iMapper: Interaction-guided Scene Mapping from Monocular Videos

ARON MONSZPART, University College London
PAUL GUERRERO, University College London
DUYGU CEYLAN, Adobe Research
ERSIN YUMER, Uber ATG
NILOY J. MITRA, University College London and Adobe Research

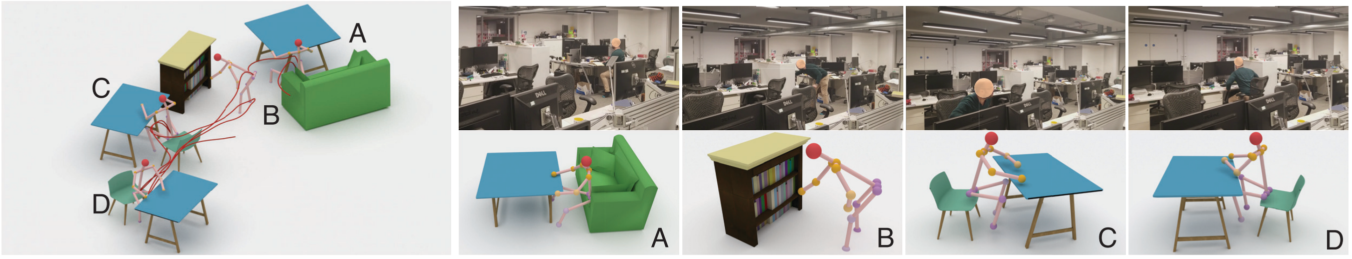


Fig. 1. **Interaction-guided scene mapping.** We present iMapper that discovers potential human-object *interactions* in an input monocular video and utilizes them to infer the object layout of the recorded scene containing medium to heavy occlusion. We show the final generated 3D scene as well as recovered interactions (Scene13). Note that although *only* the 2D human joint detection (left) is available to our algorithm, here we additionally show reference video frames (corresponding to A, B, C) to help judge the original scene layout. Please refer to supplementary video.

Next generation smart and augmented reality systems demand a computational understanding of monocular footage that captures humans in physical spaces to reveal plausible object arrangements and human-object interactions. Despite recent advances, both in scene layout and human motion analysis, the above setting remains challenging to analyze due to regular occlusions that occur between objects and human motions. We observe that the *interaction* between object arrangements and human actions is often strongly correlated, and hence can be used to help recover from these occlusions. We present iMapper, a data-driven method to identify such human-object interactions and utilize them to infer layouts of occluded objects. Starting from a monocular video with detected 2D human joint positions that are potentially noisy and occluded, we first introduce the notion of *interaction-saliency* as space-time snapshots where informative human-object interactions happen. Then, we propose a global optimization to retrieve and fit interactions from a database to the detected salient interactions in order to best explain the input video. We extensively evaluate the approach, both quantitatively against manually annotated ground truth and through a user study, and demonstrate that iMapper produces plausible scene layouts for scenes with medium to heavy occlusion. Code and data are available on the project page.

CCS Concepts: • **Computing methodologies → Scene understanding**; **Shape inference**; **Shape analysis**; *Motion capture*; *Perception.*

Additional Key Words and Phrases: shape analysis, interaction, scene layout, monocular video, occlusion

ACM Reference Format:
Aron Monszpart, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J. Mitra. 2019. iMapper: Interaction-guided Scene Mapping from Monocular Videos. *ACM Trans. Graph.* 38, 4, Article 92 (July 2019), 15 pages. https://doi.org/10.1145/3306346.3322961

1 INTRODUCTION

Computational understanding of monocular videos that capture human-object interactions in physical spaces is critical for many emerging fields such as virtual and augmented reality, smart home systems, assisted living, and robotics. Such applications require access to object arrangements embedded in the physical spaces along with the common human-object interactions performed in such spaces. For example, our future personal robot assistants should know our working habits along with the supporting object arrangements in our living rooms or workspaces. Hence, a joint understanding of scenes and human actions from the input feed is necessary.

While both scene understanding and human performance analysis are highly popular research areas, traditionally, researchers have tackled them as two separate problems. On the one hand, *scene estimation methods* such as Kinect Fusion [Newcombe et al. 2011] and Bundle Fusion [Dai et al. 2017b] can produce high-quality static indoor reconstructions, while the likes of DynamicFusion [Newcombe et al. 2015] can capture non-rigidly deforming scenes by fusing depth information across space and time. These methods, however, require the sensor to be manually moved to see around occlusions making

Authors' addresses: Aron Monszpart, Department of Computer Science, University College London, aron.monszpart.12@ucl.ac.uk; Paul Guerrero, Department of Computer Science, University College London, paul.guerrero@ucl.ac.uk; Duygu Ceylan, Adobe Research, ceylan@adobe.com; Ersin Yumer, Uber ATG, meyumer@gmail.com; Niloy J. Mitra, Department of Computer Science, University College London and Adobe Research, n.mitra@ucl.ac.uk.

0730-0301/2019/7-ART92 $15.00
https://doi.org/10.1145/3306346.3322961

the capture process cumbersome. On the other hand, *human performance capture methods* either use multiple sensors [von Marcard et al. 2017] or monocular video [Mehta et al. 2017b; Rogez et al. 2019] but assume performances to be free from object-induced occlusions. However, in monocular footage that capture human actions in physical spaces, objects and human motions regularly *occlude* each other during interactions. Filling in missing information due to such occlusions is ambiguous due to the diversity of possible scene configurations and is handled poorly by current methods.

While indoor scene configurations can be extremely rich and diverse, we observe that many of them are linked by a common thread – *they are regularly inhabited by humans*. Moreover, in similar scene configurations, humans tend to perform similar actions (*cf.,* [Krasner 2013]). Examples of such actions include sitting on sofas, picking up books from shelves, or walking around obstacles. Instead of tackling scene estimation and human performance capture separately, we propose to exploit the captured human performance to better infer plausible scene layouts.

A fundamental challenge in reaching the above goal using monocular video in natural surroundings is *occlusion* arising out of human-object interactions. A successful solution needs to tackle two problems: first, hallucinating information about partially or fully hidden objects; and second, recovering from noisy human performance estimates from monocular videos, especially in regions of medium to high occlusion. It is believed that, as humans, we focus on the interactions of the actors with the objects in a scene (referred to as 'anticipation' in [Neisser 1976]), instead of separately identifying objects and human performances. Detecting such interactions helps compensate for missing information in *both* objects and performances. For example, in the video for the scene shown in Figure 1, we can 'see' the person walking behind the desk and sitting down; from that, we can imagine both the person's sitting pose over time and the location of the unseen chair/sofa. Similarly, for the person picking up an object from the shelf.

We propose iMapper, a data-driven method, that accomplishes a similar feat by *utilizing human motions to infer object placements.* As a data-prior, we leverage a database of interactions between humans and local objects over time, which we call *scenelets* (extracted from the PiGraph dataset [Savva et al. 2016]). Our key observation is that state-of-the-art methods (see Section 8) are now reliable for detecting visible parts of the human performance, and hence the local objects being interacted with, even if partially or fully occluded, in the scenelets matched to such human performance detections can be used to provide good candidate object layouts associated with such detected human-object interactions.

Starting from a monocular video, we use a state-of-the-art human pose detector to identify initial joint estimates over time and analyze them to identify 'informative' space time snapshots representing potentially informative interactions. We then utilize the snapshots to retrieve matching local scenelets and solve a global optimization to extract a consistent subset of these scenelets, arrange them, and inherit their associated objects and human actions to produce both static objects and a human performance that are mutually consistent, and agree with the input video.

We extensively evaluated iMapper on a range of scenes of varying complexity. Our quantitative evaluation against manually annotated ground truth data, and qualitative evaluation through a user study demonstrate that iMapper produces realistic and plausible object layout and human-object interaction estimation even for scenes with significant amounts of occlusion.

In summary, our main contributions are:

(i) proposing the first method that discovers and utilizes human-object interactions to produce a 3D scene layout from a monocular video showing human interactions in natural settings;

(ii) extracting informative space-time human action snapshots and matching them to a scenelet database; and

(iii) combining the matched candidate scenelets into a consistent 3D scene layout and human performance by a novel global optimization, and evaluating the proposed iMapper method on a range of challenging examples.

2 RELATED WORK

We now discuss selected papers across four main related topics to better position our approach.

Scene analysis and synthesis. With the advances in acquisition technologies, several large-scale indoor reconstruction datasets have been created [Chang et al. 2017; Dai et al. 2017a]. Starting from similar 3D scene collections, several previous works focus on analyzing inter-object relationships [Fisher et al. 2011; Hu et al. 2016, 2015; Xu et al. 2014; Zhao et al. 2014] and hierarchical grammars [Liu et al. 2014]. Such discovered inter-object relationships then can be used to synthesize new scene variations, *e.g.,* by replacing objects or scene parts with those in different scenes such that existing relationships are maintained [Huang et al. 2016; Zhao et al. 2016], using a Markov Chain Monte Carlo based approach [Yeh et al. 2012], or using a probabilistic graphical model [Fisher et al. 2012]. More recently, deep learning methods have been proposed to progressively synthesize plausible scenes [Wang et al. 2018]. These methods, however, require full knowledge of the 3D scene layouts instead of attempting to recover them from image or video footage.

Another line of work recovers layouts from single images [Hueting et al. 2018; Izadinia et al. 2017; Poirson et al. 2016; Satkin and Hebert 2013; Tulsiani et al. 2018] or RGBD scans [Nan et al. 2012; Shao et al. 2012] by matching individual 3D objects. Relationships between the matched objects have been used to further regularize the recovered layout [Chen et al. 2014; Schwing et al. 2013], or a collection of primitives to reconstruct accurate room geometry from images [Del Pero et al. 2013]. While our goal is to also recover an approximate 3D scene layout of a partially observed scene, we rely on detected human interactions to reason about occluded objects.

Human pose estimation. With the recent success of deep learning, we have seen advances both in 2D [Cao et al. 2017; Insafutdinov et al. 2016; Newell et al. 2016; Toshev and Szegedy 2014; Wei et al. 2016] and 3D pose estimation [Huang et al. 2014; Rogez et al. 2019; Tekin et al. 2016; Tomè et al. 2017; Zhou et al. 2016]. In particular, the recent VNect system [Mehta et al. 2017b] demonstrates impressive global pose estimation from monocular video. Many of these approaches, however, do not specifically tackle the occlusion problem and fail under moderate to heavy occlusion. Only a few pieces of existing work focus on predicting human pose, either 2D [Fu et al. 2015] or 3D [Huang and Yang 2009; Wei et al. 2012], in the presence of slight occlusions from static input images. Wei et al. [2010] and Shao et al. [2014] leverage user assistance and physical constraints to handle moderate occlusions. These methods do not reason explicitly about occlusions arising due to human-object interactions, nor use interaction priors to recover occluded interactions. In contrast, we utilize the initial human pose estimates from state-of-the-art human pose detectors to jointly reason about scene layout and human pose to synthesize both plausible scenes and human motion even in scenes with moderate to heavy occlusions.

