Evaluation of Different Algorithms of Nonnegative Matrix Factorization in Temporal PsychoVisual Modulation

Jianzhou Feng, Xiaoming Huo, *Senior Member, IEEE,* Li Song, *Member, IEEE,* Xiaokang Yang, *Senior Member, IEEE,* and Wenjun Zhang, *Fellow, IEEE*

***Abstract*—Temporal PsychoVisual Modulation (TPVM) is a newly proposed information display paradigm, which can be implemented by nonnegative matrix factorization (NMF) with additional upper bound constraints on the variables. In this paper, we study all the state-of-the-art algorithms in NMF, extend them to incorporate the upper bounds and discuss their potential use in TPVM. By comparing all the NMF algorithms with their extended versions, we find that: 1) the factorization error of the truncated alternating least squares algorithm always fluctuates throughout the iterations, 2) the alternating nonnegative least squares based algorithms may slow down dramatically under the upper bound constraints, 3) the hierarchical alternating least squares (HALS) algorithm converges the fastest and its final factorization error is often the smallest among all the algorithms. Based on the experimental results of the HALS, we propose a guideline of determining the parameter setting of TPVM, i.e., the number of viewers to support and the scaling factor for adjusting the light intensity of the images formed by TPVM. Our work will facilitate the applications of TPVM.**

***Index Terms*—Nonnegative matrix factorization, Temporal PsychoVisual Modulation, multiview video display, virtual reality.**

I. Introduction

A new information display paradigm, named Temporal PsychoVisual Modulation (TPVM), has been proposed in [1]. TPVM aims to offer different 2-D optical signals to different viewers, while using the same display medium. The architecture of a typical TPVM system consists of a high-speed display and several display synchronized modulation devices. Each modulation device modulates the amplitude of the displayed frames with a time variant weight. It is shown in psychophysics of vision that multiple optical signals in a small time interval would be fused by the human visual systems (HVS) and perceived as one (denote the duration of such time interval as the critical time t_c). Since the speed of the display is high, in t_c seconds, a certain number of emitted frames would be modulated and perceived differently by the viewers when different modulation weights are applied. The active liquid crystal (LC) glasses is one of the modulation devices that can be used in TPVM. It can switch the amount of the light pass through and enter the viewer's eyes at high frequency [2]. Since the LC glasses is a light blocking device and the light energy cannot be negative, the modulation weight it can implement should be within $[0, 1]$.

TPVM can be realized via Nonnegative Matrix Factorization (NMF) [3] with additional upper bound constraints on the variables. In order to justify this statement, we first analyze how the TPVM system works during each t_c seconds. Let f_d be the refresh rate of a digital display so that $M = t_c f_d$ atom frames $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ are emitted in the time interval. The number of pixels in each frame is denoted as N, i.e., $\mathbf{x}_i \in \mathbb{R}^N$, for $i = 1, 2, \cdots, M$. We further denote $\mathbf{X}$ as the matrix contains all these atom frames as its columns. Suppose there are totally K viewers, for $j = 1, 2, \cdots, K$, a modulation vector $\mathbf{w}_j$ is transmitted to the j-th LC glasses, and drives it to modulate the i-th atom framed by weight $\mathbf{W}_{ij}$, the i-th element of $\mathbf{w}_j$. Thus, the j-th viewer would finally perceive frame $\mathbf{y}_j = \mathbf{X}\mathbf{w}_j$, fused by modulated frame $\mathbf{W}_{ij}\mathbf{x}_i$'s, for $i = 1, 2, \cdots, M$. In real applications, the process is an inversion, i.e., $\mathbf{y}_j$'s are target images and $\mathbf{x}_i$'s and $\mathbf{w}_j$'s need to be computed in order to make $\mathbf{X}\mathbf{w}_j$'s as close to $\mathbf{y}_j$'s as possible. The computational process of $\mathbf{x}_i$'s and $\mathbf{w}_j$'s is in fact a constrained NMF problem:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{W}} \quad & \|s\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2, \\ \text{subject to} \quad & 0 \le \mathbf{X} \le 1, 0 \le \mathbf{W} \le 1, \end{aligned} \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{N\times K}$ contains all $\mathbf{y}_j$'s as its columns and the columns of $\mathbf{W} \in \mathbb{R}^{M\times K}$ are $\mathbf{w}_j$'s; In (1), the Frobenius norm $\|\cdot\|_F$ is used to measure the distance between the target images and the reconstructed ones. $s \in [1, M]$ is a scaling factor to ensure adequate light intensity to enter the viewer's eyes. The bounded constraints on the variables, $0 \le \mathbf{X}, \mathbf{W} \le 1$, are imposed since the intensity of the pixels in digital images is within $[0, 1]$ and the LC glasses can only realize weights in $[0, 1]$ as aforementioned. After the factorization is completed, $\mathbf{x}_i$'s are displayed on the medium and modulated by LC glasses using $\mathbf{w}_j$'s to render $\mathbf{y}_j$'s. The reconstruction errors can be made small for sufficiently large M, i.e., high refresh frequency f_d, and if images $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K$ are similar to each other, such as the 2-D projections of the same 3-D

This work was supported by NSFC (61221001, 61025005), 863 project (2012AA011703), the 111 Project (B07022) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

J. Feng, L. Song, X. Yang and W. Zhang are with Shanghai Digital Media Processing and Transmission Key Lab, Shanghai Jiaotong University, Shanghai, 200240 China, e-mail: {bravefjz, song_li, xkyang, zhangwenjun}@sjtu.edu.cn.

X. Huo is with School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA, e-mail: xiaoming.huo@isye.gatech.edu.

scene with respect to smoothly changing viewing angles. Note that in TPVM, when we achieve $K > M$ and $s > 1$, the resulting displaying system can support more viewers with brighter frames than the temporal multiplexing (which achieves $K = M$ and $s = 1$).

TPVM is closely related to the multiview video technology that is discussed in, e.g., [4], [5], [6]. The multiview videos or images are the common theme. However, the aforementioned papers addressed the multiview video coding (MVC), and our paper does not work on the coding aspect at all. It is convincingly argued in [1] that the unique ability of TPVM to generate different concurrent interference-free views on the same display medium adds a valuable social dimension to applications of virtual reality (VR), augmented reality (AR), and visualization (cf. [7]). TPVM display paradigm can greatly reduce the computational power and video memory bandwidth required by real-time VR/AR applications, because a small number of atom frames, which can be precomputed for a particular 3-D environment, are used to synthesize a range of different perspective views through appropriate attenuations of the active viewing devices and psychovisual processing of HVS.

In this paper, we aim to answer the following questions: 1) how to adapt the existing NMF algorithms to meet the additional upper bound constraints on the variables? 2) how the adaption would affect the factorization errors and convergence rates of different algorithms? 3) how to set the parameter K and s in TPVM, under required visual quality, convergence rate, number of supported viewers, and power cost?

This is not the first time NMF with additional bounded constraints on the variables has been studied. Some of the NMF algorithms have already been able to deal with such constraints since they were proposed, e.g., [8], [9] and [10]. However, it would be the first time to extend most of the existing NMF algorithms to incorporate the upper bound constraints for a particular application (namely, TPVM) and carry out a comprehensive comparison between these algorithms.

In the rest of this paper, we introduce the existing NMF algorithms in Section II, and modify them to adapt to the TPVM constraints (by following their original ideas) in Section III. In Section IV, we first do experiments to evaluate different NMF algorithms with their modified version, measured by the factorization error and the convergence rate. It is shown that the truncated alternating least squares method seldom converges and alternating nonnegative least squares based algorithms may slow down dramatically under the additional constraints, especially when s is large, while the hierarchical alternating least squares (HALS) algorithm performs well consistently. Subsequently, we apply HALS to various K's and s's and based on the experimental results, we set a guideline of choosing K and s with M fixed in real TPVM applications. Section V concludes the paper and discusses some future works.

II. Existing NMF Algorithms

The data analysis technique, NMF, is very popular and has been successfully used in many different areas such as hyperspectral imaging [11], text mining [12], clustering [13], air emission control [14], blind source separation [15] and music analysis [16].

NMF problem:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{W}} \quad & \|s\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2, \\ \text{subject to} \quad & \mathbf{X} \geq 0, \mathbf{W} \geq 0 \end{aligned} \tag{2}$$

is a non-convex optimization problem with respect to variables $\mathbf{X}$ and $\mathbf{W}$. Solving the NMF is shown to be NP-hard [17]. Hence, one can only hope to find a local minimum in practice. Nowadays, most methods follow the general iterative framework: during the k-th iteration,

- fix $\mathbf{W}^k$ and find $\mathbf{X}^{k+1} \geq 0$ such that
$$\left\|s\mathbf{Y} - \mathbf{X}^{k+1}\mathbf{W}^k\right\|_F^2 \leq \left\|s\mathbf{Y} - \mathbf{X}^k\mathbf{W}^k\right\|_F^2;$$
- fix $\mathbf{X}^{k+1}$ and find $\mathbf{W}^{k+1} \geq 0$ such that
$$\left\|s\mathbf{Y} - \mathbf{X}^{k+1}\mathbf{W}^{k+1}\right\|_F^2 \leq \left\|s\mathbf{Y} - \mathbf{X}^{k+1}\mathbf{W}^k\right\|_F^2.$$

Notice that

$$\left\|s\mathbf{Y} - \mathbf{X}^{k+1}\mathbf{W}^{k+1}\right\|_F = \left\|s\mathbf{Y}^T - (\mathbf{W}^{k+1})^T(\mathbf{X}^{k+1})^T\right\|_F,$$

which means the same algorithm for finding $\mathbf{X}^{k+1}$ can be used for finding $(\mathbf{W}^{k+1})^T$. Thus only one updating algorithm need to be designed. Based on their updating approaches, numerous NMF algorithms can be broadly classified into three categories:

C1: multiplicative update algorithms,
C2: alternating least squares (ALS) algorithms, and
C3: alternating nonnegative least squares (ANLS) algorithms.

