Detecting Subtle Human-Object Interactions Using Kinect

Sebastian Ubalde[1], Zicheng Liu[2], and Marta Mejail[1]

[1] Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina
[2] Microsoft Research, Redmond, United States

Abstract. We present a method to identify human-object interactions involved in complex, fine-grained activities. Our approach benefits from recent improvements in range sensor technology and body trackers to detect and classify important events in a depth video. Combining global motion information with local video analysis, our method is able to recognize the time instants of a video at which a person picks up or puts down an object. We introduce three novel datasets for evaluation and perform extensive experiments with promising results.

Keywords: Human-object interaction, depth sensor, trajectory analysis.

1 Introduction

Detecting and identifying human-object interactions is crucial for many computer vision applications, such as video surveillance, assisted living, children monitoring and behavior understanding. This problem requires not only a proper analysis of human motion but also an accurate study of its effects on objects.

Our work aims at automatically detecting (both in time and space) points of a video at which a person picks up or puts down an object. We focus on rather small objects, the kind that can be usually found on a desk or a table. Modeling and tracking human pose is a essential for this task.

With the arrival of low cost range sensors such as Microsoft Kinect, fast and accurate pose detection has become possible. Compared with conventional visual systems, depth maps produced by range devices reliably describe the shape and geometry of objects. Furthermore, they are insensitive to shadows and lighting.

Pose information provided by the Kinect has proved robust enough for video recognition tasks involving relatively coarse, easily distinguishable movements [1–5]. This kind of movements usually result in sharp variations in motion properties, making it fairly simple to extract meaningful information from the video as a whole. However, the task faced in this paper can not rely on global motion features alone. The action of picking up an object from a table may involve roughly the same movements as many other not-picking up actions. At the same time, variability in motion characteristics within the same action can be large. When dealing with this kind of scenario, features describing the entire body motion

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 770–777, 2014.

and environment are not discriminative enough. Instead, it becomes necessary to focus on more subtle image variations, as the ones produced by changes in object position.

Our method takes advantage of global motion information to estimate interesting points. Then, a local approach is used. The spatio-temporal neighborhoods around the detected points are analyzed, searching for visual evidence of picking up (putting down) events. Successfully identifying such subtle evidence is challenging: It is known that the accuracy of structured-light devices reduces with the inverse of the depth [6], making measurements for distant objects less reliable. Furthermore, depth maps contain a significant number of artifacts, like undefined and noisy pixels [7].

As far as we are aware, there has not been much research along the line of our work. One related approach is by Gupta et al. [8, 9]. Their method takes object manipulation into account, but considers the pick up (put down) detection problem as part of a more general task. Furthermore, they work with traditional RGB videos, so the overall performance of their method is naturally limited. The paper of Packer et al. [10] uses depth sequences in combination with RGB videos, making it easier to detect the instant at which an object location changes. However, their method requires a calibration phase for the purpose of background subtraction. More importantly, just like Gupta et al., they use the pick up (put down) detection method as part of a bigger framework, without trying to improve it.

Due to the lack of known benchmarks for the problem, we recorded three datasets to evaluate our approach, consisting of approximately 180 videos, which we make publicly available for download [11].

Our main contributions are: To the best of our knowledge, we are the first to use a depth based approach to detect subtle picking up (putting down) events involving small objects. We propose a method to detect interesting points and a descriptor to encode depth information around those points. Further, we present three novel datasets which we use to extensively test our approach. The remainder of the paper is organized as follows. Section 2 describes our method. Section 3 presents the experimental results on the recorded datasets. Finally, Section 4 concludes the paper.

2 Detecting Human-Object Interaction

2.1 Kinect Data

The Microsoft Kinect is a structured-light device that calculates depth images using an infrared projector and an infrared camera. Images are computed at 320 x 240 resolution and at 30 frames per second.

Based on the raw depth information, the Kinect provides a high-level abstraction describing the image content. A human-pose tracker built on top of the work of Shotton et al. [12] labels each pixel as being either part of the human body, the background, or unknown, and predicts the 3D position of several body joints (hand, wrist, elbow, etc.).

We represent a depth video as a matrix V, where $V(r, c, f)$ is the depth value of the pixel located at row r and column c for frame f. We use a matrix V_l to store background/actor/unexplained labels for each pixel and frame. Further, we model the trajectory described by a specific joint j using a vector-valued function $\gamma_j : \mathbb{N} \to \mathbb{N}^3$, where $\gamma_j(f)$ is the 3D position of joint j at frame f.

