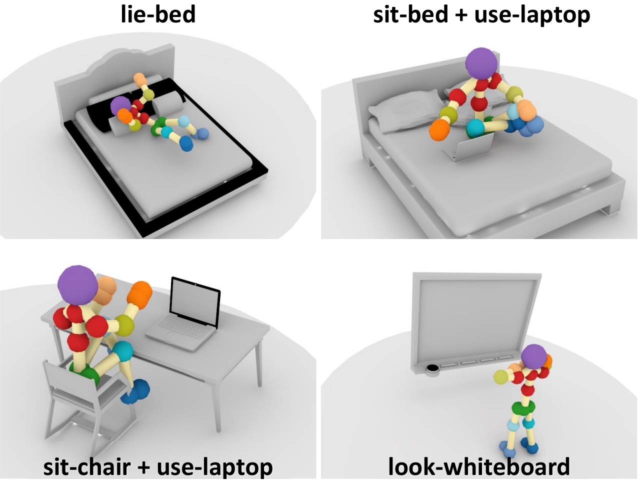

Figure 8: *Interaction snapshots generated by sampling PiGraphs learned from various action observations.*

In addition to these histogram priors, we can also visualize priors on the occupancy of space surrounding the pose of a person during an interaction by projecting all observations of interacting object surfaces in the pose-centric coordinate frame and aggregating counts into a dense voxel grid. Figure 9 shows several examples of these "interaction volume" constructions which capture common sense facts involving body part contact and attention.

8.2 Generated Interaction Snapshots

Figure 8 shows several examples of generated interaction snapshots. The input to each of these is the indicated set of verb-noun pairs. Overall, the generated interaction snapshots capture many relations between the human pose and objects that would have to be manually specified: positioning of hips on the sittable surfaces of chairs and beds, and the orientation and limb configuration of the pose for looking at and using laptops and whiteboards. More examples of interaction snapshots are provided in Figure 1.

8.3 Evaluation

To quantitatively evaluate the generated interaction snapshots, we perform a human judgment study. We establish a baseline condition

