A FAST FEATURE EXTRACTION METHOD

Jing Pan[1], *YanweiPang*[2], *Xuelong Li*[3], *Yuan Yuan*[4], *and Dacheng Tao*[5]

[1]Electronic Engineering Department, Tianjin University of Technology and Education, China
[2]School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China
[3]Birkbeck College, University of London
[4]School of Electronic Information Engineering, Aston University
[5]School of Computer Engineering, Nanyang Technology University, Singapore 639798

ABSTRACT

A fast subspace analysis and feature extraction algorithm is proposed which is based on fast Haar transform and integral vector. In rapid object detection and conventional binary subspace learning, Haar-like functions have been frequently used but true Haar functions are seldom employed. In this paper we have shown that true Haar functions can be successfully used to accelerate subspace analysis and feature extraction. Both the training and testing speed of the proposed method is higher than conventional algorithms. Experimental results on face database demonstrated its effectiveness.

Index Terms— Feature extraction, subspace analysis, fast Haar transform, integral vector.

1. INTRODUCTION

In many subspace analysis algorithms [1][2][3][4][5] [9][10], the basis vectors can be obtained by solving a standard or generalized eigenvalue decomposition problem. The solutions of many other subspace analysis algorithms have to be solved by iterative manner since their objection functions are not convex.

Given a basis vector $\mathbf{u} \in \mathbb{R}^N$ obtained by one of the above algorithms and a high-dimensional vector $\mathbf{x} \in \mathbb{R}^N$, the corresponding feature $y \in \mathbb{R}$ is computed by direct inner-product $y = \mathbf{u}^T \mathbf{x}$. This inner-product involves N floating-point multiplications and $N-1$ floating-point additions. Usually N is very large and thus the computation of the direct inner product is computational intensive. In most applications, feature extraction is only one of the steps to perform a computer vision or data mining task and many other steps have to compete with the feature extraction step for the limited computational resources. Therefore, it is desirable to reduce the computational cost of the feature extraction process.

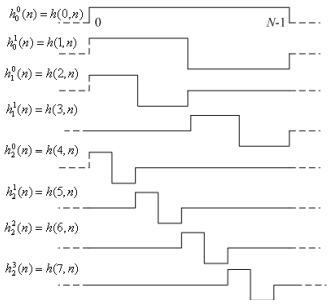

Fig. 1. The first 8 Haar basis functions.

To extract features with less computational cost, we propose in this paper an effective feature extraction algorithm which we call FHT-PCA. Not only the feature extraction process (the process of obtaining y) but also the training process are very fast. This characteristic makes the proposed method applicable to lots of cases where subspace-based feature extraction is involved. The advantages of the FHT-PCA comes from our introducing fast Haar transform (FHT) [6] into feature extraction and subspace learning.

A related work is B-PCA [7]. In B-PCA, Haar-like functions in stead of true Haar functions are selected from a very large dictionary. The selection process is too low to be applied for mid-size images let alone large-size images.

2. FAST HAAR TRANSFORM

The proposed algorithm is fast because fast Haar transform (FHT) [6] and integral vector are used.

The basis functions of the Haar transform a complete set of orthonormal rectangle basis functions which exhibit the unique characteristic of having both global and local function properties (see Fig. 1). In Fig. 1, $h_i^j(n)$ explicitly reflects the scale and position of the function by specifying i and j

Given a signal $\mathbf{x} = [x(0)\ x(1) \cdots x(N-1)] \in \mathbb{R}^N$, the Haar transform is given by

$$y(k) = \sum_{n=0}^{N-1} x(n)h(k,n), \qquad k = 0,...,N-1 \ . \tag{1}$$

The outcome of the Haar transform is $\mathbf{y} = [y(0)\ y(1) \cdots y(N-1)]$ where $y(k)$ is the coefficient of $\mathbf{x}$ associated to the Haar function $h(k,n)$.

