

Received April 19, 2020, accepted May 8, 2020, date of publication May 26, 2020, date of current version June 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997857

A Novel Image Description With the Stretched Natural Vector Method: Application to Face Recognition

WENHUI DONG^{1,2}, AND STEPHEN S.-T. YAU¹, (Fellow, IEEE)

¹Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China

²School of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710062, China

Corresponding author: Stephen S.-T. Yau (yau@uic.edu)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 11961141005, and in part by the Tsinghua University Start-Up Fund and Tsinghua University Education Foundation Fund under Grant 042202008.

Dedicated to Professor Roger W. Brockett on the occasion of his 82 birthday

ABSTRACT Nowadays there is growing research interest in designing high performance algorithms for automatic facial recognition systems, and an efficient computational approach is required. Accurate face recognition, however, is difficult due to facial complexity. In this paper, we propose a novel and efficient facial image representation named the Stretched Natural Vector (SNV) method which is defined on the intensity values in a grayscale image matrix, where each entry in an intensity matrix records the level of gray at a single pixel in a $m \times n$ array. We prove that the SNV defined in this context can distinguish photo matrices in strict one-to-one fashion. This is to say it is theoretically possible to fully recover a grayscale image matrix from the corresponding complete SNV. Experiments on a number of datasets demonstrate that our truncated SNV method compares favorably both in recognition accuracy and efficiency (measured in wall-clock time) against “Full-Pixel” algorithm, Principal Component Analysis (PCA) method, and even its widely used variants – two dimensional PCA (2DPCA) method and two dimensional Euler PCA (2D-EPCA) method.


INDEX TERMS Stretched natural vector (SNV), face recognition, principal component analysis (PCA).

I. INTRODUCTION

Face recognition is one of the most important methods for biometrical recognition. It has wide applications, including criminal identification, security systems, image and film processing, and human-computer interaction. Over the past few years, people from many different fields of science and engineering have addressed the challenging tasks of computer vision and real time pattern recognition. The task of visually matching images of the same person, which are obtained from different cameras distributed over non-overlapping locations of potentially substantial distances and time differences, is called person re-identification [11]. Solving the re-identification problem is primarily focused on the feature representation and distance measure. In [29], the authors designed and obtained a more effective metric for the re-identification problem. However, it is extremely

challenging to extract effective features from different facial images under the influences of illumination, pose, and facial expression. Improved accuracy remains a goal and mistakes occur in all applications, whether the work is done by machines or humans or both.

Face recognition mainly involves three parts, including pre-processing, feature extraction and recognition. Principal Component Analysis (PCA) [25] is one of the classic feature extraction methods based on statistical characteristics. The main idea of PCA method is to convert the matrix of the original images into vectors, through calculating the total scattering matrix of the image vector and getting a set of biggest standard orthogonal vector as the optimal projection axis. As we all know, face image infers high dimensions, conventional PCA spends much computational costs to handle high-dimensional data. Therefore, based on the effectiveness of PCA in feature extraction, researchers are tremendously interested in developing its variations.

The associate editor coordinating the review of this manuscript and approving it for publication was Ananya Sen Gupta .

As opposed to PCA, two-dimensional PCA (2DPCA) method [30] is based on two-dimensional (2D) image matrices rather than one-dimensional (1D) vectors. Instead of transforming the image matrix into a vector and then computing its covariance matrix, an image covariance matrix in 2DPCA method is constructed directly using the original image matrices, and correspondingly, its optimal projection axis is derived for image feature extraction. Compared with PCA, this approach avoids the damage to the spatial structure distribution of original face image, but with less time cost obtained a higher recognition accuracy. In [17], a modified method called modular 2DPCA method is presented in which the face images are divided into smaller images and the PCA method is applied on each of them. However, 2DPCA and its modifications extract the features from the original image matrices and need more coefficients for image representation than PCA, which is time-consuming. Diagonal principal component analysis (DiaPCA) [34] as a novel subspace method in contrast to standard PCA, directly seeks the optimal projective vectors from diagonal face images without image-to-vector transformation. While in contrast to 2DPCA, it reserves the correlations between variations of rows and those of columns of images. The application of wavelet transform in face recognition have been discussed in numerous papers and research articles. In [19], they introduce the application of digital curvelet transform in conjunction with different dimensionality reduction tools, particularly applying to the problem of facial feature extraction from 2D images. As the l_2 -norm employed by standard PCA is not robust to outliers, researchers in [16] propose a kernel PCA method, which is called Euler PCA (EPCA). In particular, they utilize a robust dissimilarity measure based on the Euler representation of complex numbers and such complex representation induces much smaller kernel matrix and principal subspaces. Further, combining the advantages of 2DPCA with EPCA, researchers in [27] introduce a two-dimensional Euler PCA (2D-EPCA) algorithm, which leans projection matrix on the 2D pixel matrix of each image without reshaping it into 1D long vector, and uncovers nonlinear relationships among features by mapping data onto complex representation. Although the approaches have achieved promising results, most of them cannot obtain meaningful components of facial images since the feature matrix is extracted by the PCA or PCA-based extension methods. In addition, 2DPCA or its modification methods [5]–[7] are proposed to reduce the computational cost of the standard PCA algorithm, but the performance of 2DPCA in reducing computational complexity and recognition rate is not satisfying.

Some other powerful tools are used for data reduction and feature extraction, such as LDA [4], [15], and Independent Component Analysis (ICA) [3], [31]. Recently, other dimension reduction techniques [9], [10] have been proposed. These are some local features extraction methods, not like as PCA and its variants as holistic methods. Researchers investigate the representation of face images by means of local binary

pattern features that are scattered in [1], [8], [12], [21], [22], [24], [32], [33]. In [2], a novel descriptor based on local binary pattern texture features extracted from local facial regions is presented.

In this paper, a novel approach coined the Stretched Natural Vector (SNV) method is proposed for image representation, which is based on two-dimensional grayscale image matrix directly. As for the construction of the SNV, the first group of the components in SNV are the quantities of the grayscales, which define the number of all level grayscales, respectively. The second group of components in SNV are the average locations across rows and columns in the image matrix of all level grayscales, respectively. Meanwhile, the third group of components in SNV are the normalized higher order central moments, which can represent the distribution of the grayscales with respect to their corresponding average locations, respectively. The physical meaning of an image about counts, locations, and distributions of the grayscales are fully considered in the construction of SNV, i.e., the first, second and third groups of components.

The proposed method can reduce data dimensions and computational loads without discarding too much information. We have proven that the SNV defined in this context can distinguish image matrices in strict one-to-one fashion. Concretely to say, it is theoretically possible to fully recover a grayscale pixel matrix from the corresponding complete SNV.

A series of experiments are performed on three face image datasets for the purpose of testing the SNV method. At the same time, the Euclidean metric is widely used on the dataset to find a similar image of a given query image by measuring the distance between those images. We use a truncated SNV method to match face images to people in order to compute recognition accuracy and time consuming, comparing with the basic PCA method, its variants–2DPCA and 2D-EPCA methods and the “Full-Pixel” algorithm, which is detailed in (18). In addition, Face recognition experiments on the commonly used ORL dataset, and two more face datasets which we have detailed in section IV, show the superiority of our proposed method over classical PCA method, 2DPCA method, 2D-EPCA method, and “Full-Pixel” algorithm in terms of recognition accuracy, and also in reduced CPU time. Along with the training sample numbers increasing, the experimental results also indicate that the SNV method is more computationally efficient than PCA algorithm, and owns an acceptable and compatible computational time costs.

The remainder of this paper is organized as follows. The proposed SNV method is described in section II. In section III we give our key contribution, namely a mathematical proof that the assignment of an SNV to a image pixel matrix is one-to-one. In section IV we introduce truncated SNV and compare the recognition accuracy of our truncated SNV method to classical PCA, 2DPCA, 2D-EPCA and “Full-Pixel” algorithms on the three referred datasets. Finally, we state our conclusions in the last section.

