

# Palmprint Template Protection Scheme with Matrix Transformation

Hengjian Li, Jian Qiu, Changzhi Yu

ICCCV '18: Proceedings of the 2018 International Conference on Control and Computer Vision June 2018

Pages 3–7

A report by-

Navneet Kumar Chaurasiya(2016167)

Shivani A Mehta(2016248)

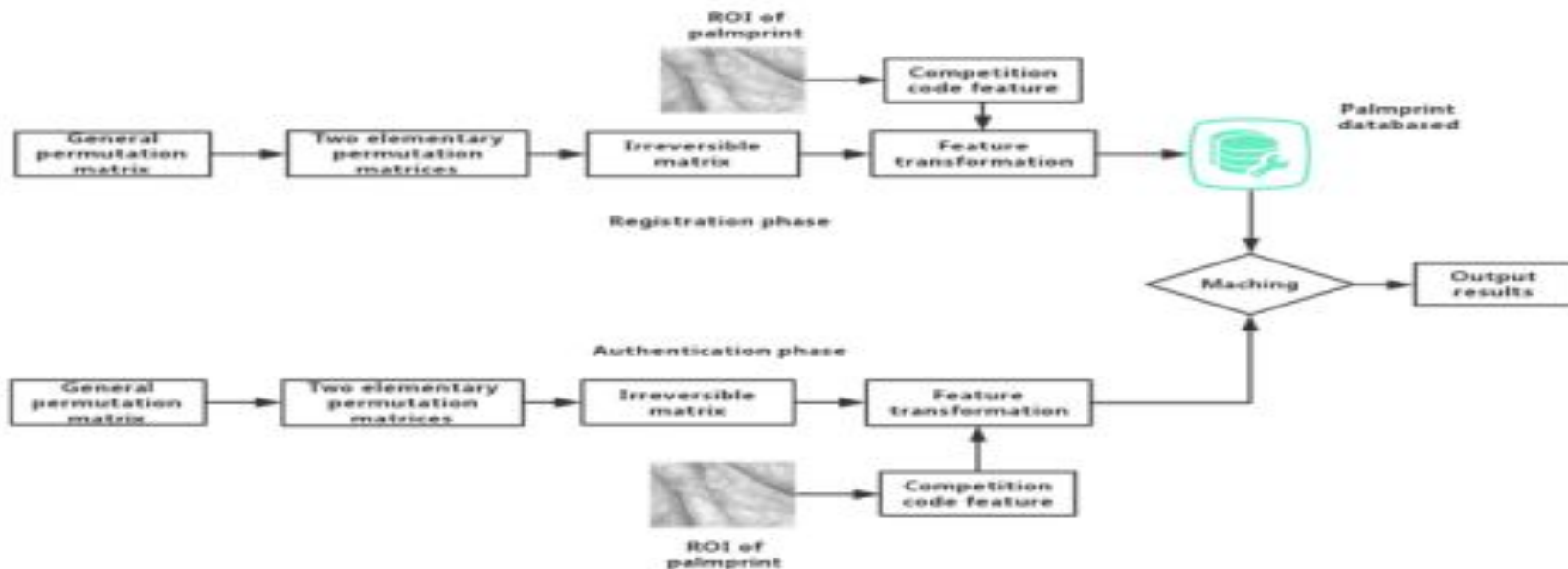
TA - Gourav Siddhad

# About

Firstly, the competition code features of original palmprint is extracted through the Gabor filters. Then, a general permutation matrix is generated randomly and two elementary permutation matrices are obtained by changing any two rows of it. Next, an irreversible matrix is generated by XORing operation. Finally, cancelable palmprint templates are produced by multiplying the irreversible matrix and the original palmprint feature. The experimental results show that our cancelable palmprint scheme can not only ensure high safety but also meet the recognition accuracy requirements.

# Summary

## The proposed system:



# Identification/Authentication Scenario

## Identification Scenario

---

First we analyse the identification scenario, which corresponds to 1:M classification problem. Biometric identification answers the question “Who are you?”. It is usually applied in a situation where an organization needs to identify a person. The organization captures a biometric from that individual and then searches in a database in order to correctly identify the person.