What is used in the proposed algorithm is fast Haar transform (FHT) [6]. The FHT needs only $2\times(N-1)$ addition operations to get $\mathbf{y}$ while the ordinary Haar transform requires $N \times N$ floating point multiplications and $N\times(N-1)$ addition operations. Most importantly, the FHT neither needs to construct the Haar functions nor store them. If an image of size 128×128 is vectorized to a long vector, the storage size of the corresponding Haar functions is 1GByte for the ordinary Haar transform. By contrast, FHT needs only to allocate memory for the signal and an intermediate vector whose size is identical to that of the signal.

3. FAST FEATURE EXTRACTION BASED ON FHT

3.1. Fast Projection onto a Given Basis Vector

Assume a basis vector $\mathbf{u} \in \mathbb{R}^N$ is given by some subspace learning algorithm such as PCA or LDA. The signal in the original high-dimensional data space is $\mathbf{x} \in \mathbb{R}^N$. The traditional process of extracting the feature $y \in \mathbb{R}$ of $\mathbf{x}$ corresponding to $\mathbf{u}$ is to compute y by the following inner product operation:

$$y = \mathbf{u}^T\mathbf{x} = \sum\nolimits_{i=0}^{N-1} u_i x_i \ . \tag{2}$$

This requires N floating point multiplications and $N-1$ floating point additions. We propose to reduce the computational cost by utilizing Haar transform and integral vector.

The Haar functions described in Section 2 can be expressed as vectors in $\mathbb{R}^N$. So in this section we will denote the Haar function $h(i,n)$ by the vector $\mathbf{v}_i \in \mathbb{R}^N$ with $i = 0,...,N-1$. The general form of $\mathbf{v}_i$ (illustrated in Fig. 2) is

$$v_{ij} = \begin{cases} a & l \le j < m \\ -a & m \le j < r \\ 0 & elsewise \end{cases}, \tag{3}$$

where v_{ij} is the j th entry of $\mathbf{v}_i$ and a is the magnitude of the Haar function. l, m, and r specify the locations of discontinuity of the Haar function. The Haar transform of $\mathbf{u}$ is

$$c_i = \mathbf{v}_i^T\mathbf{u}, \quad i = 0,...,N-1 \ . \tag{4}$$

By discarding the small-valued coefficients and reserving the largest K coefficients, $\mathbf{u}$ can be approximated by

$$\mathbf{u} = \sum_{i=1}^{N-1} c_i\mathbf{v}_i \approx \tilde{\mathbf{u}} = \sum_{i=1}^{K-1} c_i\mathbf{v}_i, \quad c_0 > c_1 > \cdots > c_{K-1}, \ K \ll N \ . \tag{5}$$

The condition $K \ll N$ is important for compact representation and for our fast feature extraction algorithm.

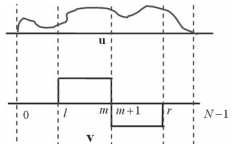

Fig. 2. The bottom row shows the general form of a Haar function $\mathbf{v}$ while the top row is a basis vector $\mathbf{u}$. The length of $\mathbf{v}$ and $\mathbf{u}$ are both $N-1$.

In the following part we will show how one can use (5) to fasten feature extraction. Substituting (5) into (2) yields

$$y = \mathbf{x}^T\mathbf{u} \approx \mathbf{x}^T\tilde{\mathbf{u}} = \sum_{i=0}^{K-1}\left(c_i\left(\mathbf{x}^T\mathbf{v}_i\right)\right) \tag{6}$$

Because of the many constant and zero entries in $\mathbf{v}_i$ (see Fig. 2), the item $\mathbf{x}^T\mathbf{v}_i$ in (6) is potentially very fast to compute. Equation (7) indicates that $\mathbf{x}^T\mathbf{v}_i$ can be calculated by using many addition operations and one multiplication operations:

$$\mathbf{x}^T\mathbf{v}_i == a\left(\sum_{i=l}^{m} x_i - \sum_{i=m+1}^{r} x_i\right) \tag{7}$$

