

Supplementary material for Rule Of Thumb: Deep derotation for improved fingertip detection

BMVC 2015 Submission # 72

1 Introduction

In this document we present some additional information to supplement a number of ideas from the main paper.

2 Derotation

In order to further facilitate understanding the effects of derotation and its variance reduction behavior we took the full test set derotated by DeROT and compared it with the non-derotated test set. The results can be seen in Figure 1.

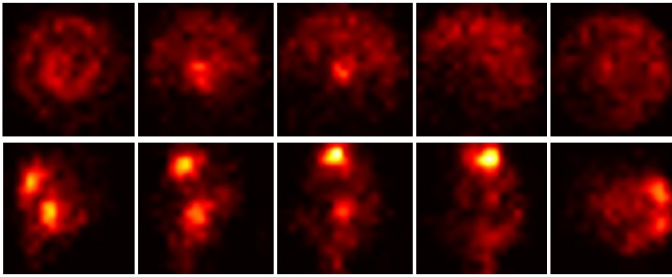


Figure 1: The top row from left to right shows an accumulation map for the individual finger locations (pinky, ring, middle, index, thumb) over all 10000 HandNet test images. The distribution of locations for every fingertip has high variance. The bottom row illustrates the effect of applying the proposed DeROT derotation method. The reduction of variance is noticeable in the much more concentrated distributions. It is interesting to note the bi-modal distribution for each fingertip which indicates that on average a fingertip spends most of it's time either extended (the mode further from the center) or curled (the mode close to the center). It is also useful to observe that the thumb has a more concentrated distribution to the right of it's accumulation map. This is sensible considering that the method is based on the heuristic of positioning the thumb in this fashion.

3 RDT and CNN input

For the features of an RDT we follow [4, 5]. An attribute vector with K features for any given pixel is defined as being $x = (x_1, x_2 \dots x_K)$ such that $x_k = z \left(q + \frac{o_{k,1}}{z(q)} \right) - z \left(q + \frac{o_{k,2}}{z(q)} \right)$ where

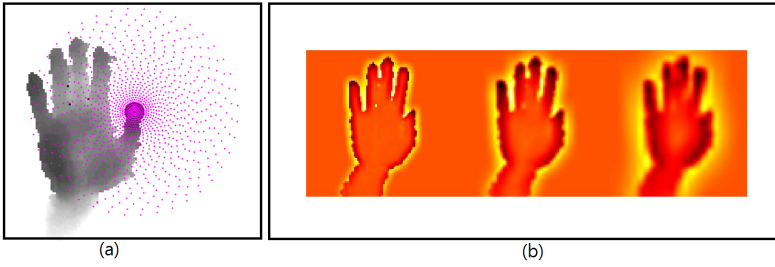


Figure 2: a) The image shows a depth map of a hand with the overlay of the offsets (in purple) that we use when generating the attributes per pixel. The example pixel in this case is in the center of the thumb. Random, pre-computed pairs of these offsets are used to generate the 1200 features for every example in the datasets. b) The input to the CNN for both regression and heatmap prediction is a triplet (of sizes 96×96 , 48×48 , 24×24) generated from an input depth map. Note that to highlight the difference as a result of gaussian smoothing we have upsampled each of the three inputs to the same size.

$q \in \mathbb{N}^2$ is an image coordinate. $z : \mathbb{N}^2 \rightarrow \mathbb{R}$ is the scene depth map supplied by the camera and $o_{k,[1:2]} \in \mathbb{N}^2$ are the two offsets defining the k_{th} attribute. Figure 2 shows the exponential offset distribution that we use which is based on BRISK [8].

4 Pipeline

The focus of the main paper is primarily on derotation and DeROT. However it is instructive to see the pipeline of the fingertip detection with the specific details as described for each fingertip detector. We display a highlevel overview of the process in Figure 3.

