Fast Algorithms for Structured Robust Principal Component Analysis

Mustafa Ayazoglu, Mario Sznaier and Octavia I. Camps[*]
Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115
http://robustsystems.ece.neu.edu

Abstract

A large number of problems arising in computer vision can be reduced to the problem of minimizing the nuclear norm of a matrix, subject to additional structural and sparsity constraints on its elements. Examples of relevant applications include, among others, robust tracking in the presence of outliers, manifold embedding, event detection, inpainting and tracklet matching across occlusion. In principle, these problems can be reduced to a convex semi-definite optimization form and solved using interior point methods. However, the poor scaling properties of these methods limit the use of this approach to relatively small sized problems. The main result of this paper shows that structured nuclear norm minimization problems can be efficiently solved by using an iterative Augmented Lagrangian Type (ALM) method that only requires performing at each iteration a combination of matrix thresholding and matrix inversion steps. As we illustrate in the paper with several examples, the proposed algorithm results in a substantial reduction of computational time and memory requirements when compared against interior-point methods, opening up the possibility of solving realistic, large sized problems.

1. Introduction

During the past few years considerably attention has been devoted to the *Robust PCA problem:* [2, 3] decomposing a given data matrix $\mathbf{D}$ as $\mathbf{D} = \mathbf{A} + \mathbf{E}$ where $\mathbf{A}$ has low rank and E is sparse. Intuitively, this problem seeks to recover an underlying low rank matrix $\mathbf{A}$ from experimental measurements $\mathbf{D}$ corrupted by outliers $\mathbf{E}$ that are sparse but can have large magnitude. Examples of applications include image and video restoration [8], video surveillance [2], background substraction [7], image alignment [10], removing shadows from face images [2], and motion segmentation in the presence of outliers [6]. While in principle the problem above is NP–hard, it has been shown [2] that if the matrix of outliers $\mathbf{E}$ is sufficiently sparse (relative to the rank of $\mathbf{A}$) and the sparsity pattern of $\mathbf{E}$ is random, then $\mathbf{A}$ can be recovered from $\mathbf{D}$ by simply solving the following (convex) optimization problem[1]:

$$\min_{\mathbf{A},\mathbf{E}} \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 \text{ subject to: } \mathbf{D} = \mathbf{A} + \mathbf{E} \quad (1)$$

where $\|.\|_*$ denotes the nuclear norm. In turn, this problem can be efficiently solved using a number of methods that include, in addition to interior point, Iterative Thresholding (IT), Accelerated Proximal Gradient (APG) and Augmented Lagrange Multipliers (ALM) [9]. Unfortunately, these methods cannot handle the case where the matrices $\mathbf{A}$ and $\mathbf{E}$ are subject to structural constraints.

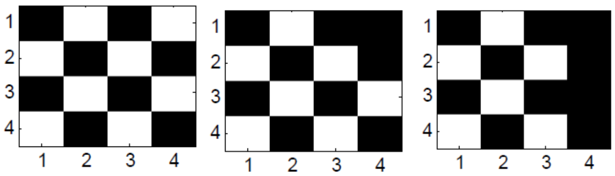


Figure 1. Restoring a textured imaged. Left: Original image. Center: Image corrupted with a single outlier at position (1,4). Right: Imaged restored using RPCA

To motivate the need to incorporate these constraints, consider for instance the problem of restoring the image shown in Figure 1, a checkerboard where a single entry has been corrupted by an outlier. The pixel values of the original and corrupted images, $\mathcal{I}$ and $\mathcal{I}_c$, are given by

$$\mathcal{I} = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \end{bmatrix} \mathcal{I}_c = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \end{bmatrix} \quad (2)$$

It can be easily shown that restoring the image by solving

$$\mathcal{I}_r = argmin\, \|\mathcal{I}_r\|_* + 0.1\|E\|_1 \text{ subject to: } \mathcal{I}_r = \mathcal{I}_c + E$$

[*] This work was supported in part by NSF grants IIS–0713003 and ECCS–0901433, AFOSR grant FA9550–09–1–0253, and the Alert DHS Center of Excellence under Award Number 2008-ST-061-ED0001.

[1] Consistent numerical experience shows that this approach typically succeeds even when these conditions do not hold.

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

yields the solution:

$$\mathcal{I}_r = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \end{bmatrix}, E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which is clearly incorrect. On the other hand, pursuing the Hankel based inpainting approach proposed in [11] leads to the following problem:

$$\begin{aligned} \mathbf{H}_{\mathcal{I}_r} \quad &= argmin \, \|\mathbf{H}_{\mathcal{I}_r}\|_* + \lambda \mathbf{H}_E \\ &\text{subject to:} \mathbf{H}_{\mathcal{I}_r} = \mathbf{H}_{\mathcal{I}_c} + \mathbf{H}_E \end{aligned} \tag{3}$$

where, for a given $n \times m$ matrix $\mathbf{X}, \mathbf{H}_{\mathcal{X}}$ denotes the block circulant Hankel matrix

$$\mathbf{H}_{\mathcal{X}} \doteq \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \dots & \mathbf{R}_n \\ \mathbf{R}_2 & \mathbf{R}_3 & \dots & \mathbf{R}_1 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_n & \mathbf{R}_1 & \dots & \mathbf{R}_{n-1} \end{bmatrix}. \tag{4}$$

where $\mathbf{R}_i$ denotes the i^{th} column of $\mathbf{X}$. In this case, "dehankelizing" the solution to the optimization problem (3) indeed recovers the correct image. Note that Problem (3) does not fall under the form (1), due to the constraint that $\mathbf{H}_{\mathcal{I}_r}$ (and hence $\mathbf{H}_E$) must have a block-circulant Hankel structure. Indeed, problem (3) is a special case of the *Structured Robust PCA problem* (SRPCA) addressed in this paper:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{E}} \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 \text{ subject to:} \\ &\mathbf{D} = \mathbf{A} + \mathbf{E} \text{ and structural constraints on } \mathbf{A} \text{ and } \mathbf{E} \end{aligned} \tag{5}$$

Since this problem is convex, it can be solved for instance using interior point methods. However, while convergence of these methods is usually fast, they have poor scaling properties (typically for an $n \times n$ matrix, the complexity of each iteration is $\mathcal{O}(n^6)$), and hence their use is restricted to small size problems. On the other hand, the existence of structural constraints prevents direct use of the fast first order methods developed to solve RPCA problems. Motivated by these difficulties, in this paper we present a fast, computationally efficient algorithm for solving SRPCA problems. As in the case of the state–of–the–art methods for solving the unconstrained case, the proposed method uses only first-order information, hence avoiding the computational complexity of interior-points methods, and converges Q-linearly (or Q-superlinearly) to the optimum. On the other hand, it can handle a variety of both structural and semi-definite constraints. These results are illustrated with several examples drawn from a broad spectrum of computer vision problems.

