Practical Matrix Completion and Corruption Recovery using Proximal Alternating Robust Subspace Minimization

Yu-Xiang Wang · Choon Meng Lee · Loong-Fah Cheong · Kim-Chuan Toh

arXiv:1309.1539v2 [cs.CV] 28 Oct 2014

Abstract Low-rank matrix completion is a problem of immense practical importance. Recent works on the subject often use nuclear norm as a convex surrogate of the rank function. Despite its solid theoretical foundation, the convex version of the problem often fails to work satisfactorily in real-life applications. Real data often suffer from very few observations, with support not meeting the randomness requirements, ubiquitous presence of noise and potentially gross corruptions, sometimes with these simultaneously occurring.

This paper proposes a Proximal Alternating Robust Subspace Minimization (PARSuMi) method to tackle the three problems. The proximal alternating scheme explicitly exploits the rank constraint on the completed matrix and uses the ℓ_0 pseudo-norm directly in the corruption recovery step. We show that the proposed method for the non-convex and non-smooth model converges to a stationary point. Although it is not guaranteed to find the global optimal solution, in practice we find that our algorithm can typically arrive at a good local minimizer when it is supplied with a reasonably good starting point based on convex optimization. Extensive experiments with challenging synthetic and real data demonstrate that our algorithm succeeds in a much larger range of practical problems where convex optimization fails, and it also outperforms various state-of-the-art algorithms.

Keywords matrix completion · matrix factorization · RPCA · robust · low-rank · sparse · nuclear norm · non-convex optimization · SfM · photometric stereo

Y.X. Wang, C.M. Lee, L.F. Cheong, K.C. Toh
National University of Singapore
E-mail: {yuxiangwang, leechoonmeng, eleclf, mattohkc} @nus.edu.sg

1 Introduction

Completing a low-rank matrix from partially observed entries, also known as matrix completion, is a central task in many real-life applications. The same abstraction of this problem has appeared in diverse fields such as signal processing, communications, information retrieval, machine learning and computer vision. For instance, the missing data to be filled in may correspond to plausible movie recommendations (Koren et al 2009; Funk 2006), occluded feature trajectories for rigid or non-rigid structure from motion, namely SfM (Hartley and Schaffalitzky 2003; Buchanan and Fitzgibbon 2005) and NRSfM (Paladini et al 2009), relative distances of wireless sensors (Oh et al 2010), pieces of uncollected measurements in DNA micro-array (Friedland et al 2006), just to name a few.
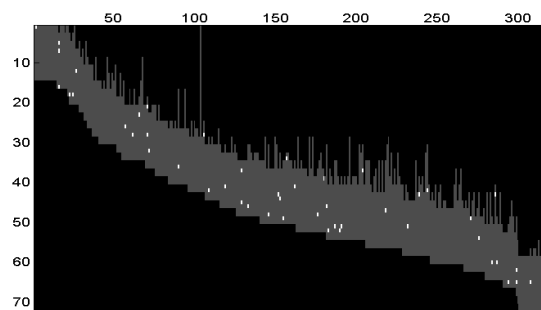

Fig. 1 Sampling pattern of the Dinosaur sequence: 316 features are tracked over 36 frames. Dark area represents locations where no data is available; sparse highlights are injected gross corruptions. Middle stripe in grey are noisy observed data, occupying 23% of the full matrix. The task of this paper is to fill in the missing data and recover the corruptions.

The common difficulty of these applications lies in the scarcity of the observed data, uneven distribution of the support, noise, and more often than not, the presence of gross corruptions in some observed entries. For instance, in the movie rating database Netflix (Bennett et al 2007), only less than 1% of the entries are observed and 90% of the observed entries correspond to 10% of the most popular movies. In photometric stereo, the missing data and corruptions (arising from shadow and specular highlight as modeled in Wu et al (2011b)) form contiguous blocks in images and are by no means random. In structure from motion, the observations fall into a diagonal band shape, and feature coordinates are often contaminated by tracking errors (see the illustration in Fig. 1). Therefore, in order for any matrix completion algorithm to work in practice, these aforementioned difficulties need to be tackled altogether. We refer to this problem as **practical matrix completion**. Mathematically, the problem to be solved is the following:

Given	$\Omega, \widehat{W}_{ij}$ for all $(i,j) \in \Omega,$
find	$W, \tilde{\Omega},$
s.t.	$\mathrm{rank}(W)$ is small; $\mathrm{card}(\tilde{\Omega})$ is small;
	$\lvert W_{ij} - \widehat{W}_{ij}\rvert$ is small $\forall (i,j) \in \Omega / \tilde{\Omega}.$

where Ω is the index set of observed entries whose locations are not necessarily selected at random, $\tilde{\Omega} \in \Omega$ represents the index set of corrupted data, $\widehat{W} \in \mathbb{R}^{m \times n}$ is the measurement matrix with only $\widehat{W}_{ij \in \Omega}$ known, i.e., its support is contained in Ω. Furthermore, we define the projection $\mathcal{P}_\Omega : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{|\Omega|}$ so that $\mathcal{P}_\Omega(\widehat{W})$ denotes the vector of observed data. The adjoint of $\mathcal{P}_\Omega$ is denoted by $\mathcal{P}_\Omega^*$.

Extensive theories and algorithms have been developed to tackle some aspect of the challenges listed in the preceding paragraph, but those tackling the full set of challenges are far and few between, thus resulting in a dearth of practical algorithms. Two dominant classes of approaches are nuclear norm minimization, e.g. Candès and Recht (2009); Candès et al (2011); Candès and Plan (2010); Chen et al (2011), and matrix factorization, e.g., Koren et al (2009); Buchanan and Fitzgibbon (2005); Okatani and Deguchi (2007); Chen (2008); Eriksson and Van Den Hengel (2010). Nuclear norm minimization methods minimize the convex relaxation of rank instead of the rank itself, and are supported by rigorous theoretical analysis and efficient numerical computation. However, the conditions under which they succeed are often too restrictive for it to work well in real-life applications (as reported in Shi and Yu (2011)

and Jain et al (2012)). In contrast, matrix factorization is widely used in practice and are considered very effective for problems such as movie recommendation (Koren et al 2009) and structure from motion (Tomasi and Kanade 1992; Paladini et al 2009) despite its lack of rigorous theoretical foundation. Indeed, as one factorizes matrix W into UV^T, the formulation becomes bilinear and thus optimal solution is hard to obtain except in very specific cases (e.g., in Jain et al (2012)). A more comprehensive survey of the algorithms and review of the strengths and weaknesses will be given in the next section.

In this paper, we attempt to solve the practical matrix completion problem under the prevalent case where the rank of the matrix W and the cardinality of $\tilde{\Omega}$ are upper bounded by some known parameters r and N_0 via the following non-convex, non-smooth optimization model:

$$\begin{aligned} \min_{W,E} \quad & \tfrac{1}{2}\|\mathcal{P}_\Omega(W - \widehat{W} + E)\|^2 + \tfrac{\epsilon}{2}\|\mathcal{P}_{\overline{\Omega}}(W)\|^2 \\ \text{s.t.} \quad & \mathrm{rank}(W) \le r, \ W \in \mathbb{R}^{m \times n} \\ & \|E\|_0 \le N_0, \ \|E\| \le K_E, \ E \in \mathbb{R}_\Omega^{m \times n} \end{aligned} \tag{1}$$

where $\mathbb{R}_\Omega^{m \times n}$ denotes the set of $m \times n$ matrices whose supports are subsets of Ω and $\|\cdot\|$ is the Frobenius norm; K_E is a finite constant introduced to facilitate the convergence proof. Note that the restriction of E to $\mathbb{R}_\Omega^{m \times n}$ is natural since the role of E is to capture the gross corruptions in the observed data $\widehat{W}_{ij \in \Omega}$. The bound constraint on $\|E\|$ is natural in some problems when the true matrix W is bounded (e.g., Given the typical movie ratings of 0-10, the gross outliers can only lie in [-10, 10]). In other problems, we simply choose K_E to be some large multiple (say 20) of $\sqrt{N_0} \times \mathrm{median}(\mathcal{P}_\Omega(\widehat{W}))$, so that the constraint is essentially inactive and has no impact on the optimization. Note that without making any randomness assumption on the index set Ω or assuming that the problem has a unique solution (W^*, E^*) such that the singular vector matrices of W^* satisfy some inherent conditions like those in Candès et al (2011), the problem of practical matrix completion is generally ill-posed. This motivated us to include the Tikhonov regularization term $\tfrac{\epsilon}{2}\|\mathcal{P}_{\overline{\Omega}}(W)\|^2$ in (1), where $\overline{\Omega}$ denotes the complement of Ω, and $0 < \epsilon < 1$ is a small constant. Roughly speaking, what the regularization term does is to pick the solution W which has the smallest $\|\mathcal{P}_{\overline{\Omega}}(W)\|$ among all the candidates in the optimal solution set of the non-regularized problem. Notice that we only put a regularization on those elements of W in $\overline{\Omega}$ as we do not wish to perturb those elements of W in the fitting term. Finally, with the Tikhonov regularization and the bound constraint on

$\|E\|$, we can show that problem (1) has a global minimizer.

By defining $H \in \mathbb{R}^{m \times n}$ to be the matrix such that

$$H_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ \sqrt{\epsilon} & \text{if } (i,j) \notin \Omega, \end{cases} \quad (2)$$

and those elements of E and $\widehat{W}$ in $\overline{\Omega}$ to be zero, we can rewrite the objective function in (1) in a compact form, and the problem becomes:

$$\begin{aligned} \min_{W,E} \quad & \tfrac{1}{2}\|H \circ (W + E - \widehat{W})\|^2 \\ \text{s.t.} \quad & \text{rank}(W) \le r, \; W \in \mathbb{R}^{m \times n} \\ & \|E\|_0 \le N_0, \; \|E\| \le K_E, \; E \in \mathbb{R}_\Omega^{m \times n}. \end{aligned} \quad (3)$$

In the above, the notation "$\circ$" denotes the element-wise product between two matrices.

We propose PARSuMi, a proximal alternating minimization algorithm motivated by the algorithm in Attouch et al (2010) to solve (3). This involves solving two subproblems each with an auxiliary proximal regularization term. It is important to emphasize that the subproblems in our case are non-convex and hence it is essential to design appropriate algorithms to solve the subproblems to global optimality, at least empirically. We develop essential reformulations of the subproblems and design novel techniques to efficiently solve each subproblem, provably achieving the global optimum for one, and empirically so for the other. We also prove that our algorithm is guaranteed to converge to a limit point, which is necessarily a stationary point of (3). We emphasize here that the convergence is established even though one of the subproblems may not be solved to global optimality. Together with the initialization schemes we have designed based on the convex relaxation of (3), our method is able to solve challenging real matrix completion problems with corruptions robustly and accurately. As we demonstrate in the experiments, PARSuMi is able to provide excellent reconstruction of unobserved feature trajectories in the classic Oxford Dinosaur sequence for SfM, despite structured (as opposed to random) observation pattern and data corruptions. It is also able to solve photometric stereo to high precision despite severe violations of the Lambertian model (which underlies the rank-3 factorization) due to shadow, highlight and facial expression difference. Compared to state-of-the-art methods such as GRASTA (He et al 2011), Wiberg ℓ_1 (Eriksson and Van Den Hengel 2010) and BALM (Del Bue et al 2012), our results are substantially better both qualitatively and quantitatively.

Note that in (3) we do not seek convex relaxation of any form, but rather constrain the rank and the corrupted entries' cardinality directly in their original forms. While it is generally not possible to have an algorithm guaranteed to compute the global optimal solution, we demonstrate that with appropriate initializations, the faithful representation of the original problem often offers significant advantage over the convex relaxation approach in denoising and corruption recovery, and is thus more successful in solving real problems.

The rest of the paper is organized as follows. In Section 2, we provide a comprehensive review of the existing theories and algorithms for practical matrix completion, summarizing the strengths and weaknesses of nuclear norm minimization and matrix factorization. In Section 3, we conduct numerical evaluations of predominant matrix factorization methods, revealing those algorithms that are less-likely to be trapped at local minima. Specifically, these features include parameterization on a subspace and second-order Newton-like iterations. Building upon these findings, we develop the PARSuMi scheme in Section 4 to simultaneously handle sparse corruptions, dense noise and missing data. The proof of convergence and a convex initialization scheme are also provided in this section. In Section 5, the proposed method is evaluated on both synthetic and real data and is shown to outperform the current state-of-the-art algorithms for robust matrix completion.

2 A survey of results

2.1 Matrix completion and corruption recovery via nuclear norm minimization

Recently, the most prominent approach for solving a matrix completion problem is via the following nuclear norm minimization:

$$\min_W \left\{ \|W\|_* \;\middle|\; \mathcal{P}_\Omega(W - \widehat{W}) = 0 \right\}, \quad (4)$$

in which $\text{rank}(X)$ is replaced by the nuclear norm $\|X\|_* = \sum_i \sigma_i(X)$, where the latter is the tightest convex relaxation of rank over the unit (spectral norm) ball. Candès and Recht (2009) showed that when sampling is uniformly random and sufficiently dense, and the underlying low-rank subspace is *incoherent* with respect to the standard bases, then the remaining entries of the matrix can be exactly recovered. The guarantee was later improved in Candès and Tao (2010); Recht (2009), and extended for noisy data in Candès and Plan (2010); Negahban and Wainwright (2012) relaxed the equality constraint to

$$\|\mathcal{P}_\Omega(W - \widehat{W})\| \le \delta.$$

Using similar assumptions and arguments, Candès et al (2011) and Chandrasekaran et al (2011) concurrently

	MC (Candès and Recht 2009)	RPCA (Candès et al 2011)	NoisyMC (Candès and Plan 2010)	StableRPCA (Zhou et al 2010)	RMC (Li 2013)	RMC (Chen et al 2011)
Missing data	Yes	Yes	Yes	No	Yes	Yes
Corruptions	No	Yes	No	Yes	Yes	Yes
Noise	No	No	Yes	Yes	No	No
Deterministic Ω	No	No	No	No	No	Yes
Deterministic $\tilde{\Omega}$	No	No	No	No	No	Yes

Table 1 Summary of the theoretical development for matrix completion and corruption recovery.

proposed solution to the related problem of robust principal component analysis (RPCA) where the low-rank matrix can be recovered from sparse corruptions (with no missing data[1]). This is formulated as

$$\min_{W,E} \left\{ \|W\|_* + \lambda\|E\|_1 \;\middle|\; W + E = \widehat{W} \right\}. \qquad (5)$$

Noisy extension and improvement of the guarantee for RPCA were provided by Zhou et al (2010) and Ganesh et al (2010) respectively. Chen et al (2011) and Li (2013) combined (4) and (5) and provided guarantee for the following

$$\min_{W,E} \left\{ \|W\|_* + \lambda\|E\|_1 \;\middle|\; \mathcal{P}_\Omega(W + E - \widehat{W}) = 0 \right\}. \qquad (6)$$

In particular, the results in Chen et al (2011) lifted the uniform random support assumptions in previous works by laying out the exact recovery condition for a class of deterministic sampling (Ω) and corruptions ($\tilde{\Omega}$) patterns.

We summarize the theoretical and algorithmic progress in practical matrix completion achieved by each method in Table 1. It appears that researchers are moving towards analyzing all possible combinations of the problems; from past indication, it seems entirely plausible albeit tedious to show that the noisy extension

$$\min_{W,E} \left\{ \|W\|_* + \lambda\|E\|_1 \;\middle|\; \|\mathcal{P}_\Omega(W + E - \widehat{W})\| \le \delta \right\} \qquad (7)$$

will return a solution stable around the desired W and E under appropriate assumptions. Wouldn't that solve the practical matrix completion problem altogether?

The answer is unfortunately no. While this line of research have provided profound understanding of practical matrix completion itself, the actual performance of the convex surrogate on real problems (e.g., movie recommendation) is usually not competitive against nonconvex approaches such as matrix factorization. Although convex relaxation is amazingly equivalent to the original problem under certain conditions, those well versed

in practical problems will know that those theoretical conditions are usually not satisfied by real data. Due to noise and model errors, real data are seldom truly low-rank (see the comments on Jester joke dataset in Keshavan et al (2009)), nor are they as incoherent as randomly generated data. More importantly, observations are often structured (e.g., diagonal band shape in SfM) and hence do not satisfy the random sampling assumption needed for the tight convex relaxation approach. As a consequence of all these factors, the recovered W and E by convex optimization are often neither low-rank nor sparse in practical matrix completion. This can be further explained by the so-called "Robin Hood" attribute of ℓ_1 norm (analogously, nuclear norm is the ℓ_1 norm in the spectral domain), that is, it tends to steal from the rich and give it to the poor, decreasing the inequity of "wealth" distribution. Illustrations of the attribute will be given in Section 5.

Nevertheless, the convex relaxation approach has the advantage that one can design *efficient* algorithms to find or approximately reach the *global* optimal solution of the given convex formulation. In this paper, we take advantage of the convex relaxation approach and use it to provide a powerful initialization for our algorithm to converge to the correct solution.

2.2 Matrix factorization and applications

Another widely-used method to estimate missing data in a low-rank matrix is matrix factorization (MF). It is at first considered as a special case of the weighted low-rank approximation problem with $\{0, 1\}$ weight by Gabriel and Zamir in 1979 and much later by Srebro and Jaakkola (2003). The buzz of Netflix Prize further popularizes the missing data problem as a standalone topic of research. Matrix factorization turns out to be a robust and efficient realization of the idea that people's preferences of movies are influenced by a small number of latent factors and has been used as a key component in almost all top-performing recommendation systems (Koren et al 2009) including BellKor's Pragmatic Chaos, the winner of the Netflix Prize (Koren 2009).

