Minimizing the Maximal Rank

Erik Bylow[1] Carl Olsson[1] Fredrik Kahl[1,2] Mikael Nilsson[1]
[1] Centre for Mathematical Sciences
Lund University, Sweden
[2] Department of Signals and Systems
Chalmers University of Technology, Sweden

{erikb,calle,micken}@maths.lth.se fredrik.kahl@chalmers.se

Abstract

In computer vision, many problems can be formulated as finding a low rank approximation of a given matrix. Ideally, if all elements of the measurement matrix are available, this is easily solved in the L_2-norm using factorization. However, in practice this is rarely the case. Lately, this problem has been addressed using different approaches, one is to replace the rank term by the convex nuclear norm, another is to derive the convex envelope of the rank term plus a data term. In the latter case, matrices are divided into sub-matrices and the envelope is computed for each subblock individually. In this paper a new convex envelope is derived which takes all sub-matrices into account simultaneously. This leads to a simpler formulation, using only one parameter to control the trade-of between rank and data fit, for applications where one seeks low rank approximations of multiple matrices with the same rank. We show in this paper how our general framework can be used for manifold denoising of several images at once, as well as just denoising one image. Experimental comparisons show that our method achieves results similar to state-of-the-art approaches while being applicable for other problems such as linear shape model estimation.

1. Introduction

Low rank approximation and PCA type procedures are important in many disciplines, for example, statistics, bioinformatics, compression and prediction. In computer vision it has been proven useful for applications such as nonrigid and articulated structure from motion [5, 23, 12], photometric stereo [3], optical flow [13] and linear shape models [8, 22]. The rank of the approximating matrix typically describes the complexity of the solution. Therefore one seeks to find a low rank factorization $UV^T \approx M$. If the measurement matrix M is complete and the rank of the approximating matrix is known, then the best approximation, in a least squares sense, can be computed in closed form using the singular value decomposition (SVD) [10].

Alternatively the problem can be formulated as minimization of the objective function

$$f(X) = \mu \operatorname{rank}(X) + \|X - M\|_F^2. \tag{1}$$

Here μ is a parameter that controls the trade-off between data fit and rank. While the solution is easy to compute using SVD the optimization problem itself is non-convex and non-differentiable. As a consequence it is difficult to modify the formulation without having to resort to heuristic optimization approaches. For example, in case there are missing entries and/or outliers the optimization problem is substantially more difficult. In structure from motion, recent approaches [11, 20] attack these problems by optimizing jointly over fixed size U and V matrices. As a consequence the rank has to be predetermined and the quality of the result is dependent on initialization.

To achieve flexible formulations that are independent of initialization, researchers have instead started to consider convex surrogates of the rank function. Most commonly the convex nuclear norm, or sum-of-singular-values penalty is used [18, 6, 17, 2, 7]. One reason for its popularity is that it can be shown that if the locations of the missing entries are random the approach gives the best low rank approximation [6]. In many computer vision applications missing entry locations are highly correlated which makes the approach break down.

An additional downside of using the nuclear norm is that it has a bias to small solutions. Due to its definition it penalizes both small and large singular values equally. Indeed its proximal operator corresponds to soft thresholding [6]. In contrast, the desirable operation of hard thresholding, which is performed when solving (1) with SVD, leaves the larger singular values *unchanged*.

A convex formulation that only penalizes the small singular values was recently proposed in [16]. It is shown that the convex envelope of (1) is given by

$$f^{**}(X) = \sum_{i=1}^{n} \left(\mu - [\sqrt{\mu} - \sigma_i(X)]_+^2\right) + \|X - M\|_F^2, \tag{2}$$

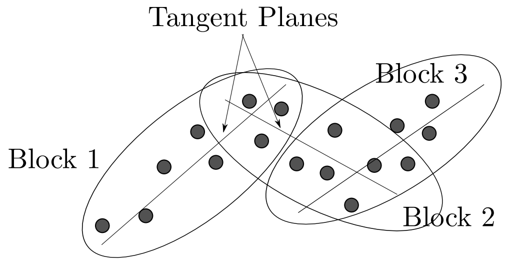

Figure 1: A simple illustration of how to estimate tangent planes of the manifold.

