Robust Matrix Factorization with Unknown Noise

Deyu Meng
Xi'an Jiaotong University
dymeng@mail.xjtu.edu.cn

Fernando De la Torre
Carnegie Mellon University
ftorre@cs.cmu.edu

Abstract

Many problems in computer vision can be posed as recovering a low-dimensional subspace from high-dimensional visual data. Factorization approaches to low-rank subspace estimation minimize a loss function between an observed measurement matrix and a bilinear factorization. Most popular loss functions include the L_1 and L_2 losses. L_2 is optimal for Gaussian noise, while L_1 is for Laplacian distributed noise. However, real data is often corrupted by an unknown noise distribution, which is unlikely to be purely Gaussian or Laplacian. To address this problem, this paper proposes a low-rank matrix factorization problem with a Mixture of Gaussians (MoG) noise model. The MoG model is a universal approximator for any continuous distribution, and hence is able to model a wider range of noise distributions. The parameters of the MoG model can be estimated with a maximum likelihood method, while the subspace is computed with standard approaches. We illustrate the benefits of our approach in extensive synthetic and real-world experiments including structure from motion, face modeling and background subtraction.

1. Introduction

Many computer vision, machine learning and statistical problems can be formulated as one of learning a low dimensional linear model. These linear models have been widely used in computer vision to solve problems such as structure from motion [39], face recognition [43], photometric stereo [19], object recognition [40], motion segmentation [41] and plane-based pose estimation [36].

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \Re^{d \times n}$ (see [1] for notation) be a matrix where each column $\mathbf{x}_i$ is a d-dimensional measurement. Standard approaches to subspace learning optimize

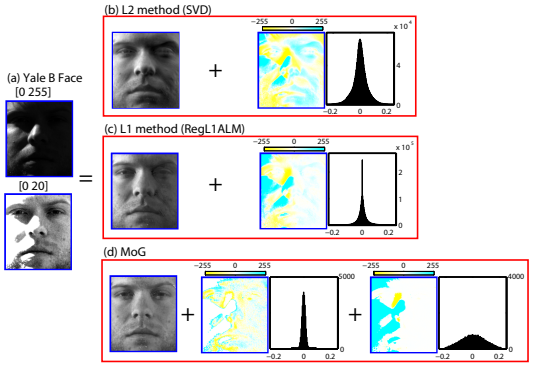

Figure 1. (a) Original face ($\mathbf{X}$). The upper image is the same as the lower one, with different display ranges. Observe the amplified camera noise in the dark region of the face. (b) The reconstructed image ($\mathbf{UV}^T$), the error image ($\mathbf{E} = \mathbf{X} - \mathbf{UV}^T$) and histogram of the error computed with the L_2 loss. (c) Same as (b) but with L_1 loss. (d) The reconstructed image and the two Gaussian errors, with smaller and larger variances, obtained by our method. (Figure better seen in color and to see details zoom on a computer screen.)

the Low Rank Matrix Factorization (LRMF) error:

$$\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{W} \odot (\mathbf{X} - \mathbf{UV}^T) \right\|_{L_p}, \tag{1}$$

where $\mathbf{U} \in \Re^{d \times r}$ and $\mathbf{V} \in \Re^{n \times r}$ are low-dimensional matrices ($r < \min(d, n)$), $\mathbf{W}$ is the indicator matrix of the same size as $\mathbf{X}$. $w_{ij} = 0$ if x_{ij} is missing and 1 otherwise. $\| \cdot \|_{L_p}$ denotes an L_p norm, and most popular approaches use L_2 and L_1 norm.

A main advantage of minimizing the L_2 norm is that the optimization problem is smooth and there are multiple fast numerical solvers [35, 31, 9, 44, 28, 32]. A closed-form solution for $\mathbf{U}$ and $\mathbf{V}$ can be computed with the Singular Value Decomposition (SVD) when all data is available (no missing data). However, the L_2 norm is only optimal for Gaussian noise and provides biased estimates in presence of outliers and non-Gaussian noise distributions. In order to introduce robustness to outliers, the L_2 can be replaced by robust functions [11] or the L_1 norm [18, 39, 22, 13, 20, 14, 43]. Unfortunately, these approaches do not have closed-form solution and lead to non-smooth optimization problems.

[1] Bold uppercase letters denote matrices, bold lowercase letters denote vectors, and non-bold letters represent scalar variables. $\mathbf{d}_i$ and $\mathbf{d}^i$ represent the i^{th} column and row vectors of the matrix $\mathbf{D}$, respectively, and d_{ij} denotes the scalar in the i^{th} row and j^{th} column of $\mathbf{D}$. $\odot$ denotes the Hadamard product (component-wise multiplication). L_p denotes the power p norm of a matrix, that is $\|\mathbf{D}\|_{L_p} = \sum_{i,j} |d_{ij}|^p$.

While L_2 or L_1 norms are only optimal if the noise follows a Gaussian or Laplacian distribution, this is not the case in most real problems. For instance, consider the case of face images of one subject taken under varying illumination conditions (e.g., Fig. 1(a) illustrates one image of the Yale B database [16]). Under the assumption that the face is a Lambertian surface, the faces under different point light sources can be recovered by a three dimensional subspace (e.g., [34]). If the diffusion or background illumination is considered in the model the subspace will be of dimension four [21]. However, in real images there are different types of noise sources. First, the face is not a perfect Lambertian surface, and there are cast shadows. Second, due to the camera range settings there might be pixels that are saturated and there exist specular reflections (especially in people with glasses). Third, the camera noise ("read noise") is amplified in the dark areas [30] (see Fig. 1(a)). These different types of noise can have different distributions, and minimizing either the L_2 or L_1 loss is unlikely to produce a good model to factorize the illumination component (see Fig. 1(b) and Fig. 1(c), respectively).