II. CONSTRUCTION OF THE STRETCHED NATURAL VECTOR (SNV) ON A TWO-DIMENSIONAL IMAGE PIXEL MATRIX

A grayscale image having $m \times n$ pixels, with 256 possible intensities at each pixel (i.e., with 8 bits there are $2^8 = 256$ possible values) determines an $m \times n$ matrix Q with entries $q(i, j) \in K = \{0, 1, 2, 3, \dots, 255\}$. The perspective in the construction of SNV is that the information is captured by the k -valued matrix entry locations for all values $k \in K$. Thus, we say that the SNV is determined by a two-dimensional distribution on a matrix.

Denote by n_k the cardinality of $q^{-1}(k)$. Thus for $k \in K = \{0, 1, 2, \dots, 255\}$, n_k is the total number of k -valued pixels in the grayscale image. Then $N = mn = \sum_{k=0}^{255} n_k$ is the total number of pixels. Furthermore, let $q^{-1}(k) = \{(i_{s,k}, j_{s,k}) | s = 1, 2, \dots, n_k\}$ be the set of all k -valued pixel positions.

Here is a small example of a hypothetical photo intensity matrix.

$$\begin{pmatrix} 3 & 25 & 5 & 34 & 45 & 34 & 255 & 0 \\ 4 & 34 & 32 & 56 & 45 & 57 & 234 & 189 \\ 23 & 45 & 56 & 77 & 34 & 1 & 0 & 6 \end{pmatrix}_{3 \times 8} \quad (1)$$

In the example matrix above, $q^{-1}(34)$ consists of four pixels. The value $k = 34$ shows up as following

$$\begin{aligned} &\{(i_{1,34}, j_{1,34}), (i_{2,34}, j_{2,34}), (i_{3,34}, j_{3,34}), (i_{4,34}, j_{4,34})\} \\ &= \{(1, 4), (1, 6), (2, 2), (3, 5)\}. \end{aligned}$$

For an actual photo intensity matrix, there would be 256 sets capturing similar information. Now we describe an SNV corresponding to a given photo matrix. It is easily seen that a similar SNV could be constructed from any two-dimensional matrix with values in a finite set. Theoretically, there is a big messy mathematical object even in the case of an integer-valued or a real-valued matrix with infinitely many possible entry values, but it will be of some importance for us that the set K is finite.

- 1) The first group of components in the SNV are the 256 counts $(n_0, n_1, \dots, n_{255})$. They're all non-negative integers bounded by the size of the matrix and some of them could be zero. In our little example above, $n_{34} = 4$ and most of the n_k 's are 0 since most intensity values don't show up in the matrix.
- 2) The second group of components in the SNV are mean pixel locations $\vec{\mu}_k = (\mu_{1,k}, \mu_{2,k})$ for intensity values $k = 0, 1, \dots, 255$.

$$\mu_{1,k} := \frac{\sum_{s=1}^{n_k} i_{s,k}}{n_k} = \frac{i_{1,k} + i_{2,k} + \dots + i_{n_k,k}}{n_k}, \quad (2)$$

$$\mu_{2,k} := \frac{\sum_{s=1}^{n_k} j_{s,k}}{n_k} = \frac{j_{1,k} + j_{2,k} + \dots + j_{n_k,k}}{n_k}, \quad (3)$$

where $(i_{s,k}, j_{s,k}) \in q^{-1}(k)$, for all $s = 1, 2, \dots, n_k$.

Let's look at our example, where $\vec{\mu}_{34} = (\mu_{1,34}, \mu_{2,34}) = (\frac{1+1+2+3}{4}, \frac{4+6+2+5}{4}) = (\frac{7}{4}, \frac{17}{4})$. Define $\vec{\mu}_k := (0, 0)$ if $n_k = 0$. For all other k 's, these vector components are positive rationals and not necessarily

integers. The SNV will have components from $\vec{\mu}_0$ through $\vec{\mu}_{255}$.

- 3) The third group of parameters that we include in the SNV are the normalized higher order central moments. There is a set of D 's for each $k \in K$. Let $D_{k,0,0} := 0$ for $k \in K$. For any other exponent pair (r, s) , we define

$$D_{k,r,s} := \sum_{t=1}^{n_k} \frac{(i_{t,k} - \mu_{1,k})^r \cdot (j_{t,k} - \mu_{2,k})^s}{(n_k)^{r+s} \cdot N^{r+s-1}} \quad (4)$$

where $k \in K$, r is the arbitrary non-negative integer, and $s = 0, 1, 2, \dots, n_k$.

Plugging $r = 0$ and $s = 0$ into the (4) yields only normalized counts $n_k \cdot N$ which are already captured. $D_{k,1,0}$ and $D_{k,0,1}$ turn out to be zero, a property of the location means $\mu_{1,k}$ and $\mu_{2,k}$. Thus central moments of combined degree 0 and 1 can be omitted. Alternatively, we could keep $D_{k,0,0}$, $D_{k,1,0}$, $D_{k,0,1}$, omitting the n_k 's and $\vec{\mu}_k$'s. However, that we keep the counts and location means just to make (4) easy to write. The SNV will have components $D_{0,2,0}, \dots, D_{255,2,0}$, then $D_{0,1,1}, \dots, D_{255,1,1}$, then $D_{0,0,2}, \dots, D_{255,0,2}$, then D 's with combined $r + s$ degree 3, and so forth.

After a bit of reordering, the SNV is given as follows with D 's ordered lexicographically starting at $k = 0$ with degree two.

$$\begin{aligned} &< n_0, \mu_{1,0}, \mu_{2,0}, D_{0,0,2}, D_{0,1,1}, D_{0,2,0}, \dots \\ & n_1, \mu_{1,1}, \mu_{2,1}, D_{1,0,2}, D_{1,1,1}, D_{1,2,0}, \dots \\ & \dots \\ & n_{255}, \mu_{1,255}, \mu_{2,255}, D_{255,0,2}, D_{255,1,1}, D_{255,2,0}, \dots > \end{aligned} \quad (5)$$

Obviously, higher central moments converge to 0 for a random generated distribution matrix since for any given k ,

$$\begin{aligned} D_{k,r,s} &= \sum_{t=1}^{n_k} \frac{(i_{t,k} - \mu_{1,k})^r \cdot (j_{t,k} - \mu_{2,k})^s}{(n_k)^{r+s} \cdot N^{r+s-1}} \\ &\leq \sum_{t=1}^{n_k} \frac{\max_{\{t \in \{1, 2, \dots, n_k\}\}} |i_{t,k}|^r \cdot |j_{t,k}|^s}{(n_k)^{r+s} \cdot N^{r+s-1}} \\ &\leq \frac{\max_{\{t \in \{1, 2, \dots, n_k\}\}} |i_{t,k}|^r \cdot |j_{t,k}|^s}{(n_k)^{r+s-1} \cdot N^{r+s-1}} \\ &\leq \frac{N^r \cdot N^s}{(n_k)^{r+s-1} \cdot N^{r+s-1}} = \frac{N}{(n_k)^{r+s-1}}, \end{aligned} \quad (6)$$

where $N = \sum_{k=0}^{255} n_k$.

It is clear that $n_k \geq 1$, otherwise, there is no any k -th grayscale distributed in the matrix.

From the viewpoint of probability, suppose that the expectation value of any level grayscale is $n_k = N/256$ (uniform distribution) for an image with the total stencil points N of the given distribution matrix.

Actually, for the given distribution matrix in m -by- n , we naturally get the total entries of the matrix is $N := mn$. Therefore,

$$\lim_{r+s} \frac{N}{(n_k)^{r+s-1}} = \lim_{r+s} \frac{N}{(N/256)^{r+s-1}} = \lim_{r+s} \frac{256^{r+s-1}}{N^{r+s-2}}. \quad (7)$$

Clearly, this limit can converge to 0 as $r + s$ increases. Specifically, for example, as for the given distribution matrix in 196-by-196, which has $N = 2^{16}$, the limit of (7) is shown in Tabel 1.

TABLE 1. The decay of higher central moments for the grayscale.

$r + s$	$\lim_{r+s} \frac{N}{(n_k)^{r+s-1}}$
3	1.7060
4	0.0114
5	7.5757×10^{-5}
6	5.0484×10^{-7}
7	3.3642×10^{-9}

That is, evidently, the higher central moments converge to zero very quickly as $r + s$ increases.