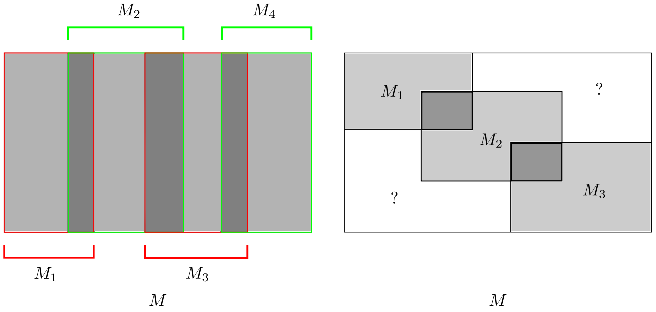

Figure 2: Illustration of how measurement matrices can divided into blocks. *Left*: Blocks for tangent spaces. *Right*: Example of block division with missing data.

where $[\cdot]_+$ denotes truncation at 0 and $\sigma_i(X)$, $i = 1, \dots, n$ are singular values of X. Since f^{**} is the convex envelope of f their minimum values coincide and $f^{**}(X)$ is a lower bound $f(X)$ for every X. Furthermore, singular values that are larger than $\sqrt{\mu}$ get a constant penalty, which is similar to hard thresholding.

In this paper we are interested in problems where multiple matrices of the same unknown rank need to be estimated. One example where this appears is in manifold estimation. All the tangent spaces of a connected manifold have the same dimension, equal to the dimension of the manifold. Locally a d-dimensional manifold can be thought of as a d-dimensional tangent space. Therefore approximating the data with a d-dimensional manifold can be thought of as locally approximating data with low rank matrices (all of rank d). Another problem that can be cast in the same framework is the missing data problem. In [16] it was solved by applying the objective (1) on complete sub-blocks of the measurement matrix. To achieve the same rank on all subblocks, one μ-parameter for each block had to be selected. Optimal parameter selection is a major obstacle for this approach.

More specifically, in this paper we propose an approach where a trade-off between the maximal rank of a set of matrices and their fit to observed data is penalized. In contrast to the approach in [16] we consider all matrices at the same time. The formulation, which has only one parameter, ensures that the estimated matrices are of the same (unknown) rank. Our main technical contribution is that we derive an expression for the convex envelope and show that its proximal operator is equivalent to a convex cone problem. This allows efficient optimization using an ADMM approach [4]. We present several applications where this framework can be applied, including manifold denoising and missing data problems.

2. Regularization With the Maximal Rank

In applications like manifold estimation, one seeks to estimate a manifold by its tangent spaces, where the tangent spaces have a lower dimension than the ambient space, see Figure 1. In particular, all tangent spaces have the same dimension which is also equal to the dimension of the manifold. To achieve this, one can divide the data points in a measurement matrix M into different neighborhoods. These neighborhoods form blocks (or sub-matrices) M_j of M, see left of Figure 2. Note that in this case, the blocks M_j have the same number of rows as the matrix M, but they may vary in sizes and typically have no missing data. We will show how one can compute a low-rank approximation X where all blocks X_j corresponding to M_j have the same rank. These low-rank approximations correspond to the low dimensional (affine) tangent spaces. Further details of the specific formulation for this application will be given in Section 3.1. In this section, we will work with a more general formulation.

We let $\hat{M} = (M_1, M_2, ..., M_b)$ be a collection of measurement matrices that we wish to approximate with $\hat{X} = (X_1, X_2, ..., X_b)$, where X_j, $j = 1, \dots, b$, are of the same (unknown) rank. Note that the matrices M_j in $\hat{M}$ need not to have the same sizes, and thus $\hat{M}$ should be regarded as a collection of measurement matrices. Our objective function will be of the form

$$\min_{\hat{X}} \mu r(\hat{X}) + \|\hat{X} - \hat{M}\|^2, \tag{3}$$

where the regularization term is

$$r(\hat{X}) = \max(\,\mathrm{rank}(X_1),\ \mathrm{rank}(X_2), ...,\ \mathrm{rank}(X_b)), \tag{4}$$

the data fit is measured by

$$\|\hat{X} - \hat{M}\|^2 = \sum_{j=1}^{b} \|X_j - M_j\|_F^2, \tag{5}$$

and $\|\cdot\|_F$ is the regular Frobenious norm. The parameter μ controls the trade-off between rank and data fit. In practice we are interested in solutions where the ranks of the X_j matrices are the same. It can be seen that the regularizer (4) will achieve this under the assumption that the

M_j matrices are all of full rank. If for some j we have $r(\hat{X}) >$ rank(X_j) then the data term $\|X_j - M_j\|_F^2$ can be reduced by adding another singular value to X_j without affecting any other term.