To address this issue, this paper proposes a simple but effective approach to LRMF with unknown noise. The key idea is to model the noise as a Mixture of Gaussians (MoG) [26], which is an universal approximator to any continuous density function [25]. Thus, it subsumes prior popular L_2 and L_1 models (the Laplace distribution can be equivalently expressed as a scaled MoG [2]). Fig. 1(d) illustrates how the proposed MoG noise model can better account for the different types of noise and provide a better estimate of the underlying face. The parameters of our proposed model, subspace-MoG, can be estimated with the traditional Expectation-Maximization (EM) under a Maximum Likelihood Estimation (MLE) framework. The effectiveness of our MoG method is shown in synthetic, Structure From Motion (SFM), face modeling and background subtraction experiments.

2. Previous work

The L_2-norm LRMF with missing data has been studied in the statistical literature since the early 80's. Gabriel and Zamir [15] proposed a weighted SVD technique that used alternated minimization (or criss-cross regression) to find the principal subspace of the data. De la Torre and Black [11] proposed Robust PCA by changing the L_2 norm to a robust function to deal with outliers. They used the Iteratively-Reweighted Least-Squares (IRLS) to solve the problem. This approach can handle missing data by setting weights to zero in the IRLS algorithm, but it is prone to local minima. Srebro and Jaakkola [35] proposed the Weighted Low-rank Approximation (WLRA) algorithm, that uses EM or conjugate gradient descent depending on the complexity of the structure of the problem. To avoid local minima, Buchanan and Fitzgibbon [6] added regularization terms to Eq. (1) and modified the Levenberg-Marquardt (LM) algorithm to estimate the variables ($\mathbf{U}$, $\mathbf{V}$) jointly. Chen [9] later proposed modifications of LM algorithms to improve its efficient by solving smaller linear system in every iteration. Okatani and Deguchi [31] showed that a Wiberg marginalization strategy on $\mathbf{U}$ or $\mathbf{V}$ provides a robust initialization, but its high memory requirements make it impractical for medium-size datasets. Aguiar et al. [1] introduced a globally optimal solution to L_2 LRMF with missing data under the assumption that the missing data has a special Young diagram structure. More recently, Zhao and Zhang [44] introduced the SALS method that constrains the components of $\mathbf{X}$ lie within a range, and considers the L_2 LRMF as a constrained model. Mitra et al. [28] showed that the matrix factorization problem can be formulated as a low-rank semidefinite program and proposed an augmented Lagrangian method. However, all of these works minimize an L_2 error that is only optimal for Gaussian noise.

In order to introduce robustness to outliers, Ke and Kanade [20] suggested replacing the L_2 loss with the L_1 norm, minimized by alternated linear or quadratic programming (ALP/AQP). A more efficient method called PCAL1 was then proposed by Kwak [22]. This method maximizes the L_1 norm of the projected data. Similarly to the L_2 Wiberg approach [31], Eriksson and Hengel [14] experimentally showed that the alternated convex programming approach frequently does not converge to the desired point, and introduced the L_1Wiberg approach to address this. Very recently, Zheng et al. [45] extended [14] by adding a nuclear norm regularizer on $\mathbf{V}$ and the orthogonality constraints in $\mathbf{U}$, which resulted in improvements on the structure from motion problem. In the compressed sensing literature, Wright et al. [43] proposed a Robust PCA method using recent advances in rank minimization. A major advantage of this approach lies in its convex formulation even in the case of sparse outliers and missing data. These methods, however, optimize an L_1 norm error and are thus only optimal for Laplacian noise.

Beyond these deterministic LRMF methods, there has been several probabilistic extensions of matrix factorizations. Factor analysis (FA) [4] is a probabilistic extension of PCA that assumes normally distributed coefficients ($\mathbf{U}$) and a diagonal Gaussian noise model. An instance of FA is the probabilistic Principal Component Analysis (P-PCA) [29, 33, 38] model. Unlike FA, PPCA assumes an isotropic Gaussian noise models. Other probabilistic extensions include the mixture of PPCA [37] that extends P-PCA by considering a mixture model in which the components are probabilistic PCA models (Mixture PCA). Recently, some probabilistic frameworks for robust matrix factorization [42, 23, 3] have been further proposed and model the noise with a Laplacian or student-t distributions. Un-

like previous work, we model our noise as a MoG and not a particular unimodal distribution.

3. LRMF with MoG noise

This section proposes a new LRMF method with a MoG noise model, a new matrix factorization method that accounts for multi-modal noise distributions.

3.1. The subspace-MoG model

In LRMF, each element x_{ij} $(i = 1, 2, \cdots, d, j = 1, 2, \cdots, n)$ of the input matrix $\mathbf{X}$ can be modeled as

$$x_{ij} = (\mathbf{u}^i)^T \mathbf{v}^j + \varepsilon_{ij}, \tag{2}$$

where $\mathbf{u}^i$ and $\mathbf{v}^i$ are the i^{th} row vectors of $\mathbf{U}$ and $\mathbf{V}$, respectively, and ε_{ij} denotes the noise in x_{ij}. It can be easily shown that the L_2 or L_1 LRMF model (1) corresponds to the MLE of the problem when ε_{ij} is independently sampled from a Gaussian or Laplace distribution, respectively. To deal with more complex problems in computer vision, it is natural to use a MoG to model the noise. Since it is an universal approximator to any continuous distributions [25]. For instance, a Laplacian distribution can be equivalently expressed as a scaled MoG [2].

