21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Real-time hand status recognition from RGB-D imagery

Andrew D. Bagdanov, Alberto Del Bimbo, Lorenzo Seidenari, Lorenzo Usai
Media Integration and Communication Center, University of Florence, Italy
{*bagdanov,delbimbo,seidenari*}*@dsi.unifi.it, lorenzo.usai@gmail.com*

Abstract

One of the most critical limitations of Kinect^{TM}-based interfaces is the need for persistence in order to interact with virtual objects. Indeed, a user must keep her arm still for a not-so-short span of time while pointing at an object with which she wishes to interact. The most natural way to overcome this limitation and improve interface reactivity is to employ a vision module able to recognize simple hand poses (e.g. open/closed) in order to add a state to the virtual pointer represented by the user hand. In this paper we propose a method to robustly predict the status of the user hand in real-time. We jointly exploit depth and RGB imagery to produce a robust feature for hand representation. Finally, we use temporal filtering to reduce spurious prediction errors. We have also prepared a dataset of more than 30K depth-RGB image pairs of hands that is being made publicly available. The proposed method achieves more than 98% accuracy and is highly responsive.

1. Introduction

The recent advances in low cost depth sensing cameras such as the Kinect sensor from Microsoft have boosted the capabilities of Natural User Interfaces (NUI). These sensors stream synchronized pairs of RGB-depth (RGB-D) frames. With these sensors the cost and difficulty of some computer vision tasks can be largely alleviated. People and object tracking both benefit from a calibrated depth image by aiding in target segmentation and providing accurate real world coordinates. Most approaches for articulated body tracking in RGB imagery are extremely costly and unstable and therefore are not suitable as drivers for NUI. Moreover in NUI applications there are strict real-time constraints.

Articulated human body tracking is greatly simplified with the use of depth imagery. Recently, a technique to detect human body parts in real-time to provide measurements to a body tracker has been proposed by Shotton et al. [8]. They trained decision forests on features computed with the difference in depth of random displacements around body parts pixels. This technique has been improved by Girshick [3] by combining the features of [8] with a regressor in order to directly predict the body joint position from depth data.

Despite the ability to track a human body in real-time, the implementation of generic user interfaces remains an open problem. An accurate skeletal configuration enables game and application designers to improve the quality of interaction. Users can control a virtual self directly and basic activities can be recognized just by comparing the sequence of body configurations with existing templates. Nevertheless some classic user interaction paradigms are lacking. Classical gesture analysis approaches go a long way towards addressing these needs [1], but access to real-time depth imagery opens new horizons.

Applications that require pointing, clicking, drag-and-drop and all of the gestures that stem from this basic vocabulary are not generally supported by body-driven platforms. The main issue is the difficulty of adding a state to the coordinates of the body without constraining the user position in space. The usual work-around is based on persistence: a user must keep her arm still for a not-so-short span of time while pointing at an object which she wants to interact with. This kind of interaction is usually limited to sections of an application where the user spends less than 5% of her time – like a setup menu or the start screen of an application. We propose to overcome this limitation and improve interface reactivity by employing a vision module able to recognize simple hand poses in order to add a state to the virtual pointer represented by the user hand.

Recognition of hand poses and gestures with depth imagery has been addressed in the past. Oikonomidis et al. developed a tracker for an ar-

ticulated hand model using a particle swarm optimization [6]. Their technique is computationally extremely expensive and requires GPU acceleration. Moreover, their benchmark is conducted on synthetic data and as far we can determine there are no results (even qualitative) that show the system tracking hands at a distance larger than 1 m. Suryanarayan et al. [9] proposed a compressed 3D shape descriptor computed on depth images of hands. They state that the interaction distance is set to 1.5 m. Finally Ren et al. [7] defined a finger Earth Mover's Distance that allows partial matching of segmented hand shapes. Their system works at close distances, but requires the user to wear a black wrist band in order to correctly segment the hand.

2. Method Overview

2.1 Hand extraction and segmentation

The first step in our hand status recognizer is the segmentation of hand regions from the background. Given the real-world center of mass coordinates of a hand $\overline{h}$, we threshold the user map M_u with a value of $D_h + 10$cm where D_h is the depth at the center of mass of the hand. In this way we obtain the hand map M_h. Finally, the binary image M_h is projected in the RGB coordinate system to extract only the pixels of the user hand. M_u is obtained by applying background subtraction to the depth sequence, while $\overline{h}$ can be obtained by averaging the pixel coordinates of the hand body part detected using [8] or performing a double thresholding on the depth image and extracting only pixels $p_h \in \left[\overline{D}_u + \delta, \max(D_u)\right]$, where D_u are the depth pixels of a user and $\overline{D}_u$ is her center of mass. We found $\delta = 15$cm to be a reasonable setting for all users.