2. Preliminaries

In this section we summarize, for ease of reference, the notation used in the paper and some key background results.

2.1. Notation

$\mathbf{M}^T$	Transpose of the matrix $\mathbf{M}$
Trace$\{\mathbf{M}\}$	Trace of the square matrix $\mathbf{M}$.
$\sigma_i(\mathbf{M})$	i^{th} largest singular value of $\mathbf{M}$.
$\circ$	Hadamard product of matrices: $(\mathbf{A} \circ \mathbf{B})_{i,j} = \mathbf{A}_{i,j}\mathbf{B}_{i,j}$.
$\langle \mathbf{M}, \mathbf{N} \rangle$	Inner product in the space of square $n \times n$ matrices defined as $\langle \mathbf{M}, \mathbf{N} \rangle \doteq$ Trace$\{\mathbf{M}^T\mathbf{N}\}$
$\|\mathbf{M}\|_F$	Frobenious norm: $\|\mathbf{M}\|_F^2 \doteq \langle \mathbf{M}, \mathbf{M} \rangle =$ Trace$\{\mathbf{M}^T\mathbf{M}\}$
$\|\mathbf{M}\|_*$	Nuclear norm: $\|\mathbf{M}\|_* = \sum_i \sigma_i(\mathbf{M})$.
$\|\mathbf{M}\|_1$	ℓ_1 norm: $\|\mathbf{M}\|_1 = \sum_{i,j} \|M_{ij}\|$.
vect$(\mathbf{M})$	Matrix vectorizing operator: $\mathbf{m} =$ **vect**$(\mathbf{M})$ is a vector formed by stacking the columns of $\mathbf{M}$.
mat$(\mathbf{a}, n)$	Vector to matrix operation: $\mathbf{mat}(\mathbf{a}, n) \doteq \begin{bmatrix} \mathbf{a}(1 : n-1) & \mathbf{a}(n : 2n-1) \dots \end{bmatrix}$ in the sequel, the dimension n will be omitted when clear from the context.
$\mathcal{D}(x, \tau, w)$	Weighted soft thresholding operation: $\mathcal{D}(x, \tau, w) = \max\{0, sign(x)(\|x\| - \tau w)\}$ When applied to matrices, $\mathcal{D}(.,.,.)$ acts on each element, by shrinking each element in the matrix by its corresponding weight w(i,j) and τ product. In the sequel, by a slight abuse of notation we will use $\mathcal{D}(x, \tau)$ when $w = 1$.

2.2. Augmented Lagrangian Method

Consider a constrained optimization problem of the form:

$$min_{\mathbf{X}} f(\mathbf{X}) \text{ subject to } h(\mathbf{X}) = 0 \tag{6}$$

The Augmented Lagrangian Method (ALM) seeks to solve this problem by forming the augmented Lagrangian:

$$L(\mathbf{X}, \mathbf{Y}, \mu) \quad = f(\mathbf{X}) + \langle \mathbf{Y}, h(\mathbf{X}) \rangle + \frac{\mu}{2} \|h(\mathbf{X})\|_F^2 \tag{7}$$

and proceeding (iteratively) as follows:

Algorithm 2.2: GENERAL ALM

```
while not converged do
  1. $\mathbf{X}_{k+1} = argmin_X L(\mathbf{X}_k, \mathbf{Y}_k, \mu_k)$
  2. $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \rho h(\mathbf{X}_{k+1})$
  3. $\mu_{k+1} = \rho\mu_k$
end while
```

It can be shown [1] that if μ is an increasing sequence and f and h are smooth, then the algorithm above converges at least Q-linearly to the optimal solution. These results were further extended in [9] to problems of the form (1), where the objective is not continuously differentiable everywhere.

2.3. The Structured RPCA problem

The general form of the SRPCA problem addressed in this paper can be formally stated as:

Problem 1 (SRPCA). *Given a data matrix D and weights w_i, $\mathbf{W}_1$ and $\mathbf{W}_F$, solve*

$$\begin{array}{l} \min_{\mathbf{A},\mathbf{E}} \sum_i w_i \sigma_i(\mathbf{A}) + \|\mathbf{W}_e \circ \mathbf{E}\|_1 + ... \\ \quad ... \frac{1}{2} \|\mathbf{W}_F \circ \mathbf{E}\|_F^2 \\ \textit{subject to } \mathbf{D} = \mathbf{A} + \mathbf{E},\ \mathbf{A} \in \mathcal{S}_A \textit{ and } \mathbf{E} \in \mathcal{S}_E \end{array} \tag{8}$$

where $\mathcal{S}_A$, $\mathcal{S}_E$, the sets that define the structure of $\mathbf{A}$ and $\mathbf{E}$ are of the form $\mathcal{S}_X = \{X : X = \textbf{mat}(\mathbf{S}\mathbf{x})\}$, where $\mathbf{S}$ is a given matrix with full column rank and $\mathbf{x}$ is an arbitrary vector of appropriate dimensions.

Remark 1. *Note that the structure above is quite general, capturing all cases where there is a linear dependence amongst the elements of the matrix under consideration. For instance, if $\mathbf{A} \in R^{3\times 3}$ is restricted to have a Hankel structure, then it can be written as $\mathbf{A} = \textbf{mat}(\mathbf{S}_H \mathbf{a})$ where $\mathbf{a} \in R^5$ and*

$$\mathbf{S}_H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Remark 2. *The objective function (8) above can be thought of as a convex relaxation of the problem of recovering a (structured) low rank matrix from measurements corrupted both by noise and outliers. Using the re-weighted nuclear and ℓ_1 norms as surrogates for rank and cardinality, respectively, leads precisely to this problem. Note also that if $w_i = 1$, $\mathbf{W}_F = 0$, $\mathbf{W}_e = \lambda\mathbf{1}$ and $\mathcal{S}_E = \mathcal{S}_A = R^{n\times n}$, then the RPCA problem (1) is recovered.*