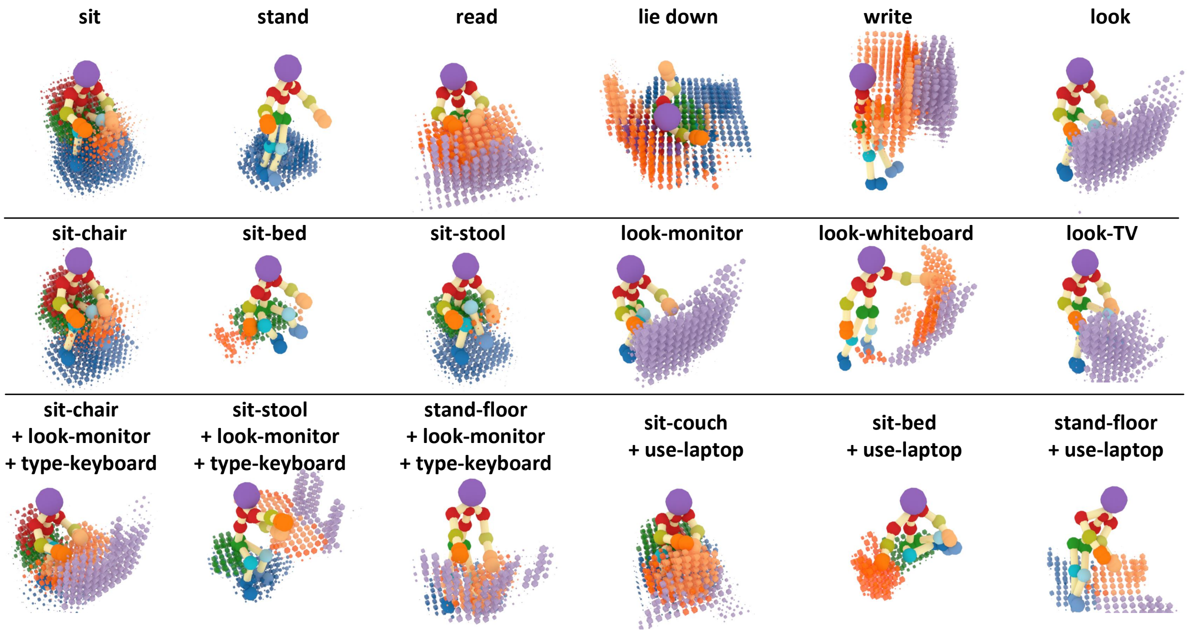

Figure 9: *PiGraph interaction volumes for some actions. The voxels around the pose are colored corresponding to body part, and their size is determined by the probability of a contact or gaze linkage occurring with the given body part. These priors summarize intuitive but rarely stated and represented facts about human pose–geometry coupling. For example: looking means gazing at geometry in front of one's head; when sitting in chairs there is usually a backrest part behind the torso, in contrast to sitting in bed and sitting on a stool.*

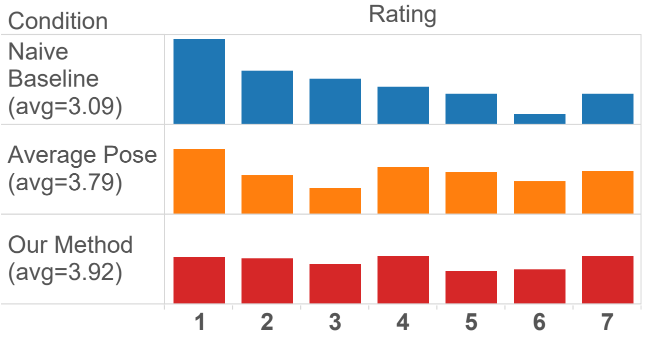

Figure 10: *Rating distributions for the interaction snapshot quality study (higher is better, 2072 judgments across conditions). The naive baseline performs worst as expected. Using the average pose instead of sampling and scoring poses results in a lower perceived quality. This illustrates the benefit of the pose priors.*

by using the PiGraph to only predict the objects and their support hierarchy. We then position these objects independent of the pose. We also tested a simplified version of our approach which always uses the average pose for each PiGraph and thus completely skips pose sampling. For each of these conditions, and the full method, we generated 5 interaction snapshots from 37 PiGraphs learned using our training dataset.

We then recruited 75 participants on Amazon Mechanical Turk to judge the quality of the interaction snapshots. Participants were presented with screenshots of interaction snapshots chosen at random from among the conditions, along with the input verb-noun action specification. They were asked to provide a rating on a 1-7 Likert scale indicating how well the scene matched the given description (1 for "very poorly" and 7 for "very well"). Participants were instructed to focus on whether the scene depicts the specified interactions, whether all stated objects are present, and whether there are artifacts such as collisions or unusual object positions.

We collected 2072 judgments for the 555 stimuli screenshots (37 PiGraphs, 5 snapshots, 3 conditions, 3.7 ratings per stimulus on average). The rating distributions for each condition are plotted in Figure 10. As expected, the naive baseline has the lowest score (average of 3.09), the simplified method with average poses is higher (3.79), and the full method has the highest average rating (3.92).

8.4 Retargeting for Novel Interactions

Using the learned PiGraphs we can generate novel interactions for verb-noun pairs not observed in the training data (see Figure 11). To generate these, we aggregate the PiGraphs corresponding to all possible verb targets. We then take the joint probabilities and features associated with these that were previously observed, and use them with the new verb noun pair.

8.5 Limitations

Though the generated snapshots produce good results for a variety of input actions, our approach has important limitations and failure cases. Our sampling scheme fails to sample good configurations when the initial parameters are not ideal. For example, if the first object in an interaction (e.g., a table) is badly positioned behind the pose, recovery is difficult. More advanced sampling methods can help mitigate this limitation. The most significant limitation of our approach is the restriction to static scenes.

