Accepted by Computer Vision and Image Understanding , 2014

Tracking object poses in the context of robust body pose estimates ☆

John Darby *, Baihua Li, Nicholas Costen

School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester M1 5GD, UK

A B S T R A C T

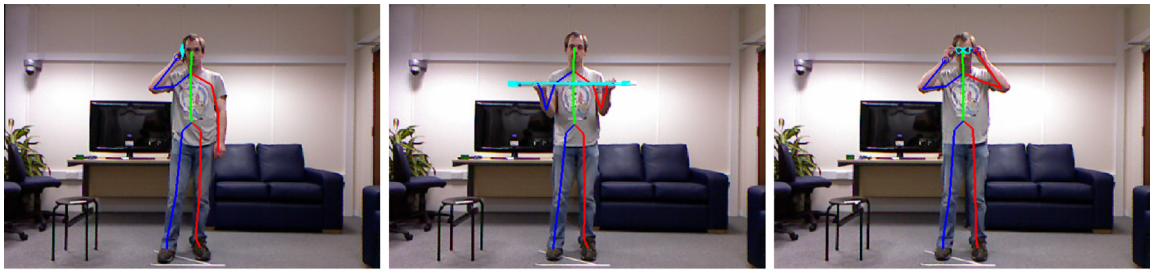
This work focuses on tracking objects being used by humans. These objects are often small, fast moving and heavily occluded by the user. Attempting to recover their 3D position and orientation over time is a challenging research problem. To make progress we appeal to the fact that these objects are often used in a consistent way. The **body poses** of different people using the same object tend to have similarities, and, when considered relative to those body poses, so do the respective **object poses**. Our intuition is that, in the context of recent advances in body-pose tracking from RGB-D data, robust object-pose tracking during human-object interactions should also be possible. We propose a combined generative and dis-criminative tracking framework able to follow gradual changes in object-pose over time but also able to re-initialise object-pose upon recognising distinctive body-poses. The framework is able to predict object-pose relative to a set of independent coordinate systems, each one centred upon a different part of the body. We conduct a quantitative investigation into which body parts serve as the best predictors of object-pose over the course of different interactions. We find that while object-translation should be predicted from nearby body parts, object-rotation can be more robustly predicted by using a much wider range of body parts. Our main contribution is to provide the first object-tracking system able to estimate 3D translation and orientation from RGB-D observations of human-object interactions. By tracking precise changes in object-pose, our method opens up the possibility of more detailed computational reason-ing about human-object interactions and their outcomes. For example, in assistive living systems that go beyond just recognising the actions and objects involved in everyday tasks such as sweeping or drinking, to reasoning that a person has "missed sweeping under the chair" or "not drunk enough water today".

1. Introduction

This paper attempts to track the 3D pose of objects being used by humans. Although similar efforts have been made in the past, they have usually focused on the localisation of objects using bounding boxes (e.g. [1,2]). Here we try to recover the full 3D translation and orientation of objects over time, our ultimate goal being to automate deeper reasoning about human-object interactions and their outcomes. Tracking 3D object pose is challenging because the objects are often small relative to the person using them, fast-moving, and heavily occluded. However, we observe that humans are often able to estimate both the object class and 3D pose from the corresponding body pose alone, see for example Fig. 1a showing a person using a fully occluded mobile phone. From this observation we deduce that both the *body-poses* and relative *object-poses* seen in many human-object interactions feature reasonably high levels of consistency. Therefore, if body-poses can be estimated robustly our intuition is that object-poses can be usefully predicted from them. We examine this claim in this work.

If good body-pose estimates are available then, as a simple first step, we might try to locate the corresponding object-pose based on the position of the participant's hand. For example, this presumption is used as a first step in object localisation by [1,3,4]. However, where interactions are with larger objects (e.g. brooms) or are more complex (e.g. two-handed: Fig. 1b; or involving the transfer of objects between body parts: Fig. 1c) we believe that a more sophisticated framework is necessary. In particular, while the position of the dominant hand may sometimes serve as a good predictor for object translation, we anticipate that it may not always be the best (or the only good) predictor of object orientation.

In order to test our ideas we set about the task of learning the 3D spatial and rotational relationships between body parts and objects during human-object interactions. This is in contrast to previous studies which have learned 2D spatial relationships between human-object centroids [2,5,6] or part-object centroids [7]. Additionally, where other work has attempted to learn



(a) (b) (c)

Fig. 1. Human-object interactions with body-poses and object-poses superimposed: (a) talking on a mobile phone; (b) lifting weights; and (c) putting on glasses.

aggregated models across the duration of the interaction (e.g. "hats are always on top of heads" [5]), a key aim of our approach is to determine *when* different body parts offer good predictions of object-pose. For example, the hands while picking up and putting on a hat, but the head once wearing the hat.

In order to study the relationship between body and object, we learn a large number of body-pose → object-pose mappings from labelled training data. We have found that, in general, this mapping does not remain one-to-one during human-object interactions. That is, there are times during the interaction where nearby poses in body space map to very different poses in object space. This means that the problem is not suited to a pure *discriminative* approach where we infer each object-pose solely from the current body-pose. On the other hand, a pure *generative* approach where we gradually "update and test" the pose of the object between consecutive images is also likely to fail due to a lack of good image features (the object is often small, motion blurred and partially occluded).

Instead, we propose that the problem of object tracking during human-object interactions is best addressed in a *combined* generative + discriminative (G + D) tracking framework (e.g. [8–10]). The idea is that the body → object pose mappings can be used in a discriminative strand, able to initialise tracking and to re-initialise at points in the interaction where the mapping is (near) one-to-one. However, for periods where the mapping is multivalued, we can rely on a second generative strand to gradually update the object-pose between frames and test against available image evidence. We bring about this combination using a particle-based Bayesian approach that extends our earlier work [11] on the importance sampling framework [12].

Our wider research goal is to automate deeper reasoning about human-object interactions by computer vision systems. Previous works have studied human-object interactions in order to improve reasoning about objects (e.g. "is that a jug?" [1]), about human actions (e.g. "is he pouring water from the jug?" [6]) and about scenes (e.g. "where did he leave the jug?" [13]). By pursuing a more detailed description of object-pose during interactions we hope to pave the way for deeper reasoning about outcomes, such as "did he pour all of the water out of the jug?", "has the floor by the table been thoroughly swept?", or "what is she taking a photograph of?". A critical ingredient in this type of reasoning is an accurate 3D description of the changing object-pose over time, and this is the specific aim of this work. Future applications could include vision-based assisted living systems for the elderly, able to reason about the upkeep of the home (e.g., cleanliness, consumption of foodstuffs) by observing human-object interactions in detail.

We make the following contributions:

- We present an approach that gives full 3D estimates of object-pose (translation and orientation) during human-object interactions. To the best of our knowledge this is the first paper to provide this level of detail from a single sensor ([14] do so with multiple, synchronised video cameras and static backgrounds).
- The approach is able to automatically initialise itself at the first frame, track subsequent object-pose changes with a generative particle set, and perform "soft" re-initialisations through the introduction of discriminative particles in variable numbers.
- These new predictions about object-pose can be made relative to any part of the body and we introduce methods for selecting the best parts for predicting: (i) translation; and (ii) rotation, given the current point in the interaction.

By using a large dataset of labelled human-object interactions we are able to demonstrate quantitatively the value of the approach over the use of the hand (e.g. [1]) or randomly chosen body parts (e.g. [11]) for prediction. We also show the importance of the combined G + D scheme over a purely generative approach.

2. Related work

Human-object interactions have been studied in a number of different contexts. By far the most popular has been that of human action recognition, where a number of works [2,5,15,16] have combined a study of object-pose with an already well developed literature on human pose estimation in order to improve action recognition rates. However, other authors have also studied human-object interactions in order to improve human pose estimation [14,17], object detection and tracking [1,3,13,18,4], or even both [7]. In this paper we are specifically interested in the accurate tracking of object-pose. However, we anticipate this is best served by learning about the relationship between body-pose and object-pose. This is something that all the works above have addressed, to some degree, and we review the various contributions below, dividing work between the various sensor modalities studied.

RGB images: Gupta et al. [6] and Prest et al. [5] learn 2D spatial relationships between human-centred and object-centred bounding boxes. But these models do not vary with changes in body-pose, e.g. a bike is always below the rider, a tennis racquet above the server's head. Yao and Fei-Fei [7] learn 2D spatial relationships between individual body parts and objects using a discretised search space around the centre of each body part, similar to the spatial histograms of [16]. By learning separate distributions for different body-pose clusters ("atomic poses"), they avoid aggregating these relationships over time. Desai et al. [16] also report the importance of learning body-pose specific, or "pose-aware" spatial histograms for good action recognition rates. These single-image approaches are impressive, but do require unoccluded, short-exposure images of human-object interactions that allow the application of state-of-the-art object detectors (e.g. [19]). They are also limited to providing 2D bounding boxes around objects.

RGB video: Detecting and tracking objects through human-object interactions in RGB video is challenging. Insights are provided by the confusion between object detectors in [6] and the poor performance of adaptive trackers (e.g. [20–22]) evaluated in [11]. Gupta et al. [6] are able to improve object detections by

considering the movement of the participant's hands, but they do not present object tracking results. In later work Prest et al. [2] present an approach that tracks and merges between individual object detections across a whole video sequence. Again, they use these tracks to capture a 2D spatial relationship between human-object centroids over time, improving action recognition rates. What they present is a batch approach, perhaps not suited to a pure tracking context, but is able to give impressive object tracking (2D bounding boxes) on the dataset from [6]. One outstanding question (also for similar approaches to RGB-D video [4]) is whether the approach relies on being able to track hands in between good object detections (and merges).

RGB-D video: Depth data has made possible the tracking of 3D object-translation (3D bounding boxes) during human-object interactions. In part this has been because RGB-D data allows for accurate body-pose estimation via the use of various "black box" frameworks, e.g. [8,23], Kinect [24], and OpenNI [25]. To date, however, spatial models of the body-object interaction have tended to be much simpler than those used in RGB images; usually assuming that objects are near hands [1,3,4]. Other approaches have chosen not to model the body-object relationship at all (for the purpose of object tracking), instead relying on the visibility of objects in the depth map [13,18], something which is difficult to guarantee.

Gall et al. [1] limit their search for objects to within 25 cm of the participant's most active hand. They mask out limbs and extract the object by finding connected components in the depth image. Object classification is subsequently achieved via action classification of body-poses. Similarly, Kjellström et al. [3] perform object detections only within the neighbourhood of the participant's hands, which are tracked in 3D using stereo observations. They are able to improve subsequent object classifications using context from the movement of the participant's hand, and the pose of their individual fingers [26]. Koppula et al. [4] also restrict object detection to the vicinity of the hands, as tracked by the OpenNI framework [25]. They then track between detections using a particle filtering approach and merge tracks in a batch framework reminiscent of [2], their ultimate goal being improved action classification.

If an RGB-D sensor can be used to collect images of objects from a near distance and with minimal occlusion, then tracking 3D object-pose (translation and orientation) though large rotations is possible [27,28]. For human-object interactions such a viewpoint is usually impossible and occlusions of the object (by the human) are often severe. As an object is moved further away from an RGB-D sensor such as Kinect there is no guarantee it will continue to reflect the structured light source, and it may disappear from the depth map (see Section 7 for further discussion). Both [13,18] rely on objects remaining visible in the depth map during interactions. Packer et al. [13] additionally assume a static background behind objects so that the depth map can be background subtracted, and Pieropan et al. [18] assume objects start on a flat surface from which they can be detected by variations in colour and/or depth. Again, both approaches only model spatial relationships between objects and hands in their subsequent work on improving object and action recognition. None of these approaches are able to track object orientation, only 3D bounding boxes.

RGB-D sensors offer the richest data available from a single viewpoint, but many of the best models of the spatial relationships between body and object have come from the literature on single RGB images. Our aim is to combine the 3D body-pose estimates that RGB-D sensors can generate with a richer model of the spatial relationships between body parts and objects to achieve 3D object tracking (translation and orientation) without the need for multiple synchronised sensors (e.g. [14,17]). A preliminary version of this paper was described in [11]. The work described here differs in the following ways: (1) we separate the translation and rotation components of object-pose between their own, independent, part predictors; (2) rather than assigning these components to random part predictors [11], we introduce measures to automatically determine the best choice at any particular instant; (3) we present a "soft" approach to re-initialisation, dynamically adjusting the mixing fraction of generative and discriminative particles; (4) we present a 3D (rather than 2D [11]) error evaluation on a larger number of participants which allows us to demonstrate robustly the efficacy of (1–3) in reducing object-pose tracking errors.

