Scene Reconstruction and Analysis from Motion

Changgu Kang and Sung-Hee Lee*

Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

Abstract

Human-object interaction is important information for scene creation and understanding. Most previous studies obtain the interaction-contextual information from the observed data on human-object interaction, but the data collection requires significant amount of time and effort, as well as state-of-the art capturing technique. In addition, the observation-based approach cannot be applied to virtual objects well. As a viable alternative, we propose a novel method to reconstruct synthetic scenes purely from captured motions and to analyze the interaction-contextual information of the synthetic scenes and motions. The scene reconstruction process searches for 3D objects from an object database that match the captured motions, which is achieved by constructing abstract objects containing contact-related information inferred from captured motions. Scene analysis process obtains interaction-contextual information, including *interaction behavior*, *functionality of an object*, and *the interaction space of an object*. We demonstrate the effectiveness of our method through a number of experiments.

Keywords: Human-object interaction, Scene reconstruction, Scene analysis
2010 MSC: 68T05, 68T45, 68U01

1. Introduction

Creating a scene or analyzing information on interaction with a scene are of interest to researchers in many fields including computer graphics, computer vision, and HCI [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Human interaction behavior occurs frequently in daily life, and thus is an important factor for creating and understanding a particular situation or scene. For example, when we observe, in a single scene, human-object interactions such as 1) some people sitting on a chair, 2) some holding books, 3) some looking at a board in front of them, and 4) a person writing on the board, we can infer that the scene is a classroom.

Human-object interaction includes an actor, interaction behaviors, and a target object along with its functionality and interaction space. A number of studies have been conducted to infer such context, using various sensor data such as image, point cloud, and 3D geometry data of objects. Savva et al. found an area for interaction behavior in an environment through action maps created by using motion capture data and geometry information of the captured space [4]. By using geometric information and its structural features, Hu et al. analyzed the functionality of the object [5]. Kim et al. created an appropriate interaction pose by using a stochastic model learned from 3D models [6]. Grabner et al. searched suitable positions for a static sitting pose in a 3D-scanned space, after learning the geometry information of sittable 3D models [7]. Kang and Lee proposed a method to create contact poses for a given environment by using contact-related features extracted from sample poses [12].

In most studies, the interaction-contextual information is learned by using the observation data on the interaction between the actor and object, but this approach has some limitations. The acquisition of the observation data requires significant time, effort and state-of-the art capturing techniques, and thus the data is not widely available. The insufficiency of data may cause the learned model to be overfitted to the training data. From a computer graphics perspective, we need to be able to analyze the virtual objects of which shape may not be easily found in the real world. In this case, the observation-based approach cannot be applied in a straightforward manner. In addition, the existing

*Corresponding author
Email address: sunghee.lee@kaist.ac.kr (Changgu Kang and Sung-Hee Lee)

machine learning-based approaches can only generate static contextual information on the human-object interaction (e.g., contact pose rather than contact motion).

In order to overcome these limitations, we propose a novel method to reconstruct synthetic scenes from captured motions, and to analyze the interaction-contextual information of the synthetic scenes and motions. For the scene reconstruction we search for 3D objects from a 3D object database that match the capture motions and place them in appropriate orientations and positions. Our method is characterized by constructing abstract objects, which contain contact-related information inferred from capture motions. Through scene analysis, we obtain contextual information on the interaction, including *interaction behavior*, *functionality of an object*, and *the interaction space of an object*.

Interaction behavior: Motion data, if it contains object interaction, generally consists of a sequence of actions: moving to a target - transition - interaction with an object - transition - moving out (or to another target). Transition is a preparatory or a finishing action before or after the actual interaction occurs, and thus is an important part of human-object interaction. We developed a method to segment a motion sequence into this series of actions. The proposed method defines the suitable feature vector reflecting the human-object interaction motion and obtains a Gaussian Process Regressor that classifies the actions.

Functionality: We define an object's functionality as the interaction motions related with the object, such as "sitting down" and "lying". In this paper, the functionality of an object is represented as the pair of the object's surface and the contacting body part, e.g., seat-hip. The information on an object's functionality can be used to create a new interaction motion for the object. The functionality of an object is derived straightforwardly from the process of reconstructing synthetic scenes.

Interaction space: In order for a human to interact with an object, some volume of an empty space surrounding the object is necessary so that a user can transition to the object and make poses with respect to the object. The information on the interaction space is useful for designing the object layout in a scene as well as for creating or modifying motions that interact with the object. We developed a method to compute the interaction space of an object by finding the spaces where the transition and interaction actions can be applied to an object through random sampling.

The analyzed information by our method can be used for many purposes. Interaction space and functionality of objects are useful for automatic arrangement of 3D objects according to a given human-object interaction scenario. In addition, the interaction space of an object can be used to generate interaction motions customized to that object.

The remaining part of the paper proceeds as follows. We discuss related work in Secion 2. Section 3 provides an overview of the proposed framework, and Sections 4 and 5 detail the procedure for reconstructing abstract scenes and virtual scenes. Section 6 reports our experiment. Section 7 discusses the advantages and limitations of our work and concludes the paper.

2. Related Work

Our goal is to reconstruct synthetic scenes from captured motions and analyze interaction context with the scene and the motion. This section reviews previous studies closely related to our method.

Scene Reconstruction. With the advent of low-cost depth cameras, many researchers have conducted studies on reconstructing scenes from RGB-D data [1]. Given low quality data generated by a 3D scanner, the method of [1] reconstructs synthetic scenes by using prior knowledge learned from a scene database. Firstly, as an intermediate representation, 3D scan data is represented by the scene template in which geometric and activity properties are also embedded. The activity is expressed as a continuous distribution on a 2D floor. Then, a scoring function selects suitable 3D objects that match the scene template. [3] reconstructs plausible 3D scenes from low-quality RGB-D data based on the contextual relationship between 3D objects learned from the scene database. [13] segments 3D space represented as RGB-D data into semantic regions such as sofa, floor, bed, and background, and then creates synthetic scenes by retrieving 3D models that fit the semantic regions. [14] reconstructs a scene by extracting dominant planes from the scanned environment and matching objects to the planes. In the preprocessing step, the objects are segmented into planes for efficient matching tests.

[3, 13, 14, 8] define relationships among objects or parts by using geometry information in order to reconstruct scenes, whereas [1] defines interaction information between an object and an actor. They all require a pre-processing or training step and an object database. In other lines of research, [2] proposed a method to rearrange objects by using relationship between objects, and [15] developed a method to combine multiple virtual scenes to create a complex scene.

Interaction behavior. [16, 17] investigated methods to create an appropriate human pose against an input object or environment. The fitness of a pose with respect to an environment is measured by a probabilistic model, which has been learned from observed human activity or downloaded 3D models. [6] also developed a method to create interaction poses for an arbitrarily given object. This is realized by a stochastic model trained with a set of sample objects to which appropriate contact points are annotated by hand. The method of [12] finds a set of candidate contact points from input objects and then searches for human poses that are physically balanced while realizing the contact points. [18] proposed a 4D human-object interaction model which defines relationships between an object and an actor for a specific event or object recognition. The 4D human-object interaction model is a 3D spatial domain which includes a type of human pose and objects, and an 1D temporal domain which includes a continuous chronological order of the events (e.g., approach the dispenser, fetch water, and leave the dispenser). [2] also models object-object and human-object relations. The objects are rearranged to preserve the relations between the objects for a particular scene and generate an appropriate interaction pose with respect to the object. In order to model human-object relations, [2] developed the Infinite Latent Conditional Random Field.

Functionality of objects. [1] finds the functionality and the interaction space in a given scene by using an activity model, which is created by using manually tagged interaction-related data to each object. [19] finds corresponding parts with similar function between the objects by computing shape similarity between objects through Graph Kernels [20]. [5] analyses the functionality of the object using geometric information and structural features of an object. They extract the Interaction Bisector Surface [21] and Interaction Region of the objects in a given scene, and structurize objects as a type of a tree according to functionality of the objects.

