Cascaded Filtering for Biometric Identification using Random Projections

Atif Iqbal and Anoop M. Namboodiri
Center for Visual Information Technology
IIIT, Hyderabad 500032, India
Email: {atif.iqbal@research.,anoop@}iiit.ac.in

***Abstract*—Biometric identification often involves explicit comparison of a probe template against each template stored in a database. This approach becomes extremely time-consuming as the size of the database increases. Filtering approaches use a light-weight comparison to reduce the database to smaller set of candidates for explicit comparison. However, most existing filtering schemes use specific features that are hand-crafted for the biometric trait at each stage of the filtering. In this work, we show that a cascade of simple linear projections on random lines can achieve significant levels of filtering. Each stage of filtering consists of projecting the probe onto a specific line and removal of database samples outside a window around the probe. The approach provides a way of automatic generation of filters and avoids the need of developing specific features for different biometric traits. The method also provides us with a variety of parameters such as the projection lines, the number and order of projections, and the window sizes to customize the filtering process to a specific application. Experimental results show that using an ensemble of projections reduce the search space by 60% without increasing the false negative identification rate.**

I. Introduction

Biometric identification has a long history and was used in criminal and prisoner identification as early as 1870s. One of the early problems that was encountered in identification of a person from a large number of potential candidates was the time required for explicit comparison of a biometric sample against each possible candidate. Around 1900, Sir Edward Henry developed an indexing scheme for fingerprint based identification, where each fingerprint was classified into one of five classes (right loop, left loop, whorl, arch, and tented arch) based on the overall ridge flow in the fingerprint. This reduces the search space to one-fifth, albeit with some misses due to misclassification of samples into the above categories.

With the advent of computer based biometric identification the speed of matching has improved considerably obviating the need of indexing in most databases. However, electronic storage and comparison has also enabled several large scale identification programs such as voter registration programs, national ID projects and large-scale AFIS in law enforcement. Many of these datasets contain tens of millions of records and a single identification request can take a significant amount of time even with the advances in computing speeds. The search time becomes an important factor in the success and failure of such systems.

Approaches to reducing the search time falls into two categories: indexing and filtering. Indexing, as mentioned before, classifies a probe as belonging to a specific class (or a few classes), and uses only that part of the dataset from the same class for explicit comparisons. The process is extremely quick as the time required for classification of the probe is independent of the database size. However approach assumes that the biometric trait can be partitioned into mutually exclusive set of classes and classification into these classes is accurate. Filtering approaches relaxes this assumption and uses a simple light-weight matcher to compare the probe against each entry in the database. All samples that are potential candidates from this matching process is passed on to the next stage for further comparisons.

We note that the ideal feature representation of strong biometric trait used for identification is not well-suited for indexing as the inter-class distances tend to be close to each other as evidenced by low variance of the imposter distribution. Similarly, the length of feature representation and the comparison mechanisms used in practice for strong biometric traits makes it too heavy for use in a filtering process. As supported by experimental evidences, a direct approach to indexing biometric data such as the use of indexing structures like KD-Trees on the feature representations of a strong biometric does not yield satisfactory results. To overcome these difficulties, researchers and biometric practitioners have proposed a variety of features and matching strategies, often tailored to a biometric trait for the purpose of indexing and filtering. As we are using the indexing or filtering stage as a precursor to explicit matching, we would like to keep the False Non-Identification Rate (FNIR), very close to zero, while pruning the database as much as possible. FNIR indicates the probability that a probe with a matching record in the database would return a no-match after the entire identification process.

Automatic classification of fingerprints into the Henry classes was explored by Jain *et al.* [8], yielding a system with 12.4% FRR. A similar work by Ratha *et al.* [16] yielded a False reject rate(FRR) of 10% with search space pruned to 25% of the original database. In an experiment conducted by Cappelli *et al.* [3]on NIST Special Database 4, it was shown that the distribution of Fingerprint population was non-uniform with 2 of the 5 Henry classes they considered holding nearly 65% of the population. Note that the FNIR (corresponds to FRR in this case) is too high for most practical purposes and often one has to search more than one bin in the database for every probe. This further reduces the effectiveness of the method.