3. An ALM approach to Structured RPCA

The key point in applying an algorithm of the form (2.2) to Problem 1 is to develop a computationally efficient way of finding the minimizers in the first step. As we show next, this can be accomplished by using a combination of thresholding and matrix inversion steps. Recall that in the case of RPCA, at each iteration, the explicit solution to

$$\min_{\mathbf{A},\mathbf{E}} \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 + \langle \mathbf{Y}, \mathbf{D} - \mathbf{A} - \mathbf{E}\rangle + \frac{\mu}{2}\|\mathbf{D} - \mathbf{A} - \mathbf{E}\|_F^2$$

is given by [9]:

$$\begin{array}{rcl} \mathbf{A}^{k+1} & = & \mathbf{U}\mathcal{D}(\mathbf{\Sigma}, \mu_k^{-1})\mathbf{V}^T \\ \mathbf{E}^{k+1} & = & \mathcal{D}(\mathbf{D} - \mathbf{A}^{k+1} + \frac{1}{\mu_k}\mathbf{Y}_k, \lambda\mu_k^{-1}) \end{array}$$

where $\mathbf{U\Sigma V}^T$=svd$(\mathbf{D} - \mathbf{E}_k + \frac{1}{\mu_k}\mathbf{Y}_k)$ The main barrier in applying ALM type methods to solve Problem 1 is that, contrary to the situation above, when $\mathbf{A}$ and $\mathbf{E}$ are subject to structural constraints, the resulting problem does not admit an explicit solution at each step. As we show next, this difficulty can be circumvented by adding new variables $\mathbf{J}$ and $\mathbf{T}$, subject to the constraints $\mathbf{J} = \mathbf{A}$ and $\mathbf{T} = \mathbf{E}$. While this seems a trivial step, when incorporated to the augmented Lagrangian, these new variables and constraints allow for decoupling the problem of minimizing a combination of the nuclear and ℓ_1 norm from that of enforcing the structural constraints. In turn, this allows for finding computationally efficient closed form solutions at each iteration. It is worth mentioning that in this approach, the structural constraints on $\mathbf{A}$ and $\mathbf{E}$ are not enforced in the intermediate steps (but, due to convergence are guaranteed to hold for the optimal solution). This intermediate constraint relaxation results in substantial speed-up vis-a-vis methods that enforce the constraint at each step.

3.1. An Exact ALM method

When the new variables and constraints are added to the problem, the resulting augmented Lagrangian is given by:

$$\begin{array}{l} L(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \mu) = \sum_i w_i \sigma_i(\mathbf{J}) + \|\mathbf{W}_e \circ \mathbf{T}\|_1 \\ + \frac{1}{2} \|\mathbf{W}_F \circ \mathbf{S}_E \mathbf{e}\|_F^2 + \langle \mathbf{Y}_1, \mathbf{J} - \textbf{mat}(\mathbf{S}_A \mathbf{a})\rangle \\ + \left\langle \mathbf{S}_A^T \textbf{vec}(\mathbf{Y}_2), \mathbf{S}_\mathbf{A}^\mathbf{T} \textbf{vec}(\mathbf{D}) - \mathbf{S}_\mathbf{A}^\mathbf{T}\mathbf{S}_\mathbf{A}\mathbf{a} - \mathbf{S}_\mathbf{A}^\mathbf{T}\mathbf{S}_\mathbf{E}\mathbf{e} \right\rangle \\ + \langle \mathbf{Y}_3, \mathbf{T} - \textbf{mat}(\mathbf{S}_E \mathbf{e})\rangle + (\mu/2)(\|\mathbf{J} - \textbf{mat}(\mathbf{S}_A \mathbf{a}))\|_F^2 \\ + \left\|\mathbf{S}_\mathbf{A}^\mathbf{T} \textbf{vec}(\mathbf{D}) - \mathbf{S}_\mathbf{A}^\mathbf{T}\mathbf{S}_\mathbf{A}\mathbf{a} - \mathbf{S}_\mathbf{A}^\mathbf{T}\mathbf{S}_\mathbf{E}\mathbf{e}\right\|_2^2 \\ + \|\mathbf{T} - \textbf{mat}(\mathbf{S}_E \mathbf{e})\|_F^2) \end{array}$$

where $\mathbf{X} \doteq (\mathbf{J}, \mathbf{a}, \mathbf{T}, \mathbf{e})$. Note that this problem is convex and hence can be solved by successively minimizing $L(.,,)$ with respect to each of the elements of $\mathbf{X}$. At the beginning of the $(k+1)^{th}$ iteration, the values of $\mathbf{a}^k, \mathbf{T}^k, \mathbf{e}^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k, \mathbf{Y}_3^k, \mu^k$ are known and the goal is to minimize L with respect to $\mathbf{J}^{k+1}$. A standard completion of the square argument shows that the optimal $\mathbf{J}^{k+1}$ is given by:

$$\begin{array}{l} \mathbf{J}^{k+1} = argmin_J \sum_i w_i \sigma_i(\mathbf{J}) \\ + \frac{\mu^k}{2}(\left\|\mathbf{J} - \textbf{mat}(\mathbf{S}_A \mathbf{a}^k) + \mathbf{Y}_1^k/\mu^k\right\|_F^2) \end{array}$$

a problem whose explicit solution is given by [12]:

$$\mathbf{J}^{k+1} = \mathbf{U}\mathcal{D}(\mathbf{\Sigma}, 1/\mu^k, \mathbf{W})\mathbf{V}^T$$

where $\mathbf{U\Sigma V}^T$=svd$(\mathbf{mat}(\mathbf{S}_A\mathbf{a}^k) - \mathbf{Y}_1^k/\mu^k)$ and $\mathbf{W}$ is a diagonal matrix with entries $W_{ii} = w_i$.

Next, consider the problem of minimizing L with respect to $\mathbf{a}$ and note that, due to the introduction of the new variable $\mathbf{J}$, L is differentiable with respect to $\mathbf{a}$. Hence, setting $\partial_a L = 0$ and using the fact that $\mathbf{S}_A$ has full column rank leads to:

$$\begin{aligned} \mathbf{a} \;\; = \;\; & \left[(\mathbf{I} + \mathbf{S}_A^T\mathbf{S}_A)\right]^{-1}\mathbf{S}_A^T\left[\mathbf{vect}(\mathbf{D}) - \mathbf{S}_E\mathbf{e} + \mathbf{vect}(\mathbf{Y_2}/\mu)\right. \\ + \;\; & \left.\left(\mathbf{S}_A^T\mathbf{S}_A\right)^{-1} S_A^T\left(\mathbf{vect}(\mathbf{J}) + \mathbf{vect}(\mathbf{Y_1}/\mu)\right)\right] \end{aligned} \tag{9}$$