Interaction space of objects. [22] estimates 3D geometry from 2D images and searches feasible positions in a scene for the postures in a pose database. [7] searches proper positions for a sitting posture in a 3D scanned space by using geometric information previously trained from sample chair models. [4] studies the functionality of a physical scene from the observed behavior of people in the scene. The trained model, called the action map, estimates the probability of interaction on the surfaces of the 3D scene and finds a feasible space for the interaction behavior. [23] proposed a method to place additional objects that are appropriate to the human actions available in an initially given sparse set of objects. By learning the relations among human poses, related object categories, and spatial configuration of the objects from annotated photos, their method allows for constructing scenes that are *behaviorally consistent*. By contrast, our work is focused on creating *contact consistent* scenes without resorting to learned data. An additional difference is that we consider the transition behavior of human movement with respect to objects whereas [23] considers only static poses. [24] proposed a descriptor for interaction between human and object, and among multiple objects. The descriptor has strong advantages in that it covers a wide variety of scenarios including fluid-solid interaction and dynamic interaction. In contrast, we focus on human-object interaction. Unlike [24] that does not differentiate the interaction context, our method subdivides interaction behaviors into approach, interaction, and release. Additionally, our method identifies the volume of the space used for the interaction between a human and an object.

	Previous work	Our work
Input data	RGB-D data	Motion data
Interaction behavior	Static pose	Motion data
Interaction space	Position for an interaction pose	3D space for an interaction motion

Table 1: Comparison between previous work [1, 3, 13] and ours.

Table 1 shows the key differences between previous work and ours. Previous work [1, 3, 13] use RGB-D data of a partial scene as input data, while our method uses captured motion in a scene as input. Our approach can only reconstruct objects that a human interacts with, but [1, 3, 13] have an advantage in that by using RGB-D data the whole scene is covered. However, processing RGB-D data is rather complicated, especially if data quality is low [13], and may require additional information such as manually drawn strokes. In contrast, the process of our work is simple and does not require additional information. Regarding the interaction behaviors, previous work considered only static poses while our method deals with a motion sequence. With respect to the interaction space, previous work only generated static poses for an object, but our method considers the 3D volume space for the interaction space. [18] also considers the transition behavior in a similar manner to us, but they only consider temporal separations for transition behavior from a given motion without spatially considering whether this transition action can be performed or not.

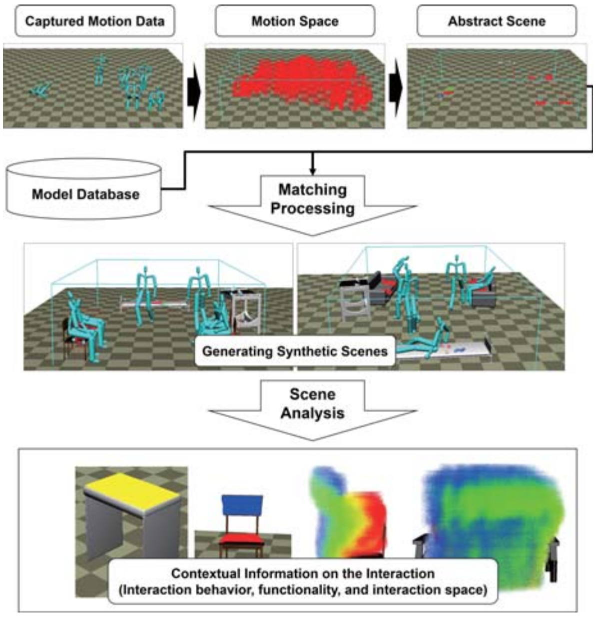


Figure 1: Overall flow of our method.

3. Overview

The proposed method reconstructs synthetic scenes suitable for the captured motions and obtains interaction context by analyzing the reconstructed synthetic scenes. Figure 1 shows the overall process of the proposed method.

Input to our method is a 3D object database and captured human motions. We assume that virtual 3D objects and human models are scaled to the actual object and human sizes. Our method aims to reconstruct a scene with only those objects with which a human makes enough amount of contact. To this end, we first estimate *contact blocks* from captured motions, small volumetric units of objects which a human makes contact with, and then construct in the order of planes, abstract objects, and abstract scenes. Actual scenes are composed by searching from a 3D object database for 3D objects that match the abstract objects and placing them in appropriate orientations and positions. The object retrieval is performed in two stages, the initial broad-phase filtering based on the heights and normal directions of the constructed planes, followed by the detailed matching between the abstract objects and 3D objects.

The interaction context is then obtained by the matched objects, their corresponding abstract objects, and human motions. In particular, we segment interaction-related subsequences from the captured motion sequence, and estimate the interaction space around

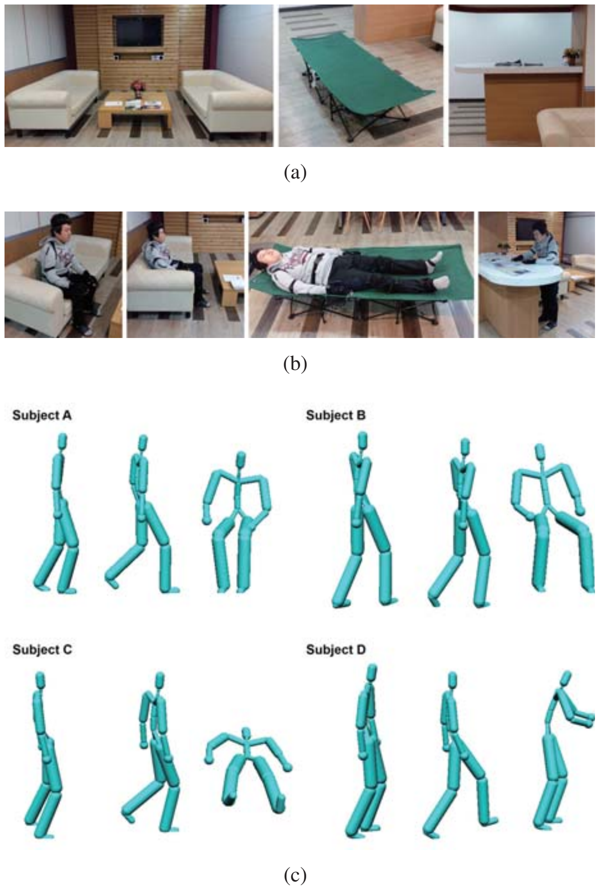


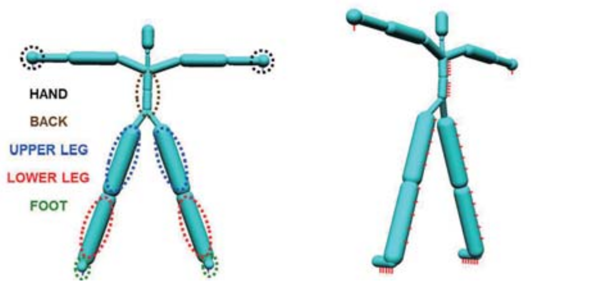
Figure 2: (a) The environment where motions are captured includes two sofas, a camp bed, and a high table. (b) Snapshots of input motions: sitting, sitting while leaning, lying, reading a newspaper. (c) Captured motions: sitting (subject A), sitting with leaning (subject B), lying (subject C), and putting hands on a table (subject D).

the object. We also register the contacting body parts to the object planes.

4. Scene Reconstruction

4.1. Data Acquisition

Motions were captured with a Perception Neuron (https://neuronmocap.com/) device, a motion capture device using inertial sensors. We captured four types of motions, i.e., sitting (subject A), sitting with leaning (subject B), lying (subject C), and putting hands on a table (subject D). Sitting and sitting with learning motions are captured twice, each captured in a different location inside a sofa. A total of 6 motion sequences were used for the experiment. Figure 2 shows the environment, screenshots of the capturing session, and the ob-



(a) (b)

Figure 3: (a) The character model used in our experiment has 21 body parts. We assume that a human contacts the environment with only a subset of body parts: the hands, back, legs, and feet. (b) Markers and their normal directions.

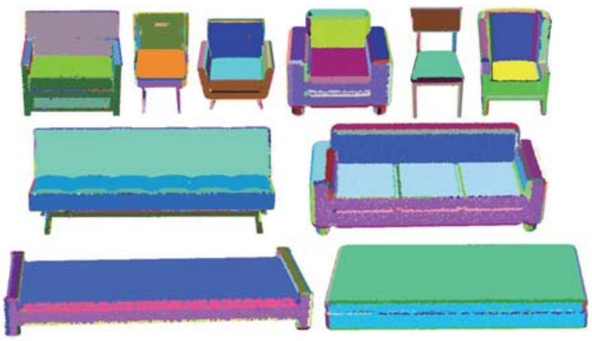


Figure 4: Objects are segmented into planes to test matching with abstract objects.

tained motion data. We did not capture the geometry of real objects.