Category *C1* and *C2* are different by their ways to insure the non-increasingness property shown in the general framework, while *C3* aims to find the exact solutions

$$\mathbf{X}^{k+1} = \arg\min_{\mathbf{X}\geq 0} \left\|s\mathbf{Y} - \mathbf{X}\mathbf{W}^k\right\|_F^2 \tag{3}$$

and

$$\mathbf{W}^{k+1} = \arg\min_{\mathbf{W}\geq 0} \left\|s\mathbf{Y} - \mathbf{X}^{k+1}\mathbf{W}\right\|_F^2, \tag{4}$$

respectively. Below is a representative list of the algorithms that have appeared in the literature. The first item is in *C1*. The next two are in *C2*. The remaining methods are in *C3*.

1) MU: the multiplicative update method in Lee and Seung [18]. However, the multiplicative update algorithm lacks the convergence guarantee [19].
2) TALS: the truncated alternating least squares method proposed in Berry *et al.* [20]. It was initially proposed by Paatero and Tapper in [14], but the algorithm in [14] did not rigorously deal with the nonnegativity constraints. The TALS algorithm also has difficulty with convergence.
3) HALS: the hierarchical alternating least squares algorithm introduced by Cichocki *et al.* [21], [22].
4) ANLS-BPP: ANLS with the block principal pivoting method in Kim and Park [23], [24].
5) ANLS-AS: ANLS with the active set method proposed by Kim and Park [25].

6) ANLS-PGRAD: ANLS with the projected gradient method in Lin [9].
7) ANLS-PQN: ANLS with the projected quasi-Newton method in Kim *et al.* [26], [27].
8) ANLS-PNM: ANLS with the projected Newton method in Gong and Zhang [28].

Matlab implementations for the above methods can be found at

- MU, HALS, ANLS-PGRAD: https://sites.google.com/site/nicolasgillis/code,
- TALS, ANLS-PQN: http://www.cs.utexas.edu/~dmkim/Source/software/,
- ANLS-BPP, ANLS-AS: http://www.cc.gatech.edu/~jingu/nmf/index.html,
- ANLS-PNM: http://sites.google.com/site/pinghuag/home/pnm.

Recently, Gillis proposed an algorithm, which is based on preprocessing the input matrix $s\mathbf{Y}$ before factorization [29]. By using the preprocessed matrix, hopefully, an unique and sparser solution would be obtained by applying any of the listed NMF algorithms. We found the idea of preprocessing inspiring. However, its adaptation with TPVM is beyond the scope of this paper. We may consider it in a future study.

III. Extending Existing Algorithms under TPVM

We give an overview of three main categories of existing algorithms for NMF and extend each particular algorithm under the additional constraints, $\mathbf{X} \leq 1$ and $\mathbf{W} \leq 1$, which are introduced in TPVM. Since the additional constraints are symmetric for $\mathbf{X}$ and $\mathbf{W}$, we just present the original algorithms and their extensions on updating $\mathbf{X}$ with fixed $\mathbf{W}$.

A. Notations

Before explaining the algorithms, we introduce some notations.

- For matrix $\mathbf{A}$, $\mathbf{A}_{ij}$ is its component at i-th row and j-th column. $\mathbf{A}_{i:}$ is the i-th row of $\mathbf{A}$ and $\mathbf{A}_{:j}$ is the j-th column of $\mathbf{A}$.
- For matrix $\mathbf{A}$ and interval $[a, b]$, $[\mathbf{A}]_{[a,b]}$'s component
$$[\mathbf{A}_{ij}]_{[a,b]} = \min\{b, \max\{a, \mathbf{A}_{ij}\}\}$$
and $[\mathbf{A}]_+ \triangleq [\mathbf{A}]_{[0,+\infty)}$'s component
$$[\mathbf{A}_{ij}]_+ = \max\{0, \mathbf{A}_{ij}\},$$
where "$\triangleq$" is the *definition* symbol.
- For matrix $\mathbf{A}$, $\mathrm{vec}(\mathbf{A})$ is the concatenation of all $\mathbf{A}$'s column $\mathbf{A}_{:j}$'s.
- For $N \times M$ matrix $\mathbf{A}$ and index set $I' \subseteq I \triangleq \{(i,j)|1 \leq i \leq N, 1 \leq j \leq M\}$, $\mathrm{vec}(\mathbf{A}|I')$ is the concatenation of all $\mathbf{A}_{i,j}$'s satisfying $(i,j) \in I'$.
- For matrix $\mathbf{A}$ and $\mathbf{B}$ with the same size, $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$ is the component-wise product, i.e., $\mathbf{C}_{ij} = \mathbf{A}_{ij}\mathbf{B}_{ij}$.
- For matrix $\mathbf{A}$ and $\mathbf{B}$ with the same size, $\mathbf{C} = \frac{[\mathbf{A}]}{[\mathbf{B}]}$ is the component-wise division, i.e., $\mathbf{C}_{ij} = \mathbf{A}_{ij}/\mathbf{B}_{ij}$.
- For matrix $\mathbf{A}$ and $\mathbf{B}$ with the same size, $\mathbf{C} = \max\{\mathbf{A}, \mathbf{B}\}$ (resp., $\mathbf{C} = \min\{\mathbf{A}, \mathbf{B}\}$) is the component-wise maximum (resp., minimum), i.e., $\mathbf{C}_{ij} = \max\{\mathbf{A}_{ij}, \mathbf{B}_{ij}\}$ (resp., $\mathbf{C}_{ij} = \min\{\mathbf{A}_{ij}, \mathbf{B}_{ij}\}$).
- The objective function for updating $\mathbf{X}$ with $\mathbf{W}$ fixed is $g(\mathbf{X}) \triangleq \frac{1}{2}\|s\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2$.

B. Multiplicative Update (MU) Algorithm

The MU algorithm updates $\mathbf{X}$ component-wise by multiplying its current value by some factor that depends on the quality of the approximation. The particular method is

$$\mathbf{X} \leftarrow \mathbf{X} \circ \frac{[s\mathbf{Y}\mathbf{W}^T]}{[\mathbf{X}\mathbf{W}\mathbf{W}^T]}, \tag{5}$$

which can be further interpreted into its gradient descent form

$$\mathbf{X} \leftarrow \mathbf{X} + \frac{[\mathbf{X}]}{[\mathbf{X}\mathbf{W}\mathbf{W}^T]} \circ (s\mathbf{Y}\mathbf{W}^T - \mathbf{X}\mathbf{W}\mathbf{W}^T), \tag{6}$$

where $s\mathbf{Y}\mathbf{W}^T - \mathbf{X}\mathbf{W}\mathbf{W}^T = -\nabla g(\mathbf{X})$ is the descent direction and $[\mathbf{X}]/[\mathbf{X}\mathbf{W}\mathbf{W}^T]$ consists all the component-wise step sizes. This group of step sizes insure the nonnegativity of $\mathbf{X}$ and guarantee non-increasing $g(\mathbf{X})$ through updating [18].

Two issues may be concerned in the implementation. One is how to avoid division by zero when $\mathbf{X}\mathbf{W}\mathbf{W}^T$ contains zero entries. The other is that some entries in $\mathbf{X}$ or $\mathbf{W}$ may be zero due to a bad initialization or the finite machine precision, and these entries will remain zero in all subsequent iterations. In [30], the authors proposed a new technique, which replaces entries in $\mathbf{X}$ and $\mathbf{W}$ smaller than a small quantity δ to be δ. Using this technique, $\mathbf{X}\mathbf{W}\mathbf{W}^T$ will no longer contains any zero entries and MU is proven to converge [31].

To fit for the additional constraint, we set up an upper bound of the component-wise step size to be

$$\eta = \min\left\{\frac{[\mathbf{X}]}{[\mathbf{X}\mathbf{W}\mathbf{W}^T]}, \frac{[\mathbf{1}_N\mathbf{1}_M^T - \mathbf{X}]}{[s\mathbf{Y}\mathbf{W}^T]}\right\}. \tag{7}$$

It can be shown that:

Theorem 1. *The updated* $\mathbf{X}' = \mathbf{X} + \eta \circ (s\mathbf{Y}\mathbf{W}^T - \mathbf{X}\mathbf{W}\mathbf{W}^T)$ *with* η *defined in (7) satisfies*

- $0 \leq \mathbf{X}' \leq 1$ *and*
- $g(\mathbf{X}') \leq g(\mathbf{X})$.

The proof is referred to Appendix A.

C. Alternating Least Squares Methods

Denote $I \triangleq \{(i,j)|1 \leq i \leq N, 1 \leq j \leq M\}$ as the index set of all $\mathbf{X}$'s components. For any $I' \subseteq I$, the ALS scheme updates concatenated components $\mathrm{vec}(\mathbf{X}|I')$ with two steps: a least squares step followed by a projection step. The least squares step solves

$$\mathrm{vec}(\mathbf{X}|I')^* = \arg\min_{\mathrm{vec}(\mathbf{X}|I')} g(\mathbf{X}) \tag{8}$$

with $\mathrm{vec}(\mathbf{X}|I \setminus I')$ fixed. It is an unconstrained quadratic program on $\mathrm{vec}(\mathbf{X}|I')$ and the closed-form solution $\mathrm{vec}(\mathbf{X}|I')^*$ can be easily derived. The projection step sets $\mathrm{vec}(\mathbf{X}|I') \leftarrow [\mathrm{vec}(\mathbf{X}|I')^*]_+$, which insures the nonnegativity of the updated $\mathbf{X}$.

In the TALS algorithm, $I' = I$ which means all the entries of $\mathbf{X}$ are updated simultaneously. The approximation error may increase during the updating due to the projection step.