Similarly, using a completion of squares argument to solve for $\mathbf{T}$ leads to:

$$\begin{aligned} \mathbf{T}^{k+1} \quad & = argmin_T \tfrac{1}{\mu^k}\left\|\mathbf{W}_e \circ \mathbf{T}\right\|_1 \\ & + \tfrac{1}{2}\left\|\mathbf{T} - \mathbf{mat}(\mathbf{S_E e}) + \mathbf{Y_3^k}/\mu^\mathbf{k}\right\|_F^2 \end{aligned}$$

As before, this problem admits an explicit solution given by:

$$\mathbf{T}^{k+1} = \mathcal{D}(\mathbf{mat}(\mathbf{S}_E\mathbf{e}^k) - \frac{\mathbf{Y}_3^k}{\mu^k}, \frac{1}{\mu^k}, \mathbf{W_e})$$

Finally, setting $\partial_e L = 0$ and solving for $\mathbf{e}$ yields:

$$\begin{aligned} \mathbf{e} = (\mathbf{S}_E^T\mathbf{S}_A\mathbf{S}_A^T\mathbf{S}_E + \mathbf{S}_E^T\mathbf{S}_E + \tfrac{1}{\mu}\mathbf{S}_E^T\hat{\mathbf{W}}_F^T\hat{\mathbf{W}}_F\mathbf{S}_E)^{-1} \\ (\mathbf{S}_E^T\left[\mathbf{vect}(\mathbf{T}) + \mathbf{vect}(\mathbf{Y_3}/\mu)\right] \\ + \mathbf{S}_E^T\mathbf{S}_A\mathbf{S}_A^T\left[\mathbf{vect}(\mathbf{D}) - \mathbf{S}_A\mathbf{a} + \mathbf{vect}(\mathbf{Y_2}/\mu)\right]) \end{aligned} \tag{10}$$

Here $\hat{\mathbf{W}}_F = diag(\mathbf{vect}(\mathbf{W}_F))$

Iteratively repeating the steps above leads to the optimal $\mathbf{X}^{k+1}$, for given values of the Lagrange multipliers $\mathbf{Y}_i^k$. Once $\mathbf{X}^{k+1}$ is available, the method proceeds as in the standard ALM case, updating the Lagrange multipliers using:

$$\begin{aligned} \mathbf{Y}_1^{k+1} & = \mathbf{Y}_1^k + \mu^k(\mathbf{J}^{k+1} - \mathbf{mat}(\mathbf{S}_A\mathbf{a}^{k+1})) \\ \mathbf{Y}_2^{k+1} & = \mathbf{Y}_2^k + \mu^k(\mathbf{D} - \mathbf{mat}(\mathbf{S}_A\mathbf{a}^{k+1}) - \mathbf{mat}(\mathbf{S}_E\mathbf{e}^{k+1})) \\ \mathbf{Y}_3^{k+1} & = \mathbf{Y}_3^k + \mu^k(\mathbf{T}^{k+1} - \mathbf{mat}(\mathbf{S}_E\mathbf{e}^{k+1})) \end{aligned} \tag{11}$$

and setting $\mu^{k+1} = \rho\mu^k$, for some $\rho > 1$. The complete algorithm is summarized next.

Algorithm 3.1: SRPCA ALGORITHM EALM

```
$\rho > 1$, $\mathbf{m}_0$
$\mathbf{a}^0 = \mathbf{S}_A^+\mathbf{vect}(\mathbf{D})$ ($\mathbf{S}^+$ is the pseudo inverse)
$\mathbf{J}^0 = \mathbf{D}$, $\mathbf{e}^0 = \mathbf{0}$, $\mathbf{T}^0 = \mathbf{0}$
$\mathbf{Y}_1^0 = \mathbf{0}$, $\mathbf{Y}_2^0 = \mathbf{0}$, $\mathbf{Y}_3^0 = \mathbf{0}$
while not converged do (outer loop)
    while not converged do (inner loop)
      1. $\mathbf{J}^{k+1} = \mathbf{U}\mathcal{D}(\mathbf{\Sigma}, 1/\mu^k, \mathbf{W})\mathbf{V}^T$,
      $\mathbf{U\Sigma V}^T$=svd$(\mathbf{mat}(\mathbf{S}_A\mathbf{a}^k) - \mathbf{Y}_1^k/\mu^k)$
      2. Solve $\mathbf{a}^{k+1}$ using eq. (9)
      3. $\mathbf{T}^{k+1} = \mathcal{D}(\mathbf{mat}(\mathbf{S}_E\mathbf{e}^k) - \frac{\mathbf{Y}_3^k}{\mu^k}, \frac{1}{\mu^k}, \mathbf{W_e})$
      4. Solve $\mathbf{e}^{k+1}$ using eq. (10)
    end while (inner loop)
    5. Update the lagrange multipliers using (11)
    6. $\mu^{k+1} = \rho\mu^k$
end while (outer loop)
```

Theorem 1. *The sequence $\mathbf{X}^k$ generated by Algorithm 3.1 converges to a solution of Problem 1. Further $L(\mathbf{X}^k, \mathbf{Y}^k)$ converges as $\mathcal{O}(\frac{1}{\mu^k})$ to the optimal value of the objective.*

Proof. See the Appendix □

3.2. An Inexact ALM method

While the algorithm above is guaranteed to converge Q-linearly to the optimal solution, the exact minimization of L in steps 1–4 could entail many iterations. Motivated by [9], we propose to avoid these iterations by considering an *inexact* ALM algorithm, where $(\mathbf{J}, \mathbf{a}, \mathbf{T}, \mathbf{e})$ are updated only once in each iteration. The entire algorithm can be summarized as

Algorithm 3.2: WEIGHTED SRPCA IALM ALGORITHM

```
$\rho > 1$, $\mathbf{m}_0$
$\mathbf{a}^0 = \mathbf{S}_A^+\mathbf{vect}(\mathbf{D})$ ($\mathbf{S}^+$ is the pseudo inverse)
$\mathbf{J}^0 = \mathbf{D}$, $\mathbf{e}^0 = \mathbf{0}$, $\mathbf{T}^0 = \mathbf{0}$
$\mathbf{Y}_1^0 = \mathbf{0}$, $\mathbf{Y}_2^0 = \mathbf{0}$, $\mathbf{Y}_3^0 = \mathbf{0}$
while not converged do
    1. Run the inner loop of Algorithm (3.1) once.
    2. Update the lagrange multipliers using (11)
    3. $\mu^{k+1} = \rho\mu^k$
end while
```

It can be shown that, under mild conditions on the sequence μ^k, the algorithm above also generates sequences $\mathbf{A}^k$ and $\mathbf{E}^k$ that converge to the solution of Problem 1, although in this case the rate of convergence is hard to ascertain. Nevertheless, consistent numerical experience shows that this algorithm has good convergence properties.

