Functional Categorization of Objects using Real-time Markerless Motion Capture

Juergen Gall[1] Andrea Fossati[1] Luc van Gool[1,2*]
BIWI, ETH Zurich ESAT-PSI / IBBT, KU Leuven
{gall,fossati}@vision.ee.ethz.ch vangool@esat.kuleuven.be

Abstract

Unsupervised categorization of objects is a fundamental problem in computer vision. While appearance-based methods have become popular recently, other important cues like functionality are largely neglected. Motivated by psychological studies giving evidence that human demonstration has a facilitative effect on categorization in infancy, we propose an approach for object categorization from depth video streams. To this end, we have developed a method for capturing human motion in real-time. The captured data is then used to temporally segment the depth streams into actions. The set of segmented actions are then categorized in an unsupervised manner, through a novel descriptor for motion capture data that is robust to subject variations. Furthermore, we automatically localize the object that is manipulated within a video segment, and categorize it using the corresponding action. For evaluation, we have recorded a dataset that comprises depth data with registered video sequences for 6 subjects, 13 action classes, and 174 object manipulations.

1. Introduction

Challenging computer vision tasks like object detection [7] or action recognition [24] consist of recognizing and localizing objects or motions of a specific class in images or videos. This means that the objects of interest are already categorized and the instances within a class are assumed to share a certain similarity, which is usually learned from the appearance. What we propose is different from such classic paradigm in two important ways: We believe that the way objects are used should count at least as much as their appearance for their categorization. As shown in the literature, for applications like autonomous robotics, a categorization based on functional similarity is more task-relevant [29, 27, 16]. These approaches learn rather the affordance [13] than the appearance of objects. Moreover, we want to build a system that does not make use of prior information about the objects, to achieve more generality. This is inspired by unsupervised methods for object recognition, also referred to as object discovery techniques [31], which categorize objects from a set of unlabeled data, instead of relying on a given categorization. While unsupervised categorization of objects using an appearance-based similarity measure has been of particular interest in the last years [31], categorizing based on functional similarity in an unsupervised fashion, which is addressed in this work, has received little attention.

(a) (b)

Figure 1. Our approach extracts manipulated objects from video data and categorizes them according to their functionality (a). To this end, the motion of the subject is captured in real-time and the video sequences are segmented and clustered in an unsupervised manner. In this work, the processing is performed on low-resolution depth data (b).

Our approach is motivated by psychological studies that give evidence that human demonstration has a facilitative effect on categorization in infancy. In the study of Booth [4], infants had to discriminate objects of two categories that were similar in appearance. When the manipulator was visible, infants were more likely to learn to differentiate between the two categories. They learned it also more rapidly than infants that observed only static objects or object manipulations without the additional cue of the human agent.

In this work, we follow the concept of a human agent as stimulus for object categorization as illustrated in Fig. 1. To this end, we introduce a 3D upper body tracker that cap-

*This research has been supported by funding from the EC projects IURO and RADHAR and the SNF project NCCR IM2.

tures the motion of the human agent automatically and in real-time. As input data, we rely on depth streams which are captured by a low-resolution depth sensor. Such sensors are recently becoming widely available and inexpensive. Based on the extracted motion of the agent, we temporally segment the data, extract the manipulated objects, and categorize the objects based on the segmented motions. The categorization is performed in an unsupervised manner.

In this work, we present four main contributions:

- We propose an approach for unsupervised categorization of objects based on depth data streams and extracted motion capture data.
- To capture the agent, we propose a novel depth-based approach for real-time pose tracking that combines the benefits of body part detection and efficient skeleton-based pose estimation. In contrast to previous work [12], our approach handles occlusions, which is essential for observing object manipulations.
- For functional categorization, we introduce a novel similarity measure for human motions. To this end, we extract a set of key poses and transform each motion segment into a string of key poses. The human actions are then compared using a modified Levenshtein distance [18] that takes the distance of key poses into account. This measure is more robust to variations among subjects than classical dynamic time warping approaches [25] applied to motion data directly.
- We evaluate the approach for object categorization on a newly recorded dataset that comprises depth and video data of 6 subjects, 13 action classes, and 174 object manipulations.[1]

The concept of categorizing objects based on the human motion observed during object manipulation has several practical advantages. In autonomous robotics, home assistance, or scene understanding, modeling all potential categories a-priori exceeds the capacity of many platforms. In our approach, a-priori knowledge is required only for the human agent in terms of the human motion capture approach. Additional objects are extracted and categorized according to their relevance which is inferred from the captured agent.