Therefore, in this paper we will assume that each ε_{ij} in Eq. (2) is a sample from a MoG distribution $p(\varepsilon)$, defined as

$$p(\varepsilon) \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\varepsilon|0, \sigma_k^2),$$

where $\mathcal{N}(\varepsilon|0, \sigma^2)$ denotes the Gaussian distribution with mean 0 and variance σ^2. $\pi_k \geq 0$ is the mixing proportion where $\sum_{k=1}^{K} \pi_k = 1$. Then, the probability of each element x_{ij} of $\mathbf{X}$ can be written as

$$p(x_{ij}|\mathbf{u}^i, \mathbf{v}^j, \mathbf{\Pi}, \mathbf{\Sigma}) = \sum_{k=1}^{K} \pi_k p(x_{ij}|k),$$

where $p(x_{ij}|k) = \mathcal{N}(x_{ij}|(\mathbf{u}^i)^T \mathbf{v}^j, \sigma_k^2)$, $\mathbf{\Pi} = \{\pi_1, \pi_2, \cdots, \pi_K\}$, and $\mathbf{\Sigma} = \{\sigma_1, \sigma_2, \cdots, \sigma_K\}$. The likelihood of $\mathbf{X}$ can then be written as

$$\begin{aligned} p(\mathbf{X}|\mathbf{U}, \mathbf{V}, \mathbf{\Pi}, \mathbf{\Sigma}) &= \prod_{i,j \in \Omega} p(x_{ij}|(\mathbf{u}^i)^T \mathbf{v}^j, \mathbf{\Pi}, \mathbf{\Sigma}) \\ &= \prod_{i,j \in \Omega} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_{ij}|(\mathbf{u}^i)^T \mathbf{v}^j, \sigma_k^2), \end{aligned}$$

where Ω is the index set of the non-missing entries in $\mathbf{X}$.

Given the likelihood, our aim is to maximize the log-likelihood function w.r.t the MoG parameters $\mathbf{\Pi}, \mathbf{\Sigma}$ and the LRMF parameters $\mathbf{U}, \mathbf{V}$, that is:

$$\begin{aligned} &\max_{\mathbf{U}, \mathbf{V}, \mathbf{\Pi}, \mathbf{\Sigma}} \mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{\Pi}, \mathbf{\Sigma}) \\ &= \sum_{i,j \in \Omega} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_{ij}|(\mathbf{u}^i)^T \mathbf{v}^j, \sigma_k^2). \end{aligned} \tag{3}$$

In the following we will refer to the problem (3) as the subspace-MoG model.

3.2. EM algorithm

The EM [12] algorithm can be used to estimate the parameters $(\mathbf{U}, \mathbf{V}, \mathbf{\Pi}, \mathbf{\Sigma})$ that maximize the likelihood function of the subspace-MoG model. Recall that in the standard EM algorithm for MoG there is a mean for each cluster and in our case all clusters share the variables $\mathbf{U}, \mathbf{V}$. Our proposed algorithm will iterate between calculating responsibilities of all Gaussian components (E Step) and maximizing the parameters $\mathbf{\Pi}, \mathbf{\Sigma}$ and $\mathbf{U}, \mathbf{V}$ of the model (M Step).

E Step: Assume a latent variable z_{ijk} in the model, with $z_{ijk} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{ijk} = 1$, indicating the assignment of the noise ε_{ij} to a specific component of the mixture. The posterior responsibility of mixture k $(= 1, 2, \cdots, K)$ for generating the noise of x_{ij} $(i = 1, 2, \cdots, d,\ j = 1, 2, \cdots, n)$ is then calculated by ([12]):

$$E(z_{ijk}) = \gamma_{ijk} = \frac{\pi_k \mathcal{N}(x_{ij}|(\mathbf{u}^i)^T \mathbf{v}^j, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_{ij}|(\mathbf{u}^i)^T \mathbf{v}^j, \sigma_k^2)}. \tag{4}$$

The M step maximizes the upper bound given by the E-step w.r.t. $\mathbf{U}, \mathbf{V}, \mathbf{\Pi}, \mathbf{\Sigma}$ [12]:

$$E_{\mathbf{Z}}\, p(\mathbf{X}, \mathbf{Z}|\mathbf{U}, \mathbf{V}, \mathbf{\Pi}, \mathbf{\Sigma}) =$$

$$\sum_{i,j \in \Omega} \sum_{k=1}^{K} \gamma_{ijk} \left(\log \pi_k - \log \sqrt{2\pi} \sigma_k - \frac{(x_{ij} - (\mathbf{u}^i)^T \mathbf{v}^j)^2}{2\sigma_k^2} \right). \tag{5}$$

An easy way to solve this maximization problem is to alternatively update the MoG parameters $\mathbf{\Pi}, \mathbf{\Sigma}$ and the factorized matrices $\mathbf{U}, \mathbf{V}$ as follows:

Update $\mathbf{\Pi}, \mathbf{\Sigma}$: Closed-form updates for the MoG parameters (for $k = 1, 2, \cdots, K$) are [12]:

$$\begin{aligned} N_k &= \sum_{i,j} \gamma_{ijk}, \quad \pi_k = \tfrac{N_k}{N}, \\ \sigma_k^2 &= \tfrac{1}{N_k} \sum_{i,j} \gamma_{ijk} (x_{ij} - (\mathbf{u}^i)^T \mathbf{v}^j)^2. \end{aligned} \tag{6}$$

Update $\mathbf{U}, \mathbf{V}$: The components of Eq. (5) related to $\mathbf{U}$ and $\mathbf{V}$ can then be re-written as follows:

$$\begin{aligned} &\sum_{i,j \in \Omega} \sum_{k=1}^{K} \gamma_{ijk} \left(-\frac{(x_{ij} - (\mathbf{u}^i)^T \mathbf{v}^j)^2}{2\sigma_k^2} \right) \\ = &- \sum_{i,j \in \Omega} \left(\sum_{k=1}^{K} \frac{\gamma_{ijk}}{2\sigma_k^2} \right) (x_{ij} - (\mathbf{u}^i)^T \mathbf{v}^j)^2 \\ = &- \left\| \mathbf{W} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T) \right\|_{L_2}, \end{aligned} \tag{7}$$

where the element w_{ij} of $\mathbf{W} \in \Re^{d\times n}$ is

$$w_{ij} = \begin{cases} \sqrt{\sum_{k=1}^{K} \frac{\gamma_{ijk}}{2\sigma_k^2}}, & i,j \in \Omega \\ 0 \quad , & i,j \notin \Omega \end{cases}. \tag{8}$$

It is interesting to observe that the maximization of (7) is exactly equivalent to the Weighted L_2 LRMF problem. We can use any off-the-shelf algorithms, such as the Alternated Least Squares (ALS) [11], WLRA [35] and DN [6] to update $\mathbf{U}, \mathbf{V}$ in our method. We adopted the ALS due to its simplicity of implementation and good performance. The optimization process is summarized in Algorithm 1.