Experiment	SRPCA	SDP
Prediction (fps)	3.7	2.1
Tracklet Matching (secs per pair)	1.23	650
Outlier Cleaning (secs per track)	25	N/A

Table 1. Comparison of the Running Times of SRPCA vs. SDP for three different applications

4. Applications

In this section we illustrate the advantages of the proposed method using three different applications: (i) robustly predicting future positions of a target, (ii) tracklet matching across occlusion, and (iii) outlier detection and removal from long trajectories. Table 1 compares the performance of SRPCA against previous approaches based upon recasting these problems into a Semi-Definite optimization form and using conventional SDP solvers. These experiments were performed on a Dual Core 2.53GHz, 24GB RAM computer. SRPCA was implemented in Matlab, with $\rho = 1.05$, $\delta = 1e-2$, $\mu_0 = 1e-2$, $\mathbf{S}_A = \mathbf{S}_E = \mathbf{S}_{\text{Hankel}}$, and $\mathbf{W}_F = 0$ (since in the applications of interest it suffices to minimize a combination of the nuclear and ℓ^1 norms). The semi-definite programs were solved using the cvx package in conjunction with the sedumi SDP solver. As shown in the table, in all cases SRPCA resulted in a substantial reduction of the computational time. Further, the memory requirements of the SDP solver prevented its use in the outlier removal example, even on a 24GB machine. The videos of the examples are available in the supplementary material.

4.1. Target Location Prediction

As shown in [5], future positions of a moving target can be predicted by solving a rank minimization problem. Specifically, if past measurements $y_1, y_2, ...y_t$ are available, it can be shown that, under mild conditions, the next position y_{t+1} satisfies:

$$y_{t+1} = \underset{y}{argmin}\, rank \mathbf{H}_y$$

where

$$\mathbf{H}_y \doteq \begin{bmatrix} y_1 & y_2 & y_3 & .. & y_m \\ y_2 & y_3 & .. & .. & y_{m+1} \\ y_3 & y_4 & \ddots & \vdots & \vdots \\ \vdots & \vdots & .. & .. & \vdots \\ y_{r-1} & y_r & .. & y_{t-2} & y_{t-1} \\ y_r & y_{r+1} & .. & y_{t-1} & y \end{bmatrix}$$

In the case of measurements corrupted by bounded noise, $\hat{y}_k = y_k + e_k$, with $\|e\| \leq \eta$, y_t can be predicted by solving the following optimization problem:

$$y_{t+1} = \underset{y,e}{argmin}\, rank\ \mathbf{H}_{\hat{y}} + \mathbf{H}_e \text{ s.t.} \|e\| \leq \eta$$

where $\mathbf{H}_{\hat{y}}$ and $\mathbf{H}_e$ denote the Hankel matrices associated with the sequences $\hat{y}$ and $\hat{e}$. Since minimizing rank is NP-hard, [5] proposed to solve this problem using a re-weighted nuclear norm heuristics that, at each step seeks to minimize $\sum_i w_i \sigma_i (\mathbf{H}_{\hat{y}} + \mathbf{H}_e)$, subject to $\|e\| \leq \eta$. While this problem does not exactly fit the SRPCA formalism, it can be modified to do so by handling the constraint on the norm via a penalty function, leading to:

$$y_{t+1} = \underset{y,e}{argmin} \sum_i w_i \sigma_i (\mathbf{H}_{\hat{y}} + \mathbf{H}_e) + \|\mathbf{W}_e \circ \mathbf{H}_e\|$$

The effectiveness of using a combination of an SRPCA based predictor and a particle filter to achieve sustained tracking in the presence of occlusion is illustrated in Fig. 2. Here the goal was to track the ball when visible and predict its location when occluded or coming out of an occlusion. As shown there, the combination SRPCA and particle filter successfully accomplished this task, even though the player with the ball is undergoing complex motions, the camera is panning and the ball is occluded by a second player for about 6 frames. In this particular example $\mathbf{W}_e$ was chosen to be

$$\mathbf{m} = [\mathbf{1}_{N_h}, \mathbf{0}_{N_o+1}], w_n(j) = 1/(\sum_{i=1}^{N}, |S(i,j)|)$$
$$\mathbf{w_e} = \mathbf{w}_n \circ \mathbf{m}, \mathbf{vec}(\mathbf{W_e}) = \mathbf{S}\mathbf{w_e}$$

where N_o is an estimate of the duration of the occlusions to be handled and $\mathbf{1}_N$ and $\mathbf{0}_N$ denote vectors of dimension N with elements all ones or zeros, respectively. The purpose of $\mathbf{w}_n$ is to cancel the repetitions that might occur in some elements of $\mathbf{E}$ matrix due to the structure $\mathbf{S}$

4.2. Tracklet Matching

In this section we consider the problem of tracklet matching, that is, establishing the identities of multiple targets across occlusion. As shown in [4], this problem can also be reduced to a Hankel rank minimization. Briefly, given a pair of tracklets b^i and a^j (before and after the occlusion, respectively), the idea is to attempt to connect them by finding the missing pixels via rank minimization. To illustrate this point with a simple example, assume that each tracklet has 2 points and the gap between tracklets is two frames. Then, the missing pixels can be found by solving:

$$y_{1:2} = \underset{y}{argmin}\, rank \mathbf{H}_{b,y,a}\ \mathbf{H}_{b,y,a} \doteq \begin{bmatrix} b_1 & b_2 & y_1 & y_2 \\ b_2 & y_1 & y_2 & a_1 \\ y_1 & y_2 & a_1 & a_2 \end{bmatrix}$$

This procedure induces a *similarity measure* between tracklets $s(b^i, a^j) \doteq \min_y rank \mathbf{H}_{b,y,a}$ that can be used to pair-