Human-centric shape analysis. Earlier work that uses observations of how humans interact with objects focuses on tasks such as object and event recognition [Delaitre et al. 2012; Gupta et al. 2009; Wei et al. 2013] and action detection [Yao et al. 2011]. Kim et al. [2014] propose a shape analysis tool based on a human-object affordance model that can be used for many applications including correspondence estimation, shape retrieval, and view selection. Subsequently, Fu et al. [2017a] utilize a similar model to generate new objects by combining functional parts of existing objects, while Pirk et al. [2017a; 2017b] introduce the concept of *interaction landscapes* as a descriptor of an object based on the type of interactions it can be involved in. More recently, Gkioxari et al. [2018] predict human-verb-object instances from a single image to characterize human-object interactions. These methods focus on individual objects and their (potential) interactions. Our goal is to use partially observed human motion to recover a full scene and a consistent human performance.

Human-centric scene synthesis. Recent efforts in scene synthesis incorporate human actions into scene analysis. Datasets of typical human actions have been used to infer where specific actions can take place in a scene [Savva et al. 2014], to regularize scene synthesis [Fu et al. 2017b; Ma et al. 2016], or to regularize reconstruction of a scene layout from incomplete 3D scans [Fisher et al. 2015; Jiang et al. 2016]. Although these methods use models of typical human actions as priors, they do not explicitly use action or motion cues as input. Thus, unlike our method, they fail to recover occluded objects.

We were inspired by methods that use human action or motion cues to reason about the scene geometry. Fouhey et al. [2012] combine human pose estimates with appearance and other geometric cues to estimate the room cuboid and free (walkable) space inside the room from a single image or a time-lapse video. Frank et al. [2015] recover an object layout from a manually defined set of human actions. Object types and shapes are inferred from specific motions that the human performs, like tracing the edges of a table with the hands. Similar to ours, objects are recovered by observing human motion during interactions, however, the type of motions that the actor needs to perform preclude working with natural videos.

The recent work of Savva et al. [2016] analyzes interaction snapshots, *i.e.,* action and pose labeled RGBD sequences to learn *prototypical interaction graphs (PiGraphs)* to link attributes of the human pose to the surrounding objects. They show how PiGraphs can be utilized to generate scenes that correspond to *static* interaction snapshots (*e.g.,* lie on bed). Kang et al. [2017] focus on a similar goal of scene synthesis that explores motion cues. However, their input is an occlusion-free 3D human motion sequence in contrast to our target scenes with moderate to severe occlusions.

3 SCENELETS: REPRESENTATION AND DESCRIPTORS

iMapper heavily relies on identifying and utilizing human-object interactions for scene layout and mapping. We start by describing how we represent the space of possible interactions as a database of ‘scenelets’, introduce suitable descriptors to query into this database, and the important notion of *interaction-saliency* to identify informative scenelets with strongly correlated human-object interactions before presenting the iMapper algorithm in the next section.

As our human-object interaction database, we use the PiGraphs dataset [Savva et al. 2016] that contains a set of scenes captured by a commodity depth sensor, each containing a human performance and a set of associated labeled objects (*e.g.,* tables, sofas, chairs, bookshelves, *etc.*). From this dataset, we extract short sequences representing interactions between the human actor and scene objects. We call such short sequences of frames *scenelets*.

3.1 Scenelet Representation

Each scenelet $\mathcal{S}_l = \{\{s^l_{k,t}\}, O^l\}$ consists of a short motion clip with known 3D joint positions and a set of static objects. We denote with $s^l_{k,t}$ the location of skeleton joint k in frame t of scenelet l. Objects $O^l = \{o^l_1, \dots o^l_n\}$ of scenelet l are defined by a placement p, a rough approximation of their geometry κ, and a class label b; *i.e.,* each object is encoded as a triplet $o = (p, \kappa, b)$. We assume objects can only rotate around the up direction, leaving four degrees of freedom for the placement of an object: $p = (x, y, z, \theta)$, where x, y, and z are the location, and θ the orientation of the object. Similar to the original dataset we approximate the geometry of objects by unions of cuboids, and the label b^l_i describes the object type (*e.g.,* chair, table, bookshelf) from a predefined set of categories. Both the motion clip and the objects are stored in the local coordinate frame of the scenelet defined by the pelvis location and the forward-facing direction of the skeleton in the center frame of the motion clip. Figure 3 shows some example scenelets.

3.1.1 Scenelet parameterization. When constructing scenelets, we make a design choice regarding the time-duration of the motion clip used for each scenelet based on two factors. First, the speed at which an interaction is performed should not affect the contents of a scenelet. For example, if a scenelet captures a fast ‘sitting-down’ performance then a slower version of the same interaction should also be captured by a single scenelet. This property is necessary to ensure that interactions captured by scenelets are comparable. Second, we assume that interactions are local in space, *i.e.,* the actor does not traverse a large distance during the interaction.

We select the duration of scenelets such that the skeleton (*i.e.,* the pelvis joint) traverses a constant arc length as this satisfies both time invariance and locality requirements. For increased robustness, we smooth the pelvis trajectory when computing the arc length with 10 iterations of a moving average. We set the spatial extent of the smoothing kernel to an arc length radius of 1 cm (the input skeletons were recorded at 30 Hz).

3.1.2 Scenelet construction. The PiGraphs dataset describes human performance with a set of 16 joint locations per frame. We start by sampling the center of each scenelet's motion clip at regularly spaced intervals on the arc length of the pelvis joint's trajectory. The start and end of the motion clip in each scenelet is then defined as this center point ± half the scenelet arc length. The objects of the scenelet are chosen as the subset of objects within roughly an arm's reach (within 1 m radius when projected to the ground plane) of the actor at any point in the motion clip, *i.e.,* the objects the actor can potentially interact with, manually corrected where needed.

3.2 Scenelet Descriptors

In order to compare two scenelets $\mathcal{S}^1, \mathcal{S}^2$ and compute their distance $d(\mathcal{S}^1, \mathcal{S}^2)$, we define two descriptors for each scenelet: a *motion descriptor* Ψ and an *object descriptor* Φ. As explained later in this section, we utilize these descriptors to identify informative scenelets via their *interaction-saliency* scores.

3.2.1 Motion descriptor. The *motion descriptor* Ψ captures the human actor motion over the scenelet duration and compactly describes the motion clip of a scenelet with a fixed-length vector as the concatenation of a fixed number of static pose descriptors $\Psi := (\psi_1, \ldots, \psi_t)$, we use $t = 15$ samples. Static pose descriptors are based on 14 robust joint-line distances as proposed by Zhang et al. [2016]. The distance between two motion descriptors is defined using a weighted L_2 distance between the corresponding static pose descriptors, assigning more weight to center frames. The descriptors ψ_i should evenly cover the motion and be invariant to the speed of the motion. Thus, we evenly distribute these samples along the trajectory of the motion clip in a 17D space of the combined pose descriptor and global skeleton location (taken to be the 3D location of the pelvis).

3.2.2 Object descriptor. The *object descriptor* Φ encodes statistics of relative object layouts with respect to the actor and is defined as a set of histograms with one histogram per object category. The histograms capture the 2D placement of objects, projected to the ground plane. Our histograms are 5 × 5 square grids (in a coordinate frame with the center pose facing the forward direction) and each bin Φ_j describes to what extent any object of the same category in the scenelet is located in this bin. We define the value in a bin as the maximum coverage of the bin by any object. To handle both objects smaller and larger than the bin, we normalize by the smaller of either the bin area or the object area as,

$$\Phi_j = \max_i \left\{ A\left(\Lambda(o_i) \cap \phi_j\right) / \min\left(A\left(\Lambda(o_i)\right),\ A(\phi_j)\right) \right\},$$

where $\Lambda(o_i)$ is the projection of object o_i to the ground plane, ϕ_i is the part of the ground plane covered by bin i, and $A(x)$ is the area of x. Figure 2 shows an example of an object descriptor.

3.3 Interaction-saliency for Scenelets

In order to identify informative human-object interactions, we associate each scenelet with an *interaction-saliency* score. Scenelets with high *interaction-saliency* are likely to contain informative interactions with objects and they have objects within interaction range that are typical or characteristic of the scenelet's motion.

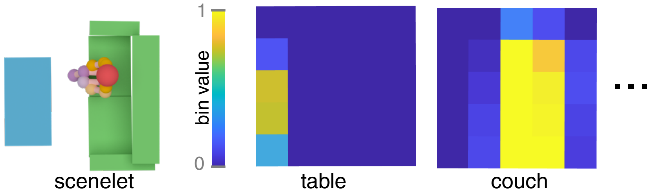


Fig. 2. **Object descriptor** Φ**.** Object descriptors compactly represent the object arrangement of a scenelet. One 5 × 5 histogram per category stores the layout of objects of this category relative to the scenelet center.

Thus, we define the uniqueness or saliency of a bin in an object descriptor to describe how typical an activation of this bin is for similar motion clips. For example, for a sitting-down motion, a couch or a chair at the center bin of the histogram will have high saliency. The main intuition behind this saliency score is to distinguish between objects that are consistently related to a given interaction, and objects that are near the motion but unrelated to the interaction. In other words, we associate repeated and consistent presence of objects as a reliable 'witness' to the particular interaction represented by the human pose. The bin-saliency of an object descriptor bin is computed as a weighted average of that bin's activation over similar motion clips, where similarity weight is defined with a Gaussian kernel in the space of motion descriptors as

$$h_j^l = \frac{\sum_{k=1}^{m} \Phi_j^k \ \mathcal{G}(d(\Psi_k, \Psi_l)|0, \sigma)\rho_k^{-1}}{\sum_{k=1}^{m} \mathcal{G}(d(\Psi_k, \Psi_l)|0, \sigma)\rho_k^{-1}},$$

where h_j^l is the saliency of bin j in scenelet l, Φ_j^l is the bin value of scenelet l, $\mathcal{G}$ is a Gaussian kernel taken over the distance d between the motion clip descriptors defined earlier (we set $\sigma = 13$), and ρ_l is the *density* of scenelets at the origin of scenelet l. We measure the density of scenelets as the spatial density of the origins of all scenelets that were obtained from the same scene in order to remove any bias introduced due to multiple scenelets taken from nearby parts of the scene. The summation is computed over all the scenelets in the database and finally, we define the *interaction-saliency* of a scenelet $\mathcal{S}^l$ as the maximum of the bin-saliency, *i.e.,*

$$H^l = \max_j(h_j^l).$$

4 ALGORITHM OVERVIEW

The input to iMapper[1] is a monocular video showing a person interacting with objects. Our goal is to synthesize a plausible scene layout along with consistent human performance that explains the input video.

When watching performance of a human actor in a scene, there are several cues that help recover plausible explanations for scene objects and performance sequences, even under partial occlusion. Specifically, both presence and absence of human-object interactions carry valuable hints: on the one hand, detecting a person interacting with objects provides information about the potential types and

[1] Project code and benchmark dataset at http://geometry.cs.ucl.ac.uk/projects/2019/imapper

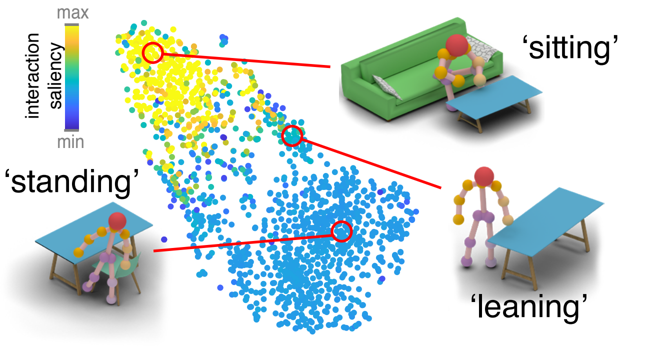


Fig. 3. **Scenelet's interaction-saliency.** We show a t-SNE embedding of *interaction-saliency* scores of the scenelets in our database. Warmer colors denote scenelets with higher *interaction-saliency*. The close-ups show scenelets with a standing sequence, a sitting sequence, and a leaning sequence. Standing sequences have low *interaction-saliency* because they are not specific to the neighboring objects.

locations of the surrounding objects. On the other hand, observing a person walking carries information about empty regions of the scene. iMapper explores such cues and proceeds in three main stages, as described next.

