SceneGrok: Inferring Action Maps in 3D Environments

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

Stanford University

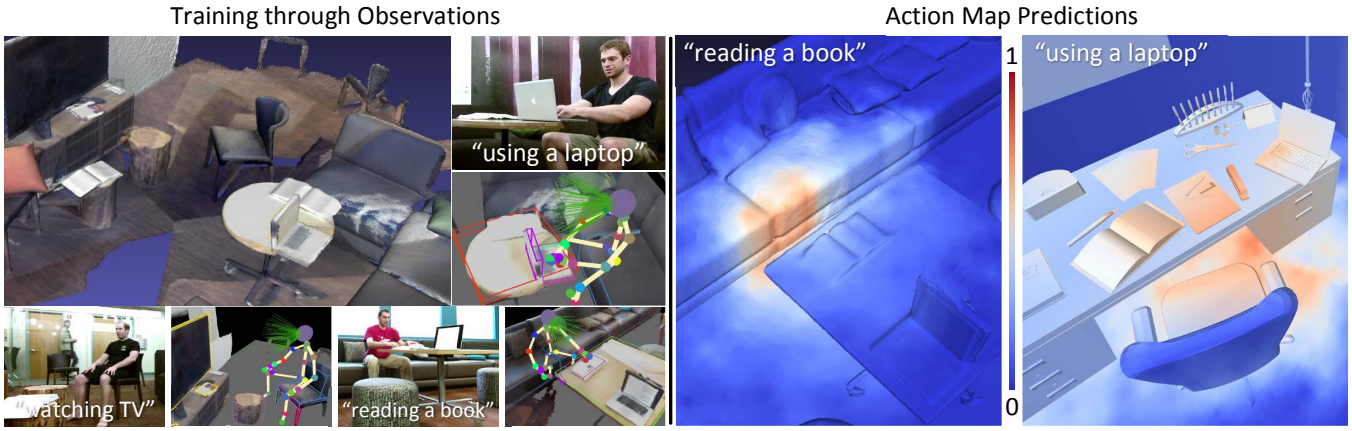

Figure 1: *We predict regions in 3D scenes where actions are likely to take place. We start by scanning the geometry of real environments using RGB-D sensors and reconstructing a dense 3D mesh (left). We then observe and track people as they interact with the captured environments (mid-left). We use these observations to train a classifier which allows us to infer the likelihood of actions occurring in regions of new, unobserved scenes. We call these predictions action maps and we demonstrate that we are able to deduce action maps for previously unobserved real and virtual scenes (see mid-right and right, respectively).*

Abstract

With modern computer graphics, we can generate enormous amounts of 3D scene data. It is now possible to capture high-quality 3D representations of large real-world environments. Large shape and scene databases, such as the Trimble 3D Warehouse, are publicly accessible and constantly growing. Unfortunately, while a great amount of 3D content exists, most of it is detached from the semantics and functionality of the objects it represents. In this paper, we present a method to establish a correlation between the geometry and the functionality of 3D environments. Using RGB-D sensors, we capture dense 3D reconstructions of real-world scenes, and observe and track people as they interact with the environment. With these observations, we train a classifier which can transfer interaction knowledge to unobserved 3D scenes. We predict a likelihood of a given action taking place over all locations in a 3D environment and refer to this representation as an *action map* over the scene. We demonstrate prediction of action maps in both 3D scans and virtual scenes. We evaluate our predictions against ground truth annotations by people, and present an approach for characterizing 3D scenes by functional similarity using action maps.

CR Categories: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—3D/stereo scene analysis;

Keywords: 3D scenes, object semantics, scene understanding

Links: DL PDF WEB VIDEO DATA

1 Introduction

We are increasingly able to capture and represent the world with high-fidelity 3D content. While these geometric representations are suitable for rendering, it is challenging to incorporate them into compelling interactive applications since we cannot readily understand the semantics that underlay our 3D representations. For example, it is still hard to automatically generate agent behaviors in 3D environments since the agent lacks knowledge of object and region functionalities. Even simple questions such as "where can I sit in this room?" are challenging tasks for computers [Grabner et al. 2011]. While we have access to large amounts of 3D content, the knowledge of how to interact with 3D environments, and what humans consider to be functionally important scene parts, is missing.

We aim to learn the functionality of 3D scenes directly from observations of people in real environments by using RGB-D input data. By observing humans interacting with objects in everyday scenes, we can empirically learn the correlation between body poses during actions and properties of a scene and its objects. Specifically, for a given action such as "using a desktop PC", our goal is to learn a model that can take a new scene we have never observed before, and predict whether an agent could perform that function at each location in a new scene. We refer to this representation as an *action map* which encodes the probability of specific actions taking place in specific regions. Examples of action maps are shown in Figure 1 (right). We observe actions in the real world, and train a model to predict action maps in new real-world and virtual environments. We demonstrate the transfer of action maps to a scene database which allows us to characterize the functionality of unknown scenes. We believe that methods for understanding 3D scenes in this manner are an important step towards bridging the semantic gap, and can be useful in a variety of applications, including virtual agent action scripting, scene understanding in robotics, and 3D scene retrieval.

The idea of understanding environments through potential human actions within them is not new; much prior work has targeted this domain [Gibson 1977; Gupta et al. 2011; Pandey and Alami 2012; Fouhey et al. 2012; Koppula et al. 2013; Koppula and Saxena 2013;

Jiang et al. 2013; Wei et al. 2013a]. However, to our knowledge, we are the first to specifically target the problem of inferring functional regions in unobserved real and virtual environments, utilizing the underlying 3D geometry. We address the functional understanding of unknown geometry through the observation of humans interactions in real environments.

We scan a variety of real-world environments such as offices, living rooms, and common areas, and we record human interactions in these scenes through RGB-D tracking and human pose data [Shotton et al. 2013]. We then ask people to label observed actions in these scenes, and we train a classifier to predict the observed actions from the scene geometry. This allows us to establish a basic functional understanding of 3D environments which we apply to new scenes where no interactions were observed and only the raw 3D geometry is provided. As a result, we can annotate new scenes with action maps that specify a likelihood for a given action at every location in the scene. Even with the limitations of current 3D scanning technology, we are able to add a basic functional understanding to 3D scenes, and we show results for both real and virtual scene datasets (see Figures 1, 7, 8).

In summary, we present a novel method for predicting action regions in unobserved 3D environments:

- We introduce a model that learns from observations how to predict possible actions over the space of 3D environments. To do this, we leverage an unsupervised feature learning method to obtain a geometric codebook.
- We evaluate our results by comparing our action map predictions against ground truth annotations provided by people.
- We demonstrate that we can predict action maps for synthetic 3D scenes, and thus characterize their functionality. We illustrate that this enables a novel form of scene retrieval by functional similarity.
- We provide a dataset of observations of people in densely reconstructed 3D scenes, along with action annotations.