In the HALS algorithm, all the M columns of $\mathbf{X}$ are updated sequentially (notice any ordering for the updating of the columns of $\mathbf{X}$ is possible), the j-th column of $\mathbf{X}$ is updated by setting $I' = \{(i,j)|1 \leq i \leq N\}$, i.e., $\mathbf{X}_{:j} = \mathrm{vec}\,(\mathbf{X}|I')$. Using such I''s, the least square problem (8) can be written as

$$\mathbf{X}^*_{:j} = \arg\min_{\mathbf{X}_{:j}} \left\| (s\mathbf{Y} - \sum_{l \neq j} \mathbf{X}_{:l}\mathbf{W}_{l:}) - \mathbf{X}_{:j}\mathbf{W}_{j:} \right\|_F^2. \quad (9)$$

It can be further decoupled into N independent quadratic programs in one variable $\mathbf{X}_{ij}$ as

$$\mathbf{X}^*_{ij} = \arg\min_{\mathbf{X}_{ij}} \left\| (s\mathbf{Y}_{i:} - \sum_{l \neq j} \mathbf{X}_{il}\mathbf{W}_{l:}) - \mathbf{X}_{ij}\mathbf{W}_{j:} \right\|_2^2. \quad (10)$$

Thus, HALS updates $\mathbf{X}_{ij} \leftarrow [\mathbf{X}^*_{ij}]_+$.

It is easy to see that, $\forall [a,b]$, $[\mathbf{X}^*_{ij}]_{[a,b]}$ is the optimal solution of (10), subject to $\mathbf{X}_{ij} \in [a,b]$. In the NMF case, $[a,b] = [0,+\infty)$ so that $\mathbf{X}_{ij} \leftarrow [\mathbf{X}^*_{ij}]_+ = [\mathbf{X}^*_{ij}]_{[a,b]}$ guarantees the non-increasingness property of HALS.

The ALS methods are amenable to the constraint $\mathbf{X} \leq 1$ by extending the projection step:

$$\mathrm{vec}\,(\mathbf{X}|I') \leftarrow [\mathrm{vec}\,(\mathbf{X}|I')^*]_{[0,1]}. \quad (11)$$

Such modification won't affect the non-increasingness property of HALS, as analyzed above.

D. Alternating Nonnegative Least Squares Methods

The ANLS scheme updates $\mathbf{X}$ as the solution of the nonnegative least squares problem (NNLS)

$$\mathbf{X}^* = \arg\min_{\mathbf{X} \geq 0} g(\mathbf{X}), \quad (12)$$

with $\mathbf{W}$ fixed. Various algorithms (belonging to the ANLS category) differ on the techniques that are used for computing the optimal solutions to the NNLS problem. The ANLS framework insures to find a stationary point. Convergence speed is the main issue for comparing the ANLS based algorithms.

1) ANLS-BPP and ANLS-AS: Function (12) can be decoupled into N independent NNLS problems:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \geq 0} \left\{ f(\mathbf{x}) \triangleq \frac{1}{2}\|s\mathbf{Y}_{i:} - \mathbf{x}^T\mathbf{W}\|_2^2 \right\}, \quad (13)$$

where each solution $\mathbf{x}^*$ corresponds to the transpose of a row $\mathbf{X}_{i:}$, for $1 \leq i \leq N$. As stated in [32], the Karush-Kuhn-Tucker (KKT) optimal conditions of (13)

$$\begin{aligned} &\lambda = \nabla f(\mathbf{x}) = \mathbf{W}\mathbf{W}^T\mathbf{x} - s\mathbf{W}(\mathbf{Y}_{i:})^T, \\ &\lambda \geq 0, \quad \mathbf{x} \geq 0, \quad \mathbf{x} \circ \lambda = 0 \end{aligned} \quad (14)$$

is a special case of the linear complementarity problem (LCP)

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad \mathbf{x} \geq 0, \quad \mathbf{y} \geq 0, \quad \mathbf{x} \circ \mathbf{y} = 0, \quad (15)$$

which seeks vectors $\mathbf{x}$ and $\mathbf{y}$ with real matrix $\mathbf{A}$ and vector $\mathbf{b}$ given. $\mathbf{x} \circ \mathbf{y} = 0$ is called the complementarity condition, which restricts either $\mathbf{x}$ or $\mathbf{y}$ to be 0 elementwise.

Thus, solving NNLS (13) is equivalent to solving LCP (14). Block principal pivoting (BPP) and active-set algorithm (AS) are methods designed for solving LCP and have been applied to NMF in ANLS-BPP and ANLS-AS, respectively. These two algorithms solve LCP iteratively, aiming to correctly divide the index set $I \triangleq \{1, \cdots, M\}$ of $\mathbf{x}$ into active set $\mathcal{A}$ and passive set $\mathcal{P}$, such that $\mathrm{vec}\,(\mathbf{x}^*|\mathcal{A}) = 0$ and $\mathrm{vec}\,(\mathbf{x}^*|\mathcal{P}) > 0$. Once $\mathcal{A}$ and $\mathcal{P}$ are derived, $\mathbf{x}^*$ is calculated as $\mathrm{vec}\,(\mathbf{x}^*|\mathcal{A}) = 0$ and $\mathrm{vec}\,(\mathbf{x}^*|\mathcal{P})$ being the solution of

$$\mathrm{vec}\left(\mathbf{W}\mathbf{W}^T\mathbf{x} - s\mathbf{W}(\mathbf{Y}_{i:})^T|\mathcal{P}\right) = \mathrm{vec}\,(\lambda|\mathcal{P}) = 0 \quad (16)$$

based on the KKT condition.

The general scheme of finding $\mathcal{A}$ and $\mathcal{P}$ is initializing $\mathcal{A} = I$ and $\mathcal{P} = \varnothing$, followed by exchanging variables between $\mathcal{A}$ and $\mathcal{P}$ iteratively until the KKT optimality conditions are satisfied.

During each iteration, $\mathcal{A}$ and $\mathcal{P}$ are used to compute a possible solution for (14). Firstly, we set $\mathrm{vec}\,(\mathbf{x}|\mathcal{A}) \leftarrow 0$ and $\mathrm{vec}\,(\lambda|\mathcal{P}) \leftarrow 0$ to fit $\mathbf{x} \circ \lambda = 0$. Then, $\mathrm{vec}\,(\mathbf{x}|\mathcal{P})$ is set by solving (16) and $\mathrm{vec}\,(\lambda|\mathcal{A}) \leftarrow \mathrm{vec}\,(\nabla f(\mathbf{x})|\mathcal{A})$. If $\mathrm{vec}\,(\mathbf{x}|\mathcal{P}) \geq 0$ and $\mathrm{vec}\,(\lambda|\mathcal{A}) \geq 0$, then the KKT conditions are satisfied, else the exceptional variables are candidates for exchanging. The main difference between ANLS-BPP and ANLS-AS is: ANLS-BPP exchanges multiple variables at a time while ANLS-AS only exchanges a single variable.

As shown in [32], when the additional constraint $\mathbf{x} \leq 1$ is introduced, the KKT optimal conditions

$$\begin{aligned} \lambda - \mu = \nabla f(\mathbf{x}) &= \mathbf{W}\mathbf{W}^T\mathbf{x} - s\mathbf{W}(\mathbf{Y}_{i:})^T \\ 0 \leq \mathbf{x} &\leq 1 \\ \lambda \geq 0 \quad &\text{and} \quad \mu \geq 0 \\ \mathbf{x} \circ \lambda = 0 \quad &\text{and} \quad (\mathbf{1}_M - \mathbf{x}) \circ \mu = 0, \end{aligned} \quad (17)$$

still constitute a LCP, which is

$$\begin{aligned} &\tilde{\lambda} = \begin{pmatrix} \mathbf{W}\mathbf{W}^T & \mathrm{I}_M \\ -\mathrm{I}_M & \mathbf{0} \end{pmatrix} \tilde{\mathbf{x}} - \begin{pmatrix} s\mathbf{W}(\mathbf{Y}_{i:})^T \\ -\mathbf{1}_M \end{pmatrix}, \\ &\tilde{\lambda} \geq 0, \quad \tilde{\mathbf{x}} \geq 0, \quad \tilde{\mathbf{x}} \circ \tilde{\lambda} = 0, \end{aligned} \quad (18)$$

where $\tilde{\lambda} = \begin{pmatrix} \lambda \\ \mathbf{1}_M - \mathbf{x} \end{pmatrix}$ and $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \mu \end{pmatrix}$.

Using the same technique in BPP and AS, $\tilde{\mathbf{x}}$ can be solved and $\mathbf{x}$ is just a part of $\tilde{\mathbf{x}}$. Notice that such extension doubles the size of index set for partition, which may largely increase the computational cost.

2) ANLS-PGRAD: The projected gradient method is applied for solving (12). It updates $\mathbf{X}$ iteratively based on gradient $\nabla g(\mathbf{X}) = (\mathbf{X}\mathbf{W} - s\mathbf{Y})\mathbf{W}^T$. In the k-th iteration, $\mathbf{X}$ is updated by

$$\mathbf{X}^{k+1} = \mathbf{X}(\eta^k) \triangleq [\mathbf{X}^k - \eta^k \nabla g(\mathbf{X}^k)]_+, \quad (19)$$

where η^k is the step size. In order to make the algorithm converge faster, an improved Armijo rule for selecting η^k is applied [9]. Given $0 < \sigma < 1$ and $0 < \beta < 1$, the improved Armijo rule finds the maximum η^k that satisfies

$$g(\mathbf{X}(\eta^k)) - g(\mathbf{X}^k) \leq \sigma \mathrm{vec}(\nabla g(\mathbf{X}^k))^T \mathrm{vec}(\mathbf{X}(\eta^k) - \mathbf{X}^k). \quad (20)$$

η^k is initialized as η^{k-1}. If (20) is satisfied, then $\eta^k \leftarrow \eta^k/\beta$ is tested until (20) is not satisfied or $\mathbf{X}(\eta^k) = \mathbf{X}(\eta^k/\beta)$. Else, $\eta^k \leftarrow \beta\eta^k$ is tested until (20) is satisfied.

When ANLS-PGRAD was proposed, the upper bound constraints $\mathbf{X} \leq 1$ was also considered [9], the particular technique is to re-denote

$$\mathbf{X}(\eta^k) \triangleq [\mathbf{X}^k - \eta^k \nabla g(\mathbf{X}^k)]_{[0,1]}, \tag{21}$$

apply the same algorithm to determine η^k, and finally update as $\mathbf{X}^{k+1} = \mathbf{X}(\eta^k)$.