---

[1] Candès et al (2011) actually considered missing data too, but their guarantee (Theorem 1.2) for (6) is only preliminary according to their own remarks. A stronger result is released later by the same group in Li (2013).

In computer vision, matrix factorization with missing data is recognized as an important problem too. Tomasi-Kanade affine factorization (Tomasi and Kanade 1992), Sturm-Triggs projective factorization (Sturm and Triggs 1996), and many techniques in Non-Rigid SfM and motion tracking (Paladini et al 2009) can all be formulated as a matrix factorization problem. Missing data and corruptions emerge naturally due to occlusions and tracking errors. For a more exhaustive survey of computer vision problems that can be modelled by matrix factorization, we refer readers to Del Bue et al (2012).

Regardless of its applications, the key idea is that when $W = UV^T$, one ensures that the required rank constraint is satisfied by restricting the factors U and V to be in $\mathbb{R}^{m \times r}$ and $\mathbb{R}^{n \times r}$ respectively. Since the (U, V) parameterization has a much smaller degree of freedom than the dimension of W, completing the missing data becomes a better posed problem. This gives rise to the following optimization problem:

$$\min_{U,V} \quad \frac{1}{2} \left\| \mathcal{P}_\Omega(UV^T - \widehat{W}) \right\|^2 \tag{8}$$

or its equivalence reformulation

$$\min_U \left\{ \frac{1}{2} \left\| \mathcal{P}_\Omega(UV(U)^T - \widehat{W}) \right\|^2 \Big| U^T U = I_r \right\} \tag{9}$$

where the factor V is now a function of U.

Unfortunately, (8) is not a convex optimization problem. The quality of the solutions one may get by minimizing this objective function depends on specific algorithms and their initializations. Roughly speaking, the various algorithms for (8) may be grouped into three categories: **alternating minimization**, **first order** gradient methods and **second order** Newton-like methods.

Simple approaches like alternating least squares (ALS) or equivalently PowerFactorization (Hartley and Schaffalitzky 2003) fall into the first category. They alternatingly fix one factor and minimize the objective over the other using least squares method. A more sophisticated algorithm is BALM (Del Bue et al 2012), which uses the Augmented Lagrange Multiplier method to gradually impose additional problem-specific manifold constraints. The inner loop however is still alternating minimization. This category of methods has the reputation of reducing the objective value quickly in the first few iterations, but they usually take a large number of iterations to converge to a high quality solution (Buchanan and Fitzgibbon 2005).

First order gradient methods are efficient, easy to implement and they are able to scale up to million-by-million matrices if stochastic gradient descent is adopted.

Therefore it is very popular for large-scale recommendation systems. Typical approaches include Simon Funk's incremental SVD (Funk 2006), nonlinear conjugate gradient (Srebro and Jaakkola 2003) and more sophisticatedly, gradient descent on the Grassmannian/Stiefel manifold, such as GROUSE (Balzano et al 2010) and OptManifold (Wen and Yin 2013). These methods, however, as we will demonstrate later, easily get stuck in local minima[2].

The best performing class of methods are the second order Newton-like algorithms, in that they demonstrate superior performance in both accuracy and the speed of convergence (though each iteration requires more computation); hence they are suitable for small to medium scale problems requiring high accuracy solutions (e.g., SfM and photometric stereo in computer vision). Representatives of these algorithms include the damped Newton method (Buchanan and Fitzgibbon 2005), Wiberg(ℓ_2) (Okatani and Deguchi 2007), LM_S and LM_M of Chen (2008) and LM_GN, which is a variant of LM_M using Gauss-Newton (GN) to approximate the Hessian function.

As these methods are of special importance in developing our PARSuMi algorithm, we conduct extensive numerical evaluations of these algorithms in Section 3 to understand their pros and cons as well as the key factors that lead to some of them finding global optimal solutions more often than others.

It is worthwhile to note some delightful recent efforts to scale the first two classes of MF methods to internet scale, e.g., parallel coordinate descent extension for ALS (Yu et al 2012) and stochastic gradient methods in "Hogwild" (Recht et al 2011). It will be an interesting area of research to see if the ideas in these papers can be used to make the second order methods more scalable.

In addition, there are a few other works in each category that take into account the corruption problem by changing the quadratic penalty term of (8) into ℓ_1-norm or Huber function

$$\min_{U,V} \quad \left\| \mathcal{P}_\Omega(UV^T - \widehat{W}) \right\|_1 , \tag{10}$$

$$\min_{U,V} \quad \sum_{(ij) \in \Omega} \text{Huber}\big((UV^T - \widehat{W})_{ij}\big). \tag{11}$$

Notable algorithms to solve these formulations include alternating linear programming (ALP) and alternating quadratic programming (AQP) in Ke and Kanade (2005), GRASTA (He et al 2011) that extends GROUSE,

---

[2] Our experiment on synthetic data shows that the strong Wolfe line search adopted by Srebro and Jaakkola (2003) and Wen and Yin (2013) somewhat ameliorates the issue, though it does not seem to help much on real data.

as well as Wiberg ℓ_1 (Eriksson and Van Den Hengel 2010) that uses a second order Wiberg-like iteration. While it is well known that the ℓ_1-norm or Huber penalty term can better handle outliers, and the models (10) and (11) are seen to be effective in some problems, there is not much reason for a "convex" relaxation of the ℓ_0 pseudo-norm[3], since the rank constraint imposed by matrix factorization is already highly non-convex. Empirically, we find that ℓ_1-norm penalty offers poor denoising ability to dense noise and also suffers from "Robin Hood" attribute. Comparison with this class of methods will be given later in Section 5, which shows that our method can better handle noise and corruptions.

The practical advantage of ℓ_0 over ℓ_1 penalty is well illustrated in Xiong et al (2011), where Xiong et al proposed an ℓ_0-based robust matrix factorization method which deals with corruptions and a given rank constraint. Our work is similar to Xiong et al (2011) in that we both eschew the convex surrogate ℓ_1-norm in favor of using the ℓ_0-norm directly. However, our approach treats both corruptions and missing data. More importantly, our treatment of the problem is different and it results in a convergence guarantee that covers the algorithm of Xiong et al (2011) as a special case; this will be further explained in Section 4.

2.3 Emerging theory for matrix factorization

As we mentioned earlier, a fundamental drawback of matrix factorization methods for low rank matrix completion is the lack of proper theoretical foundation. However, thanks to the better understanding of low-rank structures nowadays, some theoretical analysis of this problem slowly emerges. This class of methods are essentially designed for solving noisy matrix completion problem with an explicit rank constraint, i.e.,

$$\min_W \left\{ \frac{1}{2} \left\| \mathcal{P}_\Omega(W - \widehat{W}) \right\|^2 \,\middle|\, \mathrm{rank}(W) \le r \right\}. \qquad (12)$$

From a combinatorial-algebraic perspective, Kiràly and Tomioka (2012) provided a sufficient and necessary condition on the existence of an unique rank-r solution to (12). It turns out that if the low-rank matrix is *generic*, then *unique completability* depends only on the support of the observations Ω. This suggests that the incoherence and random sampling assumptions typically required by various nuclear norm minimization methods may limit the portion of problems solvable by the latter to only a small subset of those solvable by matrix factorization methods.

Around the same time, Wang and Xu (2012) studied the stability of matrix factorization under arbitrary noise. They obtained a stability bound for the optimal solution of (12) around the ground truth, which turns out to be better than the corresponding bound for nuclear norm minimization in Candès and Plan (2010) by a scale of $\sqrt{\min(m, n)}$ (in Big-O sense). The study however bypassed the practical problem of how to obtain the global optimal solution for this non-convex problem.

This gap is partially closed by the recent work of Jain et al (2012), in which the global minimum of (12) can be obtained up to an accuracy ϵ with $O(\log 1/\epsilon)$ iterations using a slight variation of the ALS scheme. The guarantee requires the observation to be noiseless, sampled uniformly at random and the underlying subspace of W needs to be incoherent—basically all assumptions in the convex approach—yet still requires slightly more observations than that for nuclear norm minimization. It does not however touch on when the algorithm is able to find the global optimal solution when the data is noisy. Despite not achieving stronger theoretical results nor under weaker assumptions than the convex relaxation approach, this is the first guarantee of its kind for matrix factorization. Given its more effective empirical performance, we believe that there is great room for improvement on the theoretical front. A secondary contribution of this paper is to find the potentially "right" algorithm or rather constituent elements of algorithm for theoreticians to look deeper into.

3 Numerical evaluation of matrix factorization methods

To better understand the performance of different methods, we compare the following attributes quantitatively for all three categories of approaches that solve (8) or (9)[4]:

Sample complexity Number of samples required for exact recovery of random uniformly sampled observations in random low-rank matrices, an index typically used to quantify the performance of nuclear norm based matrix completion.

Hits on global optimal[synthetic] The proportion of random initializations that lead to the global optimal solution on random low rank matrices with (a) increasing Gaussian noise, (b) exponentially decaying singular values.

Hits on global optimal[SfM] The proportion of random initializations that lead to the global optimal

---

[3] The cardinality of non-zero entries, which strictly speaking is not a norm.

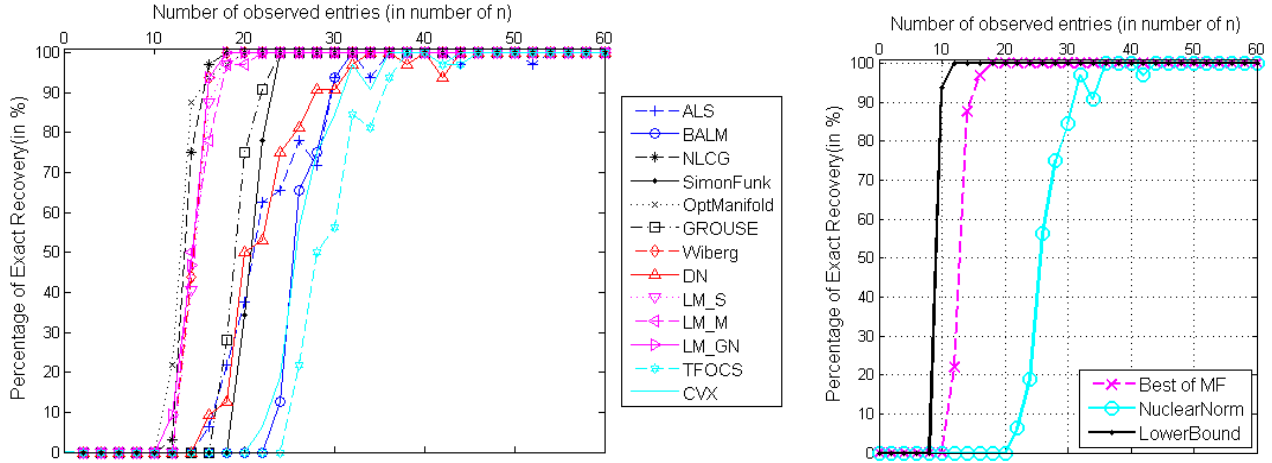[4] As a reference, we also included nuclear norm minimization that solve (4) where applicable.

Fig. 2 Exact recovery with **increasing number of random observations**. Algorithms (random initialization) are evaluated on 100 randomly generated rank-4 matrices of dimension 100×100. The number of observed entries increases from 0 to $50n$. To account for small numerical error, the result is considered "exact recovery" if the RMSE of the recovered entries is smaller than 10^{-3}. On the left, CVX (Grant and Boyd 2012) and TFOCS (Becker et al 2012) (in cyan) solves the nuclear norm based matrix completion (4), everything else aims to solve matrix factorization (8). On the right, the best solution of MF across all algorithms is compared to the CVX solver for nuclear norm minimization (solved with the highest numerical accuracy) and a lower bound (below the bound, the number of samples is smaller than r for at least a row or a column).
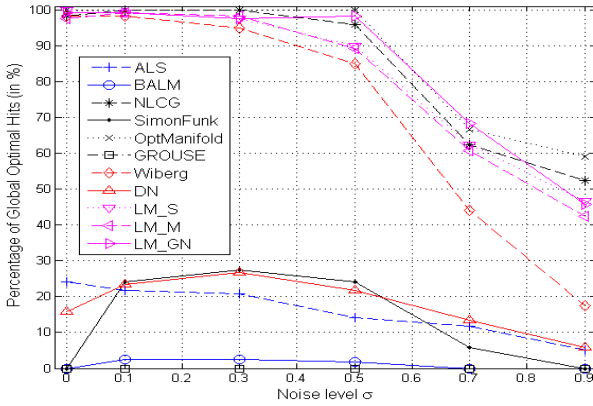

Fig. 3 Percentage of hits on global optimal with **increasing level of noise**. Five rank-4 matrices are generated by multiplying two standard Gaussian matrices of dimension 40×4 and 4×60. 30% of entries are uniformly picked as observations with additive Gaussian noise $N(0, \sigma)$. 24 different random initialization are tested for each matrix. The "global optimal" is assumed to be the solution with lowest objective value across all testing algorithm and all initializations.

Fig. 4 Percentage of hits on global optimal for **ill-conditioned low-rank matrices**. Data are generated in the same way as in Fig. 3 with $\sigma = 0.05$, except that we further take SVD and rescale the ith singular value according to $1/\alpha^i$. The Frobenious norm is normalized to be the same as the original low-rank matrix. The exponent α is given on the horizontal axis.

solution on the Oxford Dinosaur sequence (Buchanan and Fitzgibbon 2005) used in the SfM community.

The sample complexity experiment in Fig. 2 shows that the best performing matrix factorization algorithm attains exact recovery with the number of observed entries at roughly 18%, while CVX for nuclear norm minimization needs roughly 36% (even worse for numerical solvers such as TFOCS). This seems to imply that the sample requirement for MF is fundamentally smaller than that of nuclear norm minimization. As MF as-
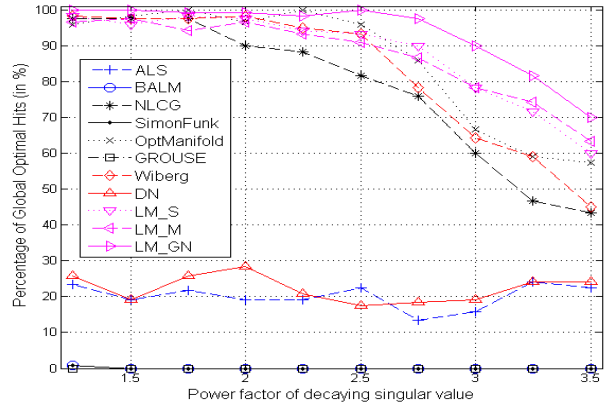
sumes known rank of the underlying matrix while nuclear norm methods do not, the results we observe are quite reasonable. In addition, among different MF algorithms, some perform much better than others. The best few of them achieve something close to the lower
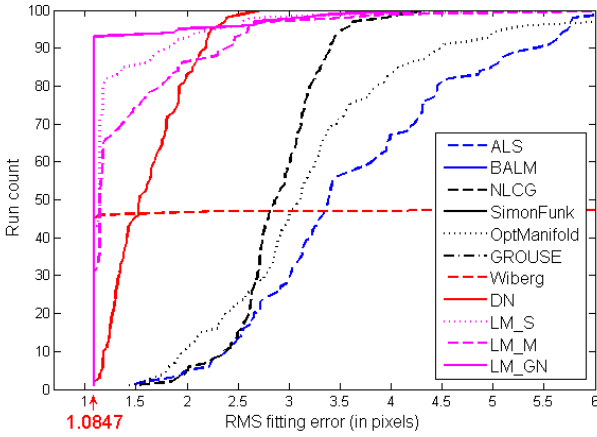
Fig. 5 Cumulative histogram on the pixel RMSE for 100 randomly initialized runs conducted for each algorithm on Dinosaur sequence. The curve summarizes how many runs of each algorithm corresponds to the global optimal solution (with pixel RMSE 1.0847) on the horizontal axis. Note that the input pixel coordinates are normalized to between [0, 1] for the experiments, but to be comparable with Buchanan and Fitzgibbon (2005), the objective value is scaled back to the original size.

bound[5]. This corroborates our intuition that MF is probably a better choice for problems with known rank.

From Fig. 3 and 4, we observe that the following classes of algorithms, including LM_X series (Chen 2008), Wiberg (Okatani and Deguchi 2007), Non-linear Conjugate Gradient method (NLCG) (Srebro and Jaakkola 2003) and the curvilinear search on Stiefel manifold (OptManifold (Wen and Yin 2013)) perform significantly better than others in reaching the global optimal solution despite their non-convexity. The percentage of global optimal hits from random initialization is promising even when the observations are highly noisy or when the condition number of the underlying matrix is very large[6].