The pyramid indexing [14] technique tries to map a feature vector into one of the pyramids centered at the mid point of the feature range. The index of the pyramid and the location of the probe within the pyramid helps to reduce the search space to points within a few pyramids in the database. The authors report considerable success with this technique, with a database pruned to 8.86% of original size with 0% FNIR in case of hand geometry. Unfortunately the method performs poorly with larger feature vectors such as Gabor responses of IRIS images. Mehrotra *et al.* [13] proposed the use of ordered DCT coefficients for indexing a dataset of IRIS images. The authors were able to prune the database to around 2.6% with an FNIR of 35.6%. The method is sensitive to the location and orientation of the samples and does not work well with other modalities such as palmprints or fingerprints.

For palmprints, Zhang *et al.* [19] proposed the use of high-level textural information to filter out a set of possible candidates for fine-grained matching using interest points. Hierarchical identification of palmprint, where a Hough transform of the principal lines is used as a feature for filtering was proposed by Li and Leung [12]. Local information extracted from line-based Hausdorff Distance (LHD) is used for further fine-level identification.

In short, we note that the feature representations and the indexing and filtering schemes developed are often tailored for a specific biometric modality. In this work, we explore the use of random linear projections as a generic method for deriving features from a given feature representation of a strong biometric for the purpose of filtering. We also propose a cascaded window based filtering scheme that would be applicable to such feature representations in an efficient manner.

A. Random Projections

The use of linear projections to reduce the dimensionality of a dataset is a well explored topic. Approaches such as Principal Component Analysis and Linear Discriminant Analysis try to find a set of projections for a given dataset that would maximize a specific objective function. In other problems such as unsupervised learning, the objective function is either not defined or cannot be optimized analytically. The distance preserving nature of linear projections into random subspaces were explored by Johnson and Lindenstrauss [10] in 1984 (JL Theorem), who showed that random projections preserve the structure of high dimensional data well in lower dimensions. Specifically, the distortion in distances, when mapping n p-dimensional points into a q-dimensional random subspace, where $q \geq O(\log(n)/\epsilon^2)$ is less than a factor of $1+\epsilon$. The method of random projections have been proven to be useful in a variety of practical applications such as dimensionality reduction, density estimation [4], data clustering [5], nearest neighbor search [11], [7], document classification [15], etc.

Random projections have also been used in biometric verification to derive lower dimensional feature representations of modalities such as face [6] and to derive cancelable representations using Multi-space Random Projections [9]. In this work, we explore the use of single random projects as weak classifiers that can act as a filtering stage for efficient biometric identification. We employ each projection as an independent filter in a cascaded fashion [18] to achieve efficient and flexible filtering. The use of cascades as a method for improving efficiency of matching for iris was also suggested in [17].

II. Filtering with Projections

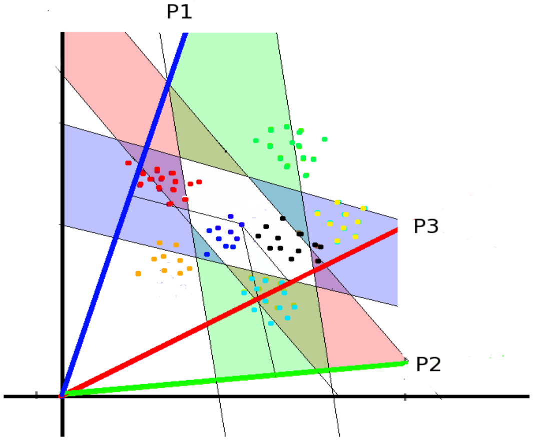

Fig. 1. Cascading random projections: P1, P2 and P3 are three projections used in a sequence. Samples that are not falling within a window of the probe are removed at each stage.

We consider each projection as a weak but efficient representation of the biometric dataset. Matching against a dataset, where each sample is represented as a scalar is extremely efficient. If the samples of each class are clustered in the projected space, it is reasonable to assume that samples of the same class will be within a window of the probe in the projected space. At each stage, we discard the samples that are outside the window. Figure 1 shows the result of projection of a set of two dimensional samples on to three projections and discarding the samples that lie outside a window. The white polygon in the middle represents the samples that are selected from the cascaded filter.

