Computer Vision Winter Workshop 2009, Adrian Ion and Walter G. Kropatsch (eds.)
Eibiswald, Austria, February 4–6
Publisher: PRIP, Vienna University of Technology, Austria

Slap Fingerprint Segmentation

Robert Hödl [1], Surinder Ram [1], Horst Bischof [1], and Josef A. Birchbauer [2]

[1]Institute for Computer Graphics and Vision, Graz University of Technology, Austria
robert.hoedl@student.tugraz.at, ram@icg.tu-graz.ac.at, bischof@icg.tugraz.at

[2]Siemens IT Solutions and Services PSE, Siemens Austria, Biometrics Center
josef-alois.birchbauer@siemens.com

Abstract *This paper presents a novel approach in segmenting multiple fingerprints from an image. Such an image is taken by a scanner capable of recording several fingers simultaneously. A combination of two-staged mean shift and ellipse-fitting algorithms as well as an elaborate subsequent set of rules is used to segment the single fingertip images. First, the mean shift which is a well-established feature-space analysis technique is used to identify the different components of the fingers. Then the orientation and size of each component is determined by the application of a robust ellipse-fitting algorithm. Finally the rules locate the fingertips. Extensive experimental evaluations demonstrate the success of the approach.*

1 Introduction

The attractiveness of fingerprints results from their uniqueness which does not change through the life of individuals [6]. Within the past few years the increasing demand of security and safety has led to a large scale application and deployment of automated fingerprint identification systems (AFIS). New laws and regulations have been introduced. Two examples which should be mentioned in this context are the US-VISIT program[1] and the Visa-Information-System (VIS)[2] of the European Union. The latter is scheduled to start in 2009.
These new rules require the reading of all ten fingerprints instead of one or two. The motivation is the dramatic improvement of the false accept and false reject rates if more than one finger is used to reference an individual.
These policies raise the problem that using market standard (single-) fingerprint readers would absolutely be not sufficient in terms of the amount of time needed for the enrollment process and the convenience of both, visitors and customs officials.
To address this problem, ten-fingerprint[3] scanners capable of recording multiple fingers at the same time, are deployed. Most of these scanners capture so called 'slap images' which are 'four-finger simultaneous plain impressions' [12] (see Fig. 1). This type of images makes it necessary to separate the biometrical interesting fingertips out of the rest of the so called 'slap fingerprint image' or 'slap image'. The whole process where the slap image is divided up into four images of the fingertips is called 'slap fingerprint segmentation' [12].

Figure 1: Typic slap fingerprint image.

The main goal of any slap fingerprint segmentation algorithm is of course the correct segmentation of the fingertips. The specific task requires handling certain constraints especially regarding performance and accuracy of the results:

- **Correct segmentation of the fingertips**
  This point expresses the basic necessity that in a four finger slap image all four fingertips should be detected and segmented in one piece. The segmentation has to be as tight-fitting as possible using a rectangular shape. Another requirement is the correct orientation of the bounding boxes. The angle of these boxes should point along the angle of the corresponding finger.
- **Correct assignment and classification of the fingers**
  The fingertips detected in the slap image need to have a unique mapping with numbers from one to four, corresponding to their location on the hand from left to right.
- **Left or right hand detection**
  The algorithm should be able to distinguish between the left and right hand.
- **Performance**
  Since the scanner is capable of providing a live preview,

[1]http://www.dhs.gov/xtrvlsec/crossingborders/
[2]http://europa.eu/scadplus/leg/en/lvb/l14517.htm
[3]Also known as '4-4-2 scanners'

it is essential that the algorithm is able to process at least four frames per second. Otherwise the live-scan preview mode would be too slow and stuttering.

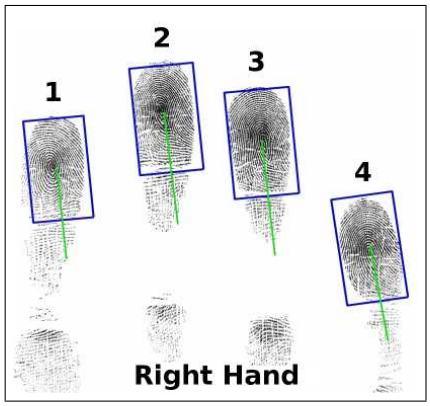


Figure 2: Typic slap fingerprint image with an example segmentation result (boxes); angles (lines) and labels added manually.