A common approach would be to simply replace the rank functions in (4) with nuclear norms. However in contrast to the rank function the nuclear norm is not scale invariant. Therefore this will result in a regularizer that penalizes the matrices unevenly. In particular if the matrices X_j have varying sizes. Furthermore, the nuclear norm is only a lower bound on the rank function on the set $\{X; \sigma_1(X) \leq 1\}$, while in contrast our convex envelope will be valid on an unbounded domain. Recall that the convex envelope is by definition the tightest possible lower-bounding convex function, hence the ideal tool for our purposes.

In the following sections we will compute the convex envelope of our formulation via conjugate functions and derive its proximal operator [19].

2.1. Conjugate Functions

To find the convex envelope of (3) we consider the conjugate function, which is by definition

$$f^*(\hat{Y}) = \max_{\hat{X}} \langle \hat{X}, \hat{Y} \rangle - \mu r(\hat{X}) - \|\hat{X} - \hat{M}\|^2, \quad (6)$$

where $\langle \hat{X}, \hat{Y} \rangle = \sum_{j=1}^{b} \mathrm{tr}(X_j^T Y_j)$. By completing squares via

$$||\hat{X} - (\hat{M} + \frac{\hat{Y}}{2})||^2 = ||\hat{X}||^2 - 2\langle \hat{X}, \hat{M} + \frac{\hat{Y}}{2} \rangle + ||\hat{M} + \frac{\hat{Y}}{2}||^2, \quad (7)$$

the maximization in (6) can be written

$$\max_{k} \max_{r(\hat{X})=k} -\|\hat{X} - \hat{Z}\|^2 + \|\hat{Z}\|^2 - \|\hat{M}\|^2 - \mu k, \quad (8)$$

where $\hat{Z} = \hat{M} + \frac{\hat{Y}}{2}$. For a fixed k the problem is separable in the matrices X_j, $j = 1, ..., b$. That is, the optimal X_j can be obtained from the SVD of Z_j giving

$$X_j = \sum_{i=1}^{k} \sigma_i(Z_j) u_i v_i^T. \quad (9)$$

Inserting into (8) we get

$$\max_{k} - \sum_{i=k+1}^{n} \|\sigma_i(\hat{Z})\|_2^2 + \|\hat{Z}\|^2 - \|\hat{M}\|^2 - \mu k. \quad (10)$$

Here $\sigma_i(\hat{Z})$ is the vector $(\sigma_i(Z_1), \sigma_i(Z_2), \ldots, \sigma_i(Z_b))$ and $\|\cdot\|_2$ is the regular euclidean vector norm. To select the maximizing k we note that

$$\mu k + \sum_{i=k+1}^{n} \|\sigma_i(\hat{Z})\|_2^2 = \sum_{i=1}^{k} \mu + \sum_{i=k+1}^{n} \|\sigma_i(\hat{Z})\|_2^2. \quad (11)$$

Since each entry in the vector $\sigma_i(\hat{Z})$ is positive and decreasing in i, its norm $\|\sigma_i(\hat{Z})\|_2^2$ will also be decreasing with i. Therefore k should be selected such that

$$\|\sigma_{k+1}(\hat{Z})\|_2^2 \leq \mu \leq \|\sigma_k(\hat{Z})\|_2^2. \quad (12)$$

This gives the conjugate function

$$f^*(\hat{Y}) = -\sum_{i=1}^{n} \min(\mu, \|\sigma_i(\hat{Z})\|_2^2) + \|\hat{Z}\|^2 - \|\hat{M}\|^2. \quad (13)$$

Recall that $\hat{Z}$ depends on $\hat{Y}$ through $\hat{Z} = \hat{M} + \frac{\hat{Y}}{2}$. Next we consider the biconjugate, which is by definition

$$\begin{aligned} f^{**}(\hat{X}) &= \max_{\hat{Y}} \langle \hat{X}, \hat{Y} \rangle - f^*(\hat{Y}) \quad (14) \\ &= \max_{\hat{Z}} 2\langle \hat{X}, \hat{Z} - \hat{M} \rangle - f^*(2\hat{Z} - 2\hat{M}). \quad (15) \end{aligned}$$