Algorithm 1 Compute Candidate list for a probe.

```
$CandidateList \Leftarrow$ {All templates in gallery}
for each projection $P_i$ do
    Retrieve projected values for $CandidateList$ for $P_i$
    Find the window around the projection of probe on $P_i$
    Remove templates outside window in $CandidateList$
end for
Return $CandidateList$
```

Although the final set of samples that are selected are independent of the order of projections, the efficiency of the cascade is clearly dependent on it. If we use projections that remove large number of impostor samples at initial stages of the classifier, the number of comparisons at later stages of the cascade can be minimized. If the projection preserves the intra-class similarity in the projected space as compared to the inter-class variations, then we can use a small window that would reject a large number of impostors without losing any genuine samples. The property that we like to maximize is close but not identical to the Fisher criterion, the ratio of between-class scatter to within-class scatter (S_B/S_w).

At projection the data outside the filter window around the probe is analyzed. If the data is from the different class (person), we call it a correct reject (rejected correctly) and if the data is from the same class, we call it a false reject (rejected falsely). The fitness of a projection i with a window W may be calculated using the following:

$$c_i = \frac{\sum\limits_{j \notin W} \neg S(j)}{\sum_j \neg S(j)} \tag{1}$$

$$f_i = \frac{\sum\limits_{j \notin W} S(j)}{\sum_j S(j)}, \tag{2}$$

where $S(j)$ is and indicator variable that takes a value 1, when j is of the same class as the probe, and N is the total number of samples. The score of the i^{th} projection is defined as the ratio:

$$Score_i = \frac{c_i}{1 + f_i}. \tag{3}$$

We note that the definition of this objective function does not yield to an analytic formulation of a minimization problem to find the optimal set of projections. The use of the Fisher criterion will give us the most discriminating set of basis vectors. However, as we note from Figure 1, the use of additional projections over a basis set of vectors can further improve the filtration process. To address this problem, we start with a large number of random projections, and select those which maximizes the above criterion function. One could also include the discriminating basis vectors along with the random projection before the selection process. Section IV compares the use of LDA vectors as a projection basis as opposed to random projections, and the effect of its combination.

A. Advantages of Random Projections

The use of random projections allow us to deal with a variety of problems encountered in other linear projection estimation techniques. As noted before, according to the Johnson-Lindenstrauss lemma[10], a random subspace of dimensionality $O(logn)$ can effectively represent n samples in any high dimensional feature space. Moreover, the use of random projections make the resulting representation to be independent of the training data, and hence addition of new data does not require changes to the random basis. We partially negate this advantage by choosing a subset of the random projections that best filter the training data. One can also produce any number of projections as desired unlike methods such as PCA or LDA that are limited by the rank of the covariance matrix or the number of classes.

Avoiding matrix inversions that are required in the computation of other linear projection methods makes the computation more numerically stable and widely applicable. The training process is also relatively less expensive.

III. Implementation Details and Challenges

Binarized feature vectors such as those used in palm and iris codes do not behave well for indexing and filtering purposes. We use the response values of the filters to carry out the indexing. Each feature is first normalized to the range $[-1, 1]$ using the sigmoidal function:

$$y = \frac{1 - e^{-sx}}{1 + e^{-sx}} \tag{4}$$

where x is a feature value of the sample and s decides the slope of the sigmoid function. We have selected $s = 1.5$ for palmprint dataset and $s = 10$ for iris. Once all the samples are projected on the random basis, they are scaled to the range $[0, 10]$. Note that the range of projected values depend on the length of the feature vector.

A. Determining Window Width and Cascade Sequence

As the filtering is a precursor to the regular identification stage, it is desirable to tune this stage in such a way that the accuracy of the identification system is not adversely affected. The width of the window should be selected such that the FNIR is very close to zero. In other words the number of genuine samples outside the window should be practically zero.