Algorithm 1: MoG algorithm for LRMF

Input: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n) \in R^{d\times n}$, index set Ω of non-missing entries of $\mathbf{X}$

Output: $\mathbf{U}, \mathbf{V}$

1 Randomly initialize $\mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{U}, \mathbf{V}$, MoG number K, small threshold ε.

2 **repeat**

3 (*E Step*): Evaluate γ_{ijk} for $i = 1, 2, \cdots, d$; $j = 1, 2, \cdots, n$; $k = 1, 2, \cdots, K$ by Eq. (4).

4 (*M Step for* $\mathbf{\Pi}, \mathbf{\Sigma}$): Evaluate π_k, σ_k^2 for $k = 1, 2, \cdots, K$ by Eq. (6).

5 (*M Step for* $\mathbf{U}, \mathbf{V}$): Evaluate $\mathbf{U}, \mathbf{V}$ by solving $\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{W} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T) \right\|_{L_2}$ through ALS, where $\mathbf{W}$ is calculated by Eq. (8).

6 (*Automatic K tuning*): If $\frac{|\sigma_i^2 - \sigma_j^2|}{\sigma_i^2 + \sigma_j^2} < \varepsilon$ for some i, j, then combine the i^{th} and j^{th} Gaussian components into a unique Gaussian by letting $\pi_i = \pi_i + \pi_j$, $\sigma_i^2 = (n_i\sigma_i^2 + n_j\sigma_j^2)/(n_i + n_j)$, where n_i is the element number in i^{th} Gaussian component, and removing the j^{th} Gaussian parameters from $\mathbf{\Pi}, \mathbf{\Sigma}$. Let $K = K - 1$.

7 **until** *converge*;

3.3. Other details of the subspace-MoG model

Number of Gaussian components: We propose a simple but effective method to automatically estimate the number of Gaussians in our model. We start with a given number of Gaussian mixtures (e.g., $K = 6$) that is large enough to fit the noise distribution in all our experiments. After each iteration (E and M step), we check if the relative deviation $\frac{|\sigma_i^2 - \sigma_j^2|}{\sigma_i^2 + \sigma_j^2}$ between variances of two Gaussian components is smaller than some small threshold ε ($\varepsilon = 0.1$ for all experiments). If so, the two mixtures are naturally seen as a similar Gaussian and can be combined. The number K is thus reduced to $K - 1$.

Local minima: Our algorithm is iterative in nature, and in each iteration we are guaranteed to not decrease the energy of the log-likelihood function (Eq. (3)). However, the log-likelihood is subject to local maxima ([12]). A commonly used strategy to alleviate this problem is to apply multiple random initializations, and select the one with the largest log-likelihood.

Termination conditions: We stop the algorithm when the change in $\mathbf{U}$ between consecutive iterations is smaller than a pre-specified small threshold, or the maximum number of iterations is reached.

Robustness to outliers: L_2 LRMF is generally considered to be sensitive to outliers. In our subspace-MoG model, however, an outlier will belong to a mixture with large variance, and w_{ij} will have a small value based on Eq. (8). This will reduce the influence of the outlier in the solution.

4. Experiments

To evaluate the performance of the proposed subspace-MoG method, we conducted extensive synthetic, Structure From Motion (SFM), face modeling and background subtraction experiments. In the synthetic and SFM experiments we analyzed the performance of our algorithm in situations when the ground truth is known and we added different types of noise to it. The experiments in face modeling and background subtraction illustrate how the the MoG noise model is a realistic assumption for visual data.

All methods were implemented in Matlab R2011b and run on a PC with Intel Q9300@2.50G (CPU) and 4GB of RAM. To properly measure the capability of various non-convex LRMF optimization models, all competing methods (except SVD and nuclear-norm based Robust PCA [43]) are run with 10 random initializations, and the best result is selected. All methods run a maximum of 100 iterations or stop when the difference between consecutive $\mathbf{U}$s is smaller than 0.01.

In all experiments, we compared our approach with several LRMF methods including Robust PCA (IRLS) [11][2]; nuclear-norm based Robust PCA [43] (NN-Robust PCA)[3]; The NN-Robust method provides the rank of the matrix as a function of the regularization parameters. In some experiments the rank is known a priori, and we have selected the regularization parameter that satisfies the required rank. two representative methods for L_2 LRMF: WLRA [35] and DN [6][4]; four state-of-the-art methods for

[2] http://www.cs.cmu.edu/~ftorre/

[3] http://perception.csl.illinois.edu/matrix-rank/sample_code.html

[4] http://www.robots.ox.ac.uk/~abm/

L_1 LRMF: ALP [20][5], L_1Wiberg [14][6], RegL1ALM [45][7] and CWM [27]. Because was not available, we implemented WLRA [35].

4.1. Synthetic experiments

Four sets of synthetic experiments were designed to evaluate the performance of our method against other LRMF methods with different types of noise. For each set of experiments, we randomly generated 30 low-rank matrices, each of size 40×20 and rank 4. Each of these matrices is generated by the multiplication of two low-rank matrices $\mathbf{U}_{gt} \in \Re^{40\times 4}$ and $\mathbf{V}_{gt} \in \Re^{20\times 4}$, and $\mathbf{X}_{gt} = \mathbf{U}_{gt}\mathbf{V}_{gt}^T$ is the ground truth matrix. Each element of $\mathbf{U}_{gt}$ and $\mathbf{V}_{gt}$ is generated from a Gaussian distribution $\mathcal{N}(0,1)$. In each experiment, we randomly specified 20% of missing entries in $\mathbf{X}_{gt}$ and further added different types of noise as follows: (1) *No noise* added. (2) *Gaussian noise* $\mathcal{N}(0,0.1)$. (3) *Sparse noise*: 20% of the entries were corrupted with uniformly distributed noise between $[-5,5]$. (4) *Mixture noise*: 20% of the entries were corrupted with uniformly distributed noise over $[-5,5]$, 20% are contaminated with Gaussian noise $\mathcal{N}(0,0.2)$, and the remaining 40% are corrupted Gaussian noise $\mathcal{N}(0,0.01)$. The noisy matrix is denoted as $\mathbf{X}_{no}$. The final performance of each method on each experiment was measured as the average over the 30 realizations and the error measured with six measures:

$$E1 = \left\|\mathbf{W} \odot (\mathbf{X}_{no} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T)\right\|_{L_1}, E2 = \left\|\mathbf{W} \odot (\mathbf{X}_{no} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T)\right\|_{L_2},$$
$$E3 = \left\|\mathbf{X}_{gt} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T\right\|_{L_1}, E4 = \left\|\mathbf{X}_{gt} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T\right\|_{L_2},$$
$$E5 = subspace(\mathbf{U}_{gt}, \widetilde{\mathbf{U}}), E6 = subspace(\mathbf{V}_{gt}, \widetilde{\mathbf{V}}),$$

where $\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}$ are the outputs of the corresponding LRMF method, and $subspace(\mathbf{U_1}, \mathbf{U_2})$ denotes the angle between subspaces spanned by the columns of $\mathbf{U_1}$ and $\mathbf{U_2}$. It is important to notice that existing methods optimize $E1$ or $E2$, but the last four measures ($E3 - E6$) are more meaningful to evaluate if the method recovers the correct subspace.

The performance of the methods are shown in Table 1. It can be observed from Table 1 that although the L_1 and L_2 LRMF methods generally perform better in terms of $E1$ and $E2$, respectively, the proposed subspace-MoG method performs best or the second best in all experiments in estimating a better subspace from noisy data (measurements $E3$-$E6$). Particularly, in the fourth set of experiments(when the noise is a mixture) our method always performs best in all errors.

[5]We used the code "l1decode_pd.m" [8] for solving the linear programming problem. The code was downloaded from "http://www-inst.eecs.berkeley.edu/~ee225B/sp08/lectures/CSmeetsML-Lecture1/codes/l1magic/Optimization".

[6]http://cs.adelaide.edu.au/~anders/code/cvpr2010.html

[7]https://sites.google.com/site/yinqiangzheng/

4.2. SFM experiments

The SFM problem can be formulated as a LRMF task [20, 14]. We used the well known dinosaur sequence[20] that contains projections of 319 points tracked over 36 frames, leading to a 319×72 matrix. The matrix contains around 77% missing data due to occlusions or tracking failures. We added four types of noise to the matrix: (1) *No noise* added. (2) *Gaussian noise* $\mathcal{N}(0,10)$. (3) *Sparse noise*: 10% of the non-missing elements were corrupted by uniformly distributed noise ($[-50,50]$). (4) *Mixture noise*: 10% of the non-missing elements were corrupted by uniformly distributed noise ($[-50,50]$), the remaining 90% were contaminated with Gaussian noise $\mathcal{N}(0,10)$. Four quantitative criteria were utilized for performance evaluation in these experiments, including:

$$E1 = \left\|\mathbf{W} \odot (\mathbf{X}_{no} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T)\right\|_{L_1}, E2 = \left\|\mathbf{W} \odot (\mathbf{X}_{no} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T)\right\|_{L_2},$$
$$E3 = \left\|\mathbf{W} \odot (\mathbf{X}_{gr} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T)\right\|_{L_1}, E4 = \left\|\mathbf{W} \odot (\mathbf{X}_{gr} - \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^T)\right\|_{L_2},$$

where $\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}$ are the outputs of the corresponding LRMF method. We compared with the same methods as in the previous experiments, with exception of L_1Wiberg because it did not fit into memory. The results are shown in Table 2.

Similar to our synthetic experiments, the proposed subspace-MoG method does not perform best among all competing methods in terms of $E1$ and $E2$. However, it always has the best or the second best performance in the error $E3$-$E6$. Especially, it performs the best in terms of both $E3$ and $E4$ in mixture noise experiments.

4.3. Face modeling experiments

This experiment aims to test the effectiveness of the proposed MoG method to build a model of the face under different illuminations. The data matrix is generated by extracting the first subset of the Extended Yale B database [16, 5], containing 64 faces of one subject with size 192×168 under different illuminations. The input matrix has a size of 32256×64. Typical images used are plotted in the first column of Fig. 2. We compared with Robust PCA (IRLS), NN-Robust PCA [43], SVD [17], RegL1ALM [45], CWM [22] and PCAL1 [22]. The SVD method is implemented with the Matlab function. Assuming perfect conditions, these face images would lie on a 4-D subspace [5]. we thus set the rank to 4 for all competing methods except nuclear-norm based Robust PCA, which automatically selects the rank. Fig. 2 compares some reconstructed images obtained by these competing methods.

From Fig. 2, it is easy to observe that the proposed method, as well as the other competing methods, are capable of removing the cast shadows and saturations in the faces, as shown in the first row of Fig. 2. However, our method performs better over the faces with large dark regions, as shown in the $2 - 4^{th}$ rows of Fig. 2. This can