2 Related Work

Human-centric understanding of environments has a long history in the concept of affordances, which was introduced by Gibson [1977]. There is a variety of related work that has looked at using human-object interactions to improve pose estimation, object recognition, action classification, and other related problem in different communities [Fritz et al. 2006; Montesano et al. 2008; Stark et al. 2008; Sun et al. 2010; Hermans et al. 2011; Goldfeder and Allen 2011; Bohg et al. 2013; Koppula and Saxena 2013; Zheng et al. 2014]. However, to the best of our knowledge, we are the first to formulate the *action map* representation which aims to predict actions over *regions* in 3D scenes, and to learn such a representation from observations in the real world.

To achieve this, we use observed human poses as an intermediary to connect geometric context to actions. Our goal is orthogonal to the task of pose prediction since we do not predict specific human poses or adjust them for novel geometry. There exists much work on pose prediction that could be integrated with our algorithm. For instance, we can leverage approaches such as Kim et al. [2014]'s *shape2pose*, to fit poses to 3D meshes. This work introduced the concept of human-centric interactions to the shape analysis literature which targets higher-level understanding of both shape collections [Mitra et al. 2013a; Mitra et al. 2013b] and 3D scenes [Fisher et al. 2011].

Another line of work hypothesizes human poses as a latent variable and predicts plausible poses in scenes in order to label objects, or determine likely placements of objects [Jiang et al. 2012; Jiang et al. 2013; Jiang and Saxena 2013]. The authors demonstrate prediction of likely positions for human poses but do not condition on different types of human actions, and intentionally do not rely on observed interactions in the input RGB-D data. In contrast, we directly use observed interactions to predict action regions in full 3D scenes making our goal orthogonal to pose prediction and object labeling.

Other approaches in computer vision aim to learn pose predictions from RGB video stream observations [Delaitre et al. 2012] or to determine affordances in new images based on inferred poses [Gupta et al. 2011; Fouhey et al. 2012]. This line of work estimates rough 3D voxel geometry from 2D images to reason about affordances in 3D space. However, a predefined set of poses is used which is not learned from observations. Similarly, Grabner et al. [2011] focus on the action of "sitting" specifically and sample virtual scenes with posed 3D human models to infer sittable objects. In contrast, we use real-world observations of a variety of human actions.

Other research has also leveraged RGB-D data and focused on learning human activities to classify objects and actions [Koppula et al. 2013; Wei et al. 2013a; Wei et al. 2013b]. Typically, these approaches take advantage of temporal features that are not available to us in static 3D scenes.

Overall, our approach is driven by the desire to add semantics to 3D content. For instance, we would like to equip virtual agents with the ability to automatically interact with novel 3D geometry. Our method relies only on 3D geometric features and not on appearance models since most existing 3D content does not have consistent color information. Furthermore, unlike most prior work, we do not require annotated object categories or labels on the raw geometry.

3 Overview

Our method aims to understand *actions* by modeling how *people* interact with their 3D environments. More specifically, we model the space of interactions between the human body and the geometry of the environment in which actions take place. This allows us to make predictions about the functionality of 3D geometry (i.e., where can actions take place) and the implied functional characteristics of virtual scenes.

We first create a dataset containing 3D reconstructions of real-world environments and pose data for human-scene interactions (Section 4). We annotate this data with action labels so that we can train a supervised classifier for predicting actions. We then partition scanned 3D scenes into disjoint sets of segments. Given recorded pose data, we identify segments which are activated by skeletons during specific actions. Using an unsupervised feature learning approach based on k-means clustering, we group activated segments using simple geometric features (Section 6.1). The cluster *centroids* span a high-dimensional space in which we embed activated segments from input observations. Using this data, we train a classifier for each action label allowing us to establish a correlation between pose observations and the underlying 3D geometry during that action (Section 5.1).

In order to infer action maps in unobserved environments, we predict the probability of the given action taking place at discrete positions (x, y) within the scene by sampling poses at the given positions with rotations θ_i. We aggregate the probability of an action label at every position (x, y) by accumulating the probabilities for all rotation samples θ_i. Thus, for every location we obtain the feasibility of that action; we call this probability distribution the action map over the environment (Section 7).

An overview of our algorithm is in Figure 1, starting with training data acquisition, and ending with action map prediction.

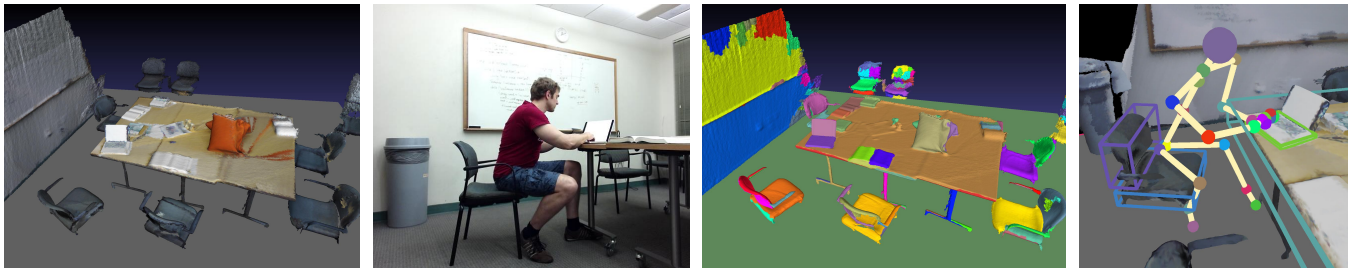

Figure 2: ***From left to right:*** *Dense 3D scan captured with an RGB-D sensor; color frame from an activity recording session taken in this scene; partitioning of the scene into disjoint segments; skeleton tracked from the same recording, with bounding boxes of active segments shown. An active segment is a segment within 30* cm *of one of the skeleton's joints.*

4 Data Acquisition

One of the key aspects of our method is that we learn the concept of action maps from real-world observations. To this end, we acquire 3D training data using RGB-D sensors.

We first obtain 3D reconstructions of all training scenes using a Microsoft Kinect sensor. In order to generate corresponding 3D meshes, we use a real-time volumetric fusion framework [Nießner et al. 2013], which regularizes out noise from the input data; Figure 2 shows an example scan. We align all reconstructions with the ground plane such that scans have a common up vector. Next, we place a Kinect One sensor at a static position within the physical environment. The sensor is used to record people as they carry out common daily activities; i.e., *actions*. In the RGB-D recordings, we capture the human poses within the 3D scene at 30Hz using the skeleton tracking provided by the Kinect SDK [Shotton et al. 2013]. These poses contain the 3D position of 24 skeletal joints, which we map to the coordinate frame of the reconstructed 3D scene. A typical recording session lasts four minutes and between two and four recordings were taken per scene.