2. Related Work

Markerless Motion Capture Recent surveys [19] reveal that markerless motion capture is a very active field of research. Our tracking approach is mostly related to the work of Bregler *et al*. [5] where the kinematic chain is represented by twists. We also use twists since they can be elegantly linearized for pose estimation. Since the original work [5] relies on local optimization and optical flow as feature, it is prone to tracking errors. To overcome these limitations, a multi-layer approach [11] has been proposed. While the first layer uses a global optimization technique for pose estimation that is related to [6], the second layer refines the silhouette and the pose using local optimization and twists. Although the approach performs very well on the HumanEva benchmark [28], it is not suitable for real-time applications.

Recently a few techniques for pose estimation from time-of-flight (TOF) cameras have been proposed. In [17, 35], variants of the iterative closest point algorithm have been used for upper body estimation. While these works rely on local optimization, which makes them prone to errors, Ganapathi *et al*. [12] propose using body part detectors [23] for full body motion capture. The detectors make the approach robust to local minima but it is assumed that the person is not occluded. Since the algorithm is implemented on a GPU, framerates around 5 frames per second are achieved. Our approach combines local optimization with twists [5] and body parts detectors. In contrast to [12], a triangulation of the surface is not needed and the detections can be sparse. This is very important in the context of object manipulation where body parts like hands are frequently occluded.

Another important point is that the use of key poses has already been suggested in the human tracking literature [9, 26]. They have mostly been used to improve and initialize the tracking algorithm, while in our case they are adopted as action descriptors. A related idea has also been presented in [32], where 3D exemplars have been used to generate 2D sequences used for supervised action recognition.

Functional Similarity Functional similarity has been already proposed in the 90's for object recognition [29, 27], where objects are modeled in terms of functional parts. While [29] associates functionality with specific shape primitives, recent approaches extract features that are relevant for functionality from video data.

When the video sequences are already labeled with the object class and the motion class that are involved in object manipulation, motion and appearance cues can be combined to improve object and action recognition [20, 14, 8]. Differently than in our case, in [16] it is assumed that objects and actions of interests are already categorized, whereas the relations between the two types of categories are unknown. The relations are inferred from video data and represented as pairs between action and object classes like "drink-cup" or "drink-glass". The learned relations can then be used for object and action recognition.

There are only few works that have addressed ob-

[1]The dataset is publicly available at http://www.vision.ee.ethz.ch/~gallju/projects/dyncat/dyncat.html.

(a) Detections (b) Depth discontinuities (c) Human contours (d) Correspondences (e) Estimated Pose

Figure 2. Pose estimation from depth data. (a) Depth image with detections. While the face *(blue)* can be reliably detected, the hand detections *(red)* are sparse due to occlusions and previously unobserved object contact. (b) Depth discontinuities extracted from depth data. (c) Contours of the model. The contours and the depth discontinuities are matched to obtain correspondences. (d) Correspondences obtained from depth matching along projection rays *(yellow)*, contour matching *(green)*, and the body part detectors *(red)*. (e) Estimated pose overlaid on the intensity image.

ject clustering based on functional similarity. In [21], appearance-based categorization is applied to video data where tracked feature points are segmented and used as features for categorization. While this approach categorizes moving objects like a car or a tram directly from observed motion patterns, the works described in [22, 30, 1, 33] are more in the spirit of categorization based on agent-produced motions. In [22], human trajectories in an office environment are used to segment the camera views into regions where similar human behaviors have been observed. This concept has been extended to street scenes to categorize and label elements like roads or sidewalks that are very similar in appearance [30]. In [1], a rule-based approach is proposed to extract scene graphs that represent spatio-temporal correlations between objects. It is assumed that all the relevant objects can be segmented and the scene graphs model whether the regions are visible, connected, or occlude each other. This approach, however, does not generalize to real-world data since it takes any segmented region into account and does not distinguish between relevant and irrelevant regions. In [33], activities are inferred from the used objects that are observed and identified by video and RFID sensors.