	MoG	IRLS[11]	WLRA[35]	DN[6]	ALP[20]	L1Wiberg[14]	RegL1ALM[45]	CWM[27]	NN-Robust PCA[7]
					No Noise (log values)				
$E1$	-18.5	$\mathbf{-26.5}$	-2.17	-11.4	0.587	-8.73	-12.6	-0.222	-2.86
$E2$	-42.2	$\mathbf{-56.8}$	-9.38	-27.6	-4.27	-21.1	-30.0	-4.83	-10.8
$E3$	-17.9	$\mathbf{-25.9}$	-1.50	-11.0	1.03	-8.44	-11.9	0.324	-2.20
$E4$	-40.5	$\mathbf{-55.2}$	-7.06	-27.3	-3.37	-20.7	-27.9	-3.65	-8.73
$E5$	-23.8	$\mathbf{-32.5}$	-7.05	-18.9	-5.28	-14.6	-17.4	-5.35	-7.75
$E6$	-23.9	$\mathbf{-31.9}$	-8.15	-19.4	-6.18	-16.1	-18.4	-5.95	-8.83
					Gaussian Noise				
$E1$	40.0	40.0	40.0	40.0	37.5	$\mathbf{35.7}$	$\mathbf{35.7}$	39.1	40.0
$E2$	$\mathbf{3.99}$	$\mathbf{3.99}$	$\mathbf{3.99}$	$\mathbf{3.99}$	5.20	5.08	5.09	5.59	$\mathbf{3.99}$
$E3$	$\mathbf{39.3}$	$\mathbf{39.3}$	$\mathbf{39.3}$	$\mathbf{39.3}$	49.3	47.95	48.02	51.9	$\mathbf{39.3}$
$E4$	$\mathbf{3.29}$	$\mathbf{3.29}$	$\mathbf{3.29}$	$\mathbf{3.29}$	5.31	4.91	4.93	6.38	$\mathbf{3.29}$
$E5$	$\mathbf{0.0455}$	$\mathbf{0.0455}$	0.0456	$\mathbf{0.0455}$	0.0561	0.0541	0.0545	0.0636	$\mathbf{0.0455}$
$E6$	$\mathbf{0.0295}$	$\mathbf{0.0295}$	$\mathbf{0.0295}$	$\mathbf{0.0295}$	0.0395	0.0368	0.0367	0.0466	$\mathbf{0.0295}$
					Sparse Noise				
$E1$	400.6	519.1	518.5	519.1	403.6	$\mathbf{395.1}$	425.8	425.7	523.1
$E2$	1317.7	827.5	827.5	$\mathbf{827.3}$	1174.8	1270.8	1176.1	1125.1	834.7
$E3$	$\mathbf{54.1}$	624.1	623.2	624.4	159.9	67.4	446.4	278.0	628.0
$E4$	100.9	991.1	1002.3	995.4	171.2	$\mathbf{99.4}$	8984.9	342.4	976.05
$E5$	0.289	0.741	0.740	0.742	0.347	$\mathbf{0.279}$	0.978	0.471	0.733
$E6$	$\mathbf{0.0169}$	0.583	0.581	0.584	0.164	0.0284	0.698	0.338	0.587
					Mixture Noise				
$E1$	419.7	516.9	516.9	516.8	412.9	$\mathbf{404.5}$	417.9	430.8	520.5
$E2$	1274.3	$\mathbf{829.0}$	$\mathbf{829.0}$	$\mathbf{829.0}$	1119.1	1147.1	1124.5	1120.4	836.3
$E3$	$\mathbf{149.5}$	616.1	615.7	616.1	242.1	192.4	375.4	291.6	618.4
$E4$	$\mathbf{189.1}$	956.1	951.7	956.2	276.5	213.5	4245.4	336.6	955.9
$E5$	$\mathbf{0.374}$	0.692	0.691	0.692	0.427	0.377	0.696	0.461	0.701
$E6$	$\mathbf{0.155}$	0.579	0.579	0.579	0.230	0.175	0.530	0.346	0.589

Table 1. Six measures of error for the synthetic example with different noise models. The best and the second best results in each experiment are highlighted in bold and italic, respectively.

	MoG	IRLS[11]	WLRA[35]	DN[6]	ALP[20]	RegL1ALM[45]	CWM[27]	NN-Robust PCA[7]
					No Noise			
$E1$	0.442	1.83	8.24	0.490	4.85	$\mathbf{0.291}$	7.71	4.98
$E2$	1.13	3.18	12.1	$\mathbf{1.12}$	12.9	1.47	26.4	7.96
$E3$	0.442	1.83	8.24	0.490	4.85	$\mathbf{0.291}$	7.71	4.98
$E4$	1.13	3.18	12.1	$\mathbf{1.12}$	12.9	1.47	26.4	7.96
					Gaussian Noise			
$E1$	6.70	7.03	11.2	6.73	8.48	$\mathbf{6.14}$	12.2	9.05
$E2$	8.48	8.93	14.9	$\mathbf{8.43}$	15.7	9.31	21.9	12.2
$E3$	4.55	5.01	9.41	$\mathbf{4.49}$	8.05	5.39	11.6	7.04
$E4$	5.87	6.43	13.2	$\mathbf{5.85}$	14.2	6.91	20.7	10.2
					Sparse Noise			
$E1$	3.24	4.95	10.48	4.53	6.42	$\mathbf{2.85}$	10.3	7.44
$E2$	9.31	8.27	14.90	$\mathbf{7.90}$	17.7	9.22	19.6	11.5
$E3$	1.29	4.08	9.34	3.58	4.91	$\mathbf{0.524}$	9.02	6.41
$E4$	4.28	6.13	13.09	5.49	15.8	$\mathbf{1.98}$	18.0	9.68
					Mixture Noise			
$E1$	8.27	8.70	12.8	8.42	10.6	$\mathbf{7.67}$	13.1	10.0
$E2$	11.2	11.4	16.9	$\mathbf{10.99}$	18.2	12.3	23.8	13.3
$E3$	$\mathbf{5.82}$	5.97	9.99	5.83	9.19	6.31	11.7	7.19
$E4$	$\mathbf{7.81}$	7.88	13.8	7.82	15.5	8.44	21.5	9.83

Table 2. Performance comparison of the competing methods in the SFM experiments. The best and the second best results in each experiment are highlighted in bold and italic, respectively.

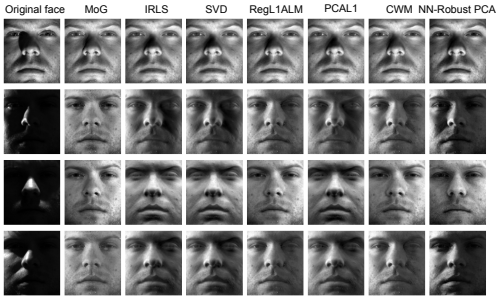

Figure 2. From left to right: original face images, faces reconstructed by MoG, Robust PCA (IRLS), SVD, RegL1ALM, PCAL1, CWM and nuclear-norm based Robust PCA, respectively.

be easily explained by Fig. 1(d). Unlike existing methods, our approach is able to model a mixture of Gaussians noise. One mixture, with large variance models shadows, saturated regions and outliers in the face (see the blue and yellow areas in the second noise image of Fig. 1(d)). The other mixture, with smaller variance, accounts for the camera noise which are specially amplified in the dark area of the face (see the first noise image of Fig. 1(d)).