The common attribute of the four algorithms is that they are all based on the model (9) which parameterizes the factor V as a function of U and then optimizes over U alone. This parameterization essentially reduces the problem to finding the best subspace that fits the data. What is different between them is the way they update the solution in each iteration. OptManifold and NLCG adopt a Strong Wolfe line search that allows the algorithm to take large step sizes, while the second order methods approximate each local neighborhood with

a convex quadratic function and jump directly to the minimum of the approximation. This difference appears to matter tremendously on the SfM experiment (see Fig. 5). We observe that only the second order methods achieve global optimal solution frequently, whereas the Strong Wolfe line search adopted by both OptManifold and NLCG does not seem to help much on the real data experiment like it did in simulation with randomly generated data. Indeed, neither approach reaches the global optimal solution even once in the hundred runs, though they are rather close in quite a few runs. Despite these close runs, we remark that in applications like SfM, it is important to actually reach the global optimal solution. Due to the large amount of missing data in the matrix, even slight errors in the sampled entries can cause the recovered missing entries to go totally haywire with a seemingly good local minimum (see Fig. 6). We thus refrain from giving any credit to local minima even if the $\mathrm{RMSE_{visible}}$ error (defined in (13)) is very close to that of the global minimum.

$$\mathrm{RMSE_{visible}} := \frac{\|\mathcal{P}_\Omega(W_{\mathrm{recovered}} - \widehat{W})\|}{\sqrt{|\Omega|}}. \qquad (13)$$

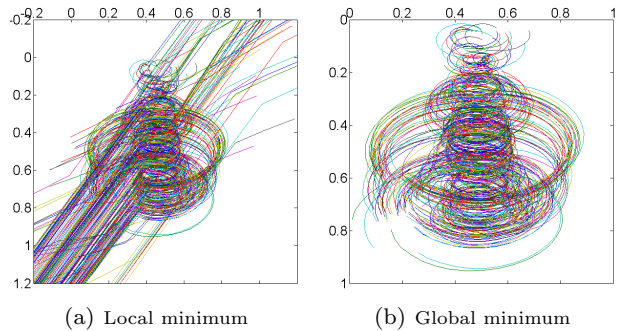(a) Local minimum   (b) Global minimum

Fig. 6 Comparison of the feature trajectories corresponding to a local minimum and global minimum of (8), given partial uncorrupted observations. Note that $\mathrm{RMSE_{visible}} = 1.1221$pixels in (a) and $\mathrm{RMSE_{visible}} = 1.0847$pixels in (b). The latter is precisely the reported global minimum in Okatani and Deguchi (2007); Buchanan and Fitzgibbon (2005) and Chen (2008). Despite the tiny difference in $\mathrm{RMSE_{visible}}$, the filled-in values for missing data in (a) are far off.

Another observation is that LM_GN seems to work substantially better than other second-order methods with subspace or manifold parameterization, reaching global minimum 93 times out of the 100 runs. Compared to LM_S and LM_M, the only difference is the use of Gauss-Newton approximation of the Hessian. According to the analysis in Chen (2011), the Gauss-Newton Hessian provides the only non-negative convex

---

[5] The lower bound is given by the percentage of randomly generated data that have at least one column or row having less than r samples. Clearly, having at least r samples for every column and row is a necessary condition for exact recovery.

[6] When $\alpha = 3.5$ in Fig. 4, r^{th} singular value is almost as small as the spectral norm of the input noise.

	DN	Wiberg	LM_S	LM_M	LM_GN
No. of hits at global min.	2	46	42	32	93
No. of hits on stopping condition	75	47	99	93	98
Average run time(sec)	324	837	147	126	40
No. of variables	(m+n)r	(m-r)r	mr	(m-r)r	(m-r)r
Hessian	Yes	Gauss-Newton	Yes	Yes	Gauss-Newton
LM/Trust Region	Yes	No	Yes	Yes	Yes
Largest Linear system to solve	$[(m+n)r]^2$	$|\Omega| \times mr$	$mr \times mr$	$[(m-r)r]^2$	$[(m-r)r]^2$

Table 2 Comparison of various second order matrix factorization algorithms

quadratic approximation that preserves the so-called "zero-on-$(n-1)$-D" structure of a class of nonlinear least squares problems, into which (8) can be formulated. Compared to the Wiberg algorithm that also uses Gauss-Newton approximation, the advantage of LM_GN is arguably the better global convergence due to the augmentation of the LM damping factor. Indeed, as we verify in the experiment, Wiberg algorithm fails to converge at all in most of its failure cases. The detailed comparisons of the second order methods and their running time on the Dinosaur sequence are summarized in Table 2. Part of the results replicate that in Chen (2008); however, Wiberg algorithm and LM_GN have not been explicitly compared previously. It is clear from the Table that LM_GN is not only better at reaching the optimal solution, but also computationally cheaper than other methods which require explicit computation of the Hessian[7].

To summarize the key findings of our experimental evaluation, we observe that: (a) the fixed-rank MF formulation requires less samples than nuclear norm minimization to achieve exact recovery; (b) the compact parameterization on the subspace, strong line search or second order update help MF algorithms in avoiding local minima in high noise, poorly conditioned matrix setting; (c) LM_GN with Gauss-Newton update is able to reach the global minimum with a very high success rate on a challenging real SfM data sequence.

4 (Partially Majorized) Proximal Alternating Robust Subspace Minimization for (3)

Our proposed PARSuMi method for problem (3) works in two stages. It first obtains a good initialization from an efficient convex relaxation of (3), which will be described in Section 4.6. This is followed by the minimization of the low rank matrix W and the sparse matrix E alternatingly until convergence. The efficiency of our PARSuMi method depends on the fact that the two inner minimizations of W and E admit efficient solutions,

which will be derived in Sections 4.1 and 4.3 respectively. Specifically, in step k, we compute W^{k+1} from either

$$\min_W \frac{1}{2}\|H \circ (W - \widehat{W} + E^k)\|^2 + \frac{\beta_1}{2}\|H \circ (W - W^k)\|^2$$
$$\text{subject to} \quad \text{rank}(W) \leq r, \tag{14}$$

or its quadratic majorization[8], and E^{k+1} from

$$\min_E \frac{1}{2}\|H \circ (W^{k+1} - \widehat{W} + E)\|^2 + \frac{\beta_2}{2}\|E - E^k\|^2$$
$$\text{subject to} \quad \|E\|_0 \leq N_0, \ \|E\| \leq K_E, \ E \in \mathbb{R}_\Omega^{m \times n}, \tag{15}$$

where H is defined as in (2). Note that the above iteration is different from applying a direct alternating minimization of (3). We have added the proximal regularization terms $\|H \circ (W - W^k)\|^2$ and $\|E - E^k\|^2$ to make the objective functions in the subproblems coercive and hence ensuring that W^{k+1} and E^{k+1} are well defined. As is shown in Attouch et al (2010), the proximal terms are critical to ensure the critical point convergence of the sequence. In addition, we have added a quadratic majorization safeguard step when computing W^{k+1}. This is to safeguard the convergence even if our computed W^{k+1} fails to be a global minimizer of (14). Further details of the algorithm and the proof of its convergence are provided in the subsequent sections.

4.1 Computation of W^{k+1} in (14)

Our solution for (14) consists of two steps. We first transform the rank-constrained minimization (14) into an equivalent subspace fitting problem, then solve the new formulation using LM_GN.

Motivated by the findings in Section 3 where the most successful algorithms for solving (12) are based on the formulation (9), we will now derive a similar equivalent reformulation of (14). Our reformulation of (14) is motivated by the N-parametrization of (12) due to Chen (2008), who considered the task of matrix completion as finding the best subspace to fit the partially

---

[7] Wiberg algoirthm takes longer time mainly because it sometimes does not converge and exhausts the maximum number of iterations.

[8] We will explain this further shortly.

observed data. In particular, Chen proposes to solve (12) using

$$\min_N \left\{ \frac{1}{2} \sum_i \hat{w}_i^T (I - \mathbb{P}_i) \hat{w}_i \,\middle|\, N^T N = I \right\} \qquad (16)$$

where N is a $m \times r$ matrix whose column space is the underlying subspace to be reconstructed, N_i is N but with those rows corresponding to the missing entries in column i removed. $\mathbb{P}_i = N_i N_i^+$ is the projection onto span(N_i) with N_i^+ being the Moore-Penrose pseudo inverse of N_i, and the objective function minimizes the sum of squares distance between $\hat{w}_i$ to span(N_i), where $\hat{w}_i$ is the vector of observed entries in the ith column of $\widehat{W}$.

4.1.1 N-parameterization of the subproblem (14)

First define the matrix $\overline{H} \in \mathbb{R}^{m \times n}$ as follows:

$$\overline{H}_{ij} = \begin{cases} \sqrt{1 + \beta_1} & \text{if } (i,j) \in \Omega \\ \sqrt{\epsilon + \epsilon \beta_1} & \text{if } (i,j) \notin \Omega. \end{cases} \qquad (17)$$

Let $B^k \in \mathbb{R}^{m \times n}$ be the matrix defined by

$$B_{ij}^k = \begin{cases} \frac{1}{\sqrt{1+\beta_1}} (\widehat{W}_{ij} - E_{ij}^k + \beta_1 W_{ij}^k) & \text{if } (i,j) \in \Omega \\ \frac{\epsilon \beta_1}{\sqrt{\epsilon + \epsilon \beta_1}} W_{ij}^k & \text{if } (i,j) \notin \Omega. \end{cases} \qquad (18)$$

Define the diagonal matrices $\mathbb{D}_i \in \mathbb{R}^{m \times m}$ to be

$$\mathbb{D}_i = \text{diag}(\overline{H}_i), \quad i = 1, \ldots, n \qquad (19)$$

where $\overline{H}_i$ is the ith column of $\overline{H}$. It turns out that the N-parameterization for the regularized problem (14) has a similar form as (16), as shown below.

Proposition 1 (Equivalence of subspace parameterization) *Let $\mathbb{Q}_i(N) = \mathbb{D}_i N (N^T \mathbb{D}_i^2 N)^{-1} N^T \mathbb{D}_i$, which is the $m \times m$ projection matrix onto the column space of $\mathbb{D}_i N$. The problem (14) is equivalent to the following problem:*

$$\min_N \quad f(N) := \frac{1}{2} \sum_{i=1}^n \|B_i^k - \mathbb{Q}_i(N) B_i^k\|^2 \qquad (20)$$

$$\text{subject to} \quad N^T N = I, \ N \in \mathbb{R}^{m \times r}$$

where B_i^k is the ith columns of B^k. If N_* is an optimal solution of (20), then W^{k+1}, whose columns are defined by

$$W_i^{k+1} = \mathbb{D}_i^{-1} \mathbb{Q}_i(N_*) B_i^k, \qquad (21)$$

is an optimal solution of (14).

Proof We can show by some algebraic manipulations that the objective function in (14) is equal to

$$\frac{1}{2} \|\overline{H} \circ W - B^k\|^2 + \text{constant}$$

Now note that we have

$$\{W \in \mathbb{R}^{m \times n} \mid \text{rank}(W) \leq r\}$$
$$= \{NC \mid N \in \mathbb{R}^{m \times r}, C \in \mathbb{R}^{r \times n}, N^T N = I\}. \ (22)$$

Thus the problem (14) is equivalent to

$$\min_N \{f(N) \mid N^T N = I, N \in \mathbb{R}^{m \times r}\} \qquad (23)$$

where

$$f(N) := \min_C \frac{1}{2} \|\overline{H} \circ (NC) - B^k\|^2.$$

To derive (20) from the above, we need to obtain $f(N)$ explicitly as a function of N. For a given N, the unconstrained minimization problem over C in $f(N)$ has a strictly convex objective function in C, and hence the unique global minimizer satisfies the following optimality condition:

$$N^T ((\overline{H} \circ \overline{H}) \circ (NC)) = N^T (\overline{H} \circ B^k). \qquad (24)$$

By considering the ith column C_i of C, we get

$$N^T \mathbb{D}_i^2 N C_i = N^T \mathbb{D}_i B_i^k, \quad i = 1, \ldots, n. \qquad (25)$$

Since N has full column rank and D^i is positive definite, the coefficient matrix in the above equation is nonsingular, and hence

$$C_i = (N^T \mathbb{D}_i^2 N)^{-1} N^T \mathbb{D}_i B_i^k.$$

Now with the optimal C_i above for the given N, we can show after some algebra manipulations that $f(N)$ is given as in (20). □

We can see that when $\beta_1 \downarrow 0$ in (20), then the problem reduces to (16), with the latter's $\hat{w}_i$ appropriately modified to take into account of E^k. Also, from the above proof, we see that the N-parameterization reduces the feasible region of W by restricting W to only those potential optimal solutions among the set of W satisfying the expression in (21). This seems to imply that it is not only equivalent but also advantageous to optimize over N instead of W. While we have no theoretical justification of this conjecture, it is consistent with our experiments in Section 3 which show the superior performance of those algorithms using subspace parameterization in finding global minima and vindicates the design motivations of the series of LM_X algorithms in Chen (2008).

4.1.2 LM_GN updates

Now that we have shown how to handle the regularization term and validated the equivalence of the transformation, the steps to solve (14) essentially generalize those of LM_GN (available in Section 3.2 and Appendix A of Chen (2011)) to account for the general mask H. The derivations of the key formulae and their meanings are given in this section.

In general, Levenberg-Marquadt solves the non-linear problem with the following sum-of-squares objective function

$$\mathcal{L}(x) = \frac{1}{2} \sum_{i=1:n} \|y_i - f_i(x)\|^2, \qquad (26)$$

by iteratively updating x as follows:

$$x \leftarrow x + (J^T J + \lambda I)^{-1} J^T \mathbf{r},$$

where $J = [J_1; \ldots; J_n]$ is the Jacobian matrix and J_i is the Jacobian matrix of f_i; $\mathbf{r}$ is the concatenated vector of residual $r_i := y_i - f_i(x)$ for all i, and λ is the damping factor that interpolates between Gauss-Newton update and gradient descent. We may also interpret the iteration as a Damped Newton method with a first order approximation of the Hessian matrix using $J^T J$.

Note that the objective function of (20) can be expressed in the form of (26) by taking $x := \text{vec}(N)$, data $y_i := B_i^k$, and function

$$f_i(x := \text{vec}(N)) = \mathbb{Q}_i(N) B_i^k = \mathbb{Q}_i y_i$$

Proposition 2 *Let $\mathcal{T} \in \mathbb{R}^{mr \times mr}$ be the permutation matrix such that $\text{vec}(X^T) = \mathcal{T}\text{vec}(X)$ for any $X \in \mathbb{R}^{m \times r}$. The Jacobian of $f_i(x) = \mathbb{Q}_i(N)y_i$ is given as follows:*

$$J_i(x) = (\mathbb{A}_i^T y_i)^T \otimes ((I - \mathbb{Q}_i)\mathbb{D}_i) + [(\mathbb{D}_i r_i)^T \otimes \mathbb{A}_i]\mathcal{T}. (27)$$

Also $J^T J = \sum_{i=1}^n J_i^T J_i$, $J^T r = \sum_{i=1}^n J_i^T r_i$, where

$$J_i^T J_i = (\mathbb{A}_i^T y_i y_i^T \mathbb{A}_i) \otimes (\mathbb{D}_i (I - \mathbb{Q}_i)\mathbb{D}_i)$$
$$+ \mathcal{T}^T[(\mathbb{D}_i r_i r_i^T \mathbb{D}_i) \otimes (\mathbb{A}_i^T \mathbb{A}_i)]\mathcal{T} \qquad (28)$$

$$J_i^T r_i = \text{vec}(\mathbb{D}_i r_i (\mathbb{A}_i^T y_i)^T). \qquad (29)$$

In the above, $\otimes$ denotes the Kronecker product.

Proof Let $\mathbb{A}_i = \mathbb{D}_i N(N^T \mathbb{D}_i^2 N)^{-1}$. Given sufficiently small δN, we can show that the directional derivative of f_i at N along δN is given by

$$f_i'(N + \delta N) = (I - \mathbb{Q}_i)\mathbb{D}_i \delta N \mathbb{A}_i^T y_i + \mathbb{A}_i \delta N^T \mathbb{D}_i r_i.$$

By using the property that $\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$, we have

$$\text{vec}(f_i'(N + \delta N)) = [(\mathbb{A}_i^T y_i)^T \otimes ((I - \mathbb{Q}_i)\mathbb{D}_i)]\text{vec}(\delta N)$$
$$+ [(\mathbb{D}_i r_i)^T \otimes \mathbb{A}_i]\text{vec}(\delta N^T)$$

Algorithm 1 Levenberg-Marquadt method for (14)

Input: $\widehat{W}, E^k, W^k, \bar{H}$, objective function $\mathcal{L}(x)$ and initial N^k; numerical parameter $\lambda, \rho > 1$.
Initialization: Compute $y_i = B_i^k$ for $i = 1, \ldots, n$, and $x^0 = \text{vec}(N^k)$, $j = 0$.
while not converged **do**
   1. Compute $J^T \mathbf{r}$ and $J^T J$ using (29) and (28).
   2. Compute $\Delta x = (J^T J + \lambda I)^{-1} J^T r$
  while $\mathcal{L}(x + \Delta x) < \mathcal{L}(x)$ **do**
    (1) $\lambda = \rho \lambda$.
    (2) $\Delta x = (J^T J + \lambda I)^{-1} J^T r$.
  end while
   3. $\lambda = \lambda/\rho$.
   4. Orthogonalize $N = \text{orth}[\text{reshape}(x^j + \Delta x)]$.
   5. Update $x^{j+1} = \text{vec}(N)$.
   6. Iterate $j = j + 1$
end while
Output: $N^{k+1} = N$, W^{k+1} using (21) with N^{k+1} replacing N_*.

From here, the required result in (27) follows.

To prove (28), we make use of the following properties of Kronecker product: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ and $(A \otimes B)^T = A^T \otimes B^T$. By using these properties, we see that $J_i^T J_i$ has four terms, with two of the terms containing involving $\mathbb{D}_i(I - \mathbb{Q}_i)\mathbb{A}_i$ or its transpose. But we can verify that $\mathbb{Q}_i \mathbb{A}_i = \mathbb{A}_i$ and hence those two terms become 0. The remaining two terms are those appearing in (28) after using the fact that $(I - \mathbb{Q}_i)^2 = I - \mathbb{Q}_i$. Next we prove (29). We have

$$J_i^T r_i = \text{vec}(\mathbb{D}_i(I - \mathbb{Q}_i)r_i(\mathbb{A}_i^T y_i)^T) + \mathcal{T}^T \text{vec}(\mathbb{A}_i^T r_i r_i^T \mathbb{D}_i).$$

By noting that $\mathbb{A}_i^T r_i = 0$ and $\mathbb{Q}_i r_i = 0$, we get the required result in (29). $\qquad \square$

The complete procedure of solving (14) is summarized in Algorithm 1. In all our experiments, the initial λ is chosen as $1e - 6$ and $\bar{\rho} = 10$.