3. Overview of the method

We address the following problem: a participant performs a human-object interaction (e.g. making a phone call) and is recorded using an RGB-D sensor, we wish to track the 3D pose of the object – or *object-pose* – over time. The use of an RGB-D sensor means that it is possible to extract robust estimates of the participant's joint locations – or *body-pose* – at each instant (e.g. using [8,23–25]). We therefore formulate our object-pose tracking problem relative to the sequence of estimated body-poses. A visual summary of the following two sections is given in Fig. 2.

3.1. Training

Our training phase centres around a collection of 3D {body, object} pose pairs for participants performing different object-interactions. We discuss the creation of the specific corpus of training data used in this paper in Section 6, but given any training sequence of this form, the following steps are general:

TR1: *For all training data pairs:* compute location insensitive encodings of body-pose, relative to a coordinate system centred on the pelvis (Section 4.1); compute local coordinate systems, or *part predictors*, for each body part in the current body-pose by performing translation and rotation operations on the pelvis coordinate system; compute object-poses relative to each part predictor (Section 4.2). These relative object-pose configurations are sampled from to generate hypotheses during tracking.
TR2: *For every body-pose:* find associated cluster of nearest neighbours in body-space; compute the variation in **object-translation** across nearest neighbours, relative to every part predictor; compute variation in **object-rotation** across nearest neighbours, relative to every part predictor; compute the median score across all part predictors for each measure (Section 4.3). These scores are used to determine the suitability of a particular body-pose for re-initialisation of the object-translation and/or object-rotation.
TR3: *For every body-pose:* find associated cluster of nearest neighbours in body-space. Across all poses in a given cluster and relative to each part predictor: compute the average proximity of the object, and the average *changes* in object-translation and object-rotation between the current and the next training pair (Section 4.4). These values are used to determine which part predictors are suitable for use when re-initialising the object-pose.

3.2. Testing

Given a new test RGB-D video showing a human-object interaction:

TE1: Estimate participant's body-pose and use it to compute a location insensitive encoding relative to a coordinate system centred on the pelvis. Find the closest body-pose in the training data and use the set of object-poses associated with its cluster of nearest body-pose neighbours to initialise a full set of

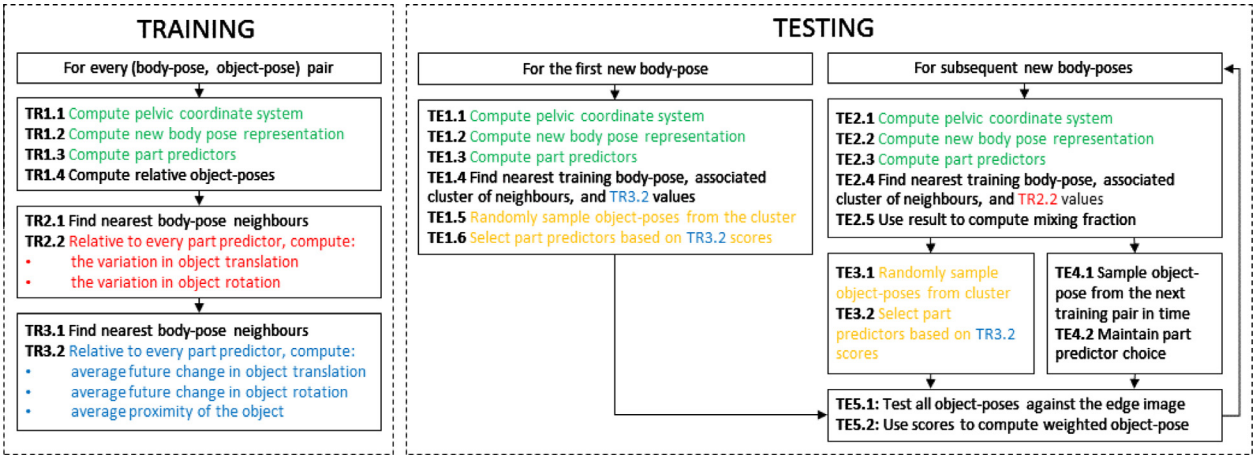


Fig. 2. Global workflow for the proposed system: key steps are colour coded to highlight repetition. More detail on each training and testing process is given in Section 3, and also shown diagrammatically in Fig. 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

object-poses, selecting suitable part predictors based on the values in TR3 (Section 5). Skip to **TE5**.

TE2: For propagation of particles to the subsequent frame: consider the *next* body-pose estimate in the test video. Compute the location insensitive encodings of body-pose, relative to a coordinate system centred on the pelvis. Find the closest body-pose in the training data. Based on the associated variation scores in TR2 for the associated cluster (suitability for re-initialisation), compute the mixing fraction (Section 5.5) between generative and discriminative particles.

TE3: Generate some fraction of discriminative re-initialisation particles (Section 5.4) by sampling from the object-poses associated with the cluster of nearest body-pose neighbours. Select suitable part predictors based on the values in TR3.

TE4: Propagate the remaining fraction of particles using a generative model of object-pose dynamics that moves through consecutive pose pairs in the training data (Section 5.3).

TE5: Compare all particles with the current RGB image using an edge-based observation likelihood model to compute weights (Section 5.6) and the tracking result (Section 6.2). Return to TE2 and repeat until the entire video has been processed.

4. Learning body-object mappings

During the training phase, our algorithm learns a mapping between body-poses and object-poses during human-object interactions. To do this we require 3D estimates of the locations of the joints in the human body as well as 3D estimates of the associated object-pose. Each observation from an RGB-D sensor $\mathbf{z}_t = [\mathbf{r}_t, \mathbf{d}_t]$ consists of a colour (RGB) image $\mathbf{r}_t$ and depth (D) image $\mathbf{d}_t$, and given these data there are a number of vision algorithms (e.g. [8,23–25]) able to produce robust 3D joint estimates, $\mathbf{b}_t$. Here we use the Kinect sensor [24] to record training interactions. We then manually label each observation with a 3D object-pose $\mathbf{o}_t$ in a post-processing step, using the depth and RGB images for guidance. A full discussion of our dataset follows in Section 6 but here we describe the general steps that must be applied to these 3D pose pairs, regardless of how they are acquired.

4.1. Body pose

At each time t, Kinect gives 3D location estimates for 20 different body joints, $\{\mathbf{j}_{t,i}\}_{i=1}^{20}$. For invariance to rotations and translations of the participant relative to the sensor we shift these coordinates into a new basis $H_t^* = \{\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t\}$ centred on the participant's hips. We define a vector $\mathbf{x}_t = \mathbf{j}_{t,\mathrm{LHip}} - \mathbf{j}_{t,\mathrm{RHip}}$ running between the hips, and $\mathbf{y}'_t = \mathbf{j}_{t,\mathrm{Spine}} - \mathbf{j}_{t,\mathrm{CHip}}$ running between the hip centre and spine. We then cross these vectors to get a perpendicular vector, $\mathbf{z}_t = \mathbf{x}_t \times \mathbf{y}'_t$ before finally replacing $\mathbf{y}'_t$ with $\mathbf{y}_t = \mathbf{z}_t \times \mathbf{x}_t$ to give an orthogonal basis $H_t^* = \{\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t\}$. A body-pose at a particular instant, $\mathbf{b}_t \in \mathbb{R}^{60}$, is then given by the concatenated locations of all 20 joints relative to H_t^* and scaled to unit height. The process is illustrated in Fig. 3a.

4.2. Relative object-poses

Given a 3D object-pose (translation and orientation) we then calculate object-poses relative to each one of the participant's body parts. To do this we generate a new basis for each one of the $l \in [1, \ldots, 19]$ parts of the kinematic tree, $\boldsymbol{\lambda}_{t,l} = \mathbf{j}_{t,j} - \mathbf{j}_{t,i}$, by translating the basis H_t^* to lie with its origin at the parent joint $\mathbf{j}_{t,i}$ and rotating it so the original $\mathbf{z}_t$-axis lies along the body part, pointing through the child joint $\mathbf{j}_{t,j}$. This is done by rotating H_t^* through a positive angle $\theta_{t,l}$ about a pivot vector $\mathbf{p}_{t,l} = \boldsymbol{\lambda}_{t,l} \times \mathbf{z}_t$, where $\theta_{t,l} = \cos^{-1}(\hat{\boldsymbol{\lambda}}_{t,l} \cdot \hat{\mathbf{z}}_t)$. The new basis $H_{t,l} = \{\mathbf{x}_{t,l}, \mathbf{y}_{t,l}, \mathbf{z}_{t,l}\}$ is referred to as the lth *part predictor*. The process is illustrated in Fig. 3b.

The object's pose relative to the lth part predictor is given by $\mathbf{o}_{t,l} = [\mathbf{a}_{t,l}, \mathbf{q}_{t,l}]$, where $\mathbf{a}_{t,l} \in \mathbb{R}^3$ is a translation (again scaled to unit height) and $\mathbf{q}_{t,l} \in \mathbb{R}^4$ is a quaternion rotation, both relative to $H_{t,l}$. At time t, we store the full set of 19 relative object-poses as the matrix $\mathbf{O}_t = [\mathbf{o}_{t,1}, \ldots, \mathbf{o}_{t,19}]$, and denote the pose for the lth part predictor at time t by $\mathbf{O}_{t,(:,l)}$ where $(:, l)$ denotes the lth column of the matrix and $(1:3, l)$ and $(4:7, l)$ give the translation and rotation, respectively.

For the ith participant performing a given object-interaction we have a collection of body-poses $\mathcal{B}_i = \{\mathbf{b}_1^i, \ldots, \mathbf{b}_N^i\}$ and associated object-poses $\mathcal{O}_i = \{\mathbf{O}_1^i, \ldots, \mathbf{O}_N^i\}$. From here on we reserve the use of the index $n \in [1, \ldots, N]$ for training data and use the index $t \in [1, \ldots, T]$ for new test data.

4.3. Characterising the body-object pose mapping

Generative object tracking during human-object interactions is difficult because of a lack of good image evidence. For this reason we wish to take any available opportunity to re-initialise object-poses. To this end we are interested in any situations where the relationship between body-poses and object-poses is unambiguous, or close to a one-to-one mapping. To examine the mappings for a particular action class, we find a set of nearest body-pose neighbours for *every* body-pose $\mathbf{b}_n^i \in \mathcal{B}_i$ by considering their Euclidean separations in body-pose space. The set $\mathcal{C}_n \subset [1, \ldots, N]$ holds the indices to the cluster of neighbours which are within the

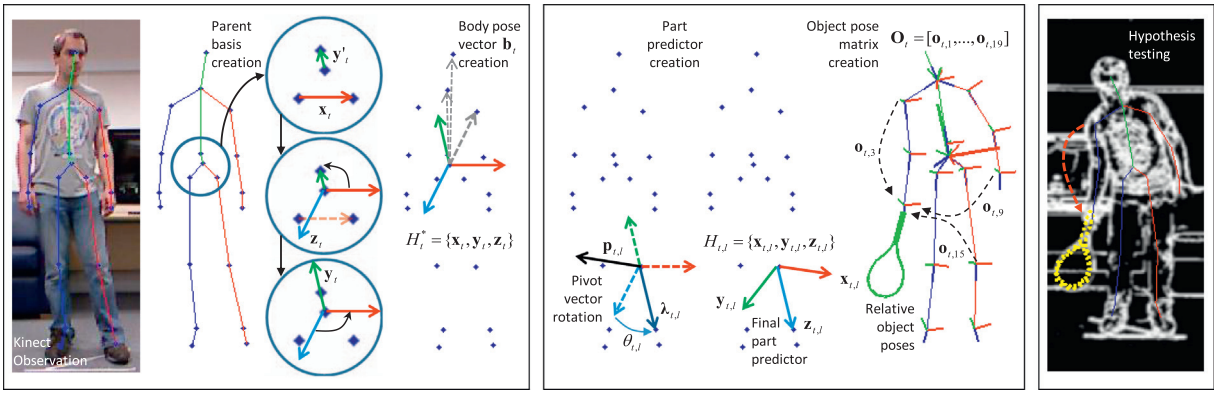


Fig. 3. Body-pose and object-pose: (a) a parent basis H_t^* is built from joints in the pelvis and used to construct a body-pose vector $\mathbf{b}_t$ from the relative locations of other joints (Section 4.1); (b) local bases are created for every body part by translating and rotating H_t^* and are used to learn a matrix of relative object-poses, $\mathbf{O}_t$ (Section 4.2); (c) during tracking these local bases, or *part predictors*, are used to produce object-pose hypotheses for evaluation (Section 5).