4.4. Background subtraction experiments

This experiment compares our approach in the problem of background subtraction [11]. We built a background model by performing LRMF in seven video sequences pro-

vided by [24][8] (600 frames of 128×160 pixels) and one in [11][9] (506 frames of 120×160 pixels). The sequences include variations due to lighting changes, people walking, shadows, etc. See Fig. 3 and Fig. 4.

We applied Robust PCA (IRLS) [11]; NN-Robust PCA [43]; the state-of-the-art method for L_2 LRMF: SVD [17]; three state-of-the-art methods for L_1 LRMF: RegL1ALM [45], CWM [22], PCAL1 [22]; and our subspace-MoG (MoG). The dimension of the subspace was set to 6 to the videos from [24] and 15 for the video in [11].

Fig. 3 illustrates the results of running different LRMF methods in the videos provided by [24]. We observe that all methods can provide a good background model. However, the proposed subspace-MoG method provides a more accurate model that decomposes the foreground information into three components with different variance from small to large: (1) background variation corresponding mostly to camera noise; (2) shadows alongside the foreground object; (3) moving objects in the foreground (see $2-4^{th}$ rows of Fig. 3). The foreground extracted by the other competing methods is more coarse because it merges the object and its shadow (see Frame 402 for easy visualization). A more obvious case of the good performance of our method can be observed from the video from [11], as seen in Fig. 4. Our method reconstructs better illumination variations and is not biased by the random people walking, shadows, specular reflections, or motion of the tree.

5. Conclusions

This paper proposes a new low-rank factorization method to estimate subspaces with an unknown noise distribution. The noise is modeled as a MoG, and the parameters of the subspace-MoG model are learned from data automatically. Compared to existing L_2 and L_1 LRMF methods that are optimal for Gaussian or Laplacian noises, our method performs better (on average) in a wide variety of synthetic and real noise experiments. Our method has proven useful in modeling different types of noise in faces under different illuminations and background subtraction. A limitation of our approach is the non-convexity of the cost function. Currently, we are exploring spectral approaches to improve robustness to local minima. Finally, adding robustness to different types of noise can be similarly applied to other component analysis methods (e.g., linear discriminant analysis, normalized cuts) that are formulated as least-squares problems [10].

Acknowledgement

This research was supported by 973 Program of China with No. 3202013CB329404 and the NSFC projects with No. 61373114, 11131006, 6107505. Fernando De la Torre was partially supported by Grant CPS-0931999 and NSF IIS-1116583. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[8] http://perception.i2r.a-star.edu.sg/bk_model/bk_index

[9] http://www.cs.cmu.edu/~ftorre/codedata.html

References

[1] P. M. Q. Aguiar, J. M. F. Xavier, and M. Stosic. Spectrally optimal factorization of incomplete matrices. In *CVPR*, 2008. 2

[2] D. Andrews and C. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36(1):99–102, 1974. 2, 3

[3] C. Archambeau, N. Delannay, and M. Verleysen. Robust probabilistic projections. In *ICML*, 2006. 2

[4] D. J. Bartholomew. *Latent Variable Models and Factor Analysis*. Charles Griffin, 1987. 2

[5] R. Basri and D. W. Jacobs. Lambertian reflection and linear subspaces. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25:218–233, 2003. 5

[6] A. Buchanan and A. Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *CVPR*, 2005. 2, 4, 6

[7] E. Candès, X. D. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58, 2011. 6

[8] E. Candès and J. Romberg. *l1-MAGIC: recovery of sparse signals via convex programming*. Technical Report, California Institute of Technology, 2005. 5

[9] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *International Journal of Computer Vision*, 80:125–142, 2008. 1, 2

[10] F. De la Torre. A least-squares framework for component analysis. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 34(6):1041–1055, 2012. 7

[11] F. De la Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:117–142, 2003. 1, 2, 4, 6, 7

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, B*, 39(1):1–38, 1977. 3, 4

[13] C. Ding, D. Zhou, X. F. He, and H. Y. Zha. R1-PCA: Rotational invariant l1-norm principal component analysis for robust subspace factorization. In *ICML*, 2006. 1

[14] A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l_1 norm. In *CVPR*, 2010. 1, 2, 5, 6

[15] K. R. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21:489–498, 1979. 2

[16] 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 Pattern Analysis and Machine Intelligence*, 23:643–660, 2001. 2, 5

[17] G. H. Golub and C. F. van Loan. *Matrix Computation*. Maryland: Johns Hopkins University Press, 1989. 5, 7

[18] H. Ji, C. Q. Liu, Z. W. Shen, and Y. H. Xu. Robust video denoising using low rank matrix completion. In *CVPR*, 2010. 1

[19] C. Julià, F. Lumbreras, and A. D. Sappa. A factorization-based approach to photometric stereo. *International Journal of Imaging Systems and Technology*, 21:115–119, 2011. 1

[20] Q. F. Ke and T. Kanade. Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, 2005. 1, 2, 5, 6

[21] J. J. Koenderink and A. J. van Doorn. The generic bilinear calibration-estimation problem. *International Journal of Computer Vision*, 23(3):217–234, 1997. 2

[22] N. Kwak. Principal component analysis based on l1-norm maximization. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 30:1672–1680, 2008. 1, 2, 5, 7

[23] B. Lakshminarayanan, G. Bouchard, and C. Archambeau. Robust bayesian matrix factorisation. In *AISTATS*, 2011. 2