We will show in the next section that the information in the SNV is enough to theoretically determine the entire grayscale photo matrix. In section IV we will see that there is practical value in the truncated ENV which discards moments of combined degree three or higher.

III. RELATION BETWEEN A TWO-DIMENSIONAL MATRIX AND ITS CORRESPONDING SNV

A crucial contribution of this paper is that different photo matrices determine different SNVs, which is completely stated in Theorem 1.

The basic idea is that for each k , a photo determines a set of row locations $\{i_1, \dots, i_{n_k}\}$, which are roots of a symmetric polynomial whose coefficients would have to match those from another photo having the same SNV, because the coefficients are functions of some of the SNV components. The difficulty is that more is required than just showing the SNV determines the set of row locations and the set of column locations for both photos. It must be shown that the exact locations (i.e., the (i, j) row-column pairs) are common to both photos having the same SNV. Namely, it is easier to know the set of k -valued locations in one direction than it is to know them in both directions. The main theorem is presented as following.

Theorem 1: Suppose the finite set $K = \{0, 1, 2, \dots, 255\}$ has a distribution on the two-dimensional $m \times n$ matrix, such as that shown on (1), then the corresponding SNV determines the distribution, i.e., the SNV determines all the matrix entries.

Before we give the complete proof of the main theorem, which can be referred to Appendix, we need to state several lemmas. Firstly, Newton’s identities [18] (also known as the Newton-Girard formula) relate different types of symmetric polynomials, namely power sums and elementary symmetric polynomials. Evaluated at the roots of a monic polynomial \mathbf{P} in one variable, they allow expressing the sums of the k -th powers of all roots of \mathbf{P} (counted with their multiplicity) in terms of the coefficients of \mathbf{P} , without actually finding those roots.

The following lemma is the full statement.

Lemma 2 ([18]): Let x_1, x_2, \dots, x_n be variables, and

$$p_k(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i^k = x_1^k + x_2^k + \dots + x_n^k,$$

for $k \geq 1$. Namely, p_k is the sum of k^{th} powers of the variables x_1, \dots, x_n . Define $e_k(x_1, x_2, \dots, x_n)$ as the elementary symmetric polynomial, namely the sum of all distinct products of k distinct variables), for $k \geq 0$ shown below,

$$\begin{aligned} e_0(x_1, x_2, \dots, x_n) &= 1, \\ e_1(x_1, x_2, \dots, x_n) &= x_1 + x_2 + \dots + x_n, \\ e_2(x_1, x_2, \dots, x_n) &= \sum_{1 \leq i < j \leq n} x_i x_j, \\ &\dots \dots \\ e_n(x_1, x_2, \dots, x_n) &= x_1 x_2 \dots x_n, \\ e_k(x_1, x_2, \dots, x_n) &= 0, \text{ for } k > n. \end{aligned} \tag{8}$$

Then Newton’s identities can be stated as:

$$k e_k(x_1, x_2, \dots, x_n) = \sum_{i=1}^k (-1)^{i-1} e_{k-i}(x_1, x_2, \dots, x_n) \cdot p_i(x_1, x_2, \dots, x_n) \tag{9}$$

for all $n \geq 1$ and $k \geq 1$.

Also, one has

$$0 = \sum_{i=k-n}^k (-1)^{i-1} e_{k-i}(x_1, x_2, \dots, x_n) \cdot p_i(x_1, x_2, \dots, x_n) \tag{10}$$

for all $k > n \geq 1$.

Lemma 3: Let \mathbb{Z}^+ be a set of all positive integers, given a fixed element in \mathbb{Z}^+ denoted by N , i.e., $N \in \mathbb{Z}^+$, for arbitrary elements $m, n, a, b \in \mathbb{Z}^+$, and $m \leq N, n \leq N, a < b \leq N$, then we have

$$a^L \cdot m < b^L \cdot n, \tag{11}$$

if $L \in \mathbb{Z}^+$ and $L \geq \left\lceil \log_{\frac{b}{a}} \frac{N}{a} \right\rceil + 1$.

Proof: Let $c := \frac{b}{a}$ and we can easily know that $c > 1$ since $a < b$. Therefore, the function \log_c^x is an increasingly monotonous function with respect to the variable x .

For $m \leq N$ and $n \leq N$, we know that $\frac{m}{n} \leq N$. And from $L \geq \left\lceil \log_{\frac{b}{a}} \frac{N}{a} \right\rceil + 1$, we get

$$c^L > N, \tag{12}$$

where the inequality strictly holds.

Subsequently, $c^L > \frac{m}{n}$, namely $a^L \cdot m < b^L \cdot n$ naturally holds, and then (11) strictly holds as well. \square

Observe that the construction of the SNV, and sequentially we have the information of all $n_k, \bar{\mu}_k$, and $D_{k,r,s}$, which have

the corresponding definitions as described in section II. Here, we simply list them out again:

$$\begin{aligned} \vec{\mu}_k &= (\mu_{1,k}, \mu_{2,k}), \\ D_{k,r,s} &= \sum_{t=1}^{n_k} \frac{(i_{t,k} - \mu_{1,k})^r \cdot (j_{t,k} - \mu_{2,k})^s}{(n_k)^{r+s} \cdot N^{r+s-1}}, \end{aligned} \quad (13)$$

where $\mu_{1,k} = \frac{\sum_{t=1}^{n_k} i_{t,k}}{n_k}$, $\mu_{2,k} = \frac{\sum_{t=1}^{n_k} j_{t,k}}{n_k}$, r is any non-negative integer, $s \in \{1, 2, \dots, n_k\}$, and $k \in K$.

Lemma 4: Given the information of all n_k , μ_k , if we define a new higher moment as:

$$\tilde{D}_{k,r,s} = \sum_{t=1}^{n_k} \frac{(i_{t,k})^r \cdot (j_{t,k})^s}{(n_k)^{r+s} \cdot N^{r+s-1}}, \quad (14)$$

then $\{D_{k,r,s}\}$ and $\{\tilde{D}_{k,r,s}\}$ can be linearly represented by each other, where r , s , and k are defined as in (4).

Proof: On the one hand, if we know the information of n_k , μ_k and $D_{k,r,s}$, then we want to use this information to obtain all $\tilde{D}_{k,r,s}$. For any r , s , according to the definition of $\tilde{D}_{k,r,s}$, we have

$$\begin{aligned} &(n_k)^{r+s} \cdot N^{r+s-1} \cdot \tilde{D}_{k,r,s} \\ &= \sum_{t=1}^{n_k} (i_{t,k})^r \cdot (j_{t,k})^s \\ &= \sum_{t=1}^{n_k} (i_{t,k} - \mu_{1,k} + \mu_{1,k})^r \cdot (j_{t,k} - \mu_{2,k} + \mu_{2,k})^s \\ &= \sum_{t=1}^{n_k} \left[\sum_{l=0}^r C(r, l) \cdot (i_{t,k} - \mu_{1,k})^l \cdot (\mu_{1,k})^{r-l} \right] \\ &\quad \cdot \left[\sum_{v=0}^s C(s, v) \cdot (j_{t,k} - \mu_{2,k})^v \cdot (\mu_{2,k})^{s-v} \right] \\ &= \sum_{t=1}^{n_k} \sum_{l=0}^r \sum_{v=0}^s C(r, l) C(s, v) \cdot (\mu_{1,k})^{r-l} (\mu_{2,k})^{s-v} \\ &\quad \cdot (i_{t,k} - \mu_{1,k})^l (j_{t,k} - \mu_{2,k})^v \\ &= \sum_{l=0}^r \sum_{v=0}^s C(r, l) C(s, v) (\mu_{1,k})^{r-l} (\mu_{2,k})^{s-v} \\ &\quad \cdot \left[\sum_{t=1}^{n_k} (i_{t,k} - \mu_{1,k})^l (j_{t,k} - \mu_{2,k})^v \right] \\ &= \sum_{l=0}^r \sum_{v=0}^s C(r, l) C(s, v) (\mu_{1,k})^{r-l} (\mu_{2,k})^{s-v} D_{k,l,v}. \end{aligned} \quad (15)$$

Then we have

$$\tilde{D}_{k,r,s} = \sum_{l=0}^r \sum_{v=0}^s \frac{C(r, l) C(s, v) (\mu_{1,k})^{r-l} (\mu_{2,k})^{s-v}}{(n_k)^{r+s} \cdot N^{r+s-1}} D_{k,l,v}. \quad (16)$$

Namely, the $\tilde{D}_{k,r,s}$ can be linearly represented by the combination of $D_{k,l,v}$, where $l = 0, 1, 2, \dots, r$ and $v = 0, 1, 2, \dots, s$.