To reduce the number of additions in (7), we define the integral vector $it(\mathbf{x},i)$ as

$$it(\mathbf{x},i) = \sum_{j=0}^{i} x_j \ . \tag{8}$$

It then follows

$$\sum_{i=l}^{m} x_i = it(\mathbf{x},m) - it(\mathbf{x},l-1) \tag{9}$$

and similarly

$$\sum_{i=m+1}^{r} x_i = it(\mathbf{x},r) - it(\mathbf{x},m) \ . \tag{10}$$

So (7) can be formulated by integral vector as

$$\begin{aligned}\mathbf{x}^T\mathbf{v}_i &= a\left(\sum_{i=l}^{m} x_i - \sum_{i=m+1}^{r} x_i\right) \\ &= a\left[\left(it(\mathbf{x},m) - it(\mathbf{x},l-1)\right) - \left(it(\mathbf{x},r) - it(\mathbf{x},m)\right)\right] \ . \end{aligned} \tag{11}$$

From (11) one can find that the inner product of $\mathbf{x}$ and $\mathbf{v}_i$ can be obtained by 3 additions and 1 multiplication. Substitute (11) into (6), one get

$$y \approx \sum_{i=0}^{K-1}\left(c_i\left(\mathbf{x}^T\mathbf{v}_i\right)\right)$$

$$= \sum_{i=0}^{K-1}\left(c_i\left(a_i\left[\left(it(\mathbf{x},m) - it(\mathbf{x},l-1)\right) - \left(it(\mathbf{x},r) - it(\mathbf{x},m)\right)\right]\right)\right) \ . \ (1$$

2)

Equation (12) tells that extracting a feature requires $3\times K+(K-1)=4\times K-1$ additions and $2\times K$ multiplications. By contrast, traditional method that computes the projection by element-wise inner product (i.e. (2)) needs $N-1$ additions and N multiplications. Take a typical configuration $K=50$ and $N=128\times128$ for example, the proposed method needs 199 addition operations and 100 multiplication operations, while traditional method needs 16,383 additions and 16384 multiplications. If d features to be extracted, the computational cost ratio s of the traditional method against our method is

$$s=\frac{d\times\left(N\times m+(N-1)\times p\right)}{(N-1)\times p+\sum_{i=1}^{d}\left[\left(4\times K_i-1\right)\times p+\left(2\times K_i\right)\times m\right]}, \quad (13)$$

where p and m are the numbers of machine cycles occupied by performing an addition operation and a multiplication operation respectively, K_i is the number of the Haar functions used to approximate the i th basis vector $\mathbf{u}_i$. In the denominator of (13), the item $(N-1)\times p$ stands for the computational cost of computing the integral vector $it(\mathbf{x},\cdot)$. The integral vector is computed once and the result can be used for extracting all the K features. The efficiency of computing the integral vector results from the following iterative formula:

$$\begin{aligned} &it(\mathbf{x},0)=x_0 \\ &it(\mathbf{x},i)=it(\mathbf{x},i-1)+x_i, \quad i=1,...,N-1 \end{aligned}. \quad (14)$$

Throughout the paper, it is assumed that multiplication operation spends much more time than addition operation, i.e. $m>p$. This is almost always true for micro controller unit (MCU). Even though $m=p$, it also holds that the denominator of (13) is smaller than the nominator (i.e. $s>1$) as long as K_i is small, which implies the computational superiority of our algorithm over the traditional method.