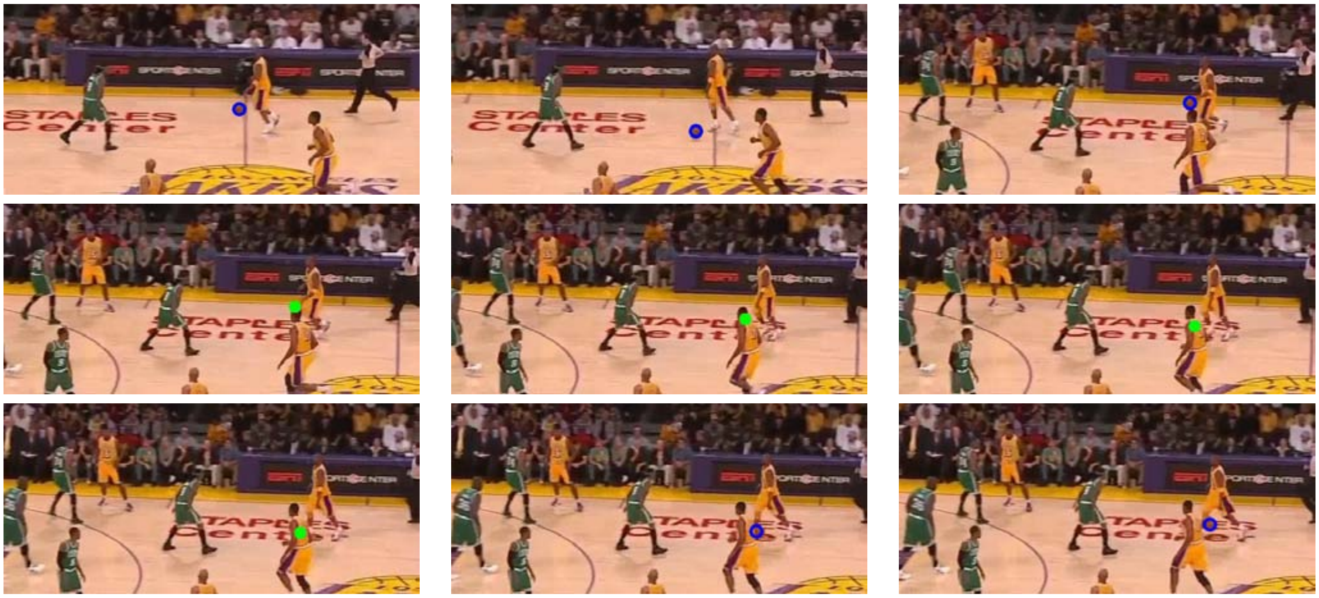


Figure 2. Using SRPCA to predict target locations under occlusion, Top right image is the ball last seen by the tracker and the bottom middle image is the recovered tracker image

wise group them. Note that in principle this approach requires solving N_bN_a rank minimization problems, where N_b and N_a denote the number of tracklets before and after the occlusion, respectively. Thus, it is of interest to develop fast algorithms for computing this similarity measure. Relaxing the rank minimization to a nuclear norm and taking into account the possible existence of outliers leads to a SRPCA problem, where all the matrices involved are constrained to have a Hankel structure. The complete algorithm is outlined below:

Algorithm 4.2: TRACKLET MATCHING WITH RSPCA

```
for each tracklet pair one ($t_1(t)$ and $t_2(t)$)
1. Let the lengths of $t_1(t)$ and $t_2(t)$ be
$N_1$ and $N_2$ separated by $N_o$ number of occluded frames
2. Construct the $\mathbf{W_e}$ as follows
$\mathbf{m} = [\mathbf{1}_{N_1}, \mathbf{0}_{N_o}\mathbf{1}_{N_2}], w_n(j) = 1/(\sum_{i=1}^{N}, |S(i,j)|)$
$\mathbf{w_e} = \mathbf{w}_n \circ \mathbf{m}, \mathbf{vect}(\mathbf{W_e}) = \mathbf{Sw_e}$
3. Construct $\mathbf{vect}(\mathbf{D}) = \mathbf{S}[t_1(.), \mathbf{0}_{N_o}, t_2(.)]^T$
4. Solve SRPCA with the defined variables,
(Here $\mathbf{a}$=combined trajectory is the output of the SRPCA)
5. s($t_1(.)$,$t_2(.)$)=$\sum_i w_i\sigma_i(mat(\mathbf{Sa}))$
end for
6. Assign closest tracklets to each
other using Distance(.,.)
```

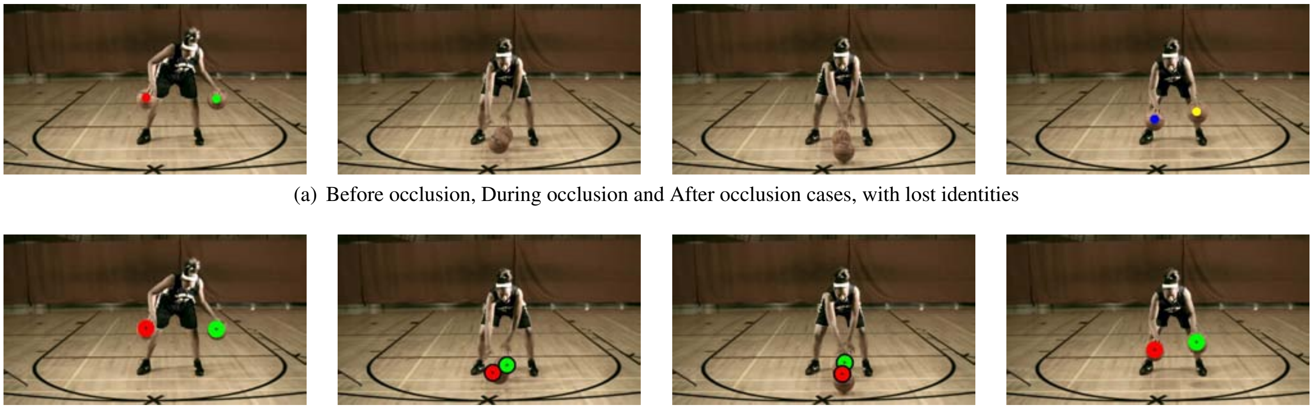
The effectiveness of this approach is illustrated in Fig. 3 where it was used to consistently label two basketballs across occlusion. Note that this problem is far from trivial due to the fact that the targets have similar appearance and bounce nearly at the same point, and are partially occluded for about 17 frames. In this case, we used SRPCA to compute a similarity measure by (approximately) minimizing the rank of the Hankel matrix with respect to the 17 missing measurements. Note that computing this measure using SRPCA only took 1.2 seconds per pair, compared to 650 seconds when using an SDP solver.

4.3. Outlier removal from long trajectories

Next, we illustrate the use of SRPCA to detect and clean outliers. Conceptually, the idea is similar to that in section 4.1: An outlier is characterized by the fact that it does not match the "dynamics" of its neighboring points and thus causes a substantial increase in the rank of the corresponding Hankel matrix. As before, relaxing rank to nuclear norm leads directly to an SRPCA type problem of the form

$$\begin{aligned}&\min \|\mathbf{A}\|_* + \|\mathbf{W_e} \circ \mathbf{E}\|_1 \text{ s. t.}\\&\mathbf{D} = \mathbf{A} + \mathbf{E}\\&\mathbf{A} \text{ and } \mathbf{E} \text{ are Hankel}\end{aligned}$$

where $\mathbf{D}$ is the Hankel matrix of the measured trajectories. Note than in here, the non-zero elements of the error matrix $\mathbf{E}$ correspond precisely to the outlier locations, while the resulting matrix $\mathbf{A}$ contains the "cleaned" trajectories.