4.2 Quadratic majorization of (14)

Recall that while the LM_GN method may be highly successful in computing a global minimizer for (14) empirically, W^{k+1} may fail to be a global minimizer occasionally. Thus before deriving the update rule for E^{k+1}, we consider minimizing a quadratic majorization of (14) as a safeguard step to ensure the convergence of the PARSuMi iterations. Recall that (14) is equivalent to

$$W^{k+1} = \underset{W}{\text{argmin}} \left\{ \frac{1}{2} \|\overline{H} \circ (W - \hat{B}^k)\|^2 \,\Big|\, \text{rank}(W) \leq r \right\}$$

where $\hat{B}^k = \overline{H}^{-1} \circ B^k$ with $\overline{H}$ and B^k defined as in (17) and (18) respectively.

For convenience, we denote the above objective function $F(W, \hat{B}^k)$. Suppose that we have positive vectors $p \in \mathbb{R}^m$ and $q \in \mathbb{R}^n$ such that

$$\bar{H}_{ij}^2 \leq p_i q_j \quad \forall\, i = 1, \ldots, m,\ j = 1, \ldots, n. \quad (30)$$

Note that the above inequality always holds if p, q are chosen to be

$$p_i = \max\{\bar{H}_{ij} \mid j = 1, \ldots, n\}, \quad i = 1, \ldots, m$$
$$q_j = \max\{\bar{H}_{ij} \mid i = 1, \ldots, m\}, \quad j = 1, \ldots, n. \quad (31)$$

Let $G^k = \nabla_W F(W^k, \hat{B}^k)$. We majorize $F(W, \hat{B}^k)$ by bounding its Taylor expansion at W^k

$$
\begin{aligned}
F(W, \hat{B}^k) - F(W^k, \hat{B}^k) &= \langle G^k, W - W^k \rangle \\
&+ \frac{1}{2}\langle W - \hat{B}^k, (\bar{H} \circ \bar{H}) \circ (W - \hat{B}^k) \rangle \\
&\leq \langle G^k, W - W^k \rangle + \frac{1}{2}\langle W - W^k, P(W - W^k)Q \rangle \\
&= \frac{1}{2}\|P^{1/2}WQ^{1/2} - U^k\|^2 - \frac{1}{2}\|P^{-1/2}G^kQ^{-1/2}\|^2 \quad (32)
\end{aligned}
$$

where $P = \mathrm{diag}(p)$ and $Q = \mathrm{diag}(q)$, $U^k = P^{1/2}W^kQ^{1/2} - P^{-1/2}G^kQ^{-1/2}$.

Proposition 3 *The minimizer of quadratic majorization function in (32) is given in closed-form by*

$$W_{\mathrm{QM}}^{k+1} \in P^{-1/2} \Pi_r(U^k) Q^{-1/2}. \quad (33)$$

Here $\Pi_r(U^k)$ denotes the set of best rank-r approximation of U^k.

The proof is simple and is given in the Appendix. Note that this is a nonconvex minimization, yet we have an efficient closed-form solution thanks to SVD.

As we shall see later in Algorithm 4, the global minimizer W_{QM}^{k+1} in (33) of the quadratic majorization function of $F(W, \hat{B}^k)$ is used as a safeguard when the computed solution W^{k+1} from (14) is inferior (which necessarily implies that W^{k+1} is not a global optimal solution) to W_{QM}^{k+1}. By doing so, the convergence of PARSuMi can be ensured, as we shall prove in Theorem 1.

4.3 Sparse corruption recovery step (15)

In the sparse corruption step, we need to solve the ℓ_0-constrained least squares minimization (15). This problem is combinatorial in nature, but fortunately, for our problem, we show that a closed-form solution can be obtained. Let $x := \mathcal{P}_\Omega(E)$. Observe that (15) can be expressed in the following equivalent form:

$$\min_x \left\{ \|x - b\|^2 \mid \|x\|_0 \leq N_0,\ \|x\|^2 - K_E^2 \leq 0 \right\} \quad (34)$$

where $b = \mathcal{P}_\Omega(\widehat{W} - W^{k+1} + \beta_2 E^k)/(1 + \beta_2)$.

Algorithm 2 Closed-form solution of (15)

Input: $\widehat{W}, W^{k+1}, E^k, \Omega$.
1. Compute b using (34).
2. Compute x using (35).
Output: $E^{k+1} = \mathcal{P}_\Omega^*(x)$.

Algorithm 3 (Partially Majorized) Proximal Alternating Robust Subspace Minimization (PARSuMi)

Input: Observed data $\widehat{W}$, sample mask Ω, parameter r, N_0. Initialization W^0 and E^0 (typically by Algorithm 5 described in Section 4.6), $k = 0$.
repeat
   1a. Solve (14) using Algorithm 1 with W^k, E^k, N^k, obtain updates $\tilde{W}^{k+1}$ and $\tilde{N}^{k+1}$
   1b. Evaluate (33) with W^k, E^k obtain updates $\hat{W}^{k+1}$.
   2. Assign $W^{k+1} = \mathrm{argmin}_{W \in \{\tilde{W}^{k+1}, \hat{W}^{k+1}\}} F(W, \hat{B}^k)$, and then assign the corresponding $N^{(k+1)}$.
   3. Solve (15) using Algorithm 2 with W^{k+1}, E^k; obtain updates E^{k+1}.
until $\|W^{k+1} - W^k\| < \|W^k\| \cdot 10^{-6}$ and $\|E^{k+1} - E^k\| < \|E^k\| \cdot 10^{-6}$
Output: Accumulation points $\overline{W}$ and $\overline{E}$

Proposition 4 *Let I be the set of indices of the N_0 largest (in magnitude) component of b. Then the nonzero components of the optimal solution x of (34) is given by*

$$x_I = \begin{cases} K_E b_I / \|b_I\| & \text{if } \|b_I\| > K_E \\ b_I & \text{if } \|b_I\| \leq K_E. \end{cases} \quad (35)$$

The proof (deferred to the Appendix) involves checking the optimality conditions of (34) assuming known support set and finding the optimal support set in a decoupled fashion.

The procedure to obtain the optimal solution of (15) is summarized in Algorithm 2. We remark that this is a very special case of ℓ_0-constrained optimization; the availability of the exact closed form solution depends on both terms in (15) being decomposable into individual (i, j) term. In general, if we change the operator $M \to H \circ M$ in (15) to a general linear transformation (e.g., a sensing matrix in compressive sensing), or change the norm $\|\cdot\|$ of the proximal term to some other norm such as spectral norm or nuclear norm, then the problem becomes NP-hard.

4.4 Algorithm

Our method is summarized in Algorithm 3. Note that we do not need to know the exact cardinality of the corrupted entries; N_0 can be taken as an upper bound of the allowable number of corruptions. As a rule of thumb, 10-15% of $|\Omega|$ is a reasonable size. The surplus

in N_0 will only label a few noisy samples as corruptions, which should not affect the recovery of either W or E, so long as the remaining $|\Omega| - N_0$ samples are still sufficient. The other parameter r is typically given by the physical model of the problem. For those problems where r is not known, choosing r is analogous to choosing the regularization parameter as in other machine learning tasks. A large r will lead to overfitting and poorly estimated missing data while an overly small r will cause underfitting of the observed data.

4.5 Convergence to a critical point

In this section, we show the convergence of Algorithm 3 to a critical point.

Note that due to the non-convex nature of the subproblem (14), Algorithm 1 is guaranteed to converge only to a local minimum. Therefore, the result in Attouch et al (2010) that requires global optimal solutions in all subproblems cannot be directly applied in our case for the critical point convergence proof. Empirically, we cannot hope LM_GN to always find the global optimal solution of (14) either, as our experiments on LM_GN in Section 3 clearly demonstrated. As a result, we design the partial majorization (Step 1b) in Algorithm 3 to safeguard against the case when the computed solution from Step 1a is not a global optimal solution. The safeguard step is powerful in that we do not need to assume anything on the computed solution of the subproblem before we can prove the overall critical point convergence.

We start our convergence proof by first defining an equivalent formulation of (3) in terms of closed, bounded sets. The convergence proof is then based on the indicator functions for these closed and bounded sets, which have the key lower semicontinuous property.

Let $K_W = 2\|\widehat{W}\| + K_E$. Define the closed and bounded sets:

$$\mathcal{W} = \{W \in \mathbb{R}^{m \times n} \mid \mathrm{rank}(W) \leq r, \|H \circ W\| \leq K_W\}$$

$$\mathcal{E} = \{E \in \mathbb{R}^{m \times n}_{\Omega} \mid \|E\|_0 \leq N_0, \|E\| \leq K_E\}.$$

We will first show that (3) is equivalent to the problem given in the next proposition.

Proposition 5 *Let* $f(W, E) := \frac{1}{2}\|H \circ (W + E - \widehat{W})\|^2$. *The problem* (3) *is equivalent to the following problem:*

$$\min\{f(W, E) \mid W \in \mathcal{W}, E \in \mathcal{E}\}. \tag{36}$$

Proof Observe that the only difference between (3) and (36) is the inclusion of the bound constraint on $\|H \circ W\|$ in (36). To show the equivalence, we only need to

show that any minimizer $(\overline{W}, \overline{E})$ of (3) must satisfy the bound constraint in $\mathcal{W}$. By definition, we know that

$$f(\overline{W}, \overline{E}) \leq f(0, 0) = \frac{1}{2}\|\widehat{W}\|^2.$$

Now for any (W, E) such that $\mathrm{rank}(W) \leq r$, $E \in \mathcal{E}$ and $\|H \circ W\| > K_W$, we must have

$$\|H \circ (W + E - \widehat{W})\| \geq \|H \circ W\| - \|H \circ (E - \widehat{W})\|$$

$$> K_W - \|E\| - \|\widehat{W}\| \geq \|\widehat{W}\|.$$

Hence $f(W, E) > \frac{1}{2}\|\widehat{W}\|^2 = f(0, 0)$. This implies that we must have $\|H \circ (\overline{W})\| \leq K_W$. □

To establish the convergence of PARSuMi, it is more convenient for us to consider the following generic problem which includes (36) as a special case. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be finite-dimensional inner product spaces, and $f : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$, $g : \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ are lower semi-continuous functions. We consider the problem:

$$\min_{x,y}\{L(x, y) := f(x) + g(y) + q(x, y)\} \tag{37}$$

where $q(x, y) = \frac{1}{2}\|Ax + By - c\|^2$ and $A : \mathcal{X} \to \mathcal{Z}$, $B : \mathcal{Y} \to \mathcal{Z}$ are given linear maps. For (36), we have $\mathcal{X} = \mathcal{Z} = \mathbb{R}^{m \times n}$, $\mathcal{Y} = \mathbb{R}^{m \times n}_{\Omega}$, $A(x) = H \circ x$, $B(y) = H \circ y$, $c = \widehat{W}$. and f, g are the following indicator functions,

$$f(x) = \begin{cases} 0 & \text{if } x \in \mathcal{W} \\ \infty & \text{otherwise} \end{cases} \quad g(y) = \begin{cases} 0 & \text{if } y \in \mathcal{E} \\ \infty & \text{otherwise} \end{cases}$$

Note that in this case, f are g are lower semicontinuous since indicator functions of closed sets are lower semicontinuous (Rudin 1987).

To denote the corresponding majorization safeguard, we define, for a fixed $(\widehat{x}, \widehat{y})$ and given $M \succ A^*A$

$$Q(x; \widehat{x}, \widehat{y}) := q(\widehat{x}, \widehat{y}) + \langle \nabla_x q(\widehat{x}, \widehat{y}), x - \widehat{x} \rangle + \frac{1}{2}\|x - \widehat{x}\|_M^2 \tag{38}$$

$$\widehat{L}(x; \widehat{x}, \widehat{y}) := Q(x; \widehat{x}, \widehat{y}) + f(x) + g(\widehat{y}) \tag{39}$$

where $\|\cdot\|_M$ is defined in the last part of Algorithm 4.

Then, we have that

$$q(x, \widehat{y}) = Q(x; \widehat{x}, \widehat{y}) - \frac{1}{2}\|x - \widehat{x}\|^2_{M-A^*A} \tag{40}$$

$$L(x, \widehat{y}) = \widehat{L}(x; \widehat{x}, \widehat{y}) - \frac{1}{2}\|x - \widehat{x}\|^2_{M-A^*A} \tag{41}$$

Consider the partially majorized proximal alternating minimization (PMPAM) outlined in Algorithm 4, which we have modified from Attouch et al (2010). The algorithm alternates between minimizing x and y, but

with the important addition of the quadratic Moreau-Yoshida regularization term (which is also known as the proximal term) in each step. The importance of Moreau-Yoshida regularization for convex matrix optimization problems has been demonstrated and studied in Liu et al (2009); Yang et al (2012); Wu et al (2011a). For our non-convex, non-smooth setting here, the importance of the proximal term will become clear when we prove the convergence of Algorithm 4. For our problem (36), the positive linear maps S and T in Algorithm 4 correspond to $\beta_1(H \circ H)\circ$ and $\beta_2 I$ respectively, where β_1, β_2 are given positive parameters. Our algorithm differs from that in Attouch et al (2010) by having the safeguard step (in Step 1b and Step 2) to ensure that critical point convergence can be achieved even if the computed solution in Step 1a is not globally optimal. Observe that one can bypass Step 1a in Algorithm 4 completely and always choose to use Step 1b. But the minimization in Step 1b based on quadratic majorization may not reduce the merit function $L(x, y_k) + \frac{1}{2}\|x - x_k\|_S^2$ as quickly as the minimization in Step 1a. Thus it is necessary to have Step 1a to ensure that Algorithm 4 converges at a reasonable speed. We note that a similar safeguard step can be introduced for the subproblem in Step 3 if the global optimality of y_{k+1} is not guaranteed.

Algorithm 4 Partially majorized proximal alternating minimization (PMPAM)

Input:$(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$; positive linear operators S and T. Choose M such that $M \succ A^*A + S$ in (38).
repeat
  1a. Compute $\widetilde{x}^{k+1} \in \arg\min\{L(x, y_k) + \frac{1}{2}\|x - x_k\|_S\}$.
  1b. Compute $\widehat{x}_{k+1} \in \arg\min\left\{\widehat{L}(x; x_k, y_k)\right\}$
  2. Consider condition (I) $L(\widetilde{x}_{k+1}, y_k) + \frac{1}{2}\|\widetilde{x}_{k+1} - x_k\|_S \leq L(\widehat{x}_{k+1}, y_k) + \frac{1}{2}\|\widehat{x}_{k+1} - x_k\|_S$. Set

$$x_{k+1} = \begin{cases} \widetilde{x}_{k+1} & \text{if condition (I) holds} \\ \widehat{x}_{k+1} & \text{otherwise.} \end{cases}$$

  3. $y^{k+1} = \arg\min\{L(x^{k+1}, y) + \frac{\beta_2}{2}\|y - y^k\|_T^2\}$
until convergence
Output: Accumulation points $\overline{x}$ and $\overline{y}$

In the above, S and T are given positive definite linear maps, and $\|x - x^k\|_S^2 = \langle x - x^k, S(x - x^k)\rangle$, $\|y - y^k\|_T^2 = \langle y - y^k, T(y - y^k)\rangle$. Note that Step 1b is to safeguard against the possibility that the computed $\widetilde{x}_{k+1}$ is not a global optimal solution of the subproblem. We assume that it is possible to compute the global optimal solution $\widehat{x}_{k+1}$ analytically.

Note that for our problem (36), the global minimizer of the nonconvex subproblem in Step 1b can be computed analytically as discussed in Section 4.2. Next we

show that any limit point of $\{(x_k, y_k)\}$ is a stationary point of L even if $\widetilde{x}_{k+1}$ computed in Step 1a is not a global minimizer of the subproblem.

Theorem 1 *Let $\{(x_k, y_k)\}$ be the sequence generated by Algorithm 4, and $(\widetilde{x}_{k+1}, \widehat{x}_{k+1})$ are the intermediate iterates at iteration k.*

(a) For all $k \geq 0$, we have that

$$L(\widehat{x}_{k+1}, y_k) + \frac{1}{2}\|\widehat{x}_{k+1} - x_k\|_S^2 + \frac{1}{2}\|\widehat{x}_{k+1} - x_k\|_{M-A^*A-S}^2$$

$$= \widehat{L}(\widehat{x}_{k+1}; x_k, y_k) \leq L(x_k, y_k), \tag{42}$$

$$L(x_{k+1}, y_k) + \frac{1}{2}\|x_{k+1} - x_k\|_S^2 \leq L(\widehat{x}_{k+1}, y_k) + \frac{1}{2}\|\widehat{x}_{k+1} - x_k\|_S^2. \tag{43}$$

(b) For all $k \geq 0$, we have that

$$L(x_{k+1}, y_{k+1}) + \frac{1}{2}\|x_{k+1} - x_k\|_S^2 + \frac{1}{2}\|y_{k+1} - y_k\|_T^2 \leq L(x_k, y_k). \tag{44}$$

Hence $\sum_{k=0}^{\infty}\|x_{k+1} - x_k\|_S^2 + \|y_{k+1} - y_k\|_T^2 < \infty$ and $\lim_{k\to\infty}\|x_{k+1} - x_k\| = 0 = \lim_{k\to\infty}\|y_{k+1} - y_k\|$.

(c) Let $\{(x_{k'}, y_{k'})\}$ be any convergent subsequence of $\{(x_k, y_k)\}$ with limit $(\bar{x}, \bar{y})$. Then $\lim_{k\to\infty} L(x_k, y_k) = \lim_{k\to\infty} L(x_{k+1}, y_k) = \lim_{k\to\infty} \widehat{L}(\widehat{x}_{k+1}; x_k, y_k) = L(\bar{x}, \bar{y})$. Furthermore $\lim_{k\to\infty}\|\widehat{x}_{k+1} - x_k\| = 0$.