4.2. Actor Representation

Figure 3 shows the character skeleton used in our experiment. The skeleton model provided by the software associated with the motion capture device was too detailed to stably extract contact information, and thus we retargeted the motion to a simplified skeleton model (Fig. 3) using Maya. The hands are modeled as spheres, and all others as capsules. We assume that a human makes contact with the environment only on a particular side of some body parts. To model this, we attach *markers* for contact points on the hands, back, legs, and feet as shown in Figure 3 (b). We measure the velocity profiles of the candidate contact points from the motion data by using the finite difference method and the Kalman filter [25].

(a) (b) (c)

Figure 5: This figure conceptually illustrates the process to obtain groups of contact blocks. (a) The boundary of the motion space is a candidate space that can be the surface area of an object. The whole space is discretized into blocks, and those blocks containing the motion boundary are marked as surface blocks. (b) Among the surface blocks, we identify contact blocks that are likely to be contacted by human. (c) Contact blocks that share the same contacting body part form a group.

4.3. Object Representation

Given a 3D mesh of an object to be included in the object database, we analyze its shapes and precompute some attributes that are used for matching the object with an input motion. Since the matching process is performed with respect to the surface planes of the 3D object, we first segment the surface planes from the object. For this, we voxelize the object by using a uniform grid and estimate the normal direction of the surface cells. Then the adjacent cells with similar normals are clustered as a plane. Figure 4 shows the segmentation results for some indoor objects.

Subsequently, in order to support an initial test that quickly filters out un-matchable objects, we precompute a filter map per object that indicates the normal directions and heights that the planes of an object can have. Specifically, a filter map is represented by a Boolean table in which the X-axis denotes the polar angles of the plane normals divided into intervals, and the Y-axis is the intervals of the plane heights. An entry (x, y) is marked true if the object has a plane with a normal and height in the ranges indicated by x and y. In our experiment, the polar angles are divided by 20 degrees into 9 number of cells, and the plane heights are divided by 0.05 meters into 35 number of cells.

4.4. Abstract Scene

To reconstruct scenes from motion data, it is necessary to estimate the existence of objects from the motion data. Certainly there could be many objects in an environment in which motion is captured, and some make contact with a human while others do not. We only attempt to reconstruct those objects that a human interacts with and hence can be estimated purely from motion, and ignore other objects. In order to simplify the processing of the spatial data, we discretize the 3D space that a motion belongs to with a uniform grid. Each cell

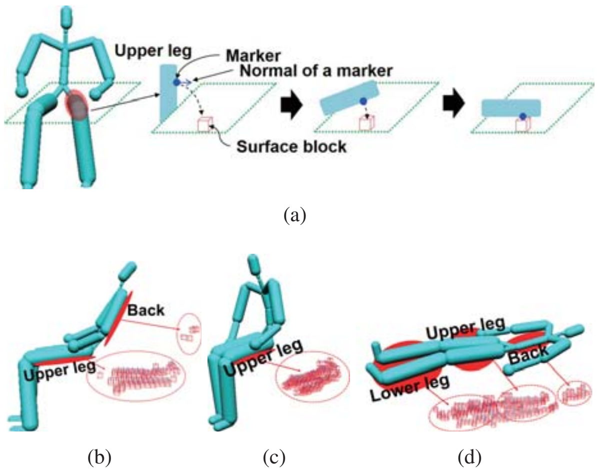


Figure 6: (a) An example of a condition under which a surface block is identified as a contact block. (b-d) The surface blocks collided with the character's markers. Normals of surface blocks are estimated from the normals of contacting markers.

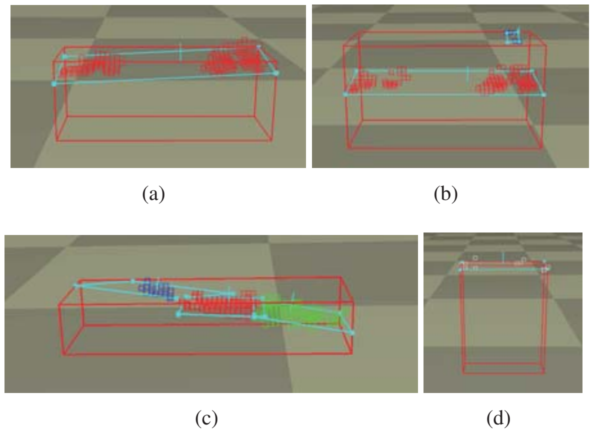


Figure 7: (a-d) show the abstract objects created from subjects A and A, subjects A and B, subject C, and subject D, respectively. The contact blocks are colored per body parts associated with the blocks. Red-hip, green-leg, blue-back, white-hand. Planes in cyan: planes made of contact blocks of the same functionality. Note that the contact blocks in (c) are slanted due to the inclination of a captured human pose.

in a grid, dubbed a *block* in this paper, serves as a unit for the spatial processing. The abstract scene is constructed from the blocks that a human interacts with through contact. A set of such blocks forms a plane, and then nearby planes form an abstract object. An abstract scene is composed of the abstract objects.

Contact Block Extraction. We analyze the motion and extract the points at which a human contacts objects. Let us call the blocks, including the points, *contact blocks*. Our first goal is to identify the contact blocks. Human cannot penetrate into solid objects, and the contact between a human and an object occurs at the boundary of the two entities. Let us find the *motion space*, a Cartesian subspace that a human performing a motion occupies, and then the motion blocks are those blocks in the motion space. The motion blocks are collected by examining the collision of a block with oriented bounding boxes of a character performing the input motion. We can also define the *surface blocks*, which are the blocks on the boundary of the motion blocks (Fig. 5). Note that the surface blocks are on the surface of a motion space, not on the surface of the environment objects. Then, the contact points must be included in the surface blocks, and we have only to examine each surface block to identify the contact blocks.

There can be numerous ways that a human can interact with an object through touch. She can touch an object at a static point, but also can rub or stroke on the object's surface. Thus, it is not straightforward to determine whether or not a human touches an object only by looking at her motions. After examining various motion data involving object interaction, we have chosen a heuristic that a contact block is a surface block satisfying all three conditions below:

- It collides with a marker on a human.
- At a certain time instance, the direction of the marker's velocity is parallel to the outward normal direction of the surface block, i.e., the maker approaches the imaginary surface in a perpendicular direction.
- At a certain time instance, the speed of the maker is nearly zero.

Figure 6 shows a case that a surface block is selected to be a contact block. When the character is sitting, a marker on the upper leg approaches a surface block. At this time, the directions of the surface marker's normal and the marker's velocity become close to parallel, and the velocity approaches zero (Fig. 6 (a)). In such a case, the surface block is categorized as a contact block. The normal of a contact block is calculated as the average of the normals of the collided markers on a surface block. Figures 6 (b)-(d) show the surface blocks collided with the character's markers. The normal of the surface blocks is similar to the direction of the corresponding part.

Abstract Objects and Scenes. Next, we extract planes from the contact blocks. To this end, the contact blocks that are close to each other, have similar normal directions, and collide with the same body part, are put into the same group. Groups that have less than three elements are considered as outliers and are eliminated. One intricacy originates from the fact that two or more people can interact with the same object at the same time. For instance, imagine that two persons sit on a sofa. By only looking at their motions, we cannot differentiate this scenario from one where two persons sit on two separate chairs. Our strategy against this redundancy is to adopt multiple solutions for the groups: If the contact blocks in a group are generated by more than one motion, we generate additional groups by dividing the group per source motion.

Subsequently, we generate planes per group. The position and normal direction of the plane is first estimated by using the principal component analysis of the contact blocks, and four corner points are generated as the min/max boundary. PCA is performed against the 3D positions of the contact blocks, and then the third component is selected as the normal direction. At this stage, the outward direction of the normal is not yet identified, which is achieved by selecting a direction closer to the average normal of the surface blocks. It is to be noted that an enough number of the contact blocks need to be included in PCA. Otherwise, noise in a few blocks may induce instability in the plane estimation, as will be discussed in Sec. 6.

We assume that each motion interacts with only one object. Therefore, the planes created from the same motion among planes are grouped as one object. In the case that two persons share an object, planes created from the two motions constitute one object. An abstract object contains the following attributes in order to find matching objects and help estimate the interaction-related information of the object.

