

# ITERATIVE BLOCK TENSOR SINGULAR VALUE THRESHOLDING FOR EXTRACTION OF LOW RANK COMPONENT OF IMAGE DATA

Longxi Chen Yipeng Liu Ce Zhu

School of Electronic Engineering / Center for Robotics / Center for Information in Medicine  
University of Electronic Science and Technology of China (UESTC), Chengdu, 611731, China  
emails: chenlongxi@std.uestc.edu.cn, {yipengliu, eczhu}@uestc.edu.cn

## ABSTRACT

Tensor principal component analysis (TPCA) is a multi-linear extension of principal component analysis which converts a set of correlated measurements into several principal components. In this paper, we propose a new robust TPCA method to extract the principal components of the multi-way data based on tensor singular value decomposition. The tensor is split into a number of blocks of the same size. The low rank component of each block tensor is extracted using iterative tensor singular value thresholding method. The principal components of the multi-way data are the concatenation of all the low rank components of all the block tensors. We give the block tensor incoherence conditions to guarantee the successful decomposition. This factorization has similar optimality properties to that of low rank matrix derived from singular value decomposition. Experimentally, we demonstrate its effectiveness in two applications, including motion separation for surveillance videos and illumination normalization for face images.

**Index Terms**— tensor principal component analysis, tensor singular value decomposition, low rank tensor approximation, block tensor

## 1. INTRODUCTION

The high-dimensional data, also referred to as tensors, arise naturally in a number of scenarios, including image and video processing, and data mining [1]. However, most of the current processing techniques are developed for two-dimensional data [2]. The principal component analysis (PCA) is one of the most widely used one in two-dimensional data analysis [3].

The robust PCA (RPCA), as an extension of PCA, is an effective method in matrix decomposition problems [4]. Suppose we have a matrix  $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ , which can be decomposed as  $\mathbf{X} = \mathbf{L}_0 + \mathbf{S}_0$ , where  $\mathbf{L}_0$  is the low rank component of the matrix and  $\mathbf{S}_0$  is the sparse component. The RPCA method has been applied to image alignment [5], surveillance video processing [6], illumination normalization for face images [7]. In most applications, the RPCA method should flatten or vectorize the tensor data so as to solve the problem in the matrix. It doesn't use the structural feature of the data effectively since the information loss involves in the operation of matricization.

Tensor robust principal component analysis (TRPCA) has been studied in [8, 9] based on the tensor singular value decomposition (t-SVD). The advantage of t-SVD over the existing methods such as canonical polyadic decomposition (CPD) [10] and Tucker decomposition [11] is that the resulting analysis is very close to that

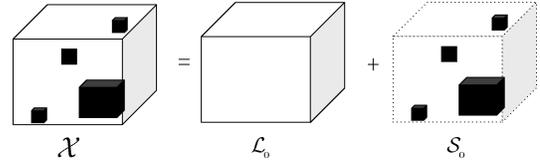


Fig. 1: Illustration of TRPCA.

of matrix analysis [12]. Similarly, suppose we are given a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and it can be decomposed into low rank component and sparse component. We can write it as

$$\mathcal{X} = \mathcal{L}_0 + \mathcal{S}_0, \quad (1)$$

where  $\mathcal{L}_0$  denotes the low rank component, and  $\mathcal{S}_0$  is the sparse component of the tensor. Fig. 1 is the illustration for TRPCA. In [8] the problem (1) is transformed to the convex optimization model:

$$\min_{\mathcal{L}_0, \mathcal{S}_0} \|\mathcal{L}_0\|_* + \lambda \|\mathcal{S}_0\|_1, \quad \text{s. t. } \mathcal{X} = \mathcal{L}_0 + \mathcal{S}_0, \quad (2)$$

where  $\|\mathcal{L}_0\|_*$  is the tensor nuclear norm (see section 2 for the definition),  $\|\mathcal{S}_0\|_1$  denotes the  $\ell_1$ -norm. In the paper [9] the problem (1) is transformed to another convex optimization model as:

$$\min_{\mathcal{L}_0, \mathcal{S}_0} \|\mathcal{L}_0\|_* + \lambda \|\mathcal{S}_0\|_{1,1,2}, \quad \text{s. t. } \mathcal{X} = \mathcal{L}_0 + \mathcal{S}_0, \quad (3)$$

where  $\|\mathcal{S}_0\|_{1,1,2}$  is defined as  $\sum_{i,j} \|\mathcal{S}_0(i, j, :)\|_F$ . The two methods solve the tensor decomposition problem based on the t-SVD.