On the other hand, if we know the information of n_k , $\vec{\mu}_k$ and $\tilde{D}_{k,r,s}$, following the similar procedure, we can get

$$D_{k,r,s} = \sum_{l=0}^r \sum_{v=0}^s \frac{C(r, l) C(s, v) (-\mu_{1,k})^{r-l} (-\mu_{2,k})^{s-v}}{(n_k)^{r+s} \cdot N^{r+s-1}} \tilde{D}_{k,l,v}. \quad (17)$$

Therefore, we finish the proof and get the conclusion. \square

IV. THE APPLICATION OF THE SNV METHOD IN FACE RECOGNITION

The proposed SNV method is used for face recognition and tested on three datasets (ORL dataset, ‘total_73_95faces’ dataset, and ‘total_151_96faces’ dataset). For the three datasets, specifically, the ORL dataset is used to evaluate the performance of SNV under conditions where the poses are varied. The ‘total_73_95faces’ dataset is employed to test the performance of the proposed method under conditions where there is a variation in lighting conditions. The ‘total_151_96faces’ dataset with complex background of face image is utilized to examine the performance when both the facial expression and background of face image are varied.

The final two datasets are taken from the Computer Vision Science Research Projects designed and maintained by Dr. Libor Spacek, which can be downloaded on the link <https://cswww.essex.ac.uk/mv/allfaces>. These datasets have different numbers of individuals, together with 20 distinct face images per person, showing very minor changes in head turn, tilt, slant, and head position in the image. Hairstyle variation was minimal as the images were taken in a single session. All images were in color, meaning three intensity values per pixel. We converted all color face images to grayscale data as the first step, which specifically says that we convert the RGB color image to the grayscale intensity image by eliminating the hue and saturation information while retaining the luminance.

It is necessary to emphasize that our proposed method is entirely based on the information of pixels of the face image, so we construct a comparison with the conventional PCA algorithm, 2DPCA method and 2D-EPCA approach for the face recognition. And further there exists a natural algorithm based on all pixels of an image, which we define as ‘‘Full-Pixel’’ algorithm. For detailed description of the ‘‘Full-Pixel’’ algorithm, if there are two image A and B both with $m \times n$, we define the Euclidean distance between A and B as:

$$\text{dist}(A, B) = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (A_{i,j} - B_{i,j})^2}. \quad (18)$$

The ‘‘Full-Pixel’’ algorithm takes advantage of the defined distance to convey the similarity between each pair images.

We tested our SNV method against classical PCA, 2DPCA, 2D-EPCA, and ‘‘Full-Pixel’’ algorithm by using only moments up to degree two. That is, the truncated SNV we used is consisted of counts, locations, and three degree-two moments for each $k \in K$. We found that a reduction in dimension of the data from $N = mn$ (i.e., from

10, 000 or more) to $6 \cdot 256 = 1536$ gave fairly accurate facial recognition, on account of the (6) and (7). Specifically, for each face image, we use the SNV listed in (19). As for PCA, 2DPCA, and 2D-EPCA methods, the maximum reduced dimensionality equals to the number of non-zero eigenvalues. When algorithms are applied across these datasets, we select different number of images of each individual for training and the remaining images of each individual for testing.

$$\begin{aligned}
 &< n_0, n_1, n_2, \dots, n_{255}, \\
 &\mu_0^1, \mu_1^1, \mu_{A_2}^1, \dots, \mu_{255}^1, \\
 &\mu_0^2, \mu_1^2, \mu_{A_2}^2, \dots, \mu_{255}^2, \\
 &D_{0,2,0}, D_{1,2,0}, D_{2,2,0}, \dots, D_{255,2,0}, \\
 &D_{0,0,2}, D_{1,0,2}, D_{2,0,2}, \dots, D_{255,0,2}, \\
 &D_{0,1,1}, D_{1,1,1}, D_{2,1,1}, \dots, D_{255,1,1} >. \quad (19)
 \end{aligned}$$

As for face classification or face recognition between a given face image named I_{given} and a query face image I_{query} , we need to detail concrete steps of the SNV method. First of all, we would like to compute their corresponding truncated SNVs to degree two high moments which is shown like (19), and note them as u_1 and u_2 correspondingly, naturally we know that $u_1, u_2 \in \mathbb{R}^{1536}$. Here, the distance between two arbitrary feature SNV $u_1 = [u_1^1, u_1^2, \dots, u_1^{1536}]$ and $u_2 = [u_2^1, u_2^2, \dots, u_2^{1536}]$, is defined by

$$\begin{aligned}
 d(u_1, u_2) &= \sqrt{(u_1^1 - u_2^1)^2 + \dots + (u_1^{1536} - u_2^{1536})^2} \\
 &= \sqrt{\sum_{i=1}^{1536} (u_i^1 - u_i^2)^2}. \quad (20)
 \end{aligned}$$

Then, we adopt minimum distance (MD) classifier to calculate the recognition rate and record the time consuming of the whole process. Note that the common Euclidean distance measure is adopted in all methods. Suppose that the training samples are u_1, u_2, \dots, u_M (where M is the total number of training samples), and that each of these samples is assigned a given identity (class) ω_k . Given a test sample u , if $d(u, u_i) = \min_j \{d(u, u_j)\}$ and $u_i \in \omega_k$, then the resulting decision is $u \in \omega_k$. In the end, the total times of true decision are recorded and denominated by the number of testing samples which is the recognition rate we defined.

A. EXPERIMENTS ON THE ORL DATASET

The ORL dataset which contains images from 40 individuals, each containing 10 different poses. For some subjects, the images were taken at different times. The faces were photographed at intervals, with varying lighting, facial expression (eyes closed/opened, smiling/not smiling), facial poses and facial details (with/without glasses, with/without beard), among other type of variations. This dataset contains both males and females. All images are in grayscale, with a resolution of 92×112 pixel. Six sample images of one person from the ORL dataset are shown in Fig.1.

First, an experiment is performed by using five image samples per class for training and the remaining images



FIGURE 1. Example images of ORL dataset (only 6 pose out of 10).

for test. Thus the total number of training samples and testing samples are both 200. The SNV method is first used as feature extraction which is defined as (19) a vector in \mathbb{R}^{1536} . As for PCA algorithm, even 2DPCA and 2D-EPCA methods, the eigenvectors are the maximum reduced dimensionality which equals to the number of non-zero eigenvalues of the training samples' covariance matrix. We also employ another kind of classifier named as Support Vector Machine (SVM) [14] for comparison in PCA algorithm. The experimental result is shown on Table 2.

TABLE 2. The performance comparison of different methods on the ORL dataset.

Methods	Recognition Rate(100%)	Time Consuming(s)
Full-Pixel + MD	0.7125	102.36
PCA + MD	0.885	1.9547
PCA + SVM	0.84	7.373
2DPCA + MD	0.915	4.77
2D-EPCA + MD	0.975	4.59
SNV + MD	0.98	2.799

Another experiment is done based on time consuming of different face image number for training from 5 to 9. The result is shown on Table 3.

TABLE 3. The time consuming of different face image number for training on the ORL dataset.