(i) *Identifying informative interactions:* In Section 5, we identify potentially informative space-time segments of the input skeletal motion as useful human-object interactions. Starting from a monocular video, we first use state-of-the-art human pose detectors [Rogez et al. 2019; Tomè et al. 2017] to generate an initial human skeleton track over time. Using the notion of *interaction-saliency* defined in Section 3, we assign scores to parts of the skeletal motion by matching against database scenelets and assessing how informative the parts are based on the *interaction-saliency* scores of these matched scenelets.

(ii) *Retrieving matched interactions:* In Section 6, we formulate an energy function to help retrieve matching scenelets corresponding to the skeletal motion segments extracted in the previous step. During this process we encourage consistency measured based on how well the fitted models explain the presence or absence of skeleton joint detections in each video frame. The matched scenelets, in turn, provide local objects as candidate completions for the occluded parts of the scene.

(iii) *Scene mapping via global optimization:* Finally, in Section 7, we solve a global optimization by minimizing an energy that additionally considers plausibility criteria (*e.g.,* path smoothness and intersection avoidance) to obtain a static scene layout and a consistent human performance that matches the input monocular video. Specifically, we formulate a selection problem to extract a subset of scenelets among the matched candidate ones to constitute our synthesized scene, and optimize the placement of these chosen scenelets and skeletons to provide a plausible explanation for the input video.

5 IDENTIFYING INFORMATIVE INTERACTIONS

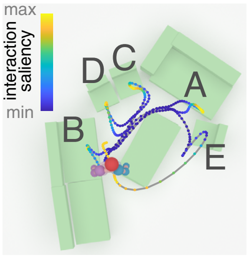
Starting from a monocular video, as preprocessing we use existing human pose trackers to form initial skeletal joint tracks over time, and then identify informative interaction segments using a data-driven notion of *interaction-saliency*.

5.1 Generating an Initial Skeletal Estimate

We apply state-of-the-art static pose detectors to obtain initial skeletal estimates from the input monocular video. In each frame, we detect the image-space skeleton of the actor consisting of n_j joint locations along with local 3D pose estimates (*i.e.,* pelvis is always at the origin). In our experiments, we tested with (i) 2D keypoints detected by CPM [Wei et al. 2016] and grouped based on the heuristic of Tomè et al. [2017] or (ii) LCR-Net++ [Rogez et al. 2019]. Given a video, for each joint k detected in frame t, $u_k^t \in \mathbb{R}^2$ denotes its image-space location and $c_k^t \in [0, 1]$ its confidence. While CPM directly provides confidence values, for LCR-Net++ we compute them using pose proposal variance (see Appendix A). We note that these initial pose estimates are often highly noisy in the presence of occlusions, *e.g.,* around human-object interactions. We next try to match the initial 2D skeletal pose estimates (using a sliding window approach) to database interactions by fitting them close to the initial 3D pelvis path to identify informative space-time segments.

5.2 Estimating Interaction-likelihood Score

Our goal is to assign scores to the video frames indicating the likelihood to contain *informative* interactions between the actor and the objects (see Figure 3), based on which we can fetch potential scene objects. We start by fitting the scenelets in our database to each video frame and use the *interaction-saliency* (see Section 3.3) of the matched scenelets weighted by their matching quality to determine the probability of an interaction.



Specifically, for any frame at time t, say $\{\mathcal{S}^i\}$ be the top $\mathbb{K}_{\text{sal}}$ matched scenelets ($\mathbb{K}_{\text{sal}}$= 20 in our tests) with corresponding *interaction-saliency* given by H^i and matching quality by w_i. This quality w_i measures how well the 3D skeletal joints in frames inside a window of $[t - \tilde{t}, t + \tilde{t}]$ frames (we use ±10 frames) match the human motion in the scenelet $\mathcal{S}^i$ according to the fitting energy defined in Equation 5 . We then define the *interaction-likelihood* score for frame t simply as the weighted average $\sum_i w_i H^i / \sum_i w_i$. The inset figure shows an example scene where the final recovered actor trajectory is color-coded based on the interaction likelihood of the corresponding frames. The regions denoted by letters have higher score indicating sitting interactions.

Our goal is to fit scenelets only to parts of the video that contain interactions, *i.e.,* are informative of the objects in the scene. Thus, we perform non-maximum suppression of the interaction-likelihood over the video frames. At this stage, we have assigned scores to the frames quantifying how informative they are for assisting in subsequent interaction-based object placement.

6 RETRIEVING MATCHED INTERACTIONS

We quantify the consistency of alignment of a given 2D skeletal track $\{u_k^t\}$ with associated detection confidence $\{c_k^t\}$ against a scenelet $\mathcal{S}$ over its placement P (see Section 3.1) using an energy function that penalizes inconsistency as,

$$L(\{u_k^t, c_k^t\}, \mathcal{S}, P) = w_r L_r + w_o L_o, \tag{1}$$

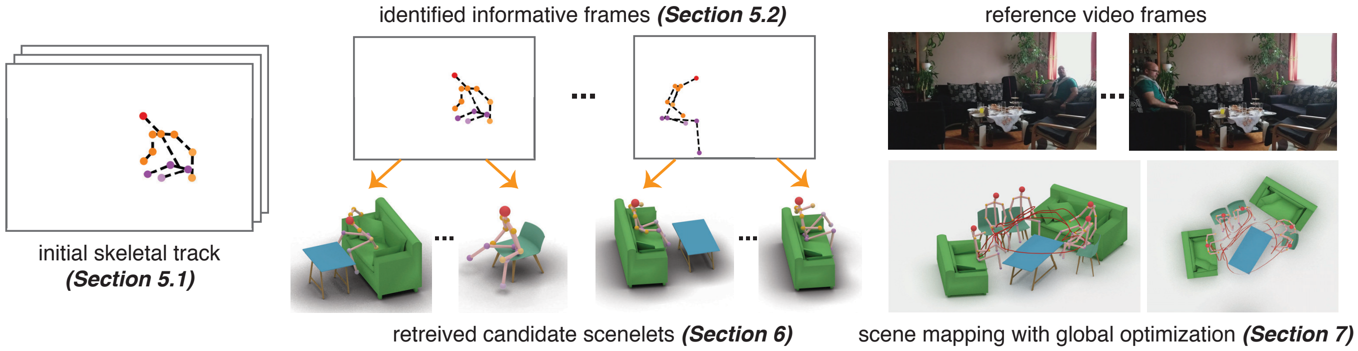

identified informative frames (Section 5.2)
reference video frames
initial skeletal track (Section 5.1)
retreived candidate scenelets (Section 6)
scene mapping with global optimization (Section 7)

Fig. 4. **Overview.** Given an input monocular video, we first extract an initial skeletal track with confidence scores (Section 5.1). We identify informative frames in the input sequence by assigning each frame an *interaction likelihood score* based on their matching scenelets (Section 5.2). For each informative frame, we identify the top matching candidate scenelets (Section 6). Finally, we perform a global optimization that selects and places such candidate scenelets to form a plausible 3D scene and human performance (Section 7). We show the final synthesized 3D scene, the recovered trajectory of the person (in red), and some of the detected interacting human poses. The reference video frames show that the synthesized 3D scene provides a plausible explanation.

where L_r measures the *reprojection* error between the 2D joints and the projection of 3D joint locations; and L_o penalizes the presence of *occlusions* of skeleton joints in the video that are not explained by occlusions in the scenelet objects. We next describe the terms.

Reprojection term (L_r). The reprojection term is a standard back projection term that penalizes the distance from the image-space 2D joint locations detected in the video to the projected scenelet joints q_k^t as,

$$L_r = \sum_t \sum_k c_k^t \left\|\Pi q_k^t - u_k^t\right\|_2^2, \tag{2}$$

where Π is the camera projection matrix (we assume the intrinsic camera parameters to be known) and u_k^t are the detected input 2D joint locations with respective confidence c_k^t.

Occlusion term (L_o). The occlusion term is a novel term that encourages consistency between joint occlusions observed in the video and occlusions of joints induced by scenelet objects as seen from the camera. In other words, we require the synthesized objects to explain observed occlusions. This cost is asymmetric, *i.e.,* missing joint detections occur either due to false negatives in joint detections, or due to occluding objects. The reverse is, however, *not* true: the joint detector may, in some cases, also predict the position of occluded joints with high confidence. We define an asymmetric occlusion error as,

$$L_o = \sum_t \sum_k F(v(q_k^t, O, \Pi), c_k^t), \tag{3}$$

where $v(q_k^t, O, \Pi)$ denotes the visibility of joint q_k^t given the scene objects O and the current camera information Π. In order to have non-zero gradients that are necessary for gradient-based solver, we define v as the signed distance of joint q_k^t to the *occlusion volume* induced by O that is the volume that remains invisible from the camera. We then define the asymmetric occlusion error F for a joint and a set of objects is as,

$$F(v, c) = \begin{cases} (c - 0.5)^2 v^2 & \text{if } c - 0.5 < 0 \text{ and } v > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $c \in [0, 1]$. Note that this function is non-zero *only* when low-confidence joint detections are explained by visible joints. Finally, we can quantify the alignment error between given 2D skeletal tracks against a scenelet $\mathcal{S}$ as,

$$L^\star(\{u_k^t, c_k^t\}, \mathcal{S}) = \min_P L(\{u_k^t, c_k^t\}, \mathcal{S}, P). \tag{5}$$

We solve Equation 5 by a gradient-based optimization and retrieve the top $\mathbb{K}_{\text{glob}} = 5$ matching scenelets. For efficiency, we only consider the best $\mathbb{K}_{\text{loc}} = 200$ scenelets from Section 5.2 in this section. Next, we describe how to globally select among these retrieved scenelets and obtain a final scene layout.

7 SCENE MAPPING VIA GLOBAL OPTIMIZATION

Our goal is to synthesize a scene consisting of 3D joint locations $q_k^t \in \mathbb{R}^3$ for each video frame, describing the human performance, and a set of objects $O = \{o_1, \dots o_{n_o}\}$. The 3D joint locations at each frame are obtained from either one of the matched candidate scenelets, or fitting a 3D skeleton to the 2D joint detections in the video; while objects are only taken from the selected candidate scenelets. Which of the two models we fit to a given part of the video depends on two factors: the estimated amount of joint occlusion observed in the video (*i.e.,* the confidence of the joint detection signal) and the estimated probability of object interactions.

First, in segments of the video with high interaction-likelihood, our task is to pick a scenelet among the top $\mathbb{K}_{\text{glob}}$ matches obtained previously, *i.e.,* we have to solve a selection problem. Second, in segments of the video with low interaction-likelihood, matched scenelets are less useful. Instead, we match the initial local 3D human pose estimates to the image-space joint detections. In the following, we describe these two scenarios, and then define a global energy that when minimized produces the final solution.