3. Pose Estimation

Pose estimation is performed on depth data (see Fig. 2). In our setup, we acquire the data with a low-resolution depth sensor. Such sensors are becoming widely available and are already part of consumer products like video game consoles. However, any source of depth data, *e.g.*, acquired by a stereo setup, could be used as well. For tracking, we rely on a skeletal model of the human body with 10 degrees-of-freedom for the joints and 6 additional parameters for the rotation and translation of the torso. The parameters are denoted by Θ. The skeleton is surrounded by a 3D triangle surface mesh that is generated from a statistical body model [15]. To this end, we use the rough height ($\pm$5cm) and the gender of the person to morph the model. Currently, gender and height are provided, but it would be also feasible to estimate such parameters directly from the depth data, *e.g.*, as in [2]. Finally, skinning weights w_{k_i} are computed [3] that specify the influence of a bone k on a vertex V_i, *i.e.*, a mesh transformation is obtained by $V_i' = \sum_k w_{k_i} T_k(\Theta) V_i$, where $T_k(\Theta)$ is the transformation matrix for bone k obtained from the pose parameters Θ.

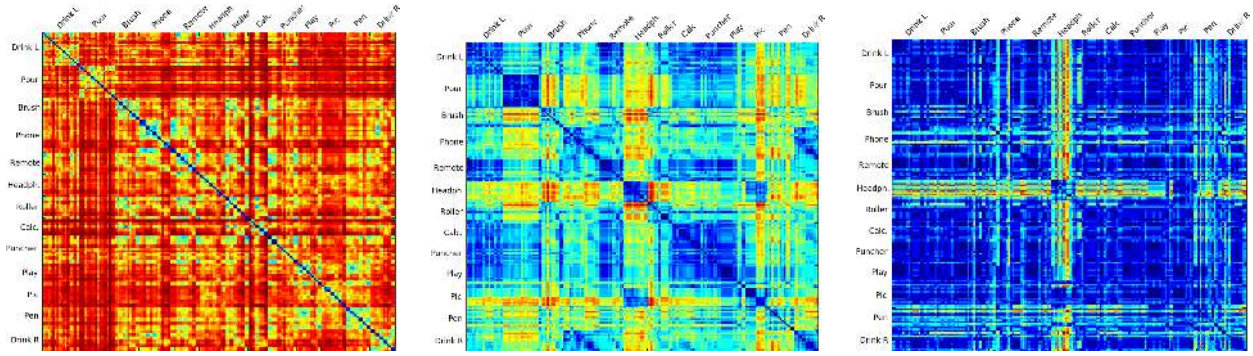
Since the camera is calibrated, each depth value can be expressed as a 3D point X. Having the vertices of the model V_i associated to some 3D point X_i, we can solve for the human pose using the twist representation $\exp(\hat{\theta\xi}) \approx I + \hat{\theta\xi}$ for the transformations $T(\Theta)$ [5] by minimizing

$$\frac{1}{2} \sum_i \left\| \left(I + \hat{\theta\xi} + \sum_{j=1}^{n_{k_i}} \theta_j \hat{\xi}_j \right) V_i - X_i \right\|_2^2 , \qquad (1)$$

where the vertex V_i on limb k_i is influenced by n_{k_i} joints according to the kinematic chain. Due to the linearization of the twists, *i.e.*, $\exp(\hat{\theta\xi}) \approx I + \hat{\theta\xi}$, we can efficiently optimize over all pose parameters.

Correspondences (X_i, V_i) are established by searching for the closest point of each visible vertex V_i in the depth image. This can be done very efficiently by following the projection ray of V_i. When a depth value z is on the ray, we take the closest 3D point X among all the depth values within a 12x12-pixel neighborhood of z. This only matches the model to the data, but the data should also explain the model. Hence, we match the edge pixels extracted from the depth image with the edge pixels extracted from the projected surface. This can be efficiently performed by computing a distance field in the image domain. For each edge pixel that matches a projected vertex, we get a correspondence. Due to occlusions and to the presence of objects, the matching can lead to correspondences with wrong depth values. To reduce wrong correspondences, we reject them when the Euclidean distance or the depth distance between the two 3D points is larger than 200 mm or 50 mm, respectively. The extraction of correspondences is shown in Fig. 2(b-d).

Since local optimization is prone to errors, we integrate detectors for the head and the hands; see Fig. 2(a). To this end, we trained two object detectors [10], one for hands and one for heads. As features, we use the raw depth data and



(a) (b) (c)

Figure 3. Similarity matrix for all the actions in our training dataset, ordered by action class (low values are red). (a) Using the dynamic time alignment proposed in the ACA algorithm does not give good results for similar actions performed by different subjects. (b) The key pose based representation gives a good clustered representation of all the actions in the dataset, which cannot be achieved if consecutive, identical key poses are not merged together (c).

the intensity image. For training, we captured around 2500 examples with a depth camera. While heads can be reliably detected, hands are more difficult to detect and very often occluded during object manipulation. Having a detection of a body part in the depth image, we establish correspondences between the vertices of the body part and the detection. Since our method is mainly driven by the local optimization, temporal sparse detections are sufficient to recover from tracking errors. Hence, we set very high detection thresholds, namely 2.5 for the hands and 2 for the face, to achieve low recall and high precision for the detectors.