Training Number	5	6	7	8	9
Time Consuming(s) for PCA	1.9547	1.8523	1.8922	2.2966	2.7318
Time Consuming(s) for SNV	2.799	2.7410	2.8998	2.5419	2.5383

B. EXPERIMENTS ON THE 'total_73_95faces' DATASET

The second is the 'total_73_95faces' dataset with 73 individuals, each with 20 face images. The background of each individual's images, which have the image resolution of 180×200 pixels as the second dataset, consist of a red curtain. As subject moves forward, background variation is caused by shadows and significant lighting arrangement. Some of grayscale face images of one person is shown on Fig.2. The first experiment on this dataset is that we select 18 face images of each individual as training samples and others are used for test, by using different methods for comparison. The experimental result is shown on Table 4.

Another experiment is done based on time consuming of different face image number for training from 10 to 19, the result is shown on Table 5.

C. EXPERIMENTS ON THE 'total_151_96faces' DATASET

The third one is the 'total_151_96faces' dataset, where there have 151 individuals, with each image resolution



FIGURE 2. Example images of ‘total_73_95faces’ dataset (only 12 pose out of 20).

TABLE 4. The performance comparison of different methods on the ‘total_73_95faces’ dataset.

Methods	Recognition Rate(100%)	Time Consuming(s)
Full-Pixel + MD	0.8685	2184.803
PCA + MD	0.9269	86.3102
PCA + SVM	0.7684	38.7054
2DPCA + MD	0.9589	28.22
2D-EPCA + MD	0.9863	21.42
SNV + MD	1.00	25.2350

TABLE 5. The time consuming of different face image number for training on the ‘total_73_95faces’ dataset.

Training Number	Time Consuming(s) for PCA	Time Consuming(s) for SNV
11	39.978	24.965
12	45.6402	25.4589
13	52.4397	25.2920
14	60.2334	25.4579
15	68.5232	25.1980
16	75.5097	25.1597
17	86.3102	25.2350
18	95.6109	24.8165
19	106.1058	25.3827



FIGURE 3. Example images of ‘total_151_96faces’ dataset (only 12 pose out of 20).

of 196×196 pixels. The images of each individual have expression variation and large head scale variation with a complex background showing glossy posters. Significant lighting changes occur due to the artificial lighting arrangement as the subject moves forward. Some of grayscale face images of one person is shown on Fig.3. We select 18 face images of each individual as training samples and others are used for test, which is the first experiment for comparison of different methods. The result is shown on Table 6.

Another experiment is done based on time consuming of different face image number for training. The result is shown on Table 7.

TABLE 6. The performance comparison of different methods on the ‘total_151_96faces’ dataset.

Methods	Recognition Rate(100%)	Time Consuming(s)
Full-Pixel + MD	0.8756	5149.03
PCA + MD	0.9036	414.87
PCA + SVM	0.7377	156.15
2DPCA + MD	0.9238	84.66
2D-EPCA + MD	0.9370	69.92
SNV + MD	0.9808	80.52

TABLE 7. The time consuming of different face image number for training on the ‘total_151_96faces’ dataset.

Training Number	Time Consuming(s) for PCA	Time Consuming(s) for SNV
11	166.75	82.54
12	196.25	81.73
13	226.17	79.50
14	262.51	80.15
15	299.81	81.765
16	331.88	79.96
17	378.21	82.10
18	414.87	80.27
19	468.28	79.83

D. EVALUATION OF THE EXPERIMENTAL RESULTS

This section evaluates the SNV method by comparing it with “Full-Pixel” algorithm, conventional PCA algorithm, and its variants – 2DPCA and 2D-EPCA algorithms on the three face images datasets which have been recommended in details above. All face images in the same dataset have been aligned according to the same pixel. To eliminate the influence of randomness, we repeat each trials twenty times and then report the final results with their average recognition rate and time consuming, and all of these experiments have been implemented using MATLAB of version R2015b. We compare our proposed method with PCA, 2DPCA, 2D-EPCA, and “Full-Pixel” algorithms in terms of their average recognition rate and time consuming. Furthermore, we adopt MD classifier to calculate the recognition rate. Especially, we also use the Support Vector Machine (SVM) [14] classifier in PCA method to get corresponding results in all three datasets.

The first kind of comparison is constructed among different methods on three datasets, and from the experimental results showing on Table 2, Table 4, and Table 6, we know that even though the “Full-Pixel” algorithm has a mildly high recognition rate, on account of using the all pixels information, it is poorly deficient in the good performance on time consuming. Therefore it is necessary for researchers to reduction dimension of data, among which our proposed method, the PCA, 2DPCA and 2D-EPCA algorithms have good performance.

From the experimental results shown on Table 2, Table 4 and Table 6, firstly, we know that our proposed method has a higher recognition accuracy than the PCA, 2DPCA, 2D-EPCA, and “Full-Pixel” algorithms across all datasets above. Secondly, the SNV method is more computationally efficient than PCA algorithm shown in Table 4, and Table 6

on time consumings, and much more efficient than “Full-Pixel” algorithm illustrated by Table 2, Table 4, and Table 6, where there is a hidden property for the SNV method that it can be on parallel computation in two aspects. The first aspect is that when we compute the SNV of one face image, we can compute each intensity value $\{0, 1, 2, \dots, 255\}$ in parallel, because in the truncated SNV, the intensity values are in equivalence place. When we compute the distance between two face images, we convert it to compute the distance between the two corresponding truncated SNVs. Therefore, we need to compute the distances of all the truncated SNVs instead of all face images. The computation of all distances of the truncated SNVs is also in parallel which is the second aspect. It is known that parallel computation is a good technique for us to reduce the time consuming, however it is difficult to achieve in a lot of real applications. Nevertheless, our proposed SNV method in the application of face recognition can be achieved in parallel.

While the resolution of the face images in the ORL dataset is lower than the other two datasets, the performance of our method in time consuming is a little worse than PCA, which illustrates that our proposed method has not very prominent merit of reduced time consuming in the not very high dimensions data. And further, our proposed method achieves a good performance in time consuming almost similar with 2DPCA algorithm and 2D-EPCA technique among all datasets.

In addition, the face image background is obviously different and complex on the last two datasets. Specifically the face images of the second dataset are in a red curtain background, and the third with complex background of glossy posters. As the experimental results shown about these final two datasets, we can know that our method performs a steadily high property in not only the recognition accuracy, but also the computational load.

The second kind of comparison is constructed to study the performance of our proposed method and PCA algorithm when randomly selecting different number of face image for training. From the experimental results showing on Table 5 and Table 7, we know that our proposed method was more efficient than PCA method. Furthermore, when the training number is larger, the time consuming of PCA method is larger than SNV method, whereas our proposed method’s time consuming is almost stable no matter how many face images of each individual for training are, as depicted in both two tables above. The good property of our proposed method in the stable time consuming is not obtained on ORL dataset shown on Table 3, which again illustrates that our proposed method has no much superiority of time consuming over PCA algorithm in the not very high dimensional data.

V. CONCLUSIONS

A new technique for image feature extraction and representation, which is called Stretched Natural Vector (SNV) method, was proposed and developed. While the proposed SNV is unique to one face image, the same individual’s different images have similar extended natural vectors which

also represents some biological character of a person. We use this property to propose a new facial recognition algorithm, i.e., the SNV method, in this paper.

What the most significant contribution of this paper is that we have successfully proved that the correspondence between a two-dimensional grayscale image matrix and its SNV is in strict one-to-one fashion. Besides, the performance of proposed SNV method is better than the “Full-Pixel” algorithm, the conventional PCA method, and its variant – 2DPCA and 2D-EPCA methods in the recognition accuracy and computational load. In addition, along with the training sample numbers increasing, the SNV method is more computationally efficient than PCA algorithm, and owns an acceptable computational time cost.

Furthermore, our proposed method is not like other existing algorithms that when the number of training samples is larger, the time consuming is higher, which is not a good performance for face recognition. While the SNV method maintain a mildly stable time consuming on the dataset no matter what the number of training sample is.

APPENDIXES

Now, we begin to prove the main theorem.

Proof of Theorem 1:

Step 1: Here we show how the SNV determines the set of row locations for a given k .