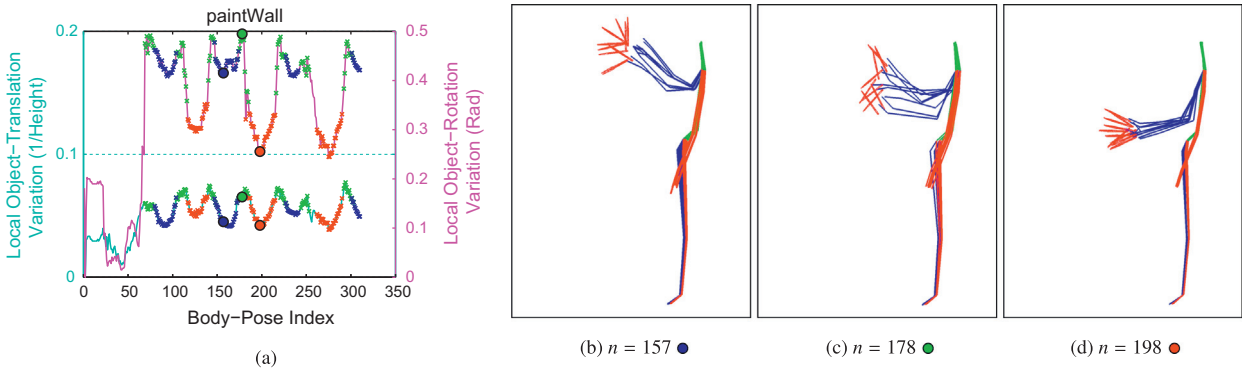


Fig. 4. Variation of object-pose across nearest body-pose neighbours (best viewed in colour): (a) median part predictor variation in object-rotation (magenta solid line) and object-translation (cyan solid line) for `paintWall`. The interaction is periodic with the participant repeatedly stroking a paintbrush up and then down a wall. Three body-poses have been highlighted by circular markers at $n = 157$ (blue, ●), $n = 178$ (green, ●) and $n = 198$ (red, ●). The nearest neighbours for each of the three body-poses are shown with crosses of the same colour. (b–d) ten random samples from each cluster are show with their associated object-poses. Notice that the blue and green clusters feature high variation in object-rotation compared to the red cluster. The blue cluster (b, ●) features the turn of the brush at the top of the participant's reach, ready to bring the opposite side in contact with the wall. The green cluster (c, ●) captures the brush moving both up *and down* the wall with the tip facing approximately 45° down and then 45° up, respectively. In contrast, the red cluster (d, ●) maps to a much tighter distribution of object-poses, with the brush held approximately level at the bottom of the stroke. At test time, body-poses that fall close to the red cluster offer a good opportunity to re-initialise the object-pose. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

distance $\delta^i_{\min}$ of the nth body-pose. The "closeness" threshold[1] for each sequence, $\delta^i_{\min}$, can be set manually, but in Section 6 we outline a simple strategy for its automatic selection.

For every body-pose in a given participant's interaction training data we then compute the spread in the associated object-translation and object-rotation values across its near neighbours in body-space. To do this we define the separation functions

$$\epsilon(\mathbf{a}_j, \mathbf{a}_k) = \sqrt{\sum_{i=1}^{3} \left(a^i_j - a^i_k\right)^2} \tag{1}$$

for the Euclidean distance between two vectors defining object-translations, and

$$\theta(\mathbf{q}_j, \mathbf{q}_k) = \arccos(|\,\mathbf{q}_j.\mathbf{q}_k\,|) = \arccos\left(\left|\sum_{i=1}^{4} q^i_j q^i_k\right|\right) \tag{2}$$

for the angle between two unit-length quaternions defining object-rotations [29]. Using these measures we then compute, for every body-pose in the training set $n \in [1, \ldots, N]$, the variation in object-translation and object-rotation relative to each part predictor

$$\phi^{\epsilon}_{n,l} = \sqrt{\frac{1}{|\,\mathcal{C}_n\,|} \sum_{c \in \mathcal{C}_n} \epsilon\left(\mathbf{a}_{c,l}, \hat{\mathbf{a}}_{n,l}\right)^2} \tag{3}$$

$$\phi^{\theta}_{n,l} = \sqrt{\frac{1}{|\,\mathcal{C}_n\,|} \sum_{c \in \mathcal{C}_n} \theta\left(\mathbf{q}_{c,l}, \hat{\mathbf{q}}_{n,l}\right)^2} \tag{4}$$

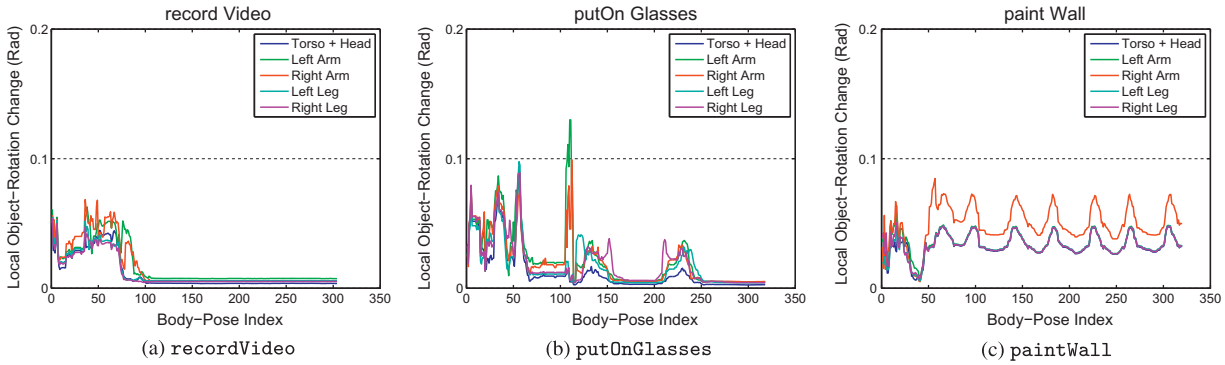
where $\hat{\mathbf{a}}_{n,l} = \frac{1}{|\mathcal{C}_n|} \sum_{c \in \mathcal{C}_n} \mathbf{a}_{c,l}$ is the average object-translation across this set and $\hat{\mathbf{q}}_{n,l} = \frac{1}{|\mathcal{C}_n|} \sum_{c \in \mathcal{C}_n} \mathbf{q}_{c,l}$, re-normalised to lie on the unit sphere [30], is the average object-rotation. We use the median value of these variations across part predictors

$$\phi^{\epsilon}_{n} = \text{median}\left(\{\phi^{\epsilon}_{n,l}\}^{19}_{l=1}\right) \tag{5}$$

$$\phi^{\theta}_{n} = \text{median}\left(\{\phi^{\theta}_{n,l}\}^{19}_{l=1}\right) \tag{6}$$

as a robust measure of the nth body-pose's suitability for re-initialisation. The lower the score, the more suitable the current body-pose is for attempting re-initialisation of the object-pose. The median

[1] Some of the human-object interactions we study are periodic and so nearby body-poses are not always nearby in terms of their training index $n \in [1, \ldots, N]$.



(a) `recordVideo` (b) `putOnGlasses` (c) `paintWall`

Fig. 5. Object-rotation: average object-rotation changes across all nearest body-pose neighbours for all part predictors. Plots are from representative participants performing three different human-object interactions: `recordVideo` (a), `putOnGlasses` (b) and `paintWall` (c). As visualising all 19 part predictors at once is difficult we have grouped them across limbs (torso + head, left arm, right arm, left leg, right leg) and plotted the stablest predictor (lowest delta score) from each limb at every instant. The effect is to highlight the stablest part predictor *per limb* over time. Notice that hands are not always the best (or only good) part predictors: once the camera is held steady (frame 100, a) part predictors right across the body stabilise; once the glasses are placed on the face (frame 105, b) the head becomes more stable than the arms during subsequent movement; while painting a wall (c) the arm that "moves with" the object is actually the least stable predictor. See text for more details.

value is an appropriate measure (rather than, say, the minimum) because all body-poses within a cluster are similar and so the spread in object-poses tends to be similar relative to every part predictor. Fig. 4 shows the spread of object-translation and object-rotation parameters across nearest body-pose neighbours for all training poses $n \in [1,\ldots,N]$ of a `paintWall` human-object interaction performed by a representative participant. Notice in Fig. 4a that while the spread in object-translations remains relatively low across the interaction (rising slightly during the fastest parts of the brushstroke), the spread of object-rotations changes quite considerably. Fig. 4b–d show body-pose clusters from the top, middle and bottom of the brushstroke along with their associated object-poses. The spread in object-rotations at the bottom of the stroke is considerably lower, meaning it provides a better opportunity to re-initialise the object-pose.

4.4. Part predictor choice

If a body-pose is found to be suitable for object-pose re-initialisation we must then decide which of the 19 part predictors to use for prediction. We use the following logic in our choice: (i) we start from the principle that the instantaneous relative object-pose predictions of all part predictors are equally valid; (ii) however, some part predictors feature much greater *changes* in relative object-pose over time than others, meaning any subsequent generative tracking strand must explore a greater range of relative object-pose configurations; (iii) we therefore favour part predictors that display smaller relative changes in object-pose over time, ensuring that, as far as is possible, changes in object-pose are brought about naturally through the changes in a participant's body-pose; (iv) finally, we also allow that the best part predictor for object-translation prediction is not necessarily also the best part predictor for object-rotation prediction. We refer to part predictors that minimise future changes in the relative object-pose as being *stable*.

To identify stable rotation predictors, we consider the changes in object-rotation between the current and next time steps, n and $n+1$ in the training data. We calculate an average value for this change across the cluster of body-pose neighbours $\mathcal{C}_n$ for each training pose $n \in [1,\ldots,N]$

$$\psi^{\theta}_{n,l} = \frac{1}{|\,\mathcal{C}_n\,|}\sum_{c\in\mathcal{C}_n}\theta(\mathbf{q}_{c,l},\mathbf{q}_{c+1,l}) \tag{7}$$

Fig. 5 shows the average rotation delta value across nearest body-pose neighbours for all part predictors and all training poses $n \in [1,\ldots,N]$ of three different interactions: `recordVideo`, `putOnGlasses` and `paintWall`. As visualising all 19 part predictors at once is difficult we have grouped them across limbs (torso + head, left arm, right arm, left leg, right leg) and plotted the stablest predictor (lowest delta score) from each limb at every instant. The effect is to highlight the stablest part predictor *per limb* over time. Note that it is rotations of the object relative to the body part that is key. In `recordVideo` (Fig. 5a), the camera is moved into position just above and in front of the participant's head at around frame 100, after which point the participant remains still and all part predictors become stable, even if they are far away from the object. In `putOnGlasses` (Fig. 5b), the glasses are placed on the face at around frame 110 and the arms become particularly unstable as they drop back down to the participant's sides. The participant then takes a large step to their left (frame 125), pauses, and then back to their right (frame 200). During this time the torso and head remain the most stable predictors, while all other limbs rotate relative to the glasses, which remain stationary on the participant's head. In `paintWall` (Fig. 5c) participants tend to hold the paintbrush at a relatively constant angle as they make brushstrokes down (and then up) the wall. Object-rotation is therefore minimised relative to their stationary limbs (torso + head, left arm, legs) rather than their moving arm.