For initialization, we use the detected head for an initial estimate of the global translation and rotation. We then estimate the torso using Eq. (1), and the full pose in a second step. After initialization, the pose parameters Θ_t are estimated for each frame where the previously estimated parameters Θ_{t-1} are used to initialize the optimization. For estimation, we iterate the two steps of computing correspondences and optimization using Eq. (1) several times. In our experiments, we used 10 iterations.

4. Functional Categorization

Dataset Using the outcome of the tracking algorithm, we have built a dataset. This dataset includes 6 subjects who perform a set of actions using several objects of different appearances and functionalities. The possible actions are "Pour liquid in a cup", "Drink with the left hand", "Drink with the right hand", "Use a brush", "Use a remote control", "Use a roller", "Use a puncher", "Use a calculator", "Make a phone call", "Wear headphones", "Play with a videogame", "Take a picture", "Use a pen". We asked the subjects to perform about 30 actions, therefore some actions are repeated using objects with the same functionality but with a different appearance. Then each action has been manually labeled with an action label, to provide an evaluation testbed for the clustering and classification phases.

Temporal Segmentation The collected sequences are then initially segmented using Aligned Cluster Analysis (ACA) [34], a very powerful technique for segmenting motion capture data. It allows to decompose an arbitrary motion capture stream by a single subject into a set of disjointed segments, each of which is corresponding to one out of a set of possible actions, in a semi-supervised way. In fact only few parameters, like the total number of actions k and their length range need to be provided. We chose to adopt this algorithm because it has a couple of very useful properties: 1) It works even if actions have an arbitrary and not pre-defined length and 2) it is robust to noise and to speed variation in the actions. Thanks to this pre-processing step, all the training data are split into single actions, which will form the basis of our functional categorization algorithm.

Categorization Although ACA works very well for splitting actions performed by a single subject, its similarity measure based on dynamical time alignment is not powerful enough to evaluate the similarity between actions performed by different subjects. This can be better evaluated by analyzing Fig. 3(a), where we show the similarity matrix computed on all the actions in our dataset, performed by 6 different subjects and ordered by class. Hence, we propose an algorithm that can better handle this situation in order to model actions in a subject-independent way, which is a key component to achieve an unsupervised action clustering.

To this end, we studied an action descriptor which can cope very well with variations among subjects and allows to compute similarities between different segments in a fast and principled way. What we propose is to cluster all the input poses, which once concatenated build the different actions, into a set of N *key poses*. To do this, we adopt the K-means algorithm, using as distance between poses the Euclidean distance of the normalized direction vectors of the limbs. In our experiments, the vector representation performed slightly better than twists or joint angles.

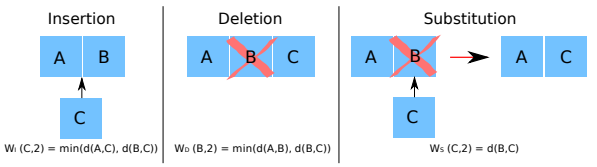


Figure 4. Illustration of the insertion, deletion, and substitution operation weights W_{I}, W_{D}, and W_{S}.

Each pose in the sequence is then substituted by the corresponding key pose, which in our case is the mean of the cluster which the pose belongs to. To have a short representation which is still consistent with the action but independent of its duration (*e.g.*, a phone call can last 10 seconds or 5 minutes, but anyway should belong to the same class), we merge all the consecutive poses which are represented by the same key pose into a single one. Also this choice can be better motivated by analyzing the similarity matrices shown in Fig. 3(b,c). The matrices clearly show that merging consecutive key poses helps in achieving a better distinction between different actions.