2.2 Hand representation

We conducted a preliminary evaluation of local descriptors and spatial configurations on a small dataset consisting of three subjects. We tested the following spatial configurations: a single whole-patch descriptor, a 2×2 grid of descriptors, and a 2×2 grid plus a central location (5 sub-patches in total). For individual descriptors we tested SIFT [5], global Hu moments[4] and the two variations of SURF descriptors [2]: SURF-64 and SURF-128. A dense grid of features is obtained by extracting patches with 50% overlap. The patch size is selected as $\frac{2}{3}r$, where $r = min(w, h)$ and w and h are the width and height of the patch, respectively. The central patch has the same size. Figure 1 shows the extraction process in detail.

Figure 1. Feature extraction procedure

Feat. \ Conf.	1	2×2	$2 \times 2 + 1$
Hu	56.00%	-	-
SIFT	82.97%	-	-
SURF-64	83.63%	86.12%	86.96%
SURF-128	86.92%	86.79%	**87.81%**

Table 1. Features and spatial layout.

In our preliminary evaluation, summarized in Table 1, SIFT and both SURF versions outperformed by far the performance of the Hu moments. This is not surprising, since the Hu moments have less discriminative power and are mainly shape descriptors. Shape can be discriminative only at high resolutions (distance < 1.5 m), while at farther distances shape becomes more ambiguous.

SURF-64 performed slightly worse than SURF-128, but both SURF variations had an edge on SIFT. SIFT descriptors are probably too discriminative and their more detailed representation of the pattern generated more noise. The single patch representation was outperformed by grid configurations, and in particular the $2 \times 2 + 1$ configuration was the one that gave the best results. The final hand representation is computed as the concatenation of the five SURF-128 descriptors for a total of 640 dimensions.

2.3 Hand status recognition

For hand status recognition we train a support vector machine (SVM) with a nonlinear RBF kernel of the form $k(x, y) = \exp(-\gamma||x - y||^2)$. The C regularization parameter and the γ kernel parameter are determined through a 5-fold cross-validation procedure on the training set. The distance of a test sample x to the SVM decision margin is:

$$f(x) = \sum_i \alpha_i y_i k(x_i, x), \qquad (1)$$

where x_i are the support vectors (SVs), y_i are their labels (+1 open, -1 closed), and α_i are the

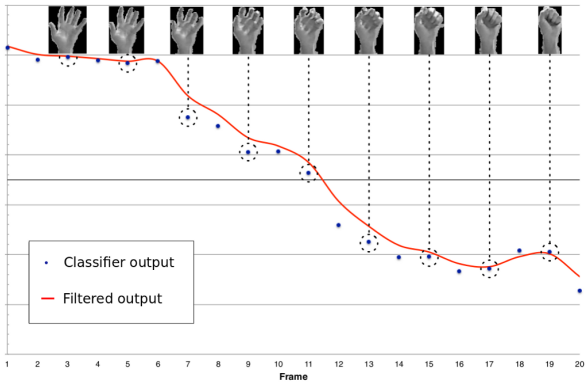


Figure 2. Filtered and unfiltered output.

Figure 3. Pose and orientation variation.

SV weights.

In practice we noticed that the SVM output $f(x)$ is quite noisy (see figure 5 for an example). Consequently, we model the SVM classification function as a noisy measurement process of the true hand state at time k: $f_k(x) = f_{k-1}(x) + \sigma$, where σ is a zero-mean, Gaussian noise term. The SVM output at time k are used to update a Kalman filter that estimates $f_k(x)$. The sign of $f_k(x)$ is then the smoothed predictor of the current hand status. As can be seen in Figure 2 the classifier output smoothly follows the hand status.

3. Experimental Results

Our dataset[1] consists of 18,188 and 15,937 RGB-D pairs of images of, respectively, open and closed hands, from nine different subjects. RGB-D pairs are synchronized and extracted at distances from 1 meter to 3 meters from the sensor. Subjects recorded sequences both with sleeves rolled up and sleeves rolled down in order to avoid bias in the dataset. A sample of processed hands taken from our dataset is shown in Figure 3.