1.1 Related research

As described in [6] already proposed fingerprint segmentation techniques rely on the application of local histograms of ridge orientations or block-wise calculations on the variance of gray-levels. Other approaches make use of the responses of Gabor filters convolved with single image blocks or use pixel-wise features to divide (single-) fingerprint images into background and foreground.

Worth mentioning in the context of slap fingerprint segmentation is the 2004 NIST organised contest called 'Slap Fingerprint Evaluation 2004' [12]. Within this contest thirteen segmentation algorithms were assessed and compared. This points out the general importance of the slap fingerprint segmentation problem. Twelve of the thirteen algorithms are under disclosure since they are intellectual property of different companies. The only public accessible algorithm is the NISTs own implementation called NFSEG. NFSEG first preprocesses and binarises the slap input image. In order to find the fingertip location and global hand orientation it 'searches for large quantities of black pixels' [13] along scanlines applied in different angles.

Another algorithm to be mentioned is described in the United States Patent 7072496 [14]. It is based on segmenting the image into disconnected regions using edge detection and a convex hull calculation on the different components. Finally the components which are expected to form a hand geometry are identified and selected.

2 Segmentation approach

In order to realise the segmentation task described in the preceding section, especially considering the performance requirements, only a segmentation algorithm suitable for real-time applications can be taken into account. One example is the so called CAMSHIFT (Continuously Adaptive Mean Shift) [10], which is based on a modified version of the mean shift algorithm [4, 2].
The CAMSHIFT algorithm is described as a robust, 'simple and computationally efficient face and colored object tracker' [10].
What makes the CAMSHIFT so interesting for this kind of task is that it was originally designed as a face tracking algorithm assuming that 'faces are somewhat elliptical' [10]. As shown in the example above, the characteristics of the different fingertips strongly resemble elliptical shapes.
The CAMSHIFT tracking process is divided into two steps where at first a mean shift is applied and secondly an ellipse fit is calculated. The adaption of those steps is described in the following sections.

2.1 Preprocessing

The original size of the images provided by the scanner (CROSSMATCH L SCAN Guardian[4]) would have made any real time analysis impossible. Since not the full resolution is needed, the images are resampled to 1/10 of their original size to 150x160 pixels.

2.2 Mean shift

This section describes how the properties of the mean shift as a well proven kernel density gradient estimator are used in identifying 'interesting regions' - the fingers - in a slap image. The mean shift algorithm is a nonparametric technique for detecting local extrema of an underlying distribution. It is based on a recursive procedure which repeatedly finds the mean location within a search window and translates itself to that location [10].

1. Choose a search window size.
2. Choose an initial location of search window.
3. Compute the mean location in the search window.
4. Center the search window at the mean location computed in step 3.
5. Repeat steps 3 and 4 until convergence (or until the mean location moves less than a preset threshold).

In the nth step, the next search window location (or centroid) is calculated by:

$$x_{n+1} = \frac{M_{10}}{M_{00}}; \quad y_{n+1} = \frac{M_{01}}{M_{00}} \tag{1}$$

where M_{00} denotes the zeroth and M_{10}, M_{01} the first order moments which are defined as:

$$\begin{aligned} M_{00} &= \sum_x \sum_y I(x_n, y_n) \\ M_{10} &= \sum_x \sum_y x_n I(x_n, y_n) \\ M_{01} &= \sum_x \sum_y y_n I(x_n, y_n) \end{aligned} \tag{2}$$

[4] http://www.crossmatch.net/

$I(x, y)$ is the pixel value at position (x,y) in the image.

The mean shift vector (MSV) - the vector by which the search window is translated - is calculated by:

$$MSV(x) = \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} - \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (3)$$

2.3 Mean shift procedure

For actually finding the different components of the fingers in the image the mean shift algorithm is applied twice:

[Mean shift I] First, equidistant seed points for mean shifts are set all over the image, having the circular search windows overlap. In a second step for every seed point the mean shift is started. A threshold on the number of pixels above a certain gray value within the search window is used to decide if it is worth starting the mean shift.
Another threshold is introduced in order to remove mean shift results which stopped converging right in-between two fingers, having the same weighting in both semicircles of the search window. The rest of the modes against which the individual mean shifts converge are stored (Fig. 3a).

