Towards Touchless Palm and Finger Detection for Fingerprint Extraction with Mobile Devices*

Christof Jonietz[1], Eduardo Monari[1], Chengchao Qu[2,1]

[1]Fraunhofer IOSB
[2]Vision and Fusion Laboratory (IES), Karlsruhe Institute of Technology (KIT)
{firstname.lastname}@iosb.fraunhofer.de

Abstract: In this paper, contactless palm and finger detection for biometric fingerprint verification/identification process with mobile devices is considered. In order to speed up the border checking verification process, we focus on capturing the whole palm in order to extract each fingertip instead of successively capturing each fingertip. The workflow comprises palm detection in order to detect the skin region within the image prior to detection of fingertips. A machine learning based algorithm with Aggregated Channel Features (ACFs) adopted for palm detection is considered. Furthermore, a geometric shape based approach for fingertip detection has been designed to reconstruct long lines along fingers. Results demonstrate the performance of both algorithms.

1 Introduction

Automatic Border Control (ABC) will allow border control authorities to check travelers in a comfortable, fast and secure way. Here, a mobile scenario where border guards are using a device as a means of the travelers identity check is considered. In the European Union's Seventh Framework Program the project *MobilePass* focuses on research and development towards technologically advanced mobile equipment at land border crossing points. This will allow border control authorities to check European, visa-holding and frequent third country travelers in a comfortable, fast and secure way. Here, contactless palm and finger detection for biometric fingerprint verification/identification process with mobile devices is considered.

Several approaches for finger detection with different sensors are discussed in the literature. In [JHB13, PK14], finger detection is based on the top-hat transform, which uses morphological operators on a hand blob by subtracting the results from the original hand blob. However, a reliable segmentation of the hand palm/fingers in the RGB image is a prerequisite of many algorithms that analyze a binary hand blob image and the performance of these algorithms crucially depends on the segmentation result. In [RYMZ13], a shape representation based on time-series curve is used for fingertip detection. In [BLL07], a

*This project has received funding from the European Union's Seventh Framework program for research, technological development and demonstration under grant agreement No. 608016.

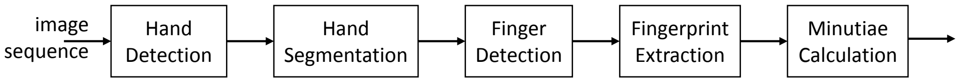

Figure 1: Processing chain.

skeleton-based approach is proposed for fingertip detection.

In order to speed up the border checking verification process, we focus on capturing the whole palm, which is presented by the person to be checked, in order to extract the fingertips, instead of successively capturing each fingertip. As a requirement for the capturing device, the resolution of the sensor should be such, that each fingerprint image can be cropped with a resolution of at least 500 dots per inch (DPI), which is the minimum resolution in a fingerprint verification/identification process. The workflow comprises palm detection in order to detect the skin region within the image prior to the detection of fingertips. A machine learning based algorithm with ACFs adopted for palm detection and finger detection, respectively, are considered. Furthermore, a fast geometric shape based approach for fingertip detection has been designed to reconstruct long lines along fingers.

The considered processing workflow is depicted in Fig. 1. In the first steps hand detection and hand segmentation are carried out, cf. Section 2. Based on the segmentation result, the fingers are detected in order to extract the fingerprint images. cf. Section 2.3.

2 Palm Detection, Segmentation, and Fingertip Extraction

2.1 Palm Detection using Aggregated Channel Feature

Object detection is one of the fundamental topics of research in the pattern recognition community. The Histograms of Oriented Gradients (HOG) introduced by Dalal and Triggs [DT05], although not state-of-the-art anymore, presented a classic approach for dealing with these tasks by combining rich feature descriptors and effective learning methods. In the past few years, a new family of features, namely the Integral Channel Features (ICF) [DTPB09] has attracted increasing interests, which integrate diversity of information into the computational efficiency using integral images. Recently, Dollár et al. proposed the Aggregated Channel Feature (ACF) in [DABP14]. The random rectangular blocks for calculating integral images in ICF [DTPB09, DBP10] are substituted by the sum of small squares. By exploiting the same learning framework, further performance and efficiency enhancements are reached.