- Object volume, calculated as the volume of its bounding box.
- The positions and normal directions of planes.
- The body parts colliding with a plane.

A bounding box is determined by the min/max positions of all the contact blocks that make up the object while the minimum height is always assumed to be zero. The body parts colliding with a plane is stored as the functionality of surface of matched 3D objects. Finally, the abstract objects constitute an abstract scene. Figure 7 shows abstract objects created by input motions.

4.5. Synthetic Scene Reconstruction

The synthetic scenes are reconstructed by finding 3D objects from a database that match the abstract objects. In this process, we search for the object m_i that matches the abstract object from the 3D object database $M = \{m_1, \ldots, m_N\}$.

The matching process is performed with respect to the surface planes of the 3D objects, and it is performed in two stages.

First, for each $m_i \in M$, we perform a pre-validation check by using its filter map. We collect a list of (polar angle, height) pairs of all planes of the abstract object, and check whether all the filter map's cells that correspond to the pairs are true. If false, which means that there is certainly at least one abstract object plane that cannot be matched to m_i, m_i is discarded from the further checking. If true, we proceed to the fine scale matching test.

The basic tool for the fine scale matching process is the cost function that measures the fitness of an object m_i with a given pose $T \in SE(3)$ to an abstract object:

$$\begin{aligned} E(T,m_i) &= w_1 f_c + w_2 f_v + w_3 f_p && (1)\\ f_c &= \sum_j \delta_c(x_j) && (2)\\ f_v &= \frac{1}{\sum_j \delta_v(x_j) + \varepsilon} && (3)\\ f_p &= dif_{dis}(p_a, p_o) + dif_{deg}(p_a, p_o) && (4) \end{aligned}$$

The collision cost f_c discourages collision between the object and the motion space. The variable $\delta_c(x_j)$ is 1 if the position x_j of an object cell collides with a motion block, and 0 otherwise. The term f_v drives the object's volume to overlap the volume of the abstract object. The variable $\delta_v(x_j)$ is 1 if x_j is included in the volume of the abstract object, and 0 otherwise. A small positive number ε prevents f_v from overflowing. For f_p, we first select p_o at the closest distance to p_a. The distance $dif_{dis}(p_a, p_o)$ is calculated as the average of the closest distance from each block of p_a to p_o. As for the angular difference $dif_{deg}(p_a, p_o)$, we measure the average of the dot products between the normals of p_a and p_o. Weight w_i controls the importance of each cost.

To obtain $E(T, m_i)$, we place the 3D object at the center of an abstract object, and compute the optimal transformation matrix T of an object m_i in terms of the cost $E(T, m_i)$ using the CMA-ES, a derivative-free optimization algorithm for multi-objective optimization [26]. The sampling range for the transformation matrix is set as follows: scale [0.9, 1.1] , translation [-15, 15] cm, and vertical rotation [0, 2π]. Then we collect a set of matched objects with those objects that have a

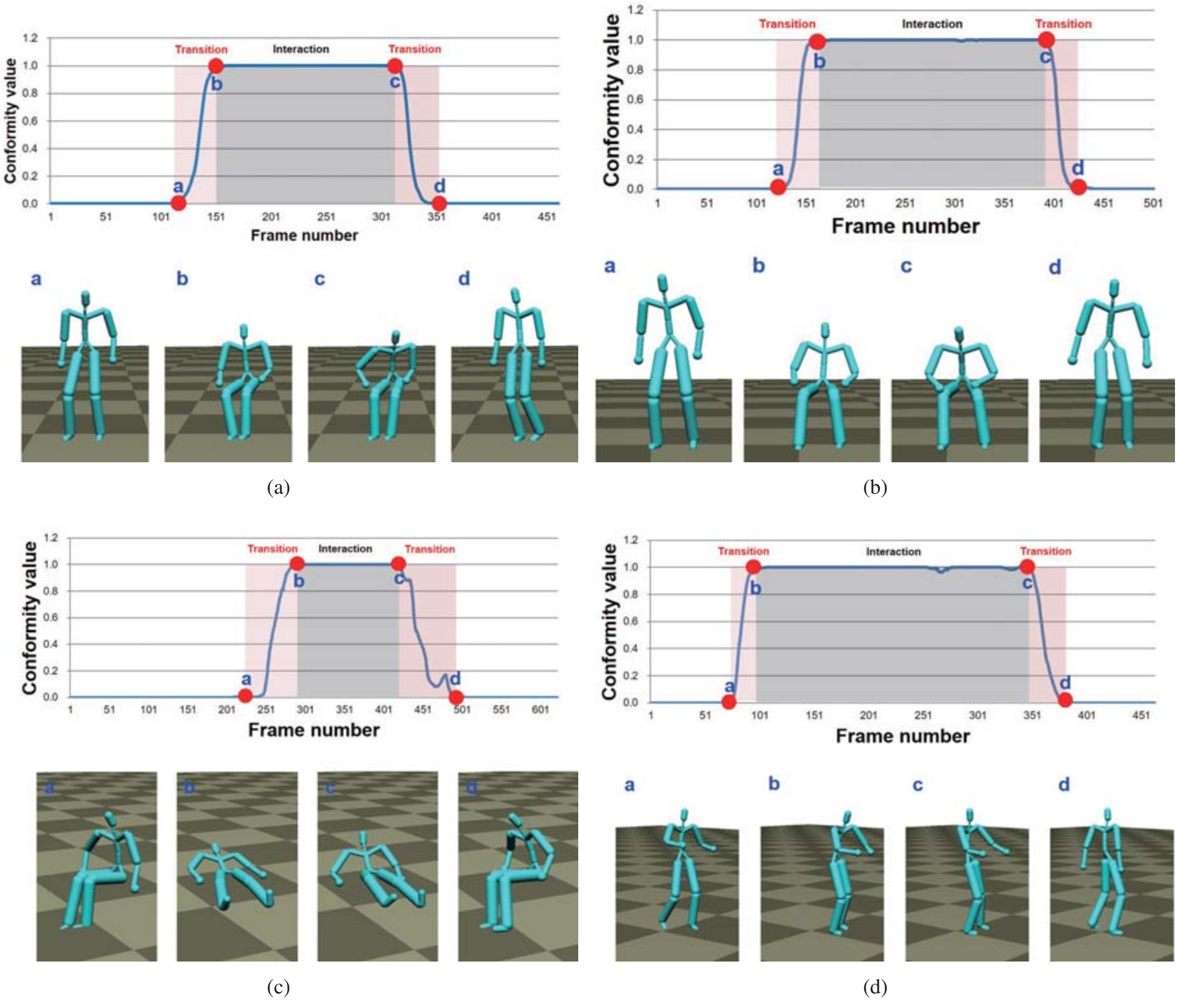


Figure 8: The top shows conformity value according to the number of frames for four types of motion data. The bottom shows the postures of segmented frame data. A pose is determined to belong to a transition if $1 \times 10^{-4} < y < 0.9$.

cost lower than a certain threshold. The reconstructed synthetic scenes are created by the combination of the matched objects per abstract object.

Note that our goal is to find 3D objects that allow for the input motion, not to retrieve objects of which representative function best matches the given motions. Therefore, a bed can be retrieved if its height matches that of an abstract object that has been constructed from sitting motions. We assumed in our experiment that the size of 3D objects in the database does not vary significantly. This is because scaling of an object may lead to undesirable object matching such as a human lie on a giant stool. However, appropriate scaling of the objects will increase the number of matched objects for an abstract object.

5. Scene Analysis

By analyzing the reconstructed synthetic scenes, our goal is to extract the interaction-related motion from the whole motion sequence and to obtain the interaction space of the objects. Additionally, we register the information of the contacting body part to the object planes. Since the abstract object has the attributes of the contacting plane and body part, we have only to transfer the information to the object's planes that correspond to the planes of the abstract object. We detail how other goals are achieved next.

5.1. Interaction behavior

In general, a motion sequence related with object interaction consists of several sub-sequences: (moving to a target point)-(transition)-(interaction)-(transition)-(moving out), and our goal is the segment them from the whole motion sequence.