We have chosen not to model any of the dynamics or temporal sequencing of interactions primarily due to the difficulty of tracking people interacting with dynamic scene geometry. The quality of the reconstructed geometry and tracked poses is limited by the resolution of the commodity range sensors we used. This makes it difficult to address interactions with smaller objects below the sensor resolution, such as with mugs or pens. Improvements in sensor technology will help to alleviate this problem. Finally, our current implementation uses simple geometric features and makes simpli-

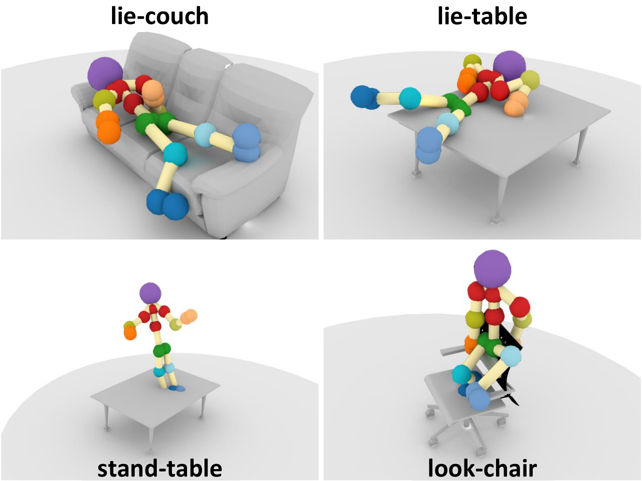

Figure 11: *Interaction snapshots for novel verb-noun combinations that were not observed in the training data. These snapshots show how a PiGraph aggregated for a given action verb can be sampled for posing against novel object categories. The bottom right example illustrates a failure case where the attempt to "look" at the chair clashes with the more common action of sitting on the chair.*

fying assumptions for building the pose and object configuration priors. We believe that more powerful models requiring fewer assumptions and using more robust geometric features will improve the quality of the learned interaction priors.

9 Applications

9.1 Text2Interaction

We can use our method to generate interaction snapshots directly from text. Using the Stanford CoreNLP pipeline [Manning et al. 2014], we analyze the input text, extract verb-noun pairs, and provide them as input to our method. This is a rudimentary approach that does not handle synonymy and complex patterns—more advanced NLP techniques can be used to map a sentence to a set of canonicalized verb-noun pairs. Alternatively, with more data we can learn PiGraphs of finer granularity that can correspond to different verbs (e.g., subtle variations of sit such as "lounge"). Figure 12 shows example natural language descriptions and the corresponding generated interaction snapshots. Our method can be incorporated into text-to-scene systems such as WordsEye [Coyne and Sproat 2001] or Chang et al. [2014]'s system for automatic generation of interactions without the need for manually defined object interaction annotation tags, which require significant manual effort.

9.2 3D Scene Analysis with Constrained Snapshots

The PiGraph representation provides a novel way to analyze 3D reconstructions of real-world environments. By densely sampling positions in the scene and evaluating the support for the PiGraph at each position, we can find a high likelihood location for the given interaction. Examples are shown in Figure 13 as saturated green heatmaps over the scene geometry. After the position is determined, we anchor the PiGraph at the maximal likelihood position and use the object arrangement priors to predict the identity of nearby voxels (cf. ground truth voxel annotations). The predicted voxels are then used as an additional term in the interaction snapshot generation, constraining the resulting object arrangement to overlap with the scanned scene. Modeling interactions is a promising avenue for research in scene reconstruction and 3D model retrieval.