3) ANLS-PQN: The projected quasi-Newton method is applied to solve the N independent NNLS problems (13). For each NNLS, $\mathbf{X}$ is iteratively updated by

$$\mathbf{x}^{k+1} = [\mathbf{x}^k - \eta^k \mathbf{D}^k \nabla f(\mathbf{x}^k)]_+ \tag{22}$$

at the k-th iteration, where η^k is the step size and $\mathbf{D}^k$ is chosen as the inverse of the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$ (or its approximated version via BFGS update [27] when the computation involving the Hessian is expensive). At each iteration, all the variables are firstly partitioned into two groups, namely the free and fixed variables. The fixed variables are the ones equal to 0 with positive derivative. Denote $I^k = \{i|\mathbf{x}_i^k = 0, (\nabla f(\mathbf{x}^k))_i > 0\}$ as the fixed set. Then ANLS-PQN restricts $\mathbf{D}_{ij}^k = 0$ for $i \in I^k$ or $j \in I^k$, and only aims to update variables in the free set based on the gradients. Finally, value of η^k is chosen by the line search [33].

ANLS-PQN handles constraint $X \leq 1$ without much additional difficulty: updating

$$\mathbf{x}^{k+1} = [\mathbf{x}^k - \eta^k \mathbf{D}^k \nabla f(\mathbf{x}^k)]_{[0,1]} \tag{23}$$

and enlarging the fixed set to contain variables equal to 1 with negative derivative

$$\begin{aligned} I^k = &\{i|\mathbf{x}_i^k = 0, (\nabla f(\mathbf{x}^k))_i > 0\} \\ &\cup \{i|\mathbf{x}_i^k = 1, (\nabla f(\mathbf{x}^k))_i < 0\}. \end{aligned} \tag{24}$$

4) ANLS-PNM: The projected Newton method [8] is applied to solve (12). It updates $\mathbf{X}$ iteratively as

$$\mathrm{vec}(\mathbf{X}^{k+1}) = [\mathrm{vec}(\mathbf{X}^k) - \eta^k (\mathbf{D}^k)^{-1} \mathrm{vec}(\nabla g(\mathbf{X}^k))]_+ \tag{25}$$

in the k-th iteration, where $\mathbf{D}^k$ is a positive definite symmetric matrix. However, an arbitrary positive definite choice of $\mathbf{D}^k$ may make it impossible to reduce the approximation error along direction $(\mathbf{D}^k)^{-1}\mathrm{vec}(\nabla g(\mathbf{X}^k))$ (i.e., $g(\mathbf{X}^{k+1}) \geq g(\mathbf{X}^k), \forall \eta^k \geq 0$). To this end, $\mathbf{D}^k$ is set by the following steps:

1: Find all the variables close to 0 with positive partial gradient

$$I^k = \left\{ i|0 \leq \mathrm{vec}(\mathbf{X}^k)_i \leq \epsilon^k, \frac{\partial g(\mathbf{X}^k)}{\partial \mathrm{vec}(\mathbf{X}^k)_i} > 0 \right\}, \tag{26}$$

where $\epsilon^k = \min\left\{\epsilon, \left\|\mathbf{X}^k - [\mathbf{X}^k - \nabla g(\mathbf{X}^k)]_+\right\|_F\right\}$ is a threshold.

2: Initialize $\mathbf{D}^k = \nabla^2 g(\mathbf{X}^k)$, whose component

$$\nabla^2 g(\mathbf{X}^k)_{ij} = \frac{\partial^2 g(\mathbf{X}^k)}{\partial \mathrm{vec}(\mathbf{X}^k)_i \partial \mathrm{vec}(\mathbf{X}^k)_j}, \tag{27}$$

since $(\nabla^2 g(\mathbf{X}^k))^{-1}\mathrm{vec}(\nabla g(\mathbf{X}^k))$ is the Newton descent direction.

3: Set $\mathbf{D}_{ij}^k = 0$ for $i \in I^k$ or $j \in I^k, i \neq j$. Using such $\mathbf{D}^k$, smaller updated approximation error is guaranteed under a proper step size when $\mathbf{X}^k$ is not a critical point. The proof can be seen in [8] (Proposition 1).

Then the value of η^k is chosen based on the Armijo-like rule [28] and $\mathbf{X}$ is updated using (25).

The additional constraint $\mathbf{X} \leq 1$ has already been considered in the original projected Newton method [8] as a special case of the general linear constraints. In summary, the modification on ANLS-PNM includes: updating

$$\mathrm{vec}(\mathbf{X}^{k+1}) = [\mathrm{vec}(\mathbf{X}^k) - \eta^k (\mathbf{D}^k)^{-1} \mathrm{vec}(\nabla g(\mathbf{X}^k))]_{[0,1]} \tag{28}$$

and finding all the variables close to 0 with positive partial gradient or close to 1 with negative partial gradient

$$\begin{aligned} I^k = &\left\{ i|0 \leq \mathrm{vec}(\mathbf{X}^k)_i \leq \epsilon^k, \frac{\partial g(\mathbf{X}^k)}{\partial \mathrm{vec}(\mathbf{X}^k)_i} > 0 \right\} \\ &\cup \left\{ i|1 - \epsilon^k \leq \mathrm{vec}(\mathbf{X}^k)_i \leq 1, \frac{\partial g(\mathbf{X}^k)}{\partial \mathrm{vec}(\mathbf{X}^k)_i} < 0 \right\}, \end{aligned} \tag{29}$$

where

$$\epsilon^k = \min\left\{\epsilon, \left\|\mathbf{X}^k - [\mathbf{X}^k - \nabla g(\mathbf{X}^k)]_{[0,1]}\right\|_F\right\}. \tag{30}$$

Compared with ANLS-PQN, ANLS-PNM is different on the following

- A general step size is chosen instead of a step size set, each one for a specific row $\mathbf{X}_{i:}$. The algorithm of selecting the step size is also different.
- The fixed set I^k is enlarged to contain variables close to the boundary within a threshold ϵ^k.
- ANLS-PNM restricts the Hessian matrix before inverting it while ANLS-PQN restricts the inversed Hessian matrix or its approximation via BFGS.

IV. Numerical Experiments

In this section, we evaluate the potential performance of TPVM using the existing NMF algorithms via numerical experiments. We consider two major questions:

- First of all, the current mathematical formulation of the TPVM is essentially a constrained NMF with upper bound constraints on $\mathbf{X}$ and $\mathbf{W}$. We have modified the corresponding algorithms of the ordinary NMF to the case including the aforementioned newly added upper bounds. Under the ordinary NMF, there are already comparison among the existing algorithms. However, when applied to TPVM, how the proposed modification would affect the convergence rates and the final factorization errors of different algorithms is unknown. One would like to see the exact impact of the modification and select the best algorithm for TPVM through numerical experiments.
- TPVM requires two algorithmic parameters: the number of supported viewer K and the scaling factor s for adjusting the intensity of target images. M is assumed to be fixed, since it mainly depends on the refresh rate f_d of the display device. On the other hand, in TPVM,

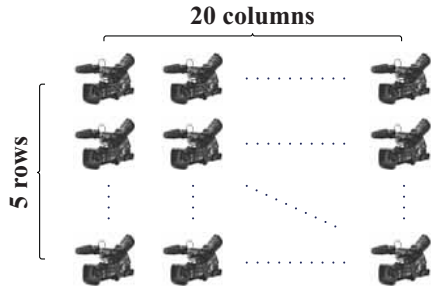


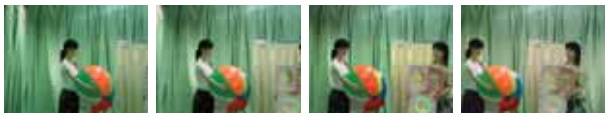
Fig. 1. Layout of a camera array

we are interested in four performance properties: 1) the visual quality, i.e., the fidelity to the original image set, 2) the convergence time of the factorization algorithm, 3) the number of supported viewer K, 4) the power used for display. We will use experiments to study the influence of these two parameters (i.e., K and s) to the above mentioned performance measures. The experimental results can help to select the optimal K and s in real applications.

A. Data sets and experiment setup

The data sets that we used for evaluation are

1) Akko: This dataset contains 30fps (frames per second) sequences of 100 views. The 100 cameras are equipped in a vertical plane as shown in Fig. 1. The resolution of each frame is 640×480. Each camera captures 290 frames in a sequence, thus the whole data set contains $100 \times 290 = 29,000$ frames totally. The data set can be downloaded from http://www.tanimoto.nuee.nagoya-u.ac.jp/~fukushima/mpegftv/Akko.htm.
2) Book: This data set contains 16.67fps sequences of 16 views. The 16 cameras are equipped in a horizontal line side by side. The origin resolution of each frame is 1024×768, while we use its downsampled version 512×384 for testing. Each camera captures 100 frames in a sequence, thus the whole data set contains $16 \times 100 = 1,600$ frames totally. The data set can be downloaded from http://sp.cs.tut.fi/mobile3dtv/stereo-video/.
3) Sea, Lake, Garden, Museum: These four data sets contain different perspective views of a particular 3-D environment. These 3-D scenes are synthesized by Microsoft Photosynth software. They can be obtained by accessing http://photosynth.net/default.aspx and searching for scene *black sea*, *Boardman Lake*, *Botanischer Garten Innsbruck MedienService* and *Museum KHM Spanish Saal*, respectively. For each environment, we capture about 200 frames under different rotation angle at the center of the environment. The resolution of each frame is 520×310.
4) Temple, Dino: These two data sets contain views sampled on a ring around the object. The resolution of each frame is 480×640 and the frame number is around 50. The data set can be downloaded from http://vision.middlebury.edu/mview/data/.



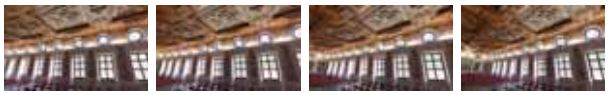
(a) Four views of Akko (Totally 100 views).

(b) Four views of Book (Totally 16 views).

(c) Four views of Sea (Totally 249 views).

(d) Four views of Lake (Totally 231 views).

(e) Four views of Garden (Totally 172 views).



(f) Four views of Museum (Totally 248 views).

(g) Five views of Temple (Totally 47 views).

(h) Five views of Dino (Totally 48 views).

Fig. 2. Samples from different data sets.