The low rank and sparse matrix decomposition has been improved by the [13]. The main idea is incorporating multi-scale structures with low rank methods. The additional multi-scale structures can obtain a more accurate representation than conventional low rank methods. Inspired by this work, we notice that the sparse component in matrix is block-distributed in some applications, e.g. shadow and motion in videos. For these images we find it is more effective to extract the low rank components in a another smaller scale of image data. Here we try to extract low rank components in block tensor data that is stacked by small scale of image data. And when we decompose the tensor data into many small blocks, it is easy to extract the principal component in some blocks that have few sparse components. We model our tensor data as the concatenation of block tensors instead of solving the RPCA problem as a whole big tensor. Fig. 2 is the illustration of concatenation of block tensors.

Based on the above motivation, we decompose the whole tensor into concatenation of blocks of the same size, then we extract low rank component of each block by minimizing the tubal rank of

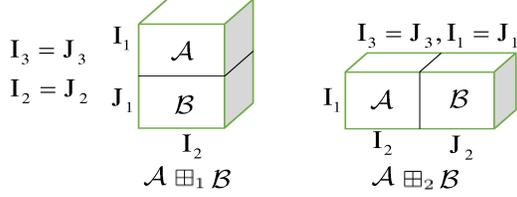


Fig. 2: Illustration of the concatenation of block tensors.

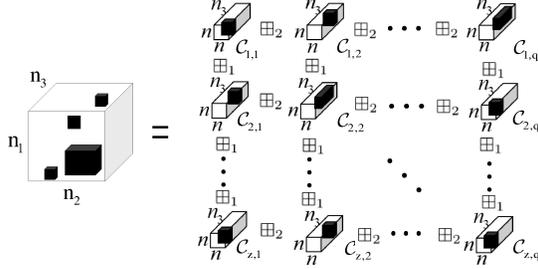


Fig. 3: Illustration of the block tensor decomposition model.

each block tensor. Fig. 3 is the illustration of our method. And we get low rank component of the whole tensor by concatenating all the low rank components of the block tensors. The proposed method can be used to some conventional image processing problems, including motion separation for surveillance videos (Section 4.1) and illumination normalization for face images (Section 4.2). The results of numerical experiments demonstrate that our method outperforms the existing methods in term of accuracy.

## 2. NOTATIONS AND PRELIMINARIES

In this section, we describe the notations and definitions used throughout the paper briefly [14, 15, 16, 17].

A third-order tensor is represented as  $\mathcal{A}$ , and its  $(i, j, k)$ -th entry is represented as  $\mathcal{A}_{i,j,k}$ .  $\mathcal{A}(i, j, :)$  denotes the  $(i, j)$ -th tubal scalar.  $\mathcal{A}(i, :, :)$ ,  $\mathcal{A}(:, j, :)$  and  $\mathcal{A}(:, :, k)$  are the  $i$ -th horizontal,  $j$ -th lateral and  $k$ -th frontal slices, respectively.  $\|\mathcal{A}\|_F = \sqrt{\sum_{i,j,k} |a_{i,j,k}|^2}$  and  $\|\mathcal{A}\|_\infty = \max_{i,j,k} |a_{i,j,k}|$  tensor kinds of tensor norms.

We can view a three-dimensional tensor of size  $n_1 \times n_2 \times n_3$  as an  $n_1 \times n_2$  matrix of tubes.  $\hat{\mathcal{A}}$  is a tensor which is obtained by taking the fast Fourier transform (FFT) along the third mode of  $\mathcal{A}$ . For a compact notation we will use  $\hat{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$  to denote the FFT along the third dimension. In the same way, we can also compute  $\mathcal{A}$  from  $\hat{\mathcal{A}}$  using the inverse FFT (IFFT).

**Definition 2.1 (t-product)** [12] The t-product  $\mathcal{E} = \mathcal{A} * \mathcal{B}$  of  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and  $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$  is an  $n_1 \times n_4 \times n_3$  tensor. The  $(i, j)$ -th tube of  $\mathcal{E}$  is given by

$$\mathcal{E}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{A}(i, k, :) \bullet \mathcal{B}(k, j, :), \quad (4)$$

where  $\bullet$  denotes the circular convolution between two tubes of same size.

**Definition 2.2 (conjugate transpose)** [14] The conjugate transpose of a tensor  $\mathcal{A}$  of size  $n_1 \times n_2 \times n_3$  is the  $n_2 \times n_1 \times n_3$  tensor

$\mathcal{A}^T$  obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slices from 2 to  $n_3$ .