2.2 Detecting Interesting Points

We follow a two step procedure that lets us focus on the important sections of the video. First, we use the trajectory of the hand to detect interesting spatio-temporal points. Sudden changes in hand speed, direction and acceleration are closely related to picking up (putting down) events, so we look for points that present these characteristics. Then, we extract a pair of short, local videos from the depth sequence at the estimated points and analyze them.

Specifically, we associate interesting points with local maxima in the curvature of the hand trajectory. Given the hand trajectory $\gamma_h : \mathbb{N} \to \mathbb{N}^3$, we compute the curvature of the hand trajectory, κ, at frame f as:

$$\kappa(f) = \frac{\|\gamma_h'(f) \times \gamma_h''(f)\|}{\|\gamma_h'(f)\|^3}, \tag{1}$$

where γ_h' and γ_h'' represent the first and second derivatives of γ_h respectively, and $\times$ is the cross product.

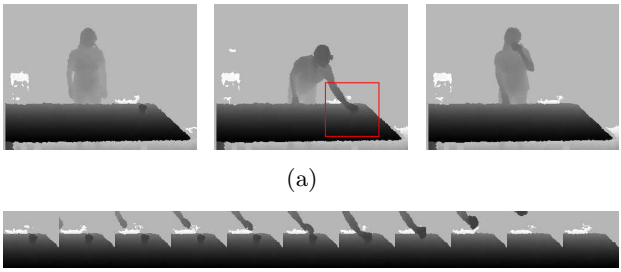
Previous to curvature computation, we smooth the trajectory using an anisotropic diffusion procedure [13]. This is an important step because the positions computed by the body tracker are usually very noisy. Anisotropic diffusion eliminates most of the noise while preserving important changes in the trajectory.

Local maxima in κ represent important changes in the motion properties [14]. As such, they are ideal candidates to capture picking up (putting down) events. Of course, many other events may result in local maxima. Nevertheless, focusing on the neighborhoods around these points dramatically reduces the search space.

2.3 Analyzing Interesting Neighborhoods

Given a frame f corresponding to a local maxima, the spatial position of the hand at f is given by $p = \gamma(f) = (x(f), y(f), z(f))$. As Kinect provides the intrinsic parameters of the depth camera, we can map p to a pixel (r, c) in frame f of the depth video V. We identify the kind of event that triggered the interesting point by analyzing two short, local spatio-temporal patches (i.e. sub matrices) extracted near (r, c, f) from V.

Fig. 1 (b) shows a short, local video extracted from the depth sequence for a typical picking up event. The segment is shown flattened with respect to time. We can easily distinguish three intervals. A first interval shows the hand approaching the object. A second interval shows the hand grasping the object. Finally, a third interval shows the hand moving away. If we compare the first and last frames of the patch, we would find a difference in the pixels values corresponding to the



(a)

(b)

Fig. 1. Example of a picking up event. (a) The 3D position of the hand is mapped to the depth image. (b) A short, local segment capturing the *picking up* event.

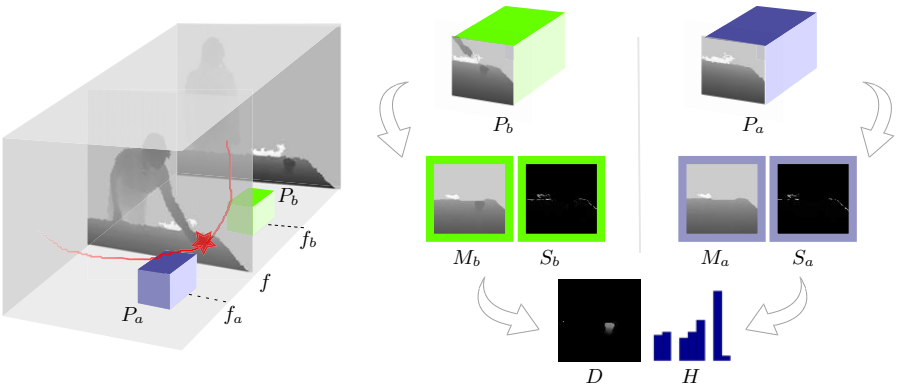


Fig. 2. Method overview

object original position. This is reasonable, because the object is present only in the frames from the first interval. In contrast, for a typical non-picking up (putting down) event, the hand first approaches and then makes contact with the table, just to be moved away after a few frames. No object is picked up (put down) . In this case, the first and last frames would not show significant differences in pixel values. This suggests that comparing frames in this fashion may be a good way to spot a picking up (putting down) event.