(d) Let $\{(x_{k'}, y_{k'})\}$ be any convergent subsequence of $\{(x_k, y_k)\}$ with limit $(\bar{x}, \bar{y})$. Then $(\bar{x}, \bar{y})$ is a stationary point of L.

The full proof is given in the Appendix. Here we explain the four parts of Theorem 1. Part(a) establishes the non-increasing monotonicity of the proximal regularized update. Leveraging on part(a), part(b) ensures the existence of the limits. Using Part(a), (b) and (c), (d) then shows the critical point convergence of Algorithm 4.

4.6 Convex relaxation of (3) as initialization

Due to the non-convexity of the rank and ℓ_0 cardinality constraints, it is expected that the outcome of Algorithm 3 depends on initializations. A natural choice for the initialization of PARSuMi is the convex relaxation of both the rank and ℓ_0 function:

$$\min\left\{f(W, E) + \lambda\|W\|_* + \gamma\|E\|_1 \mid W \in \mathbb{R}^{m \times n}, E \in \mathbb{R}_\Omega^{m \times n}\right\} \tag{45}$$

where $f(W, E) = \frac{1}{2}\|H \circ (W + E - \widehat{W})\|^2$, $\|\cdot\|_*$ is the nuclear norm, and λ and γ are regularization parameters.

Problem (45) can be solved efficiently by the quadratic majorization-APG (accelerated proximal gradient) framework proposed by Toh and Yun (2010). At the kth iteration with iterate $(\bar{W}^k, \bar{E}^k)$, the majorization step replaces (45) with a quadratic majorization of $f(W, E)$,

so that W and E can be optimized independently, as we shall see shortly. Let $G^k = (H \circ H) \circ (\bar{W}^k + \bar{E}^k + \widehat{W})$. By some simple algebra, we have

$$
\begin{aligned}
f(W, E) - f(\bar{W}^k, \bar{E}^k) &= \frac{1}{2} \|H \circ (W - \bar{W}^k + E - \bar{E}^k)\|^2 \\
&\quad + \langle W - \bar{W}^k + E - \bar{E}^k, G^k \rangle \\
&\leq \|W - \bar{W}^k\|^2 + \|E - \bar{E}^k\|^2 + \langle W - \bar{W}^k + E - \bar{E}^k, G^k \rangle \\
&= \|W - \widetilde{W}^k\|^2 + \|E - \widetilde{E}^k\|^2 + \text{constant}
\end{aligned}
$$

where $\widetilde{W}^k = \bar{W}^k - G^k/2$ and $\widetilde{E}^k = \bar{E}^k - G^k/2$. At each step of the APG method, one minimizes (45) with $f(W, E)$ replaced by the above quadratic majorization. As the resulting problem is separable in W and E, we can minimize them separately, thus yielding the following two optimization problems:

$$
W^{k+1} = \operatorname{argmin} \frac{1}{2} \|W - \widetilde{W}^k\|^2 + \frac{\lambda}{2} \|W\|_* \qquad (46)
$$

$$
E^{k+1} = \operatorname{argmin} \frac{1}{2} \|E - \widetilde{E}^k\|^2 + \frac{\gamma}{2} \|E\|_1 \qquad (47)
$$

The main reason for performing the above majorization is because the solutions to (46) and (47) can readily be found with closed-form solutions. For (46), the minimizer is given by the Singular Value Thresholding (SVT) operator. For (47), the minimizer is given by the well-known soft thresholding operator (Donoho 1995). The APG algorithm, which is adapted from Beck and Teboulle (2009) and analogous to that in Toh and Yun (2010), is summarized below.

Algorithm 5 An APG algorithm for (45)

Input: Initialize $W^0 = \bar{W}^0 = 0$, $E^0 = \bar{E}^0 = 0$, $t_0 = 1$, $k = 0$
repeat
   1. Compute $G^k = (H \circ H) \circ (\bar{W}^k + \bar{E}^k + \widehat{W})$, $\widetilde{W}^k$, $\widetilde{E}^k$.
   2. Update W^{k+1} by applying the SVT on $\widetilde{W}^k$ in (46).
   3. Update E^{k+1} by applying the soft-thresholding operator on $\widetilde{E}^k$ in (47).
   4. Update step size $t_{k+1} = \frac{1}{2}(1 + \sqrt{1 + 4t_k^2})$.
   5. $(\bar{W}^{k+1}, \bar{E}^{k+1}) = (W^{k+1}, E^{k+1}) + \frac{t_k - 1}{t_{k+1}}(W^{k+1} - W^k, E^{k+1} - E^k)$
until Convergence
Output: Accumulation points $\overline{W}$ and $\overline{E}$

As has already been proved in Beck and Teboulle (2009), the APG algorithm, including the one above, has a very nice worst case iteration complexity result in that for any given $\epsilon > 0$, the APG algorithm needs at most $O(1/\sqrt{\epsilon})$ iterations to compute an ϵ-optimal (in terms of function value) solution.

The tuning of the regularization parameters λ and γ in (45) is fairly straightforward. For λ, we use the singular values of the converged $\overline{W}$ as a reference. Starting from a relatively large value of λ, we reduce it by a constant factor in each pass to obtain a $\overline{W}$ such that its singular values beyond the rth are much smaller than the first r singular values. For γ, we use the suggested value of $1/\sqrt{\max(m, n)}$ from RPCA (Candès et al 2011). In our experiments, we find that we only need a ballpark figure, without having to do a lot of tuning. Taking $\lambda = 0.2$ and $\gamma = 1/\sqrt{\max(m, n)}$ serve the purpose well.

4.7 Other heuristics

In practice, we design two heuristics to further boost the quality of the convex initialization. These are tricks that allow PARSuMi to detect corrupted entries better and are always recommended.

We refer to the first heuristic as "Huber Regression". The idea is that the quadratic loss term in our matrix completion step (14) is likely to result in a dense spread of estimation error across all measurements. There is no guarantee that those true corrupted measurements will hold larger errors comparing to the uncorrupted measurements. On the other hand, we note that the quality of the subspace N^k obtained from LM_GN is usually good despite noisy/corrupted measurements. This is especially true when the first LM_GN step is initialized with Algorithm 5. Intuitively, we should be better off with an intermediate step, using N^{k+1} to detect the errors instead of W^{k+1}, that is, keeping N^{k+1} as a fixed input and finding coefficient C and E simultaneously with

$$
\min_{E, C} \frac{1}{2} \|H \circ (N^{k+1} C - \widehat{W} + E)\|^2 \qquad (48)
$$
$$
\text{subject to} \quad \|E\|_0 \leq N_0.
$$

To make it computationally tractable, we relax (48) to

$$
\min_{E, C} \frac{1}{2} \|H \circ (N^{k+1} C - \widehat{W} + E)\|^2 + \eta_0 \|E\|_1 \qquad (49)
$$

where $\eta_0 > 0$ is a penalty parameter. Note that each column of the above problem can be decomposed into the following Huber loss regression problem (E is absorbed into the Huber penalty)

$$
\min_{C_j} \sum_{i=1}^{m} \text{Huber}_{\eta_0/H_{ij}}(H_{ij}((N^{k+1} C_j)_i - \widehat{W}_{ij})). \qquad (50)
$$

Since N^{k+1} is known, (49) can be solved very efficiently using the APG algorithm, whose derivation is similar to that of Algorithm 5, with soft-thresholding operations on C and E. To further reduce the Robin Hood effect (that haunts all ℓ_1-like penalties) and enhance sparsity, we may optionally apply the iterative re-weighted Huber minimization (a slight variation of the method in

Candès et al (2008)), that is, solving (50) for l_{max} iterations using an entrywise weighting factor inversely proportional to the previous iteration's fitting residual. In the end, the optimal columns C_j's are concatenated into the optimal solution matrix C^* of (49), and we set

$$W^{k+1} = N^{k+1}C^*.$$

With this intermediate step between the W step and the E step, it is much easier for the E step to detect the support of the actual corrupted entries.

The above procedure can be used in conjunction with another heuristic that avoids adding false positives into the corruption set in the E step when the subspace N has not yet been accurately recovered. This is achieved by imposing a threshold η on the minimum absolute value of E^k's non-zero entries, and shrink this threshold by a factor (say 0.8) in each iteration. The "Huber regression" heuristic is used only when $\eta > \eta_0$, and hence only in a very small number of iteration before the support of E has been reliably recovered. Afterwards the pure PARSuMi iterations (without the Huber step) will take over, correct the Robin Hood effect of Huber loss and then converge to a high quality solution.

Note that our critical point convergence guarantee in Section 4.5 is not hampered at all by the two heuristics, since after a small number of iterations, $\eta \leq \eta_0$ and we come back to the pure PARSuMi.

5 Experiments and discussions

In this section, we present the methodology and results of various experiments designed to evaluate the effectiveness of our proposed method. The experiments revolve around synthetic data and two real-life datasets: the Oxford Dinosaur sequence, which is representative of data matrices in SfM works, and the Extended YaleB face dataset (Lee et al 2005), which we use to demonstrate how PARSuMi works on photometric stereo problems.

In the synthetic data experiments, our method is compared with the state-of-the-art algorithms for the objective function in (10) namely Wiberg ℓ_1 (Eriksson and Van Den Hengel 2010) and GRASTA (He et al 2011). ALP and AQP (Ke and Kanade 2005) are left out since they are shown to be inferior to Wiberg ℓ_1 in Eriksson and Van Den Hengel (2010). For the sake of comparison, we perform the experiment on recovery effectiveness using the same small matrices as in Section 5.1 of Eriksson and Van Den Hengel (2010). Other synthetic experiments are conducted with more reasonably-sized matrices. Whenever appropriate, we

also include a comparison to a variant of RPCA that handles missing data (Wu et al 2011b) which solves (6) using the augmented Lagrange multiplier (ALM) algorithm (we will call it ALM-RPCA from here onwards). This serves as a representative of the nuclear norm based methods.

The real data from the SfM and photometric stereo problems contain many challenges typical in practical scenarios. They contain large contiguous areas of missing data, and potentially highly corrupted observations which may not be sparse too. For instance, in the YaleB face dataset, grazing illumination tends to produce large area of missing data (well over 50%) and often large number of outliers too (due to specular highlights). The PARSuMi method outperformed a variety of other methods in the experiments, even uncovering hitherto unknown corruptions inherent in the Dinosaur data from SfM. The results also corroborate those obtained in the synthetic data experiments, in that our method can handle a substantially larger fraction of missing data and corruptions, thus providing empirical evidence for the efficacy of PARSuMi under practical scenarios.

For a summary of the parameters used in the experiments, please refer to the Appendix.

5.1 Convex Relaxation as an Initialization Scheme

We first investigate the results of our convex initialization scheme by testing on a randomly generated 100×100 rank-4 matrix. A random selection of 70% and 10% of the entries are considered missing and corrupted respectively. Corruptions are generated by adding large uniform noise between $[-1, 1]$. In addition, Gaussian noise $\mathcal{N}(0, \sigma)$ for $\sigma = 0.01$ is added to all observed entries. From Fig. 7, we see that the convex relaxation outlined in Section 4.6 was able to recover the error support, but there is considerable difference in magnitude between the recovered error and the ground truth, owing to the "Robin Hood" attribute of ℓ_1-norm as a convex proxy of ℓ_0. Nuclear norm as a proxy of rank also suffers from the same woe. Similar observations can be made on the results of the Dinosaur experiments, which we will show later.

Despite the problems with the solution of the convex initialization, we find that it is a crucial step for PARSuMi to work well in practice. As can be seen from Fig. 7, the detected error support can be quite accurate. This makes the E-step of PARSuMi more likely to identify the true locations of corrupted entries.
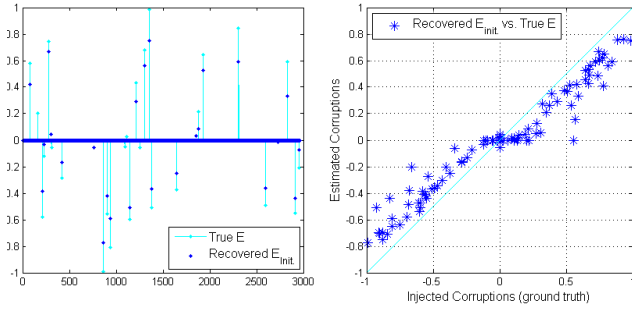
Fig. 7 The Robin Hood effect of Algorithm 5 on detected sparse corruptions E_{Init}. **Left**: illustration of a random selection of detected E vs. true E. Note that the support is mostly detected, but the magnitude falls short. **Right**: scatter plot of the detected E against true E (perfect recovery falls on the $y = x$ line, false positives on the y-axis and false negatives on the x-axis).
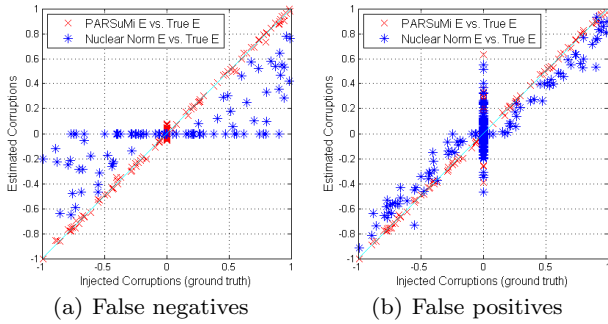

(a) False negatives (b) False positives

Fig. 8 Recovery of corruptions from poor initialization.

5.2 Impacts of poor initialization

When the convex initialization scheme fails to obtain the correct support of the error, the "Huber Regression" heuristic may help PARSuMi to identify the support of the corrupted entries. We illustrate the impact by intentionally mis-tuning the parameters of Algorithm 5 such that the initial E bears little resemblance to the true injected corruptions. Specifically, we test the cases when the initialization fails to detect many of the corrupted entries (false negatives) and when many entries are wrongly detected as corruptions (false positives). From Fig. 8, we see that PARSuMi is able to recover the corrupted entries to a level comparable to the magnitude of the injected Gaussian noise in both experiments[9].

---

[9] Note that a number of false positives persist in the second experiment. This is understandable because false positives often contaminate an entire column or row, making it impossible to recover that column/row in later iterations even if the subspace is correctly detected. To avoid such an undesirable situation, we prefer "false negatives" over "false positives" when tuning Algorithm 5. In practice, it suffices to keep the initial E relatively sparse.
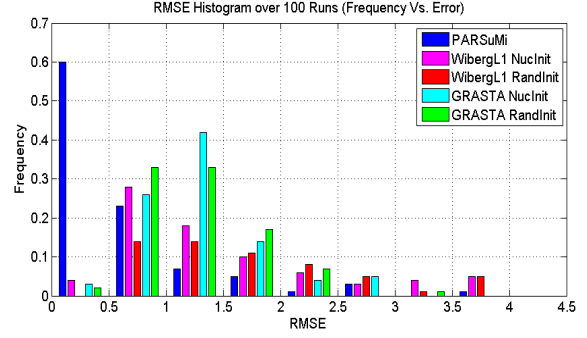

Fig. 9 A histogram representing the frequency of different magnitudes of RMSE in the estimates generated by each method.

In most of our experiments, we find that PARSuMi is often able to detect the corruptions perfectly from a simple initializations with all zeros, even without the "Huber Regression" heuristic. This is especially true when the data are randomly generated with benign sampling pattern and well-conditioned singular values. However, in challenging applications such as SfM, a good convex initialization and the "Huber Regression" heuristic are always recommended.

5.3 Recovery effectiveness from sparse corruptions

For easy benchmarking, we use the same synthetic data in Section 5.1 of Eriksson and Van Den Hengel (2010) to investigate the quantitative effectiveness of our proposed method. A total of 100 random low-rank matrices with missing data and corruptions are generated and tested using PARSuMi, Wiberg ℓ_1 and GRASTA.

In accordance with Eriksson and Van Den Hengel (2010), the ground truth low rank matrix $W \in \mathbb{R}^{m \times n}, m = 7, n = 12, r = 3$, is generated as $W = UV^T$, where $U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}$ are generated using uniform distribution, in the range [-1,1]. 20% of the data are designated as missing, and 10% are added with corruptions, both at random locations. The magnitude of the corruptions follows a uniform distribution $[-5, 5]$. Root mean square error (RMSE) is used to evaluate the recovery precision:

$$\text{RMSE} := \frac{\|W_{\text{recovered}} - W\|_F}{\sqrt{mn}}. \tag{51}$$

Out of the 100 independent experiments, the number of runs that returned RMSE values of less than 5 are 100 for PARSuMi, 78 and 58 for Wiberg ℓ_1 (with two different initializations) and similarly 94 and 93 for GRASTA. These are summarized in Fig. 9.

5.4 Recovery under varying level of corruptions, missing data and noise

To gain a holistic understanding of our proposed method, we perform a series of systematically parameterized experiments on 40×60 rank-4 matrices (with the elements of the factors U, V drawn independently from the uniform distribution on $[-1, 1]$), with conditions ranging from 0-80% missing data, 0-20% corruptions of range [-2,2], and Gaussian noise with σ in the range [0,0.1]. By fixing the Gaussian noise at a specific level, the results are rendered in terms of phase diagrams showing the recovery precision as a function of the missing data and outliers. The precision is quantified as the difference between the recovered RMSE and the oracle bound RMSE [10]. As can be seen from Fig. 10, our algorithm obtains near optimal performance at an impressively large range of missing data and outlier at $\sigma = 0.01$[11].