[Mean shift II] In the second step, the resulting modes found in step one need to be merged. This is due to the fact that although the individual mean shifts from step one converged into the direction of the same local extreme, they stop at slightly different locations (Fig. 3b and Fig. 3c). To merge these points, again the characteristic of the mean shift procedure to locate density extrema within a set of data points is used.
A new mean shift procedure is set up, taking into account some neighbourhood and stability considerations. One such consideration is that hits on the same local maxima which are very close together are weighted more heavily than ones which lie far apart. The resulting pruned mode set is considered to define the centers of different finger segments - the potential fingertip candidates (see Fig. 3d).

2.4 Ellipse fitting

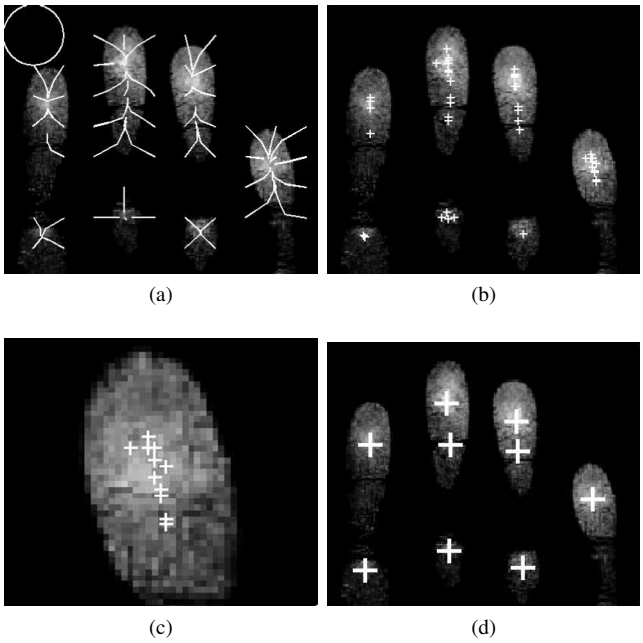
The next step after detecting the potential fingertip candidates is to calculate an ellipse fit for the regions of interest (ROI) in order to determine the size and orientation of the fingertips.

As proposed in [10] and [11] the major and minor axis lengths of the ellipses are derived using properties of the eigenvalues of the covariance matrix.

The covariance matrix of a $ROI(x_i, y_i)$ is given by:

$$Cov_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \quad [5] \quad (4)$$

with $\bar{x}$ and $\bar{y}$ being the respective medians of x_i and y_i. The resulting 2x2 matrix is called a 'robust covariance matrix' referring to the more robust results when using a median over a simple mean.



(a) (b) (c) (d)

Figure 3: (a) visualised paths of the converging mean shifts; in the top left corner the size of a search window is demonstrated; (b) white marks indicate different local modes detected; (c) detected local modes on the rightmost finger in detail; (d) merged local modes resulting from second mean shift step.

The square root of both eigenvalues $\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$ of this matrix, with $\lambda_1 > \lambda_2$, specify the lengths of the semimajor and the semiminor axis of the ellipse respectively.
The corresponding eigenvector of λ_1 is used to determine the orientation of the ellipse major axis. The angle α is calculated by a four-quadrant inverse tangent function (notably called $atan2(y, x)$).

The whole ellipse fitting step of the algorithm for the 4-finger slap images is again divided into two parts:
Within the first step, the size and orientation of each potential candidate is determined and the average angle of all ellipses is calculated. The second step then uses a search window of elliptical shape, turned into the direction of the average angle. This is done to calculate the ellipse fit again, in order to robustly compute a mean orientation of the observed set of finger components.

[Ellipse fit I] For the first step, ellipse fitting as described above is applied to all candidates, again using a circular search window. In order to calculate a robust average angle of the hand, ellipses with problematic features are identified and rejected.

We reject ellipses if:

- **size**
  The estimated size of the ellipse is smaller than a certain amount of the search window size.

- **orientation**
  The difference between the angle of the specific ellipse and the average angle of all other ellipses is too large.

[Ellipse fit II] In the second step, the ellipse fitting process is reapplied to all fingertip candidates, this time using an adapted search window of elliptical shape. The shape is given by the factors 1.1 for the semimajor and 0.9 for the semiminor axis, referring to the original search window radius. These search window ellipses are turned into the direction of the average angle and the ellipse fit is - once again - recalculated for these special ROIs.
The improvements made by executing the ellipse fit a second time, using an elliptical search window, is significant. Particularly regarding images with neighbouring fingers lying tightly together. The elliptical shape of the search window omits the incorporation of neighbouring fingers as shown in Fig. 4c. Therefore, the algorithm is able to provide a robust estimation about the global orientation of the hand depicted in the image. The correctness of this information is vital to the finger logic described in the following section.