The recordings were annotated by a volunteer who indicated the time ranges over which the subjects performed any of a prespecified set of actions. Multiple actions can occur at the same time: e.g., sitting on furniture often co-occurs with using a desktop computer. Figure 2 shows an example frame annotated as "using a laptop" and "sitting on furniture". Table 1 provides a complete list of the actions that were annotated and the size of our dataset.

One key difference between this and many existing RGB-D action datasets is that we acquire dense scans of the scene geometry for each recording generated by hundreds of input depth frames, instead of only capturing the depth information from a set of fixed sensors. For our work, this dense geometric reconstruction is critical as it enables us to transfer action information from real observations to virtual 3D environments.

Data Processing We transform the raw geometry in the dataset into a segment-based representation. For each 3D mesh, we oversegment the geometry using a graph segmentation method [Felzenszwalb and Huttenlocher 2004] with a distance metric based on surface normals [Karpathy et al. 2013]. An example of a generated scene segmentation is shown in Figure 2. With this oversegmentation, we establish a higher-level scene representation composed of coarser primitive objects and parts, which is sufficient for our purposes. In general, segmentation of reconstructed 3D scenes is a challenging research problem beyond the focus of our contributions. More sophisticated scene parsing methods (e.g., [Nan et al. 2012]) may be useful for improved segmentation of the input scenes.

Action	Scenes	Minutes
Sit on furniture	14	54
Use a desktop computer	5	15
Read a book	10	13
Use a laptop computer	7	9
Stand on the floor	12	7
Write on a whiteboard	4	7
Watch television	4	6
Total Scenes:		14
Total Recordings:		45

Table 1: *Summary of the dataset. For each action, we show the number of scenes in the database with at least once instance of that action and the total time spent observing this action across all recording sessions.*

For each skeleton at each time point, we collect a set of segments in the scene within 30 cm of each skeleton joint. We refer to these as the *active segments* for a joint. Figure 2 shows the skeleton and all the active segments for the image on the left. For a given pose p, we use $\beta_j \in p$ to denote the set of active segments for each joint j.

In addition to the 24 skeletal joints, we estimate a gaze vector based on the position of the shoulder and head joints. Since a single direction would be highly unstable, we sample the hemisphere around the estimated vector using a power-weighted cosine distribution. For every sample, we construct a ray which we intersect with the scene within a range of 2 m. All segments that are hit by more than 10% of the sample rays are taken to be in the set of active segments for the gaze of the skeleton. For notational simplicity, we consider the gaze to be an additional joint of the skeleton, with an independent set of active segments.

Given the limitations of current RGB-D sensor technology, there are many challenges in 3D reconstruction and pose tracking: significant noise in 3D scene reconstructions, tracking failures due to occlusions or complex kinematics, and limited body part localization accuracy. All these issues lead to open research problems. We present what is possible with current consumer technology and show that despite these limitations, we can predict action maps for many common actions in indoor scenarios. Progress in RGB-D sensor technology will improve the robustness and quality of the results as well as the diversity of actions that can be effectively classified.

5 Action Map Model

Our approach characterizes actions by examining the geometric properties of objects that people interact with using different parts of their body. For example, the activity of sitting is characterized

by the presence of a seat-like surface supporting the hips of the person from below. Likewise, the activity of typing on a laptop is characterized by the hand being in contact or close proximity to the laptop's keyboard surface.

More specifically, we would like to predict the probability that an action is possible while an agent's hips are located at a given (x, y) location in 2D space. We refer to this as an *action map* $M_a(x, y)$ for a given action a and point (x, y) in a scene.

To define our model for $M_a(x, y)$, we will aggregate information over human poses centered at this location. In Section 5.1, we will show how we use a supervised learning approach to learn a function for the probability $P_a(x, y, p)$ of action a being performed while a human pose p is centered at (x, y). Our learned model takes as input a set of features produces by a featurization function ψ (defined in Section 5.2) which computes a fixed-dimensional feature vector from the set of active segments β_j of each joint of the pose. L is the classifier function that has been trained from the resulting features.

$$P_a(x, y, p) \mathrel{\widehat{=}} L(\psi(\forall_j\{\beta_j \in p\}))$$

We compute our final action map score at a given location by integrating over $P_a(x, y, p)$ for different poses $p \in H_a$:

$$M_a(x, y) = \int_{p \in H_a} P(p) P_a(x, y, p) \approx \sum_{p \in \tilde{H}_a} \frac{1}{\tilde{H}_a} P_a(x, y, p)$$

Here, H_a represents the space of all poses corresponding to action a, and $P(p)$ is the probability that a human performing action a is in pose p. We estimate the integral by sampling a set of poses $\tilde{H}_a$ chosen at random from the set of all poses corresponding to that action observed in the training set. Each randomly chosen pose is assigned equal weight.

For each action map M_a, we approximate $P_a(x, y, p)$ using the class membership probability of a supervised classifier for action a. A key challenge in successfully training a supervised classifier lies in the choice of featurization function ψ. We describe several featurization schemes in Section 5.2, and find that a segment dictionary-based approach performs best (described in detail in Section 6).

5.1 Supervised Classification

For the supervised learning stage, we first create a training set of positive and negative instances for each action a that we would like to predict. Positive instances are randomly chosen from the interaction time ranges that were annotated with the action a. We randomly select negative instances from any other interactions. We select a maximum of 5000 positive and 5000 negative observation instances for each action we learn.

Once we have created a training set, we extract for each instance all segments in the scene that are activated by the current body pose $\beta_j \in p$. The set of featurized segment vectors f for each action is used to train a random decision forest classifier [Breiman 2001]. Random forests are an ensemble learning method for classification that construct sets of decision trees trained on different subsets of the available variables. We followed standard heuristics for setting the number of random trees to 10 and the number of randomly chosen features per tree to $\lceil\sqrt{k}\rceil$. In our implementation, we use the WEKA machine learning toolkit [Hall et al. 2009].

5.2 Segment Featurization

A featurization function takes an observation instance consisting of multiple active segments interacting with the joints of a pose and returns a fixed length feature vector. Thus, it addresses the variation in the number of segments in training or test samples. In our case, the variation comes from different numbers of active segments for each joint.

Simple aggregation of the raw geometric features of segments is not meaningful. For this reason, we use aggregate features that capture the identity of interacting segments. We consider three featurization functions that compute a fixed length feature vector for a given set of active segments.

Segment Presence (ψ_{pres}) Presence of a segment is the simplest feature and the baseline against which we compare the other approaches. For each joint j there is a single indicator feature which is equal to one if there is an active segment and zero otherwise. The input is the set of all active segments $\{\beta_j \in p\}$, and the output is a binary feature vector of length $|j|$.