7.1 Video Segments with High Interaction-likelihood

Here, we pick among the matched scenelets to both populate the scene with objects involved in interactions and explain occlusions of joints due to these objects. Thus, joint occlusions can help in both choosing and placing the scenelets. For a given video sequence, we would like to choose and place a scenelet such that the objects of the scenelet explain the joint occlusions observed in the video sequence.

We start by modeling the assignment of scenelets in our dataset to time intervals in the video. Given a video with n_v frames and a dataset with m scenelets, only a single scenelet can start at any frame of the video. The (unknown) scenelet assignment can therefore be expressed with a binary matrix $\mathcal{X} \in \{0,1\}^{m \times n_v}$ with

$$\mathcal{X}_{lt} = \begin{cases} 1 & \text{if scenelet } l \text{ starts at video frame } t \\ 0 & \text{otherwise.} \end{cases}$$

The constraint that scenelets should not overlap in time can be formulated as,

$$\eta_t = \sum_{l} \sum_{i=1}^{\max(t,n_l)} \mathcal{X}_{l(1+t-i)} \leq 1, \quad t = 1 \ldots n_v,$$

where n_l is the number of frames in scenelet l and η_t measures the integer number of scenelets that overlap with frame t and thus needs to be less than or equal to 1. Since only a single scenelet can start at any frame t, we model scenelet placement with one set of parameters per frame $P = \{P_1 \ldots P_{n_v}\}$, where $P_t = (x, y, z, \theta)$ is the placement of the scenelet starting at t, with x, y, and z the location and θ the orientation of the scenelet. The 3D joint locations $\hat{q}_k^t$ in video sequences covered by scenelets can then be defined as a function of the placement P and the scenelet assignment $\mathcal{X}$,

$$\widehat{q}_k^t(P, \mathcal{X}) = \sum_{l} \sum_{i=1}^{\max(t,n_l)} \mathcal{X}_{l(1+t-i)}\, T(P_{(1+t-i)}) s_k^{li}, \tag{6}$$

where s_k^{li} is the 3D position of joint k in frame i of scenelet l and $T(P_t)$ is the transformation due to placement P_t.

Finally, the objects in the scene are obtained from all scenelets that have been assigned to the scene as,

$$O(P, \mathcal{X}) = \bigcup_{\{(l,t) \,|\, \mathcal{X}_{lt}=1\}} T(P_t, O^l),$$

where we denote with $T(P, O)$ the transformation of objects in O to the placement P, *i.e.,* $T(P, O^l) = \{(T(p), \kappa, b) \mid (p, \kappa, b) \in O^l\}$.

7.2 Video Segments with Low Interaction-likelihood

Here, we fit static skeletons to each frame as the segment likely contains unoccluded human performance without object interactions. Since the number of degrees of freedom for human poses is smaller than for human-object interactions, the space of possible human poses can be covered more accurately than the space of possible human-object interactions. Thus, fitting static skeletons to the video gives better performance in unoccluded sequences that do not contain interactions.

The aforementioned 3D human pose reconstruction methods [Rogez et al. 2019; Tomè et al. 2017] retrieve the best matching 3D skeleton pose for a given frame. Such pose, however, is defined in the *local* space of the skeleton and does not give us the placement of the skeleton in the scene. We fit the retrieved 3D skeleton to our video by optimizing the 3D placement of the skeleton, *i.e.,* the variables are only placement attributes of the local poses. We fit skeletons only to frames that do not have any scenelet assignment, *i.e.,* $\eta_t = 0$. In such frames, the joint locations $\check{q}_k^t$ for video sequences are then defined as,

$$\check{q}_k^t(P, \mathcal{X}) = (1 - \eta_t)\, T(P_t) a_k^t, \tag{7}$$

where the first term is only non-zero if no scenelet is assigned to frame t, and a_k^t is the local skeleton pose computed by using Tomè et al.or LCR-Net++, $P_t = (x, y, z, \theta)$ is the placement of the skeleton in frame t, and $T(P_t)$ is the transformation to placement P_t. Combining Equations 6 and 7, we define the location of any joint q_k^t in the video as,

$$q_k^t(P, \mathcal{X}) = \widehat{q}_k^t(P, \mathcal{X}) + \check{q}_k^t(P, \mathcal{X}). \tag{8}$$

In the following, we will omit the explicit dependence of $q_k^t(P, \mathcal{X})$ and $o_i(P, \mathcal{X})$ on P and $\mathcal{X}$ for a less cluttered notation.

7.3 Scene Mapping via a Global Optimization

We have now set up our search space over possible configurations of objects and actor motions, parameterized through the scenelet and pose placements P and the assignment matrix $\mathcal{X}$. We define an energy in this space that can be minimized to obtain a plausible configuration of objects and actor motions given the observations in the video as,

$$L_{\text{global}}(\{u_k^t\}, \mathcal{X}, P) = L(\{u_k^t\}, \mathcal{S}, P) + w_s L_s + w_c L_c + w_m L_m, \tag{9}$$

where the first term denotes how well the current configuration explains the image-space joint detections as described in Equation 1. L_s encourages *smoothness* among human performance; L_c penalizes *intersections* between objects; and L_m penalizes *intersections* between the motion clip and objects.

Smoothness term (L_s). The smoothness term ensures continuity of the synthesized motion by measuring the finite difference approximation of time derivative of the synthesized joint locations as,

$$L_s = \sum_t \left\| q_\lambda^t - q_\lambda^{t-1} \right\|_2^2, \tag{10}$$

where λ is the index of the pelvis joint at video time of frame t.

Object intersection term (L_c). The object intersection term discourages object-object penetration. In our flat scene assumption, all objects are placed on the ground plane. We approximate intersections in 2D, using the projections of objects to the ground plane. Again, to obtain non-zero gradients, which are necessary to resolve intersections in a gradient-based solver, we quantify the amount of penetration using signed distance functions as,

$$L_c = - \sum_{b_i \neq b_j \wedge \theta_i \neq \theta_j} \left(\int_{\Lambda(o_i)} \delta_{o_j}^{-}(x)\, dx + \int_{\Lambda(o_j)} \delta_{o_i}^{-}(x)\, dx \right), \tag{11}$$

where $\delta_{o_i}^{-}$ is the negative part of the signed distance function of object o_i, $\Lambda(o_i)$ is the projection of object o_i to the ground plane, x is a point on the ground plane, b_i is the label of object o_i, and θ_i its orientation. We do not penalize intersecting objects that have the same label and orientation, since we assume these to be representations of the same object placed by different scenelets.

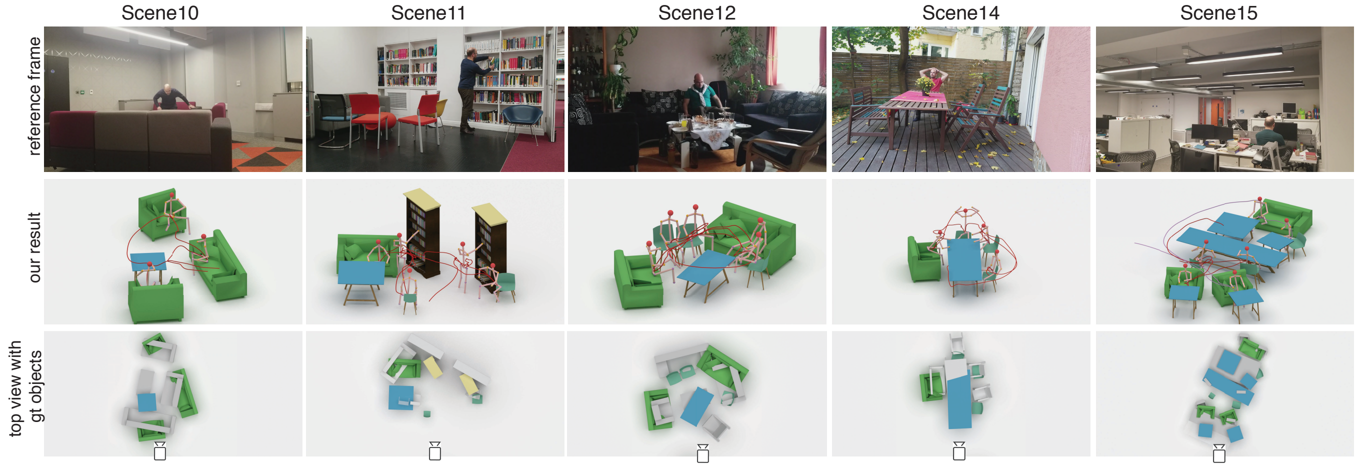


Fig. 5. **Interaction-guided scene layout.** Plausible object layout and human movement as predicted by iMapper on various monocular videos. (Bottom-row) For qualitative evaluation, we overlay, shown from top-view, estimated scene layout versus annotated ground truth. For quantitative evaluation, please refer to Tables 1 and 2. Please refer to supplemental for videos and results.

Motion intersection term (L_m). The motion intersection term discourages humans going through objects. Specifically, the trajectory of the human motion provides information about empty regions in the scene. For efficiency, we compute the intersection in 2D on the ground plane and focus on three joints only: the pelvis joint and the two knee joints. In practice, we have found that taking the maximum 2D distance of these three joints to objects allows a reasonable estimation of full 3D intersections in our scenes. We use,

$$L_m = \sum_t \max_{q \in \{q^t_\lambda,\, q^t_{\Gamma l},\, q^t_{\Gamma r}\}} \min_i \delta_{o_i}(q), \tag{12}$$

where δ_o is the signed distance function of object o, and $q^t_{\Gamma l}, q^t_{\Gamma r}$ are the left- and right-knee joints, respectively.

Global optimization. We obtain the final solution as,

$$\mathcal{X}^\star, P^\star = \operatorname*{argmin}_{\mathcal{X}, P} L_{\text{global}}(\{u^t_k\}, \mathcal{X}, P). \tag{13}$$

We can then utilize the optimized assignment $\mathcal{X}^\star$ and placement $P^\star$ to extract joint locations q^t_k and the placements of objects o_i from selected scenelets and skeletons.

The above optimization is challenging given the mix of discrete and continuous parameters, and a highly non-linear energy function. We simplify the task by progressively performing the optimization.

(i) First, we estimate *interaction-saliency* for all frames of the video (as described in Section 5.2).

(ii) Next, starting with a smaller set of candidates ($\mathbb{K}_{\text{loc}}$) for the high-interaction likelihood video segments (as described in Section 6), we evaluate a selection of computationally-efficient energy terms (Equations 2, 3, 10, 12) locally.

(iii) Finally, by committing to a smaller set of high-scoring candidates ($\mathbb{K}_{\text{glob}}$, as described in Section 7), we optimize placements P of all fitted models in the scene, both local skeletons and scenelets, using the full energy term. Starting from the local optima from (ii), we optimize Equation 13, which translates to adding Equation 11 and inter-scenelet versions of Equations 3 and 12 to the energy function.

The scenelet assignment $\mathcal{X}$ is optimized indirectly by filtering out candidates in each stage of the decomposed optimization instead of immediately invoking an integer program. For simpler scenes, we perform one optimization for all combinations of the few remaining candidates and keep the combination that results in the scene with the lowest fitting energy.