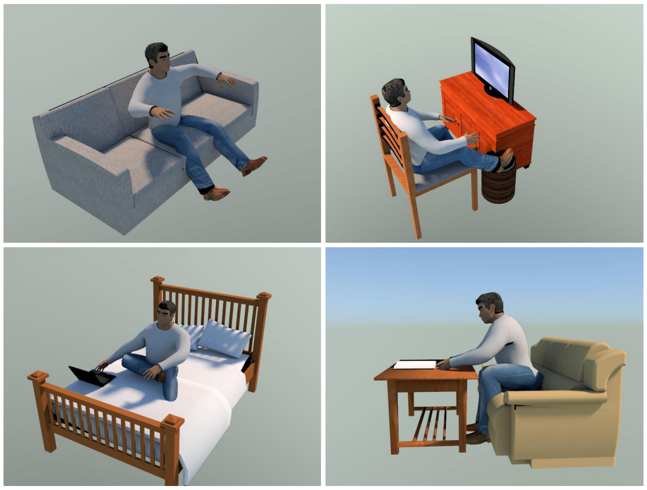

Figure 12: *Output scenes with posed virtual characters for natural language text input: "He is sitting in the couch" (top left), "He is watching TV and resting his feet on a stool" (top right), "He is sitting in bed and using a laptop" (bottom left), and "He is sitting on a couch and reading a book". The text is parsed into verb-noun pairs to generate interaction snapshots with approriate PiGraphs.*

10 Conclusion

We introduced the PiGraph representation to connect geometry and human poses during static interactions. We showed that PiGraphs can be learned from RGB-D data and used to generate a variety of interaction snapshots. There are many use cases for interaction snapshots. For instance, using snapshots as automatically generated keyframes for storyboarding animations, or as efficient high-level primitives in 3D scene modeling, or for augmented reality interfaces driven by natural language.

We presented a new framework for jointly modeling human pose and object arrangements during common interactions. Our method offers a novel view of geometry through the lens of interactions. Our hope is that by augmenting geometry with probabilistic models of human interaction, such as the PiGraphs, we can help to answer the fundamental questions of "where", "what", "how" and "when" that are necessary for 3D scene understanding. In our results, we demonstrated that interaction snapshots can be generated for both unconstrained scenarios, and for matching given scene geometry. We believe that interaction snapshots can form a building block for automating more advanced models of human interactions. An interesting line for future work would be in combining multiple interaction snapshots for animation by supporting temporal sequencing and multiple agents in 3D environments.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of Titan GPUs used for this research. This work is partially supported by NSF Award IIS-1514268. Thanks to Katherine Breeden, Angela Dai, and the anonymous reviewers for their valuable comments and suggestions for improving this paper.

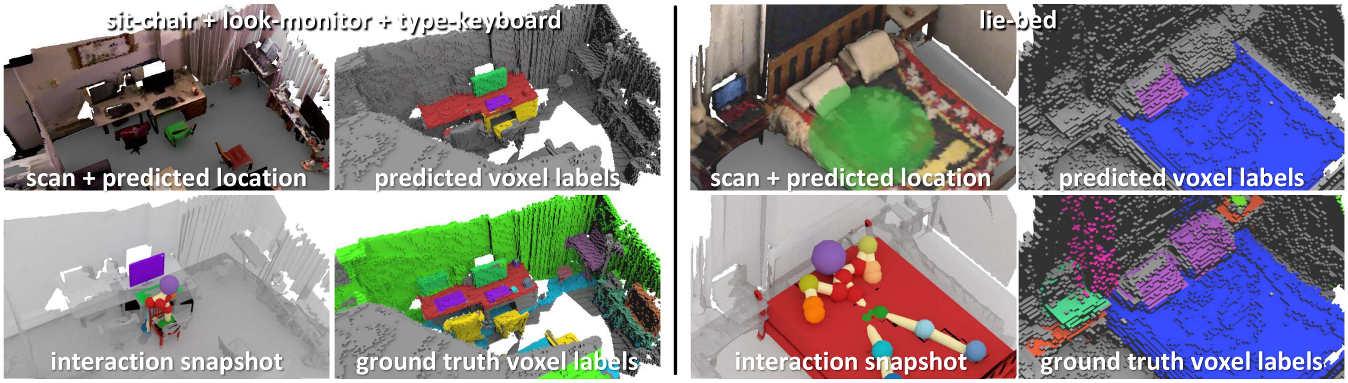
sit-chair + look-monitor + type-keyboard
scan + predicted location
predicted voxel labels
interaction snapshot
ground truth voxel labels
lie-bed
scan + predicted location
predicted voxel labels
interaction snapshot
ground truth voxel labels