(a) Before occlusion, During occlusion and After occlusion cases, with lost identities

(b) Labels provided by SRPCA: Before occlusion, Prediction during occlusion and After occlusion cases. A black edge around a solid color indicates a predicted location

Figure 3. Using SRPCA to maintain consistent labeling across occlusion.

Algorithm 4.3: OUTLIER CLEANING WITH SRPCA

```
for each corrupted trajectory $t(t)$
  1. Construct the $\mathbf{W_e}$ as follows
  $w_n(j) = 1/(\sum_{i=1}^{N}, |S(i,j)|)$, $\mathbf{vect}(\mathbf{W_e}) = \mathbf{Sw_e}$
  2. Construct $\mathbf{vect}(\mathbf{D}) = \mathbf{S}[t(t)]^T$
  3. Solve SRPCA with the defined variables
  (Here a=clean trajectory is the output of the SRPCA)
end for
```

Fig. 4 shows the results of applying the algorithm outlined above to remove outliers from trajectories that are 250 frames long, manually corrupted with outliers added at random locations with probability 0.2. As illustrated there, SRPCA was able to recover the original trajectories in about 25 seconds. On the other hand, this example could not be solved using a standard SDP solver in a computer with 24GB of RAM due to insufficient memory.

5. Conclusions

A large number of problems arising in computer vision involve minimizing a combination of the sum of the nuclear, ℓ_1 and Frobenious norms of matrices, subject to additional structural constraints. Examples of relevant applications include, among others, robust tracking in the presence of outliers, manifold embedding, event detection, inpainting and tracklet matching across occlusion. Unfortunately, the existence of structural constraints prevents the use of very efficient algorithms recently developed to solve the unconstrained case, while the use of general semi-definite optimization solvers is limited to relatively small problems, due to their poor scaling properties. The main result of this paper shows that structured nuclear norm minimization problems can be efficiently solved by using an iterative Augmented Lagrangian Type (ALM) method that only requires performing at each iteration a combination of matrix thresholding and matrix inversion steps. These results were illustrated with several examples where the proposed algorithm resulted in a substantial reduction of computational time and memory requirements when compared against interior-point methods. Research is currently underway seeking to extend these results to handle inequality and semi-definite constraints.

References

[1] D. Bertsekas. *Constrained Optimization and Lagrange Multiplier Method*. Academic Press, 1982. 3

[2] E. Candes, X. Li, Y. Ma, and J.Wright. Robust principal component analysis? *J. ACM*, (3), 2011. 1

[3] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *Siam J. Optim.*, (2):572–596, 2011. 1

[4] T. Ding, M. Sznaier, and O. Camps. Fast track matching and event detection. In *IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, June 2008. 5

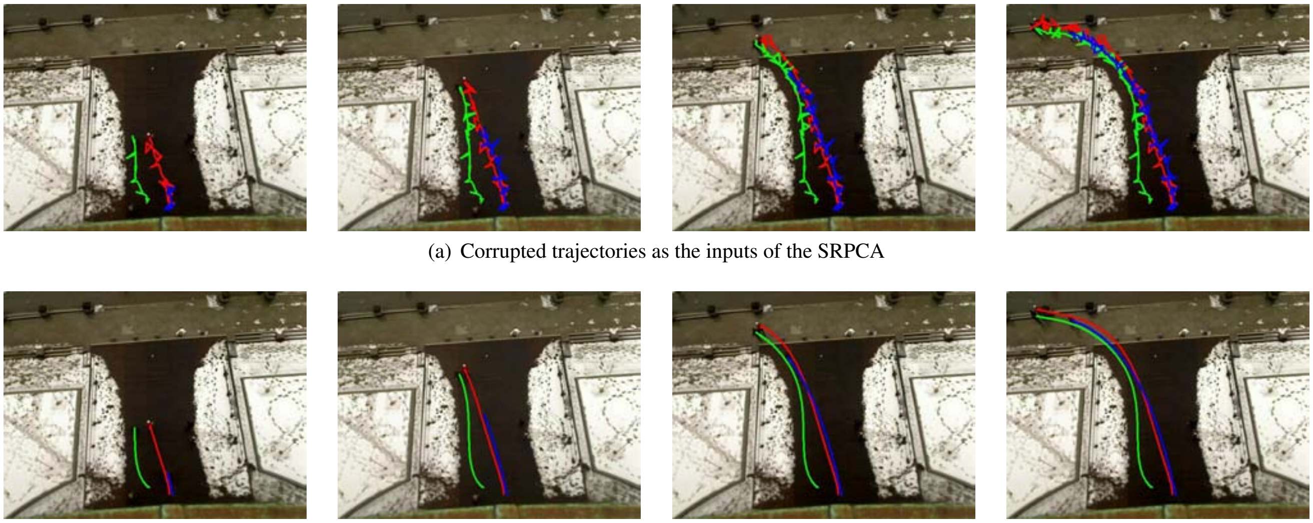
[5] T. Ding, M. Sznaier, and O. Camps. Receding horizon rank minimization based estimation with applications to visual tracking. In *Proc. 47th IEEE Conf. Dec. Control (CDC)*, pages 3446–3451, Dec 2008. 5

[6] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, pages 1801–1807, 2011. 1

[7] J. Huang, X. Huang, and D. Metaxas. Learning with dynamic group sparsity. In *IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, 2009. 1

[8] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, pages 1791–1798, 2010. 1

[9] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted



(a) Corrupted trajectories as the inputs of the SRPCA

(b) Clean trajectories, output of the SRPCA

Figure 4. Using SRPCA to clean long time trajectories.

low-rank matrices. Technical Report UILU-ENG-09-2215, arXiv:1009.5055v2, UIUC, 2009. 1, 3, 4, 8

[10] Y. Peng, A. Ganesh, J. Wright, and Y. Ma. Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, 2010. 1

[11] M. Sznaier and O. Camps. A hankel operator approach to texture modelling and inpainting. In *Texture 2005: Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, pages 125–130, 2005. 2

[12] J. Xu. Reweighted nuclear norm minimization for matrix completion. preprint, 2011. 4