For each of the three stages, described respectively in Sections 5.2, 6 and 7, we perform an optimization (using the quasi-Newton solver L-BFGS-B [Byrd et al. 1995]) over the placement parameters P. We implemented the optimization in Tensorflow[2] and optimize using a Titan X (Pascal) with 12GB memory. The gradient tolerance termination criterion is set to its minimum value 10^{-12}. We optimize fitting all scenelets (≈ 1500) to a single frame at once with the terms in Equations 2 and 10 to estimate the interaction-likelihood score. For each local maximum of the interaction-likelihood function over time, we re-optimize the top $\mathbb{K}_{\text{loc}} = 200$ fits as described in (i) in batches of 10-25 due to the larger memory requirements of our implementation of Equation 3.

8 RESULTS AND DISCUSSION

We tested iMapper on a range of input monocular videos of varying complexity. For each of these benchmark videos (recorded in-house), we also manually annotated ground truth object placements, 3D actor poses, and action labels. Table 1 shows statistics of these videos while Figure 5 shows some examples. After we qualitatively discuss some results, we report how ground truth was annotated and explain our evaluation protocol. Next, we provide quantitative evaluations, comparisons against baseline methods, and an ablation study. Please refer to the supplemental for the full input videos and annotations.

[2] http://www.tensorflow.org/api_docs/python/tf/contrib/opt/ScipyOptimizerInterface

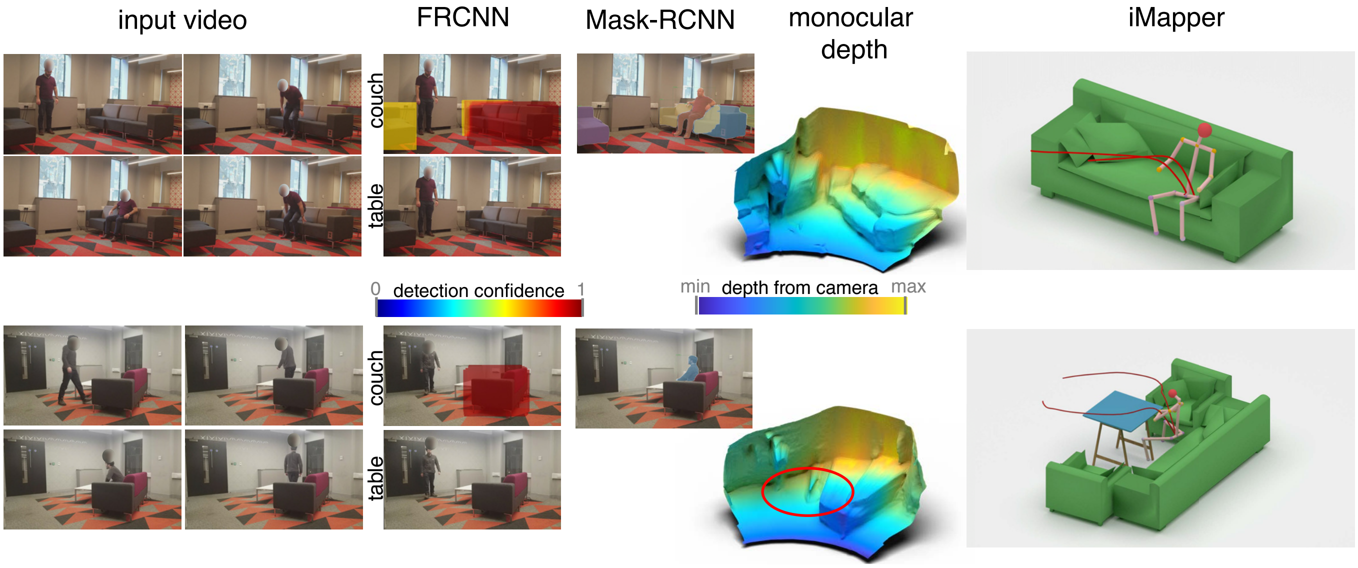


Fig. 6. **Object detection comparison.** Qualitative comparisons to state-of-the-art object detection methods. Note that FRCNN and Mask-RCNN both produce only 2D image-space segments. iMapper synthesizes 3D scenes that plausibly match the video, even in sequences where the objects and the human performance are occluded. Encircled in red in the bottom depth map are the typical errors near depth discontinuities obtained from image depth estimators.

Qualitative evaluation

We show our results on some of the 15 different scenes from the Interaction Benchmark Dataset (see Section 8.1) in Figures 1 and 5. For each scene, we show a reference video frame. The recovered trajectory of the actor's pelvis is shown as a colored line and the colored skeleton shows an occluded actor pose. To help assess that iMapper finds plausible object arrangements and actor poses for occluded parts of the scene, we also show renderings of our results from the top view along with overlaid ground truth object locations. To better visualize the detected orientation of objects, we use object bounding boxes to scale and place category specific proxy object geometries in our scenelets – please note the meshes are *not* output by our method. Please refer to supplemental for full videos of the input and the generated scenes.

Note how much information is encoded in the interactions, enabling plausible reconstruction of the original scenes. Although we cannot hallucinate objects that are not interacted with, the quality of the generated scenes improves over time as more interactions 'reveal' the true underlying scene. As shown in Figure 13, as the same environment is explored over time, our system recovers larger parts of the object arrangement. Further, perturbations to the input (*e.g.*, in the form of the same action being performed by different people, or at different times) lead to slightly different, but still plausible and consistent scenes.

8.1 Interaction Benchmark Dataset (i3DB)

We created a benchmark dataset consisting of 15 scenes by semi-automatically annotating both object locations and 3D world-space actor poses for each input video. To the best of our knowledge, this is the first benchmark dataset for scenes with medium to heavy occlusion containing both object and 3D human pose annotations over time.

8.1.1 Object location annotations. For *object locations*, we physically measured the dimensions of the scene objects and positioned objects as collections of oriented bounding boxes in the ground truth scene to minimize video reprojection error while using known camera intrinsics. We also added class labels (*e.g.*, `chair`, `table`, `shelf`) for each individual object. To evaluate the performance of iMapper and alternative methods in terms of object placement quality, we define two metrics (see Section 8.2 for comparison results).

Object Location Measure 1 (ObjCt). We define the metric ObjCt that counts the number of objects detected along with correctly labeled object class. We consider objects of interest as those that are participating in at least one interaction during the recorded video.

Object Location Measure 2 (ObjPosn). We define the metric ObjPosn that measures the mean and standard deviation of the distances between the predicted and the ground truth object centroids in the scene.

8.1.2 Human pose annotations. For human poses, we used an assisted approach to generate the ground truth since manually annotating 3D poses in each frame is not feasible for 1000s of frames. We started with estimated 2D joint locations [Wei et al. 2016] and then manually corrected them. These corrected 2D locations were then lifted first to local 3D space by running [Tomè et al. 2017] – which works well in the absence of occlusion – and then to world space 3D using the reprojection and smoothness energy terms described in Section 6. Finally, we inspected the output 3D path, moved the skeletons to fit the ground truth scene layouts from Section 8.1.1 and added these corrections as additional constraints for the hip joints to the optimization. This process was repeated until we found no more significant errors. The number of corrections depended on the amount of occlusion in the scene, typically ranging from 10-50. When Tomè et al. [2017] results deviated significantly from the actual human pose, we manually corrected the 3D poses using Blender. Finally, we used ray-tracing to establish ground truth visibility of the 3D joints using the oriented bounding boxes of the ground truth scene layouts.

To evaluate the performance of iMapper and alternative methods in terms of 3D human performance reconstruction quality, we define two metrics (see Section 8.3 for comparison results).

Human Pose Measure 1 (PoseEst). As a direct measure, we report the root mean square error over 14 joint locations that are common across the annotated 2D poses and 3D poses iMapper recovers. We compute this error both in local 3D (using origin at pelvis) space ('LC') and world space ('WC'), where available.

Human Pose Measure 2 (ActnDetn). As an indirect measure, we train a pose-based action recognition network and evaluate how well the output poses can predict the ground truth action labels manually annotated at each frame of the input video. Specifically, we use the pose-based action recognition network proposed by Luvizon et al. [2018], which takes as input 3D human poses (represented as 16 joints) for each video clip (we use a window size of 20 frames) and outputs the probability of each action class. The original network is trained using the NTU dataset [Shahroudy et al. 2016] that provides 60 action classes. Most of these action classes are performed both in sitting and standing positions (*e.g.,* eating a snack, brushing teeth) and are not covered by the scenelet dataset we use [Savva et al. 2016]. In order to avoid any ambiguities, we retrain this network using action classes that are included both in our and the NTU dataset (*i.e.,* `walk`, `sitDown`, and `standUpFromSittingPosition`).

8.2 Evaluating Object Placement Quality

Object Location Measure 1 (ObjCt). We compare our method to per-frame region detection methods, FRCNN [Ren et al. 2017] and Mask-RCNN [He et al. 2017]. We show qualitative comparisons in Figure 6, and quantitative results in terms of metric ObjCt are given for Mask-RCNN in Table 1 (column 'MR'). For Mask-RCNN, we count the number of objects where at least 50% of the object's region was detected and correctly labeled on average, over all the frames. Since FRCNN and Mask-RCNN are designed to detect visible objects, they naturally fail to detect any objects 'hidden' behind visible objects leading to a low number of detections. Other systems that rely on these methods as their primary building blocks will have similar problems in occluded regions. In comparison, iMapper detects all the objects that participate in an interaction even if they are occluded.

Object Location Measure 2 (ObjPosn). We compare against per-pixel monocular depth estimation [Chakrabarti et al. 2016] using the second metric, ObjPosn. Table 1 shows the mean and standard deviation of the distances between the predicted and the ground truth object centroids in the scene for monocular depth estimation (column 'GT+MD') and iMapper (column 'iM'). For the monocular depth map, we approximate the object centroid as the mean world position of all samples that are inside the manually annotated ground truth 2D region of an object (*i.e.,* providing an upper bound on region detectors). Objects without a single visible pixel are ignored. The depth map contains only limited information about partially or fully occluded objects, resulting in large errors. As shown in Figure 6, fourth column, even for visible regions the estimated depths are smoothed out and fail to capture the object specific layout. In contrast, iMapper produces plausible objects along with their spatial locations.