Of course, differences between the frames may have other sources, the more drastic of them being the movement of the person. Differences may also be due to noise in the depth image, as well as small random movements in the scene. We use the information provided by the body tracker (i.e. the labels in matrix V_l) to ignore pixels belonging to the person when comparing frames. We follow a frame averaging approach to mitigate noise and small movements, as explained below. Finally, we build a descriptor that encodes frame differences, and use an SVM to classify it as either *pick up (put down)* or not *pick up (put down)*.

The exact procedure followed to compute the descriptor is detailed next. We select two frames f_b and f_a such that $f_b < f < f_a$. We then consider two patches of k frames centered at (r, c, f_b) and (r, c, f_a), named P_b and P_a respectively.

We intend to detect picking up (putting down) events by comparing P_b and P_a. To handle noise and small variations, we collapse P_b into a mean image, M_b, such that:

$$M_b(r,c) = \frac{\sum_{f=1}^{k} P_b(r,c,f)}{k}. \tag{2}$$

Further, we compute a standard deviation image S_b as

$$S_b(r,c) = \sqrt{\frac{\sum_{f=1}^{k} (P_b(r,c,f) - M_b(r,c))^2}{k-1}}. \tag{3}$$

We process P_a in the same way, yielding M_a and S_a. Next, we encode differences between P_b and P_a by computing the image D as follows:

$$D(r,c) = \begin{cases} |M_a(r,c) - M_b(r,c)| \; if \;\; S_b(r,c) < \beta \\ \qquad\qquad\qquad\qquad\quad \wedge \\ \qquad\qquad\qquad\qquad\quad S_a(r,c) < \beta \\ \qquad 0 \qquad\qquad ow \end{cases},$$

where β is a predefined threshold.

Lastly, we build the descriptor H by histogramming D values, and feed H to an SVM for classification.

As the hand may occlude the object (see middle frames in Fig. 1 (b)), we select f_b and f_a based on the hand trajectory. Starting from f, we traverse γ_h, searching for points located far enough from the interesting point p. Specifically, f_b is chosen as

$$f_b = \max_{f_j < f \,\wedge\, \|p - \gamma_h(f_j)\| \geq \alpha} f_j, \tag{4}$$

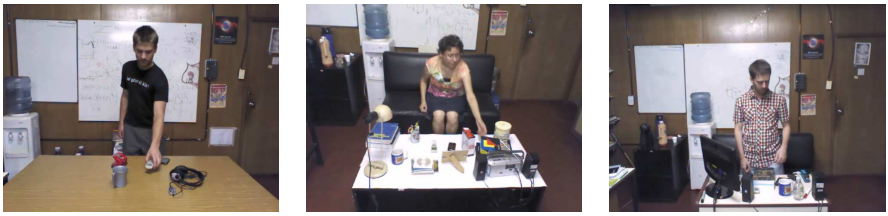
where α is a predefined threshold. That is, f_b is the first frame before f for which the hand is located at a distance of at least α from p. Similarly, f_a is the first frame *after* f for which such condition is met.

Fig. 2 summarizes our method.

3 Experiments

3.1 Datasets

We tested our method on three datasets of actors manipulating objects in different environments. The first dataset shows 11 actors interacting with 5 objects



(a) (b) (c)

Fig. 3. Example frames dataset 1 (a), dataset 2 (b) and dataset 3 (c)

(a cup, a phone, a hole puncher, a pair of headphones and a remote). Each video shows a single actor standing in front of a table. The objects are randomly placed on the table. There are 6 videos for each actor. In the first three videos, the actor interacts with each object in turn. He chooses an object, picks it up, manipulates it in some way and finally puts it down on the table. This is repeated for every object on the table. The last three videos are similar, but the actor does not actually manipulate the object. Instead, he just touches the object and then moves the hand away. Fig. 3 (a) shows an example frame.

Only the manipulated objects are present in the first dataset. This results in a rather clean table scenario that simplifies the detection task. In contrast, the second and third datasets pose more challenging environments. Objects are placed on highly cluttered surfaces. This causes a lot of of artifacts in the depth images. Furthermore, heavy occlusion is commonly found in the depth images.

The second and third datasets are organized in much the same way as the first one. Ten actors interact with 6 objects placed on a coffee table (dataset 2) or a desktop (dataset 3). Fig. 3 (b) and Fig. 3 (c) show two example frames. Note the rather chaotic arrangement of the objects.