An action is therefore represented as an L-dimensional vector A of key poses, where $1 \leq L < \infty$ and $A_i \neq A_{i+1}$ for all $1 \leq i \leq L-1$. We can now introduce the concept of *distance* between actions, which we formulate as a variation of the Levenshtein distance [18]. The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character. What we propose in our case is to consider actions as strings of key poses and give a weight W_{I}, W_{D}, or W_{S} to the operations depending on the distance between the key poses that they involve. We define then:

$$W_{\mathrm{I}}(P,i) = \begin{cases} d(P,A_1) & \text{if } i=1, \\ \min(d(P,A_{i-1}), d(P,A_i)) & \text{if } 2 \leq i \leq L, \\ d(P,A_L) & \text{if } i = L+1 \end{cases}$$

$$W_{\mathrm{D}}(P,i) = \begin{cases} d(P,A_2) & \text{if } i=1, \\ \min(d(P,A_{i-1}), d(P,A_{i+1})) & \text{if } 2 \leq i < L, \\ d(P,A_{L-1}) & \text{if } i = L \end{cases}$$

$$W_{\mathrm{S}}(P,i) = d(P,A_i),$$

where P is the key pose we need to insert, delete, or substitute at position i in the action string A, and d indicates the Euclidean distance between poses. The weights and corresponding operations are illustrated in Fig. 4.

Now that a dissimilarity measure between actions has been defined, it can be used to discover, in an unsupervised way, the data structure. Given the complete dissimilarity matrix that can be computed using our modified Levenshtein distance, clustering the training data becomes straightforward. We chose to adopt the hierarchical agglomerative clustering algorithm, using weighted average linkage, but many other techniques could have been used. The clustering results at this point only depend on the number of key poses that have been adopted and on a threshold, namely the cutoff, that is basically a stopping criterion for the clustering algorithm. An additional implementation choice that we made is to discard all the clusters that contain less than 3 elements, because they would not be descriptive enough. In case of our training data, we know the *true* data structure and we can use it to quantitatively evaluate the results obtained by the clustering algorithm (Sec. 6).

5. Object Localization

After temporal segmentation and clustering of the data, we use the estimated pose to localize the object that is manipulated within a segment. To this end, we evaluate the variance of the hand positions as trajectories in the 3D space and assume that the hand with the highest variation manipulates the object. We mask then all depth values that are within a distance of 250 mm of the active hand and not part of the human. After filtering the mask, we extract connected components and compute the bounding box. In order to obtain the object in a rather static state without motion blur, we discard elements with depth variations in a temporal neighborhood. The object localization is directly inferred from the human poses and additional scene knowledge is not used. In our implementation, we take currently only the first 30 frames of each segment into account.

6. Experiments

Setup To collect our dataset, we synchronized and calibrated a TOF camera and a standard RGB one (which is used only for visualization purposes). Data has then been collected by asking the subjects to perform a set of actions using the objects we provided. One important characteristic of our set of objects is that it contains objects with similar appearance and different functionality, *e.g.*, cell phone and videogame, and objects with different appearance and very similar functionality, *e.g.*, cell phone and landline phone. The set of possible actions has already been described in Sec. 4.

Tracking For evaluating the markerless motion capture approach proposed in Sec. 3, we have annotated the head and the hands for 2 sequences of our recorded dataset. The 3D ground truth is obtained by manually annotating every 10^{th} depth image. For a sequence without occlusions (2113 frames), we get an error of 84.3 ± 9.0 mm. A few frames of the sequence are shown in Fig. 5. For a sequence with occlusions (462 frames), the error is 85.8 ± 7.9 mm. The

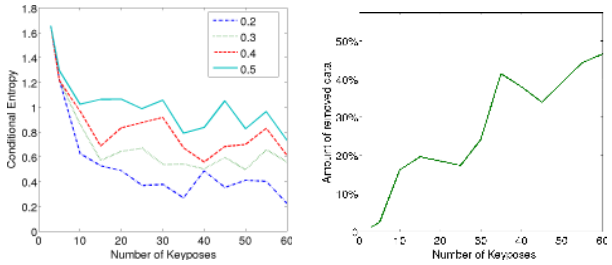
current implementation runs at 12 fps[2]. It requires 25 msec. for the face and hand detections, while the optimization including computing correspondences takes 60 msec. Higher frame rates can be achieved by parallelizing the detectors and speeding-up the closest point search for optimization.

Categorization As explained in Sec. 4, we cluster the training data using a hierarchical agglomerative clustering algorithm based on our modified Levenshtein distance. This approach depends on two parameters: The number of key poses used to describe the input sequences has an influence on the similarity matrix and therefore on the clustering. A lower number of key poses will increase the similarity between different actions and therefore generate few large clusters. On the other hand, a larger number of key poses will differentiate more the activities and bias the algorithm towards many small clusters. Another parameter is the cutoff threshold of the clustering algorithm, which indicates until which level of the hierarchy smaller clusters should be merged, and therefore also has an impact on the cluster sizes. To evaluate the effects of these parameters and the quality of the resulting clustering, we computed the conditional entropy of the outcome when varying the number of key poses and the cutoff threshold, as shown in Fig. 6(a). To compute the entropy value, we used the manual labeling of the training dataset, so that it basically measures how much uncertainty remains in the true class given the estimated clusters.