For comparison, we also displayed the results for closely related methods, e.g., ALM-RPCA (Wu et al 2011b), GRASTA (He et al 2011), DRMF (Xiong et al 2011), LM_GN (Chen 2011) as well as Algorithm 5 (our initialization). Wiberg ℓ_1 is omitted because it is too slow. Among all the methods we compared, PARSuMi is able to successfully reconstruct the largest range of matrices with almost optimal numerical accuracy. Also, the results for DRMF and LM_GN are well-expected since they are not designed to handle both missing data and outliers.

5.5 SfM with missing and corrupted data on Dinosaur

In this section, we apply PARSuMi to the problem of SfM using the Dinosaur sequence and investigate how well the corrupted entries can be detected and recovered in real data. We have normalized image pixel dimensions (width and height) to be in the range [0,1]; all plots, unless otherwise noted, are shown in the normalized coordinates.

To simulate data corruptions arising from wrong feature matches, we randomly add sparse error of the range [-2,2][12] to 1% of the sampled entries. This is a
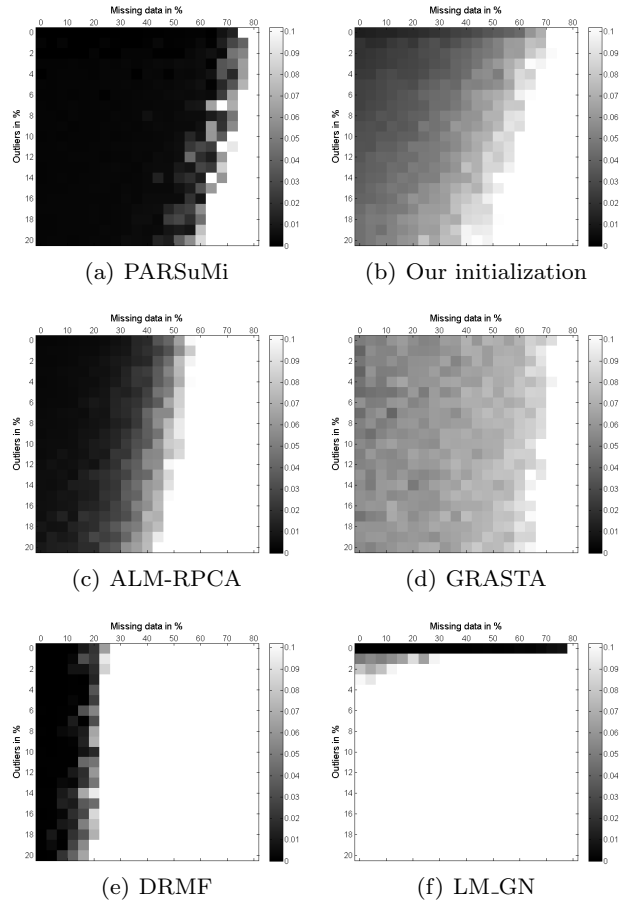

(a) PARSuMi

(b) Our initialization

(c) ALM-RPCA

(d) GRASTA

(e) DRMF

(f) LM_GN

Fig. 10 Phase diagrams (darker is better) of RMSE with varying proportion of missing data and corruptions with Gaussian noise $\sigma = 0.01$.

	PARSuMi	Wiberg ℓ_1	GRASTA
No. of success	9/10	0/10	0/10
Run time (mins): min/avg/max	2.2/2.9/5.2	76/105/143	0.2/0.5/0.6
Min RMSE (original pixel unit)	1.454	2.715	22.9
Min RMSE excluding corrupted entries	0.3694	1.6347	21.73

Table 3 Summary of the Dinosaur experiments. Note that because there is no ground truth for the missing data, the RMSE is computed only for those observed entries as in Buchanan and Fitzgibbon (2005).

more realistic (and much larger[13]) definition of outliers for SfM compared to the [-50,50] pixel range used to evaluate Wiberg ℓ_1 in Eriksson and Van Den Hengel (2010).

We conducted the experiment 10 times each for PAR-SuMi, Wiberg ℓ_1 (with SVD initialization) and GRASTA (random initialization as recommended in the original paper) and count the number of times they succeed.

---

[10] See the Appendix for details.

[11] The phase diagrams for other levels of noise look very much like Fig. 10; we therefore did not include them in the paper.

[12] In SfM data corruptions are typically matching failures. Depending on where true matches are, error induced by a matching failure can be arbitrarily large. If we constrain true match to be inside image frame [0, 1](which is often not the case), then the maximum error magnitude is 1. We found it appropriate to at least double the size to account for general matching failures in SfM, hence [−2, 2].

[13] [-50,50] in pixel is only about [-0.1,0.1] in our normalized data, which could hardly be regarded as "gross" corruptions.

As there are no ground truth to compare against, we cannot use the RMSE to evaluate the quality of the filled-in entries. Instead, we plot the feature trajectory of the recovered data matrix for a qualitative judgement. As is noted in Buchanan and Fitzgibbon (2005), a correct recovery should consist of all elliptical trajectories. Therefore, if the recovered trajectories look like that in Fig. 6(b), we count the recovery as a success.
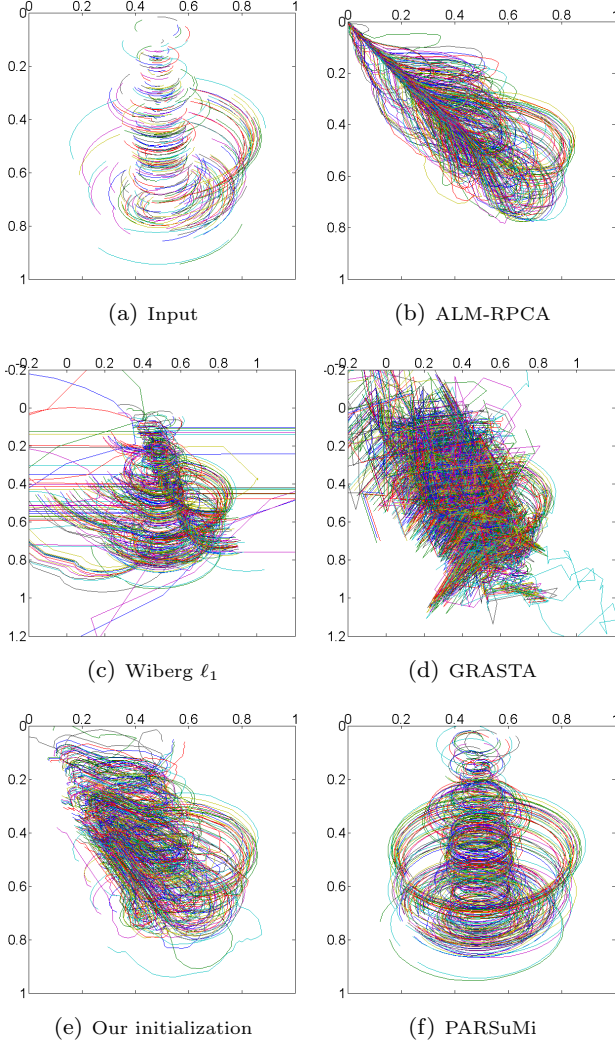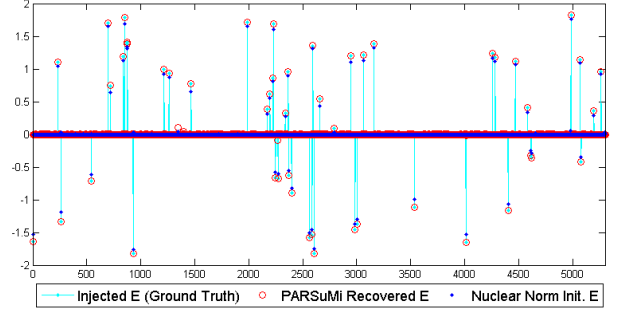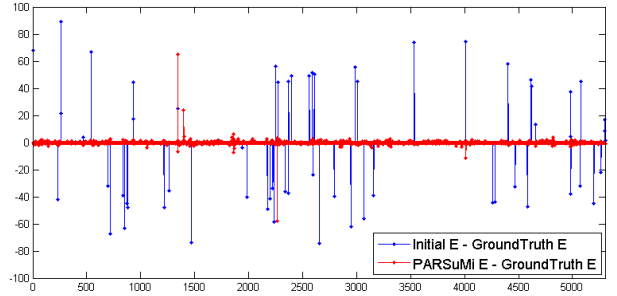

(a) Initialization via Algorithm 5 and the final recovered errors by PARSuMi (Algorithm 3)

(b) Difference of the recovered and ground truth error (in original pixel unit)

Fig. 12 Sparse corruption recovery in the Dinosaur experiments: The support of all injected outliers are detected by Algorithm 5 (see (a)), but the magnitudes fall short by roughly 20% (see (b)). Algorithm 3 is able to recover all injected sparse errors, together with the inherent tracking errors in the dataset (see the red spikes in (b)).

(a) Input   (b) ALM-RPCA

(c) Wiberg ℓ_1 (d) GRASTA

(e) Our initialization   (f) PARSuMi

Fig. 11 Comparison of recovered feature trajectories with different methods. It is clear that under dense noise and gross outliers, neither convex relaxation nor ℓ_1 error measure yields satisfactory results. Solving the original non-convex problem with (e) as an initialization produces a good solution.

The results are summarized in Table 3. Notably, PARSuMi managed to correctly detect the corrupted entries and fill in the missing data in 9 runs while Wiberg ℓ_1 and GRASTA failed on all 10 attempts. Typical feature trajectories recovered by each method are shown in Fig. 11. Note that only PARSuMi is able to recover the elliptical trajectories satisfactorily.

For comparison, we also include the input (partially observed trajectories) and the results of our convex initialization in Fig. 11(a) and 11(e) respectively.

An interesting and somewhat surprising finding is that the result of PARSuMi is even better than the global optimal solution for data containing supposedly no corruptions (and thus can be obtained with ℓ_2 method) (see Fig. 6(b), which is obtained under no corruptions in the observed data)! In particular, the trajectories are now closed.

The reason becomes clear when we look at Fig. 12(b), which shows two large spikes in the vectorized difference between the artificially injected corruptions and the recovered corruptions by PARSuMi. This suggests that there are hitherto unknown corruptions inherent in the Dinosaur data. We trace the two large ones into the raw images, and find that they are indeed data corruptions corresponding to mismatched feature points from the original dataset; our method managed to recover the correct feature matches (left column of Fig. 13).
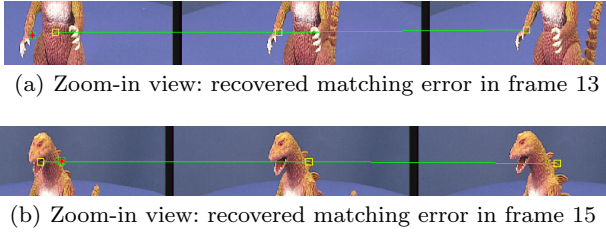
(a) Zoom-in view: recovered matching error in frame 13

(b) Zoom-in view: recovered matching error in frame 15

Fig. 13 Original tracking errors in the Dinosaur data identified (yellow box) and corrected by PARSuMi (green box with red star) in frame 13 feature 86 (a) and frame 15 feature 144 (b).

The result shows that PARSuMi recovered not only the artificially added errors, but also the intrinsic errors in the data set. In Buchanan and Fitzgibbon (2005), it was observed that there is a mysterious increase of the objective function value upon closing the trajectories by imposing orthogonality constraint on the factorized camera matrix. Our discovery of these intrinsic tracking errors explained this matter evidently. It is also the reason why the ℓ_2-based algorithms (see Fig. 6(b)) find a global minimum solution that is of poorer quality (trajectories fail to close loop).

To complete the story, we generated the 3D point cloud of Dinosaur with the completed data matrix. The results viewed from different directions are shown in Fig. 14.
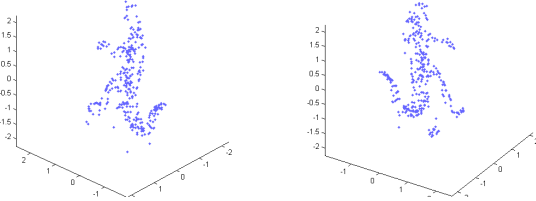

Fig. 14 3D point cloud of the reconstructed Dinosaur.

5.6 Photometric Stereo

Another intuitive application for PARSuMi is photometric stereo, a problem of reconstructing the 3D shape of an object from images taken under different lighting conditions. In the most ideal case of Lambertian surface model (diffused reflection), the data matrix obtained by concatenating vectorized images together is of rank 3.

Real surfaces are of course never truly Lambertian. There are usually some localized specular regions appearing as highlights in the image. Moreover, since there is no way to obtain a negative pixel value, all negative

Fig. 15 Illustration of the synthetic data and their surface normal. Note that there are specular regions and shadows.

inner products will be observed as zero. This is the so-called attached shadow. Images of non-convex object often also contain cast shadow, due to the blocking of light path. If these issues are teased out, then the seemingly naive Lambertian model is able to approximate many surfaces very well.

Wu et al (2011b) subscribed to this low-rank factorization model and proposed to model all dark regions as missing data, all highlights as sparse corruptions and then use a variant of RPCA (identical to (6)) to recover the full low-rank matrix. The solution however is only tested on noise-free synthetic data and toy-scale real examples. Del Bue et al (2012) applied their BALM on photometric stereo too, attempting on both synthetic and real data. Their contribution is to impose the normal constraint of each normal vector during the optimization.

We compare PARSuMi with the aforementioned two methods on several photometric stereo datasets. Quantitatively, we use the Caesar and Elephant data in Wu et al (2011b) and compare the reconstructed surface normal against the ground truth. The data is illustrated in Fig. 15 and the comparison is detailed in Table 4. As we can see, PARSuMi has the smallest reconstruction error among the three methods in all experiments.

We also conducted a qualitative comparison of the methods on a real-life data using Subject 3 in the Extended YaleB dataset since it was initially used to evaluate BALM in Del Bue et al (2012)[14]. As we do not have any ground truth, we can only compare the reconstruction qualitatively.

From Fig. 16, we can clearly see that PARSuMi is able to recover the missing pixels in the image much better than the other two methods. In particular, Fig. 16(a) and 16(b) shows that PARSuMi's reconstruction (in the illuminated half of the face) has fewest artifacts. This can be seen from the unnatural grooves that the red arrows point to in Fig. 16(b). Moreover, we know from the original image that the light comes from the right-hand-side of the subject; thus all the pixels on the left side of

---

[14] The authors claimed that it is Subject 10 (Del Bue et al 2012, Figure 9), but careful examination of all faces shows that it is in fact Subject 3.

Dataset	PARSuMi	ALM-RPCA (Wu et al 2011b)	BALM (Del Bue et al 2012)	Oracle (a lower bound)
Elephant	**7.13e-2** (16.7 min)	7.87e-2 (1.1 min)	3.55 (1.1 min)	model error
Caesar	**1.83e-1** (28.6 min)	2.71e-1 (7.2 min)	3.11 (5.2 min)	model error
Elephant $+ \mathcal{N}(0, 0.05)$	**2.35** (28.3 min)	2.62 (1.5 min)	4.37 (1.1 min)	1.70 + model error
Caesar $+ \mathcal{N}(0, 0.05)$	**2.34** (99.2 min)	2.53 (8.3 min)	4.06 (6.6 min)	1.73 + model error

Table 4 Angular error (in degree) and runtime (in minutes) comparison for the synthetic data photometric stereo experiments. The lowest estimation error is labeled in boldface. The oracle column gives the information-theoretic limit, it depends on an unknown model error as we are using a Lambertian model to deal with the data rendered by the Cook-Torrence model.

his face (e.g. the red ellipse area in Fig. 16(b)) should have negative filled-in values and therefore should be dark in the image. Neither BALM nor ALM-RPCA's reconstructed images comply to this physical law.

To see this more clearly, we invert the pixel values of the reconstructed image in Fig. 16(c). This is equivalent to inverting the direction of lighting. From the tag of the image, we know that the original lighting is $-20°$ from the subject's right posterior and $40°$ from the top, so the inverted light should illuminate the left half of his face from $20°$ left frontal and $40°$ from below. As is shown in the comparison, only PARSuMi's result revealed what should be correctly seen with a light shining from this direction.

In addition, we reconstruct the 3D depth map with the classic method by Horn (1990). In Fig. 16(d), the shape from PARSuMi reveals much richer depth information than those from the other two algorithms, whose reconstructions appear flattened. This is a known low-frequency bias problem for photometric stereo and it is often caused by errors in the surface normal estimation Nehab et al (2005). The fact that BALM and ALM-RPCA produces a flatter reconstruction is a strong indication that their estimations of the surface normal are noisier than that of PARSuMi.

From our experiments, we find that PARSuMi is able to successfully reconstruct the 3D face for all 38 subjects with little artifacts. As illustrated in Fig. 17, our 3D reconstructions of the features seem to reveal the characteristic features of subjects across different ethnic groups. Moreover, due to the robust ℓ_0 penalty, PARSuMi is able to effectively recover the input images from many different types of irregularities, e.g. specular regions, different facial expressions, or even image corruptions caused by hardware malfunctions (see Fig. 18 and 20). This makes it possible for PARSuMi to be integrated reliably into engineering systems that function with minimal human interactions [15].