Once the window width is finalized, one can re-order the cascade to make the overall process more efficient. As noted before, the order of cascade does not affect the final accuracy. However, we use only a subset of the projections that has very low false rejects. We randomly generate 1500 projections and select best 500 projections based on the scores as mentioned before. The projections are ordered in the sequence of decreasing scores as computed by Equation 3. This would minimize the total amount of comparisons as the samples that are rejected in one projection is not considered in the following projections in the cascade.

B. Effect of Feature Representation

In our experiments, we use three different feature representations for the initial feature vector (before projection) for the purpose of comparison. The first one (referred to as **F1**) is a Gabor wavelet based texture feature that is popular in Iris as well as Palmprint recognition. The response is computed by convolving the image with the following kernel:

$$G(x, y, \theta, u, \sigma) = \frac{1}{2\pi\sigma^2} e^{\{-\frac{x^2+y^2}{2\sigma^2}\}} e^{\{2\pi\iota(uxcos\theta + uysin\theta)\}}, \tag{5}$$

where $\iota = \sqrt{-1}$;u is the frequency of the sinusoidal wave; θ controls the orientation of the kernel and σ is the standard deviation of the Gaussian envelope. Palmprint is represented using a 2048 ($32 \times 32 \times 2$) dimensional feature vector, while the iris was represented using 9600 ($240 \times 20 \times 2$) dimensional vector. In our case $\theta = \frac{\pi}{4}$ and $\sigma = 0.0916$. Another set of features that was proposed for iris indexing was DCT coefficients in various subbands [13]. After normalizing for pose and illumination variations using an adaptive histogram equalization, image is divided into non-overlapping 8×8 pixel blocks and are transformed to generate DCT coefficients.

The coefficients from each block belonging to a particular subband are grouped together. Energy value E_i of each subband S_i is obtained by summing up the square of coefficients as

$$E_i = \sum S_i(x, y)^2 \tag{6}$$

The feature vector consists of different energy values obtained from 10 subbands, resulting in a 10-dimensional feature vector. The image key consists of bin number corresponding to each subband. The bin numbers for each subband are combined together in increasing order of frequency. We refer to the DCT feature representation as **F2** from now on.

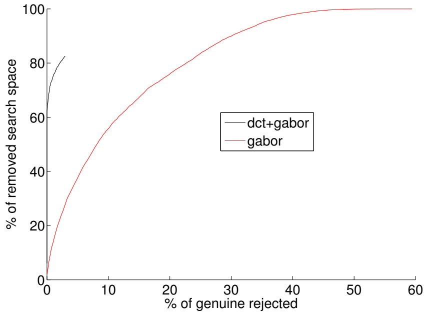

Fig. 2. Effect of F1 and F3 on filtering performance using only random projections.

Using only the Gabor response features and using a mixture of projections computed from LDA as well as random generation, we can prune 56% of the dataset with the genuine loss of 10%. However, if we use a concatenated feature vector of both Gabor Responses and DCT (referred to as **F3**). In case of palmprint images, we can prune the dataset upto 62.1% with no loss in genuine (see Figure 2).

C. Effect of Window Size

The size of the window selected also plays an important role in determining the accuracy of the system, if the size of the window is too small, we will be able to remove more of the gallery, but many genuine samples will also be removed during the process. However, if the window size is too large, it will not be able to reduce the search space considerably (see Figure 3). We need to select the optimum window size depending on the nature of the data.

IV. Experimental Results and Analysis

The experiments are performed on two different modalities: Palmprints and Iris. The PolyU database [1] of palmprint and CASIA database [2] of Iris were used for this purpose. For the palmprint, only those images in the dataset, where the complete palm was visible was considered. For iris, the entire database was considered irrespective of whether part of the iris was covered by the eyelids or not. The Palmprint dataset contains images of size 384×284 with 10 samples from each of 385 users. The iris database contains images of size 320×280 with 3 samples each from a total of 286 users. Tables I and II provides the Penetration rates achieved versus the FNIR for different databases, features and projection methods.