Obviously, we cannot use such measure for setting the parameters of our approach, since this would imply knowing the true class labels. Instead, we impose a fixed cutoff threshold and state that not more than 20% of the input data should have been removed. The amount of removed sequences depends on the clustering fragmentation, because all the sequences belonging to clusters made of less than 3 elements are not considered. The number of removed sequences can be better evaluated in Fig. 6(b), and leads us to choose $K = 30$ key poses for all our experiments.

Classification The categorization obtained on the training data can also be used as a basis to perform action classification experiments. We have developed such classification experiments in two different setups: In the first, the subjects were asked to execute some actions chosen among the ones that built the dataset, *without* physically using the objects. In the second setup, we performed leave-one-out cross validation where we used 1 subject for testing and the other 5 for training. The classification score depends on the relative frequency of a certain action class within a cluster, which we denote by $p(a|c)$, normalized by its maximal value among all the clusters. More formally, we define the

[2]CPU: Intel Core2Quad 2.83GHz (single thread); Graphics Card: NVIDIA GeForce 9800 GT.



(a) (b)

Figure 6. (a) Conditional entropy of the obtained clustering depending on the number of key poses. Results are shown for different cutoff values, depicted in different line styles. (b) Number of sequences removed from the training data depending on the adopted number of key poses. A larger number of key poses generates more smaller clusters which are removed if composed by less than 3 elements.

score S of the classification of an action a to a cluster c as

$$S(c|a) = \frac{p(a|c)}{\max_{c_i \in C} p(a|c_i)}, \tag{2}$$

where C is the set of all the clusters.

To classify a new sequence, we compute the average modified Levenshtein distance from the sequence to all the elements in each cluster, and then choose the cluster c for which this average distance is lowest. Then, knowing from the manual labeling the true class a of this sequence and of the actions belonging to c, we can compute our classification score $S(c|a)$ as described by Eq. (2). The experimental results are given in Tables 1 and 2. It is interesting to note that the classification results for the leave-one-out experiment are only slightly better than the ones obtained in the testing sequences in which the objects were *not* used. The same experiments have been carried on using ACA [34], and the obtained recognition rates averaged over all the subjects and all the actions are 16.2% for sequences without physical objects and 9.7% for sequences with objects.

Object Localization and Categorization Finally, we evaluate the object localization algorithm described in Sec. 5. To this end, we annotated the manipulated objects by a bounding box in the first frame of each action segment of the dataset. We denote the object as correctly localized when the ratio of intersection over union is greater than 0.5. For our dataset, we obtained 75.6% for recall and 83.9% for precision. By merging the outcome of the localization step and the action clustering approach and by assigning each object to the corresponding action, we obtain our unsupervised object categorization. An overview of a subset of the extracted and categorized objects is given in Fig. 7.

7. Conclusions

In this work, we have proposed an approach that automatically extracts objects from depth data streams and cat-

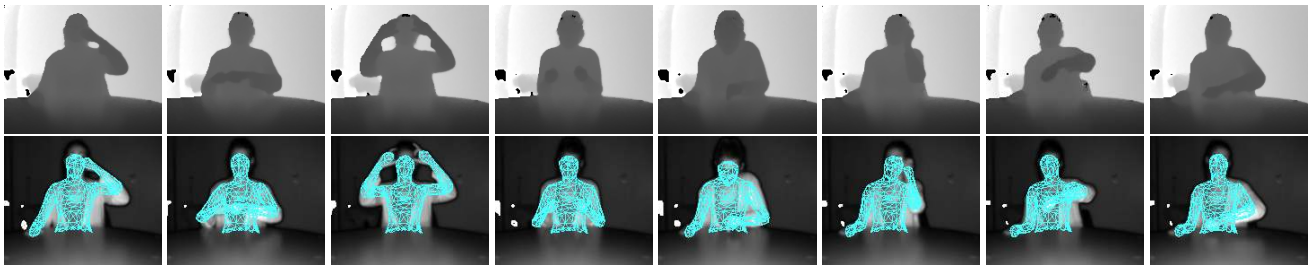


Figure 5. Depth images and estimated poses projected onto the intensity images.