Segment-Joint Interaction Score (ψ_{agg}) We compute a score for how likely a segment s is to be interacting with a specific joint j for a given action a by training a binary predictor $L_{j,a}(s) \rightarrow [0, 1]$ (as described in Section 5.1). We select the set of all segments β_j interacting with the joint as positive instances, and active segments never observed interacting $\{s \notin \beta_j\}$ as negative instances. At test time, the input is the set of all active segments $\{\beta_j \in p\}$, and the output is a feature vector of length $|j|$ which sums the interaction score for all active segments in each joint: $\sum_i L_{j,a}(s_i)$. Here, we make an independence assumption that the interaction likelihood scores for each joint-segment pair are not correlated. This scheme is similar to the feature compatibility measure used by Kim et al. [2014] to predict contact likelihood between mesh surface points and body parts.

Segment Dictionary Activation (ψ_{dic}) Using the dictionary of segment centroids $\mathcal{D}$ learned as described in Section 6, we map observed segments to encoded feature vectors f. We follow the approach of Coates and Ng [2012] and use a simple non-linear function of the per-centroid responses $\mathcal{D}^T s$ for the given segment s. We use the soft-thresholded function $\max(0, \mathcal{D}^T s - \alpha)$ where we set α to be the mean centroid response for s. Once we apply this function to each input segment, we obtain an $\mathbb{R}^k$ centroid response vector which encodes that segment's activation against each of the dictionary centroids. This form of encoding is known to perform much better than single cluster assignments or sigmoidal non-linear functions [Coates and Ng 2012]. We aggregate these encoded responses for all active segments by summation along each centroid dimension. The final aggregated centroid response $f \in \mathbb{R}^k$ is the feature vector used by the supervised learning procedure.

6 Segment Dictionary Learning

Our segment dictionary approach is inspired by unsupervised feature learning techniques based on k-means clustering that have been shown to be successful in image classification [Coates and Ng 2012]. These approaches are based on encoding "codebooks" extracted through k-means clustering on the training data – they are simple, easy to parallelize, and have few parameters. This codebook encoding scheme is a way to capture the variability of the input data compactly in a lower-dimensionality space and avoid overfitting.

We first define a feature set for our segment geometry. We then accumulate all observed interactions of tracked body parts with segments in our scene. We perform k-means clustering of all active segments in the defined feature space to obtain a set of codebook

centroids that will be used to encode particular instances of observed interactions.

6.1 Segment Geometry Features

We choose a small number of physically interpretable features to represent the segments extracted from the unstructured 3D scene input meshes. Segment features are computed with respect to an oriented bounding box (OBB) of the segment points. We constrain the OBB to have one of its axes be upwards. The bounding box dimensions are computed only with points between the 10th and 90th percentile along each dimension to provide robustness against outliers. Specifically, we use the following geometric features:

1. Vertical position of the OBB centroid above ground
2. Height of the OBB: $\max_z - \min_z$
3. Diagonal of OBB in the xy plane
4. Area of OBB in the xy plane: $\sqrt{A_{xy}(OBB)}$
5. Magnitude of the dot product of the minimum PCA vector with the world up vector.

These features define an $\mathbb{R}^5$ feature vector for each segment. We intentionally chose simple features that have direct mappings to physical properties of the object segments and characterize their functionality: distribution at different vertical heights, approximate vertical size, horizontal area available for interaction, and orientation of the dominant plane with respect to the upwards vector.

6.2 Segment Clustering

To capture the variety of different object segments that people interact with, we perform k-means clustering on the feature space we defined above, and extract a set of codebook centroids which will encode the segments in each observed interaction.

We first accumulate the set S of all active segments that have been interacted with for more than 10 seconds. We then compute the feature vectors $\widetilde{s} \in \mathbb{R}^5$ for each segment $s \in S$ and normalize them by subtracting the mean and dividing by the standard deviation:

$$s = \widetilde{s} - \text{mean}(\widetilde{S}) / \sqrt{\text{var}(\widetilde{S})}$$

We also perform a *whitening* step, a form of independent component analysis also known as *sphering* [Hyvärinen and Oja 2000]. This reduces the cross-correlation between samples, and is known to improve training performance for supervised learning [Coates et al. 2011]. We do this by using the eigenvalue decomposition of the covariance matrix of the normalized segment feature vectors $V \Sigma V^T = \text{cov}(S)$:

$$s = V(\Sigma + \epsilon I)^{-1/2} V^T s$$

where $\epsilon = 0.01$ is a small regularization constant to avoid numerical issues due to division by eigenvalues close to zero.

Given the set of normalized and whitened segment feature vectors, we perform k-means clustering to obtain a set of k codebook segment feature vectors. We use the k-means++ initialization approach to improve clustering quality and we run the algorithm until convergence [Arthur and Vassilvitskii 2007]. Once we have obtained a clustering, we form a dictionary matrix $\mathcal{D} \in \mathbb{R}^{5 \times k}$ comprised of the k cluster centroids as its columns. This dictionary can now be used to encode any segment s into a code vector that will be used as a feature vector for supervised learning. Examples of highly-activated segments for a particular centroid are shown in Figure 3.

Figure 3: *We represent our mesh segments in codebook space, which is a low-dimensional representation for embedding activated segments. Examples of highly-activated segments for one of our learned segment centroids, indicated in red, in four different scenes (top row: scanned 3D scenes, and bottom row: synthetic 3D scenes). We see that this centroid roughly corresponds to the backrest of a chair.*

7 Action Map Prediction

The trained classifiers for each action a can be used to predict the likelihood of that action given a new set of activated segments $\{\beta_j \in p\}$ for a pose p in a 3D environment. In order to retrieve activated segments within test scenes, we sample the space of the scene with a blue noise sample-based search scheme (all results shown use 2000 samples). We iterate over sampled points and place a randomly sampled skeleton out of the observations that were annotated with the given action a. The skeleton is translated so that the base of the hips is at the sampled xy position. The vertical position z above the ground remains the same as in the original observation. In addition, we evaluate several orientations θ_i at 45 degree increments; i.e., we rotate the skeleton by θ_i.

For each sample (x, y, θ_i), we retrieve activated segments $s \in \beta_j$ from the scene by nearest-neighbor lookup from each joint position. The feature vector of each activated segment s is then normalized, whitened, and its featurization $\psi(s)$ is computed. This is the same procedure as in the training phase described in Section 5.2. The encoded instance is then given a likelihood for being a positive example of an action a by the trained classifier L_a. The map of likelihood predictions $M_a(x, y)$ over the space of the scene is our output, the *action map*. We visualize these predictions as heatmaps over the scene indicating where actions are likely to take place given the evidence of the surrounding activated geometric context.