These data sets simulate several possible conditions in TPVM. As shown in Fig. 2,

- Akko and Book contain indoor sequences by cameras facing the same direction. They are used for simulating general multi-view TV watching.
- Sea, Lake, Garden and Museum (3 outdoor and 1 indoor) contain views captured at the center of a particular 3-D environment under different rotation angles. They are used for simulating observations of a place in virtual reality.
- Temple and Dino contain frames captured from a ring centered at a particular object. They are used for simu-

lating observations of an object in virtual reality.

In all the experiments, we transform the color images into gray ones to reduce the simulation time. Experiments with color videos will be a future endeavor.

We execute the experiments in MATLAB R2012b on a Windows Machine with an Intel(R) Core(TM) i7-3630QM CPU @2.40GHz and 8 GB memory. The online matlab codes of all the existing NMF algorithms are listed in Section II. We implement ANLS-PQN ourselves to achieve much faster execution speed, since the code written by its authors was not carefully optimized for high-dimensional data, in which NMF is typically used. It's unfair to compare ANLS-PQN with other algorithms by using the original code. The modified versions of these algorithms are implemented by us as described in Section III.

In the standard NMF algorithm, $\mathbf{X}$ and $\mathbf{W}$ are initialized with random nonnegative values, before the iteration starts. However, in a man-made engineering system like TPVM, everything should be deterministic, including the starting point. Thus, we design a simple deterministic initialization for $\mathbf{X}$ and $\mathbf{W}$ using the traditional time multiplexing like method:

- *Generate* $\mathbf{X}$ *by selecting* M *columns from* $\mathbf{Y}$: For $1 \leq m \leq M$, initializing $\mathbf{X}_{:m} = \mathbf{Y}_{:k}$, where $k = [K \cdot m/M]$. Operator $[a]$ rounds a to its nearest integer.
- *Factorize columns of* $s\mathbf{Y}$ *as the sum of* s *columns of* $\mathbf{X}$ (The scaling factor s is assumed to be an integer): For $1 \leq k \leq K$, we approximate $\mathbf{Y}_{:k} \approx \sum_{i=1}^{s} \mathbf{X}_{:m_i^{(k)}}$, where $\{m_1^{(k)}, \cdots, m_s^{(k)}\}$ indicate the s columns in $\mathbf{X}$ that are nearest to $\mathbf{Y}_{:k}$. This means we initialize $\mathbf{W}_{:k} = \sum_{i=1}^{s} \mathbf{e}_{m_i^{(k)}}$, where $\mathbf{e}_{m_i^{(k)}}$ is a vector with its $m_i^{(k)}$-th element equals to 1 and others equal to 0.

When $K = M$ and $s = 1$, it is exactly the traditional time multiplexing strategy. Compared with a randomly chosen starting point, the deterministic one is generally closer to a stationary point, thus the execution time required by NMF algorithms to converge can be much shorter. Through numerical experiments, we set the execution time to be 30s.

To evaluate different algorithms under the same parameter settings, K and s can be varied, we use two general measurements:

- Relative Objective Value (ROV):

$$R(K,s) = \|s\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F / \|s\mathbf{Y}\|_F. \tag{31}$$

  In the video processing society, the Signal-to-Noise Ratio (SNR), the Peak Signal-to-Noise Ratio (PSNR) and the Root Mean Square Error (RMSE) are frequently used for evaluating the quality of the reconstructed frames. In fact these three measurements are closely related to ROV, especially SNR. It is easy to see that $\|s\mathbf{Y}\|_F^2$ is the signal power and $\|s\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2$ is the noise power, thus

$$\mathrm{SNR} = \frac{1}{R(K,s)^2} \tag{32}$$

  Similarly, we can derive PSNR in dB and RMSE as

$$\mathrm{PSNR(dB)} = -20\log_{10}\frac{R(K,s)\|\mathbf{Y}\|_F}{NK}, \tag{33}$$

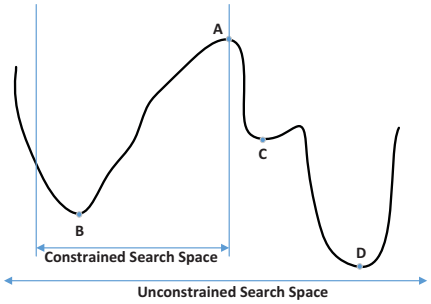


Fig. 3. Local stationary points in the constrained and unconstrained search spaces. The curve represents the function value. **A** is the starting point, **B** is the local stationary point in the constrained search space, **C** is the local stationary point in the unconstrained search space, **D** is the global minimum point in the unconstrained search space.

  and

$$\mathrm{RMSE} = \frac{R(K,s)\|s\mathbf{Y}\|_F}{\sqrt{NK}}. \tag{34}$$

- Convergence time $T(K,s)$: Based on ROV sequence r_i's and their corresponding execution time t_i's during iteration, we define $T(K,s) = t_{i^*+1}$, where

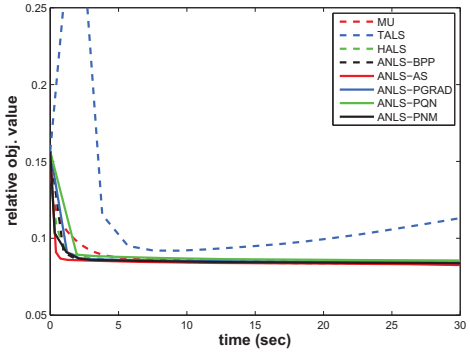
$$i^* = \max\{i | r_i > 1.05 \min_j\{r_j\}\}. \tag{35}$$

  In the definition, $\min_j\{r_j\}$ nearly equals to the ROV at the stationary point that an algorithm converges to. $0.05\min_j\{r_j\}$ is set to be the radius of the convergence region. We can guarantee that once an algorithm has been executed for $T(K,s)$ seconds, the difference between the achieved SNR (when the algorithm once stopped) and its optimal SNR would not exceed $20\log_{10}1.05 \approx 0.4$ in dB. That's why we define $T(K,s)$ as aforementioned.

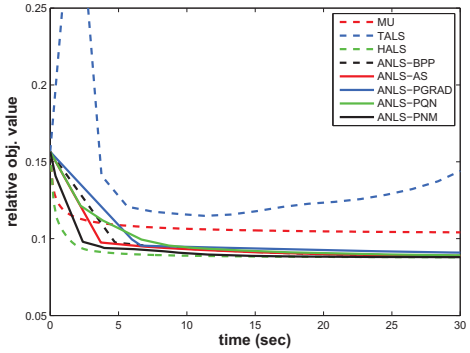
When comparing two algorithms, we suppose the visual quality is more important than the execution time used. Hence, we first compare their final ROVs, the one with smaller final ROV is better since it can achieve better reconstructed frames. If their ROVs are comparable, we will then compare their convergence time, and the one with smaller convergence time is considered better. In our experiments, the time spent to compute the objective value is excluded from the execution time.

B. Impact of Additional Constraints in NMF

After adding additional constraints to the ordinary NMF, one should expect that the convergence of certain algorithm will be slowed down; because each iteration step has more restriction. On the other hand, the relative objective value should increase in principle, because the search space is reduced. However, sometimes this may be violated since the NMF algorithms can only find local stationary point close to the starting point. An illustrative 1-Dimensional example is shown in Fig 3. In this case, the starting point **A** is at the boundary of the constrained search space, an optimization algorithm finding local stationary point close to **A**, may find point **B** and **C** as the local stationary points with and without constraints. We can observe the function value at point **B** is actually smaller than



(a) Without constraint $X, W \leq 1$



(b) With constraint $X, W \leq 1$

Fig. 4. Relative objective value vs. execution time of all algorithms with and without additional constraint using data set 'Akko' ($K = 20$, $M = 15$, $s = 2$).

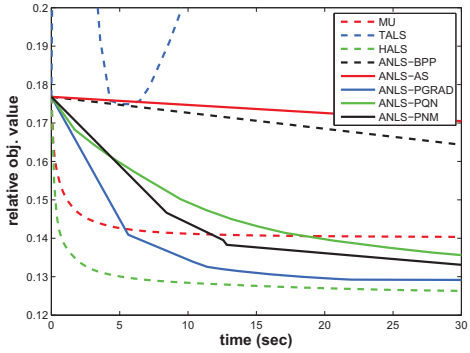


Fig. 5. Relative objective value vs. execution time of all algorithms with additional constraint under data set 'Lake' ($K = 40$, $M = 20$, $s = 3$).

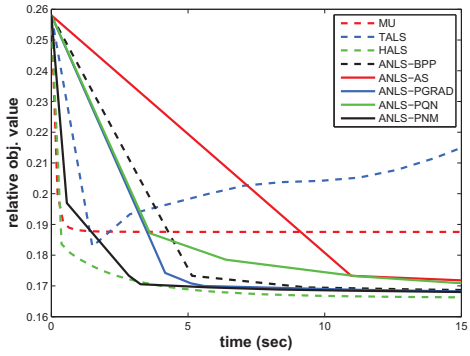


Fig. 6. Relative objective value vs. execution time of all algorithms with additional constraint under data set 'Dino' ($K = 40$, $M = 20$, $s = 1$).

the one at point **C**, though the search space is constrained. If the optimization algorithm can find the global minimum points, then the function value is guaranteed to increase with the constraints, see the function value at point **B** and **D**.

In our experiments, we observe the above trend. To further illustrate the slow-down of the convergence and the increase of the relative error, Fig. 4 and Table I give a pairwise comparison. The specific parameters are: $K = 20$, $M = 15$ and $s = 2$. $\mathbf{Y}$ contains the first frames from camera 1 to camera 20 of data set 'Akko' so we have $N = 640 \times 480$. Among all the 8 algorithms, TALS is the particular one whose relative error curve fluctuates and shows difficulty in convergence. The same phenomenon was also observed in [24], [25]. Therefore, we didn't list its converge time and relative objective value difference in Table I. Besides, the performance of the remaining seven algorithms are quite close under the ordinary NMF case (see Fig. 4 (a)). However, the impact of the modification can be significantly different (see Fig. 4 (b)). Firstly, using the MU algorithm, the relative error increases the most. The difference is bolded in Table I. As for the convergence time, ANLS-BPP, ANLS-AS, ANLS-PGRAD and ANLS-PQN increase the most in this experiment. The convergence time for HALS remains almost the same and achieves the minimum relative error under the additional constraints. The reason for this phenomenon may be the following: the modification on HALS is a simple thresholding for each column of $\mathbf{X}$, while for algorithms belonging to the ANLS category, the complicated modification leads to their inferior performance (e.g., in ANLS-AS and ANLS-BPP, the modification doubles the size of the LCP problem (14) to be (18)).