## Authentication Scenario

---

Second, we analyse the authentication scenario, which corresponds to 1:1 problem or binary classification. The question that is asked is: “Can you prove who you are?”. A system will challenge someone to prove their identity and the person has to respond in order to allow them access to a system or service. For example, a person touches their palm on a sensor embedded in a smartphone, used by the authentication solution as part of a challenge/response system.

# Results

## Multiclass SVM Model

```
Mdl =  
ClassificationECOC  
    ResponseName: 'Y'  
    CategoricalPredictors: []  
    ClassNames: [1 2 3 4 5 6 7 8 9 10 11 12 13 14]  
    ScoreTransform: 'none'  
    BinaryLearners: {780x1 cell}  
    CodingName: 'onevsone'
```

[Properties, Methods](#)

## Cross Validation

```
CVMdl =  
classreg.learning.partition.ClassificationPartitionedECOC  
    CrossValidatedModel: 'ECOC'  
    PredictorNames: {1x20480 cell}  
    ResponseName: 'Y'  
    NumObservations: 236  
    KFold: 4  
    Partition: [1x1 cvpartition]  
    ClassNames: [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]  
    ScoreTransform: 'none'
```

[Properties, Methods](#)

# Results

X= 352x20480

Y=352x1

ExtractGaborFeatures.mlx   X = 352x20480																																				
	g150	g151	g152	g153	g154	g155	g156	g157	g158	g159	g160	g161	g162	g163	g164	g165	g166	g167	g168	g169	g170	g171	g172	g173	g174	g175	g176	g177	g178	g179	g180	g181	g182			
200	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0			
209	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0			
210	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0			
211	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0			
212	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0			
213	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0			
214	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0			
215	1	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0			
216	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	1	0	0	1	0	0	1	0	0	0			
217	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1			
218	1	0	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0			
219	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1			
220	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0		
221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0			
222	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0			
223	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0			
224	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0			
225	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1			
226	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0			
227	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	1			
228	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0			
229	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	1	0			
230	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0			
231	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	1	0	0	0	1	0	0			
232	0	1	0	0	0	1	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1			
233	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	0	1	0	0			
234	1	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0		
235	0	1	1	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0		
236	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0		
237	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
238	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
239	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
240	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1		

ExtractGaborFeatures.mlx   Y = 352x1			
		↑	
304	52		
305	52		
306	53		
307	53		
308	53		
309	53		
310	53		
311	53		
312	54		
313	54		
314	54		
315	54		
316	54		
317	55		
318	55		
319	55		
320	55		
321	55		
322	55		
323	56		
324	56		
325	56		
326	56		
327	56		
328	56		
329	57		
330	57		
331	57		

# Problem Faced

1. Not able to use a set of Gabor Filters for feature extraction. This is due to the fact that large feature vectors were computationally time consuming for a 8GB Laptop. Therefore, only one Gabor Filter was used.
2. Lack of information about SVM Classification in paper.
3. According to the proposed method, there were 100 classes each having 6 palmprint images, where 3\*100 images were used for training and remaining 300 for testing. We implemented this but our K-Fold generalization error always turned out to be >80%.
4. We used **fitcecoc** model of MATLAB for multiclass SVM and tried to train the model with various parameters as mentioned below:
  - a. Coding = 'allpairs' and 'onevsone', 'binarycomplete', 'denserandom', 'onevsall', 'ordinal', 'sparsrandom', 'ternarycomplete'
  - b. Learners = 'discriminant', 'kernel', 'knn', 'linear', 'naivebayes', 'svm', 'tree'
  - c. Crossval = 2,3,4,5,6,7,8,9,10
  - d. Used k-fold loss function to measure generalization error.

# Improvised Solution

We employed Gabor filter for image enhancement and Harris Detector for detecting Lines and Edges in Palmprint ROI. We then extracted the features and used Docker for securing the palmprint features.

---



# Conclusion

Competition code features of palmprint are extracted by Gabor filters. The extracted features of the competing code occupy a small space, and the extraction accuracy is high. Irreversible matrix is generated by general permutation matrix. The key space is large enough. The cancelable palmprint templates are produced by multiplying the irreversible matrix and the original palmprint feature. In this paper, the method based on matrix transformation can not only protect the original palmprint template, but also satisfies the recognition accuracy