8.3 Evaluating Actor Pose Quality

Human Pose Measure 1 (PoseEst). Most monocular 3D pose detection methods compute only *local* 3D poses, *i.e.,* joint locations relative to pelvis, limiting our choice of baselines for our first metric. We compare to Tomè et al. [2017], and LCRNet++ [Rogez et al. 2019] which both output local 3D joint locations, but do not provide world-space coordinates. Therefore, we use our method to optimize and lift their local predictions to world space (referred to as Tome3D and LCRNet++3D, respectively) and compare them in Table 2. While errors in local coordinates are reported in cm units, we report the errors in world coordinates as a fraction of the top-view 2D diagonal of the axis-aligned box of the ground truth path to be invariant to scene sizes. We define five categories based on the amount of

Table 1. **Performance statistics.** List of scenes presented showing frame count (ct., N), fraction of frames with occlusion (frac., η), number of objects in the scene (obj., s_o) and number of objects with interactions (s_i). For comparison, we list number of objects with interactions detected by Mask-RCNN (MR, n_{MR}) and by iMapper (iM, n_{iM}). Also, we show quality of depth estimation error in cm by ground truth mask + MonoDepth (GT+MD, $\mu_{\text{GM}}(\sigma_{\text{GM}})$) and iMapper as mean (s.d.) (iM, $\mu_{\text{iM}}(\sigma_{\text{iM}})$) compared against ground truth annotations. Note that the scenes are ordered based on increasing difficulty which we assess by higher $N * \eta$ value.

scene	ct.	frac.	obj.	MR	iM	GT+MD	iM
	N	η	$s_i(s_o)$	n_{MR}	n_{iM}	$\mu_{\text{GM}}(\sigma_{\text{GM}})$	$\mu_{\text{iM}}(\sigma_{\text{iM}})$
Scene1	80	0.00	1 (2)	2	1	120 (-)	**45** (–)
Scene2	49	0.84	2 (4)	2	2	197 (24)	**135** (13)
Scene3	130	0.57	2 (4)	4	2	120 (10)	**72** (06)
Scene4	115	0.77	2 (2)	2	2	203 (12)	**70** (57)
Scene5	120	0.82	1 (1)	1	1	191 (-)	**25** (–)
Scene6	148	0.93	3 (4)	2	3	230 (52)	**72** (41)
Scene7	254	0.65	3 (3)	1	3	243 (98)	**70** (55)
Scene8	182	1.00	4 (4)	2	4	229 (48)	**53** (49)
Scene9	189	0.98	4 (4)	2	4	216 (50)	**69** (38)
Scene10	224	0.97	4 (4)	1	4	259 (101)	**51** (21)
Scene11	539	0.49	6 (11)	5	6	174 (73)	**83** (29)
Scene12	380	0.84	5 (5)	4	5	98 (36)	**57** (32)
Scene13	348	1.00	7 (14)	8	7	263 (117)	**81** (43)
Scene14	430	0.86	5 (5)	5	5	242 (68)	**58** (28)
Scene15	600	0.98	12 (14)	7	12	287 (120)	**68** (24)
μ	252.5	0.78	4.1 (5.4)	3.2	4.1	205 (62.2)	**67** (33.5)

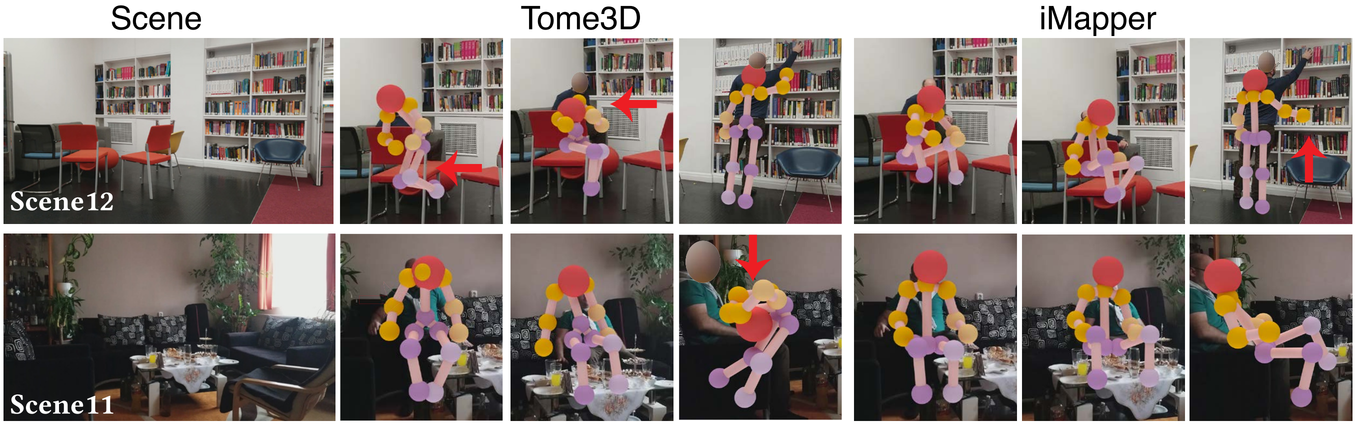


Fig. 7. **Human pose detection comparison.** Qualitative comparisons to Tomè et al. [2017], which produces image-space and local 3D poses. Note that Tomè and colleagues do not compute world space positions of the skeletons. For better comparison, we position them in world coordinates using the hip locations as estimated by iMapper. Relevant differences between the methods are marked with red arrows. Note that our method gives plausible skeleton poses in many cases whereas the method of Tomè et al. fails, especially in occluded areas.

error: `BAD(> 25cm)`, `FAIR(20 − 25cm)`, `FINE(15 − 20cm)`, `GOOD(10 − 15cm)`, `EXCELLENT(< 10cm)`.

Both Tome3D and LCRNet++3D are designed to capture the actual poses seen in the input sequence and thus achieve reasonable performance on average. However, they often produce unrealistic poses in occlusion regions as seen in the supplemental video. On the other hand, the goal of our method is to generate plausible poses even in highly occluded interaction regions. While showing the same interactions, the poses in the matched scenelets may differ from the ground truth poses resulting in lower accuracy (we distinguish between accuracy and plausibility). Nevertheless, the accuracy of the poses generated by iMapper are still in `FINE − GOOD` range. Qualitative comparisons to Tome3D in Figure 7 verify these observations. In non-occluded frames, their poses are very close to ours (up to the smoothness term L_s). In occluded frames however, Tome3D returns unrealistic poses while iMapper continues to generate plausible poses in alignment with the discovered interaction. Figure 8 shows how such errors evolve over time for the three methods as well as the number of occluded joints in the ground truth.

Table 2. **PoseEst evaluation.** Comparing iMapper against Tome3D and LCR-Net++3D, which both return only local coordinates (LC) and we use our optimization to lift them to world coordinates (WC). LC units are in cm, WC units are in fraction of the top-view 2D diagonal of the axis-aligned bounding box of the ground truth path.

	Tome3D		LCR-Net++3D		iMapper	
	LC	WC	LC	WC	LC	WC
Scene4	FINE 19.9 (+70%)	FAIR .197 (+0.2%)	FAIR 21.5 (+83%)	FAIR .196	**GOOD** 11.7	BAD .225 (+15%)
Scene5	FINE 18.5 (+13%)	FINE .103 (+11%)	FAIR 21.9 (+34%)	**GOOD** .093	FINE 16.4	FINE .123 (+32%)
Scene7	BAD 26.3 (+39%)	FAIR .166 (+108%)	FAIR 20.1 (+6%)	FINE .123 (+54%)	FINE 19.0	**GOOD** .079
Scene10	FINE 16.6	**GOOD** .083 (+30%)	FAIR 22.3 (+34%)	FINE .135 (+111%)	FINE 19.1 (+15%)	**GOOD** .064
Scene11	**GOOD** 12.5	FINE .139 (+32%)	**GOOD** 14.0 (+12%)	FINE .105	FINE 17.3 (+39%)	FINE .127 (+20%)
Scene12	**GOOD** 13.2	FINE .114 (+64%)	**GOOD** 13.3 (+1%)	**GOOD** .070	FINE 16.2 (+23%)	**GOOD** .080 (+14%)
Scene13	FAIR 23.9	FINE .111 (+96%)	BAD 25.9 (+8%)	**GOOD** .057	BAD 28.1 (+17%)	**GOOD** .059 (+4%)
Scene14	**GOOD** 11.7	**GOOD** .089 (+45%)	**GOOD** 12.8 (+9%)	**GOOD** .065 (+6%)	**GOOD** 14.8 (+27%)	**GOOD** .062
μ	**17.11**	0.117	18.33	0.095	18.47	**0.087**

We also compare to Vnect [Mehta et al. 2017b] that provides world-space 3D pose estimates. A qualitative comparison is given in Figure 9. Since we do not have direct access to their source code, we compare to this method quantitatively using the same scene as in Figure 9 which is from the *MPI-INF-3DHP* dataset [Mehta et al. 2017a], where a relatively high-quality ground truth is available. While Vnect achieves an RMSE error of .399 and 28.9 in world-space and local-space respectively, iMapper achieves a lower error of .235 and 23.5.

Human Pose Measure 2 (ActnDetn). In Table 3, we compare the results of a pose-based action detection network [Luvizon et al. 2018] on pose sequences obtained by Tome3D, LCRNet++3D, and iMapper. We report precision and accuracy for the duration of the sequence when interactions happen (by excluding the regions where the ground truth action label is `walking`) to focus evaluation around frames with occlusion. While Tome3D achieves reasonable precision, it does not produce reasonable 3D poses in occluded regions leading to low recall. While LCRNet++3D achieves higher recall, the estimated poses are often not accurate leading to low precision. In comparison, iMapper generally improves both in terms of precision and recall. Specifically, when we consider regions where interactions occur, iMapper improves by a large margin by recovering plausible poses from matching scenelets.

To verify that the scenes and human performance captures that iMapper generates are aligned with user expectations, we also conducted a comparative user study. For each question, we showed the users an input video of initial 2D pose estimates which are often potentially noisy and miss joints in occlusion regions. Note that

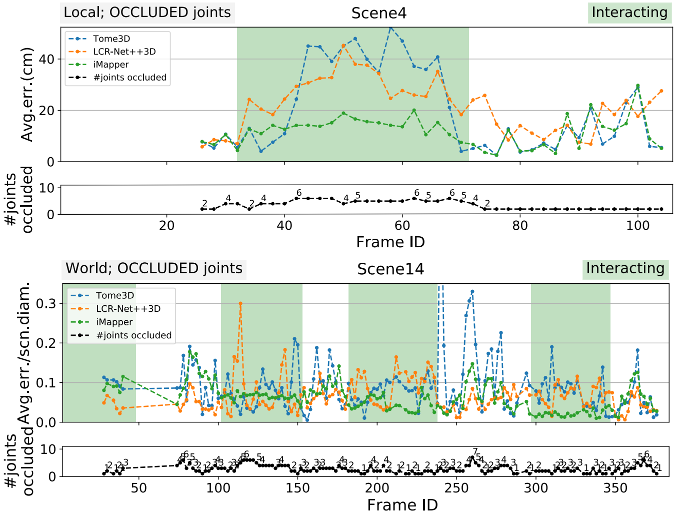


Fig. 8. Comparison of the position accuracy of occluded joints over time. Green background denotes time intervals that contain interactions. Lower values are better. For world space, error metrics have been normalized by the top-view diagonal of the scene bounding box. Our algorithm detects and processes interactions and hence recovers from space-time segments with occlusions, as shown by the lower average error in green regions. However, outside interactions, the algorithm fits static skeletons to each frame, resulting in comparable performance to other methods. LCR-Net++3D tends to place ankle joints intersecting the floor during interactions, whilst Tome3D often yields physically not possible human poses. Note that although our error is not lowest in all frames, the poses detected by iMapper typically represent more plausible interactions, as shown in Table 3.

Table 3. **ActnDetn evaluation.** Comparison of pose-based action detection [Luvizon et al. 2018] on human performance reconstruction by iMapper against those from Tome3D and LCR-Net++3D. We report both precision and recall for the parts of the input sequences that are annotated to be non-`walking` in the ground truth. We also report mean and (std. dev.) precision and recall across all the scenes. The corresponding pose-based action detection numbers for all frames (*i.e.,* `walking` or not) across scenes in this table are quite similar and included in supplemental material.