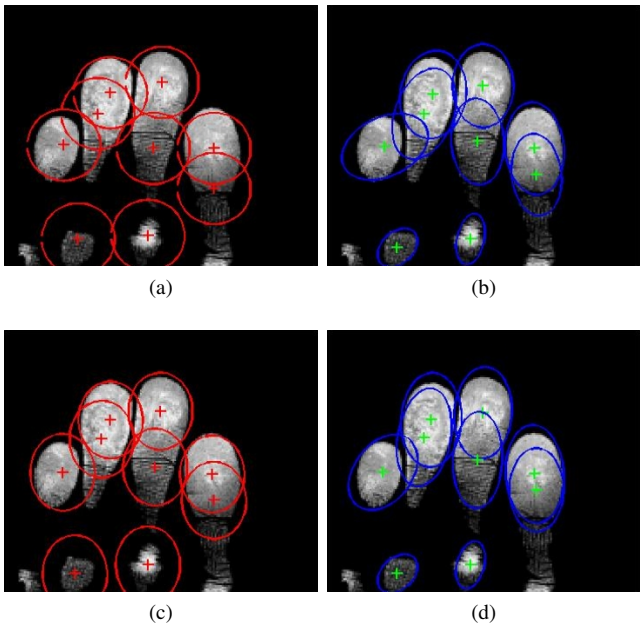


Figure 4: Visualising the advantage of the reapplied ellipse fit; (a) circular search windows used in first run; (b) result after first run - note the toppled ellipse at the leftmost finger; (c) elliptical search windows applied in second run of the ellipse fit - note the difference of inclusion at the leftmost search window; (d) nearly perfect (in size and orientation) fitted ellipses resulting from second ellipse fit step.

2.5 Finger logic

The task of the finger logic is to interpret the results gained from the previous mean shift and ellipse fitting steps. It has to detect and appoint the four correct fingertips and determine if the hand is either a left or right hand.
In order to register the four wanted fingertips, the image is 'turned upright' (i.e. orientation normalised) according to the average angle of the hand. This has the effect that the modes of the single fingers lie in straight lines parallel to each other. The finger logic starts with the leftmost mode found. It advances from left to right and identifies modes belonging to one finger and prunes all but the topmost one. For doing so, the topmost mode figuratively casts a conical shaped 'spreading shadow' over all others lying beneath it. Modes that are located within the shadow are sorted out (see Fig. 5).
The decision on a left or a right hand is done using the unique feature of the middle finger of being the hands longest finger. So the logic decides on a right hand if the second fingertip from the left is the 'highest' in the orientation normalised image. A left hand is detected if the third finger from the left is the longest. If the longest-finger feature is true for one of the other fingers the algorithm returns an 'unknown hand' type.

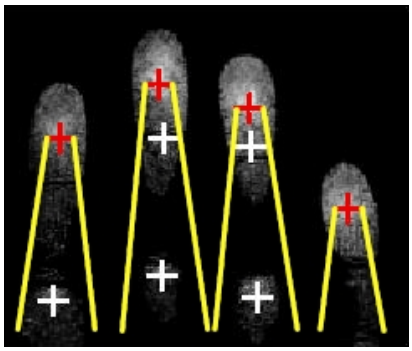


Figure 5: Orientation normalised example illustrating the procedure of the finger logic. Red crosses determine the selected modes. Yellow lines indicate the 'spreading shadow' for sorting out the remaining modes (white crosses).

2.6 Flowchart

The workflow of the whole algorithm is shown in Fig. 6.

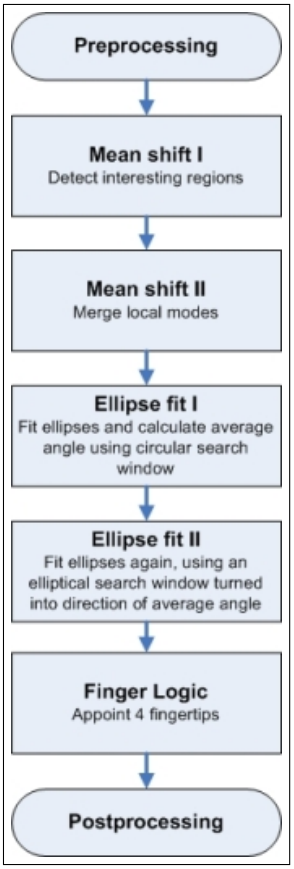


Figure 6: Flowchart illustrating the processing steps of the slap segmentation algorithm.