8 Results

We first quantitatively evaluate the performance of our approach and the three featurization functions against ground truth annotations provided by people. We then present examples of action map predictions on scanned test scenes, and on synthetic 3D scenes from a database assembled by prior work [Fisher et al. 2012]. Finally, we show how an approach using action maps provides a functional similarity metric for scenes and can be used for scene retrieval.

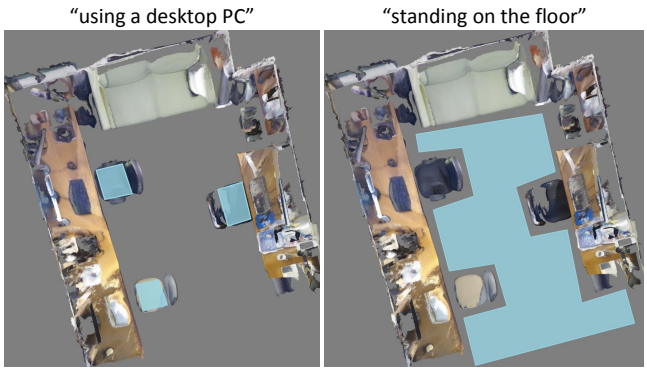

Figure 4: *Ground truth action map annotations provided by participants for "using a desktop PC" (left) and "standing on the floor" (right) for an office scene in our dataset.*

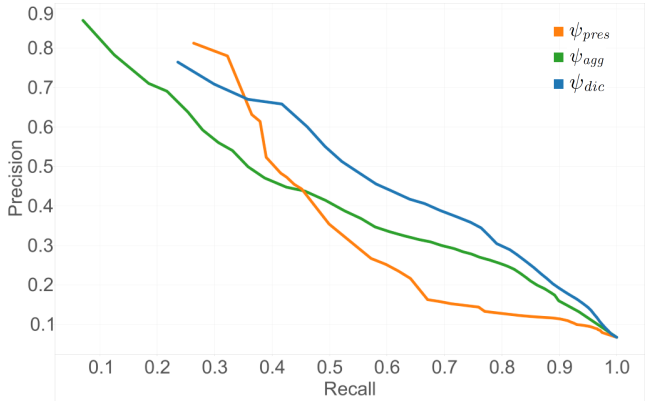

Figure 5: *Precision vs recall plots for the three featurization approaches we use, computed against ground truth annotations:* ***orange:*** *segment presence baseline ψ_{pres},* ***green****: joint-segment score baseline ψ_{agg}, and* ***blue:*** *segment dictionary activation ψ_{dic}.*

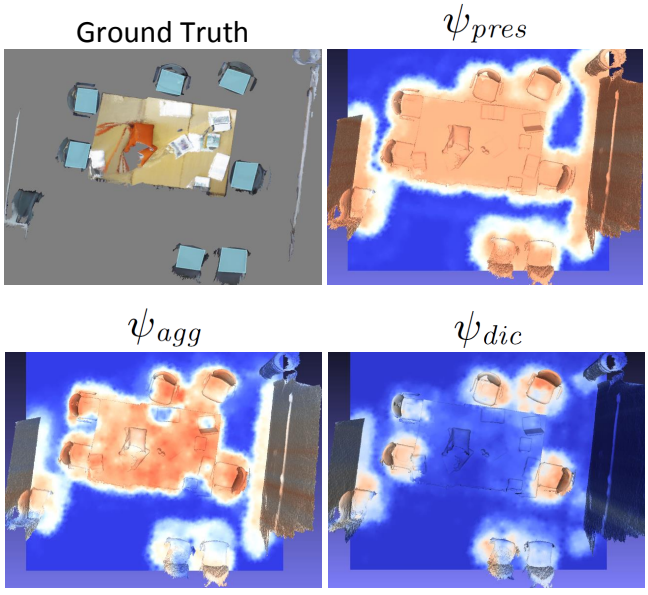

Figure 6: *Qualitative comparison of the "sitting on furniture" action map prediction using the three featurization functions described in Section 5.2, for the scene shown in the middle row of Figure 7. We compare: segment presence baseline ψ_{pres}, joint-segment score baseline ψ_{agg}, and segment dictionary activation ψ_{dic}.*

8.1 Evaluation

To evaluate the performance of our approach, we compare our action maps against ground truth annotations provided by people. We asked three volunteers to annotate our dataset of scanned 3D environments (see Table 1). The volunteers were provided with a 3D view of the scanned scene they could navigate to disambiguate objects. They provided annotations as sketched out regions in a top-down view of each scene. The study participants were instructed to give 2D sketches of the regions where each of a list of actions could plausibly take place for an adult, assuming the objects in the scene remain static. Figure 4 gives example ground truth masks for one of the scenes, and the complete prompt given to the volunteers is provided in the supplemental materials.

Using these ground truth annotations, we compared the predictive performance of each of the featurization schemes presented in Section 5.2. We first establish a test set using four of the scenes in our dataset, and train on the remaining scenes. We then compare the output action map predictions against the annotated ground truth 2D masks. We compute averaged precision-recall curves over all actions for each approach which we plot in Figure 5.

The simple segment presence baseline ψ_{pres} has the lowest performance (orange). The aggregation of segment scores ψ_{agg} performs better (green) and the overall performance is further improved using the segment dictionary activation approach with $k = 100$ centroids ψ_{dic} (blue). The maximum F1 scores (harmonic mean of precision and recall) were 36.4% for the presence baseline, 45.0% for the segment score aggregation baseline, and 52.2% for the segment dictionary approach.

Figure 6 visualizes the behavior of the featurization functions plotted in Figure 5. The presence baseline ψ_{pres} is not able to distinguish between different segments and has a large proportion of false positives. The joint-segment score baseline ψ_{agg} can discriminate between different geometric properties of segments but still has a significant false positive rate. The segment dictionary activation function ψ_{dic} most accurately captures the ground truth annotation. All the remaining results presented in this paper use the ψ_{dic} featurization function.

8.2 Action Map Predictions

To demonstrate prediction of action maps for novel scenes, we show a series of results where we use scanned scenes as unobserved test instances and train on all remaining scenes. We then predict action maps in the test scene. We visualize the action map classification predictions as heat maps ranging from high confidence that the action cannot be performed (saturated blue) to high confidence that the action can be performed (saturated red). Figure 7 shows these results for several actions and test scenes.

In addition to testing on scanned 3D environments in the dataset we collected, we also examine the generalization of our action map predictions to existing synthetic 3D scenes. This is a challenging scenario since the geometry of synthetic scenes differs significantly compared to the scans acquired with RGB-D sensors on which we train. However, the geometric features we use are defined predominantly on oriented bounding boxes of mesh segments acquired through a normal-based distance metric, which can be compared between real and virtual scenes. Figure 8 shows example predictions on a few scenes from the corpus of Fisher et al. [2012].