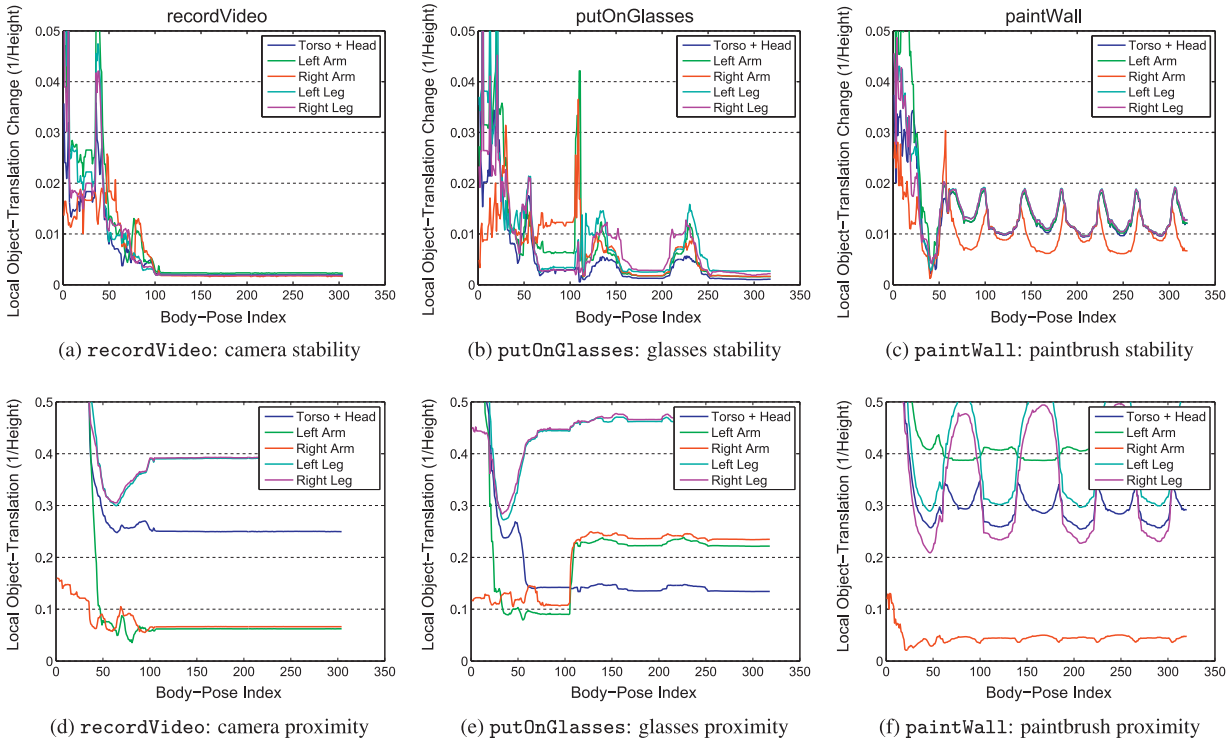
To identify stable translation predictors we can calculate equivalent delta scores for object-translation

$$\psi^{\epsilon}_{n,l} = \frac{1}{|\,\mathcal{C}_n\,|}\sum_{c\in\mathcal{C}_n}\epsilon(\mathbf{a}_{c,l},\mathbf{a}_{c+1,l}) \tag{8}$$

However, we note that much other work has had success in predicting object locations from the hand (see also Section 2). Our intuition is that this is because, for many interactions, the hand is the *closest* body part to the object. For the interactions we study this is not always the case, and so as an alternative to measuring stability, we also consider the *proximity* of the object to *every* part predictor

$$\psi'^{\epsilon}_{n,l} = \frac{1}{|\,\mathcal{C}_n\,|}\sum_{c\in\mathcal{C}_n}\epsilon(\mathbf{a}_{c,l},\mathbf{0}) \tag{9}$$

Fig. 6 shows a comparison between the average translation delta value (top row) and proximity value (bottom row) across nearest body-pose neighbours for all limbs and all training poses $n \in [1,\ldots,N]$ of the interactions: `recordVideo`, `putOnGlasses` and `paintWall`. In terms of stability, the initial picture is similar to that with object-rotation: for `recordVideo` (Fig. 6a) all part predictors become stable once the camera is in position; for `putOnGlasses` (Fig. 6b) the torso and head are most stable once



(a) `recordVideo`: camera stability (b) `putOnGlasses`: glasses stability (c) `paintWall`: paintbrush stability

(d) `recordVideo`: camera proximity (e) `putOnGlasses`: glasses proximity (f) `paintWall`: paintbrush proximity

Fig. 6. Object-translation: (top row) average object-translation changes across all nearest body-pose neighbours for all part predictors; (bottom row) average object proximity across all nearest body-pose neighbours for all part predictors. Plots are from representative participants performing three different human-object interactions: `recordVideo` (a, d), `putOnGlasses` (b, e) and `paintWall` (c, f). As visualising all 19 part predictors at once is difficult we have grouped them across limbs (torso + head, left arm, right arm, left leg, right leg) and plotted the stablest/closest predictor from each limb at every instant. The effect is to highlight the stablest (top row) or closest (bottom row) part predictor *per limb* over time. Part predictors can be far away from an object but still be stable (`recordVideo`, a); the hands can become unstable once an object moves to another body part (`putOnGlasses`, b); unlike for object-rotation, the body part that "moves with" the object does tend to be more stable (`paintWall`, c). For some smaller objects, the participant's strongest hand remains closest to the object at all times by some way (`paintWall`, f); however this is not true where objects can transfer between body parts (`putOnGlasses`, e); or where the object is held in both hands (`recordVideo`, d).

the glasses are placed on the face. However, for `paintWall` (Fig. 6c), the right arm is now the *most* stable predictor, minimising changes in relative object-translation as it moves with the paintbrush. We can see this relationship in the proximity plot: the right hand is at all times the closest body part to the paintbrush (Fig. 6f), and this is fairly typical for many small objects. However, for objects that are held two-handed, either hand remains equally proximate (e.g. `recordVideo`, Fig. 6d) or where objects can be moved between limbs, the hands can become quite distant (e.g. `putOnGlasses`, Fig. 6e).

5. Object tracking

During tracking we try to recover a new object-pose estimate $\hat{\mathbf{o}}_t$ given each new body-pose estimate $\hat{\mathbf{b}}_t$ and its associated RGB image $\hat{\mathbf{r}}_t$. We use a particle-based Bayesian approach to combine generative and discriminative object-pose hypotheses. In earlier work [11] we have used the importance sampling framework [12] to bring about this combination. Importance sampling allows for the combination of discriminative particles based on the *current* observation with generative particles propagated from the posterior approximation at the *previous* timestep. New discriminative particles are reweighted based on the likelihood of them having occurred given the location of the last posterior approximation (particle set) and the *dynamical model* used for particle propagation.

However, we have found that the use of this corrective term leads to a dilemma. For a successful generative strand we hope to adopt a dynamical model that is as restrictive as possible but not more so. That is, a model that spreads particles over a small enough subset of the object-pose space that good coverage can be achieved with sensible particle numbers, but that we can also be confident will envelope the next solution. Such a model will not tolerate large jumps across the pose space by particles. Therefore, in the very scenario where discriminative particles are most useful – when they regain a track after the particle set has drifted – they will be subject to severe reweightings in the importance sampling framework.

The original importance sampling formulation gets round this problem with a third flavour of *initialisation* particle: a discriminative particle that is not reweighted. However, the interplay between these two types of discriminative particle then becomes difficult to interpret, and choosing the constant-valued mixing fractions for each particle type a challenging and experimental process. In this work we do not reweight our discriminative particles but instead concentrate on dynamically adjusting the mixing fraction based on the suitability of the current observation for re-initialisation (rather than introducing a constant, arbitrary fraction at every timestep).

5.1. Particle filtering

Particle filtering facilitates a generative approach to object tracking by maintaining an approximation to the posterior $p(\mathbf{o}_t \mid \mathcal{Z}_t)$, where $\mathcal{Z}_t = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$ is the set of all observations, with set of P particles, $\{(\mathbf{o}_t^{(p)}, \pi_t^{(p)})\}_{p=1}^{P}$. The pth particle consists of an object-pose estimate, $\mathbf{o}_t^{(p)}$ and associated weighting $\pi_t^{(p)} \approx p(\mathbf{z}_t \mid \mathbf{o}_t^{(p)})$ based on agreement with the observation $\mathbf{z}_t$. Particles are dispersed by a dynamical model $p(\mathbf{o}_t \mid \mathbf{o}_{t-1})$ between

observations. The task of tracking objects through human-object interactions is sufficiently challenging (due to their speed, small size and regular occlusion) that we do not attempt to recover an unconstrained object-pose $\mathbf{o}_{t,l} \in \mathbb{R}^7$, but instead spread particles through our corpus of training data. Such a generative approach, similar to that in [31], prevents us from arriving at impossible object-poses and might be described as being at the discriminative end of the spectrum of generative tracking approaches.

5.2. Particle structure

For a given human-object interaction we have a collection of body-poses $\mathcal{B}_i = \{\mathbf{b}_1^i, \ldots, \mathbf{b}_N^i\}$ and associated object-poses $\mathcal{O}_i = \{\mathbf{O}_1^i, \ldots, \mathbf{O}_N^i\}$ for each of a number of training participants, indexed by $i \in [1, 2, \ldots, S]$. Particles define an object-pose via particular indices into this collection of training data. Our chosen particle structure allows us to parameterise object-rotation and translation completely independently, taking advantage of the independent treatment presented in Section 4.3 (see also Section 5.4). Each particle holds: an index to the current training participant being used for object-translation prediction $i_{T,t} \in [1, 2, \ldots, S]$, and the participant being used for object-rotation prediction $i_{R,t} \in [1, 2, \ldots, S]$; an index to the current part predictor being used for object-translation prediction $l_{T,t}^{(p)} \in [1, 2, \ldots, 19]$, and the part predictor being used for object-rotation prediction $l_{R,t}^{(p)} \in [1, 2, \ldots, 19]$; and an index to the actual pose pairing used for object-translation $n_{T,t}^{(p)} \in [1, \ldots, N]$, and the pairing used for object-rotation $n_{R,t}^{(p)} \in [1, \ldots, N]$. The new particle structure is given by $(\breve{\mathbf{o}}_t^{(p)}, \pi_t^{(p)})$, where $\breve{\mathbf{o}}_t^{(p)} = (i_{T,t}^{(p)}, i_{R,t}^{(p)}, l_{T,t}^{(p)}, l_{R,t}^{(p)}, n_{T,t}^{(p)}, n_{R,t}^{(p)})$.

5.3. Generative particles

Generative particles are sampled from the particle set at $(t-1)$ (initialisation is discussed in the following section) and undergo simple propagation through the training data. The most obvious choice of dynamical model is to increment the pth particle's indices to the training data by one, e.g. $n_{T,t}^{(p)} = n_{T,t-1}^{(p)} + 1$ and similarly for $n_{R,t}^{(p)}$. However, we instead choose a noisy and slightly inflated dynamical model that enables generative particles to support a simple form of dynamic time warping; moving through the training data at a variable speed. The pth particle's index into the training data is updated as

$$n_{T,t}^{(p)} \sim \text{round}\left(|N(n_{T,t-1}^{(p)}, \sigma_T^2)|\right) \tag{10}$$

$$n_{R,t}^{(p)} \sim \text{round}\left(|N(n_{R,t-1}^{(p)}, \sigma_R^2)|\right) \tag{11}$$

where σ_T and σ_R are chosen empirically. All other elements of $\breve{\mathbf{o}}_t^{(p)}$ (indices for participants and part predictors) remain constant. As a first step in propagation between frames we update *all* our particles using this generative model. In the next two sections we describe the discriminative update of translation and/or rotation for some *fraction* of the particles in this new distribution.

5.4. Discriminative particles

There are moments during human-object interactions when the object-pose can be accurately inferred from the body-pose alone; that is, ϕ_n^ϵ and/or ϕ_n^θ for the closest body-pose in the training data are low, see Fig. 4. In order to exploit this fact we introduce discriminative particles, to complement the generative particles described in the previous section. Following [12], discriminative particles are sampled from an *importance function* conditioned on the current observation, $g(\breve{\mathbf{o}}_t \mid \mathbf{z}_t)$. To generate the pth discriminative particle from a new RGB-D observation $\mathbf{z}_t$ of a given interaction, we take the following steps:

D1: extract an estimate of the participant's new body-pose $\hat{\mathbf{b}}_t$ from the new observation (Section 4.1) and select a new participant index i_t by computing the nearest body-pose across all our training participant data $\mathcal{B}_i \forall i \in [1, \ldots, S]$, retaining the index n_t of the closest pose.

D2: set $i_{R,t}^{(p)} = i_t$; randomly select a new index for the **object-rotation** $n_{R,t}^{(p)}$ from the set of nearest neighbours (indexed by $\mathcal{C}_{n_t}$) to the winning body-pose (those which are within the distance $\delta_{\min}^{i_t}$ of $\mathbf{b}_{n_t}^{i_t}$); select a new rotation part predictor $l_{R,t}^{(p)}$ as that which minimises Eq. (7) for all values of $l \in [1, \ldots, 19]$.