The objective function in (15) can be written

$$\sum_{i=1}^{n} \min(\mu, \|\sigma_i(\hat{Z})\|_2^2) - \|\hat{Z} - \hat{X}\|^2 + \|\hat{X} - \hat{M}\|^2. \quad (16)$$

Using von Neumann's trace theorem it can be seen that the optimal Z_j has to have an SVD with the same U and V as X_j. Therefore the optimization can be reduced to a search over the singular values of the Z_j, $j = 1, ..., b$, giving the convex envelope

$$f^{**}(\hat{X}) = \mathcal{R}_\mu(\hat{X}) + \|\hat{X} - \hat{M}\|^2, \quad (17)$$

where

$$\mathcal{R}_\mu(\hat{X}) = \max_{\hat{Z}} \sum_{i=1}^{n} \min(\mu, \|\sigma_i(\hat{Z})\|_2^2) - \|\sigma_i(\hat{Z}) - \sigma_i(\hat{X})\|_2^2. \quad (18)$$

2.2. The Proximal Operator of f^{**}

The maximization over the singular values in (18) does not seem to have any closed form solution. Evaluation of $f^{**}(\hat{X})$ therefore has to be done by numerically maximizing the (concave) objective function. At first glance it may therefore seem as though minimization of f^{**} would involve a search over numerical evaluations of f^{**}. Fortunately this can be avoided. In this section we show that the proximal operator

$$\mathrm{prox}_{f^{**}}(\hat{Y}) = \arg\min_{\hat{X}} f^{**}(\hat{X}) + \rho\|\hat{X} - \hat{Y}\|^2, \quad (19)$$

which is the basis for ADMM can be computed using a single cone program. The trick is to switch the order of minimization and maximization and thereby obtain a closed form solution for $\hat{X}$. If $\rho > 0$ the objective function is closed, proper convex-concave, continuous and the optimization can be restricted to a compact set. Switching optimization order is therefore justified by the existence of a saddle point, see [19]. To find the optimal $\hat{X}$ we consider the terms of (19) that contain $\hat{X}$

$$-\|\hat{Z}-\hat{X}\|^2 + \|\hat{X}-\hat{M}\|^2 + \rho\|\hat{X}-\hat{Y}\|^2. \tag{20}$$

It can be seen (e.g., by taking derivatives of (20)) that the optimal $\hat{X}$ is given by

$$\hat{X} = \hat{Y} + \frac{\hat{M}-\hat{Z}}{\rho}. \tag{21}$$

Inserting into (20) and completing squares gives

$$-\frac{\rho+1}{\rho}\left\|\hat{Z}-\hat{W}\right\|^2 + C, \tag{22}$$

where

$$\hat{W} = \frac{\rho\hat{Y}+\hat{M}}{\rho+1} \tag{23}$$

and

$$C = \frac{2\rho+1}{\rho}\|\hat{M}\|^2 + \rho\|\hat{Y}\|^2 - \rho\|\hat{Y} + \frac{\hat{M}}{\rho}\|^2. \tag{24}$$

Note that C is independent of $\hat{Z}$. In practice we are only interested in finding the optimizers $\hat{Z}$ and $\hat{X}$ and not the objective value itself. Hence we can ignore C. We therefore need to maximize

$$\sum_{i=1}^{n}\min(\mu, \|\sigma_i(\hat{Z})\|_2^2) - \frac{\rho+1}{\rho}\left\|\hat{Z}-\hat{W}\right\|^2. \tag{25}$$

The terms in the sum only depend on the singular values of the matrices Z_j, $j = 1, \ldots, b$. For the second term we have

$$\left\|\hat{Z}-\hat{W}\right\|^2 = \|\hat{Z}\|^2 - 2\sum_{j=1}^{b}\langle Z_j, W_j\rangle + \|\hat{W}\|^2. \tag{26}$$

By von Neumann's trace theorem $\langle Z_j, W_j\rangle \le \sum_{i=1}^{n}\sigma_i(Z_j)\sigma_i(W_j)$ one sees that the SVD of Z_j has the same U and V as the SVD of W_j. Therefore (25) simplifies to

$$\sum_{i=1}^{n}\left(\min(\mu, \|\sigma_i(\hat{Z})\|_2^2) - \frac{\rho+1}{\rho}\left\|\sigma_i(\hat{Z})-\sigma_i(\hat{W})\right\|_2^2\right). \tag{27}$$