Now the question is how to determine K_i. Instead of assigning a fixed value for K_i, we propose to increase the value of K_i from 1 to the value so that the angle $angle(\mathbf{u},\tilde{\mathbf{u}})$ between $\mathbf{u}$ and $\tilde{\mathbf{u}}$

$$angle(\mathbf{u},\hat{\mathbf{u}})=\frac{180}{\pi}\frac{\mathbf{u}^T\tilde{\mathbf{u}}}{\left(\|\mathbf{u}\|\right)\left(\|\tilde{\mathbf{u}}\|\right)} \quad (15)$$

is just equal to or less than a pre-defined threshold θ such as 30 degree. The meaning of $\mathbf{u}$ and $\tilde{\mathbf{u}}$ is the same as that in (5).

3.2. Subspace Learning

In the above subsection, it is assumed that the basis vector $\mathbf{u}$ is given beforehand. In this subsection, we describe how to obtain the basis vectors.

Let $\mathbf{X}=[\mathbf{x}_1\ \mathbf{x}_2\cdots\mathbf{x}_M]$ be the M training samples with $\mathbf{x}_i\in\mathbb{R}^N$. We adopt the framework in [7] to iteratively compute the d basis vectors $\mathbf{U}=[\mathbf{u}_1\cdots\mathbf{u}_d]$. Each basis vector is obtained by applying a subspace learning algorithm on the reconstruction error matrix and outputting the most important basis vector $\mathbf{u}$ and finally approximating $\mathbf{u}$ with the K Haar functions as stated in last subsection. In the experimental section, we choose PCA as the subspace learning algorithm.

Before iteration, the reconstruction error matrix $\mathbf{E}$ is initialized to be the training matrix $\mathbf{X}$. At the beginning of t th iteration, $t-1$ basis vectors $\tilde{\mathbf{U}}=[\tilde{\mathbf{u}}_1\cdots\tilde{\mathbf{u}}_{t-1}]$ are available, the reconstruction error matrix is then updated by

$$\begin{aligned} &\mathbf{E}=\mathbf{X}-\tilde{\mathbf{X}} \\ &\tilde{\mathbf{X}}=\tilde{\mathbf{U}}\left(\tilde{\mathbf{U}}^T\tilde{\mathbf{U}}\right)^{-1}\tilde{\mathbf{U}}^T\mathbf{X} \end{aligned}. \quad (16)$$

The second row of (16) is the reconstruction formula for non-orthogonal basis vectors.

Because we use FHT to compute the coefficients of Haar functions and then approximate the corresponding basis vector with the largest K Haar functions, the whole training process is very faster than that of B-PCA [7].

4. EXPERIMENTAL RESULTS

We conducted experiments on the ORL database. For training, we randomly selected 5 images per subject, and used the remaining images for testing.

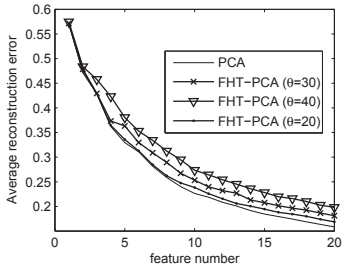

Fig. 3. The average reconstruction error.

The proposed method FHT-PCA is PCA-based. Because PCA is optimal in the sense of minimum reconstruction error, we how how the average reconstruction error e

$$e=\frac{1}{M}\sum\nolimits_{i=1}^{M}\|\mathbf{x}_i-\tilde{\mathbf{x}}_i\| \quad (17)$$

varies with respect to the number of basis vectors used. The results of FHT-PCA and PCA are shown in Fig. 3. As can be seen, as θ decreases the FHT-PCA approaches PCA. But if θ is too small, the number of Haar functions will be very large and the computation cost will increase. In the

following experiments, we will choose $\theta = 30$ as a good trade-off. Fig. 4 shows that the angle θ (see eq. (15)) between $\mathbf{u}$ and $\tilde{\mathbf{u}}$ decreases quickly w.r.t. the number of selected Haar functions.