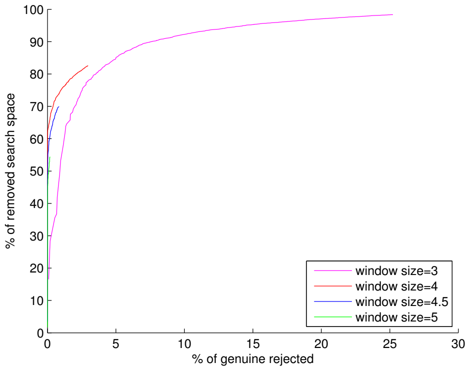

Fig. 3. Effect of window size on filtering using feature set F3.

Feature	Palmprint PolyU		Iris CASIA	
	Pen.	FNIR	Pen.	FNIR
F1	64.8%	10.3%	38.9%	21.2%
Pyramid Indexing	98.9%	0.15%	88.8%	5.4%
F2	-	-	35.6%	2.6%
F3	37.9%	0%	33.3%	10.0%

TABLE I
FNIR AND FILTERING RATES WITH VARIOUS FEATURES ON PALMPRINT AND IRIS DATASETS.

Feat.	Random		LDA		Combined	
	Pen.	FNIR	Pen.	FNIR	Pen.	FNIR
F1	65.0%	10.3%	46.1%	10.0%	42.4%	10.7%
F3	37.0%	0%	41.0%	5.1%	24.8%	1.0%

TABLE II
FNIR AND FILTERING RATES WITH VARIOUS METHODS OF SELECTION THE PROJECTIONS USING THE PALMPRINT DATASET.

A. Cost analysis

We now analyze the cost advantage of carrying out a filtering stage before explicit matching. Each stage of filtering would remove a part of the dataset from consideration, thereby improving the speed of the overall system. However, as the number of projections in the cascade increases, the returns starts diminishing, and at some point the cost of the projection and matching would override the cost advantage due to filtering. Figure 4 shows a plot of the reduction in search space (100-penetration) versus the FNIR for various lengths of the cascade. The green dots indicate the lengths of $1, 51, 101, \ldots, 451$. We note that after around 100 projections, the reduction in penetration rate is not significant.

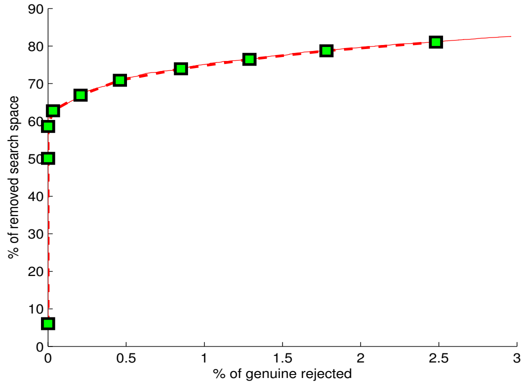

Fig. 4. Data pruned after each set of 50 projections, starting with 1. The improvement in pruning reduces as the number of projections increase.

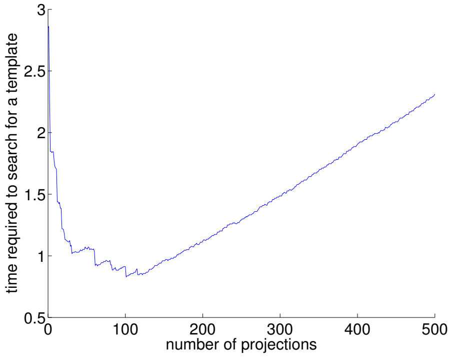

Fig. 5. Overall time taken (seconds) for identification as the number of projections in the cascade increases.

To compute the actual gain in speed, we carry out an experiment with a sample probe and the time taken for identification for various lengths of the cascade was determined. Figure 5 shows a graph between the overall time taken for identification (in seconds) and the number of projections in the cascade. This experiment was conducted on the PolyU Palmprint database. As expected, the returns of adding further filtering stages reduces and then reverses as the number of projections cross a limit (104). The time required for explicit comparison of a template against all samples in the database was around 2.86 seconds. However, as part of the samples are filtered out, the total time required for comparison decreases, and with a filtering pipeline of 104 random projections, the time required for an identification drops to 0.84 seconds. Note that the actual time will depend on the specific probe being used. However, the overall trend remains the same.