Subj	Pour	Drink R	Brush	Phone	Remote	Headph.	Roller	Calc.	Puncher	Play	Pic	Pen	Drink L	Avg
1	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	0.86	0.00	0.00	-	0.78
2	1.00	-	0.00	1.00	0.50	1.00	0.67	1.00	0.00	0.29	1.00	0.86	0.57	0.66
3	1.00	-	0.50	1.00	0.00	1.00	1.00	0.00	0.20	0.86	0.50	0.00	1.00	0.59
4	0.00	-	1.00	1.00	0.50	1.00	1.00	0.00	0.27	0.86	1.00	1.00	1.00	0.72
5	0.50	-	0.50	0.83	0.50	1.00	0.00	1.00	0.17	1.00	0.50	0.86	1.00	0.66
6	0.17	1.00	1.00	0.43	0.50	0.00	0.67	1.00	1.00	0.29	1.00	0.00	-	0.59
Avg	0.61	1.00	0.58	0.88	0.50	0.83	0.72	0.67	0.44	0.69	0.67	0.45	0.89	0.69

Table 1. Action classification results on testing sequences in which subjects were not using the physical objects. The maximum score of 1 is obtained if an action is assigned to the cluster in which the frequency of that action is the highest among all the clusters. The lowest score of 0 is obtained when an action is assigned to a cluster in which that action is not represented.

egorizes them according to their functionality in an unsupervised manner. The functionality is inferred from the captured human motion observed during object manipulation. Our experiments have shown that the categories obtained by our method have a semantic interpretation. Our current approach is limited by the detail of motion that is captured. For instance, functionalities that differ in subtle hand motions cannot be extracted from the low-resolution depth data. However, this is not a principled limitation of our approach, which can also be applied to high resolution color data. In general, we regard functionality as a complementary cue to appearance for unsupervised object categorization. The obtained functional categories can be further processed to obtain finer categories based on appearance or to infer the relation between functionality and appearance.

References

[1] E. Aksoy, A. Abramov, F. Wörgötter, and B. Dellen. Categorizing object-action relations from semantic scene graphs. In *Proc. Int. Conf. on Rob. and Automation*, 2010.

[2] A. Balan and M. Black. The naked truth: Estimating body shape under clothing. In *Proc. ECCV*, pages 15–29, 2008.

[3] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.

[4] A. Booth. The facilitative effect of agent-produced motions on categorization in infancy. *Infant Behavior and Development*, 23(2):153 – 174, 2000.

[5] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *IJCV*, 56(3):179–194, 2004.

[6] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *IJCV*, 61(2):185–205, 2005.

[7] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010.

[8] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. *Proc. CVPR*, 2008.

[9] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. From canonical poses to 3-d motion capture using a single camera. *PAMI*, 32(7):1165–1181, 2010.

[10] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *Proc. CVPR*, 2009.

[11] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture – a multi-layer framework. *IJCV*, 87(1):75–92, 2010.

[12] V. Ganapathi, C. Plagemann, S. Thrun, and D. Koller. Real time motion capture using a single time-of-flight camera. In *Proc. CVPR*, 2010.

[13] J. Gibson. *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin Company, 1979.

[14] A. Gupta and L. Davis. Objects in action: An approach for combining action understanding and object perception. In *Proc. CVPR*, 2007.

[15] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 2(28), 2009.

[16] H. Kjellström, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *CVIU*, 2010.

[17] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *Proc. Int. Conf. on Rob. and Automation*, 2006.

[18] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 1966.

[19] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2):90–126, 2006.

[20] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *Proc. ICCV*, 1999.

Subj	Pour	Drink R	Brush	Phone	Remote	Headph.	Roller	Calc.	Puncher	Play	Pic	Pen	Drink L	Avg
1	0.56	1.00	0.67	1.00	1.00	1.00	1.00	1.00	1.00	0.60	1.00	0.92	0.00	0.83
2	0.30	1.00	1.00	1.00	0.25	0.00	0.50	1.00	0.50	1.00	1.00	0.00	1.00	0.66
3	0.64	1.00	0.50	1.00	0.50	1.00	1.00	1.00	0.23	0.60	1.00	1.00	1.00	0.81
4	0.50	1.00	0.21	0.75	0.50	1.00	0.00	1.00	0.62	0.50	0.50	0.86	0.25	0.59
5	0.01	1.00	0.50	0.21	0.50	1.00	1.00	1.00	0.00	0.94	1.00	0.50	1.00	0.67
6	0.80	1.00	0.93	1.00	0.50	1.00	0.50	1.00	1.00	0.00	1.00	0.86	1.00	0.81
Avg	0.47	1.00	0.63	0.83	0.54	0.83	0.67	1.00	0.56	0.61	0.92	0.69	0.71	0.73

Table 2. Action classification results on training sequences in which subjects were using the physical objects. If a subject has performed the same actions several times, an average of the classification score is given for that action. Before computing these results, each subject has been removed from the training dataset.