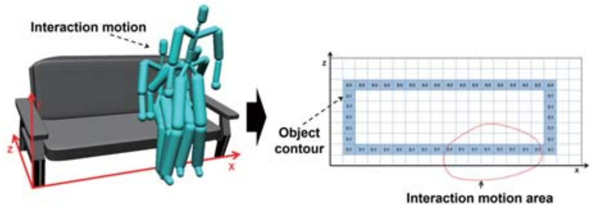
Our approach to this problem is to use a regression model: we train a function $y = f(\mathbf{x})$ that estimates the conformity value y of a feature vector $\mathbf{x}$ from an input pose with respect to a target object and extract the poses that have conformity higher than a certain threshold. In the process of creating abstract objects in Sec. 4.4, we detect the poses in contact with objects. Let us call them *canonical poses*. We train a regression function such that the conformity of the poses similar to the canonical poses is 1, and that of the dissimilar pose is 0. This approach enjoys the benefit of a simple structure for the regressor as it only deals with the similarity between poses and does not need to consider the temporal connectivity of the poses. We designed the feature vector to take the following aspects into account:

- Interacting behavior should be close enough to an object.
- Interacting behavior should have similar movement characteristics with respect to the contacting body parts.

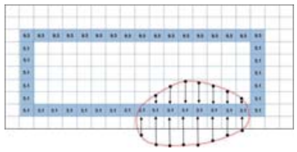
We employ the Gaussian Process (GP) regression model [27] for the regressor (Eq. 5). After training the motion data ($\mathbf{x}$) of the interaction interval as $y = 1$, $y \in [0,1]$ is determined according to the degree of similarity to the frame data($\mathbf{x}$) in the motion. Training data is simple: it includes the pairs $(\mathbf{x}, y = 1)$ for the canonical poses. The feature vector $\mathbf{x}$ is defined as a multi-dimensional vector in which the first three elements are the relative position ($\in R^3$) of the center of mass of a human with respect to the reference frame of an object, and the remaining elements are the velocities ($\in R^3$) of the markers in contact with the objects in the canonical poses. For instance, if n markers get in contact with an object in a motion sequence, the dimension of $\mathbf{x}$ is $3+3n$. The kernel function for the covariance matrix is modeled as the radial basis function (Eq. 6), of which parameter σ is obtained by maximizing the log likelihood $\log p(g|x,\sigma)$ (Eq. 7) with the Nelder-Mead simplex method [28]. Therefore, the GP is only trained to output 1 for $\mathbf{x}$ of the canonical poses, and the conformity values of other sub-sequences are determined from the covariance matrix.

$$y = GPR(\mathbf{x}) \tag{5}$$

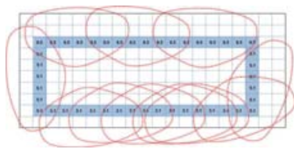
$$k(\mathbf{x},\mathbf{x}') = \exp\left[\frac{-||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right] \tag{6}$$



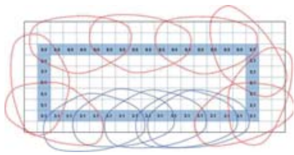
(a) Object and interaction behavior projected on a horizontal plane.



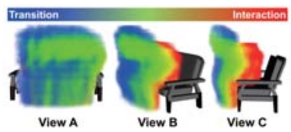
(b) Closest boundary cells to sample points on the boundary of projected motion space.



(c) Projected motion spaces for sample transformations.



(d) Feasible transformations (blue).



(e) Interaction space, colored as the conformity value of a cell, from 0 (blue) to 1 (red).

Figure 9: Procedure to estimate an object's interaction space from its interaction motion.

$$\log\ p(g|\mathbf{x},\sigma) = -\frac{1}{2}g^T K^{-1} g - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{7}$$

Figure 8 shows the result of extracting interaction motion from the motion data. One can see that transition and interaction motion are appropriately segmented according to conformity value. The poses corresponding to the change of states are shown on the bottom.

5.2. Interaction space of objects

We describe a method to estimate the interaction space of an object from its related interaction behavior obtained in the previous Subsection. Our method is based on two assumptions:

(1) The interaction behavior with respect to an object occurs in the interaction space.
(2) The interaction behavior can be rigid-transformed to a region in an object that is geometrically similar to the region taken by the original interaction behavior.

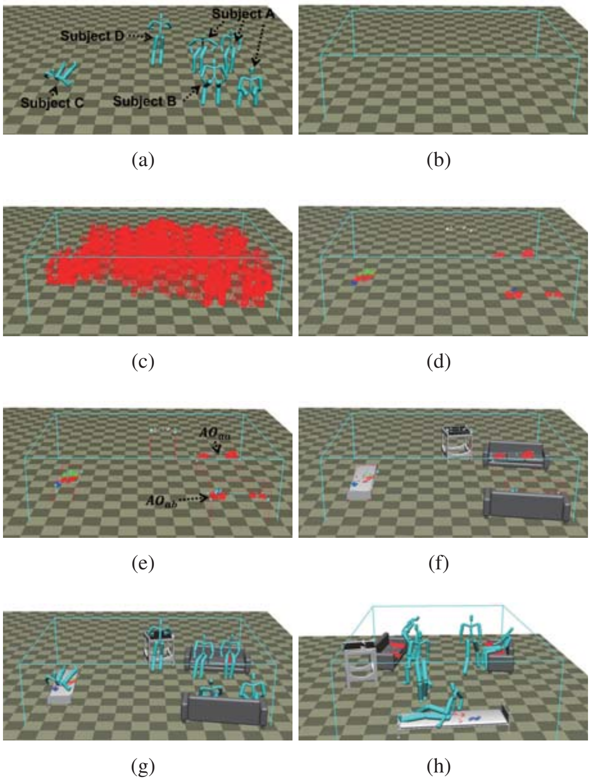


Figure 10: (a) to (d) show the process of making surface blocks from captured motions.(d) white: hand-contacted blocks, red: hip-contacted blocks, green: leg-contacted blocks, and blue: back-contacted blocks. See Fig. 11 (a) for the enlarged image. (e) and (f) show the result of matching 3D objects to abstract objects. AO_{aa} is an abstract object created from motions for subjects A and A. AO_{ab} is an abstract object created from motions for subjects A and B. (g) and (h) show the generated scenes.

Figure 9 shows the procedure to estimate an object's interaction space from the interaction behavior. The main point of the method is to find the proper range of rigid transformations that an interaction motion can take with respect to an object, and we achieve this by associating the motion space for the interaction behavior with the information in its spatial relationship with the object. To this end, we first project the object onto a horizontal 2D grid, find the boundary cells, and append the object's height at the location of the boundary cells (Fig. 9 (a)). The position and height value of the boundary cells serve as the feature of the object's shape. In addition, we project the motion space of the interaction motion onto the same 2D grid. Then, the spatial relation between the interaction behavior and the object is defined as the distance from the projected motion space to the closest boundary cell and its height value. This is realized by collecting a set of sample points on the boundary of the projected motion space, followed by finding the closest boundary cells and then storing its distance and height value. Note that we store the signed distance (negative distance to the sample points inside the object's boundary) in order to differentiate whether a sample point should be inside the object or not. The distance d_i^o and height value h_i^o for each sample point i are used to find suitable transformation of the interaction behavior.

We collect the feasible transformations of the interaction motion by random sampling. To increase the hit rate of the sampling, the sampling is bounded to the area obtained by extending the object's boundary by the maximum length between the sample points (Fig. 9 (c)). For each test transformation T, we find the closest boundary cell from every transformed sample point and compute its signed distance $d_i(T)$ and read its height value $h_i(T)$. The cost of the transformation is measured by

$$c(T) = \frac{1}{N}\sum_{i=1}^{N}(d_i^o - d_i(T))^2 + (h_i^o - h_i(T))^2 \quad (8)$$

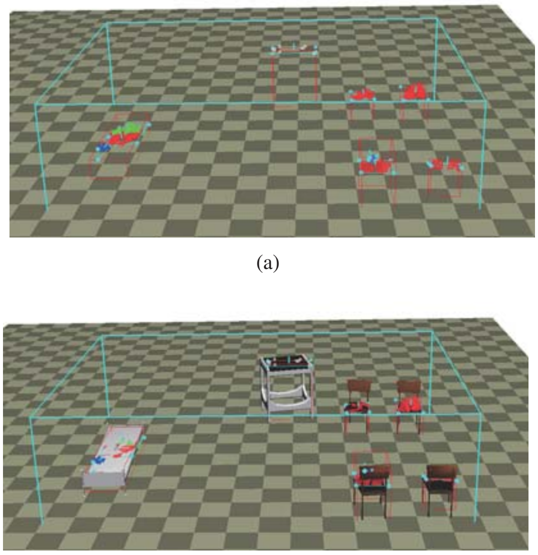
where N is the number of sample points.