3.2 Method Evaluation

We evaluate our interesting point detector as follows. We manually label the frames at which the hand makes or brakes contact with an object, for all the videos in the first dataset. Then, we use our method based on local maxima to automatically select frame numbers. Finally, we compare both procedures: For each manually recorded frame number f on a given video, we consider the closest frame number f' among those selected by our method for that video, and we take the difference $|f - f'|$. The mean difference over all the manually recorded frames is 13.77. Considering that a typical picking up (putting down) action spans approximately 80 frames, this result indicates that it is highly unlikely for our method to miss a picking up (putting down) frame. Further, only 2% of the total number of frames in the dataset were automatically selected, suggesting that our method is also very precise.

Next, we evaluate the proposed descriptor. We perform three separate tests, one for each dataset. We use leave-one-out cross-validation on the actors. That is,

we divide the videos into n sets (n may be 11 or 10 depending on the dataset), each including exactly the videos of one actor. In each of n experiments, we train an SVM using the descriptors extracted from videos of $n-1$ sets and test on the descriptors extracted from videos of the remaining set. Descriptors are extracted at the interesting points detected by our method. For each test, we report confusion matrices and average precision over the n experiments. Results are summarized in tables 1, 2 and 3.

The average accuracy for the first dataset is 96.72%, which is impressive considering that objects as small as phones and remotes were used in the videos.

Table 1. Results for the first dataset

		Actual		
		pick up	not pick up	Total
Predicted	pick up	345	12	357
	not pick up	26	775	801
	Total	371	787	1158

The average accuracy for the second dataset is 93.52%. This is still quite good. Note, however, that recall decreased with respect to the first dataset: 84.93% of the pick up (put down) events where classified as such. In contrast, 93% of the pick up (put down) events were correctly classified for the first dataset. Differences between both tests are reasonable, given the more challenging scenario of the second dataset.

Table 2. Results for the second dataset

		Actual		
		pick up	not pick up	Total
Predicted	pick up	417	16	433
	not pick up	74	882	956
	Total	491	898	1389

The average accuracy for the third dataset is 88.7%, with only 74.34% of the pick up (put down) events correctly classified. The rather sharp decrease in performance with respect to the previous tests is reasonable if we take into account the highly challenging setting of the third dataset.

Table 3. Results for the third dataset

		Actual		
		pick up	not pick up	Total
Predicted	pick up	342	24	366
	not pick up	118	773	891
	Total	460	797	1257

4 Conclusions

We presented a method for identifying human-object interactions from depth sequences. Our approach accurately detects significant events based on the trajectory described by the actor's hand. On relatively clean environments, it achieves impressive accuracy for pick up (put down) recognition. When dealing with highly cluttered scenarios, performance is still promising. In the future, we hope to explore the potential benefits of combining RGB information with depth data for classification. Also, we plan to integrate the proposed method into a general framework for human-object interaction understanding.

References

1. Oreifej, O., Liu, Z.: Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences. In: CVPR 2013, pp. 716–723 (2013)
2. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR 2012, pp. 1290–1297 (2012)
3. Vieira, A., Nascimento, E., Oliveira, G., Liu, Z., Campos, M.: Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 252–259. Springer, Heidelberg (2012)
4. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: CVPR4HB 2010, pp. 9–14 (2010)
5. Sung, J., Ponce, C., Selman, B., Saxena, A.: Human activity detection from RGBD images. In: AAAI workshop on Pattern, Activity and Intent Recognition, PAIR (2011)
6. Mehrotra, S., Zhang, Z., Cai, Q., Zhang, C., Chou, P.A.: Low-complexity, near-lossless coding of depth maps from kinect-like depth cameras. In: MMSP, pp. 1–6. IEEE (2011)
7. Camplani, M., Salgado, L.: Efficient spatio-temporal hole filling strategy for Kinect depth maps. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 8290 (February 2012)
8. Gupta, A., Davis, L.: Objects in action: An approach for combining action understanding and object perception. In: CVPR 2007, pp. 1–8 (2007)
9. Gupta, A., Kembhavi, A., Davis, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. PAMI 31, 1775–1789 (2009)
10. Packer, B., Saenko, K., Koller, D.: A combined pose, object, and feature model for action understanding. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1378–1385 (June 2012)
11. Datasets, `http://www-2.dc.uba.ar/grupinv/imagenes/subalde/`
12. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. In: CACM, vol. 56, pp. 116–124 (January 2013)
13. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. In: CVWS 1987, pp. 16–22 (1987)
14. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. In: IJCV, vol. 50, pp. 203–226 (November 2002)