# Hand recognition using geometric classifiers

Yaroslav Bulatov* Sachin Jambawalikar† Piyush Kumar‡ Saurabh Sethia†

## 1 Introduction

Biometric recognition systems find applications in security systems with varying requirements. While finger printing and iris based systems work well for high security applications, they are not as suitable for medium and low security applications because of privacy concerns. Hand Geometry based verification systems find more acceptance because hand geometry is not considered distinctive enough to establish a positive identity.

Hand geometry recognition systems may provide three kinds of services. Verification, classification and identification. For verification the user provides her identity along with the hand geometry and the system verifies her identity. For classification the user does not provide any identity information but is known to be legitimate. For identification the user does not provide any identity information other than the hand geometry and may be an intruder. The system tries to identify the individual or deny access.

**Previous work** Jain et.al. developed a hand geometry based verification system and used it for a prototype web security system [3]. They obtained 10 hand images each from 50 users but discarded 140 of them because users didn't do what was expected of them. From the remaining images they report a false acceptance rate (FAR) of 2% and a false rejection rate (FRR) of 15%.

Jain and Duta [4] developed a verification system that aligns finger contours and measure the mean alignment error between them. They experimented with 353 images from 53 persons and report FAR of 2% and FRR of 3.5%. Raul Sanchez-Reillo et. al. [6] report experiments with a system similar to the one in [4]. However they implemented verification as well as identification. They used a database of 200 hand images from 20 people and report 97% success in identification and error rates below 10% in verification. Öden et. al. [5] report on a system for identification and verification using implicit polynomials. They combine their method with geometric features and achieve 95% success in identification and 99% success in verification.

## 2 Data Collection and Feature Extraction

Previous work on hand geometry recognition used images obtained from a set up that includes a digital camera. Most of them also have pegs or similar mechanism to guide the hand in a somewhat consistent position. We however collected data using a document scanner. Thus no special set up was required. Also the users were free to keep their hands anywhere on the scanner. The only instruction given to the users was to keep their fingers separated and have their hand roughly vertical on the scanner surface. Users were free to stretch their hands to whatever level they felt comfortable. In fact some of the users were encouraged to stretch their hands to different extents for different scans so as to generate difficult data. We took around 10 scans of the right hand of 70 people to obtain a total of 714 images.

Thirty different features were extracted from each image. See figure 1. We omit description of our feature extraction algorithms from this abstract.

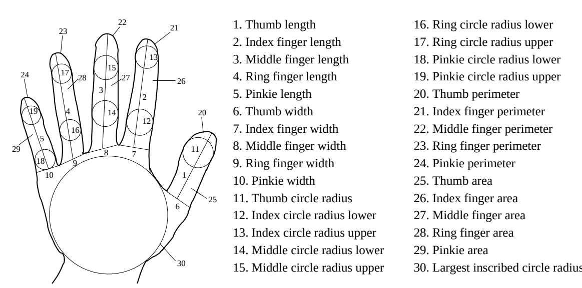


**Figure 1.** Feature extraction

*Computer Science, Oregon State University, Corvallis, OR 97331, USA.
†Biomedical Engineering, State University of New York at Stony Brook, NY 11794, USA.
‡Computer Science, State University of New York at Stony Brook, NY 11794, USA.

| Training set size | Nearest box classifier | Minimum enclosing ball classifier |
|---|---|---|
| 3 | 2.81% | 1.37% |
| 4 | 1.74% | 1.15% |
| 5 | 1.72% | 0.92% |

**Table 1.** Misclassification rates.

## 3 Nearest box classifier and verifier

In this section we describe a simple classifier and verifier that we use to implement hand geometry verification, classification, and identification. For each person we pick a small number (3 to 5) of hand images as training set. We find the bounding box for each training set in the 30 dimensional feature space. For a query feature vector, the distance to these bounding boxes in $L_\infty$ metric is used as a measure of similarity. The distance along each feature axis is normalized using the maximum variability observed for that feature in the training set. For verification, we use a experimentally determined threshold to decide whether a query feature vector is close enough to a given training set. For classification we simply classify the query point to the set with nearest bounding box. Identification is implemented by classifying the query point assuming it to be legitimate followed by verification.

Figures 2 and 3 show the average plots of false acceptance rate (FAR) and false rejection rate (FRR) for different values of the threshold. The plots show that while keeping FAR less than 1%, we can achieve FRR below 3% for verification and below 6% for identification. Table 1 reports the average misclassification rates.