3 Results

3.1 Testdata

In order to make specific segmentation results measurable, three datasets named USER, POS and NARROW were created. Each dataset contains 50 different slap images taken from five test subjects with respectively five images from the left and the right hand.
The names of the datasets correspond to their characteristics. For the USER dataset the subjects were asked to intuitively place their hands on the scanner. The POS dataset was intended to measure the performance of the algorithm on images with distorted orientation of the hand. The NARROW dataset contains images where the users press their fingers very tightly together. Surprisingly, it was nearly impossible to produce images without a space in-between the fingers.

3.2 Annotation

With the implementation of a special annotation tool it was possible to manually annotate and label the ground truth of the provided testdata. Each finger in the image was assigned the corresponding number from left to right. The best fitting polygons were designated and the corresponding angles defined (shown in Fig. 7). At last a decision between the left and right hand could be made.

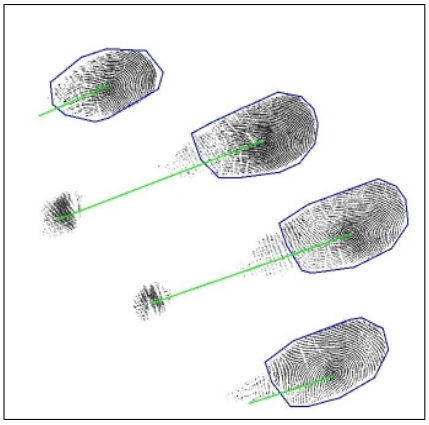


Figure 7: Example annotation showing how the best fitting polygons and angles were annotated manually.

3.3 Evaluation

For the purpose of evaluating the performance of the algorithm, different comparable measures were tested. Some of the evaluation work was done under the aspect of parametrising an optimal window size used for the mean shift and the ellipse fitting process.

3.3.1 Overlap score The amount of overlap between the ground truth and the segmented regions serves as a measure of performance. The overlap score [8] (also known as the 'Jaccard similarity' [3]) is used for comparing the similarity and diversity of sample datasets. It is defined as the size of the intersection divided by the size of the union of the sample datasets (with G being the ground truth):

$$OVERLAP(G,S) = \frac{AREA(G \cap S)}{AREA(G \cup S)} \tag{5}$$

When the two regions are totally disjoint the overlap score is 0. When both regions match completely the overlap score is 1. As shown in Tab. 1 and Fig. 8 the results were determined particularly for every dataset.

WS	USER	POS	NARROW	**TOTAL**
17	0.76	0.64	0.74	**0.71**

Table 1: Overlap score for different datasets using a window size of 17 pixels.

Since the total overlap score averages to 0.71, it is obvious that the bounding boxes produced by the algorithm very closely match the polygons of the ground truth. It again has to be considered that polygons are being compared to rectangular boxes.
The variance with the POS dataset can be explained by the bigger error made when calculating the average angle (see Section 3.3.3) and the finger logic not being able to appoint the correct ellipses.

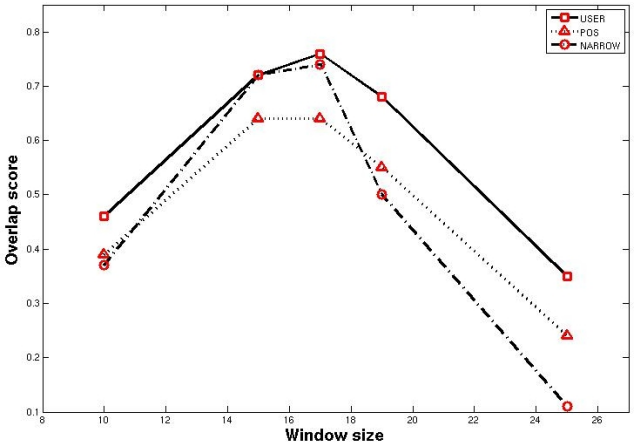


Figure 8: Overlap score with respect to the window size for the different datasets. It can be seen that the maximum overlap score is reached when using a window size of 17 pixels.