---

[15] For the best of our knowledge, all previous works that use this dataset for photometric 3D reconstruction manually removed a number of images of poor qualities, e.g.Del Bue et al (2012)
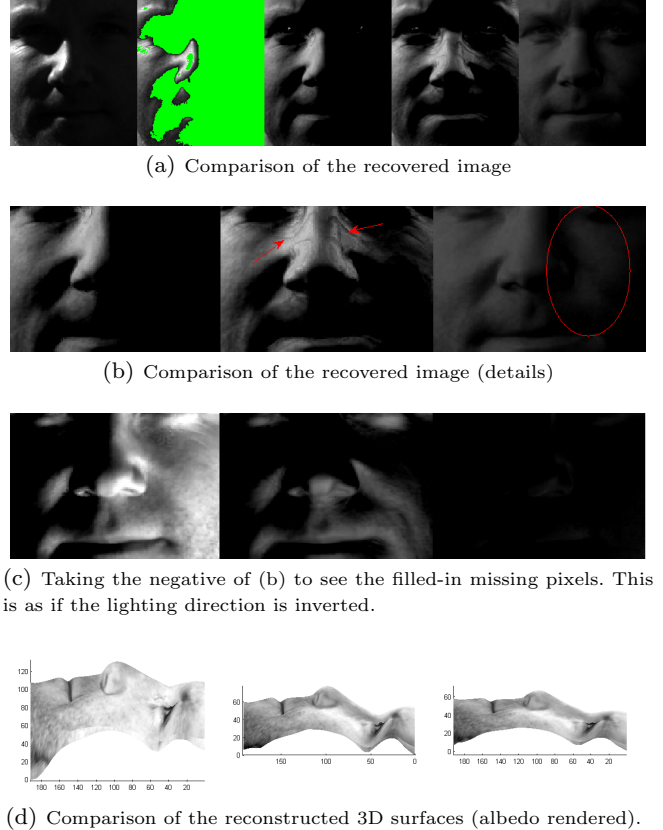

(a) Comparison of the recovered image

(b) Comparison of the recovered image (details)

(c) Taking the negative of (b) to see the filled-in missing pixels. This is as if the lighting direction is inverted.

(d) Comparison of the reconstructed 3D surfaces (albedo rendered).

Fig. 16 Qualitative comparison of algorithms on Subject 3. From left to right, the results are respectively for PARSuMi, BALM and ALM-RPCA. In (a), they are preceded by the original image and the image depicting the missing data in green.

5.7 Speed

The computational complexity of PARSuMi is cheap for some problems but not for others. Since PARSuMi uses LM_GN for its matrix completion step, the numerical cost is dominated by either solving the linear system $(J^T J + \lambda I)\delta = J\mathbf{r}$ which requires the Cholesky factorization of a potentially dense $mr \times mr$ matrix, or the computation of J which requires solving a small linear system of normal equation involving the $m \times r$ matrix N for n times. As the overall complexity of $O(\max(m^3 r^3, mnr^2))$ scales merely linearly with num-

(a) Subject 02

(b) Subject 5

(c) Subject 10

(d) Subject 15
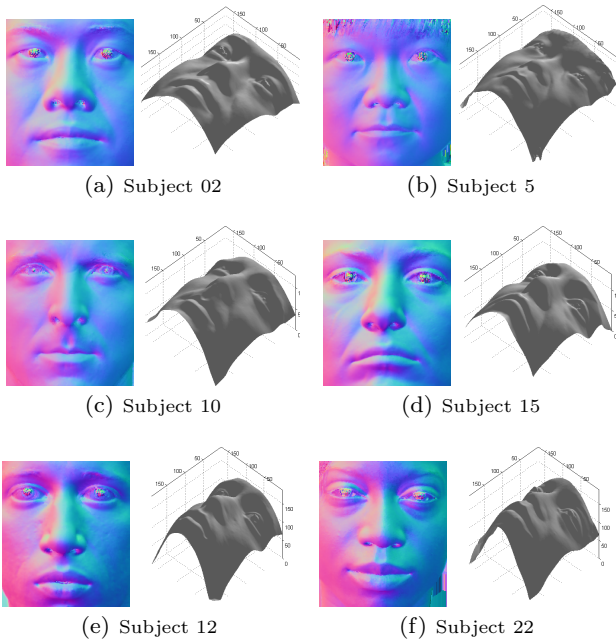
(e) Subject 12

(f) Subject 22

Fig. 17 The reconstructed surface normal and 3D shapes for Asian (first row), Caucasian (second row) and African (third row), male (first column) and female (second column), in Extended YaleB face database.(Zoom-in to look at details)
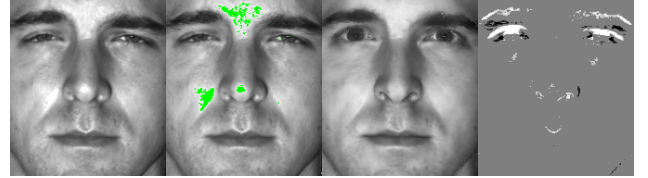
ber of columns n but cubic with m and r, PARSuMi is computationally attractive when solving problems with small m and r, and potentially large n, e.g., photometric stereo and SfM (since the number of images is usually much smaller than the number of pixels and feature points). However, for a typical million by million data matrix as in social networks and collaborative filtering, PARSuMi will take an unrealistic amount of time to run.

Experimentally, we compare the runtime between our algorithm and Wiberg ℓ_1 method in our Dinosaur experiment in Section 5.5. Our Matlab implementation is run on a 64-bit Windows machine with a 1.6 GHz Core i7 processor and 4 GB of memory. We see from Table 3 that there is a big gap between the speed performance. The near 2-hour runtime for Wiberg ℓ_1 is discouragingly slow, whereas ours is vastly more efficient. On the other hand, as an online algorithm, GRASTA is inherently fast. Examples in He et al (2011) show that it works in real time for live video surveillance. However, our experiment suggests that it is probably not appropriate for applications such as SfM, which requires a higher numerical accuracy.
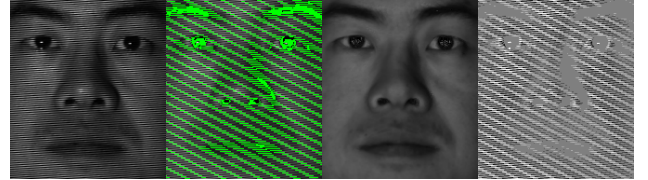
The runtime comparison for the photometric stereo problems is shown in Table 4. We remark that PAR-SuMi is roughly ten times slower than other methods. The pattern is consistent for the YaleB face data too, where PARSuMi takes 23.4 minutes to converge while
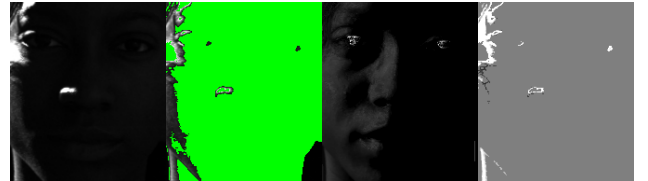

(a) Cast shadow and attached shadow are recovered. Region of cast shadow is now visible, and attached shadow is also filled with meaningful negative values.

(b) Facial expressions are set to normal.

(c) Rare corruptions in image acquisition are recovered.

(d) Light comes 20 degrees from behind and 65 degrees from above).

Fig. 18 Illustrations of how PARSuMi recovers missing data and corruptions. From left to right: original image, input image with missing data labeled in green, reconstructed image and detected sparse corruptions.

BALM and RPCA takes only 4.8 and 1.7 minutes respectively.

We note that PARSuMi is currently not optimized for computation. Speeding up the algorithm for application on large scale dataset would require further effort (such as parallelization) and could be a new topic of research. For instance, the computation of Jacobians J_i and the evaluation of the objective function can be easily done in parallel and the Gauss-Newton update (a positive definite linear system of equations) can be solved using the conjugate gradient method; hence, we do not even need to store the matrix in memory. Furthermore, since PARSuMi seeks to find the best subspace, perhaps using only a small portion of the data columns is sufficient. If the subspace is correct, the rest of the columns can be recovered in linear time with our iterative reweighted Huber regression technique (see Section 4.7). A good direction for future re-

search is perhaps on how to choose the best subset of data to feed into PARSuMi.

6 Conclusion

In this paper, we have presented a practical algorithm (PARSuMi) for low-rank matrix completion in the presence of dense noise and sparse corruptions. Despite the non-convex and non-smooth optimization formulation, we are able to derive a set of update rules under the proximal alternating scheme such that the convergence to a critical point can be guaranteed. The method was tested on both synthetic and real life data with challenging sampling and corruption patterns. The various experiments we have conducted show that our method is able to detect and remove gross corruptions, suppress noise and hence provide a faithful reconstruction of the missing entries. By virtue of the explicit constraints on both the matrix rank and cardinality, and the novel reformulation, design and implementation of appropriate algorithms for the non-convex and non-smooth model, our method works significantly better than the state-of-the-art algorithms in nuclear norm minimization, ℓ_2 matrix factorization and ℓ_1 robust matrix factorization in real life problems such as SfM and photometric stereo.

Moreover, we have provided a comprehensive review of the existing results pertaining to the "practical matrix completion" problem that we considered in this paper. The review covered the theory of matrix completion and corruption recovery, and the theory and algorithms for matrix factorization. In particular, we conducted extensive numerical experiments which reveals (a) the advantages of matrix factorization over nuclear norm minimization when the underlying rank is known, and (b) the two key factors that affect the chance of ℓ_2-based factorization methods reaching global optimal solutions, namely "subspace parameterization" and "Gauss-Newton" update. These findings provided critical insights into this difficult problem, upon the basis which we developed PARSuMi as well as its convex initialization.

The strong empirical performance of our algorithm calls for further analysis. For instance, obtaining the theoretical conditions for the convex initialization to yield good support of the corruptions should be plausible (following the line of research discussed in Section 2.1), and this in turn guarantees a good starting point for the algorithm proper. Characterizing how well the following non-convex algorithm works given such initialization and how many samples are required to guarantee high-confidence recovery of the matrix remain open questions for future study.

Other interesting topics include finding a cheaper but equally effective alternative to the LM_GN solver for solving (20), parallel/distributed computation, incorporating additional structural constraints, selecting optimal subset of data for subspace learning and so on. Step by step, we hope this will eventually lead to a practically working robust matrix completion algorithm that can be confidently embedded in real-life applications.

A Appendix

A.1 Proofs

Proof (Proof of Proposition 4) Given a subset I of $\{1, \ldots, |\Omega|\}$ with cardinality at most N_0 such that $b_I \neq 0$. Let $J = \{1, \ldots, |\Omega|\} \backslash I$. Consider the problem (34) for $x \in \mathbb{R}^{|\Omega|}$ supported on I, we get the following:

$$v_I := \min_{x_I} \left\{ \|x_I - b_I\|^2 + \|b_J\|^2 \mid \|x_I\|^2 - K_E^2 \leq 0 \right\},$$

which is a convex minimization problem whose optimality conditions are given by

$$x_I - b_I + \mu x_I = 0, \ \mu(\|x_I\|^2 - K_E^2) = 0, \ \mu \geq 0$$

where μ is the Lagrange multiplier for the inequality constraint. First consider the case where $\mu > 0$. Then we get $x_I = K_E b_I / \|b_I\|$, and $1 + \mu = \|b_I\| / K_E$ (hence $\|b_I\| > K_E$). This implies that $v_I = \|b\|^2 + K_E^2 - 2\|b_I\|K_E$. On the other hand, if $\mu = 0$, then we have $x_I = b_I$ and $v_I = \|b_J\|^2 = \|b\|^2 - \|b_I\|^2$. Hence

$$v_I = \begin{cases} \|b\|^2 + K_E^2 - 2\|b_I\|K_E & \text{if } \|b_I\| > K_E \\ \|b\|^2 - \|b_I\|^2 & \text{if } \|b_I\| \leq K_E. \end{cases}$$

In both cases, it is clear that v_I is minimized if $\|b_I\|$ is maximized. Obviously $\|b_I\|$ is maximized if I is chosen to be the set of indices corresponding to the N_0 largest components of b. □

Proof (Proof of Theorem 1) (a) The equality in (42) follows directly from (41). By the minimal property of $\widehat{x}_{k+1}$, we have that

$$\widehat{L}(\widehat{x}_{k+1}; x_k, y_k) \leq \widehat{L}(\xi; x_k, y_k) \quad \forall \xi \in \mathcal{X}. \tag{52}$$

Thus when $\xi = x_k$, we get $\widehat{L}(\widehat{x}_{k+1}; x_k, y_k) \leq \widehat{L}(x_k; x_k, y_k) = L(x_k, y_k)$, and the required inequality in (42) follows. On the other hand, the inequality (43) follows readily from the definition of x_{k+1}.

(b) If $x_{k+1} = \widetilde{x}_{k+1}$, then from the definition of x_{k+1} and (42), we have that,

$$L(x_{k+1}, y_k) + \frac{1}{2}\|x_{k+1} - x_k\|_S^2$$
$$\leq L(\widehat{x}_{k+1}, y_k) + \frac{1}{2}\|\widehat{x}_{k+1} - x_k\|_S \leq L(x_k, y_k). \tag{53}$$

On the other hand, if $x_{k+1} = \widehat{x}_{k+1}$, we have from (42) that

$$L(x_{k+1}, y_k) + \frac{1}{2}\|x_{k+1} - x_k\|_S^2 \leq L(x_k, y_k). \tag{54}$$

By the minimal property of y_{k+1}, we have that $\forall\, \eta \in \mathcal{Y}$

$$L(x_{k+1}, y_{k+1}) + \frac{1}{2}\|y_{k+1} - y_k\|_T^2 \leq L(x_{k+1}, \eta) + \frac{1}{2}\|\eta - y_k\|_T^2.$$

In particular, when $\eta = y_k$, we get

$$L(x_{k+1}, y_{k+1}) + \frac{1}{2}\|y_{k+1} - y_k\|_T^2 \leq L(x_{k+1}, y_k). \quad (55)$$

By combining (53)-(54) and (55), we get the inequality (44).

(c) Note that by using the result in part (b), we also have $\lim_{k'\to\infty} x_{k'+1} = \bar{x}$ and $\lim_{k'\to\infty} y_{k'+1} = \bar{y}$. From (42), (43) and (52), we have $\forall\, k \geq 0$, $\xi \in \mathcal{X}$

$$L(x_{k+1}, y_k) + \frac{1}{2}\|x_{k+1} - x_k\|_S^2 \leq \widehat{L}(\xi; x_k, y_k). \quad (56)$$

Thus $\forall\, \xi \in \mathcal{X}$

$$\limsup_{k'\to\infty} f(x_{k'+1}) + q(\bar{x}, \bar{y}) \leq f(\xi) + Q(\xi; \bar{x}, \bar{y}). \quad (57)$$

By taking $\xi = \bar{x}$, we get

$$\limsup_{k'\to\infty} f(x_{k'+1}) \leq f(\bar{x}) + Q(\bar{x}; \bar{x}, \bar{y}) - q(\bar{x}, \bar{y}) = f(\bar{x}). \quad (58)$$

On the other hand, since f is lower semicontinuous, we have that $\liminf_{k'\to\infty} f(x_{k'+1}) \geq f(\bar{x})$. Thus $\lim_{k'\to\infty} f(x_{k'+1}) = f(\bar{x})$. Similarly, we can show that $\lim_{k'\to\infty} g(y_{k'+1}) = g(\bar{y})$. As a result, we have

$$\lim_{k'\to\infty} L(x_{k'+1}, y_{k'+1}) = L(\bar{x}, \bar{y}). \quad (59)$$

Since $\{L(x_k, y_k)\}$ is a nonincreasing sequence, the above result implies that

$$\lim_{k\to\infty} L(x_k, y_k) = L(\bar{x}, \bar{y}) = \inf_k L(x_k, y_k).$$

Also, (53)-(54) and (55) implies that

$$\lim_{k\to\infty} L(x_{k+1}, y_k) = L(\bar{x}, \bar{y}).$$

From (42) and (43), we have

$$L(x_{k+1}, y_k) + \frac{1}{2}\|x_{k+1} - x_k\|_S^2$$
$$\leq \widehat{L}(\widehat{x}_{k+1}; x_k, y_k) \leq L(x_k, y_k).$$

Thus $\lim_{k\to\infty} \widehat{L}(\widehat{x}_{k+1}; x_k, y_k) = L(\bar{x}, \bar{y})$.
Now by (42) and (43) again, we have

$$\frac{1}{2}\|x_{k+1} - x_k\|_S^2 + \frac{1}{2}\|\widehat{x}_{k+1} - x_k\|_{M-A^*A-S}^2$$
$$\leq \widehat{L}(\widehat{x}_{k+1}; x_k, y_k) - L(x_{k+1}, y_k).$$

Thus $\lim_{k\to\infty} \|\widehat{x}_{k+1} - x_k\|_{M-A^*A-S}^2 = 0$. Since $M - A^*A - S \succ 0$, we also get $\lim_{k\to\infty} \|\widehat{x}_{k+1} - x_k\|^2 = 0$.

(d) From the optimality of $\widehat{x}_{k+1}$, we have that

$$0 \in \partial\widehat{L}(\widehat{x}_{k+1}; x_k, y_k)$$
$$= \partial f(\widehat{x}_{k+1}) + A^*(Ax_k + By_k - c) + M(\widehat{x}_{k+1} - x_k)$$
$$= \partial f(\widehat{x}_{k+1}) + A^*(A\widehat{x}_{k+1} + By_{k+1} - c) - \Delta x_{k+1}$$

where $\Delta x_{k+1} = -(M - A^*A)(\widehat{x}_{k+1} - x_k) - A^*B(y_k - y_{k+1})$. Thus

$$\Delta x_{k+1} \in \partial_x L(\widehat{x}_{k+1}, y_{k+1}). \quad (60)$$

From the optimality of y_{k+1}, we have that

$$0 \in \partial g(y_{k+1}) + B^*(Ax_{k+1} + By_{k+1} - c) + T(y_{k+1} - y_k)$$
$$= \partial_y L(\widehat{x}_{k+1}, y_{k+1}) + T(y_{k+1} - y_k) + B^*A(x_{k+1} - \widehat{x}_{k+1})$$

Hence $\Delta y_{k+1} := -T(y_{k+1} - y_k) - B^*A(x_{k+1} - \widehat{x}_{k+1}) \in \partial_y L(\widehat{x}_{k+1}, y_{k+1})$.