	Tome3D		LCR-Net++3D		iMapper	
	precision	recall	precision	recall	precision	recall
Scene4	0.00	0.00	0.00	0.00	**0.36**	**0.29**
Scene5	0.00	0.00	0.00	0.00	**0.13**	**0.09**
Scene7	0.07	0.01	0.13	0.12	**0.45**	**0.33**
Scene9	0.00	0.00	0.15	0.15	**0.57**	**0.57**
Scene10	0.08	0.03	0.11	0.11	**0.22**	**0.18**
Scene12	0.15	0.12	0.36	0.36	**0.64**	**0.51**
Scene13	0.00	0.00	0.14	0.14	**0.24**	**0.24**
Scene14	0.08	0.07	0.39	0.39	**0.50**	**0.50**
Overall	0.05 (0.06)	0.03 (0.04)	0.16 (0.15)	0.16 (0.15)	**0.39** (0.18)	**0.34** (0.17)

we do not show them the RGB videos as this is essentially also the input to iMapper and baselines. As output, we showed two different explanations of the input (*i.e.,* 3D scene and human performances) generated by iMapper and one of the two baseline methods (see Figure 10). For iMapper, we present the raw output results. For the two baselines, we accompanied the 3D pose trajectories obtained by Tome3D or LCRNet++3D with objects detected by MaskRCNN. For any object detected by MaskRCNN and matching the ground truth, we compute the 3D object centroid from a monocular depth estimate [Chakrabarti et al. 2016] and place the object to the centroid location by snapping it to the ground truth floor and manually orient the object based on the ground truth annotation.

8.4 User Study

We asked the users to select the method output that resembles the input the most judging based on how plausible the recovered interactions along with the involved objects are. We conducted the study for 5 scenes (Scene7, Scene9, Scene10, Scene13, Scene14) which resulted in a total of 15 pairs of queries (first and second ordering for any video was also flipped at random); 10 of these queries

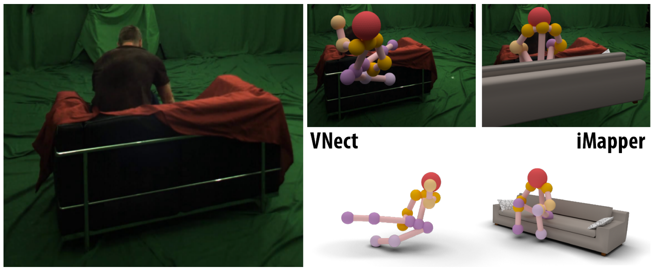


Fig. 9. **Comparison with VNect.** State-of-the-art 3D human pose detection from monocular video VNect [Mehta et al. 2017b] breaks down in regions of occlusion (VNect was not designed to handle occlusions), while iMapper continues to produce plausible results because of explicit occlusion detection and handling. Bottom row shows the recovered human pose from another camera angle for better visibility.

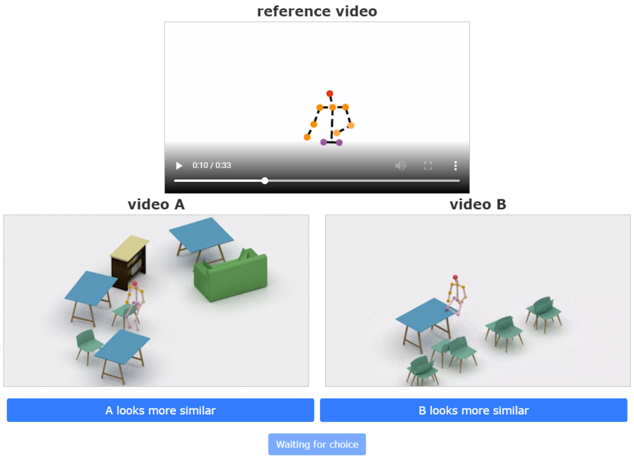


Fig. 10. **User study.** We showed users an input video of initial 2D pose estimates and two different videos selected at random among the three explanations generated by iMapper and the two of the baseline methods.

compared iMapper to alternatives whereas 5 queries compared the two alternative methods to each other. We had 25 unique users answer the queries for a total of 590 answers. We asked each query twice in random order to measure the consistency of the users and weighted the answers by the consistency. Users preferred iMapper output vs LCRNet++3D 99% the time and always preferred iMapper output over Tome3D. When comparing LCRNet++3D vs Tome3D, users preferred Tome3D 91% of the time. There is a strong user preference for iMapper outputs since the generated scenes are consistent with the input motion. In contrast, alternative methods most of the time result in missing objects involved in interactions due to occlusions, and lower quality of captured human performance, specifically in the occlusion regions.

8.5 Ablation Study

We report the effect of removing some of the terms from our optimization in Figure 11. For example, the occlusion term has a heavy influence if multiple joint locations are occluded in the video. Recall that our occlusion term is assymmetric, so that more occlusion always has less cost. The couch can thus be optimized to be closer to the camera to occlude more of the scene (the camera is at the bottom center of the image). The smoothness and reprojection terms also affect the result since they are used throughout the entire pipeline. Omitting the smoothness term can lead to path deformations, while leaving out the reprojection term removes the anchoring of the motions to the video and permits the smoothness term to possibly contract the path.

Hold-one-out validation. We also evaluated iMapper on a hold outset taken from the original PiGraphs dataset. We pick a single scene and remove the scenelets that were generated from this scene from our database, accounting for about 10% of our scenelets. We compare to the ground truth objects given in the PiGraphs dataset through manually established correspondence, and recorded deviations in world-space object placement using the ObjPosn metric. The mean reconstruction error was 110 cm (s.d. 56 cm) with all relevant objects detected.

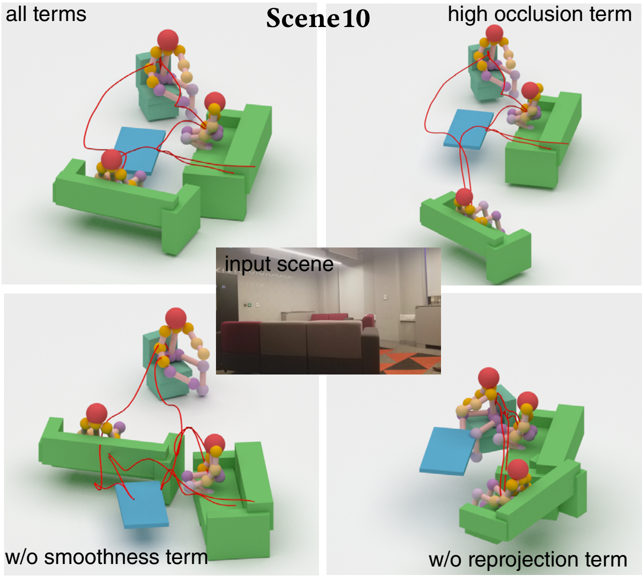


Fig. 11. **Ablation study.** We test the effect of various terms on the solution for the scene shown in the center.

8.6 Limitations

We next discuss the current limitations of our method.

(i) In our formulation, we directly inherit human performance from the selected scenelets. Hence, we cannot correct any noise in the human motion as recorded in the interaction database. This can be addressed either by using a higher fidelity data capture or adaptively denoising the motion sequence.

(ii) Since our method only uses 2D joint detections from the raw video and in absence of any image-space cues (*i.e.,* the objects are occluded), the method cannot provide any estimate about the size and extent of the objects, and hence cannot often distinguish between say a chair or a sofa.

(iii) Our method expects enough visible movement in the input video to trigger sufficient fraction of 2D joints to be detected. Otherwise, subsequent scenelet matching will fail, and in absence of matched scenelets, the object layout will remain incomplete. Similarly, occluded objects that remain un-interacted by the actor are likely to go undetected.

(iv) Finally, since our method builds on the expectation that people react similarly in similar settings, it will naturally get confused when this assumption is broken. For example, if a person decides to hand-walk, or use a sofa as a bed, *etc.*

9 CONCLUSION AND FUTURE WORK

We presented iMapper that takes as input a monocular footage of a person interacting with objects in a physical space and produces a plausible scene layout along with consistent human performances explaining the footage. iMapper detects and leverages human-object interactions in the input video to resolve medium to high level of occlusions that occur in such footage. At the heart of iMapper lies a novel data-driven method to assign *interaction-likelihood* scores to video segments that help identify space-time moments when matched human-object interactions provide reliable cues about the surrounding scene layout. Our method retrieves corresponding interaction-salient scenelets as candidates fitted to the informative video segments, and then selects and positions them to form a global scene layout and 3D human pose estimates. We introduced the **i**3DB benchmark dataset for evaluating quality of interactions computed by different methods. Our qualitative and quantitative evaluation demonstrate that iMapper produces realistic scene layouts as well as 3D pose estimates.

9.1 Future Directions

Exciting research directions lay ahead as we are only starting to capture, analyze, and understand the space of (human) interactions, or *interaction landscapes* (*cf.,* [Pirk et al. 2017b]). Below we discuss some of the immediate issues.

Capturing richer interaction databases. Current datasets only capture limited variety of interactions, both in terms of different types of interactions and variance for each interaction type. For example, we miss examples of interactions with small objects (*e.g.,* picking up a cup/glass, using pots and pans in kitchens, lifting a bag or suitcase), or examples of the different ways that people sit in sofas, couches, chairs, *etc.* While significant progress has been made in capturing static environments at high geometric detail, capturing interactions remains *fundamentally* difficult because of heavy occlusion arising due to the interactions. One possibility is to separate the capture of static geometry (*e.g.,* with mobile 3D scanners) from the capture of interactions using a mix of sensors such as IMU sensors, RGBD scanners, markers, *etc.*

Utilizing scene priors. So far we used signals only from human-object interactions. However, in scenes with heavy occlusion, scenelet matching with partial (occluded) information may not be sufficient to accurately ground object positions. One possibility is to additionally use scene statistics and local context, as has been heavily utilized in scene synthesis research, to regularize the interaction-based layout reconstructions.

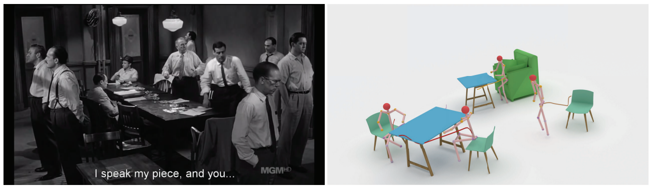


Fig. 12. iMapper result on an input scene with multiple actors.

Handling multiple actors. A next opportunity will be to extend iMapper to handle multiple actors. This will involve extending the 2D human pose detector to multiple actors by tracking instance correspondence over time. Although our method naturally generalizes to this scenario, we have to obtain suitable priors to handle human-human interactions. Figure 12 shows a first result.

Recovering interactions over large timescales. As shown in Figure 13, iMapper has only a chance of recovering scene arrangements once people interact with parts of the environment. This suggests that the approach gets better as we 'observe' the scene over larger timescales, ideally days or weeks. However, then our static scene assumption breaks down as objects are going to be shifted and moved around. Hence, we would like to extend our approach to also capture space-time object movements, starting with rigid movement of objects, such as a moving chair.

ACKNOWLEDGMENTS

This project was supported by an ERC Starting Grant (SmartGeometry StG-2013-335373), ERC PoC Grant (SemanticCity), Google Faculty Awards, Google PhD Fellowships, Royal Society Advanced Newton Fellowship and gifts from Adobe. We thank our anonymus reviewers for their input, and Sebastian Friston, Tuanfeng Y. Wang, Moos Hueting, Drew MacQuarrie, Denis Tomè, Dushyant Mehta, Grégory Rogez, Manolis Savva and Veronika Monszpart-Benis for their help and advice.