8.3 Scene Similarity through Action Descriptors

We formulate a simple descriptor based on integrating the set of predicted action maps over the space of each scene. This results

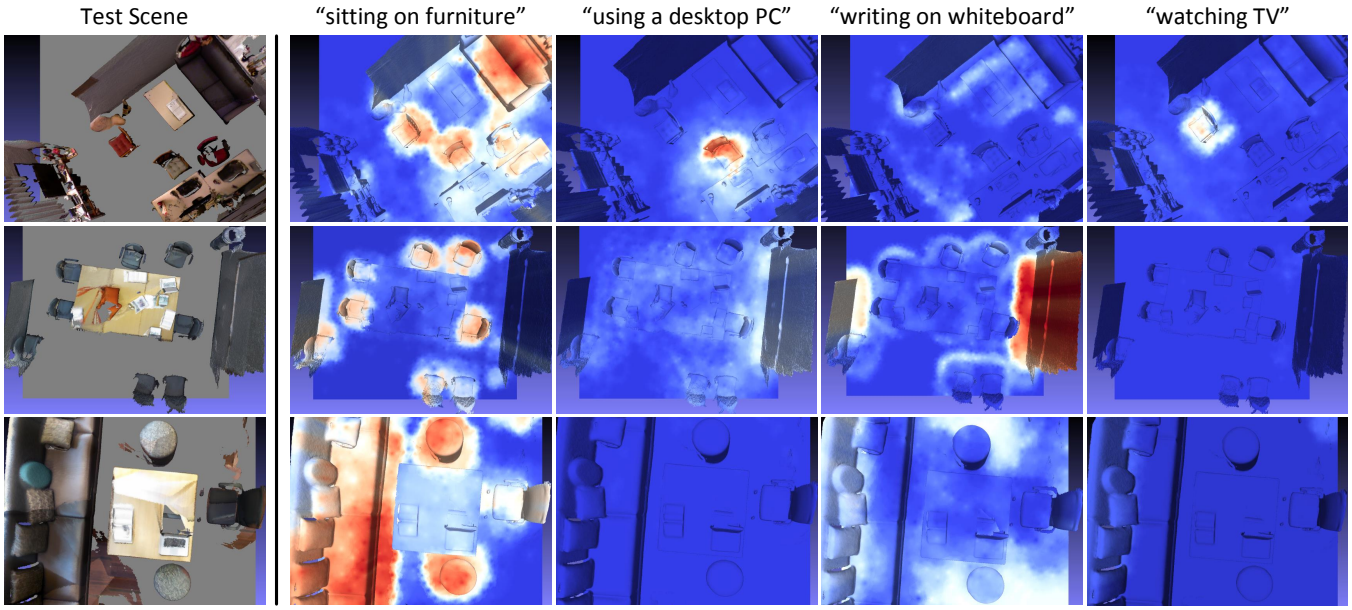

Figure 7: *Predictions of different action maps on three test scenes (first column). In the second column high likelihood of sitting is correctly predicted on couches, chairs, and stools with varying geometry. Only the first scene with the desktop workstation has a high likelihood for “using a desktop PC” in the third column. The fourth column shows predictions for “writing on a whiteboard” which are sensitive to false positives due to segments with vertical planes similar to whiteboards. Finally, the fourth column shows action map predictions for “watching TV” which are high for the first scene with the chair positioned in front of the TV. The whiteboard-like segment on the left in the second scene is deceptively similar to a TV but there are no spurious predictions for TV watching, likely due to the surrounding segments not being arranged for watching TV.*

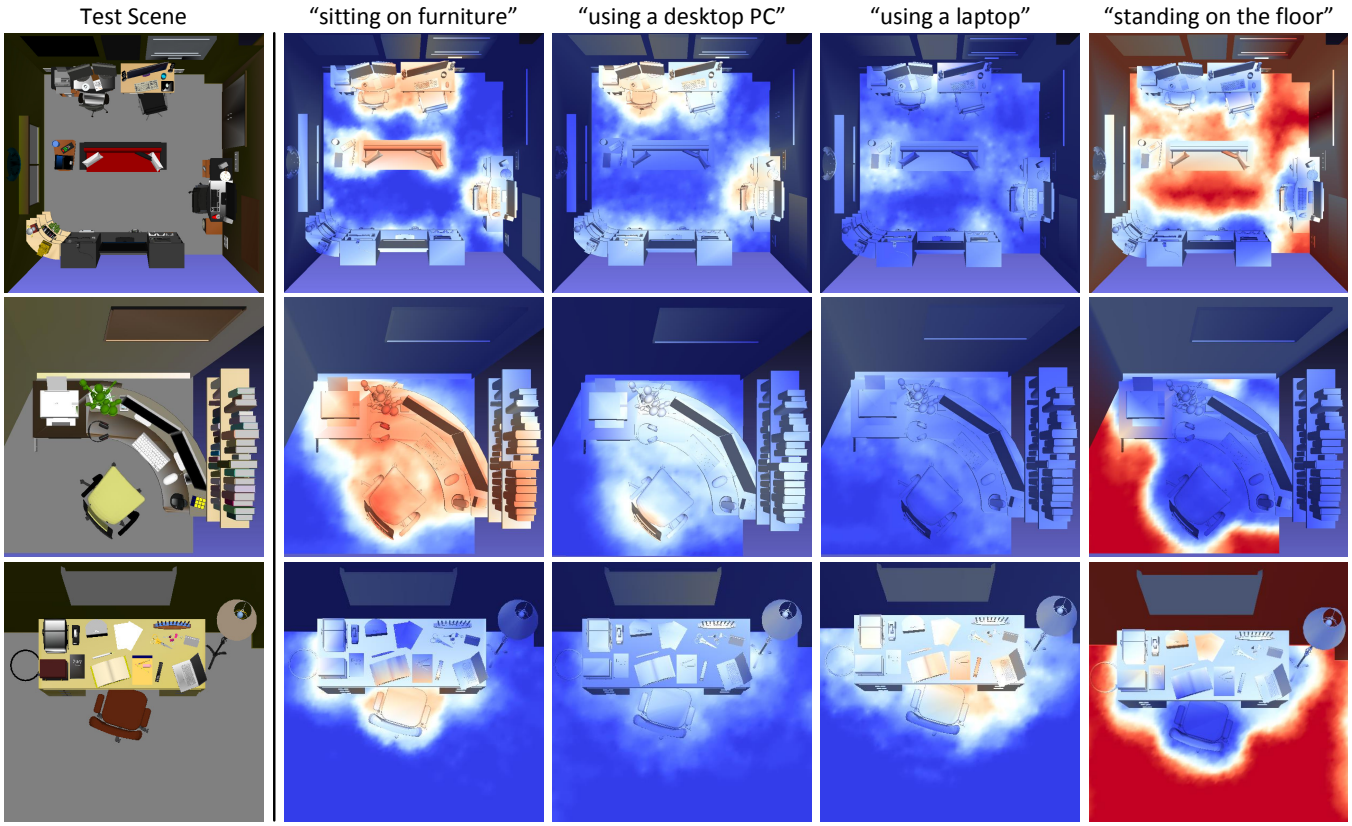

Figure 8: *Action map predictions on some synthetic 3D scenes. The leftmost column shows the input scenes. The sitting prediction for the second scene correctly activates the chair but suffers from a high false positive rate, the “using laptop” prediction in the third column for the third row is shifted to the right of the chair due to the uncommon placement of the laptop. We note that the predictions for “standing on the floor” recover the free space in the scene where a person could stand without explicitly considering collisions.*