D3: set $i_{T,t}^{(p)} = i_t$; randomly select a new index for the **object-translation** $n_{T,t}^{(p)}$ from the set of nearest neighbours (indexed by $\mathcal{C}_{n_t}$) to the winning body-pose; select a new translation part predictor $l_{T,t}^{(p)}$ as that which minimises Eq. (8) (or alternatively, Eq. (9)) for all values of $l \in [1, \ldots, 19]$.

In the original formulation of importance sampling, the new sample is reweighted based on its likelihood given the previous particle set and the generative dynamical model. Here we do not apply the reweighting factor (for those interested in the original formulation, our approach can be viewed as [12] with $r = 0$), but instead focus on appropriately varying the mixing fractions of generative and discriminative particles (which are held constant in [12]) to reflect the suitability of the current observation for re-initialisation. We also use discriminative particles to initialise our tracker. At the first frame the entire particle set is created as described above, based only on the first body-pose estimate.

5.5. Mixing fractions

In Section 4.3 we have defined measures of the level of variation in the body-object pose mapping for both translational and rotational components. When one or both of the values $\phi_{n_t}^\epsilon$ or $\phi_{n_t}^\theta$ for the closest training pose n_t are small, we wish to take the opportunity to re-initialise object-poses based on the current observation. To this end we compute two probability values, α and β, which determine the likelihood that we perform the discriminative update steps for rotation (**D2**) and translation (**D3**), respectively, for each particle in the set.

We compute these two values from zero-centred univariate Gaussian distributions

$$\alpha_t = p(\phi_{n_t}^\theta \mid 0, \sigma_\theta^2) \tag{12}$$

$$\beta_t = p(\phi_{n_t}^\epsilon \mid 0, \sigma_\epsilon^2) \tag{13}$$

which are normalised to give $p(0 \mid 0, \sigma_\theta^2) = p(0 \mid 0, \sigma_\epsilon^2) = 1$, and where σ_θ^2 and σ_ϵ^2 are determined empirically. The effect is to selectively update *either or both* of a particle's object-translation and object-rotation components depending on what the dataset tells us about the new body-pose and the nature of its mapping to the object-pose space.

5.6. Measurement density

Through the use of generative and discriminative particles we can create a set of hypotheses about the current object-pose $\{\breve{\mathbf{o}}_t^{(p)}\}_{p=1}^P$ based on both the last posterior $p(\breve{\mathbf{o}}_{t-1} \mid \mathcal{Z}_{t-1})$ and current observation $\mathbf{z}_t$, respectively. The final step is to test each of these hypotheses against the current observation $\mathbf{z}_t$ in order to calculate a set of associated likelihood weightings $\{\pi_t^{(p)}\}_{p=1}^P$ for each particle.

Table 1
Object-pose tracking accuracy in centimetres: each value is the mean and standard deviation across sequences for 6 different participants. (Lowest error score highlighted in bold.)

	G (hand)	G + D (hand)	G + D (random)	G + D (stable)	G + D (stable-proximate)
pourCream	12.2 ± 2.4	9.67 ± 2.4	11.6 ± 4.5	10.5 ± 4.4	$\mathbf{8.63 \pm 2.5}$
answerPhone	9.16 ± 0.94	6.59 ± 0.64	8.64 ± 1.2	7.70 ± 1.8	$\mathbf{6.49 \pm 0.60}$
drinkFromMug	18.7 ± 4.3	13.0 ± 3.6	$\mathbf{9.42 \pm 2.4}$	9.73 ± 3.8	11.9 ± 3.0
recordVideo	9.78 ± 1.8	7.15 ± 1.2	11.2 ± 3.6	9.89 ± 3.6	$\mathbf{6.02 \pm 1.7}$
liftWeights	25.9 ± 5.0	20.7 ± 3.2	10.5 ± 1.8	$\mathbf{10.3 \pm 2.3}$	11.5 ± 1.3
playFlute	15.4 ± 5.2	13.7 ± 6.3	$\mathbf{9.38 \pm 1.4}$	10.7 ± 2.4	11.9 ± 4.9
hammerNail	18.2 ± 2.5	$\mathbf{12.2 \pm 2.8}$	20.2 ± 6.1	14.1 ± 5.6	12.3 ± 2.8
putOnGlasses	21.6 ± 7.9	13.5 ± 2.7	9.32 ± 3.0	$\mathbf{7.87 \pm 3.2}$	8.32 ± 3.7
shakeVinegar	6.92 ± 1.3	6.88 ± 1.7	11.1 ± 3.1	8.46 ± 1.5	$\mathbf{6.27 \pm 1.7}$
magnifyText	9.05 ± 1.6	7.41 ± 0.78	10.5 ± 2.3	11.4 ± 1.7	$\mathbf{6.70 \pm 0.99}$
putOnShoe	12.6 ± 1.7	11.0 ± 1.6	10.8 ± 3.8	11.0 ± 3.2	$\mathbf{8.53 \pm 1.3}$
hitGolfBall	65.6 ± 6.8	35.7 ± 9.0	32.8 ± 10	31.0 ± 7.8	$\mathbf{29.0 \pm 6.9}$
sweepFloor	52.6 ± 11	50.9 ± 7.2	30.9 ± 6.8	29.2 ± 8.5	$\mathbf{26.6 \pm 9.1}$
paintWall	18.8 ± 1.6	12.9 ± 2.0	14.9 ± 2.2	10.7 ± 2.3	$\mathbf{8.84 \pm 1.9}$
hitTennisBall	34.4 ± 4.2	33.2 ± 2.3	38.9 ± 5.5	29.4 ± 4.8	$\mathbf{26.7 \pm 4.0}$
Average	22.1 ± 17	17.0 ± 13	16.0 ± 9.9	14.1 ± 8.3	$\mathbf{12.6 \pm 7.9}$

5.6.1. Edge map comparison

To evaluate the pth particle's weighting $\pi_t^{(p)}$, we compare it with a chamfer image computed from the latest RGB image $\mathbf{r}_t \in \mathbf{z}_t$. The chamfer image is calculated by convolving $\mathbf{r}_t$ with a gradient-based edge detection mask, thresholding and smoothing the results with a Gaussian mask and rescaling values into the range $[0, 1]$. Each pixel in the resulting image $\hat{\mathbf{r}}_t$ contains a value proportional to its proximity to an edge in the original image.

The pth particle's object-pose is then given by the translation

$$\mathbf{a}_t^{(p)} = \mathbf{O}^{i_{T,t}^{(p)}}_{n_{T,t}^{(p)},(1:3,l_{T,t}^{(p)})} \cdot h_t \tag{14}$$

where h_t is the test participant's current height estimate, and the rotation

$$\mathbf{q}_t^{(p)} = \mathbf{O}^{i_{R,t}^{(p)}}_{n_{R,t}^{(p)},(4:7,l_{R,t}^{(p)})} \tag{15}$$

These values are defined relative to the $l_{T,t}^{(p)}$th and $l_{R,t}^{(p)}$th part predictors, respectively, which we compute from the latest body-pose estimate $\hat{\mathbf{b}}_t$ (see also Section 4.2) in order to configure the object. Following [32], the object is then projected into the chamfer image $\hat{\mathbf{r}}_t$ and a set of equally spaced sample points $\mathcal{R}^{(p)}$ computed around its boundaries (see Fig. 3c for an example). These points are used to compute a sum of squared differences between the object edges and the image edges

$$\Sigma^{(p)} = \frac{1}{|\mathcal{R}^{(p)}|} \sum_{r \in \mathcal{R}^{(p)}} (1 - \hat{\mathbf{r}}(r))^2 \tag{16}$$

where $\hat{\mathbf{r}}(r)$ gives the value of the chamfer image at the rth sample point. Finally, we calculate the particle's weight as

$$\pi_t^{(p)} = \exp[-\Sigma^{(p)}] \tag{17}$$

and normalise across the whole set to give $\sum_{p=1}^{P} \pi_t^{(p)} = 1$.

6. Experimental results

6.1. Data

We have collected a database of 90 12-s videos ($N = 360$ frames) showing 6 participants (5 male, 1 female, aged 25–40) performing 15 separate human-object interactions. These interactions are listed in Table 1. The database was recorded using Kinect (via the Kinect for Windows SDK) and each observation $\mathbf{z}_n = [\mathbf{r}_n, \mathbf{d}_n, \mathbf{b}_n]$ consists of an RGB image $\mathbf{r}_n$, a depth image $\mathbf{d}_n$, and a body-pose estimate $\mathbf{b}_n$. We then manually labelled the sequences with 3D object-poses for each instant $\mathbf{o}_n$. To do this we wrote a keyframing UI that allowed for easy 3D rotation of the object relative to the 3D skeleton, and immediately projected adjustments into the associated RGB and depth streams for comparison. Intermediate object-poses were recovered using SLERP. The objects used in labelling were selected from the Google 3D Warehouse to match our real objects as closely as possible, and are shown in Fig. 7.

6.2. Error evaluation

To evaluate the accuracy of tracking we use the weighted particle set to compute an expected object-pose translation

$$E(\mathbf{a}_t) = \sum_{p=1}^{P} \pi_t^{(p)} \cdot T\left[\mathbf{O}^{i_{T,t}^{(p)}}_{n_{T,t}^{(p)},(1:3,l_{T,t}^{(p)})} \cdot h_t\right] \tag{18}$$

and rotation

$$E(\mathbf{q}_t) = \sum_{p=1}^{P} \pi_t^{(p)} \cdot T\left[\mathbf{O}^{i_{R,t}^{(p)}}_{n_{R,t}^{(p)},(4:7,l_{R,t}^{(p)})}\right] \tag{19}$$

where the function $T[]$ applies a transformation between the local part predictor basis and the global Kinect coordinate system, centred on the sensor. This allows for a weighted average of object-poses that may be defined relative to different part predictors. We then configure our tracking object using the expected pose

$$\hat{\mathbf{o}}_t = [E(\mathbf{a}_t), E(\mathbf{q}_t)] \tag{20}$$

and calculate an average vertex-vertex error score (in cm) with our ground truth label. The vertices used for comparison are highlighted with red markers in Fig. 7.

6.3. Object-pose tracking

We used the proposed approach to track object-poses during each of the 15 human-object interactions for every participant. The tracker is supplied with the object class and participant handedness but must then initialise and track the object-pose for the remainder of the sequence. All experiments were conducted on *unknown participants*, meaning that we included no training data (body-poses or object-poses) from the participant being tested. This resulted in a set of $S = 5$ training participants for each

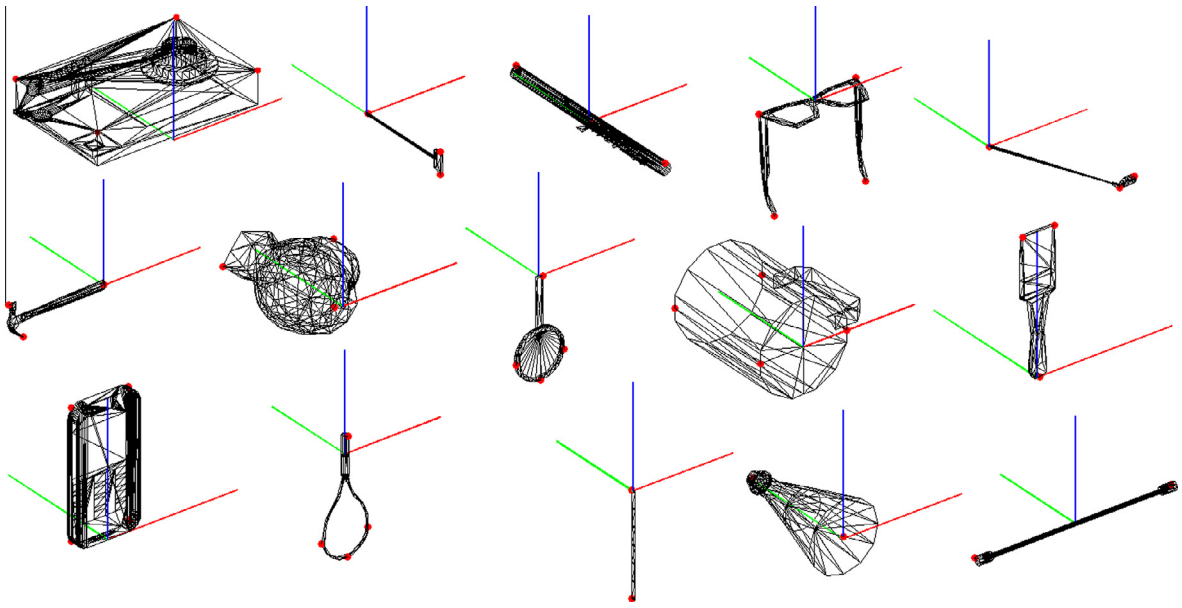


Fig. 7. Individual objects from Google 3D Warehouse and virtual markers (red) used for error evaluation. Left-to-right starting from top left: camera, floorbrush, flute, glasses, golf club, hammer, jug, magnifying glass, mug, paintbrush, mobile phone, tennis racquet, shoehorn, vinegar and weightbar. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

experiment. We used $P = 100$ particles, the values $\sigma_G = \sigma_D = 1$ for generative propagation and $\sigma_\theta^2 = \pi/16$ and $\sigma_\epsilon^2 = 0.1$ for the calculation of the discriminative-generative mixing fractions. The former is an angle between quaternions on the unit sphere and the latter a fraction of the participant's body height. We chose the closeness threshold for each training sequence, δ^i_{min}, by computing the distances between every constituent body-pose and the nearest body-pose across all *other* training participants, then taking the median separation value. We found that approximating inter-participant variation in this way provided a useful model for intra-participant variation, which is more difficult to capture automatically.[2] Smaller thresholds are recovered for more controlled interactions with little stylistic variation (e.g. `liftWeights`) and larger thresholds for underconstrained interactions such as `hitTennisBall` (where the ball is imagined).