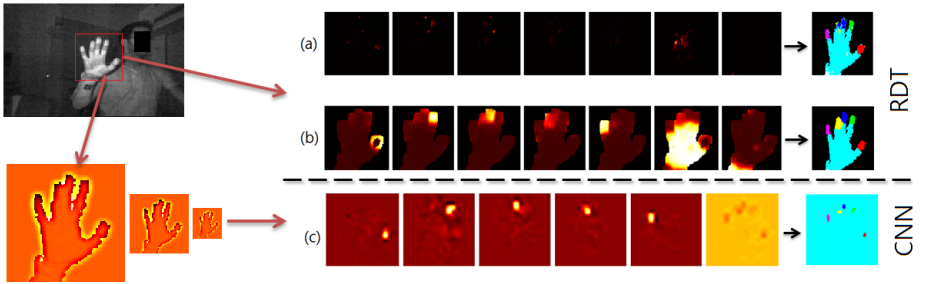


Figure 3: Here we show the pipeline from input image to final label classification. The IR image on the far left shows a user with their hand in front of the camera. a) RDT: The hand is then segmented and passed onto the tree which produces a posterior estimation per fingertip as described in the main body of the paper. There are 7 posterior images representing thumb, index, middle, ring, pinky, plam, wrist. They produce the noisy labeled image to the far right. b) RDT: This is the method proposed in the paper for reducing the noise. We first smooth each posterior channel and use that for labelling. The result is considerably less noisy than the previous approach and it is ready for blob extraction. c) CNN: The hand is cropped and subtractive LCN is used to produce an input triplet for the CNN. The heatmaps displayed here are upsampled to 128×128 and ordered by thumb, index, middle, ring, pinky, no-fingertip. The final label image has smaller label blobs than the RDT.

5 CNN architecture

We base our CNN on the architecture described in [8]. The authors of that work use a different training framework (Torch7 [9]) and there are likely to be differences between our architecture and theirs. However we believe our network closely approximates the one described

in that paper. We use Caffe [1] for defining our two CNN networks. Figure 4 and Figure 5 show the full structure displayed by using Caffe’s network visualization capabilities. For regression output we use 9 parameters and for the heatmap output we use $18 \times 18 \times 5 = 1620$ output values. It is likely that both architectures could be updated to perform the subtractive local contrast normalization directly and therefore use only a single input channel but this has not yet been implemented.

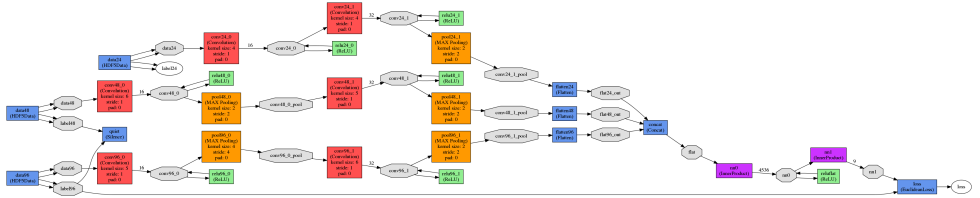


Figure 4: This is the network layout for the full hand orientation regression. We use Caffe’s HDF5 input format as opposed to leveldb because of its simpler interface and available tools in Matlab. Label96 is used to store the 9 orientation parameters for every batch input. Label48 and Label24 are silenced.

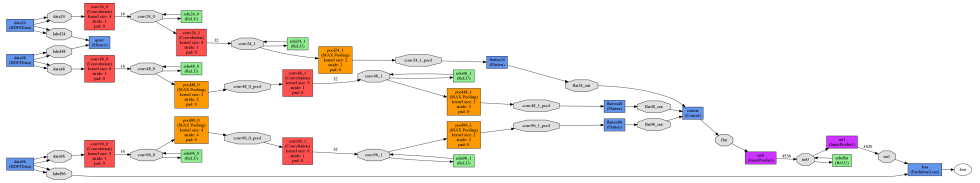


Figure 5: This is the network layout for a fingertip heat map predictor. We use Caffe’s HDF5 input format as opposed to leveldb because of its simpler interface and available tools in Matlab. Label96 is used to store the heatmap data for every batch input. Label48 and Label24 are silenced.

References

[1] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning.

[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[3] C. Keskin, F. Kiraç, Y. Emre Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013.

[4] 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 single depth images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, 2011.

[5] L. Stefan, C. Margarita, and S. Roland Yves. Brisk: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision (ICCV)*, pages 2548–2555. IEEE, 2011.

[6] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOC)*, 33, 2014.