Palm or hand detection in the context of *MobilePass* resembles that of pedestrian detection in the aforementioned work. With the constrained recording condition, the subjects are guided to show upright palm in front of the camera, analog to the upper body of pedestrians. However, the fingers can be spread to a variable extent, which is like the human legs. Considering the similarity between the two detection tasks, we explore the possibility of employing ACF for detection palms.

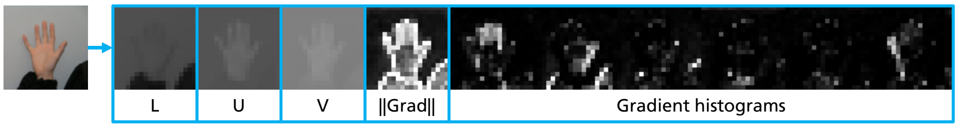

Figure 2: Illustration of the computed ACF from an example palm image.

Features. A bunch of feature combinations with regarding to different color space, intensity and gradient were experimented in [DTPB09], which yields a clear advantage of taking into account color information than the pure grayscale and gradient feature. LUV stands out among the color spaces. Moreover, the gradient magnitude and 6 gradient orientation bins are also included, resulting in totally 10 channels. Before and after dividing the channels into 4×4 blocks for summing up the pixels, pre-smoothing and post-smoothing with a $\frac{[1,2,1]}{4}$ filter are conducted. An example of the computed ACFs from a palm image is illustrated in Fig. 2. It is in particular obvious that the aggregated gradient magnitude and histograms can characterize the hand shape in a compact representation.

Multiscale ACF Approximation. The standard routine for object detection is based on sliding windows over the multiscale image pyramids. As a fine-scale pyramid is essential for successive detection, more than 50 scale steps are needed to process an image of 640×480 pixels. However, computing ACFs dozens of times on the image pyramid turns out to be costly. To circumvent the drawback of the conventional pipeline, the rich image features are ideally computed as few times as possible. Regarding to ACF, since the color channels can be directly resized to match different scale spaces, Dollár et al. [DBP10, DABP14] proposed a fast resampling scheme for the HOG features in ACF. Based on the observation that the statistics of natural images conform to the power law w.r.t. the ratio of scales, they proved that for the shift-invariant ACFs, a similar approximation also holds $\frac{\mathbf{f}_{\mathbf{\Omega}}(\mathbf{I}_{s_1})}{\mathbf{f}_{\mathbf{\Omega}}(\mathbf{I}_{s_0})} = \left(\frac{s_1}{s_0}\right)^{-\lambda_{\mathbf{\Omega}}}$, where $\mathbf{f}_{\mathbf{\Omega}}$, given the input image $\mathbf{I}$ of scale s_0, computes the channel feature $\mathbf{\Omega}$, which has its own corresponding scale factor $\lambda_{\mathbf{\Omega}}$. Accordingly, assuming $\lambda_{\mathbf{\Omega}}$ is obtained by least squares fitting of the training images, ACF channels $\mathbf{f}_{\mathbf{\Omega}}(\mathbf{I}_{s_1})$ of an arbitrary scale s_1 are simply a recalculation of that in the original scale space s_0, yielding

$$\mathbf{f}_{\mathbf{\Omega}}(\mathbf{I}_{s_1}) = \mathbf{f}_{\mathbf{\Omega}}(\mathbf{I}_{s_0}) \cdot \left(\frac{s_1}{s_0}\right)^{-\lambda_{\mathbf{\Omega}}}. \tag{1}$$

It is suggested that computing $\mathbf{f}_{\mathbf{\Omega}}$ only once per octave (resizing with doubled or halved size), and approximating the intermediate scales by Eq. (1) suffices to find an ideal trade-off between accuracy and speed. Moreover, by virtue of the aggregation of the features, evaluating ACF approximation $\mathbf{f}_{\mathbf{\Omega}}(\mathbf{I}_s)$ is even faster than resizing the image $\mathbf{I}_s$ itself.