Let $\delta_{k,j,0} = \tilde{D}_{k,j,0} \cdot n_k^j \cdot N^{j-1}$, where $j \in \{1, 2, \dots, n_k\}$. Then the set of $\delta_{k,j,0}$ ’s can be obtained by the set of $\tilde{D}_{k,j,0}$ ’s and vice versa. For a fixed k , the δ ’s are multiples of the \tilde{D} ’s by a fixed factor that doesn’t depend on j and which is known from one of the SNV components. The δ ’s are raw power sums whereas the \tilde{D} ’s are normalized by their denominator factors and tend to 0 as $j \rightarrow \infty$.

$$\begin{aligned} \delta_{k,1,0} &= i_{1,k} + i_{2,k} + \dots + i_{n_k,k}, \\ \delta_{k,2,0} &= (i_{1,k})^2 + (i_{2,k})^2 + \dots + (i_{n_k,k})^2, \\ &\dots\dots \\ \delta_{k,n_k,0} &= (i_{1,k})^{n_k} + (i_{2,k})^{n_k} + \dots + (i_{n_k,k})^{n_k}, \end{aligned} \quad (21)$$

where $k = 0, 1, 2, \dots, 255$.

To illustrate this step in the theorem clearly, we focus on a single the grayscale k as an illustration. The row values corresponding to k with multiplicities, $i_{1,k}, i_{2,k}, \dots, i_{n_k,k}$, are roots of a symmetric polynomial

$$\begin{aligned} a_0 + a_1x + a_2x^2 + \dots + a_{n_k}x^{n_k} \\ = (x - i_{1,k})(x - i_{2,k}) \dots (x - i_{n_k,k}). \end{aligned} \quad (22)$$

Let p_d ($d = 1, 2, \dots, n$) be the elementary symmetric polynomials in $i_{1,k}, i_{2,k}, \dots, i_{n_k,k}$, i.e., $p_1 = -a_{n_k-1}, p_2 = a_{n_k-2}, \dots, p_{n_k} = (-1)^{n_k} a_0$.

Then $p_1 = \sum_{s=1}^{n_k} i_{s,k}, p_2 = \sum_{s<t} i_{s,k}i_{t,k}, \dots, p_{n_k} = i_{1,k}i_{2,k} \dots i_{n_k,k}$.

By Lemma 2, we have

$$\delta_{k,d,0} - p_1\delta_{k,d-1,0} + \dots + (-1)^{d-1}p_{d-1}\delta_{k,1,0} + (-1)^d p_d = 0, \quad (23)$$

where $d = 1, 2, \dots, n_k$, and p_d is the elementary symmetric polynomials in $i_{1,k}, i_{2,k}, \dots, i_{n_k,k}$.

And a_i can be obtained by $\delta_{k,j,0}$ as shown below:

$$\begin{cases} a_{n_k} = 1 \\ a_{n_k-1} = (-1)\delta_{k,1,0} \\ a_{n_k-2} = \frac{1}{2}((\delta_{k,1,0})^2 - \delta_{k,2,0}) \\ a_{n_k-3} = (-1)^3 \frac{1}{6}((\delta_{k,1,0})^3 - 3\delta_{k,1,0}\delta_{k,2,0} + 2\delta_{k,3,0}) \\ a_{n_k-4} = \frac{1}{24}((\delta_{k,1,0})^4 - 6(\delta_{k,1,0})^2\delta_{k,2,0} + 3(\delta_{k,2,0})^2 \\ + 8\delta_{k,1,0}\delta_{k,3,0} - 6\delta_{k,4,0}) \\ \dots \dots \end{cases} \quad (24)$$

As a result, the coefficients of the symmetric polynomial

$$a_0 + a_1x + a_2x^2 + \dots + a_{n_k}x^{n_k} = (x - i_{1,k})(x - i_{2,k}) \dots (x - i_{n_k,k}^k) \quad (25)$$

can be conformed, and the set of all roots can be obtained.

Let r_k be the number of distinct roots, which we expect to occur with multiplicities. We would like to identify the roots $i_{1,k}, i_{2,k}, \dots, i_{n_k,k}$ in a non-decreasing order then label them as shown below.

$$\begin{aligned} i_{1,k} &= i_{2,k} = \dots = i_{l_1,k} < \\ i_{l_1+1,k} &= i_{l_1+2,k} = \dots = i_{l_2,k} < \\ i_{l_2+1,k} &= i_{l_2+2,k} = \dots = i_{l_3,k} < \\ &\dots \dots \\ i_{l_{r_k-1}+1,k} &= i_{l_{r_k-1}+2,k} = \dots = i_{l_{r_k},k}, \end{aligned} \quad (26)$$

where $l_t - l_{t-1}$ is the multiplicity of the t^{th} root $l_{r_k} = n_k$. We put them all in a set denoting by S_k^0 . Namely,

$$S_k^0 := \{i_{1,k}, i_{2,k}, \dots, i_{l_1,k}, i_{l_1+1,k}, i_{l_1+2,k}, \dots, i_{l_2,k}, i_{l_2+1,k}, i_{l_2+2,k}, \dots, i_{l_3,k}, \dots, i_{l_{r_k-1}+1,k}, i_{l_{r_k-1}+2,k}, \dots, i_{l_{r_k},k}\}. \quad (27)$$

For the step 2, we will do some preparations and give some notations. We put the equal elements on one set respectively, and get the following sets:

$$\begin{aligned} G_k^{1,0} &:= \{i_{1,k}, i_{2,k}, \dots, i_{l_1,k}\}, \\ G_k^{2,0} &:= \{i_{l_1+1,k}, i_{l_1+2,k}, \dots, i_{l_2,k}\}, \\ G_k^{3,0} &:= \{i_{l_2+1,k}, i_{l_2+2,k}, \dots, i_{l_3,k}\}, \\ &\dots \dots, \\ G_k^{r_k,0} &:= \{i_{l_{r_k-1}+1,k}, i_{l_{r_k-1}+2,k}, \dots, i_{l_{r_k},k}\}, \end{aligned} \quad (28)$$

where for $\forall x_1, x_2 \in G_k^{t,0}, t = 1, 2, \dots, r_k$, we have $x_1 = x_2$. We now have $x < y$, if $\forall x \in G_k^{s,0}, \forall y \in G_k^{t,0}, 1 \leq s < t \leq r_k$. Similarly, we can find all other $i_{1,k}, i_{2,k}, \dots, i_{n_k,k}, k = 0, 1, 2, \dots, 255$, respectively, and get the similarly corresponding sets as shown below.

$$\begin{aligned} G_0^{1,0}, G_0^{2,0}, \dots, G_0^{r_0,0}, G_1^{1,0}, G_1^{2,0}, \dots, G_1^{r_1,0}, \\ G_2^{1,0}, G_2^{2,0}, \dots, G_2^{r_2,0}, \dots, G_{255}^{1,0}, G_{255}^{2,0}, \dots, G_{255}^{r_{255},0}. \end{aligned} \quad (29)$$

Note that if we say, given the finite integer sets A and $B, A < B$, it means that $x < y$, where $\forall x \in A$ and $\forall y \in B$. At the similar notation expressing, given the finite integer sets C and $D, C \ll D$, it means that $u \ll v$, for $\forall u \in C, \forall v \in D$.

Therefore, once we find all $i_{1,k}, i_{2,k}, \dots, i_{n_k,k}, k = 0, 1, 2, \dots, 255$, we can say that we have recovered the two-dimensional distribution matrix on one direction. It is not sufficient, however, to determine all of the exact positions of all elements in index set K . Henceforth, what is mostly significant is that when we have determined the positions of all levels grayscales on one direction, how we can uniquely get the other direction. We will give the specific answers as step 2 shown below.