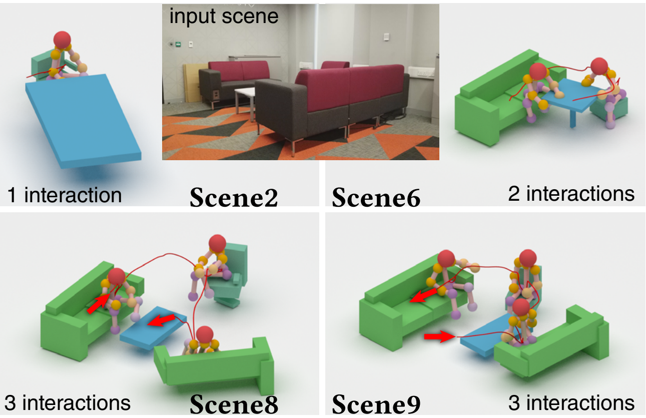


Fig. 13. **Progressive scene exploration.** We show results of four different videos taken from the same scene. As interactions with more objects are made available, we can recompute the results to synthesize additional objects. Variations of scene explorations, for example performing the interactions in reverse order, as shown in the bottom row, give slightly different, but comparable and plausible results.

REFERENCES

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. In *SISC.*

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *IEEE CVPR.*

Ayan Chakrabarti, Jingyu Shao, and Greg Shakhnarovich. 2016. Depth from a Single Image by Harmonizing Overcomplete Local Network Predictions. In *NIPS.*

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *3DV.*

Kang Chen, Yu-Kun Lai, Yu-Xin Wu, Ralph Martin, and Shi-Min Hu. 2014. Automatic Semantic Modeling of Indoor Scenes from Low-quality RGB-D Data Using Contextual Information. In *ACM SIGGRAPH Asia.*

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017a. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE CVPR.*

Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. 2017b. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. In *ACM TOG.*

Luca Del Pero, Joshua Bowdish, Bonnie Kermgard, Emily Hartley, and Kobus Barnard. 2013. Understanding Bayesian Rooms Using Composite 3D Object Models. In *IEEE CVPR.*

Vincent Delaitre, David F. Fouhey, Ivan Laptev, Josef Sivic, Abhinav Gupta, and Alexei A. Efros. 2012. Scene semantics from long-term observation of people. In *ECCV.*

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based Synthesis of 3D Object Arrangements. In *ACM SIGGRAPH Asia.*

Matthew Fisher, Manolis Savva, and Pat Hanrahan. 2011. Characterizing structural relationships in scenes using graph kernels. In *ACM SIGGRAPH.*

Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric Scene Synthesis for Functional 3D Scene Modeling. In *ACM SIGGRAPH Asia.*

David F. Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A. Efros, Ivan Laptev, and Josef Sivic. 2012. People Watching: Human Actions as a Cue for Single-View Geometry. In *ECCV.*

Barbara Frank, Michael Ruhnke, Maxim Tatarchenko, and Wolfram Burgard. 2015. 3D-reconstruction of indoor environments from human activity. In *IEEE ICRA.*

Lianrui Fu, Junge Zhang, and Kaiqi Huang. 2015. Beyond Tree Structure Models: A New Occlusion Aware Graphical Model for Human Pose Estimation. In *IEEE ICCV.*

Qiang Fu, Xiaowu Chen, Xiaoyu Su, and Hongbo Fu. 2017a. Pose-Inspired Shape Synthesis and Functional Hybrid. In *IEEE TVCG.*

Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. 2017b. Adaptive Synthesis of Indoor Scenes via Activity-Associated Object Relation Graphs. *ACM SIGGRAPH Asia.*

Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. 2018. Detecting and Recognizing Human-Object Interactions. In *IEEE CVPR*.

Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. 2009. Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition. In *IEEE PAMI*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *IEEE ICCV*.

Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. 2016. Learning How Objects Function via Co-analysis of Interactions. In *ACM TOG*.

Ruizhen Hu, Chenyang Zhu, Oliver van Kaick, Ligang Liu, Ariel Shamir, and Hao Zhang. 2015. Interaction Context (ICON): Towards a Geometric Functionality Descriptor. In *ACM TOG*.

Chun-Hao Huang, Edmond Boyer, Nassir Navab, and Slobodan Ilic. 2014. Human Shape and Pose Tracking Using Keyframes. In *IEEE CVPR*.

Jia-Bin Huang and Ming-Hsuan Yang. 2009. Estimating Human Pose from Occluded Images. In *ACCV*.

Shi-Sheng Huang, Hongbo Fu, and Shi-Min Hu. 2016. Structure guided interior scene synthesis via graph matching. In *Graphical Models*.

Moos Hueting, Pradyumna Reddy, Ersin Yumer, Vladimir G. Kim, Nathan Carr, and Niloy J. Mitra. 2018. SeeThrough: Finding Objects in Heavily Occluded Indoor Scene Images. In *3DV*.

Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. In *ECCV*.

Hamid Izadinia, Qi Shan, and Steven M Seitz. 2017. IM2CAD. In *CVPR*.

Yun Jiang, Hema S. Koppula, and Ashutosh Saxena. 2016. Modeling 3D Environments Through Hidden Human Context. In *IEEE PAMI*.

Changgu Kang and Sung-Hee Lee. 2017. Scene reconstruction and analysis from motion. In *Graphical Models*.

Vladimir G. Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. 2014. Shape2Pose: Human-Centric Shape Analysis. In *ACM SIGGRAPH*.

Leonard Krasner. 2013. *Environmental Design and Human Behavior.* Elsevier.

Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G. Kim, Qixing Huang, Niloy J. Mitra, and Thomas Funkhouser. 2014. Creating Consistent Scene Graphs Using a Probabilistic Grammar. In *ACM SIGGRAPH Asia*.

Diogo C. Luvizon, David Picard, and Hedi Tabia. 2018. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In *IEEE CVPR*.

Rui Ma, Honghua Li, Changqing Zou, Zicheng Liao, Xin Tong, and Hao Zhang. 2016. Action-driven 3D Indoor Scene Evolution. In *ACM SIGGRAPH Asia*.

Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. 2015. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE CVPR*.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*.

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. 2017a. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. In *3DV*.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017b. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. In *ACM SIGGRAPH*.

Liangliang Nan, Ke Xie, and Andrei Sharf. 2012. A Search-classify Approach for Cluttered Indoor Scene Understanding. In *ACM SIGGRAPH Asia*.

Ulric Neisser. 1976. *Environmental Design and Human Behavior.* W. H. Freeman.

Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*.

Sören Pirk, Olga Diamanti, Boris Thibert, Danfei Xu, and Leonidas J. Guibas. 2017a. Shape-Aware Spatio-Temporal Descriptors for Interaction Classification. In *IEEE ICIP*.

Sören Pirk, Vojtech Krs, Kaimo Hu, Suren Deepak Rajasekaran, Hao Kang, Yusuke Yoshiyasu, Bedrich Benes, and Leonidas J. Guibas. 2017b. Understanding and Exploiting Object Interaction Landscapes. In *ACM SIGGRAPH Asia*.

Patrick Poirson, Phil Ammirato, Cheng-Yang Fu, Wei Liu, Jana Kosecká, and Alexander C. Berg. 2016. Fast Single Shot Detection and Pose Estimation. In *3DV*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *IEEE PAMI*.

Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. 2019. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. In *IEEE PAMI*.

Scott Satkin and Martial Hebert. 2013. 3DNN: Viewpoint Invariant 3D Geometry Matching for Scene Understanding. In *IEEE CVPR*.

Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2014. SceneGrok: Inferring Action Maps in 3D Environments. In *ACM SIGGRAPH Asia*.

Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2016. PiGraphs: Learning Interaction Snapshots from Observations. In *ACM SIGGRAPH*.

Alexander G. Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. 2013. Box in the Box: Joint 3D Layout and Object Reasoning from Single Images. In *IEEE ICCV*.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *IEEE CVPR*.

Tianjia Shao*, Aron Monszpart*, Youyi Zheng, Bongjin Koo, Weiwei Xu, Kun Zhou, and Niloy Mitra. 2014. Imagining the Unseen: Stability-based Cuboid Arrangements for Scene Understanding. In *ACM SIGGRAPH Asia*. * Joint first authors.

Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. 2012. An Interactive Approach to Semantic Modeling of Indoor Scenes with an RGBD Camera. In *ACM SIGGRAPH Asia*.

Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. 2016. Direct Prediction of 3D Body Poses from Motion Compensated Sequences. In *IEEE CVPR*.

Denis Tomè, Chris Russell, and Lourdes Agapito. 2017. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. In *IEEE CVPR*.

Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *IEEE CVPR*.

Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. 2018. Factoring Shape, Pose, and Layout from the 2D Image of a 3D Scene. In *IEEE CVPR*.

Timo von Marcard, Bodo Rosenhahn, Michael J. Black, and Gerard Pons-Moll. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. In *CGF Eurographics*.

Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2018. Deep convolutional priors for indoor scene synthesis. In *ACM TOG*.

Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. 2013. Modeling 4D Human-Object Interactions for Event and Object Recognition. In *IEEE ICCV*.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *IEEE CVPR*.

Xiaolin Wei and Jinxiang Chai. 2010. VideoMocap: Modeling Physically Realistic Human Motion from Monocular Video Sequences. In *ACM TOG*.

Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. 2012. Accurate Realtime Full-body Motion Capture Using a Single Depth Camera. In *ACM TOG*.

Kai Xu, Rui Ma, Hao Zhang, Chenyang Zhu, Ariel Shamir, Daniel Cohen-Or, and Hui Huang. 2014. Organizing Heterogeneous Scene Collections Through Contextual Focal Points. In *ACM SIGGRAPH*.

Bangpeng Yao, Aditya Khosla, and Li Fei-Fei. 2011. Classifying Actions and Measuring Action Similarity by Modeling the Mutual Context of Objects and Human Poses. In *ICML*.

Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D. Goodman, and Pat Hanrahan. 2012. Synthesizing Open Worlds with Constraints Using Locally Annealed Reversible Jump MCMC. In *ACM SIGGRAPH*.

Hong-Bo Zhang, Qing Lei, Bi-Neng Zhong, Ji-Xiang Du, and JiaLin Peng. 2016. A Survey on Human Pose Estimation. In *Intelligent Automation and Soft Computing*.

Xi Zhao, Ruizhen Hu, Paul Guerrero, Niloy Mitra, and Taku Komura. 2016. Relationship Templates for Creating Scene Variations. In *ACM SIGGRAPH Asia*.

Xi Zhao, He Wang, and Taku Komura. 2014. Indexing 3D Scenes Using the Interaction Bisector Surface. In *ACM TOG*.

Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Kosta Derpanis, and Kostas Daniilidis. 2016. Sparseness Meets Deepness: 3D Human Pose Estimation from Monocular Video. In *IEEE CVPR*.

A KEYPOINT CONFIDENCE USING LCR-NET++

When using LCR-Net++, we estimate confidence of the 2D detection keypoints as

$$v_k = \underset{q_k^i \in pose\ proposals}{var} \left(q_k^i\right) / \left(1 + \exp\left(-0.2s' + 3.5\right)\right)$$

$$c_k(v_k) = 1/[1 + \exp\{-10 \exp\left(\log(v_k)/P_{99}\left(\log(v_k)\right)\right) + 20\}]$$

where, var_k denotes the variance of the 3D joint position among the grouped pose proposals, and P_{99} denotes the 99th percentile of *log* joint variances over the whole recording, assigning high confidence to low variance joint estimates, and s' is a per-pose score defined in Equation 6 in [Rogez et al. 2019].