3.3.2 Precision / Recall In order to determine the window size at which the algorithm performs best, the pixel based quality measures Precision and Recall were used too. In short Precision is the 'probability that a found pixel is correct' and Recall the 'probability that a correct pixel is found'. These two measures are calculated by [9]:

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN} \quad (6)$$

TP...True Positive, FP...False Positive, FN...False Negative

The calculation of Precision and Recall was done separately for each dataset using different window sizes. The results are presented in Tab. 2 and Tab. 3.

WS	USER	POS	NARROW	**TOTAL**
16	0.87	0.72	0.83	**0.80**
17	0.81	0.68	0.80	**0.76**
18	0.77	0.67	0.71	**0.72**

Table 2: Precision: probability that a found pixel is correct

WS	USER	POS	NARROW	**TOTAL**
16	0.90	0.78	0.87	**0.85**
17	0.94	0.79	0.92	**0.88**
18	0.93	0.80	0.86	**0.86**

Table 3: Recall: probability that a correct pixel is found

In Tab. 2 it is clearly visible that the Precision gets higher with decreasing window size. This makes sense because by using a small window size it is more likely that a found pixel is correct since regions around the fingertips are not segmented. The maximum total probability reached was 80% using a window size of 16 pixels.

The maximum Recall probability over all three datasets is reached by using a window size of 17 pixels. Since the combined mean probability of Precision and Recall at the window sizes 16 and 17 is nearly equal (∼82%), other criterions (angle error rates and left / right hand detection errors) were actually used to decide on a preferred window size.

3.3.3 Angle error rates The angle error rates were also taken into account when determining the window size. For all fingers detected in a dataset the standard deviation and the mean value were calculated for different window sizes as shown in Tab. 4 and Tab. 5.

WS	USER	POS	NARROW	**TOTAL**
16	7.3	13.9	16.0	**12.4**
17	6.5	12.7	13.9	**11.0**
18	6.1	24.0	20.7	**16.9**

Table 4: Standard deviation of angle errors in degree

WS	USER	POS	NARROW	**TOTAL**
16	5.5	9.1	7.3	**7.3**
17	5.0	8.2	6.2	**6.5**
18	4.6	11.0	10.1	**8.5**

Table 5: Mean value of angle errors in degree

These two tables show that the best result of a minimal angle error is achieved by using a window size of 17 pixels. It has to be mentioned that a standard deviation of 11° and a mean value of 6.5° are satisfying results. This is regarding any fingerprint matching algorithms which have to deal with 'involuntary rotations up to ±20° with respect to vertical orientation' [6].

3.3.4 Left / right hand detection errors An error made by the finger logic when deciding on a left or right hand was recognized as such, if either a wrong or unknown hand type was returned. The performance of the algorithm using different window sizes is presented in Tab. 6.

WS	USER	POS	NARROW	**TOTAL**
16	4	16	20	**13.3**
17	6	8	20	**11.3**
18	2	18	22	**14.0**

Table 6: Left / right hand detection errors in percent

3.3.5 Window size Based on the fact that the algorithm performed best using a window size of 17 pixels in four out of five tested scenarios it is clear that this window size was the first choice.

3.4 Examples

This section presents segmentation examples generated by the algorithm. The results shown in Fig. 9 are very satisfying with respect to the segmentation requirements defined in Section 1.

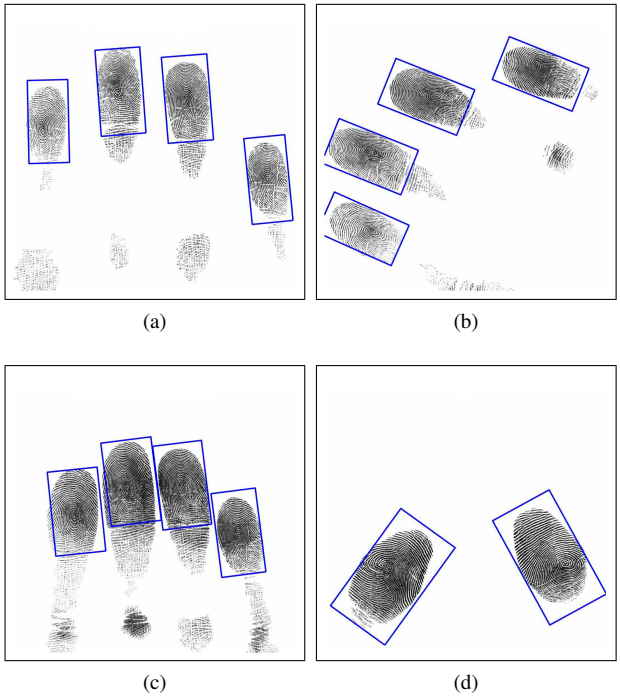


Figure 9: (a) image taken from the USER dataset; (b) orientation distorted image out of POS dataset; (c) difficult image from NARROW dataset - result is satisfactory; (d) two-finger thumb image - note the orientation independency.