Step 2: Using the information captured in the SNV components $D_{k,L,1}, D_{k,L,2}, \dots, D_{k,L,n_k}$, we consider the values of $(i_{s,k})^L j_{s,k}, s = 1, 2, \dots, n_k$ for an arbitrary positive integer L , in a slightly modified mimicry of step 1. Put the those integers in a set of H_k^L . Then we have

$$H_k^L := \{(i_{1,k})^L j_{1,k}, (i_{2,k})^L j_{2,k}, \dots, (i_{n_k,k})^L j_{n_k,k}\}. \quad (30)$$

By using Lemma 3, if $L \gg 1$, then for any $r < s$, we can obtain that (using t as the indexing variable in both sets)

$$\{(i_{t,k})^L j_{t,k}\}_{l_{r-1} < t \leq l_r} \ll \{(i_{t,k})^L j_{t,k}\}_{l_{s-1} < t \leq l_s}. \quad (31)$$

Use the notation

$$\begin{aligned} R_{k,s}^L &:= \{(i_{t,k})^L j_{t,k} \mid l_{s-1} < t \leq l_s\} \\ &= \{(i_{l_{t-1}+1,k})^L j_{l_{t-1}+1,k}, \dots, (i_{l_s,k})^L j_{l_s,k}\}, \end{aligned} \quad (32)$$

where $t = 1, 2, \dots, r_k$. We get the following sets in a strictly increasing order,

$$\begin{aligned} R_{k,1}^L &:= (i_{1,k})^L j_{1,k}, (i_{2,k})^L j_{2,k}, \dots, (i_{l_1,k})^L j_{l_1,k} \\ R_{k,2}^L &:= \{(i_{l_1+1,k})^L j_{l_1+1,k}, (i_{l_1+2,k})^L j_{l_1+2,k}, \dots, (i_{l_2,k})^L j_{l_2,k}\} \\ R_{k,3}^L &:= \{(i_{l_2+1,k})^L j_{l_2+1,k}, (i_{l_2+2,k})^L j_{l_2+2,k}, \dots, (i_{l_3,k})^L j_{l_3,k}\} \\ &\dots \\ R_{k,r_k}^L &:= \{(i_{l_{r_k-1}+1,k})^L j_{l_{r_k-1}+1,k}, (i_{l_{r_k-1}+2,k})^L j_{l_{r_k-1}+2,k}, \\ &\dots, (i_{l_{r_k},k})^L j_{l_{r_k},k}\} \end{aligned} \quad (33)$$

i.e., $R_{k,1}^L < R_{k,2}^L < R_{k,3}^L < \dots < R_{k,r_k}^L$. The strict ordering holds for all $k = 0, 1, 2, \dots, 255$.

Step 3: From step 1 and step 2, we can get the other direction's information just as shown below.

From (26) and (28), we take the operations shown as (34),

$$\begin{aligned} E_k^1 &:= \frac{R_{k,1}^L}{(i_{1,k})^L} = \{j_{1,k}, j_{2,k}, \dots, j_{l_1,k}\}, \\ E_k^2 &:= \frac{R_{k,2}^L}{(i_{l_2,k})^L} = \{j_{l_1+1,k}, j_{l_1+2,k}, \dots, j_{l_2,k}\}, \\ E_k^3 &:= \frac{R_{k,3}^L}{(i_{l_3,k})^L} = \{j_{l_2+1,k}, j_{l_2+2,k}, \dots, j_{l_3,k}\}, \\ &\dots \dots \\ E_k^{r_k} &:= \frac{R_{k,r_k}^L}{(i_{l_{r_k},k})^L} = \{j_{l_{r_k-1}+1,k}, j_{l_{r_k-1}+2,k}, \dots, j_{l_{r_k},k}\}. \end{aligned} \quad (34)$$

Actually, the other direction's information are determined by the sets $E_k^1, E_k^2, E_k^3, \dots, E_k^{r_k}$ computed by the (34). Then $S_k(i_t, k, j_t, k), t \in \{1, 2, \dots, n_k\}$ are obtained from the set of $G_k^{l,0}$ and E_k^l , where $l \in \{1, 2, \dots, r_{n_k}\}$. The concrete operation is shown below,

$$S_k(i_t, k, j_t, k) \in \{(x, y) | x \in G_k^{1,0}, y \in E_k^1\} \cup \{(x, y) | x \in G_k^{2,0}, y \in E_k^2\} \cup \dots \cup \{(x, y) | x \in G_k^{r_{n_k},0}, y \in E_k^{r_{n_k}}\}. \quad (35)$$

Similarly, we can find all $S_k(i_t, k, j_t, k)$, for $k = 0, 1, 2, \dots, 255$ and accordingly $t \in \{1, 2, \dots, n_k\}$. Therefore, the corresponding grayscale intensity distribution matrix of a given image can be recovered based on all $S_k(i_t, k, j_t, k), k = 0, 1, 2, \dots, 255, t \in \{1, 2, \dots, n_k\}$.

On the step 2 and 3, we know that we can recover an matrix's distribution from the SNV. There is another thing we need to explain for the uniqueness.

Step 4: Consider the set of $M_{m,n}(K)$ of $m \times n$ matrices with entries in the set $K = \{0, 1, 2, \dots, 255\}$. We will think of such a matrix as a function q from the set of entry locations $\{(i, j) | 1 \leq i \leq m; 1 \leq j \leq n\}$ to the set K of possible entry values. Given two distinct matrices A and B , there is an integer L for which their SNV's truncated at degree L are different. That implies the full SNV's are different.

The first components of an SNV are the cardinalities $n_k = |q^{-1}(k)|$ of the level sets. For A and B to have the same SNV means $n_k(A) = n_k(B)$ for each $k = 0, \dots, 255$.

Start with a single matrix A . For each $k \in K$, we define a function

$$f_k(x, y) = \prod_{q(i,j)=k} \left((x-i)^2 + (y-j)^2 \right) = \prod_{(i,j) \in q^{-1}(k)} \left((x-i)^2 + (y-j)^2 \right) = \prod_{(i,j) \in q^{-1}(k)} \left(x^2 - 2ix + i^2 + y^2 - 2jy + j^2 \right) = \prod_{(i,j) \in q^{-1}(k)} \left(x^2 + y^2 - 2ix - 2jy + (i^2 + j^2) \right). \quad (36)$$

For fixed $k \in K$, there are n_k matrix entries locations in $q^{-1}(k)$ and hence n_k factors in the product. Each function f_k is polynomial of combined degree $2n_k$ in x and y . We have $\forall (i, j) \in q^{-1}(k), f_k(x, y) = 0$. Furthermore, it is clear that for $f_k(x, y)$ to equal 0, one of the factors must equal 0 and (x, y) must be one of the locations in $q^{-1}(k)$. Thus $f_k(i, j) = 0 \iff q(i, j) = k$.

The (i, j) pairs in $q^{-1}(k)$ appear symmetrically in the expression for f_k . I.e. if the matrix entry locations in $q^{-1}(k)$ are ordered $(i_1, j_1), (i_2, j_2), \dots, (i_{n_k}, j_{n_k})$ and $f_k(x, y)$ is written as $((x - i_1)^2 + (y - j_1)^2) \cdot ((x - i_2)^2 + (y - j_2)^2) \cdot \dots \cdot ((x - i_{n_k})^2 + (y - j_{n_k})^2)$, the ordering doesn't matter.

Assume A has the same SNV as B . We aim to show A and B have the same functions f_k and hence the same level sets for each $k \in K$. We do this by showing the coefficient of the

monomial $x^a y^b$ in f_k can be extracted from components in the SNV.