**Definition 2.3 (identity tensor)** [14] The identity tensor  $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$  is a tensor whose first frontal slice is the  $n \times n$  identity matrix and all other frontal slices are zero.

**Definition 2.4 (orthogonal tensor)** [14] A tensor  $\mathcal{Q}$  is orthogonal if it satisfies

$$\mathcal{Q}^T * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^T = \mathcal{I}. \quad (5)$$

**Definition 2.5 (f-diagonal tensor)** [14] A tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.

**Definition 2.6 (t-SVD)** [14] For  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , the t-SVD of  $\mathcal{A}$  is given by

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T \quad (6)$$

where  $\mathcal{U}$  and  $\mathcal{V}$  are orthogonal tensors of size  $n_1 \times n_1 \times n_3$  and  $n_2 \times n_2 \times n_3$  respectively, and  $\mathcal{S}$  is a f-diagonal tensor of size  $n_1 \times n_2 \times n_3$ .

We can obtain this decomposition by computing matrix singular value decomposition (SVD) in the Fourier domain, as it shows in Algorithm 1. Fig. 4 illustrates the decomposition for the three-dimensional case.

---

### Algorithm1 : t-SVD for 3-way data

---

**Input:**  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

$\mathcal{D} \leftarrow \text{fft}(\mathcal{A}, [], 3)$ ,

**for**  $i = 1$  to  $n_3$ , **do**

$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathcal{D}(:, :, i))$ ,

$\hat{\mathcal{U}}(:, :, i) = \mathbf{U}$ ,  $\hat{\mathcal{S}}(:, :, i) = \mathbf{S}$ ,  $\hat{\mathcal{V}}(:, :, i) = \mathbf{V}$ ,

**end for**

$\hat{\mathcal{U}} \leftarrow \text{ifft}(\hat{\mathcal{U}}, [], 3)$ ,  $\hat{\mathcal{S}} \leftarrow \text{ifft}(\hat{\mathcal{S}}, [], 3)$ ,  $\hat{\mathcal{V}} \leftarrow \text{ifft}(\hat{\mathcal{V}}, [], 3)$ ,

**Output:**  $\mathcal{U}, \mathcal{S}, \mathcal{V}$

---

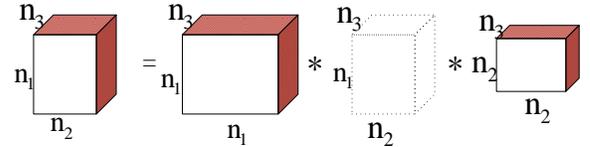


Fig. 4: Illustration of the t-SVD of an  $n_1 \times n_2 \times n_3$  tensor.

**Definition 2.7 (tensor multi-rank and tubal rank)** [9] The tensor multi-rank of  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is a vector  $\mathbf{r} \in \mathbb{R}^{n_3}$  with its  $i$ -th entry as the rank of the  $i$ -th frontal slice of  $\hat{\mathcal{A}}$ , i. e.  $r_i = \text{rank}(\hat{\mathcal{A}}(:, :, i))$ . The tensor tubal rank, denoted by  $\text{rank}_t(\mathcal{A})$ , is defined as the number of nonzero singular tubes of  $\mathcal{S}$ , where  $\mathcal{S}$  is from  $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ , i. e.

$$\text{rank}_t(\mathcal{A}) = \#\{i : \mathcal{S}(i, :, :) \neq 0\} = \max_i r_i \quad (7)$$

**Definition 2.8 (tensor nuclear norm: TNN)** [9] The tensor nuclear norm of  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  denoted by  $\|\mathcal{A}\|_*$  is defined as the sum of the singular values of all the frontal slices of  $\hat{\mathcal{A}}$ . The TNN of  $\mathcal{A}$  is equal to the TNN of  $\text{blkdiag}(\hat{\mathcal{A}})$ . Here  $\text{blkdiag}(\hat{\mathcal{A}})$  is a block diagonal matrix defined as follows:

$$\text{blkdiag}(\hat{\mathcal{A}}) = \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(n_3)} \end{bmatrix}, \quad (8)$$

where  $\hat{A}^{(i)}$  is the  $i$ -th frontal slice of  $\hat{A}$ ,  $i = 1, 2, \dots, n_3$ .