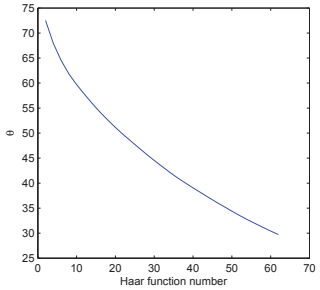

Fig. 4. The angle θ between $\mathbf{u}$ and $\hat{\mathbf{u}}$ decreases quickly w.r.t. the number of selected Haar functions.

Then we examine the recognition performance of the FHT-PCA. Fig. 5 shows the recognition results. It is observed from Fig. 5 that the recognition performance of FHT-PCA approaches that of PCA very well. This implies that the recognition performance is degenerated little by approximating the basis vector with small number of Haar functions.

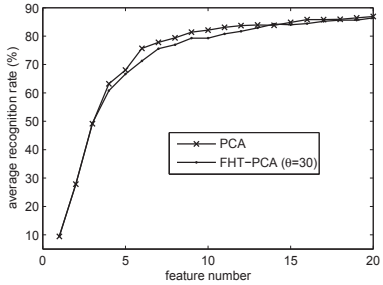

Fig. 5. Recognition rates of FHT-PCA and PCA

Fig 3 and Fig. 5 have shown that both the representation capacity and recognition accuracy of the FTH-PCA are comparable with PCA. But these properties of the FHT-PCA are meaningful only if its computational cost is more economical than that of PCA. The computational cost ratio of PCA against FHT-PCA is shown in Fig. 6. Fig. 6 shows that the computational cost of PCA is several times larger than that of FHT-PCA.

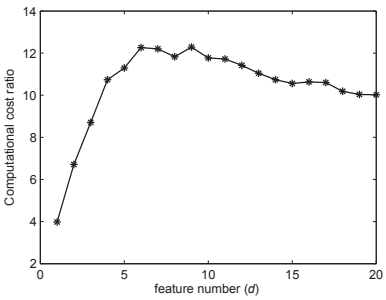

Fig. 6. The computational cost ratio of PCA against FHT-PCA.

These experimental results have given evidences that FHT-PCA is much faster than PCA without significant decreasing the representation and recognition performance.

5. CONCLUSIONS

We introduced an effective subspace analysis and feature extraction algorithm based on fast Haar transform and integral vector. In face detection and recognition, Haar-like functions have been frequently used but true Haar functions are seldom employed. In this paper we have shown that true Haar functions can be successfully used to accelerate subspace analysis and feature extraction. Owing to high-speed of fast Haar transform, the proposed algorithm is very fast for training. By utilizing integral vector, it is also very fast for extracting features.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for the valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China under grant no. 60605005 and the Open Project Program of the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

REFERENCES

[1] Dong Xu, Shuicheng Yan, et al., "Enhanced marginal Fisher analysis for human gait recognition and content based image retrieval," *IEEE Trans. Image Process.*, 2007.

[2] Dacheng Tao, Xuelong Li, Xingdong Wu, and Steve J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 10, pp. 1700-1715, 2007.

[3] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 27, no. 3, pp. 328-340, 2005.

[4] Deng Cai, Xiaofei He, and Jiawei Han, "SRDA: An efficient algorithm for large scale discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 1-12, 2008.

[5] Matthew Turk and Alex Pentalnd, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[6] Peter R. Roeser and M. E. Jernigan, "Fast Haar transform," *IEEE Transactions on. Computer*, vol. 3-31, no. 2, pp. 175-177, 1982.

[7] F. Tang, R. Crabb, and H. Tao, "Representing images using nonorthogonal Haar-like bases," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2120-2134, 2007.

[8] ORL face database, http:// www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, 2008.

[9] Y. Fu, L. Cao, G. Guo, and T. S. Huang "Multiple feature fusion by subspace learning," *ACM International Conference on Image and Video Retrieval*, pp. 127-134, 2008.

[10] Y. Fu, S. Yan and T. S. Huang, "Correlation Metric for Generalized Feature Extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 30, no. 12, 2008.