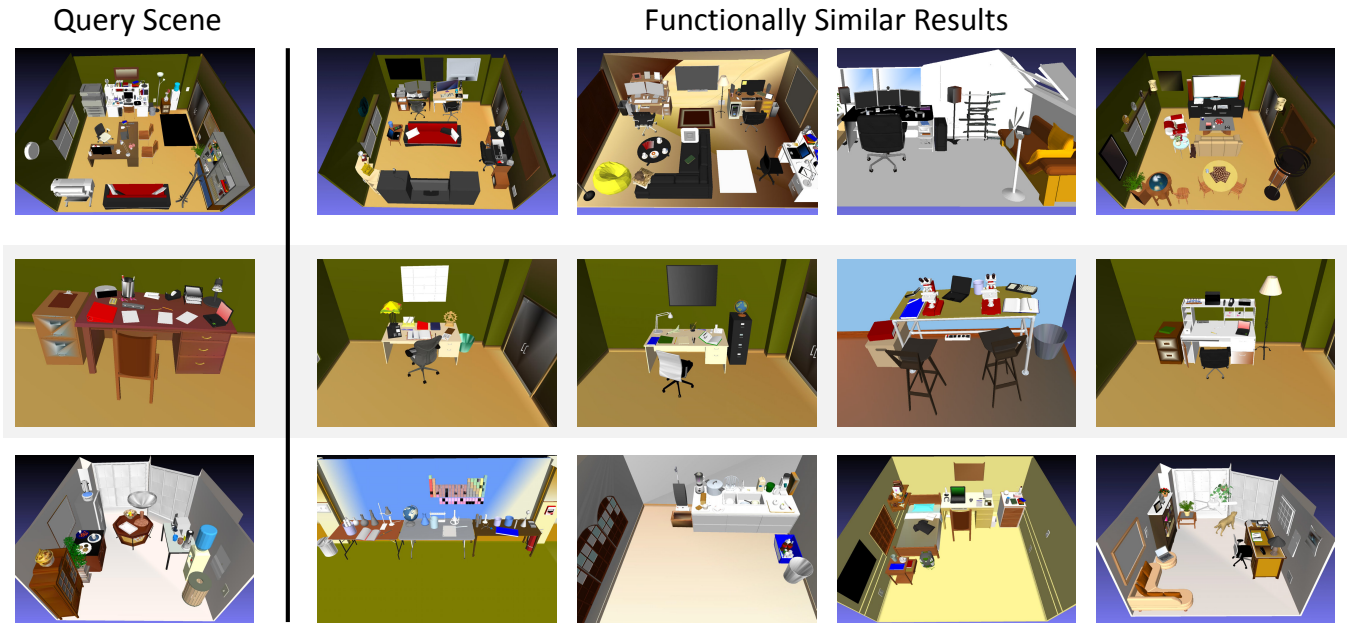

Figure 9: *Retrieval of functionally similar scenes. The scenes on the right are the top four most similar scenes by action map profile to the query scene on the left.*

in a feature vector of dimension $|A|$, the number of predicted action maps. Our intent is to show that such a simple approach can still retrieve functionally similar scenes. We see potential for action map-based scene similarity as another useful dimension of comparison that can be leveraged for scene indexing and retrieval.

Figure 9 demonstrates an application of action descriptors to retrieve functionally similar scenes given query scenes. Scenes are ranked according to their Euclidean distance from the query scene in the space of the $|A|$-dimensional action descriptors. We use the action map set described in Table 1, trained on all our training recordings. The first row of retrieved results supports the "using a desktop computer" action present in the query scene, except for the last scene which supports "watching TV". In the second row, all scenes support either the "using a laptop computer" or "reading a book" action exemplified in the query scene. The third row does not contain many meaningful results; this is because our method is not trained on many actions that are relevant to the kitchenette query scene. Note that unlike many existing algorithms for comparing scene geometry, this method does not rely upon text labels or categories assigned to the objects in the environment in any way.

9 Limitations

Many limitations of our approach are due to noisy input of current RGB-D sensors. Partially scanned scenes and segmentation failures can cause errors in the predicted action maps. The noise regularization of volumetric depth map fusion partly mitigates some of these issues; however, three categories of failures cases remain: geometric similarity failures, obstructions, and poor signal isolation.

Geometric similarity failures. The most common failure of our algorithm occurs when an incorrect similarity is computed between two objects or object parts. If two similar objects are not found to be similar in our geometric feature space, then we will not correctly transfer actions from training scenes to test scenes. Likewise, if two dissimilar objects are found to be similar, then spurious activations can occur. The bottom row of Figure 10 shows one example where an open book is found to be geometrically similar to a laptop. Consequently, this region activates as "use a laptop computer" even though no laptop is present. Geometric similarity failures can occur for many reasons: our 3D scans are both incomplete and contain inherent noise; there is natural variation in the shape and size of objects; and our per-segment geometric features do not perfectly characterize the shape and material of an object. With improved scanning quality, incorporating more sophisticated geometric features should become viable.

Obstructions. Many actions become difficult or impossible to complete when obstructing objects are present. One example is shown in the top row of Figure 10. Here, the presence of a keyboard makes the chair very uncomfortable for sitting, but our algorithm still classifies this area with very high sitting activation. This is because we do not explicitly construct negative examples to train our classifiers. Instead, negative examples for an action are drawn from any poses not explicitly labeled for that action. Creating explicit negative examples to exemplify scenarios which inhibit actions should improve our ability to correctly account for the presence of interfering objects.

Poor signal isolation. For some types of actions, a large number of segments may be activated by a pose performing this action. For example, an agent using a desktop computer might be gazing at a monitor and have joints near a keyboard, a mouse, a mousepad, a desk, the floor plane, headphones, a mug, a speaker, and a notepad. However, not all of these objects are required for using a desktop computer. With insufficient training data, it is likely that our classifier will not correctly deduce which objects are most relevant to an interaction, potentially resulting in both false positive and false negative classifications. This problem is made more severe by the fact that our skeleton part tracking is vulnerable to occlusions and is not accurate to more than approximately 15cm.