Training. Despite of the sparsity of hand and palm datasets, we were able to collect 345 images from 3 publicly available hand gesture datasets, i.e. the Sébastien Marcel Static Hand Posture Database [Mar99], the Database for Hand Gesture Recognition (HGR) [NGK14], and the hand gesture dataset acquired with Leap Motion and the Kinect devices [MDZ14]. The hand gesture "five" in these datasets resembles the upright open palm scenario in *MobilePass*. Selected samples of the combined dataset are shown in Fig. 3, demonstrating a large variation in gesture shape, image quality and resolution, illumination and background clutter, etc. All these nuisance factors are extremely challenging for training a powerful ACF detector.

Figure 3: Example images of our dataset for training the ACF palm detector.

We manually annotated the training images with square bounding boxes and resized the hand crops to 50×50 pixels. Boosted tree with soft cascades [FHT00] is leveraged to train and combine 2048 depth-two trees over all candidate ACF channel lookups. Bootstrapped learning employed by the Viola-Jones detector [VJ01] is also exploited. For sampling negative samples, the INRIA pedestrian dataset [DT05] is used.

2.2 Palm Segmentation

Given the bounding box of the detected palm, provided by the ACF detector as described in Sec. 2.1, the next step is to determine the subset of pixels which belong to the hand. This segmentation step allows for sub-sequential shape analysis of the hand, and as a consequence to search for finger tips.

One straight forward approach for segmentation of body parts is using skin color. In literature, several approaches exist which mainly try to detect pixels with human skin tones, without any prior knowledge on images content [LP10, FAKK02]. However, due to the applied palm detection step as pre-processing, in our case skin color estimation and segmentation can be designed in a more robust way.

Our palm segmentation approach is sub-divided in the following 4 steps:

Pixel-based Skin Tone Detection. For pixel-based skin color detection, we use the algorithms proposed in [CCCM09]. While most alternative approaches basically try to remove illumination component from images to obtain an illumination invariant color classifier, the authors in [CCCM09] claims, that illumination is also an important feature for pixel-wise skin tone classification.

The algorithm first determines a grayscale map of the RGB color image given an usual transformation matrix $\mathbf{a} = [0.298936, 0.587043, 0.140209]^\top$ by standard product operation $\mathbf{I}'(\mathbf{x}) = [r(\mathbf{x}), g(\mathbf{x}), b(\mathbf{x})] \otimes \mathbf{a}$. The resulting 1D image $\mathbf{I}'$ is considered as the grayscale map of the original image, taking into account all color channels. Additionally to $\mathbf{I}'$ a second illumination map $\hat{\mathbf{I}}'$ is determined, considering green and blue channel components only. Red channel is discarded from this grayscale map, since it is the most contributing one in skin pixels, $\hat{\mathbf{I}}'(\mathbf{x}) = \max\left(g(\mathbf{x}), b(\mathbf{x})\right)$. Now, a skin color probability map is generated by pixel wise signal error calculation as $\mathbf{E}(\mathbf{x}) = \mathbf{I}'(\mathbf{x}) - \hat{\mathbf{I}}'(\mathbf{x})$.

Finally, given a large training dataset containing regions with skin from persons of a range of races/cultures with extreme variation of lighting effect, the interval of $\mathbf{E}(\mathbf{x})$ for the majority of skin pixels has been estimated. Given a lower and upper bound, skin pixels are classified as follows:

$$\mathbf{M}_{skin}(\mathbf{x}) = \begin{cases} 1 & \text{if } 0.02511 \leq \mathbf{E}(\mathbf{x}) \leq 0.1177 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

One important reason for choosing this skin tone detection method was also its efficiency and real-time capability. No complex color space transform is needed and reduction of color space dimensionality from 3D (RGB) to 1D (normalized grayscale) allows for very low computational load and as a consequence efficient pixel-wise classification.

(a) Original image with ACF detector ROI.

(b) Image after skin color segmentation.

(c) Clutter removal, blob filling and morphology.

Figure 4: Hand segmentation by skin color.

