Real-time Hand Gesture Recognition on Unmodified Wearable Devices

Jie Song, Gábor Sörös, Fabrizio Pece, Otmar Hilliges
Department of Computer Science
ETH Zurich
{jsong, soeroesg, fabrizio.pece, hilliges}@inf.ethz.ch

Abstract

We present a machine learning technique for recognizing discrete gestures and estimating continuous 3D hand position for mobile interaction. Our multi-stage random forest pipeline jointly classifies hand shapes and regresses metric depth of the hand from a single RGB camera. Our technique runs in real time on unmodified mobile devices, such as smartphones, smartwatches, and smartglasses, complementing existing interaction paradigms with in-air gestures.

1. Introduction

Personal wearable computers, such as smartphones, smartwatches and smartglasses, provide ubiquitous access to digital information and play now an important role in our everyday life. Yet, it is unclear what is the easiest and most natural way to interact with such devices, in order to effectively access and consume digital content. Clearly, a major hindrance to seamless interaction is posed by small touch screens and diminutive buttons – the de facto standard for wearable interfaces. Hence, we argue that the current wearable interaction paradigm can be effectively complemented by natural hand gestures around the devices.

Researchers investigated in-air gestural interaction via hardware modifications or dedicated wearable sensors (e.g., IR proximity sensors [5], muscle sensors [7] or external cameras [4]), but such solutions pose a serious limitation to seamless interaction. Previous work has also successfully expanded smartglasses' input capabilities (e.g., [1], [8], [12]), but available solutions do not provide means for gesture sensing and rely on external infrastructure.

The recent emergence of consumer depth cameras has enabled a number of high fidelity, interactive gestural systems and fine-grained 3D hand-pose estimation [3, 6, 11]. The current state of the art can be categorized into methods relying either on model fitting and temporal tracking [6, 11], or on data-driven approaches [3]. In wearable scenarios, however, the usage of conventional depth sensors is prohibitive due to power consumption, heat dissipation and size. Recently Fanello et al. [2] try to overcome these limitations by learning a mapping from color to depth in a camera surrounded with IR illuminants. However, such solution requires hardware modification.

Our contribution builds upon and extends existing research on gesture recognition on mobile devices. Our work, though, differs from existing solutions as it leverages only RGB cameras, lending itself to run on *unmodified* devices. We propose a Random Forest (RF) based algorithm to extend the interaction space around mobile devices by detecting rich gestures performed in front of any wearable camera. The algorithm runs in real time on off-the-shelf mobile devices including resource-constrained smartphones, smartwatches, and smartglasses. In [9] we introduced a data-driven gesture recognition approach that enables mid-air interaction on unmodified portable devices. While the method in [9] is limited to 2D gestures, in [10] we extended this framework to a hybrid classification-regression scheme which is capable of successfully learning a direct mapping from 2D color images to 3D hand positions plus gestures.

2. Method

Our algorithm consists of established image processing steps interwoven with a new, staged classification–regression process. All components have been carefully designed for real-time performance, even on ultra-mobile and resource-constrained devices.

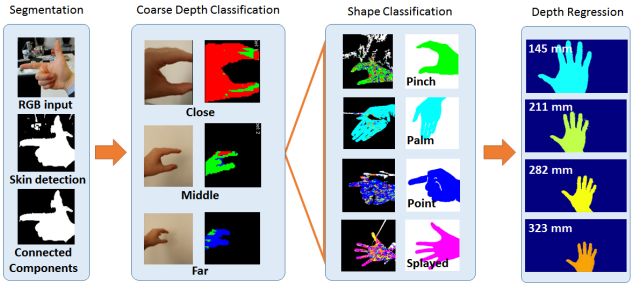

Figure 1. Classification-regression pipeline: 1.) Hand segmentation; 2.) Coarse depth classification; 3.) Hand shape classification; 4.) Fine-grained depth regression (average hand depth)

2.1. Segmentation and Pre-processing

Our method relies on binary masks of the hand, segmented from the background. Currently no method can provide perfect segmentation under arbitrary conditions in real time. Hence, we opted for a simple color thresholding technique which is a good compromise between true positives and false negatives, while it suffers from *false positives* (see Fig. 1). However, the subsequent classification algorithm can be made robust to this type of noise as detailed below.