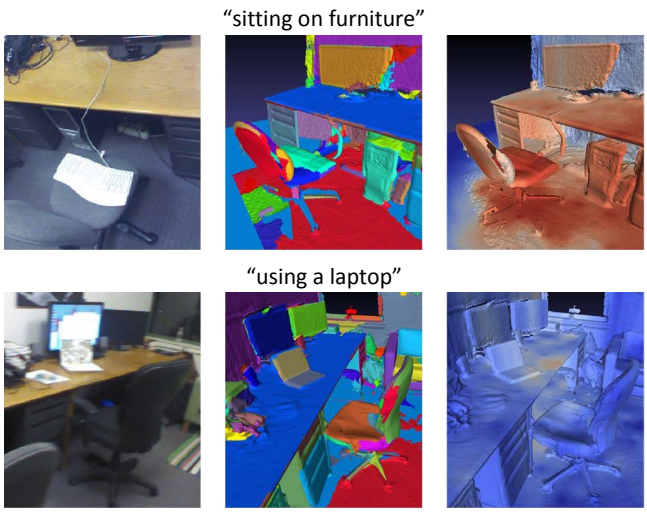

Figure 10: ***Failure cases. Middle column:*** *segmentation of each scene.* ***Right column:*** *Visualization of the estimated action map likelihood.* ***Top row:*** *our algorithm suggests high activation for sitting despite the presence of a keyboard interfering with sitting.* ***Bottom row:*** *our algorithm suggests high activation for using a laptop, despite the fact that no laptop is nearby.*

10 Discussion and Future Work

We presented a method to annotate arbitrary 3D scenes with action maps which estimate the probability of a given action occurring at each location in the scene. Our model for generating action maps is trained from real-world observations on a pre-specified set of possible actions. These action maps are one step towards the broader goal of a functional understanding of scenes and can be used to both augment existing applications in computer graphics as well as enable many new avenues of research.

Object-based action maps. Though we did not focus on connecting action observations to semantically distinct objects within the reconstructed 3D scene, we empirically observed a strong correlation of body part contact and distinct object parts. A per-body-part decomposition of action maps is an interesting avenue for future work which can be useful in object segmentation, categorization and functional annotation. The functionality of objects arises from how human body parts interact with object parts so this form of annotation would be useful for informing virtual agent behavior.

Agent action scripts. The actions performed by real agents interacting with an environment are not independent events. Creating automated agents for 3D environments requires modeling the correlation and causation underlying why agents perform actions and in which order. For example, we might generate an "action script" that describes how an agent interacts with a studio apartment, including events such as "get food from fridge", "sit on couch", and "switch TV channel". When attempting to execute such a script on an environment, the action maps presented in this paper are required whenever the agents needs to know where to be in order to perform an action. The action script itself might even be learned from observations of real agents using a similar training set. The ability to automatically generate autonomous agents to populate virtual worlds has significant applications in games and films.

Action annotations. Recent work has looked at understanding 3D scenes by comparing the geometric relationships between objects [Fisher et al. 2011; Xu et al. 2014]. In these methods, scenes are represented as graphs, objects, or collections of objects are nodes in the graph and edges are spatial relationships between nodes. These representations could be augmented with action annotations using action maps, where some nodes represent "localized actions" – regions in space where a given action is likely to occur. These action nodes could be connected to other objects or actions via edge-based relationships, enabling these methods of relating scenes to incorporate both geometric and functional properties of scenes.

Functional scene synthesis. To mitigate the burden of content generation, methods have been developed to automate the synthesis of 3D scenes [Fisher et al. 2012; Xu et al. 2013]. These methods work by studying the spatial relationships between objects observed in example scenes. However, there is only an indirect attempt to make the generated scenes provide the same types of functionality present in the input examples. By building upon these methods, we would like to create a system that can generate scenes that better capture both the geometric and functional aspects of the training scenes. We might seek to generate a scene whose action maps resemble either the input training scene or a manually specified objective. For example, we could ask for a scene that contains 10 chairs and supports "using a laptop computer", "watching a television", and "writing on a whiteboard", using these functional constraints to guide the placement of the objects towards desirable locations.

Acknowledgements

We would like to thank Angela Dai for the video voice over. This research was supported by the Max Planck Center for Visual Computing & Communication and a Stanford Graduate Fellowship. We gratefully acknowledge the support of NVIDIA Corporation with the donation of Titan GPUs used for this research. We also thank the anonymous reviewers for their comments and suggestions for improving this paper.

References

ARTHUR, D., AND VASSILVITSKII, S. 2007. k-means++: The advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms*.

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

BREIMAN, L. 2001. Random forests. *Machine learning*.

COATES, A., AND NG, A. Y. 2012. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*.

COATES, A., NG, A. Y., AND LEE, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *ICAIS*.

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., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. In *ACM TOG*.

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

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*.

FRITZ, G., PALETTA, L., BREITHAUPT, R., ROME, E., AND DORFFNER, G. 2006. Learning predictive features in affordance based robotic perception systems. In *IROS*.

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

GOLDFEDER, C., AND ALLEN, P. K. 2011. Data-driven grasping. *Autonomous Robots*.

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

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

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*.

HERMANS, T., REHG, J. M., AND BOBICK, A. 2011. Affordance prediction via learned object attributes. In *ICRA Workshop on Semantic Perception, Mapping, and Exploration*.

HYVÄRINEN, A., AND OJA, E. 2000. Independent component analysis: algorithms and applications. *Neural networks*.

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*.

KARPATHY, A., MILLER, S., AND FEI-FEI, L. 2013. Object discovery in 3D scenes via shape analysis. In *ICRA*.

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*.

MITRA, N., WAND, M., ZHANG, H. R., COHEN-OR, D., KIM, V., AND HUANG, Q.-X. 2013. Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses*.

MITRA, N. J., PAULY, M., WAND, M., AND CEYLAN, D. 2013. Symmetry in 3d geometry: Extraction and applications. In *Computer Graphics Forum*.

MONTESANO, L., LOPES, M., BERNARDINO, A., AND SANTOS-VICTOR, J. 2008. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*.

NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM TOG*.

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

PANDEY, A. K., AND ALAMI, R. 2012. Taskability graph: Towards analyzing effort based agent-agent affordances. In *RO-MAN, 2012 IEEE*.

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*.

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*.

SUN, J., MOORE, J. L., BOBICK, A., AND REHG, J. M. 2010. Learning visual object categories for robot affordance prediction. *IJRR*.

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., CHEN, K., FU, H., SUN, W.-L., AND HU, S.-M. 2013. Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM TOG*.

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*.

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*.