Let us look again at our little 3×8 example (1). The function f_{34} is a polynomial of degree eight given by

$$f_{34}(x, y) = \left((x-1)^2 + (y-4)^2 \right) \cdot \left((x-1)^2 + (y-6)^2 \right) \cdot \left((x-2)^2 + (y-2)^2 \right) \cdot \left((x-3)^2 + (y-5)^2 \right) = \left(x^2 + y^2 - 2x - 8y + 17 \right) \cdot \left(x^2 + y^2 - 6x - 10y + 34 \right). \quad (37)$$

It is easy to see that the monomials of degree 8 are $x^8 + C(4, 1)x^6y^2 + C(4, 2)x^4y^4 + C(4, 3)x^2y^6 + y^8$ and the constant term is $(1^1 + 4^2)(1^2 + 6^2)(2^2 + 2^2)(3^2 + 5^2) = (17)(37)(8)(34)$. The degree 8 coefficients would be exactly the same if some other four matrix entry locations made up $q^{-1}(34)$ but the constant term is informative. Each contribution is a product of i^2 factors from some of the entry locations and j^2 factors from the remaining locations. Thus, an accurate descrip-

tion of the constant term is $\sum_{S \subset q^{-1}(k)} \left(\prod_{(i,j) \in S} i^2 \prod_{(i,j) \in q^{-1}(k) \setminus S} j^2 \right)$

which would be a tedious mess to write out in its entirety. A thorough description might list summands coming from first the empty subset, then the four singleton subsets, then the six 2-element subsets, the 3-element subsets, and finally the 4-element subset whose contribution is $(4^2)(6^2)(2^2)(5^2)$. More generally, for any intermediate monomial of the form $x^r y^s$ the coefficient will be even messier, consisting of sums of products where some of the factors are x^2 , some are $2ix$, some are $i^2 + j^2$, some are y^2 , etc. The proof doesn't depend on a detailed description of each coefficient. It only depends on symmetry with respect to the ordering of entry locations. So by using Lemma 2 again, we know that if the matrices A and B has the same SNV, namely each component is same, naturally the $D_{k,r,s}$ is same, then the coefficients of polynomial $f_k(x, y)$ of the matrices A and B are same as well. Hence, the uniqueness holds.

On the other hand, for a given grayscale intensity distribution matrix of the grayscale image, we can count the numbers n_k , compute μ_k , and get the normalized central moments $D_{k,r,s}$ from the procedure of constructing the SNV of the two-dimensional distribution in matrix.

In terms of the Lemma 4, naturally, we can get the information of $\tilde{D}_{k,r,s}$ as well.

Therefore, we have successfully proved that the correspondence between a two-dimensional distribution matrix and its SNV is one-to-one. \square

ACKNOWLEDGMENT

Stephen S.-T. Yau is grateful to National Center for Theoretical Sciences (NCTS) for providing excellent research environment while part of this research was done.

REFERENCES

- [1] T. Ahonen, M. Pietikäinen, A. Hadid, and T. Maenpää, "Face recognition based on the appearance of local regions," in *Proc. 17th Int. Conf. Patt. Recognit.*, Aug. 2004, pp. 153–156.
- [2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
- [3] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1450–1464, Nov. 2002.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
- [5] Y. Choi, S. Ozawa, and M. Lee, "Incremental two-dimensional kernel principal component analysis," *Neurocomputing*, vol. 134, pp. 280–288, Jun. 2014.
- [6] K. Cui, Q. Gao, H. Zhang, X. Gao, and D. Xie, "Merging model-based two-dimensional principal component analysis," *Neurocomputing*, vol. 168, pp. 1198–1206, Nov. 2015.
- [7] H. Du, Q. Hu, M. Jiang, and F. Zhang, "Two-dimensional principal component analysis based on Schatten p -norm for image feature extraction," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 55–62, Oct. 2015.
- [8] X. Feng, M. Pietikäinen, and A. Hadid, "Facial expression recognition with local binary patterns and linear programming," *Patt. Recogn. Image Analy.*, vol. 15, no. 2, pp. 546–548, 2005.
- [9] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.
- [10] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1087–1099, Jul. 2012.
- [11] S. Gong, M. Cristani, S. Yan, and C. Loy, *Person Re-Identification*. New York, NY, USA: Springer, 2014.
- [12] A. Hadid, M. Pietikäinen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2004, pp. 797–804.
- [13] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognit.*, vol. 29, no. 8, pp. 1233–1244, Aug. 1996.
- [14] K. Lee, Y. Chung, and H. Byun, "SVM based face verification with feature set of small size," *Electron. Lett.*, vol. 38, no. 15, pp. 787–789, 2002.
- [15] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using LDA-based algorithms," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 195–200, Jan. 2003.
- [16] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Euler principal component analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 498–518, 2013.
- [17] Q. Y. Li and H. Y. Quan, "The dimension reduction method of face feature parameters based on modular 2DPCA and PCA," *Appl. Mech. Mater.*, vols. 687–691, pp. 4037–4041, Nov. 2014.
- [18] D. G. Mead, "Newton's identities," *Amer. Math. Monthly*, vol. 99, no. 8, pp. 749–751, Oct. 1992.
- [19] T. Mandal, Q. M. Jonathan Wu, and Y. Yuan, "Curvelet based face recognition via dimension reduction," *Signal Process.*, vol. 89, no. 12, pp. 2345–2353, Dec. 2009.
- [20] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 84–91.
- [21] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.
- [22] Y. Rodriguez and S. Marcel, "Face authentication using adapted local binary pattern histogram," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 321–332.
- [23] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
- [24] C. Shan, S. Gong, and P. W. McOwan, "Robust facial expression recognition using local binary patterns," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, pp. 914–917.
- [25] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
- [26] C. Guzel Turhan and H. S. Bilge, "Class-wise two-dimensional PCA method for face recognition," *IET Comput. Vis.*, vol. 11, no. 4, pp. 286–300, Jun. 2017.
- [27] H. B. Tan, X. Zhang, N. Y. Guan, D. C. Tao, X. H. Huang, and Z. G. Luo, "Two-dimensional Euler PCA for face recognition," in *Proc. Int. Conf. Multimedia Modeling*, vol. 8936, 2015, pp. 548–559.
- [28] A. Vadivel, A. Majumdar, and S. Sural, "Performance comparison of distance metrics in content-based image retrieval applications," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2003, pp. 159–164.
- [29] Z. Wang, R. Hu, C. Liang, Y. Yu, J. Jiang, M. Ye, J. Chen, and Q. Leng, "Zero-shot person re-identification via cross-view consistency," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 260–272, Feb. 2016.
- [30] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.
- [31] P. C. Yuen and J. H. Lai, "Face representation using independent component analysis," *Pattern Recognit.*, vol. 35, no. 6, pp. 1247–1257, Jun. 2002.
- [32] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Advances in Biometric Person Authentication (Lecture Notes in Computer Science)*, vol. 3338, S. Z. Li, J. Lai, T. Tan, G. Feng, and Y. Wang, Eds. Berlin, Germany: Springer, 2004, pp. 179–186.
- [33] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 786–791.
- [34] D. Zhang, Z.-H. Zhou, and S. Chen, "Diagonal principal component analysis for face recognition," *Pattern Recognit.*, vol. 39, no. 1, pp. 140–142, Jan. 2006.



WENHUI DONG was born in Hubei, China, in 1993. She received the Ph.D. degree in mathematics from Tsinghua University, Beijing, China, in 2020.

She is currently a Postdoctoral Fellow with the School of Mathematics and Information Science, Shaanxi Normal University, Xi'an, China. Her research interests include nonlinear filtering, estimation algebras, and numerical analysis.



STEPHEN S.-T. YAU (Fellow, IEEE) received the Ph.D. degree in mathematics from the State University of New York at Stony Brook, Stony Brook, NY, USA, in 1976.

He was a member of the Institute of Advanced Study at Princeton, from 1976 to 1977 and from 1981 to 1982, and a Benjamin Pierce Assistant Professor with Harvard University, from 1977 to 1980. After that, he joined the Department of Mathematics, Statistics and Computer Science (MSCS), University of Illinois at Chicago (UIC), and has served for more than 30 years. In 2005, he was named UIC Distinguished Professor. From 2005 to 2011, he was a joint Professor with the Department of Electrical and Computer Engineering, MSCS, UIC. After his retirement in 2012, he joined Tsinghua University, Beijing, China, where he is currently a full-time Professor with the Department of Mathematical Sciences. His research interests include nonlinear filtering, bioinformatics, complex algebraic geometry, CR geometry, and singularities theory.

Dr. Yau was awarded the Sloan Fellowship, in 1980, the Guggenheim Fellowship, in 2000, and the AMS Fellow Award, in 2013. He was the General Chairman of the IEEE International Conference on Control and Information, which was held at the Chinese University of Hong Kong, in 1995. He has been the Founder and the Managing Editor of the *Journal of Algebraic Geometry*, since 1991. He has also been the Founder and the Editor-in-Chief of *Communications in Information and Systems*, since 2000.

• • •