Figure 13: *Two PiGraphs used to analyze reconstructed 3D scenes, label voxels, and generate interaction snapshots constrained to match regions of the scene that can support the specified interaction with high likelihood.*

References

BAI, Y., SIU, K., AND LIU, C. K. 2012. Synthesis of concurrent object manipulation tasks. *ACM Trans. Graph. 31*, 6, 156.

BOHG, J., MORALES, A., ASFOUR, T., AND KRAGIC, D. 2013. Data-driven grasp synthesis—a survey.

CHANG, A. X., SAVVA, M., AND MANNING, C. D. 2014. Learning spatial knowledge for text to 3D scene generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

COYNE, B., AND SPROAT, R. 2001. WordsEye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.

DE LA TORRE, F., HODGINS, J., MONTANO, J., VALCARCEL, S., AND MACEY, J. 2009. Guide to the Carnegie Mellon university multimodal activity (CMU-MMAC) database. *Robotics Institute, Carnegie Mellon University*.

DELAITRE, V., FOUHEY, D. F., LAPTEV, I., SIVIC, J., GUPTA, A., AND EFROS, A. A. 2012. Scene semantics from long-term observation of people. In *ECCV*.

FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. 2004. Efficient graph-based image segmentation. *IJCV*.

FISHER, M., RITCHIE, D., SAVVA, M., FUNKHOUSER, T., AND HANRAHAN, P. 2012. Example-based synthesis of 3D object arrangements. In *ACM TOG*.

FISHER, M., SAVVA, M., LI, Y., HANRAHAN, P., AND NIESSNER, M. 2015. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Transactions on Graphics (TOG) 34*, 6, 179.

FOUHEY, D. F., DELAITRE, V., GUPTA, A., EFROS, A. A., LAPTEV, I., AND SIVIC, J. 2012. People watching: Human actions as a cue for single view geometry. In *ECCV*.

GIBSON, J. 1977. The concept of affordances. *Perceiving, acting, and knowing*.

GRABNER, H., GALL, J., AND VAN GOOL, L. 2011. What makes a chair a chair? In *CVPR*.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIĆ, Z. 2004. Style-based inverse kinematics. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 522–531.

GUO, S., SOUTHERN, R., CHANG, J., GREER, D., AND ZHANG, J. J. 2014. Adaptive motion synthesis for virtual characters: a survey. *The Visual Computer*, 1–16.

GUPTA, A., SATKIN, S., EFROS, A. A., AND HEBERT, M. 2011. From 3D scene geometry to human workspace. In *CVPR*.

HU, R., ZHU, C., VAN KAICK, O., LIU, L., SHAMIR, A., AND ZHANG, H. 2015. Interaction context (icon): Towards a geometric functionality descriptor. *ACM Trans. Graph. 34*, 4 (July), 83:1–83:12.

HUANG, H., KALOGERAKIS, E., AND MARLIN, B. 2015. Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces. *Computer Graphics Forum 34*, 5.

JIANG, Y., AND SAXENA, A. 2013. Infinite latent conditional random fields for modeling environments through humans. In *RSS*.

JIANG, Y., LIM, M., AND SAXENA, A. 2012. Learning object arrangements in 3D scenes using human context. *arXiv preprint arXiv:1206.6462*.

JIANG, Y., KOPPULA, H., AND SAXENA, A. 2013. Hallucinated humans as the hidden context for labeling 3D scenes. In *CVPR*.