From part (b) and (c), we have that

$$\lim_{k'\to\infty} \|\widehat{x}_{k'+1} - x_{k'}\| = 0, \quad \lim_{k'\to\infty} \|\widehat{x}_{k'+1} - x_{k'+1}\| = 0,$$
$$\lim_{k'\to\infty} \widehat{x}_{k'+1} = \bar{x}, \quad \lim_{k'\to\infty} y_{k'+1} = \bar{y}.$$

Thus

$$\lim_{k'\to\infty} \Delta x_{k'+1} = 0 = \lim_{k'\to\infty} \Delta y_{k'+1}.$$

By the closedness property of ∂L (Clarke 1990, Proposition 2.1.5), we get

$$(0, 0) \in \partial L(\bar{x}, \bar{y}).$$

Thus $(\bar{x}, \bar{y})$ is a stationary point of L. □

A.2 Software/code used

The point cloud in Fig. 14 are generated using VincentSfM-Toolbox (Rabaud n.d.). Source codes of BALM, GROUSE, GRASTA, Damped Newton, Wiberg, LM_X used in the experiments are released by the corresponding author(s) of Del Bue et al (2012); Balzano et al (2010); He et al (2011); Buchanan and Fitzgibbon (2005); Okatani and Deguchi (2007); Wu et al (2011b) and Chen (2008). In particular, we are thankful that Balzano et al (2010) and Wu et al (2011b) shared with us a customized version of GROUSE and ALM-RPCA that are not yet released online. For Wiberg ℓ_1 (Eriksson and Van Den Hengel 2010), we have optimized the computation for Jacobian and adopted the commercial LP solver: cplex. The optimized code performs identically to the released code in small scale problems, but it is beyond the scope for us to verify for larger scale problems. In addition, we implemented SimonFunk's SVD ourselves. The ALS implementation is given in the released code package of LM_X. For OptManifold, TFOCS and CVX, we use the generic optimization packages released by the author(s) of Wen and Yin (2013); Becker et al (2011); Grant and Boyd (2008) and customize for the particular problem. For NLCG, we implement the derivations in Srebro and Jaakkola (2003) and used the generic NLCG package (Overton n.d.).

A.3 Additional experimental results

Illustration of the decomposition on Subject 3 of Extended YaleB dataset is given in Fig. 19. Additional qualitative comparisons on the recovery of the image is given in Fig. 20.

A.4 The lower bounds in the experiments

– The lower bound in Fig. 2: the lower bound is obtained by the data set that contains less than r data points per-column and per-row. It is clear from Kiràly and Tomioka (2012) that this is an easy-to-check necessary condition of recoverability.
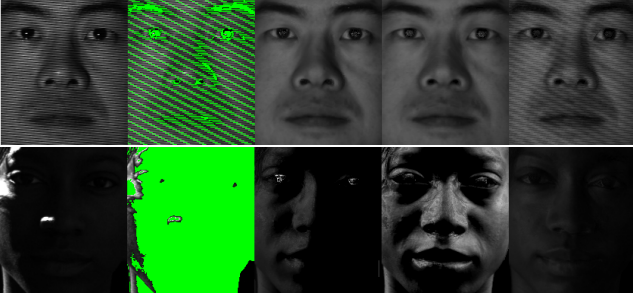
Fig. 20 Additional comparisons in the quality of face image recovery. From left to right, they are original image, missing data mask (in green), results for PARSuMi, BALM and missing-RPCA.

- *The oracle RMSE for Phase Diagram*: We also adapt the oracle lower bound from Candès and Plan (2010) to represent the theoretical limit of recovery accuracy under noise. Our extended oracle bound under both sparse corruptions and Gaussian noise is:

$$\text{RMSE}_{oracle} = \sigma \sqrt{\frac{(m + n - r)r}{p - e}}, \tag{61}$$

This is used for benchmarking in our phase diagram experiments.

- *The oracle angular error in Table 4:* For the Caesar and Elephant experiments, we use (61) (ignoring corruptions by taking $e = 0$) but transformed it by taking

$$\arcsin \sqrt{1 - (n \cdot \hat{n})^2},$$

where $\hat{n}$ is the surface normal obtained by an oracle projection of the noisily observed image.

A.5 Summary of parameters used in the experiments

- *Parameters in our formulation*: We assume r (the underlying rank) to be known. N_0 is chosen to be an upper bound of the number of corrupted entries. In experiments, we use 120% of the actual number of corruptions. In practice, we should choose $N_0 = 0.1|\Omega|$ or $0.15|\Omega|$. $\epsilon = 1e - 10$ (almost negligible). $K_E = 20\sqrt{N_0 \times \text{median}(\mathcal{P})_\Omega(\widehat{W})}$ (very large, negligible). In theory, we only need $\epsilon > 0$ and $K_E < \infty$ to ensure the convergence. In practice, unless it is meaningful to choose an effective K_E, we will choose it large enough so that it has no impact on the optimization.
- *Parameters for PARSuMi*: $\beta_1 = \beta_2 = \frac{1e-3}{\sqrt{\max m,n}}$. For Algorithm 1, $\rho = 10$ and initial $\lambda = 1e - 6$.
- *Parameters for APG*: $\gamma = \frac{1}{\sqrt{\max m,n}}$ $\lambda = 0.2$

References

Attouch H, Bolte J, Redont P, Soubeyran A (2010) Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. Mathematics of Operations Research 35(2):438–457

Balzano L, Nowak R, Recht B (2010) Online identification and tracking of subspaces from highly incomplete information. In: Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, IEEE, pp 704–711

Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences 2(1):183–202

Becker SR, Candès EJ, Grant MC (2011) Templates for convex cone problems with applications to sparse signal recovery. Mathematical Programming Computation 3(3):165–218

Becker SR, Candès EJ, Grant MC (2012) TFOCS: Templates for first-order conic solvers. http://cvxr.com/tfocs/

Bennett J, Lanning S, Netflix N (2007) The Netflix Prize. In: In KDD Cup and Workshop in conjunction with KDD

Buchanan AM, Fitzgibbon AW (2005) Damped Newton algorithms for matrix factorization with missing data. In: IJCV, vol 2, pp 316–322

Candès E, Plan Y (2010) Matrix completion with noise. Proc IEEE 98(6):925–936

Candès E, Recht B (2009) Exact matrix completion via convex optimization. Foundations of Computational Mathematics 9(6):717–772

Candès E, Li X, Ma Y, Wright J (2011) Robust principal component analysis? Journal of the ACM 58(3)

Candès EJ, Tao T (2010) The power of convex relaxation: Near-optimal matrix completion. Information Theory, IEEE Transactions on 56(5):2053–2080

Candès EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ℓ_1 minimization. Journal of Fourier Analysis and Applications 14(5-6):877–905

Chandrasekaran V, Sanghavi S, Parrilo P, Willsky A (2011) Rank-sparsity incoherence for matrix decomposition. SIAM Journal on Optimization 21:572–596

Chen P (2008) Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. IJCV 80(1):125–142

Chen P (2011) Hessian matrix vs. gauss-newton hessian matrix. SIAM Journal on Numerical Analysis 49(4):1417–1435

Chen Y, Jalali A, Sanghavi S, Caramanis C (2011) Low-rank matrix recovery from errors and erasures. In: Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on, IEEE, pp 2313–2317

Clarke FH (1990) Optimization and nonsmooth analysis, vol 5. Siam

Del Bue A, Xavier J, Agapito L, Paladini M (2012) Bilinear modeling via augmented lagrange multipliers (balm). Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(8):1496 –1508

Donoho DL (1995) De-noising by soft-thresholding. Information Theory, IEEE Transactions on 41(3):613–627

Eriksson A, Van Den Hengel A (2010) Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L1 norm. CVPR pp 771–778

Friedland S, Niknejad A, Kaveh M, Zare H (2006) An Algorithm for Missing Value Estimation for DNA Microarray Data. In: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, IEEE, vol 2, p II

Funk S (2006) Netflix update: Try this at home. http://sifter.org/~simon/journal/20061211.html

Gabriel KR, Zamir S (1979) Lower rank approximation of matrices by least squares with any choice of weights. Technometrics 21(4):489–498

Ganesh A, Wright J, Li X, Candès EJ, Ma Y (2010) Dense error correction for low-rank matrices via principal component pursuit. In: Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on, IEEE, pp 1513–1517

Grant M, Boyd S (2008) Graph implementations for nonsmooth convex programs. In: Blondel V, Boyd S, Kimura H (eds) Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, pp 95–110

Grant M, Boyd S (2012) CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx

Hartley R, Schaffalitzky F (2003) Powerfactorization: 3d reconstruction with missing or uncertain data. In: Australia-Japan advanced workshop on computer vision, vol 74, pp 76–85

He J, Balzano L, Lui J (2011) Online robust subspace tracking from partial information. arXiv preprint arXiv:11093827

Horn BK (1990) Height and gradient from shading. International journal of computer vision 5(1):37–75

Jain P, Netrapalli P, Sanghavi S (2012) Low-rank matrix completion using alternating minimization. arXiv preprint arXiv:12120467

Ke Q, Kanade T (2005) Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: CVPR, vol 1, pp 739–746

Keshavan R, Montanari A, Oh S (2009) Low-rank matrix completion with noisy observations: a quantitative comparison. In: Communication, Control, and Computing, pp 1216–1222

Kiràly F, Tomioka R (2012) A combinatorial algebraic approach for the identifiability of low-rank matrix completion. In: Langford J, Pineau J (eds) Proceedings of the 29th International Conference on Machine Learning (ICML-12), Omnipress, New York, NY, USA, ICML '12, pp 967–974

Koren Y (2009) The Bellkor solution to the Netflix grand prize. Netflix prize documentation

Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37

Lee KC, Ho J, Kriegman DJ (2005) Acquiring linear subspaces for face recognition under variable lighting. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27(5):684–698

Li X (2013) Compressed sensing and matrix completion with constant proportion of corruptions. Constructive Approximation 37(1):73–99

Liu Y, Sun D, Toh K (2009) An implementable proximal point algorithmic framework for nuclear norm minimization. Mathematical Programming 133(1-2):1–38

Negahban S, Wainwright MJ (2012) Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. The Journal of Machine Learning Research 13:1665–1697

Nehab D, Rusinkiewicz S, Davis J, Ramamoorthi R (2005) Efficiently combining positions and normals for precise 3d geometry. In: ACM Transactions on Graphics (TOG), ACM, vol 24, pp 536–543

Oh S, Montanari A, Karbasi A (2010) Sensor network localization from local connectivity: Performance analysis for the mds-map algorithm. In: Information Theory Workshop (ITW), 2010 IEEE, IEEE, pp 1–5

Okatani T, Deguchi K (2007) On the wiberg algorithm for matrix factorization in the presence of missing components. IJCV 72(3):329–337

Overton ML (n.d.) NLCG: Nonlinear conjugate gradient. http://www.cs.nyu.edu/faculty/overton/software/nlcg/index.html

Paladini M, Bue AD, Stosic M, Dodig M, Xavier J, Agapito L (2009) Factorization for non-rigid and articulated structure using metric projections. CVPR pp 2898–2905

Rabaud V (n.d.) Vincent's Structure from Motion Toolbox. http://vision.ucsd.edu/~vrabaud/toolbox/

Recht B (2009) A simpler approach to matrix completion. arXiv preprint arXiv:09100651

Recht B, Re C, Wright S, Niu F (2011) Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In: Advances in Neural Information Processing Systems 24, pp 693–701

Rudin W (1987) Real Analysis, vol 84. McGraw-Hill, New York

Shi X, Yu PS (2011) Limitations of matrix completion via trace norm minimization. ACM SIGKDD Explorations Newsletter 12(2):16–20

Srebro N, Jaakkola T (2003) Weighted low-rank approximations. In: Proceedings of the 20th International Conference on Machine Learning (ICML-2003), vol 20, p 720

Sturm P, Triggs B (1996) A factorization based algorithm for multi-image projective structure and motion. In: 4th European Conference on Computer Vision, Springer, pp 709–720

Toh K, Yun S (2010) An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific J Optim 6:615–640

Tomasi C, Kanade T (1992) Shape and motion from image streams under orthography: a factorization method. IJCV 9(2):137–154

Wang YX, Xu H (2012) Stability of matrix factorization for collaborative filtering. In: Langford J, Pineau J (eds) Proceedings of the 29th International Conference on Machine Learning (ICML-12), Omnipress, New York, NY, USA, ICML '12, pp 417–424

Wen Z, Yin W (2013) A feasible method for optimization with orthogonality constraints. Mathematical Programming pp 1–38

Wu B, Ding C, Sun D, Toh KC (2011a) On the Moreau-Yosida regularization of the vector k-norm related functions. Preprint

Wu L, Ganesh A, Shi B, Matsushita Y, Wang Y, Ma Y (2011b) Robust photometric stereo via low-rank matrix completion and recovery. In: ACCV, pp 703–717

Xiong L, Chen X, Schneider J (2011) Direct robust matrix factorizatoin for anomaly detection. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, pp 844–853

Yang J, Sun D, Toh KC (2012) A proximal point algorithm for log-determinant optimization with group lasso regularization. To appear in SIAM J Optimization

Yu HF, Hsieh CJ, Si S, Dhillon I (2012) Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In: Data Mining (ICDM), 2012 IEEE 12th International Conference on, IEEE, pp 765–774

Zhou Z, Li X, Wright J, Candès E, Ma Y (2010) Stable principal component pursuit. In: Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on, IEEE, pp 1518–1522
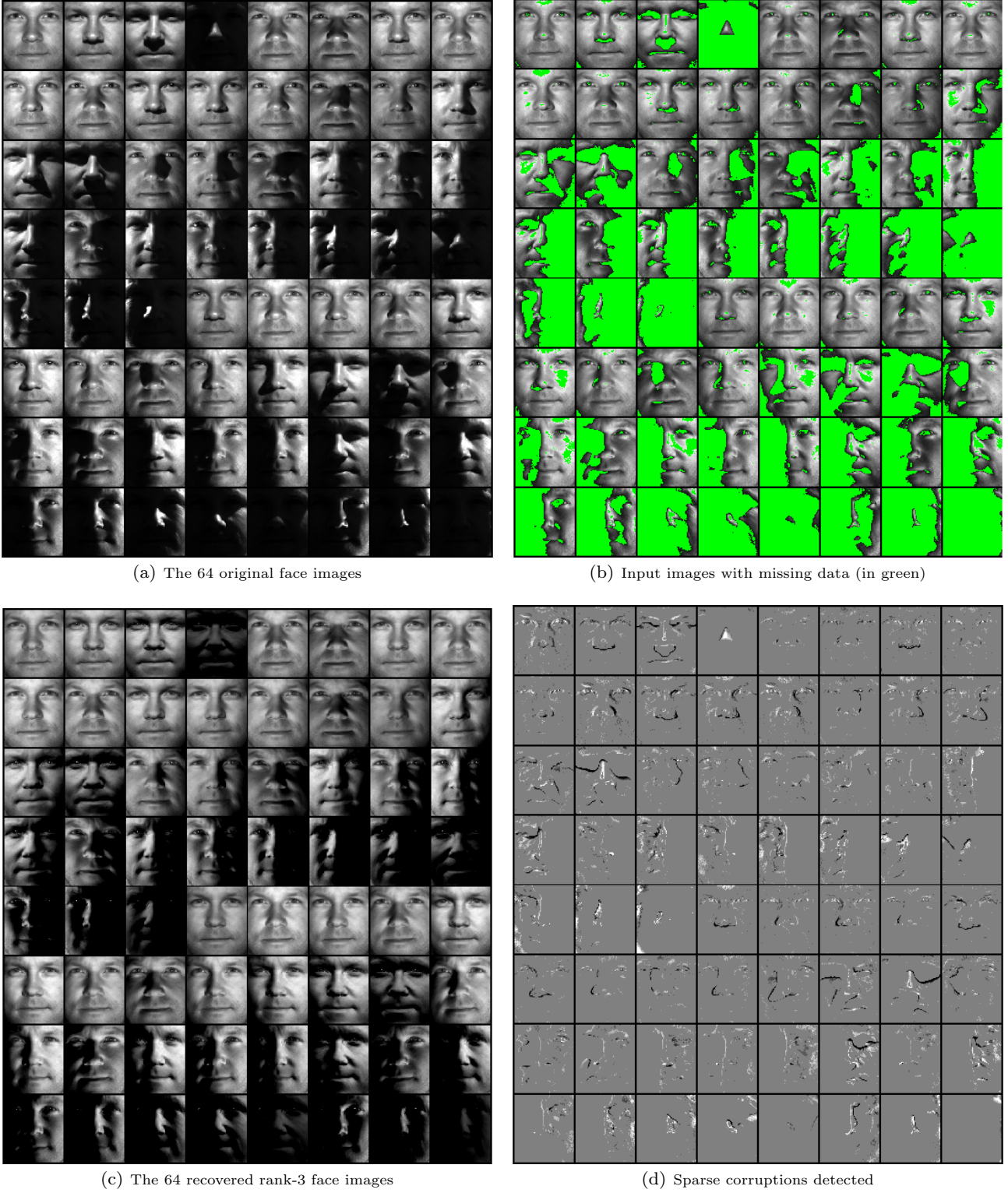
(a) The 64 original face images

(b) Input images with missing data (in green)

(c) The 64 recovered rank-3 face images

(d) Sparse corruptions detected

Fig. 19 Results of PARSuMi on Subject 3 of Extended YaleB. Note that the facial expressions are slightly different and some images have more than 90% of missing data. Also note that the sparse corruptions detected unified the irregular facial expressions and recovered those highlight and shadow that could not be labeled as missing data by plain thresholding.