Connected Component Analysis for Clutter Removal / De-noising. Given the skin pixel segmentation image as shown in Fig. 4b, in a second step a connected component analysis is performed for blob analysis and clutter removal. All small segments with an unreasonable too small area related to palm region-of-interest (ROI) are classified as clutter. In our examples, we choose 5% of ROI area as threshold.

Also segments outside palm detector ROI are discarded in further processing. Finally, the remaining blob with the largest size is selected as hand segment candidate for further processing. The result is a segmentation image, containing blobs of skin colored areas of significant size, only.

Post-Processing / Morphology (hole filling). Given the hand segment candidate as described above, in a final post-processing morphological filters are applied for hole filling. Hereby, a background flood fill approach, followed by a binary image inversion is applied. The resulting segmentation result is shown in Fig. 4c.

2.3 Fingertip Extraction

The main idea of the finger detection algorithm is to associate the “left” and “right” edges to fingers (edge-pairing) and to extract them by their respective tips and angles. The edge-pairing algorithm is subdivided into the following steps.

Hand palm segmentation. The input RGB image (Fig. 5a) is segmented in order to extract the finger edges. Hand segmentation is based on the algorithm in Section 2.2.

Converting the Binary Blob Image into a Contour Line. In order to process and examine the blobs contained in the binary input image (Fig. 5b) efficiently, the blobs are converted into contour lines, cf. Fig. 5c. A contour is a sequence of pixels located along the boundaries of the blob.

(a) Input RGB image. (b) Segmented palm. (c) Line segments. (d) Edges.

Figure 5: Finger detection by edge pairing.

Line Segmentation of the Contour Line. The most important recognition features of fingers are approximately long lines along them. In order to detect such lines and deduce the presence of fingers, it is necessary to isolate them from the contour line of a blob by isolating line segments within a contour. Line segments are detected by analyzing the variation of the tangent angle on the contour. Within a line segment the variation is low, i.e. the tangent angle remains almost constant.

Forming fingers by matching edges. The formation of fingers based on the previously computed line segments is provided by edge pairing. A pair of edges is consequently interpreted as the "left" and "right" edge of the fingers, respectively. The method is illustrated in Fig. 5d, where the line segments used for reconstruction of paired edges are plotted in the same color. An edge can only be combined with another edge. Unpaired edges are discarded and are not recognized as a finger. In order to find the best match for each edge, a match quality metric for a hypothetical pair of edges is calculated, which depends on the orientation of edges, the distance between center points, the maximum allowed pairing angle, and the length of edges.

3 Results

Results are provided for the ACF palm detector and the finger extraction algorithm based on assigning associated edges through pairing in the following.

Results of the ACF approach for palm detection are presented in Fig. 6. Obviously, the trained detector localize the palms independent of subjects and shape. Robustness against closed and spread fingers is demonstrated. By virtue of the relatively simple background, no false positives are seen in the test cases, although the confidence threshold is set to a very low level.

Results of the proposed finger extraction algorithm based on assigning associated edges through pairing are presented. In Fig. 7, the finger axes and the ROIs are depicted. The finger roots and tips are labeled as red and blue circles, respectively. The ROIs containing that part of the finger are used for a later biometric verification or identification process. After computing the finger axis and the fingertip, cf. Section 2.3, the size of the ROI is determined dynamically in order to take into account individual finger sizes. The width of the ROI corresponds to the width of the finger and is determined by examining the binary blob-image on a virtual line perpendicular to the finger-axis. The height of the ROI depends on its width and is determined by multiplying the finger width with a constant factor (here: 1.7).

Figure 6: Example results of palm detection by the ACF detector.

Here, different results for palms and fingers are presented. The most important recognition feature of fingers are approximately long lines along them. Since this algorithm has been designed to reconstruct long lines along fingers, only well separated fingers can be reconstructed, i.e., results with spread fingers are shown here. Since extraction of edges is based on an reliably segmented palm in order to deduce the presence of fingers, the segmentation of the palm is crucial. If these two conditions are fulfilled, the dynamic ROI can be determined reliably, as depicted in Fig. 7.