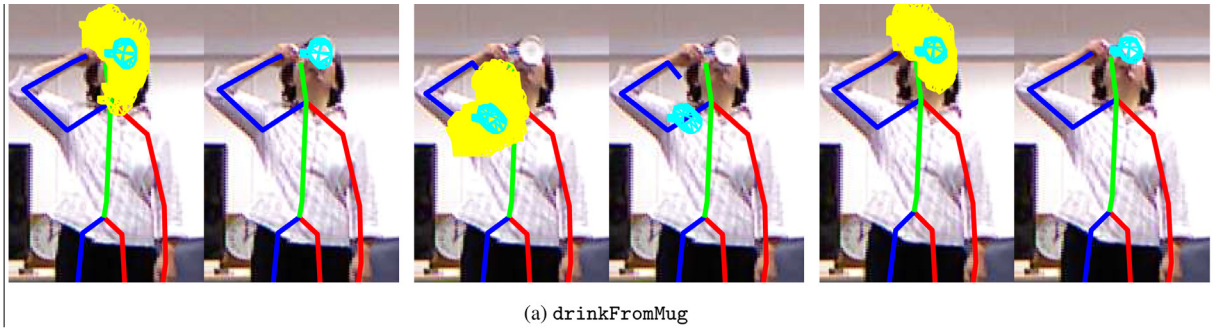
The average vertex error score in Section 6.2 was computed at every frame of every participant's sequence. A *sequence error* was then calculated as the mean across all frames in a sequence. The rows of Table 1 give the mean and standard deviation across all 6 participants' sequence errors for every human-object interaction. The columns of Table 1 then correspond to the following parameter settings:

G (hand): using only generative particles and only the hand as a part predictor. This approach is broadly representative of other approaches (e.g. [1]) that have used only the location of the hand to drive object tracking.
G + D (hand): using both generative and discriminative particles and only the hand as a part predictor. As above, but re-initialisation is now possible through the introduction of discriminative particles.
G + D (random): using both generative and discriminative particles and randomly chosen part predictors for rotation and translation. Representative of the approach in [11].
G + D (stable): using both generative and discriminative particles and translation and rotation part predictors chosen by minimising Eqs. (8) and (7) respectively. This approach favours stable part predictors which minimise the changes in relative object-pose over time.
G + D (stable-proximate): using both generative and discriminative particles and rotation and translation part predictors chosen by minimising Eqs. (7) and (9) respectively. As above, but while the rotational part predictor is chosen as the most stable, the translational part predictor is chosen as that which is closest (proximate) to the object.

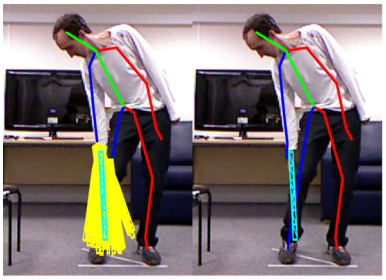
6.4. Tracking performance

Table 1 shows that hypothesising object-pose estimates based only on a participant's hand [column 1, **G (hand)**] gives the highest average tracking error (22.1 cm), and the highest individual errors in 10/15 of the human-object interactions. Introducing discriminative particles [column 2, **G + D (hand)**] improves the average tracking accuracy (17.0 cm) and individual accuracies in all 15 human-object interactions, but errors remain high in a number of cases, e.g. `liftWeights`. Using randomly selected part predictors [column 3, **G + D (random)**] produces a small further reduction in the average tracking error (16.0 cm) but worsens tracking accuracy in 8 of the interactions. Interestingly however, this randomised approach achieves the best scores across all conditions for two of the interactions: `drinkFromMug` and `playFlute`. Selecting both the translation and rotation part predictors as those which minimise the future change in relative object-pose [column 4, **G + D (stable)**] produces a lower average tracking error again (14.1 cm), and gives improvements across the more dynamic interactions that involve large object-rotations, e.g. `hitGolfBall`, `sweepFloor`, `hitTennisBall`. However, for many of the more static interactions, e.g. `pourCream`, `recordVideo`, `magnifyText`, tracking errors are higher than when based on the hand alone. Predicting translation from the closest part predictor and rotation from the stablest future predictor [column 5, **G + D (stable-proximate)**] produces the lowest average tracking error of all our experiments (12.6 cm). It also produced the lowest tracking errors in 10/

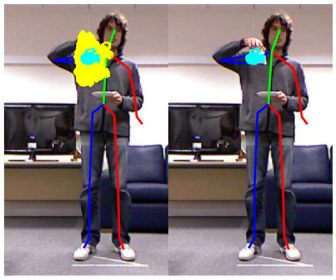
[2] In cyclic interactions, the broad aim is to identify similar body-poses across the different cycles of a participant's own interaction with the object. For some interactions the differences between cycles are greater than for others.



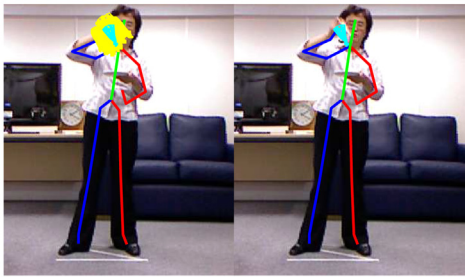
(a) drinkFromMug



(b) putOnShoe



(c) pourCream



(d) shakeVinegar

Fig. 8. The effect of localised body-pose estimation errors on object-pose tracking for **G + D (stable-proximate)**. Each image pair shows all particle hypotheses in yellow (left) and the expected object-pose in cyan (right): The top row (a) shows three instants from a `drinkFromMug` sequence where the dominant hand has been incorrectly estimated in the middle image pair. Object-pose estimates are good just before and after the hand-pose estimation error, but because the hand is always proximal in the training data the hand-pose estimation error causes a large object-translation error. In the majority of cases, localised body-pose estimation errors affect part predictors that play a much less important role in object-pose prediction. For example (b), (c) and (d) all show localised errors in the non-dominant arm that have no impact on object-pose tracking. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

15 of the individual interactions. Fig. 13a–o alternate through the six participants showing results from the **G + D (stable-proximate)** approach with rotated 3D views of the recovered body- and object-poses. Videos (including failure cases) are also available in the Supplementary materials.

6.5. Failure cases

In this section we cover the important failure cases for the **G + D (stable-proximate)** system. This allows us to show some detailed results images, and additionally highlight a number of positive aspects of the approach.

6.5.1. Localised body-pose estimation errors

The first failure case is seen when a part predictor that is uniquely stable (rotation prediction) or proximal (translation prediction) in the training set is incorrectly estimated during tracking (localised body-pose error). For example, Fig. 8a shows three frames from a `drinkFromMug` sequence where an incorrectly estimated right hand temporarily disrupts tracking in the middle image. The dominant hand is proximal to the mug almost without exception in the training set and the particle set is therefore focused around a quite narrow range of object-translations, all relative to the hand. When, in the second image pair, the hand pose incorrectly flips through 90°, all object-translation estimates move with it.

In the majority of cases localised body-pose estimation errors have no impact on object-tracking accuracy. See for example the errors in the non-dominant arm in Fig. 8b–d. The problem only arises when one part predictor that is significantly "better" than all other part predictors (in terms of rotation or translation prediction) is incorrectly estimated. Even in these cases the object-pose track recovers as soon as the body-pose estimate recovers (Fig. 8a, right-hand image pair). Although it is slightly less accurate in general, the impact of such events on **G + D (random)** is much less because it constantly uses *all* part predictor for object-pose prediction, see Fig. 9.

6.5.2. Global body-pose estimation errors

The second failure case is seen when there is a complete failure in body-pose tracking (global body-pose error). If no body parts are correctly estimated then all part predictors will give incorrect object-pose estimates and object tracking will fail. Kinect suffers global body-pose errors only very rarely on our dataset, because the vast majority of body-poses are front facing. One time when this is not the case is when participants turn to their right and stoop down to pick up the weight bar from the stool in `liftWeights`. For example, see Fig. 10a where body-pose estimation breaks down completely, and object-pose tracking with it.

As soon as the person stands up and faces the camera again, Kinect's discriminative tracking algorithm [24] is able to recover body-pose estimation. When a good opportunity for object-pose re-initialisation subsequently occurs (participant stands with arms straight, preparing to lift) discriminative particles allow object-pose tracking to recover, Fig. 10b.

6.5.3. High particle diversity, lack of image evidence

The third failure case is seen when there is genuine diversity in the way an object is held by different people, and there is insufficient image evidence to resolve the rotational diversity in the resulting particle set. A good example is the underarm tennis swing (`hitTennisBall`, Fig. 11a) where, as participants were asked to imagine the ball, there is considerable variation in performance. This is particularly true during the backswing where some use their wrist to swing the racquet through a force-generating arc and some do not, but the associated body-pose estimates are very similar. Similar twisting of the wrist tended to occur (or not) at the start and end points of the golf swing (`hitGolfBall`, Fig. 11b). In these situations the set of object-pose hypotheses become considerably more diverse and we must rely on the observation density

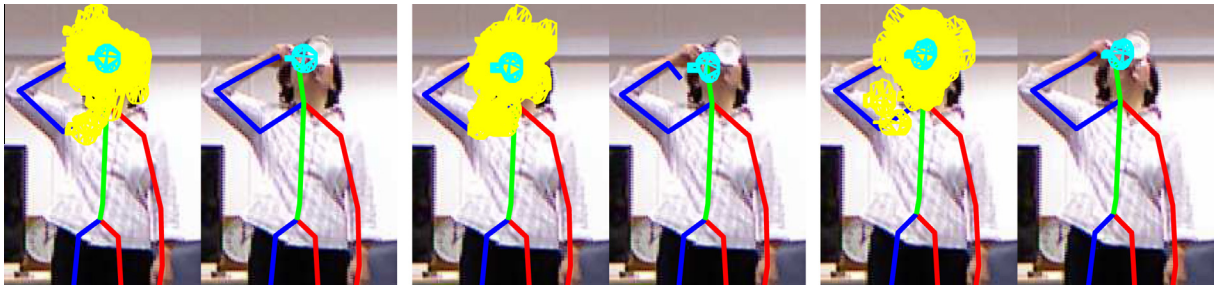
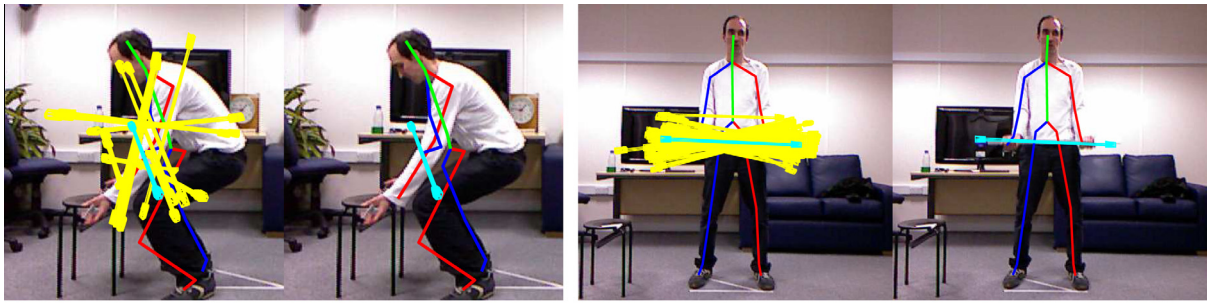
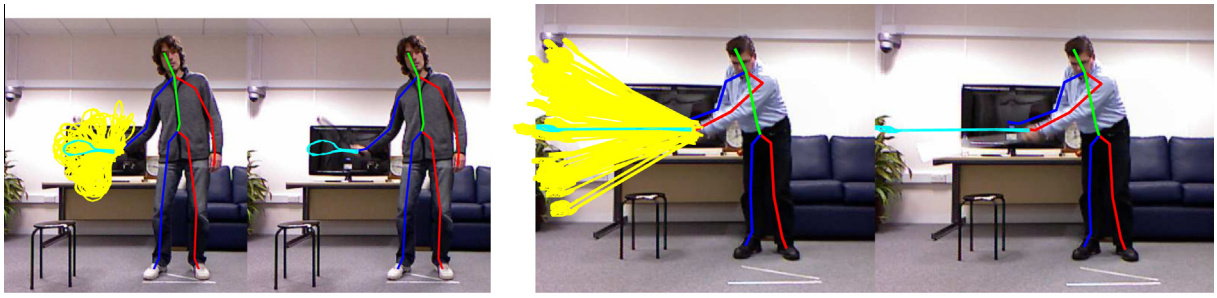


Fig. 9. The effect of localised body-pose estimation errors on object-pose tracking for **G + D (random)**. The figure shows three instants from a `drinkFromMug` sequence (same instants as Fig. 8a) where the dominant hand has been incorrectly estimated in the middle image pair. Object-pose estimates are not quite as accurate just before and after the hand estimation error, but the effect of the hand-pose estimation error is minimal because all 19 part predictors are constantly being used to make object-translation estimates.