4 Conclusion

In the context of future work it is necessary to mention the possible improvement of the underlying PDF on which the mean shift procedure is working upon. While evaluating the implementations of different techniques to generate artificial PDFs, like 'coherence' (COH) [1] or 'orientation certainty level' (OCL) [7], it became clear that both have major deficiencies. Both are very time-consuming to calculate. Moreover COH showed very strong responses at the edges of the fingertips while OCL was very noise sensitive. Due to the mentioned performance issues the available choices seem very limited.
Another desirable improvement could be realised by implementing an adaptive window size. The characteristics of a hand differs greatly from person to person. In the fields of the applications mentioned, the algorithm should be able to scale better between small and fragile as well as big and tall hands.
There is also room for improvement if the finger logic would be based upon a 'hand geometry model'. This model would allow introducing plausibility checks of results suggested by the finger logic and the possibility to correct wrong decisions.

In conclusion, it can be said that this paper presents a novel approach to segment the fingertips out of a slap image. The used technique of combining a mean shift clustering with an ellipse fit calculation and an elaborate logic produces accurate segmentation results. The two-staged mean shift clustering is used to gain knowledge about the different finger locations. The ellipse fit as well as the reapplied ellipse fit calculation is able to assign robust size and orientation hints to the different components of the fingers. The subsequent logic then seeds out the wanted fingertips. The success of the algorithm is shown clearly by the presented experimental results and error rates.
The main advantages of this approach lie in its real time capabilities, its robustness concerning tightly aligned fingers and orientation distorted images.

Acknowledgement

This work has been funded by the Biometrics Center of Siemens IT Solutions and Services PSE, Siemens Austria.

References

[1] Dass S. Chen Y. and Jain A. Fingerprint quality indices for predicting authentication performance. Technical report, Michigan State University, Dep. of Computer Science and Engineering, 2005.
[2] Yizong Cheng. Mean shift, mode seeking, and clustering. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, 1995.
[3] Hill D. Crum W., Camara O. Generalized overlap measures for evaluation and validation in medical image analysis. In *IEEE Transactions on Medical Imaging*, volume 25, 2006.
[4] Comaniciu D. and P. Meer. Mean shift: A robust approach towar feature space analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, 2002.
[5] Bartsch H.-J. *Taschenbuch Mathematischer Formeln*. Fachbuchverlag Leipzig, 20 edition, 2004.
[6] Maltoni D. Jain A. K. *Handbook of Fingerprint Recognition*. Springer-Verlag New York, Inc., 2003.
[7] Jiang X. Lim E. and Yau W. Fingerprint quality and validity analysis. In *IEEE ICIP*, 2002.
[8] Efros A. Malisiewicz T. Improving spatial support for objects via multiple segmentations. Technical report, Carnegie Mellon University, Robotics Institute, 2007.
[9] McCallum A. Peng F. Accurate information extraction from research papers using conditional random fields. Technical report, University of Massachusetts, Dep. of Computer Science, 2004.
[10] Bradski R. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal Q2*, 1998.
[11] Hou H. Tsai D. and Su H. Boundary based corner detection using eigenvalues of covariance matrices. In *Pattern Recognitnion Letters*, volume 20, pages 31–40, 1999.

[12] Watson C. Indovina M. Ulery B., Hicklin A. and Kwong K. Slap fingerprint segmentation evaluation 2004 analysis report. Technical report, National Institute of Standards and Technology, 2005.

[13] Tabassi E. Wilson C. McCabe R. Janet S. Watson C., Garris M. and Ko K. *User's Guide to Export Controlled Distribution of NIST Biometric Image Software*, 1998.

[14] Sankar P. Zhen-Ping Lo P. Slap print segmentation system and method. United States Patent No. 7072496, 1987.