We select HALS as the one used for real TPVM applications, since it achieves the minimum relative error within the shortest execution time in most cases. Fig. 5 and Fig. 6 verify the above conjecture under different data sets and parameter settings of TPVM. An interesting finding from our experiments is that, as s increases, the strength of HALS becomes more evident, see Fig. 5 of $s = 3$. Note that HALS updates each column of $\mathbf{X}$ successively, which is very straightforward and efficient. While the ANLS based algorithms optimize the columns of $\mathbf{X}$ simultaneously, aiming to exploit the structure of $s\mathbf{Y}$'s columns reflected by weight coefficients in $\mathbf{W}$. When $s = 1$, matrix $\mathbf{W}$ is very sparse, i.e., the target images only need to be factorized by one or two atom frames, which makes the ANLS based algorithms quite similar to HALS. As s increases, $\mathbf{W}$ becomes much denser since one target image $s\mathbf{Y}_{:j}$ needs about s atom frames with upper bound $\mathbf{X}_{:i} \leq 1$ to match its intensity. The denser $\mathbf{W}$ is, the more time consuming the ANLS based algorithms are (e.g., solving (18) in ANLS-BPP and ANLS-AS, or inverse of matrix derived from Hessian matrix $\mathbf{W}\mathbf{W}^T$ in ANLS-PQN and ANLS-PNM).

C. Choice of Algorithmic Parameters in TPVM

In the TPVM, one needs to decide K and s based on four major performance measures:

TABLE I
QUANTITATIVE COMPARISON RESULTS OF ALL ALGORITHMS WITH (ORIGINAL) AND WITHOUT (MODIFIED) ADDITIONAL CONSTRAINT IN FIG. 4.

Algorithm	Relative objective value (%)			Convergence time (sec)		
	Original	Modified	Difference	Original	Modified	Difference
MU	8.41	10.41	**2.00**	4.4	4.6	0.2
TALS	11.3	14.51	-	-	-	-
HALS	8.32	8.79	0.47	3.1	2.9	-0.2
ANLS-BPP	8.37	8.86	0.49	1.8	14.0	**12.2**
ANLS-AS	8.26	8.86	0.60	1.2	13.4	**12.2**
ANLS-PGRAD	8.43	9.04	0.61	2.1	14.0	**11.9**
ANLS-PQN	8.56	8.90	0.34	2.0	13.8	**11.8**
ANLS-PNM	8.38	8.80	0.42	2.1	8.4	6.3

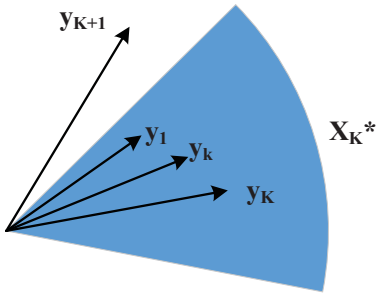


Fig. 7. Illustrative example for explaining (36).

1) The relative objective error $R(K,s)$: It reflects the visual quality of the views displayed by TPVM. The smaller $R(K,s)$ is, the better. One would like to know how $R(K,s)$ varies under different K and s. Note that the value of $R(K,s)$ here is the one derived when an NMF algorithm has converged.
2) The convergence time $T(K,s)$: We hope the display system can show multiviews under high frame rate f and f is upper bounded by $1/T(K,s)$. The smaller $T(K,s)$ is, the better. One would like to know how $T(K,s)$ varies under different K and s.
3) The supported viewer number K: The larger K is, the better.
4) The display power $P(s)$ (defined in Appendix B): The smaller $P(s)$ is, the better. In Appendix B, we derive that $P(s) \propto M/s$.

We assume M is unadjustable for a given display device. As shown in Section I, $M = t_c f_d$. In general, the flicker fusion frequency $f_c = 1/t_c$ is around 60Hz, at which rate most viewers can't resolve the temporally rapid changing optical signals. On the other side, modern optoelectronic displays can operate at refresh rates f_d that are 120Hz, 240Hz and beyond. In [34], Kulik *et al.* proposed a 360Hz display system to generate six stereoscopic views based on time multiplexing and polarization. The above corresponds to $M = 12$ case for the 2-D display and we fix this M in our experiments.

We first consider how $R(K,s)$ would be affected by K and s,

- When analyzing the effect of K, we assume $\mathbf{Y}_{K+1} = (\mathbf{Y}_K \quad \mathbf{y}_{K+1})$, which means a new frame $\mathbf{y}_{K+1}$ is added into the target image matrix $\mathbf{Y}_K$, when K increases to $K+1$. Suppose $(\mathbf{X}_K^*, \mathbf{W}_K^*)$ is the stationary point an NMF algorithm converges to, so that $R(K,s) = \|s\mathbf{Y}_K - \mathbf{X}_K^*\mathbf{W}_K^*\|_F / \|s\mathbf{Y}_K\|_F$. If

$$\min_{0\le \mathbf{w}_{K+1}\le 1} \frac{\|s\mathbf{y}_{K+1} - \mathbf{X}_K^*\mathbf{w}_{K+1}\|_2}{\|s\mathbf{y}_{K+1}\|_2} \gg R(K,s), \quad (36)$$

  then one would expect $R(K+1,s) > R(K,s)$. An illustrative example is shown in Fig. 7, where $\mathbf{Y}_K$ lies in the cone generated by $\mathbf{X}_K^*$ (i.e., $R(K,s) = 0$), while $\mathbf{y}_{K+1}$ is far away. When a frame $\mathbf{y}_{K+1}$ contains objects outside the previous frames $\mathbf{Y}_K$, (36) is likely to be satisfied, since these objects lead to large residue in $s\mathbf{y}_{K+1} - \mathbf{X}_K^*\mathbf{w}_{K+1}$, which decreases the average reconstruction quality.
- For increasing value of s, one should also expect $R(K,s)$ to increase. The reason is that the $R(K,s)$ of problem

$$\min_{\mathbf{X},\mathbf{W}} \|s\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F, \quad (37)$$
$$\text{subject to } 0 \le \mathbf{X} \le 1, 0 \le \mathbf{W} \le 1$$

  is the same as $R(K,s)$ of the following problem

$$\min_{\mathbf{X},\mathbf{W}} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F, \quad (38)$$
$$\text{subject to } 0 \le \mathbf{X} \le 1/s, 0 \le \mathbf{W} \le 1.$$

  The above equivalence is provable by doing a change of variable to $\mathbf{X}$. The latter has smaller feasible space, hence it should lead to larger minimal value, which consequently leads to a larger $R(K,s)$.

As for $T(K,s)$, K and s affects not only the factorization process, but also the initial points. Therefore it is much harder to predict how the change of K and s will impact the convergence speed. We can only observe it through numerical experiments.

In Table II, by using the HALS algorithm on data set 'Sea', we list $R(K,s)$ and $T(K,s)$ under a range of K's and s's. It can be seen that $R(K,s)$ increases when either K or s increases. $T(K,s)$ doesn't have a consistent trend relating to changes in K or s. In most cases, it varies between a narrow interval $[1,2]$. When the same experiment is done on other data sets, we observe the same patterns on $R(K,s)$ and $T(K,s)$. Compare among all the data sets, $R(K,s)$ of 'Temple' and 'Dino' are larger than others. The views captured on these two data sets are on different positions from different angles, hence the successive frames are not so similar comparing with the other data sets. The particular value of $R(K,s)$ may vary

TABLE II
$R(K,s)$ AND $T(K,s)$ UNDER VARYING K'S AND s'S USING DATA SET 'SEA' ($M = 12$). WHEN $K = M = 12$ AND $s = 1$, THE INITIALED VALUE OF $(\mathbf{X}, \mathbf{W}) = (\mathbf{Y}, I_M)$ IS ALREADY THE OPTIMAL, WHICH MAKES BOTH $R(K,s)$ AND $T(K,s)$ TO BE 0.0.

$s\backslash K$	12	14	16	18	20	22
1	0.0	8.0	10.6	12.3	13.6	14.7
2	5.2	9.1	11.4	13.0	14.2	15.2
3	7.8	10.9	13.0	14.5	15.6	16.6
4	9.8	12.5	14.2	15.6	16.6	17.5
5	11.7	14.2	15.6	16.8	17.7	18.5

(a) $R(K,s)$ (%)

$s\backslash K$	12	14	16	18	20	22
1	0.0	1.8	1.1	1.2	1.4	1.1
2	3.4	1.7	1.2	0.9	0.9	0.9
3	8.2	4.0	2.0	1.3	1.4	1.3
4	12.1	2.5	2.4	1.9	1.9	2.2
5	3.9	1.9	1.9	1.5	2.0	1.8

(b) $T(K,s)$ (sec)

under different data sets, however, the trend on $R(K,s)$ is consistent. Thus, we can conclude that the choice of K mainly relates to the tradeoff between $R(K,s)$ and K itself. Larger K support more viewers, however leads to worse visual quality. The choice of s mainly relates to the tradeoff between $R(K,s)$ and $P(s)$. Smaller s leads to better visual quality while spend more energy. In practice, the decision for choosing K and s highly relies on $R(K,s)$, however, it is impossible to do the factorization under all the combination of K's and s's. A realistic way is to sample $R(K,s)$ at some points and estimate the whole surface using certain interpolation methods. We shall concern this problem in the continuing work.