**Definition 2.9 (standard tensor basis)** [12] The column basis, denoted as  $\hat{e}_i$ , is a tensor of size  $n \times 1 \times n_3$  with its  $(i, 1, 1)$ -th entry equaling to 1 and the rest equaling to 0. Naturally its transpose  $\hat{e}_i^T$  is called row basis.

### 3. ITERATIVE BLOCK TENSOR SINGULAR VALUE THRESHOLDING

We decompose the whole tensor which satisfies the incoherence conditions into many small blocks of the same size. And the third dimension of the block size should be the same as the third dimension of the tensor. That is to say, given an input tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and its corresponding block size, we propose a multi-block tensor modeling that models the tensor data  $\mathcal{X}$  as the concatenation of block tensors. And each block tensor can be decomposed into two components, i.e.  $\mathcal{X}_p = \mathcal{L}_p + \mathcal{S}_p$ ,  $p = 1, \dots, P$ , where  $\mathcal{L}_p$  and  $\mathcal{S}_p$  denote the low rank component and sparse component of block tensor  $\mathcal{X}_p$  respectively.

As observed in RPCA, the low rank and sparse decomposition is impossible in some cases [4]. Similarly, we are not able to identify the low rank component and sparse component if the tensor is of both low rank and sparsity. Similar to the tensor incoherence conditions [8], we assume the block tensor data  $\mathcal{L}_p$  in each block satisfies some block tensor incoherence conditions to guarantee successful low rank component extraction.

**Definition 3.1 (block tensor incoherence conditions)** For  $\mathcal{L}_p \in \mathbb{R}^{n \times n \times n_3}$ , assume that  $\text{rank}_t(\mathcal{L}_p) = r$  and it has the  $t$ -SVD  $\mathcal{L}_p = \mathcal{U}_p * \mathcal{S}_p * \mathcal{V}_p^T$ , where  $\mathcal{U}_p \in \mathbb{R}^{n \times r \times n_3}$  and  $\mathcal{V}_p \in \mathbb{R}^{n \times r \times n_3}$  satisfy  $\mathcal{U}_p^T * \mathcal{U}_p = \mathcal{I}$  and  $\mathcal{V}_p^T * \mathcal{V}_p = \mathcal{I}$ , and  $\mathcal{S}_p \in \mathbb{R}^{r \times r \times n_3}$  is an  $f$ -diagonal tensor. Then  $\mathcal{L}_p$  satisfies the tensor incoherence conditions with parameter  $\mu$  if

$$\max_{i=1, \dots, n} \|\mathcal{U}_p^T * \hat{e}_i\|_F \leq \sqrt{\frac{\mu r}{n n_3}} \quad (9)$$

$$\max_{j=1, \dots, n} \|\mathcal{V}_p^T * \hat{e}_j\|_F \leq \sqrt{\frac{\mu r}{n n_3}} \quad (10)$$

and

$$\|\mathcal{U}_p * \mathcal{V}_p^T\|_\infty \leq \sqrt{\frac{\mu r}{n^2 n_3^2}} \quad (11)$$

The incoherence condition guarantees that for small values of  $\mu$ , the singular vectors are not sparse. Then the tensor  $\mathcal{L}_p \in \mathbb{R}^{n \times n \times n_3}$  can be decomposed into low rank component and sparse component.

For extracting the low rank component from every block, we process the tensor nuclear norm of  $\mathcal{L}_p$ , i. e.  $\|\mathcal{L}_p\|_{\text{TNN}} = \|\text{blkdiag}(\hat{\mathcal{L}}_p)\|_*$ . Here we can use singular value thresholding operator in the Fourier domain to extract the low rank component of the block tensor [18, 19]. The proposed method is called iterative block tensor singular value thresholding (IBTSVT). The thresholding operator used here is the soft one  $D_\tau$  as follows:

$$D_\tau(\mathcal{L}_p) = \text{sign}(\text{blkdiag}(\hat{\mathcal{L}}_p))(|\text{blkdiag}(\hat{\mathcal{L}}_p)| - \tau)_+, \quad (12)$$

where “ $(\cdot)_+$ ” keeps the positive part.

After we extract the low rank component  $\mathcal{L} = \mathcal{L}_1 \boxplus \mathcal{L}_2 \boxplus \dots \boxplus \mathcal{L}_P$ , where  $\boxplus$  denotes concatenation operation, we can get the sparse component of the tensor by computing the  $\mathcal{S} = \mathcal{X} - \mathcal{L}$ . See Algorithm 2 in details.