KALLMANN, M., AND THALMANN, D. 1999. *Modeling objects for interaction tasks*. Springer.

KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model of component-based shape synthesis. *ACM Transactions on Graphics 31*, 4.

KANG, C., AND LEE, S.-H. 2014. Environment-adaptive contact poses for virtual characters. In *Computer Graphics Forum*, vol. 33, Wiley Online Library, 1–10.

KIM, V. G., CHAUDHURI, S., GUIBAS, L., AND FUNKHOUSER, T. 2014. Shape2Pose: Human-centric shape analysis. *ACM TOG*.

KOPPULA, H. S., AND SAXENA, A. 2013. Anticipating human activities using object affordances for reactive robotic response. *RSS*.

KOPPULA, H., GUPTA, R., AND SAXENA, A. 2013. Learning human activities and object affordances from RGB-D videos. *IJRR*.

LEE, K. H., CHOI, M. G., AND LEE, J. 2006. Motion patches: building blocks for virtual environments annotated with motion data. In *ACM Transactions on Graphics (TOG)*, vol. 25, ACM, 898–906.

MANNING, C. D., SURDEANU, M., BAUER, J., FINKEL, J., BETHARD, S. J., AND MCCLOSKY, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL): System Demonstrations*.

MARDIA, K. V., AND JUPP, P. E. 2009. *Directional statistics*, vol. 494. John Wiley & Sons.

MIN, J., AND CHAI, J. 2012. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics (TOG) 31*, 6, 153.

NIESSNER, M., ZOLLHÖFER, M., IZADI, S., AND STAMMINGER, M. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM TOG*.

OFLI, F., CHAUDHRY, R., KURILLO, G., VIDAL, R., AND BAJCSY, R. 2013. Berkeley MHAD: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, IEEE, 53–60.

SAVVA, M., CHANG, A. X., HANRAHAN, P., FISHER, M., AND NIESSNER, M. 2014. SceneGrok: Inferring action maps in 3D environments. *ACM TOG*.

SHAPIRO, A. 2011. Building a character animation system. In *Motion in Games*. Springer, 98–109.

SHOTTON, J., SHARP, T., KIPMAN, A., FITZGIBBON, A., FINOCCHIO, M., BLAKE, A., COOK, M., AND MOORE, R. 2013. Real-time human pose recognition in parts from single depth images. *CACM*.

SRA, S. 2012. A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of $I_s(x)$. *Computational Statistics 27*, 1, 177–190.

STARK, M., LIES, P., ZILLICH, M., WYATT, J., AND SCHIELE, B. 2008. Functional object class detection based on learned affordance cues. In *Computer Vision Systems*.

TENORTH, M., BANDOUCH, J., AND BEETZ, M. 2009. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, IEEE, 1089–1096.

WEI, P., ZHAO, Y., ZHENG, N., AND ZHU, S.-C. 2013. Modeling 4D human-object interactions for event and object recognition. In *ICCV*.

WEI, P., ZHENG, N., ZHAO, Y., AND ZHU, S.-C. 2013. Concurrent action detection with structural prediction. In *ICCV*.

XU, K., MA, R., ZHANG, H., ZHU, C., SHAMIR, A., COHEN-OR, D., AND HUANG, H. 2014. Organizing heterogeneous scene collection through contextual focal points. *ACM TOG*.

YU, L.-F., YEUNG, S. K., TANG, C.-K., TERZOPOULOS, D., CHAN, T. F., AND OSHER, S. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics 30*, 4, 86.

YUMER, M. E., CHAUDHURI, S., HODGINS, J. K., AND KARA, L. B. 2015. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015) 34*.

ZHENG, B., ZHAO, Y., YU, J. C., IKEUCHI, K., AND ZHU, S.-C. 2014. Detecting potential falling objects by inferring human action and natural disturbance. In *ICRA*.