If the size of dataset is very large such that it cant be fit in memory, then it is divided into chunks. Each chunk is pruned independently. As the process directly lends to parallelization, each chunk can be pruned at the same time on a different machine.

V. Conclusions and Future Work

We have presented a generic approach to biometric filtering using cascaded filtering by random projections. The results show that we can reduce the search space by over 63% with no increase in the false non-identification rate (FNIR). The approach is flexible to use different feature sets and their combinations to carry out the projection. As each sample can be projected independently during the training, and hence the computational cost of inserting a new sample into the database is minimal. The approach also allows a high degree of parallelization or pipelined processing.

We are currently exploring the possibility of creating more complex filtration stages with formally characterized fitness functions.

References

[1] Polyu palmprint database. *http://www.comp.polyu.edu.hk/ biometrics*.

[2] Center for biometrics and security research. *http://www.cbsr.ia.ac.cn/IrisDatabase.htm*.

[3] R. Cappelli, D. Maio, D. Maltoni, and L. Nanni. A two-stage fingerprint classification system. *Proc. of ACM SIGMM workshop on Biometrics methods and applications*, Berkeley, CA Nov. 2003, pp. 95–99.

[4] S. Dasgupta. Experiments with random projection. *Proc. of Uncertainty in Artificial Intelligence*, Stanford CA, Jun. 2000, pp. 143–151.

[5] X. Fern and C. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. *International Conference on Machine Learning*, Washington ,DC Aug. 2003 pp. 186–193

[6] N. Goel, G. Bebis, and A. V. Nefian. Face recognition experiments with random projections. *Proc. of SPIE Conference on Biometric Technology for Human Identification*, Orlando, FL, Mar 2005.

[7] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Proc. of* 30^{th} *ACM Symposium on Theory of Computing*, Dallas, TX, May, 1998, pp 604–613,

[8] A. Jain and S. Pankanti. *Fingerprint classification and matching*. A. Bovik(ed.) Academic Press, April 2000.

[9] A. T. B. Jin. Cancellable biometrics and multispace random projections. *Conference on Computer Vision and Pattern Recognition Workshop(CVPRW'06)*, New York, NY, Jun 2006

[10] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. In *Contemporary Mathematics*, volume 26, pages 189–206, 1984.

[11] J. Kleinberg. Two algorithms for nearest-neighbor search in higher dimensions. *Proc. of ACM Symposium on Theory of Computing*, El Paso, Texas, May 1997, pp 599–608.

[12] F. Li and M. K. Leung. Hierarchical identification of palmprint using line-based hough transform. *Proc of International Conference of Pattern Recognition*, Hong Kong, Oct 2006 pp 149–152.

[13] H. Mehrotra, G. B. Srinivas, B. Majhi, and P. Gupta. Indexing iris biometric database using energy histogram of dct subbands. *Journal of Communication in Computer and Information Sciences*, 40, pp 194–204, 2009.

[14] A. Mhatre, S. Chikkerur, and V. Govindaraju. Indexing biometric databases using pyramid technique. *Proc. of Audio and Video-based Biometric Person Authentication*, NY, jul. 2005. pp 841-849

[15] C. Papadimitriou, P. Raghvan, H. Tamaki, and S. Vempala. Latent semantic analysis: A probabilistic analysis. *Proc. of ACM Symposium On the principles of Database Systems*, Seattle, jun 1998 pp 159–168.

[16] N. Ratha, K. Karu, S. Chen, and A. Jain. A real-time matching system for large fingerprint databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), Aug 1996.

[17] Z. Sun, Y. Wang, T. Tan, and J. Cui. Improving iris recognition accuracy via cascaded classifiers. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(3):pp 435–441, August 2005.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. of Computer Vision and Pattern Recognition*, Kauai, HI, Dec 2001, pp 511–518.

[19] J. Youa, W. Lia, and D. Zhang. Hierarchical palmprint identification via multiple feature extraction. *Pattern Recognition*, 35(4):847–859, April 2002.