In our method, the block size can't be too large. The large size of the block will make the sparse part contain some low rank component. And if the size of the block is too small, the computational

---

#### Algorithm 2: IBTSVT

---

**Input:** tensor data  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

**Initialize:** given  $\mu, \eta, \epsilon, \tau$ , and

block tensors  $\mathcal{X}_p$  of size  $n \times n \times n_3$ ,  $p = 1, \dots, P$

**while** not converged **do**

1. Update  $\eta := \eta \times \mu$ ,

2. Update  $\tau := \tau / \eta$ ,

3. Compute  $\mathcal{X}_p := D_\tau(\mathcal{X}_p)$ ,  $p = 1, \dots, P$ .

**end while:**

$\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \leq \epsilon$  at  $(k+1)$ -th step.

**Output:**  $\mathcal{L} = \mathcal{X}_1 \boxplus \mathcal{X}_2 \boxplus \dots \boxplus \mathcal{X}_P$

---

time will be long. Because the number of t-SVDs is large. Generally, we can choose our block size  $2 \times 2 \times n_3$ .

In our algorithm, we choose  $\mu = 1.8$ ,  $\eta = 1$ ,  $\epsilon = 10^{-2}$ . But the thresholding parameter  $\tau$  is difficult to determine. Here we can get a value by experience. As discussed in [8], the thresholding parameter could be  $\tau = 1/\sqrt{nn_3}$  for every block. This value is for denoising problem in images or videos, where the noise is uniformly distributed. But for different applications, it should be different from  $1/\sqrt{nn_3}$ . Because in these applications, the sparse component in data is not uniformly distributed, such as shadow in face images and motion in surveillance videos.

## 4. EXPERIMENTAL RESULTS

In this section, we conduct numerical results to show the performance of the method. We apply IBTSVT method on two different real datasets that are conventionally used in low rank model: motion separation for surveillance videos (Section 4.1) and illumination normalization for face images (Section 4.2).

### 4.1. Motion Separation for Surveillance Videos

In surveillance video, the background only changes its brightness over the time, and it can be represented as the low rank component. And the foreground objects are the sparse component in videos. It is often desired to extract the foreground objects from the video. We use the proposed IBTSVT method to separate the foreground component from the background one.

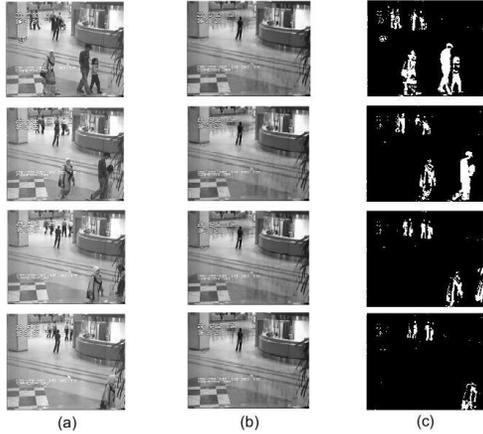
We use the surveillance video data used in [6]. Each frame is of size  $144 \times 176$  and we use 20 frames. The constructed tensor is  $\mathcal{X} \in \mathbb{R}^{144 \times 176 \times 20}$  and the selected block size is  $2 \times 2 \times 20$ . The thresholding parameter is  $\tau = 20/\sqrt{nn_3}$ .

Fig. 5 shows one of the results. We can find that IBTSVT method correctly recovers the background, while the sparse component correctly identifies the moving pedestrians. It shows the proposed method can realize motion separation for surveillance videos.

### 4.2. Illumination normalization for face images

The face recognition algorithms are sensitive to shadow or occlusions on faces [7], and it's important to remove illumination variations and shadow on the face images. The low rank model is often used for face images [20].

In our experiments, we use the Yale B face database [7]. Each face image is of size  $192 \times 168$  with 64 different lighting conditions. We construct the tensor data  $\mathcal{X} \in \mathbb{R}^{192 \times 168 \times 64}$  and choose the block size  $2 \times 2 \times 64$ . We set the thresholding parameter  $\tau = 20/\sqrt{nn_3}$ .



**Fig. 5:** IBTSVT on a surveillance video. (a) original video; (b) low rank component that is video background; (c) sparse component that represents the foreground objects of video.

We compare the proposed method with multi-scale low rank matrix decomposition method [13] and low rank + sparse method [4]. Fig. 6 shows one of the comparison results. The IBTSVT method can result in almost shadow-free faces. In contrast, the other two methods can only recover the faces with some shadow.