The interaction space of an object is constructed as the union of the motion spaces of the transformed interaction behaviors that have $c(T)$ less than a threshold. We also assign a conformity value from 0 to 1 to the cells in the interaction space as the maximum conformity value of the pose passing through the cell (Fig. 9 (e)).

6. Experiments

We constructed a 3D object database, which consists of a total of 100 objects of chairs, sofas, beds, tables, cabinets and desks. Some chairs have backrests and some do not. Every object was downloaded from the Internet (http://tf3dm.com/ and https://archive3d.net/), and roughly scaled to match the size of the human subject.

Figure 10 shows the process of creating synthetic scenes from motion. We first compute the bounding box from the motion data and perform voxelization (the size of voxel: 2.8cm) (Figs. (a, b)). Next, we exclude blocks colliding with motions (Fig. (c)) and extract surface blocks by the method in Sec. 4.5. If surface blocks form a group, they create an abstract plane. A group of abstract planes create an abstract object (Fig. (e)). Finally, a synthetic scene is reconstructed through matching 3D objects in the database to abstract objects (Figs. (g) and (h)). We empirically set $w_1 = 0.001$, $w_2 = 0.1$, and



(a)

(b)

Figure 11: (a) When the contact blocks created by different motions are not merged, multiple abstract objects are reconstructed instead of one large object. (b) The separated abstract objects have been matched to multiple chairs.

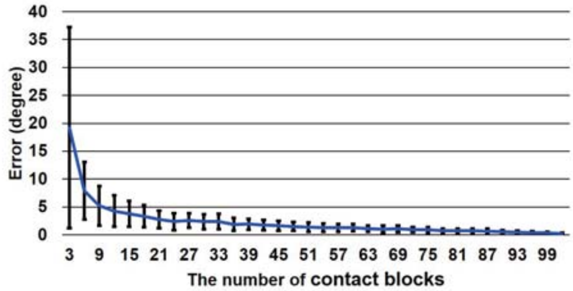
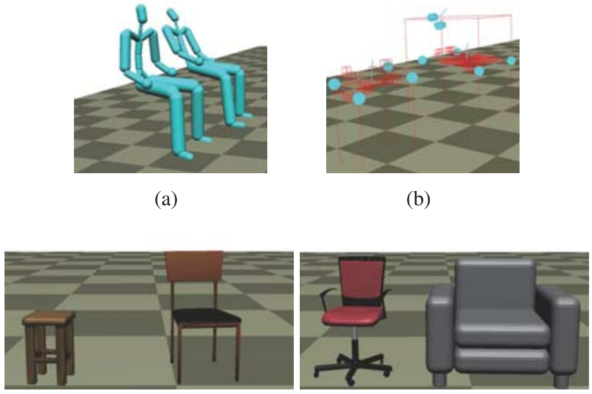


Figure 12: Mean and standard deviation of the normal estimation error per the number of contact blocks used for the estimation, for the motion of Subject A.



(a) (b)

(c) (d)

Figure 13: Different types of chairs are reconstructed by the motions of the Subject A and B.

$w_3 = 0.04$ for Eq. 1, and an object is determined to match the abstract object if the cost is less than a threshold, which ranges from 0.25 to 0.32 depending on the input motion.

Figure 11 shows the results of the abstract scene when the contact blocks created by different motions are not merged, and each create separate abstract objects. In this case, four chairs are reconstructed, which is in contrast with the case where the contact blocks are merged and generate a united abstract object, such as the sofas in Fig. 10 (e) and (f).

In order to test the sensitivity of the number of contact blocks for the estimation of the normal plane, we measured the mean and the standard deviation of the error, which is calculated as the angle between the normal direction estimated by using only a certain number of randomly sampled contact points from the normal direction obtained by using all contact points. Figure 12 shows the error for the sitting motion (Subject A) that has a total of 104 extracted contact blocks. It shows that the average error decreases rapidly with the number of contact blocks and falls below 10 degrees when six or more contact blocks are used.

Figure 13 shows the results generated from a sitting motion (subject A) and a sitting with leaning motion (subject B). In the case of sitting with leaning, an abstract plane is created to support the back, and thus a 3D model with a backrest is necessary (the chair in Fig. (d) and the sofa in Fig. (e)). Otherwise, the 3D models selected may have a backrest or not (the stool in Fig. (d) and the chair in Fig. (e)).

Figure 14 shows the reconstructed synthetic scenes. Figures 14 (a-h) are the cases when the contact blocks from different motions create abstract objects separately, and Figs. 14 (i-p) are when contact blocks from different motions are shared.

We use three sitting motions and one sitting with leaning motion. Two abstract objects are created for sitting behavior as shown in Fig. 10 (a) and (e). One abstract object (AO_{aa}) is created by sharing motions of subject A and subject A, and another one (AO_{ab}) is created by sharing the motions of subject A and subject B. The abstract objects are matched to the sofa models, but the number of matched objects is different. AO_{ab} needs back support, so the right sofa of Fig. 14 (i) without a backrest cannot be matched to AO_{ab}. For the same

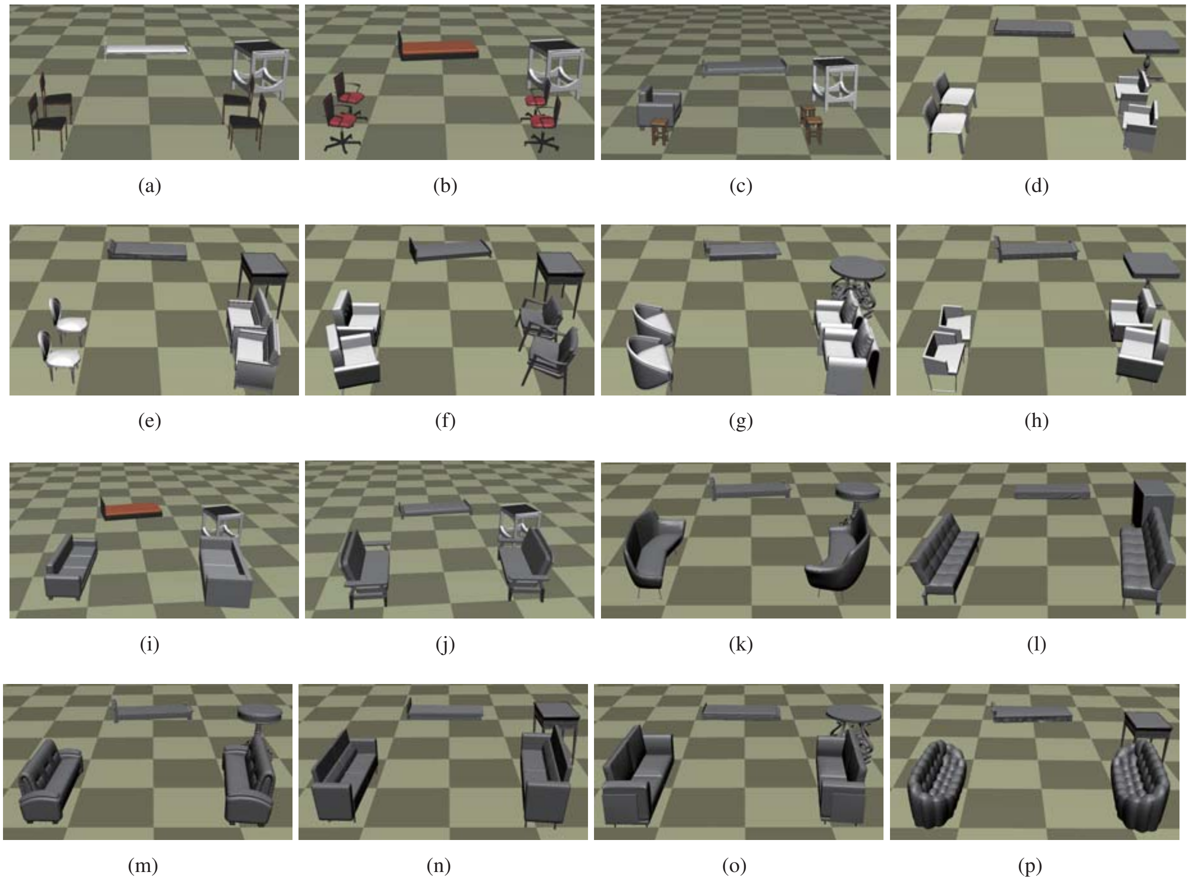


Figure 14: Various synthetic scenes generated with input motion. (a-h) 3D objects generated separately for each abstract object. (i-p) 3D objects generated with shared abstract object.