In the above experiments, the initial points are independently derived on each frame. While in video sequences, the factorization result of last frame can be used as the initial point of the current frame. It can help to decrease the convergence time dramatically. As shown in Fig. 8, we compare two strategies using sequence 'Book': one uses the factorization results of the last frame as the starting point for the current frame, the other reinitializes the starting point using the time multiplexing like method described in Section IV-A. In both strategies, we constrained the execution time for factorizing views from $K = 16$ cameras with $s = 2$ to be 1 second. The total frame number is 100. The two curves illustrate how the relative objective value changes under different frame index respectively. One can see the strength of the new initialization strategy from its smaller relative objective errors. This experiment shows that using the previous factorization results lead to a better starting point for multiview video sequences.

V. CONCLUSION AND FUTURE WORK

In this paper, we implemented the Temporal PsychoVisual Modulation system by solving an NMF problem with upper bound constraints on the variables. We first adapted all the state-of-the-art NMF algorithms to meet the additional constraints, following their original ideas. Then in the numerical experiments, we compared these algorithms based on their

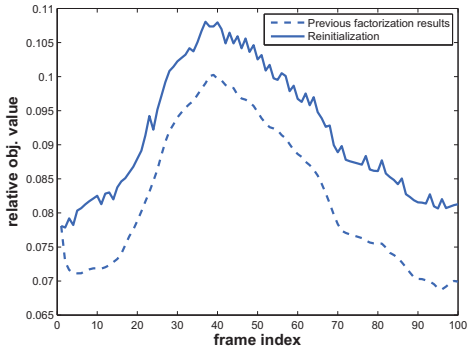


Fig. 8. Comparison between initialization using previous factorization results and reinitializing on data set 'Book' ($K = 16$, $M = 12$, $s = 2$, execution time for each frame is 1s).

factorization errors and convergence rates. We selected HALS for TPVM, because it is generally the fastest to yield the smallest factorization error. TALS doesn't converge in most cases with large factorization error, and the performance of the ANLS based algorithms can be largely affected when the constraints are added. Finally, we studied how the four performance measurements (the visual quality, the processing time, the number of supported viewers and the power cost) of TPVM would be affected by the parameters K and s, and proposed a possible way to choose the proper values of K and s. Our work is a foundation that will facilitate different applications of TPVM.

The TPVM implemented in this paper is just the first step, which leaves much space to be improved. Our future work includes:

- Accelerating the constrained NMF algorithms: As shown in [30], the algorithms for ordinary NMF can be further accelerated by applying different iterating strategies and stoping rules during the optimization. These techniques can be further investigated in the TPVM case.
- Utilizing the property of images: We haven't exploited the sparsity of natural images in this paper, e.g., the coefficients of images under wavelet transform are sparse. When a proper transform is applied, the factorization process may be speed up dramatically (e.g., embedding existing NMF algorithms into the framework of multilevel methods in [35]). From the other side, the image transform may be seen as a prefiltering step that makes the reconstructed frames more suitable for watching.
- Using other divergence measures to evaluate the performance of TPVM: In this paper, the Frobenius norm is used to measure the approximation divergence. There are still lots of other possible measures, such as the Kullback-Leibler divergence [18] and the Structural Similarity (SSIM) divergence [36]. Sometimes, we may require the optimization of the worst reconstructed view, or assign different importance for different views. These lead to other objective functions.
- Utilizing the normal view in TPVM: In [1] and [37], the authors show that the normal view $\mathbf{y}_0 = \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_M$ in TPVM can also be useful, which can lead

to exciting applications in 2D/3D display, information security and aiding. The optimization problem is similar to (1), while a new target image $\mathbf{y}_0$ is added, and the weight vector is fixed to be all 1's. Thus some more modifications need to be done on the existing NMF algorithms.

APPENDIX A
PROOF FOR THEOREM 1

Firstly, we prove that $0 \leq \mathbf{X}' \leq 1$. It is easy to see that $s\mathbf{Y}\mathbf{W}^T \geq 0$, $\mathbf{X}\mathbf{W}\mathbf{W}^T \geq 0$ and

$$\eta = \min\left\{\frac{[\mathbf{X}]}{[\mathbf{X}\mathbf{W}\mathbf{W}^T]}, \frac{[\mathbf{1}_N\mathbf{1}_M^T - \mathbf{X}]}{[s\mathbf{Y}\mathbf{W}^T]}\right\} \geq 0,$$

thus

$$\begin{aligned}\mathbf{X}'_{ij} &= \mathbf{X}_{ij} + \eta_{ij}(s\mathbf{Y}\mathbf{W}^T - \mathbf{X}\mathbf{W}\mathbf{W}^T)_{ij} \\ &\geq \mathbf{X}_{ij} - \eta_{ij}(\mathbf{X}\mathbf{W}\mathbf{W}^T)_{ij} \\ &\geq \mathbf{X}_{ij} - \mathbf{X}_{ij} = 0\end{aligned}$$

and

$$\begin{aligned}\mathbf{X}'_{ij} &= \mathbf{X}_{ij} + \eta_{ij}(s\mathbf{Y}\mathbf{W}^T - \mathbf{X}\mathbf{W}\mathbf{W}^T)_{ij} \\ &\leq \mathbf{X}_{ij} + \eta_{ij}(s\mathbf{Y}\mathbf{W}^T)_{ij} \\ &\leq \mathbf{X}_{ij} + (\mathbf{1}_N\mathbf{1}_M^T - \mathbf{X})_{ij} = 1\end{aligned}$$

To prove the non-increasingness property, we use Definition 1, Lemma 1 and Lemma 2 from [18].

Definition 1. *$G(\mathbf{x}, \mathbf{x}')$ is an auxiliary function for $F(\mathbf{x})$ if the conditions*

$$G(\mathbf{x}, \mathbf{x}') \geq F(\mathbf{x}), \quad G(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}) \tag{39}$$

are satisfied.

Lemma 1. *If G is an auxiliary function, then F is non-increasing under the update*

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}^t). \tag{40}$$

Proof. $F(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \leq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t)$. □

Lemma 2. *If $K(\mathbf{x})$ is the diagonal matrix*

$$K_{ab}(\mathbf{x}^t) = \delta_{ab}(\mathbf{W}\mathbf{W}^T\mathbf{x}^t)_a/(\mathbf{x}^t)_a \tag{41}$$

then

$$\begin{aligned}G(\mathbf{x}, \mathbf{x}^t) = F(\mathbf{x}^t) &+ \nabla F(\mathbf{x}^t)^T(\mathbf{x} - \mathbf{x}^t) \\ &+ \frac{1}{2}(\mathbf{x} - \mathbf{x}^t)^T K(\mathbf{x}^t)(\mathbf{x} - \mathbf{x}^t)\end{aligned} \tag{42}$$

is an auxiliary function for

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{v} - \mathbf{W}^T\mathbf{x}\|_2^2 \tag{43}$$

where $\mathbf{v}$ is an arbitrary vector.

Proof. See reference [18]. □

Here we propose a new Lemma based on Lemma 2,

Lemma 3. *If $K'(\mathbf{x})$ is the diagonal matrix*

$$K'_{ab}(\mathbf{x}^t) = \delta_{ab}\max\left\{(\mathbf{W}\mathbf{W}^T\mathbf{x}^t)_a/(\mathbf{x}^t)_a, (\mathbf{W}\mathbf{v})_a/(\mathbf{1}_M - \mathbf{x})_a\right\} \tag{44}$$

then

$$\begin{aligned}G'(\mathbf{x}, \mathbf{x}^t) = F(\mathbf{x}^t) &+ \nabla F(\mathbf{x}^t)^T(\mathbf{x} - \mathbf{x}^t) \\ &+ \frac{1}{2}(\mathbf{x} - \mathbf{x}^t)^T K'(\mathbf{x}^t)(\mathbf{x} - \mathbf{x}^t)\end{aligned} \tag{45}$$

is an auxiliary function for

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{v} - \mathbf{W}^T\mathbf{x}\|_2^2 \tag{46}$$

Proof.

$$\begin{aligned}G'(\mathbf{x}, \mathbf{x}^t) = G(\mathbf{x}, \mathbf{x}^t) &\\ &+ \frac{1}{2}(\mathbf{x} - \mathbf{x}^t)^T(K'(\mathbf{x}^t) - K(\mathbf{x}^t))(\mathbf{x} - \mathbf{x}^t),\end{aligned} \tag{47}$$

where $G(\mathbf{x}, \mathbf{x}^t) \geq F(\mathbf{x})$ is the one used in Lemma 2. From definition, it is easy to see $K'(\mathbf{x}^t) - K(\mathbf{x}^t)$ is semi-positive definite, thus

$$G'(\mathbf{x}, \mathbf{x}^t) \geq G(\mathbf{x}, \mathbf{x}^t) \geq F(\mathbf{x}). \tag{48}$$

□

Using Lemma 1 and Lemma 3, F is non-increasing under the update

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} G'(\mathbf{x}, \mathbf{x}^t) = \mathbf{x}^t - K'(\mathbf{x}^t)^{-1}\nabla F(\mathbf{x}^t). \tag{49}$$

For $1 \leq i \leq N$, replacing $\mathbf{x}^{t+1}$ by $(\mathbf{X}'_{i:})^T$, $\mathbf{x}^t$ by $(\mathbf{X}_{i:})^T$ and $\mathbf{v}$ by $s(\mathbf{Y}_{i:})^T$, we have

$$F_i(\mathbf{x}) = \frac{1}{2}\|s\mathbf{Y}_{i:} - \mathbf{x}^T\mathbf{W}\|_2^2 \tag{50}$$

is non-increasing under update

$$\mathbf{X}'_{i:} = \mathbf{X}_{i:} + \eta_{i:} \circ (s\mathbf{Y}\mathbf{W}^T - \mathbf{X}\mathbf{W}\mathbf{W}^T)_{i:}. \tag{51}$$

APPENDIX B
THE RELATION BETWEEN POWER $P(s)$ AND s

In the traditional time multiplexing, the light influx of each rendered view decreases linearly in M [1]. To ensure the adequate intensity for viewing, one need to provide more energy to enlarge the influx of the atom frames. However, TPVM provides the other way by adjusting the scaling factor s.