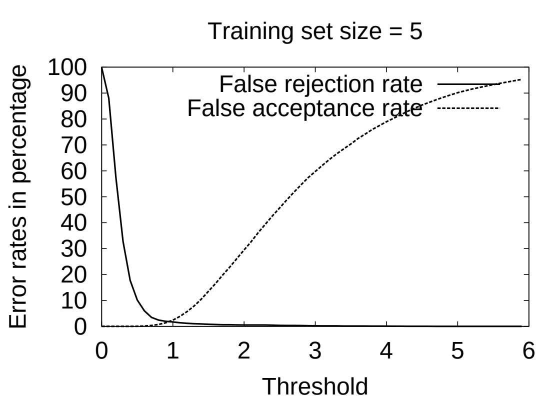


**Figure 2.** Verification error rates

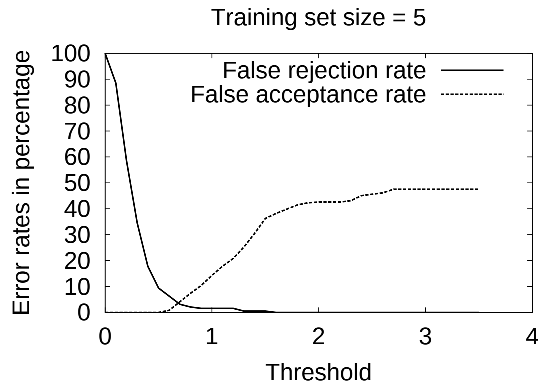


**Figure 3.** Identification error rates

## 4 Minimum enclosing ball classifier

In this section we describe a novel classifier which performs better than the nearest box classifier and is also of general interest for other classification problems. The classifier is based upon ideas from mathematical programming, core sets [1], support vector clustering [2] and pattern classification.

This classifier first maps all feature vectors to a higher dimensional space using a Gaussian kernel [2], employs a novel way to approximate minimum enclosing balls for each training set in the mapped space and then uses the Voronoi diagram of the centers for classification.

In the absence of information about the probability distribution of data, there is little theoretical justification of using one window width over another for the Gaussian Kernel. Nevertheless the classifier seems to work well in practice(See Figure 4) for large range of window widths. The approximate method of computing minimum enclosing balls used in this classifier can also be used for support vector clustering. We omit the algorithmic descriptions from this abstract. One drawback of this method is that we don't know how to take care of outliers efficiently in practice, although theoretically this is known [1]. In practice, we observed that the classifier works well for a few outliers.

## 5 Conclusion

We have developed a hand recognition system that uses hand geometry for verification, classification, and identification of individuals. We tested the system on a database of 714 hand images from 70 people. Our experiments show that hand geometry based recognition systems can be used for medium security applications. Stronger claims about such a system can only be made after conducting experiments at a larger scale.

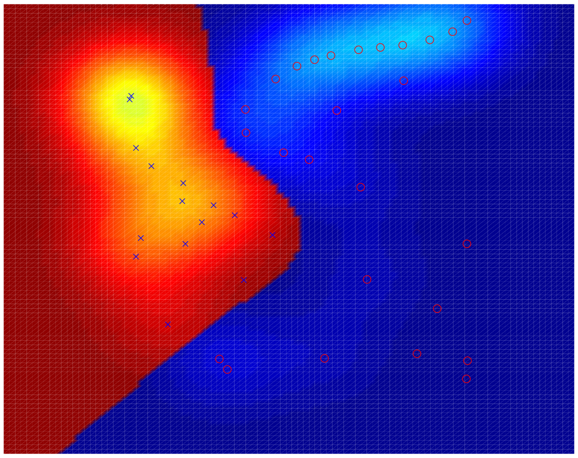

**Figure 4.** An example classification of a point set in $R^2$

## References


[1] Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. *Proc. 34th Annu. ACM Sympos. on Theory of Computing*, pages 250–257, 2002.

[2] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *revised version Jan 2002*, 2002.

[3] A.K. Jain, A. Ross, and S. Pankanti. A prototype hand geometry-based verification system. In *Proceedings of 2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication*, pages 166–171, March 1999.

[4] Anil K. Jain and Nicolae Duta. Deformable matching of hand shapes for verification. In *Proceedings of International Conference on Image Processing*, October 1999.

[5] Cenker Öden, Aytül Erçil, Vedat Taylan Yildiz, Hikmet Kirmizitas, and Burak Büke. Hand recognition using implicit polynomials and geometric features. In Josef Bigün and Fabrizio Smeraldi, editors, *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, volume 2091 of *Lecture Notes in Computer Science*, pages 336–341. Springer, June 2001.

[6] Raul Sanchez-Reillo, Carmen Sanchez-Avila, and Ana Gonzalez-Marcos. Biometric identification through hand geometry measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1168–1171, October 2000.