	Scene		Subject (only individual scene)			
	Sharing	Individual	A	B	C	D
# abstract planes	7	9	1	2	3	1
# abstract objects	4	6	1	1	1	1
# matched 3D objects	56	67	28	25	35	9
# filtered 3D objects	-	-	29	30	40	62

Table 2: The number of abstract planes, abstract objects, matched 3D objects generated in our experiment.

Subject	A	B	C	D
# Voxels	11K	15K	19K	14K
Creating an abstract object (sec.)	0.67	0.88	1.56	0.77
Matching 3D objects (sec.)	0.44	0.38	0.31	0.39

Table 3: The number of voxels used for the processing and the durations measured for each processing step. Creating an abstract object: time for creating an abstract object from motions. Matching 3D objects: average time for matching an abstract object per 3D object.

reason, the number of objects matched to Subject B is smaller than to Subject A.

Table 2 shows the number of abstract planes, abstract objects, and matched 3D objects generated in our experiment. Figure 15 shows the results of 3D objects matching from sitting motion in order of minimum cost computed by Equation 1. Table 3 shows time for creating an abstract object and matching 3D objects. Time for matching 3D objects is proportional to the size of database because we compare with every object in the database. Simple rules that cull out unmatched object can save time for matching. Figure 16 shows models matched from each motions.

Figure 17 visualizes of functionality of surfaces with respect to the interaction of the 3D model. Different colors indicate different functionalities. Figures 18 shows the results of estimating interaction spaces for 3D objects. We can see that the conformity value is high (red color) in the space close to the surface where interac-

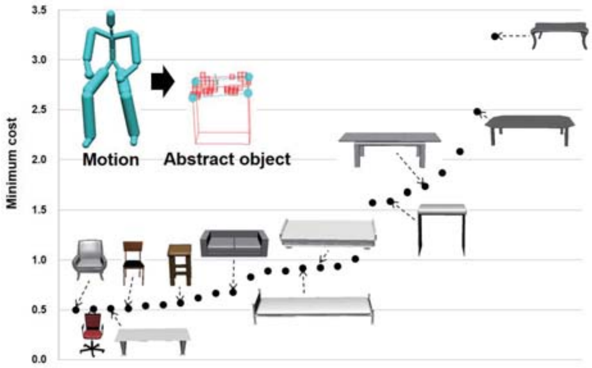
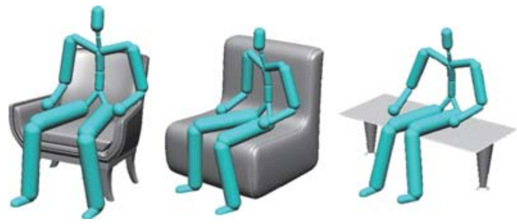


Figure 15: Minimum cost computed from subject A (sitting motion) for selected 3D objects. The minimum cost computed by Eq. 1 increases in the order of chair, sofa, bed, and table. The object with the third lowest cost was chosen as a table because it is low enough to sit on, as shown in Fig. 16 (a).



(a)

(b)

Figure 16: Selected models from (a) sitting motion and (b) hands-putting motion

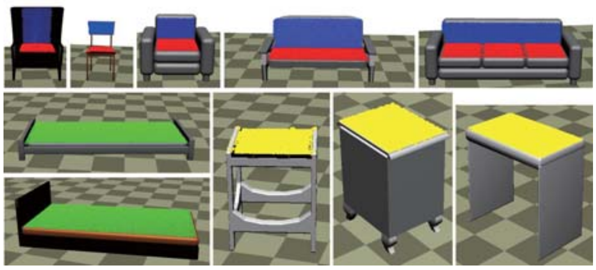


Figure 17: The functionality of 3D model (red: upper leg (hip) support; blue: back support; yellow: hand support; green: back, upper leg, and lower leg support).

tions take place. Figure 18 (c) shows a case where the interaction space completely surrounds the object because the stool has an axially symmetric shape, and Fig. 18 (f) is a similar case where a human can stand in any direction of the table and put her arms on the table. Figure 18 (e) shows that the interaction space of a sofa occupies the space evenly in front and above the sofa.

Extracting interaction behaviors from human-ground interaction. We performed additional experiments on extracting interaction behaviors from human-ground interactions. Figure 19 (left) shows the interaction behaviors segmented from a motion including sitting and lying on the ground. As there is no object, canonical poses are manually selected from sitting and lying poses. Four interaction behaviors are successfully segmented from the motion. Figure 19 (right) shows an experimentation result on locomotion including stepping on a low object. This is a more challenging scenario than previous examples because interaction behavior, i.e., the sequence from stepping on to stepping down from an object, is not distinctive enough from a normal walking in terms of the velocities of the contacting markers. As a result, a sequence (c to d) is classified as a transition despite that human would regard it as a part of an interaction sequence.

7. Limitations and Future Work

This section discusses several limitations of our approach and possible future directions for improvement.

Our method assumes that the 3D models in the database are appropriately scaled so that they match the sizes of real objects. This assumption allows us to vary the 3D model's scale in the range of 0.9 and 1.1. However, since the 3D models found in the Internet often have widely varying scales, a manual processing to adjust the scales of 3D models was necessary. An automatic method to find a proper range of scales of given 3D models depending on their properties will remove the manual process and improve the utility of our method.

To increase the matching speed, we use the filter map to cull out objects before performing fine scale test. However, the time complexity of our matching process is still linear to the number of objects in the database, which may significantly slow down the matching process for a very large database. In this case, an efficient organization of the objects in the database, e.g., the k-d tree with respect to the filter map features, will increase the filtering speed. In addition, for a very large database, finding every matchable objects would not be necessary,

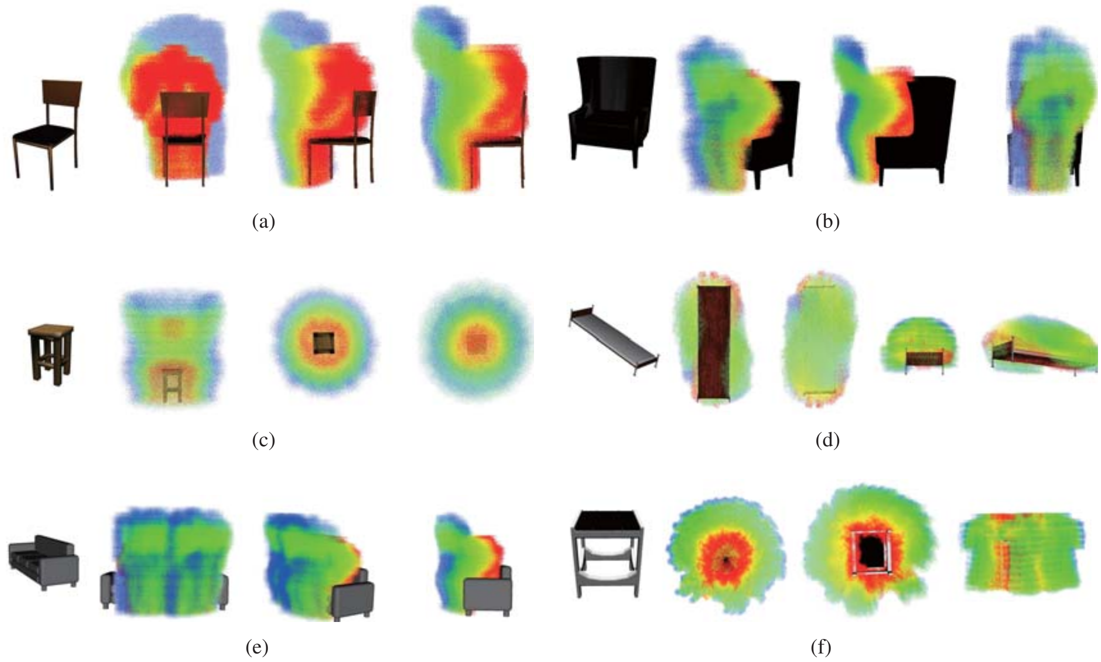


Figure 18: The estimated interaction space for various chairs and other types of furniture.

and a sampling-based approach that finds only a user-specified number of objects will be enough.