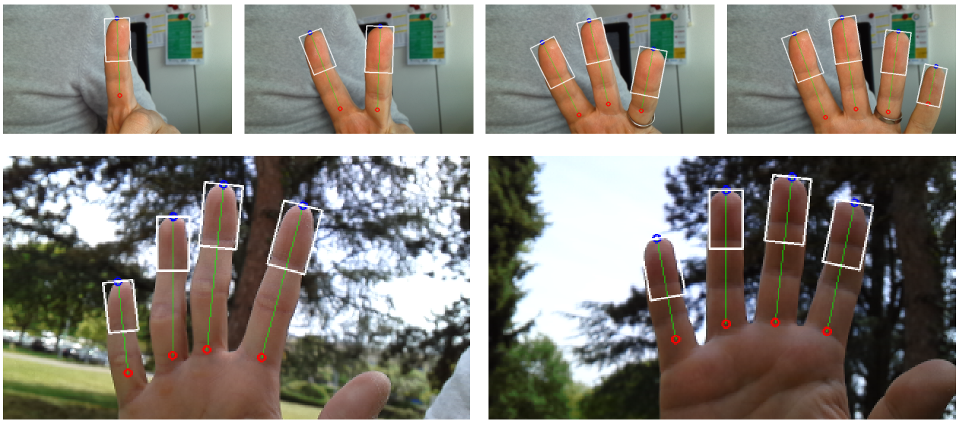

Figure 7: Results of the proposed fingertip detection algorithm in different scenarios.

4 Conclusions

In this paper, contactless palm and finger detection for the biometric recognition process for mobile devices has been considered. ACF hand detection and finger detection based on edge-pairing are proposed. Evaluation based on a measurement campaign in different indoor and outdoor scenarios demonstrate the suitability of the geometric approach by edge pairing in extracting fingertips for a biometric identification/verification process.

References

[BLL07] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):449–462, March 2007.

[CCCM09] A. Cheddad, J. Condell, K. Curran, and P. McKevitt. A new colour space for skin tone detection. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 497–500, Nov 2009.

[DABP14] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast Feature Pyramids for Object Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, Aug 2014.

[DBP10] P. Dollár, S. Belongie, and P. Perona. The Fastest Pedestrian Detector in the West. In *Proceedings of the British Machine Vision Conference*, pages 68.1–68.11. BMVA Press, 2010.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, June 2005.

[DTPB09] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *Proceedings of the British Machine Vision Conference*, pages 91.1–91.11. BMVA Press, 2009.

[FAKK02] A. Farrukh, A. Ahmad, M. I. Khan, and N. Khan. Automated segmentation of skin-tone regions in video sequences. In *Students Conference, 2002. ISCON '02. Proceedings. IEEE*, volume 1, pages 122–128, Aug 2002.

[FHT00] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statististics*, 28(2):337–407, 04 2000.

[JHB13] S. B. Jemaa, M. Hammami, and H. Ben-Abdallah. Data-mining process: application for hand detection in contact free settings. *Image Processing*, 7(8):742–750, November 2013.

[LP10] H. C. V. Lakshmi and S. PatilKulakarni. Segmentation Algorithm for Multiple Face Detection for Color Images with Skin Tone Regions. In *Signal Acquisition and Processing, 2010. ICSAP '10. International Conference on*, pages 162–166, Feb 2010.

[Mar99] S. Marcel. Hand Posture Recognition in a Body-face Centered Space. In *Proceedings of the Conference on Human Factors in Computer Systems (CHI)*, CHI EA '99, pages 302–303, New York, NY, USA, 1999. ACM.

[MDZ14] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1565–1569, Oct 2014.

[NGK14] J. Nalepa, T. Grzejszczak, and M. Kawulok. Wrist Localization in Color Images for Hand Gesture Recognition. In Dr. A. Gruca, T. Czachórski, and S. Kozielski, editors, *Man-Machine Interactions 3*, volume 242 of *Advances in Intelligent Systems and Computing*, pages 79–86. Springer International Publishing, 2014.

[PK14] P. Prasertsakul and T. Kondo. A fingertip detection method based on the top-hat transform. In *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–5, May 2014.

[RYMZ13] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. *Multimedia, IEEE Transactions on*, 15(5):1110–1120, Aug 2013.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518, 2001.