[24] L. Y. Li, W. M. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004. 7

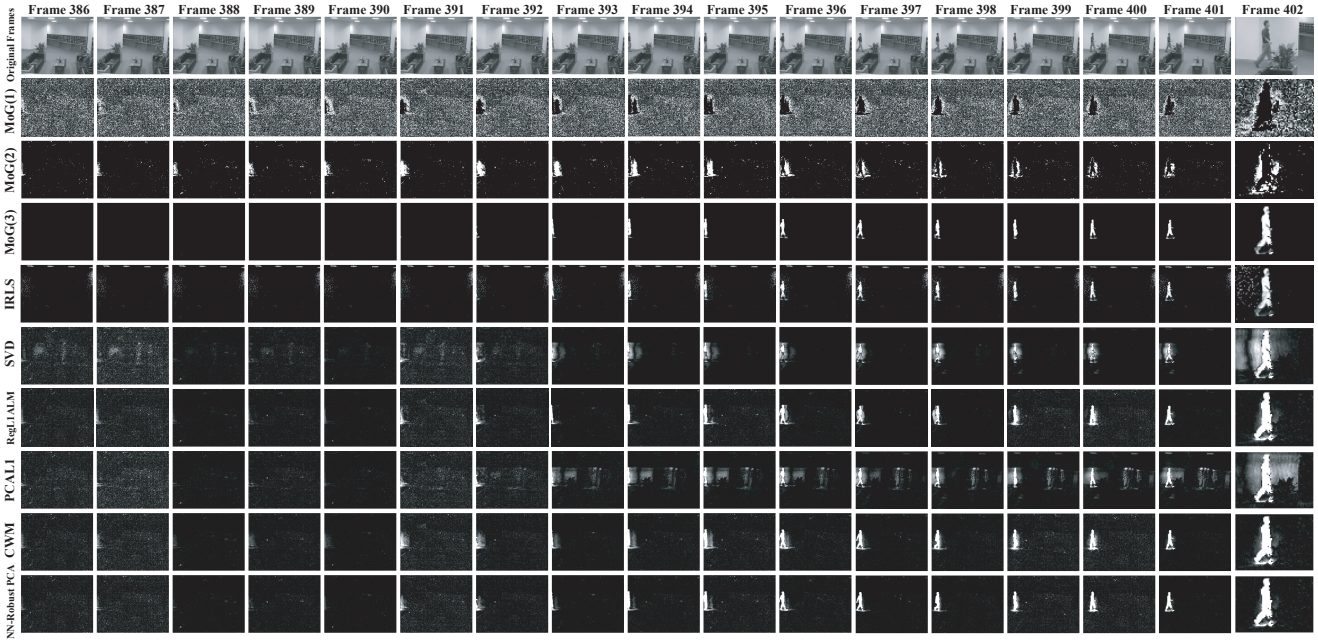

Figure 3. From top row to bottom: original Lobby frames, absolute errors computed by different methods(the details are better seen by zooming on a computer screen). The moving object region in Frame 402 is enlarged for better visualization.

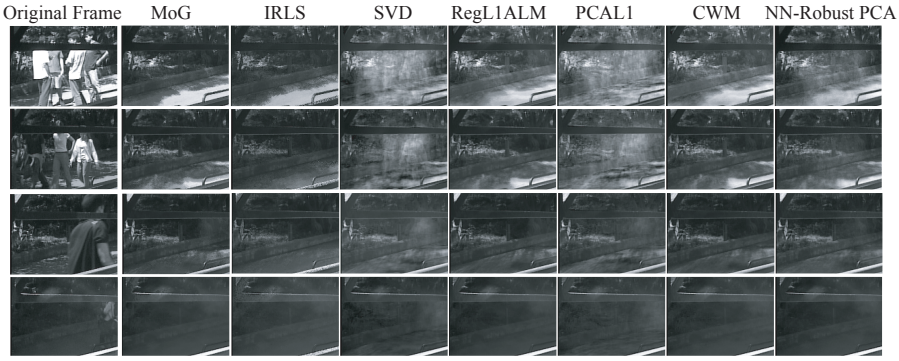

Figure 4. From left to right: original frames, reconstructed backgrounds computed by different methods.

[25] V. Maz'ya and G. Schmidt. On approximate approximations using gaussian kernels. *IMA Journal of Numerical Analysis*, 16(1):13–29, 1996. 2, 3

[26] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988. 2

[27] D. Y. Meng, Z. B. Xu, L. Zhang, and J. Zhao. A cyclic weighted median method for l1 low-rank matrix factorization with missing entries. *AAAI*, 2013. 5, 6

[28] K. Mitra, S. Sheorey, and R. Chellappa. Large-scale matrix factorization with missing data under additional constraints. In *NIPS*, 2010. 1, 2

[29] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 19:696–710, 1997. 2

[30] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, 2005. 2

[31] T. Okatani and K. Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72:329–337, 2007. 1, 2

[32] T. Okatani, T. Yoshida, and K. Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *ICCV*, 2011. 1

[33] S. Roweis. EM algorthms for PCA and SPCA. In *NIPS*, 1998. 2

[34] A. Shashua. On photometric issues in 3d visual recognition from a single 2d image. *International Journal of Computer Vision*, 21:99–122, 1997. 2

[35] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *ICML*, 2003. 1, 2, 4, 5, 6

[36] P. Sturm. Algorithms for plane-based pose estimation. In *CVPR*, 2000. 1

[37] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11:443–482, 1999. 2

[38] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61:611–622, 1999. 2

[39] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992. 1

[40] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3:71–86, 1991. 1

[41] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using power factorization and GPCA. *International Journal of Computer Vision*, 79:85–105, 2008. 1

[42] N. Y. Wang, T. S. Yao, J. D. Wang, and D. Y. Yeung. A probabilistic approach to robust matrix factorization. In *ECCV*, 2012. 2

[43] J. Wright, Y. G. Peng, Y. Ma, A. Ganesh, and S. Rao. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *NIPS*, 2009. 1, 2, 4, 5, 7

[44] K. Zhao and Z. Y. Zhang. Successively alternate least square for low-rank matrix factorization with bounded missing data. *Computer Vision and Image Understanding*, 114:1084–1096, 2010. 1, 2

[45] Y. Q. Zheng, G. C. Liu, S. Sugimoto, S. C. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust l1-norm. In *CVPR*, 2012. 2, 5, 6, 7