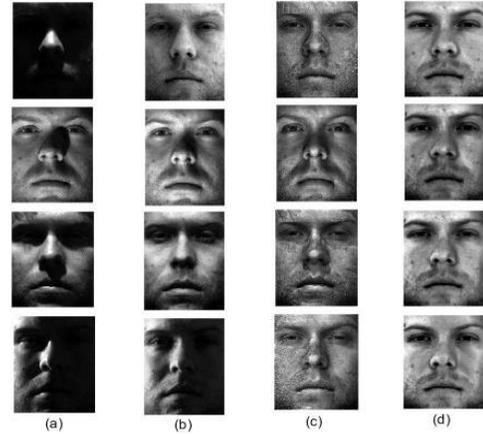
In order to further illustrate the effect of shadow elimination in the recovered face images, we carry on face detection with the recovered data from different methods. In our experiments, we employ the face detection algorithm Viola-Jones algorithm [21] to detect the faces and the eyes. The Viola-Jones algorithm is a classical algorithm which can be used to detect people’s faces, noses, eyes, mouths, and upper bodies. In the first experiment we put all face images into one image of JPG format. Then we use the algorithm to detect faces in the newly formed image. In the second experiment, we use the algorithm to detect the eyes of every face image. The second and third columns of Table 1 show the detection accuracy ratios of Viola-Jones algorithm with different recovered face images. We test how long the three methods process the 64 face images as can be seen in the fourth column. The IBTSVT can improve the efficiency by parallel processing of the block tensors. From the result of Table 1, we can find our method gives the best detection performance, because removing shadow of face images is helpful for face detection.

	face detection	eye detection	Time (s)
Original image	0.297	0.58	NULL
Low rank + sparsity	0.375	0.70	10
Multiscale low rank	0.359	1.00	4472
IBTSVT	<b>0.844</b>	<b>1.00</b>	<b>715</b>

**Table 1:** The accuracy ratios of faces and eyes detection by Viola-Jones algorithm and the computational time to process face images.

## 5. CONCLUSIONS

In this paper, we proposed a novel IBTSVT method to extract the low rank component of the tensor using t-SVD. The IBTSVT is a good way to utilize the structural feature of tensor by solving TPCA problem in block tensor form. We have given the tensor incoherence



**Fig. 6:** Three methods for face with uneven illumination: (a) original faces with shadows; (b) low rank + sparse method; (c) multi-scale low rank decomposition; (d) IBTSVT.

conditions for block tensor data. For applications, we considered motion separation for surveillance videos and illumination normalization for face images, and numerical experiments showed its performance gains compared with the existing methods.

## 6. ACKNOWLEDGMENT

This work is supported by National High Technology Research and Development Program of China (863, No. 2015AA015903), National Natural Science Foundation of China (NSFC, No. 61602091, No. 61602091), the Fundamental Research Funds for the Central Universities (No. ZYGX2015KYQD004, No. ZYGX2015KYQD004), and a grant from the Ph.D. Programs Foundation of Ministry of Education of China (No. 20130185110010).

## 7. REFERENCES

- [1] T. G. Kolda and J. Sun, “Scalable tensor decompositions for multi-aspect data mining,” in *IEEE International Conference on Data Mining*. IEEE, 2008, pp. 363–372.
- [2] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics Springer, Berlin, 2001.
- [3] I. Jolliffe, *Principal Component Analysis*, New York: Springer, 2002.
- [4] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis,” *Journal of the ACM*, vol. 58, no. 3, pp. 1–73, 2011.
- [5] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2010.
- [6] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–72, 2004.

- [7] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
- [8] C. Lu, Y. Chen, J. Feng, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.
- [9] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," *Computer Science*, vol. 44, no. 9, pp. 3842–3849, 2014.
- [10] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2014.
- [11] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [12] Z. Zhang and S. Aeron, "Exact tensor completion using t-svd," *arXiv preprint arXiv:1502.04689*, 2015.
- [13] F. Ong and M. Lustig, "Beyond low rank + sparse: Multiscale low rank matrix decomposition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 672–687, 2015.
- [14] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.
- [15] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and Its Applications*, vol. 433, no. 7, pp. 1241–1253, 2010.
- [16] M. E. Kilmer, E. Misha, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.
- [17] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 66, no. 4, pp. 294–310, 2005.
- [18] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2008.
- [19] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra and Its Applications*, vol. 170, no. 6, pp. 33–45, 1992.
- [20] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.
- [21] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*. IEEE, 2001, vol. 1, pp. I–511.