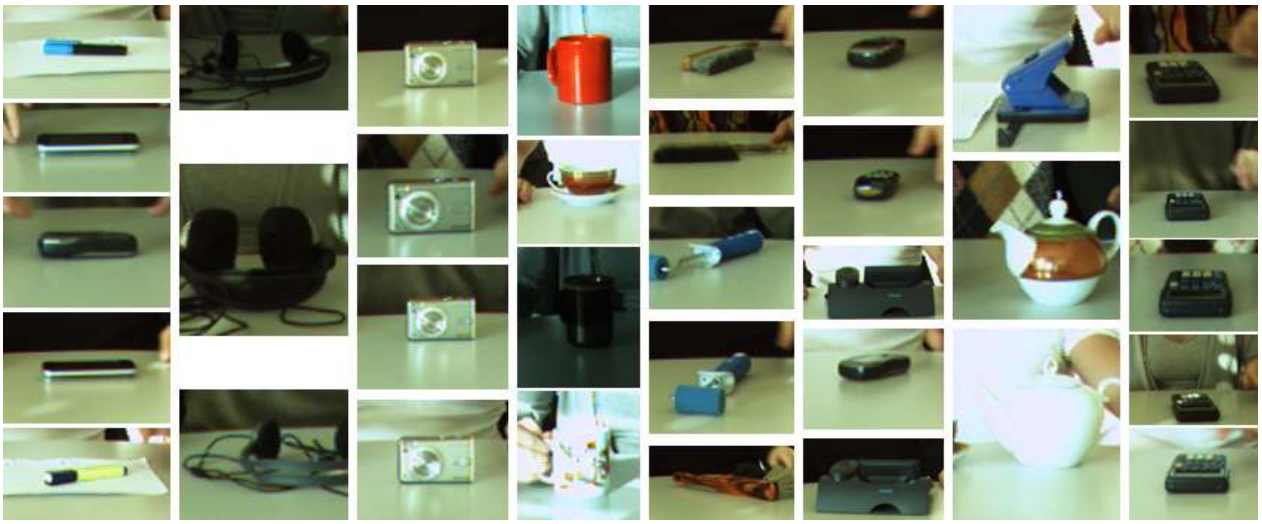


Figure 7. A subset of objects that have been extracted from the dataset and categorized according to the human motion observed during object manipulation. Each column shows a representative set of objects that are in the same class. The objects of classes like headphones, camera, or calculator are similar in appearance and functionality. The cups have a relatively high intra-class variation with respect to appearance, the variation is even higher for the telephones. Both classes are well categorized by our approach since their functionality involves similar motions. The roller and the brushes have been assigned to the same class. Although they are different in appearance and functionality, the observed movements in the dataset are very similar. The same applies to the pen and the videogame. Note that the "videogame" objects are basically smart phones that have been used for gaming instead of calling.

[21] B. Ommer, T. Mader, and J. Buhmann. Seeing the Objects Behind the Dots: Recognition in Videos from a Moving Camera. *IJCV*, 83:57–71, 2009.

[22] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Proc. ICCV*, 2005.

[23] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Proc. Int. Conf. on Rob. and Automation*, 2010.

[24] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28:976–990, 2010.

[25] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[26] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. CVPR*, pages 271–278, 2005.

[27] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Recognition by functional parts. *CVIU*, 62:164–176, 1995.

[28] L. Sigal, A. Balan, and M. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87:4–27, 2010.

[29] L. Stark and K. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *PAMI*, 13:1097–1104, 1991.

[30] M. W. Turek, A. Hoogs, and R. Collins. Unsupervised learning of functional categories in video scenes. In *Proc. ECCV*, 2010.

[31] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 88:284–302, 2010.

[32] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Proc. ICCV*, 2007.

[33] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *Proc. ICCV*, 2007.

[34] F. Zhou, F. De la Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Conf. on Aut. Face and Gestures Recognition*, 2008.

[35] Y. Zhu and B. Dariush. Constrained optimization for human pose estimation from depth sequences. In *Proc. ACCV*, 2007.