Since the process of extracting contact blocks detects only static contact made by markers approached in approximately normal directions, our method cannot deal with other types of contact such as sliding. Contact with fingers such as holding and grasping have not been considered yet. Future work that overcomes this limitation will enable reconstructing environments including challenging objects such as ladders, seesaws, slides, and jungle gyms.

In the scene analysis stage, we extracted the interaction behaviors based on the conformity value. The method gives satisfactory results when the interaction motion is distinguishable from approaching motion, but loses accuracy when the two motions are similar as shown in Fig. 19 (right). An interesting future direction to develop a better method for extracting interaction behaviors would be to combine physical characteristics of motions, such as momentum and balance, with data-driven approaches.

Lastly, in this work we have not yet provided the actual application of our method to estimate interaction space of objects. While existing motion generation methods such as the space-time optimization can utilize our interaction space, our future goal is to develop efficient methods to create realistic human-object interaction motions based on the interaction space.

Acknowledgement

This work was supported by the Global Frontier R&D Program (2015M3A6A3073743) and the Basic Science Research Program (2017R1A2B2006160) funded by NRF, MSIP, Korea.

References

[1] M. Fisher, M. Savva, Y. Li, P. Hanrahan, M. Nießner, Activity-centric scene synthesis for functional 3D scene modeling, ACM Transactions on Graphics (TOG) 34 (6) (2015) 179.

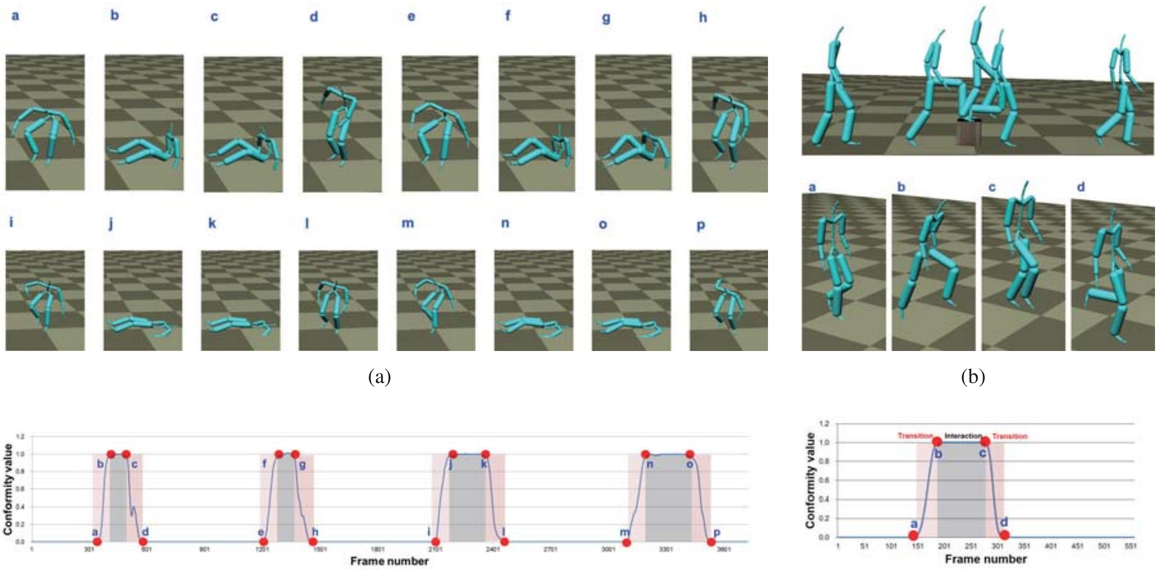
[2] Y. Jiang, H. S. Koppula, A. Saxena, Modeling 3D environments through hidden human context, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (10) (2016) 2040–2053.

[3] K. Chen, Y. Lai, Y.-X. Wu, R. R. Martin, S.-M. Hu, Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information, ACM Transactions on Graphics (TOG) 33 (6) (2014) 208.

[4] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, M. Nießner, Scenegrok: Inferring action maps in 3D environments, ACM Transactions on Graphics (TOG) 33 (6) (2014) 212.

[5] R. Hu, C. Zhu, O. van Kaick, L. Liu, A. Shamir, H. Zhang, Interaction context (icon): towards a geometric functionality descriptor, ACM Transactions on Graphics (TOG) 34 (4) (2015) 83.

[6] V. G. Kim, S. Chaudhuri, L. Guibas, T. Funkhouser, Shape2pose: Human-centric shape analysis, ACM Transactions on Graphics (TOG) 33 (4) (2014) 120.



(a) (b)

(c) (d)

Figure 19: Interaction behaviors segmented from a repetitive sitting and laying motions (left) and from a motion walking over an object (right).

[7] H. Grabner, J. Gall, L. Van Gool, What makes a chair a chair?, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 1529–1536.

[8] K. Xu, H. Huang, Y. Shi, H. Li, P. Long, J. Caichen, W. Sun, B. Chen, Autoscanning for coupled scene reconstruction and proactive object analysis, ACM Transactions on Graphics (TOG) 34 (6) (2015) 177.

[9] M. G. Helander, Handbook of human-computer interaction, Elsevier, 2014.

[10] C. Baldassano, D. Beck, L. Fei-Fei, Human-object interactions are more than the sum of their parts., Cerebral Cortex (New York, NY: 1991).

[11] K. B. Chen, R. A. Kimmel, A. Bartholomew, K. Ponto, M. L. Gleicher, R. G. Radwin, Manually locating physical and virtual reality objects, Human Factors 56 (6) (2014) 1163–1176.

[12] C. Kang, S.-H. Lee, Environment-adaptive contact poses for virtual characters, Computer Graphics Forum (CGF) 7 (33) (2014) 1–10.

[13] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, B. Guo, An interactive approach to semantic modeling of indoor scenes with an rgbd camera, ACM Transactions on Graphics (TOG) 31 (6) (2012) 136.

[14] Y. M. Kim, N. J. Mitra, D.-M. Yan, L. Guibas, Acquiring 3D indoor environments with variability and repetition, ACM Transactions on Graphics (TOG) 31 (6) (2012) 138.

[15] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, S. J. Osher, Make it home: automatic optimization of furniture arrangement, ACM Transactions on Graphics (TOG) 30 (4) (2011) 86.

[16] Y. Jiang, M. Lim, A. Saxena, Learning object arrangements in 3D scenes using human context, Proceedings of the 29th International Conference on Machine Learning (ICML) (2012) 1543–1550.

[17] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, A. A. Efros, Scene semantics from long-term observation of people, European Conference on Computer Vision (ECCV) (2012) 284–298.

[18] P. Wei, Y. Zhao, N. Zheng, S.-C. Zhu, Modeling 4D human-object interactions for event and object recognition, IEEE International Conference on Computer Vision (ICCV) (2013) 3272–3279.

[19] H. Laga, M. Mortara, M. Spagnuolo, Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes, ACM Transactions on Graphics (TOG) 32 (5) (2013) 150.

[20] Z. Harchaoui, F. Bach, Image classification with segmentation graph kernels, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007) 1–8.

[21] X. Zhao, H. Wang, T. Komura, Indexing 3D scenes using the interaction bisector surface, ACM Transactions on Graphics (TOG) 33 (3) (2014) 22.

[22] A. Gupta, S. Satkin, A. A. Efros, M. Hebert, From 3D scene geometry to human workspace, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 1961–1968.

[23] R. Ma, H. Li, C. Zou, Z. Liao, X. Tong, H. Zhang, Action-driven 3D indoor scene evolution, ACM Transactions on Graphics (TOG) 35 (6) (2016) 173.

[24] S. Pirk, V. Krs, K. Hu, S. D. Rajasekaran, H. Kang, Y. Yoshiyasu, B. Benes, L. J. Guibas, Understanding and exploiting object interaction landscapes, ACM Transactions on Graphics (TOG) 36 (3) (2017) 31.

[25] R. E. Kalman, A new approach to linear filtering and prediction problems, Journal of Basic Engineering 82 (1) (1960) 35–45.

[26] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation 15 (1) (2007) 1–28.

[27] C. E. Rasmussen, Gaussian processes for machine learning, MIT Press, 2006.

[28] J. A. Nelder, R. Mead, A simplex method for function minimization, The Computer Journal 7 (4) (1965) 308–313.