We trained our system on 8 subjects and report the accuracy on one kept out for testing; we trained the system on 31,172 (16,728 open and 14,444 closed) images and tested on the remaining

[1]http://www.micc.unifi.it/datasets/hand-pose

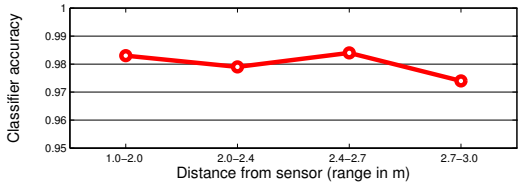


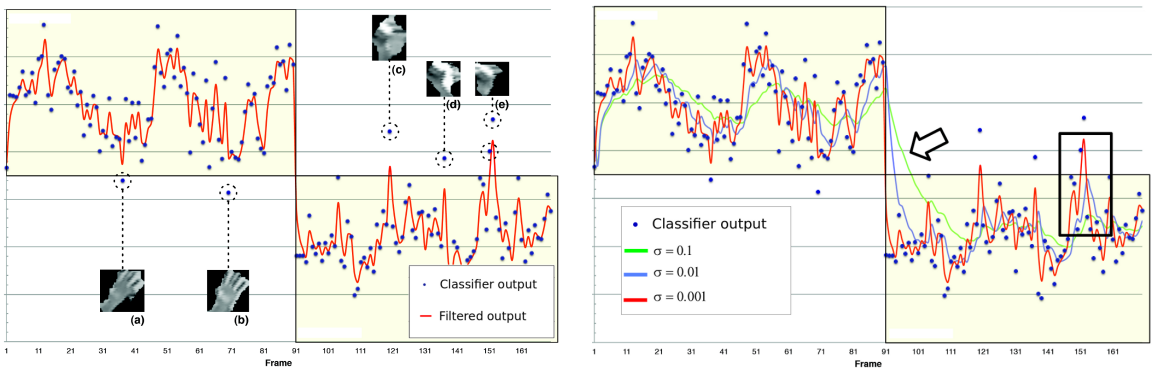
Figure 4. Accuracy at various distances.

2,953. Without the use of temporal smoothing, the system achieves fairly accurate results with an overall accuracy of 96.34%. We improved the preliminary results by almost 10% by enlarging the dataset with around 20k images. Moreover, care was taken in introducing all sources of variability in the data: rotation, scale, hand size and clothing. The use of temporal smoothing increased the accuracy further, raising it to 98.95%. In figure 4 we show the classifier accuracy (after temporal smoothing) as a function of the distance of the subject from the sensor. These results clearly indicate that, within a distance of 3m from the camera, results are extremely stable. Figure 5(a) shows the main sources of classifier errors. These are mainly due to low contrast images like frames (a) and (b) or a wrong segmentation as in frames (c)-(e). All but one of these error are removed by temporal smoothing. The classifier error in frame (e) is recoverable by increasing the filter inertia as can be seen in the black box in 5(b).

Smoothing did not impair system responsiveness; as can be seen in Figure 5(b), the transition of the detector (continuous red line) happens in just two frames and closely follows the transition of the ground truth. Increasing filter inertia (higher σ) can filter some erroneous predictions as seen in the black box, but this comes at the cost of a less prompt transition (see transition region marked by the arrow). Transition time is measured as the number of frames needed for the detector to switch from an open (closed) to a closed (open) state. Transition times are 2, 5 and 10 frames, respectively, for $\sigma = 10^{-3}$, 10^{-2} and 10^{-1}.

Our system runs at around 20 FPS on a 2.8 GHz core i7 CPU using a single core. Most of this computation time is spent in tracking and detection, while feature computation has almost no impact on performance. The use of a nonlinear kernel in the classifier negatively affects the testing time. This issue can be addressed with explicit feature mapping in order to use linear classifiers that do not require the comparison of the test pattern with all support vectors.

In Figure 6 four sample frames of the running



(a) Errors corrected by smoothing (b) Responsiveness vs. accuracy

Figure 5. Effects of temporal smoothing.

Figure 6. Hand status detection: square (open hand), circle (closed hand).

system are shown. The system can handle multiple people, but for clarity of presentation we show frames with a single user. Note how the hand status recognition is invariant with respect to rotation, arm pose, background and distance from the sensor. Videos of the system running are available online[2][3].

4. Conclusions

In this paper we report results of hand status recognition in RGB-D imagery from our dataset where hand images are extracted in the range of 1–3m. Our still effective at farther distances (∼5m), but at such distances the depth-based tracker that we use fails more often. For this reason, we believe that will be difficult to provide continuous hand localization and status recognition at such distances without improving the tracking process (which is not addressed by this work).

[2]http://vimeo.com/38687694
[3]http://vimeo.com/38687794

References

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(9):1685–1699, 2009.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Proc. of ICCV*, 2011.

[4] M. K. Hu. Visual pattern recognition by moment invariants. *Transanctions on Information Theory*, IT-8:179–187, 1962.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[6] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect, in proceedings of the 22nd british machine vision conference. In *Proc. of BMVC*, 2011.

[7] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on fingerearth movers distance with a commodity depth camera. In *Proc. of ACM MM*, 2011.

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *Proc. of CVPR*, 2011.

[9] P. Suryanarayan, A. Subramanian, and D. Mandalapu. Dynamic hand pose recognition using depth data. In *Proc. of ICPR*, 2010.