(a) (b)

Fig. 10. The effect of global body-pose estimation errors on object-pose tracking for **G + D (stable-proximate)**: (a) Kinect is unable to cope with the crouched and rotated body-pose in frame 1 of `liftWeights`. The resulting estimate's nearest neighbour in the training set is another incorrect body-pose which is not truly similar and itself has no nearest neighbours, the resulting particle set is therefore sparse, diverse and very inaccurate. (b) By frame 68 body-pose estimation has recovered and the participant has entered a pose that allows object-pose tracking to re-initialise with discriminative particles.



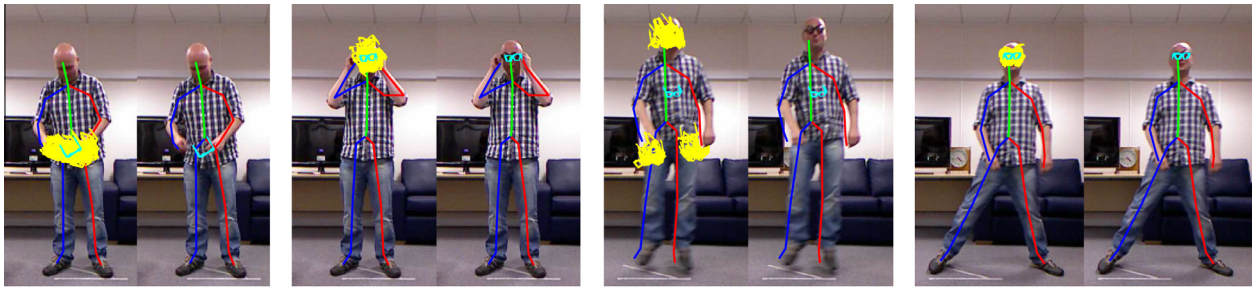
(a) `hitTennisBall` (b) `hitGolfBall`

Fig. 11. The effect of a lack of image evidence on object-pose tracking for **G + D (stable-proximate)**: (a) `hitTennisBall`; (b) `hitGolfBall`. Both interactions involve a backswing where some participants swing the object through a force generating arc with little or no change in their body-pose. This results in a diverse particle set and, where image evidence is not strong enough to resolve the uncertainty (e.g. because of the presence of background clutter (a) or motion blur (b)), tracking becomes inaccurate and noisy.

(Section 5.6) to resolve the correct answer. Where this is difficult, e.g. due to background clutter (Fig. 11a) and/or motion blur (Fig. 11b), the object-track becomes noisy.

6.5.4. Unusual interactions

The final failure case can occur when a participant performs an unusual interaction with an object. Although unusual interactions are not necessarily a problem, the important question is whether the relevant relationships between part predictors and the object hold true, see Section 7 for a full discussion. An interesting example of unusual interaction from our own dataset is shown in Fig. 12. During `putOnGlasses` participants were asked to move to their left and then their right whilst facing the camera and wearing the glasses. This particular participant chose to complete this task by jumping rather than stepping. During this period the interaction is unusual given the dataset, but the head continues to be proximal and a reliable predictor of translation, and all stationary joints in the torso and lower body continue to be good predictors of rotation. Tracking fails however, because when the participant swings their arms up in front of themselves to generate an upward force, the body-pose matches well with training participants starting to bring the glasses up to their face (body-pose in Fig. 12c is close to that in Fig. 12a). The mapping to object-poses is stable at this point in the training data and this causes discriminative object-poses to re-initialise the glasses in the participant's hands. Notice that the object-pose quickly recovers as normal interaction is resumed (the participant steps back to their right).



(a) (b) (c) (d)

Fig. 12. The impact of an unusual interaction on object-pose tracking for **G + D (stable-proximate)**. During `putOnGlasses` we asked participants to move to their left and then their right while wearing the glasses and facing Kinect. This participant chose to jump. As they throw up their hands to generate the upwards force their body-pose matches well with the start of the glasses being raised up to the face: (a) participant starts to raise glasses up to their face; (b) glasses correctly resting on the head; (c) the jumping pose is incorrectly recognised and causes discriminative object-poses to appear in the participant's hands – the expected object-pose (cyan) appears in between the three competing modes of the distribution; (d) tracking recovers as the participant resumes normal interaction and steps back to their right. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7. Discussion

Many of the more static interactions, e.g. `recordVideo`, `magnifyText`, involve periods where the object remains near-stationary with respect to all part predictors. We might expect, therefore, that all part predictors would offer equally good predictions of object-pose. However, this does not hold true, with the move to random part predictors producing a rise in tracking error for many of these interactions (column 3, Table 1) versus the use of the hand alone (column 2). This drop in accuracy is due primarily to errors in object-translation estimation. Closest body-pose matches in the training set are close but not identical, and therefore the translation predictions of part predictors (particularly distant ones) will also be, to some degree, imprecise. See for example the much wider distribution of hypotheses for **G + D (random)** in Fig. 9 compared to **G + D (stable-proximate)** in Fig. 8a. Note that distant part predictors are still stable (see for example `recordVideo` in Fig. 6a) and so matters are not improved by using the stablest part predictor (column 4). Generally speaking, the key to the good performance of the hand as a part predictor for these tasks is its proximity to the object: the object is so close to the origin of the basis that the exact translation hardly matters.

There are two cases where the use of the hand for translation prediction becomes problematic: first, where the location of the hand is incorrectly estimated; second, where the object does not remain gripped by the hand throughout the interaction. The first case explains the unusually good relative performance of using random part predictors on `drinkFromMug` and `playFlute` (column 3), as the arms were incorrectly estimated by Kinect in a number of these sequences leaving the hand a comparatively poor predictor of object location (e.g. Fig. 8a). The second case is our motivation for constantly recalculating the closest part predictor (Eq. (9)) in order that the translation predictor can, for example, move between both of the hands and the torso during `liftWeights` and from hands to head during `putOnGlasses`, see also Fig. 6e for average proximity plots from a representative participant. In fact, stable translation predictors actually marginally outperform proximate translation predictors for both these sequences (column 4). We attribute this to the fact that (unlike the static interactions mentioned above) the only other parts that are equivalently stable are *also* nearby, e.g. the shoulders (in addition to the head) for `putOnGlasses` and the forearms (in addition to the two hands) for `liftWeights`. Using these few extra parts in addition to the single proximate part may make **G + D (stable)** tracking slightly more robust to localised body-pose estimation errors in these interactions.

For object-rotation there is no sense in which we can choose the "nearest" part predictor. But by choosing the part predictor that minimises relative future changes in rotation we are able to reduce tracking errors amongst the most dynamic interactions (column 4). As discussed in Section 4.4 the future predictions of all part predictors are, in theory, equally valid, but selecting those which minimise relative rotation ensures the generative particle set is used more efficiently. Interestingly, the body parts that minimise rotational changes are not always those that are close to and "move with" the object. In `liftWeights` for example, where the angle of the weight bar is carefully maintained during repetitions, rotational changes are low relative to the torso and legs, but high relative to the moving arms. Perhaps more surprisingly the same is also true for an interaction like `paintWall`, i.e. the paintbrush is not held as a rigid extension of the forearm as might be imagined (see also Fig. 5c for average delta plots from a representative participant).

By combining the use of stable rotational part predictors with the use of the proximate translational part predictors, we get the lowest overall tracking errors (column 5). Here we briefly note some positive aspects of this approach before going on to highlight some weaknesses and areas for possible future work in the remainder of this section. First, we note that discriminatively re-initialising a fraction of the particle set improves tracking accuracy while only incurring the computational cost of a single nearest neighbour search at every frame (**D1**, Section 5.4). Second, our discriminative strand could be used to address a *single image* RGB-D object-pose estimation problem by generating and testing discriminative particles only; this topic has not received any previous attention (see Section 2). Third, the discriminative strand allows the system to recover tracking even after complete failures in body-pose estimation, e.g. Fig. 10. Finally, as our system does not strictly require RGB-D sensor observations, only good 3D body-pose estimates, it could be used to improve object-pose tracking accuracy (same number of particles) or efficiency (fewer particles) in synchronised multi-camera scenarios such as [14,17].[3]

A potential weakness of the proposed approach is its sensitivity to localised body-pose estimation errors when they affect important part predictors (see also Section 6.5). In reality this happens only rarely. Localised pose estimation errors by Kinect tend to be randomly distributed across the body and usually affect a part predictor (there are 19 in total) that is not being used (or at least heavily relied upon), e.g. Fig. 8b–d. Furthermore, localised errors are

[3] Even with 8 cameras, tracking objects through interactions is challenging: [14] use 5 times as many particles to track a stick as they do to track the body-pose of the participant interacting with it.