We define the power $P(s) = NS_pI_a$, where S_p is the area of a pixel when displayed, and I_a (the unit is W/m^2) is the maximum intensity of the atom frames, i.e., the optoelectronic system emits a light of intensity $I_a\mathbf{X}_{ij}$ at the position of pixel (i, j). Since each atom frame only last for $1/f_d$ seconds, the averaged maximum intensity of the atom frames during $1/f_c$ is only I_a/M. Hence, the observed frames by human eyes at frequency f_c would be

$$\frac{I_a\mathbf{X}\mathbf{W}}{f_d} \cdot f_c = I_a\mathbf{X}\mathbf{W}/M \approx sI_a\mathbf{Y}/M \triangleq I_o\mathbf{Y},$$

where $I_o = sI_a/M$ is the equivalent maximum intensity of the observed frames. Suppose $I_o = I_T$, where I_T is the adequate intensity for HVS, then we have

$$P(s) = NS_pI_a = NS_pI_T\left(\frac{M}{s}\right) \propto \frac{M}{s}.$$

REFERENCES

[1] X. Wu and G. Zhai, "Temporal psychovisual modulation: A new paradigm of information display," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 136–141, January 2013.

[2] L. Komitov, G. Hegde, and D. Kolev, "Fast liquid crystal light shutter," *J. Phys. D: Appl. Phys.*, vol. 44, pp. 1–5, Oct. 2011.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[4] X. Guo, Y. Lu, F. Wu, and W. Gao, "Inter-view direct mode for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1527–1532, December 2006.

[5] J. H. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New coding tools for illumination and focus mismatch compensation in multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1519–1535, November 2007.

[6] J. Lu, H. Cai, J. Lou, and J. Li, "An epipolar geometry-based fast disparity estimation algorithm for multiview image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 737–750, June 2007.

[7] R. Raskar, G. Welch, and H. Fuchs, "Spatially augmented reality," in *Proc. 1st Int. Workshop Augmented Reality*, 1998, pp. 11–20.

[8] D. Bertsekas, "Projected newton methods for optimization problems with simple constraints," *SIAM Journal on Control and Optimization*, vol. 20, no. 2, pp. 221–246, 1982.

[9] C. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[10] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, "Mahnmf: Manhattan non-negative matrix factorization," *http://arxiv.org/abs/1207.3438*, 2012.

[11] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and its Applications*, vol. 401, no. 1, pp. 29–47, 2006.

[12] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *the 26th annual int. ACM SIGIR conference on Research and development in information retrieval*, 2003, pp. 267–273.

[13] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *SIAM Int. Conf. Data Mining*, 2005, pp. 606–610.

[14] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[15] A. Cichocki, S.-I. Amari, R. Zdunek, and A. H. Phan, *Non-negative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley-Blackwell, 2009.

[16] C. Fevotte, N. Bertin, and J. L. Durrieu, "Nonnegative matrix factorization with the itakurasaito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[17] S. A. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 20, pp. 1364–1377, 2009.

[18] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," in *Advances Neural Inform. Process. Syst*, vol. 13, Nov. 2001, pp. 556–562.

[19] E. Gonzalez and Y. Zhang, "Accelerating the Lee-Seung algorithm for nonnegative matrix factorization," Rice Univ., Houston, TX, USA, Tech. Rep. TR-05-02, 2005.

[20] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Comput. Statist. Data Anal.*, vol. 52, no. 1, pp. 155–173, 2007.

[21] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," in *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, 2009, pp. 708–721.

[22] A. Cichocki, R. Zdunek, and S.-I. Amari, *Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization*, ser. Lecture Notes in Computer Science. Springer, 2007, vol. 4666, pp. 169–176.

[23] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: a new algorithm and comparisons," in *Proceedings of the 8th IEEE International Conference on Data Mining*, 2008, pp. 353–362.

[24] ——, "Fast nonnegative matrix factorization: an active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[25] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 713–730, 2008.

[26] D. Kim, S. Sra, and I. S. Dhillon, "Fast Newton-type methods for the least squares nonnegative matrix approximation problem," in *Proceedings of SIAM Conference on Data Mining*, 2007, pp. 343–354.

[27] ——, "Fast projection-based methods for the least squares nonnegative matrix approximation problem," *Statistical Analysis and Data Mining*, vol. 1, no. 1, pp. 38–51, 2008.

[28] P. Gong and C. Zhang, "Efficient nonnegative matrix factorization via projected Newton method," *Pattern Recognition*, vol. 45, pp. 3557–3565, March 2012.

[29] N. Gillis, "Sparse and unique nonnegative matrix factorization through data preprocessing," *Journal of Machine Learning Research*, vol. 13, pp. 3349–3386, November 2012.

[30] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization," *Neural Computation*, vol. 24, no. 4, pp. 1085–1105, 2012.

[31] ——, "Nonnegative factorization and the maximum edge biclique problem," *CORE Discussion paper*, 2008/64.

[32] http://en.wikipedia.org/wiki/Linear_complementarity_problem.

[33] D. Kim, S. Sra, and I. S. Dhillon, "A new projected quasi-newton approach for the non-negative least squares problem," The Univ. of Texas, Austin, TX, USA, Tech. Rep. TR-06-54, 2006.

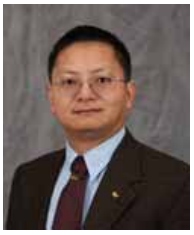
[34] A. Kulik, A. Kunert, S. Beck, R. Reichel, R. Blach, A. Zink, and B. Froehlich, "C1×6: a stereoscopic six-user display for co-located collaboration in shared virtual environments," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 188.1–188.12, December 2011.

[35] N. Gillis and F. Glineur, "A multilevel approach for non negative matrix factorization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 7, pp. 1708–1723, 2012.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[37] X. Wu and G. Zhai, "Backward compatible stereoscopic displays via temporal psychovisual modulation," in *SIGGRAPH Asia 2012 Emerging Technologies*, no. 4.

Jianzhou Feng received the B.S. degree in Electronic Engineering from Shanghai Jiao Tong University, China, in 2007. He is currently pursuing the Ph.D. degree in Electronic Engineering with Shanghai Jiao Tong University. His current research interests include sparse representation, image restoration, and nonnegative matrix factorization.



Xiaoming Huo (S'96-M'99-SM'04) received a B.S. degree in mathematics from the University of Science and Technology of China in 1993 and an M.S. degree in electrical engineering and a Ph.D. degree in statistics from Stanford University in 1997 and 1999, respectively. He is a full professor in the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta. His research interests include statistical theory and computational methodology. He has made numerous contributions on topics such as sparse representation, image coding and representation, wavelets, and statistical problems in detectability. Professor Huo represented China in the 30th International Mathematical Olympiad (IMO), which was held in Germany in 1989, and received a golden prize. Professor Huo was a Fellow of IPAM in September 2004. He received the Georgia Tech Sigma Xi Young Faculty Award in 2005. His work has led to an interview by Emerging Research Fronts in June 2006 in the field of Mathematicsin every two months, one paper is selected.

Li Song (M'08) received the B.Eng and the M.S degree from Nanjin University of Science and Technology, and the Ph.D. degree from Shanghai Jiao Tong University (SJTU) in 1997, 2000 and 2005, respectively. Then he joined SJTU and became an associate professor of department of electronic engineering. He was also a visiting professor in Santa Clara University from 2011.9 to 2012.9. He has more than 100 publications in the fields of video coding, image processing and patter recognition, 18 issued patents and several MPEG/JVT/JCTVC international standard contributions. He has served as an Associate Editor for the journal of Springer Multidimensional Systems and Signal Processing (MSSP) and a publicity chair of the 2013 IEEE International Conference on Multimedia and Expo (ICME), and a session chair of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) in 2010. He is TC members of IEEE CAS Visual Signal Processing and Communications (VSPC) and IEEE Communication Society Multimedia Communications Technical Committee (MMTC), and technical program committee for many international conferences. He is a recipient of the Okawa Foundation Research Grant in 2012, the Best Reviewer Award from International Conference on Visual Communication & Image Processing (VCIP) in 2012 and the Best Paper Award from the International Conference on Wireless Communications & Signal Processing (WCSP) in 2010.

Wenjun Zhang (M'02-SM'10-F'11) received the B.S., M.S. and Ph.D. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1984, 1987 and 1989, respectively. From 1990 to 1993, He worked as a post-doctoral fellow at Philips Kommunikation Industrie AG in Nuremberg, Germany, where he was actively involved in developing HD-MAC system. He joined the Faculty of Shanghai Jiao Tong University in 1993 and became a full professor of Electronic Engineering in 1995. As the project leader, he successfully developed the first Chinese HDTV prototype system in 1998. He was one of the main contributors to the Chinese Digital Television Terrestrial Broadcasting Standard issued in 2006. He holds more than 40 patents and published more than 90 papers in international journals and conferences. Prof. Zhang's main research interests include digital video coding and transmission, multimedia semantic processing and intelligent video surveillance. He is the vice president for Research of Shanghai Jiao Tong University and the Chief Scientist of the Chinese Digital TV Engineering Research Centre, an industry/government consortium in DTV technology research and standardization.

Xiaokang Yang (M'00-SM'04) received the B. S. degree from Xiamen University, Xiamen, China, in 1994, the M.S. degree from the Chinese Academy of Sciences, Shanghai, China, in 1997, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, in 2000. He is currently a Full Professor and Deputy Director of the Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiao Tong University. From September 2000 to March 2002, he was a Research Fellow in Centre for Signal Processing, Nanyang Technological University, Singapore. From April 2002 to October 2004, he was a Research Scientist with the Institute for Infocomm Research, Singapore. He has published over 80 refereed papers, and has led six patents. His current research interests include video processing and communication, media analysis and retrieval, perceptual visual processing, and pattern recognition. He actively participates in the International Standards such as MPEG-4, JVT, and MPEG-21. Dr. Yang received the Microsoft Young Professorship Award 2006, the Best Young Investigator Paper Award at IS&T/SPIE International Conference on Video Communication and Image Processing (VCIP2003), and awards from A-STAR and Tan Kah Kee foundations. He is a member of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society. He was the Special Session Chair of Perceptual Visual Processing of IEEE ICME2006. He is the local cochair of ChinaCom2007 and the technical program co-chair of IEEE SiPS2007.