2.2. RF based Classification and Regression

Our gesture recognizer is based on RFs, which have been successfully used for a number of vision problems, including body pose estimation and hand pose and state estimation form depth cameras. Our method relies only on shape (i.e., binary masks) to infer hand features and states. Within the classification trees we use split criteria based on binary features. Specifically, for each split node we learn shift vectors and a comparison threshold. When evaluating a pixel, we compare the values of the segmented binary mask at these shifted locations and proceed traversing the three based on the result of the comparison. Our binary features allow training of complex data-sets while keeping the computational time and memory footprint low, with no impact on precision. Conceptually, the binary and shift values encode the hand shape information and hence the probability that a given pixel belongs to the hand and to which gesture.

2.3. Multi-stage Recognition

To jointly recognize a rich set of gestures and 3D hand position, one approach is to collect a training database that covers all expected variations and train a single, potentially very deep forest through switching split objectives. While even deep trees can be evaluated in a few milliseconds, the memory footprint of the forest can quickly become an issue, as it grows exponentially with tree depth.

Instead, we opt for a multi-layered forests approach. This is a configuration where expert forests are trained for a particular task and only those images corresponding to a particular class are forwarded to a second forest, trained only on examples from this class. Each of the forests then needs to model less variation and hence can be comparatively shallow. We propose to combine multiple forests that are specialized on different tasks and modify the image before they downstream it. This is effectively a pipeline of independent but inter-related classifiers, as shown in Fig. 1.

Stage 1: Coarse Depth Classification The segmented but noisy foreground mask $S(u)$ is classified into three levels of depth (see Fig. 1). The depth classification forest (DCF) serves a dual purpose. First, it removes most of the noise coming from the simple segmentation method. Second, it constrains the variation in terms of hand appearance that the steps further down have to deal with. Currently the system is trained to distinguish gestures performed in three intervals: 9-15cm, 15-24cm, 24-39cm, corresponding to a comfortable arm pose.

Stage 2: Shape Classification Once the foreground masks pass the DCF, the corresponding shape classification forest (SCF) classifies the images into (currently) six gesture classes, one additional *no-gesture* class, and a per-pixel noise class. The latter is necessary to deal with remaining false positives from the segmentation, and the former to robustly reject *non-gesture* motion in front of the camera. Fig. 1 illustrates typical colour-coded per-pixel classification results. White pixels are classified as noise.

Stage 3: Hand depth regression On the final level we switch from classification to regression forests. Here the goal is to map from an input pixel x to an absolute depth value. Note that in contrast to the previous level here we only run one forest per gesture (they are trained only on examples of one hand shape). The continuous value $y(x|S)$ is attained as $y(x|c,S) = \sum_{l=1}^{L} w_l y_l(x|c,S))$; l is the coarse depth level and w_l are the posteriors from the first layer. The main difference from classification forest to regression forest is the entropy definition. For regression, we employ the differential entropy of the empirical continuous density $p(y|S)$, modeled as a 1D Gaussian. This reduces to $E(S) = log(\sigma_s)$, where σ is the variance of the Gaussian.

2.4. Training Data

RFs rely on good training data for high classification accuracy. To train a classifier robust to large variation in hand shapes, sizes, distances to the camera and gesture execution, we need a large, but balanced training data-set. We asked 20 subjects to perform the gestures under natural variation and recorded short sequences of each. This included one 'no-gesture' where participants casually moved their hands. We recorded $\sim 50K$ images covering enough variation in rotation, depth and appearance for training the SCF.

3. Evaluation

We have conducted several experiments to evaluate the performance of our algorithm. Here, we only summarise the main experimental results, but we remand the reader to [9, 10] for an in-depth analysis.

Gesture Recognition. Fig. 3 summarizes classification accuracy as a confusion matrix for the entire gesture set, using both half training–half test and leave-one-out validation methods. Our technique achieves a mean per-class, per-frame accuracy of 98% and 93% respectively. In practice this translates to a very robust gesture recognizer with very little temporal filtering.

Figure 2. Applications: (A) Bimanual map browsing with a magnifier lens; (B) Shooting in a mobile game; (C) Music control in the air; (D) In-air pointing in front of the smartwatch triggers a photo sharing app; (E) Finding and selecting a contact card using hand depth.