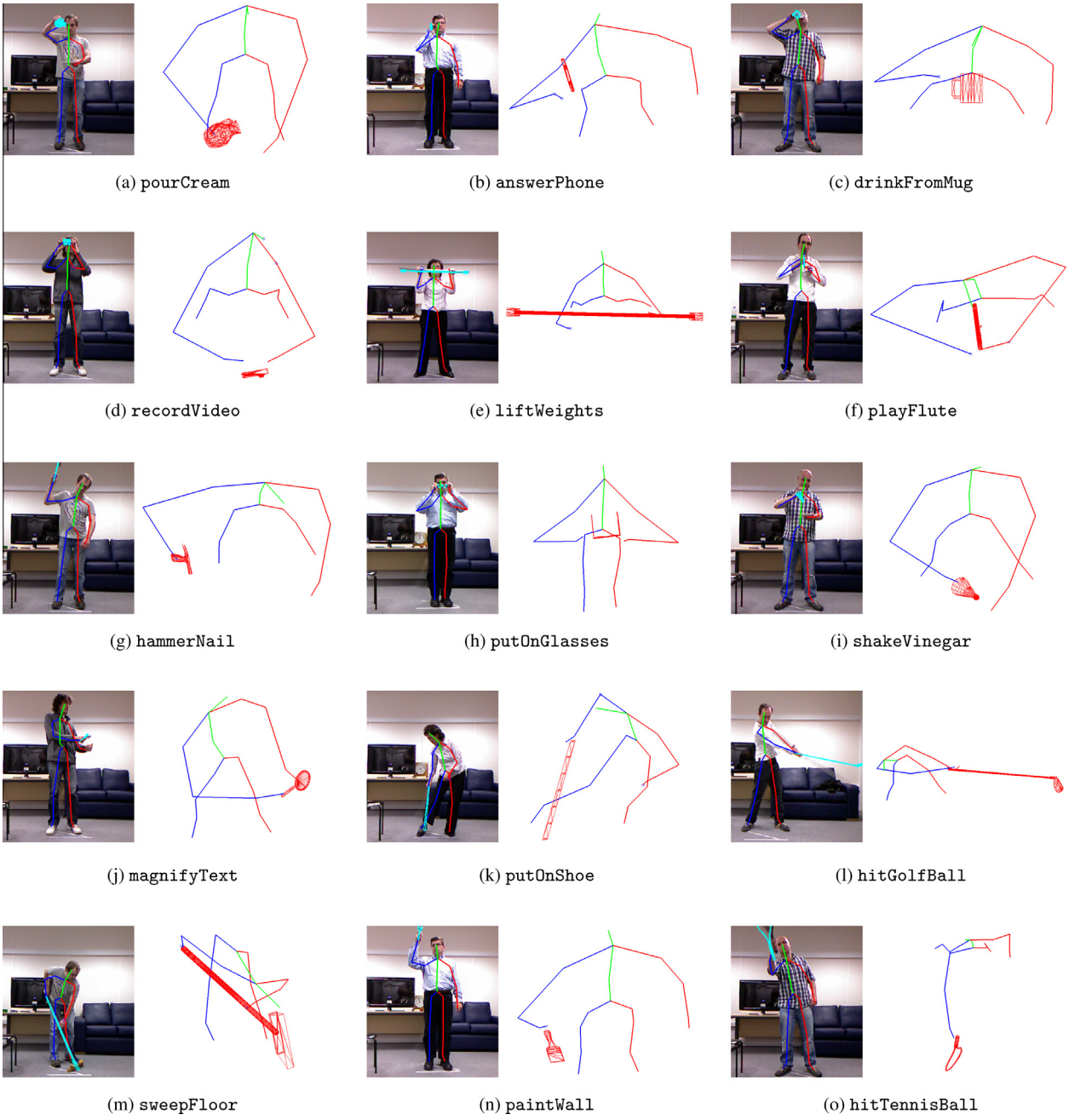


Fig. 13. Representative images from each interaction, alternating through the six participants (best viewed electronically). Each image has been paired with a view of the resulting 3D body- and object-poses, rotated to give an informative view of the interaction.

relatively rare: it is because body-pose estimation is robust that a method reliant solely on the nearest part predictor for translation estimation emerges as the most successful strategy. In situations where localised body-pose errors are anticipated (e.g. due to partial occlusion) it may be necessary to move away from relying on single body parts. Our full body method supports this and either of **G + D (random)** or **G + D (stable)** remain valuable options in this context.

Using more part predictors leads to more diverse particle sets (Fig. 9 versus Fig. 8a) and another potential weakness is the inability of the observation density to identify good object-pose candidates from bad. This is also an issue when there is genuine diversity in the way an object is held, e.g. during the backswings in Fig. 11a and b. The edge-based observation density has a positive impact on tracking, reducing errors in 14/15 interactions by an average of 2.1% across all subjects and conditions. The effect was greatest in relatively slow or static interactions (e.g. `drinkFromMug`, 7.8% reduction) or those where there was consistently a strongly contrasting background to the object (e.g. skin in `putOnGlasses`, 5.2% reduction). But it did also worsen performance in 1 interaction (`sweepFloor`, 4.3% increase). We believe this was due to the presence of a large expanse of similarly coloured background clutter (as the blue brush handle was pushed in front of the blue sofa in Fig. 13m).

Our edge-based observation density is a candidate for future work, but the way forward is not entirely clear. In line with other RGB-D approaches [1,4,13] we have tried additionally evaluating object-pose hypotheses against the depth map. This was done by

projecting the object into the Kinect depth image, performing back-face culling and evaluating the distance from a number of evenly sampled points on the object's visible surface to the nearest real surface in the depth data (using a "chamfer volume" approach similar to [33]). However, as discussed in Section 2, many objects are simply not visible in the depth image (i.e. they do not reflect the sensor's light source). For our own dataset this was true for 8/15 of the objects. Those that are visible fall so close to the surface of the body that the potential for surface confusion is high, and background subtraction (e.g. [13]) cannot help.

As an alternative, one might choose to apply more sophisticated object detection techniques to the RGB image, following the single RGB image literature [6,7,16]. The first issue here is that the RGB images provided by RGB-D sensors are of considerably lower quality than those used in single image studies. For example, where single image sports databases feature high quality (high-resolution, short-exposure) images of fast-moving tennis racquets and cricket bats, our Kinect dataset features low-resolution (640 by 480), motion-blurred RGB data for similar objects (e.g. `hitGolfBall` and `hitTennisBall`).[4] The second issue is that of occlusion. Single images of interactions have tended to contain unoccluded objects (e.g. allowing the use of "off-the-shelf" [19] object detectors in [16]), but dealing with videos requires coping with the often severe occlusion of objects by the human interacting with them. For example, building sliding-window RGB detectors for phones and shoehorns is likely to be of little or no value for the human-object interactions of `answerPhone` (Fig. 13b) and `putOnShoe` (Fig. 13k). More sophisticated appearance models that account for parts of both human and object may be the way forward, but without making viewpoint assumptions (e.g. egocentric views [15]) the learning task will be considerable.

The observation density is particularly important when dealing with unusual interactions. For example, in Fig. 12c the generative particles do a good job of representing the correct object-pose (glasses on the participant's head) but they appear no "better" in terms of the edge map than the discriminative particles that have appeared around the participant's hands (and so the expected object-pose is poor). A more common form of unusual interaction is where a participant's body-poses remain typical, but the way in which they are holding the object is quite different. These differences arise primarily as rotational changes brought about through wrist and finger manipulations that are difficult or impossible to detect (in terms of body-pose). Good examples were during the backswings in `hitTennisBall` and `hitGolfBall`. The system naturally generates a broad range of hypotheses from the varied training data (usually through a predominantly generative particle set), but is unable to identify the best candidates via the observation density, see for example Fig. 11a and b. However, the presence of motion blur and similarly coloured background clutter makes this an extremely challenging problem for any observation density.

Where unusual interactions entail both new body-poses and new object-poses – the most challenging case – the system will almost certainly fail. However, it is interesting to note that this is often quite improbable. Many of the interactions are well constrained by the nature of the objects or the task, e.g. a mug must be held level to avoid spilling, phones must be held with the speaker facing the ear to hear the conversation, heavy weights bars must be held horizontally and steadily. For example, it is instructive to imagine participants: raising up their mug of drink in a toast, pinning their phone against their ear with their shoulder whilst reaching for something, lifting the weights bar above their heads with their arms extended. In each case the stable rotation predictors in the lower body and torso should continue to give good predictions. The issue would be which body-pose in the training set the test pose happened to fall closest to (it truly matches with none) as it may result in a bad choice of proximate translation predictor (e.g., head rather than hands in these examples). Where there are no true nearest neighbours (a simple distance threshold could be applied), a more pragmatic approach may be to resort to the hands for translation prediction and sample object-poses from a wider range of body-poses.

[4] Furthermore, Kinect RGB images are not guaranteed to be exactly synchronised with depth images (and therefore body-pose data).

8. Conclusions

We have presented a system that is, to the best of our knowledge, the first able to track full 3D object-poses (translation and orientation) from RGB-D observations of human-object interactions. Our method allows independent predictions about object-pose to be made from each of the different parts of the body. We use these predictions to drive a combined generative and discriminative particle-based object-pose tracker. During tracking, the system constantly looks for opportunities to re-initialise particles based on the nature of the mapping between the body- and object-pose spaces. Where re-initialisation is possible, the best body parts from which to make predictions are selected automatically. We have found the optimal choice often proves to be different between the object's rotational and translational components. Quantitative evaluation on a large dataset has enabled us to demonstrate robustly the importance of discriminative re-initialisation versus pure generative tracking, and the value of careful part predictor selection over random choice, or the use of the hands (as is common in the literature). In constructing the proposed approach we have also resisted making assumptions about de-cluttered, or static backgrounds, or about the visibility of particular classes of object in depth data; all of which are difficult to guarantee in real-world scenarios. By recovering precise changes in object-pose, the presented methods open up the possibility for more detailed computational reasoning about human-object interactions and their outcomes.

Acknowledgments

The authors would like to thank the Dalton Research Institute for support of J. Darby.

References

[1] J. Gall, A. Fossati, L. van Gool, Functional categorization of objects using real-time markerless motion capture, in: CVPR, 2011, pp. 1969–1976.
[2] A. Prest, V. Ferrari, C. Schmid, Explicit modeling of human-object interactions in realistic videos, PAMI 35 (4) (2013) 835–848.
[3] H. Kjellström, J. Romero, D. Kragić, Visual object-action recognition: Inferring object affordances from human demonstration, CVIU 115 (1) (2011) 81–90.
[4] H. Koppula, R. Gupta, A. Saxena, Learning human activities and object affordances from RGB-D videos, IJRR 32 (8) (2013) 951–970.
[5] A. Prest, C. Schmid, V. Ferrari, Weakly supervised learning of interactions between humans and objects, PAMI 34 (3) (2012) 601–614.
[6] A. Gupta, A. Kembhavi, L.S. Davis, Observing human-object interactions: using spatial and functional compatibility for recognition, PAMI 31 (10) (2009) 1775–1789.
[7] B. Yao, L. Fei-Fei, Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses, PAMI 34 (9) (2012) 1691–1703.
[8] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real time motion capture using a single time-of-flight camera, in: CVPR, 2010, pp. 755–762.
[9] M. Salzmann, R. Urtasun, Combining discriminative and generative methods for 3D deformable surface and articulated pose reconstruction, in: CVPR, 2010, pp. 647–654.

[10] J.A. Lasserre, C.M. Bishop, T.P. Minka, Principled hybrids of generative and discriminative models, in: CVPR, 2006, pp. 87–94.
[11] J. Darby, B. Li, R.J. Cunningham, N.P. Costen, Object localisation via action recognition, in: ICPR, 2012, pp. 817–820.
[12] M. Isard, A. Blake, ICONDENSATION: unifying low-level and high-level tracking in a stochastic framework, in: ECCV, 1998, pp. 893–908.
[13] B. Packer, K. Saenko, D. Koller, A combined pose, object, and feature model for action understanding, in: CVPR, 2012, pp. 1378–1385.
[14] H. Kjellström, D. Kragić, M.J. Black, Tracking people interacting with objects, in: CVPR, 2010, pp. 747–754.
[15] H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in: CVPR, 2012, pp. 2847–2854.
[16] C. Desai, D. Ramanan, C. Fowlkes, Discriminative models for static human-object interactions, in: CVPR Workshops, 2010, pp. 9–16.
[17] B. Rosenhahn, C. Schmaltz, T. Brox, J. Weickert, D. Cremers, H.-P. Seidel, Markerless motion capture of man-machine interaction, in: CVPR, 2008.
[18] A. Pieropan, C.H. Ek, H. Kjellström, Functional object descriptors for human activity modeling, in: ICRA, 2013, pp. 1282–1289.
[19] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, PAMI 32 (9) (2010) 1627–1645.
[20] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: ECCV, 2008, pp. 234–247.
[21] S. Stalder, H. Grabner, L. Van Gool, Beyond semi-supervised tracking: tracking should be as simple as detection, but not simpler than recognition, in: ICCV Workshops, 2009, pp. 1409–1416.
[22] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: CVPR, 2009, pp. 983–990.
[23] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real-time human pose tracking from range data, in: ECCV, 2012, pp. 738–751.
[24] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: CVPR, 2011, pp. 1297–1304.
[25] OpenNI. <http://www.openni.org>.
[26] J. Romero, H. Kjellström, D. Kragić, Hands in action: real-time 3D reconstruction of hands in interaction with objects, in: ICRA, 2010, pp. 458–463.
[27] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, V. Lepetit, Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes, in: ICCV, 2011, pp. 858–865.
[28] C. Choi, H.I. Christensen, Robust 3D visual tracking using particle filtering on the SE(3) group, in: ICRA, 2011, pp. 4384–4390.
[29] D.Q. Huynh, Metrics for 3D rotations: comparison and analysis, J. Math. Imaging Vis. 35 (2) (2009) 155–164.
[30] C. Gramkow, On averaging rotations, IJCV 42 (1-2) (2001) 7–16.
[31] E.-J. Ong, A.S. Micilotta, R. Bowden, A. Hilton, Viewpoint invariant exemplar-based 3D human tracking, CVIU 104 (2) (2006) 178–189.
[32] J. Deutscher, I. Reid, Articulated body motion capture by stochastic search, IJCV 61 (2) (2005) 185–205.
[33] J. Darby, B. Li, N.P. Costen, Human activity tracking from moving camera stereo data, in: BMVC, 2008, pp. 865–874.