A. Proof of Theorem 1

Since $\|\mathbf{X}\|_{*,\mathbf{W}} \doteq \sum w_i\sigma_i(\mathbf{X})$ and $\|\mathbf{X}\|_{1,\mathbf{W}} \doteq \|\mathbf{W}\circ\mathbf{X}\|_1$ are matrix norms, direct application of Lemma 1 in [9] shows that the sequences $\mathbf{Y}_1^k$ and $\mathbf{Y}_3^k$ are bounded. To show boundedness of $\mathbf{Y}_2$, note that L is differentiable with respect to $\mathbf{a}$ and $\mathbf{e}$. Thus, from (9) it follows, using the explicit expressions for the updates of $\mathbf{Y}_i$, that

$$0 = \mathbf{S}_A^T\mathbf{vec}(\mathbf{Y}_1)^{k+1} + \mathbf{S}_A^T\mathbf{S}_A\mathbf{S}_A^T\mathbf{vec}(\mathbf{Y}_2^{k+1})$$

Since $\mathbf{S}_A$ has full column rank and $\mathbf{Y_1}$ is bounded, it follows that $\mathbf{S}_A^T\mathbf{vec}(\mathbf{Y}_2)$ is bounded. Next, note that

$$\begin{aligned}&\langle \mathbf{S}_A^T\mathbf{vec}(\mathbf{Y}_2), \mathbf{S_A^T}\mathbf{vec}(\mathbf{D}) - \mathbf{S_A^T}\mathbf{S_A}\mathbf{a} - \mathbf{S_A^T}\mathbf{S_E}\mathbf{e}\rangle\\ &+\tfrac{\mu}{2}\left\|\mathbf{S_A^T}\mathbf{vec}(\mathbf{D}) - \mathbf{S_A^T}\mathbf{S_A}\mathbf{a} - \mathbf{S_A^T}\mathbf{S_E}\mathbf{e}\right\|_2^2\\ &= \tfrac{1}{2\mu^k}\left[\|\mathbf{S_A^T}\mathbf{Y}_2^{k+1}\|_2^2 - \|\mathbf{S_A^T}\mathbf{Y}_2^k\|_2^2\right]\end{aligned}$$

Similarly,

$$\begin{aligned}\langle \mathbf{Y}_1, \mathbf{J} - \mathbf{mat}(\mathbf{S}_A\mathbf{a})\rangle + (\mu/2)(\|\mathbf{J} - \mathbf{mat}(\mathbf{S}_A\mathbf{a}))\|_F^2\\ = \tfrac{1}{2\mu^k}\left[\|\mathbf{Y}_1^{k+1}\|_2^2 - \|\mathbf{Y}_1^k\|_2^2\right]\end{aligned}$$

and

$$\begin{aligned}&\langle \mathbf{Y}_3, \mathbf{T} - \mathbf{mat}(\mathbf{S}_E\mathbf{e})\rangle + (\mu/2)(\|\mathbf{T} - \mathbf{mat}(\mathbf{S}_E\mathbf{e})\|_F^2)\\ &= \tfrac{1}{2\mu^k}\left[\|\mathbf{Y}_3^{k+1}\|_2^2 - \|\mathbf{Y}_3^k\|_2^2\right]\end{aligned}$$

Thus, from the boundedness of $\mathbf{Y}_1, \mathbf{S^T}\mathbf{Y}_2$ and $\mathbf{Y}_3$ it follows that as $\mu^k \rightarrow \infty$ then $L(\mathbf{X}^k, \mathbf{Y}^k) \rightarrow \sum_i w_i\sigma_i(\mathbf{J}) + \|\mathbf{W}_e\circ\mathbf{T}\|_1 + \frac{1}{2}\|\mathbf{W}_F\circ\mathbf{E}\|_F^2$ as $\mathcal{O}(\frac{1}{\mu^k})$. Denote by $\mathbf{A}^*$ and $\mathbf{E}^*$ the optimal solution to (8), and by L^* the corresponding value of the objective. Since $\mathbf{J} = \mathbf{A}^*$ and $\mathbf{T} = \mathbf{E}^*$ is a feasible solution to the optimization problem solved in the first step of Algorithm 3.1, it follows that $L(\mathbf{J}^k, \mathbf{a}^k, \mathbf{T}^k, \mathbf{e}^k) \leq L^*$. Hence

$$\begin{aligned}&\textstyle\sum_i w_i\sigma_i(\mathbf{J}^k) + \left\|\mathbf{W}_e\circ\mathbf{T}^k\right\|_1 + \frac{1}{2}\left\|\mathbf{W}_F\circ\mathbf{E^k}\right\|_F^2\\ &\leq L^* + \mathrm{O}(\tfrac{1}{\mu^k})\end{aligned}$$

From the boundedness of $\mathbf{Y}_1$ and $\mathbf{Y}_3$ and the update equations, it follows that $\|\mathbf{J^k} - \mathbf{A^k}\|$ and $\|\mathbf{T^k} - \mathbf{E^k}\| \rightarrow 0$ as $\mathcal{O}(\mu^{-k})$. Finally, from the boundedness of $\mathbf{S^T}\mathbf{vec}(\mathbf{Y}_2)$ it follows that $\|\mathbf{S_A^T}\mathbf{vec}(\mathbf{D}) - \mathbf{S_A^T}\mathbf{S_A}\mathbf{a^k} - \mathbf{S_A^T}\mathbf{S_E}\mathbf{e^k}\|_2 \rightarrow 0$. Hence, the sequences $\mathbf{a}^k$ and $\mathbf{e}^k$ are bounded, and thus have accumulation points $\mathbf{a}^*$, $\mathbf{e}^*$. Let $\mathbf{A}^* \doteq \mathbf{S}_A\mathbf{a}^*$ and $\mathbf{E}^* \doteq \mathbf{S}_E\mathbf{e}^*$ If the structural constraints are feasible (in the sense that there exist vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ such that $\mathbf{D} = \mathbf{S}_A\mathbf{d}_1 + \mathbf{S}_E\mathbf{d}_e$), then this last equation implies that $\mathbf{A}^* + \mathbf{E}^* \rightarrow \mathbf{D}$. Combining the arguments above with continuity of the norms, it follows that the matrices $\mathbf{A}^*$ and $\mathbf{E}^*$, which satisfy the structural constraints by construction, are such that $\sum_i w_i\sigma_i(\mathbf{A}^*) + \|\mathbf{W}_e\circ\mathbf{E}^*\|_1 + \frac{1}{2}\|\mathbf{W}_F\circ\mathbf{E}^*\|_F^2 = L^*$.