PinchOpen	0.88	0.03	0	0	0	0	0.02	0.88	0.03	0	0	0	0	0.02
PinchClose	0	0.93	0.05	0	0	0	0.01	0	0.93	0.05	0	0	0	0.01
Pointing	0.02	0.01	0.9	0.04	0	0	0.01	0.02	0.01	0.9	0.04	0	0	0.01
Gun	0	0	0.02	0.95	0	0	0	0	0	0.02	0.95	0	0	0
SplayedHand	0	0	0	0.01	0.99	0	0	0	0	0	0.01	0.99	0	0
FlatHand	0.05	0	0	0	0.01	0.99	0.11	0.05	0	0	0	0.01	0.99	0.11
No-Gesture	0.05	0.03	0.03	0	0	0.01	0.85	0.05	0.03	0.03	0	0	0.01	0.85

Figure 3. Confusion matrix (20 users). Left: half–test / half–train cross-validation; avg. accuracy 98% Right: leave–one–subject–out; avg. per-frame accuracy 93%.

Depth Estimation. To evaluate our depth estimation step, we have compared our technique against ground-truth (GT) data acquired from a depth camera[1] and against a naïve depth estimation technique based on raw hand size. Fig. 4 show depth estimates data over $2K$ frames and under gesture variation. Our method tracks the GT closely, with small recurring spikes between the coarse depth levels. In contrast, the naïve technique systematically over and undershoots the GT, with a larger avg error (17mm vs. 81mm).

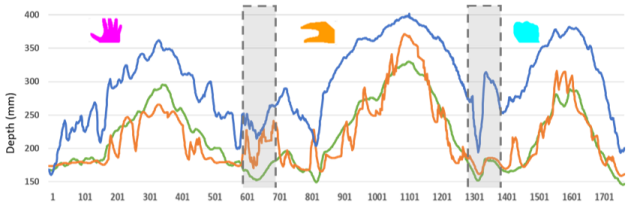

Figure 4. Depth estimation under gesture variation. GT (green) is tracked closely by Ours (orange). Naïve (blue) is worse.

4. Application Scenarios

To demonstrate the feasibility of our algorithm, we implemented several application scenarios where in-air gestures successfully complement touch input (a larger selection of applications is presented in [10, 9] and related videos). For instance, in a mapping application users can control zoom, pan and map switching with their non-dominant hand (Fig. 2, A). An in-air pinch gesture invokes a magnifier lens over a particular area of interest on the map. The lens can then be positioned by moving the hand behind the device. To showcase more complex bi-manual interaction, we interfaced our gesture recognizer with a 2D scroller game. Touch input is used to control the character position, whereas gestures are used to shoot weapons (Fig. 2, B).

The low computational requirements make our approach applicable on a wide range of devices. Fig. 2, C shows how a tablet in the kitchen can be controlled using effortless gestures without touching the screen with wet hands. In Fig. 2, D, a splayedhand gesture in front of a smartwatch triggers a photo sharing application. Finally, an important advantage of our technique is that it can recover gesture and hand position simultaneously. This allows users to jointly control discrete and continuous inputs. For example, gestures and depth may be used to invoke and browse linear list controls like selecting a contact card on smartglasses (Fig. 2, E).

[1] Creative Senz3D

5. Conclusion

We presented a robust gesture recognition algorithm that runs in real time on unmodified mobile devices with a single RGB camera. Our multi-layered RF classification-regression framework is well suited for low-memory devices. Our method exhibits high accuracy, which can be further improved if extended with temporal tracking.

References

[1] A. Colaço et al. Mime: Compact, low power 3D gesture sensing for interaction with HMD. In *UIST '13*.

[2] S. R. Fanello et al. Learning to be a depth camera for close-range human capture and interaction. In *SIGGRAPH '14*.

[3] C. Keskin et al. Hand Pose Estimation and Hand Shape Classification Using Multi-layered RDFs. In *ECCV '12*.

[4] D. Kim et al. Digits: Freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *UIST '12*.

[5] M. Ogata et al. SenSkin: adapting skin as a soft interface. In *UIST '13*.

[6] I. Oikonomidis et al. Tracking the Articulated Motion of Two Strongly Interacting Hands. In *IEEE CVPR '12*.

[7] S. Saponas et al. Enabling always-available input with muscle-computer interfaces. In *UIST '09*.

[8] M. Serrano et al. Exploring the use of hand-to-face input for interacting with head-worn displays. In *CHI '14*.

[9] J. Song et al. In-air gestures around unmodified mobile devices. In *UIST '14*.

[10] J. Song et al. Joint estimation of 3D hand position and gestures from monocular video for mobile interaction. In *CHI '15*.

[11] S. Sridhar et al. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. In *ICCV '13*.

[12] B. Zhang et al. Hobs: Head orientation-based selection in physical spaces. In *SUI '14*.