The singular values can now be determined using a cone program. To see this we introduce the auxiliary variables s_i, $i = 1, ..., n$ and write

$$\max \sum_{i=1}^{n} s_i \tag{28}$$

$$\text{s.t. } s_i \le \mu - \frac{\rho+1}{\rho}\left\|\sigma_i(\hat{Z})-\sigma_i(\hat{W})\right\|_2^2 \tag{29}$$

$$s_i \le \left\|\sigma_i(\hat{Z})\right\|_2^2 - \frac{\rho+1}{\rho}\left\|\sigma_i(\hat{Z})-\sigma_i(\hat{W})\right\|_2^2. \tag{30}$$

Note that as we are maximizing the sum of s_i, (29) or (30) will always attain equality at the optimal solution. Thus the above program is equivalent to (27). For the singular values to be feasible they have to be decreasing for each block. To enforce this we add linear constraints on the entries of the vectors $\sigma_i(\hat{Z})$ which results in the formulation

$$\max \sum_{i=1}^{n} s_i \tag{31}$$

$$\text{s.t. } \left\|\sigma_i(\hat{Z})-\sigma_i(\hat{W})\right\|_2^2 \le \frac{\rho}{\rho+1}(\mu - s_i) \tag{32}$$

$$\frac{\rho+1}{\rho}\left\|\sigma_i(\hat{Z})-\sigma_i(\hat{W})\right\|_2^2 - \left\|\sigma_i(\hat{Z})\right\|_2^2 \le -s_i \tag{33}$$

$$\sigma_1(\hat{Z}) \ge \sigma_2(\hat{Z}) \ge ... \ge \sigma_n(\hat{Z}) \ge 0. \tag{34}$$

Equation (32) is easily seen to be convex since the left side is a positive definite quadratic form and the right hand side is linear. To see that the same holds true for (33) we can rewrite this constraint as

$$\left\|\sigma_i(\hat{Z}) - (\rho+1)\sigma_i(\hat{W})\right\|_2^2 \le \rho\left(\left\|(\rho+1)\sigma_i(\hat{W})\right\|_2^2 - s_i\right). \tag{35}$$

Constraints (32) and (35) can be realized using the cone

$$\{(x_1, x_2, x_3);\ x_1x_2 \ge \|x_3\|_2^2,\ x_1 + x_2 \ge 0\}. \tag{36}$$

This type of cone (which is a rotation of the quadratic cone) is supported in SeDuMi [21] and Mosek [1] which we use to solve (19).

3. Applications

In this section, we present two applications of our framework: (i) Manifold denoising and (ii) Linear shape basis models.

3.1. Manifold Denoising

Manifold denoising can be formulated as seeking affine tangent spaces with the same dimension. If we have a set of images, possibly corrupted with noise, then the assumption is that the true uncorrupted images lie on a low-dimensional manifold. The images, represented by m_i, $i = 1, \ldots, N$, are assumed to be column-stacked so one image lies in $\mathbb{R}^n$, where n is the number of pixels. The assumption means that several points which are close to each other should be close to the tangent space of the manifold as illustrated in Figure 1. To determine neighbourhoods, we find for each image point its K-closest neighbors in the euclidean distance and consider them to be one block.

Given a set of images, stacked in a measurement matrix $M = [m_1, m_2, ..., m_N]$, we determine a collection of blocks via the neighbourhoods, $\hat{M} = (M_1, M_2, \ldots, M_b)$, see left of Figure 2. Since the images are corrupted by noise,

	Input PSNR	Output PSNR
Our method	10.4553	**17.5231**
Manifold Denoising	10.4553	15.6656

Table 1: The PSNR using different methods for denoising on the USPS Digits.

each sub-matrix M_i will have high rank, and the task is to find a low-rank approximation X_i for each M_i. Note that the different X_i share common varibles. Also, since we are interested in the affine tangent spaces, which do not necessarily go through the origin, we add the row-vise mean vector $\bar{x}_i$ of each X_i. Assuming that X_i have zero row means, that is $X_i\mathbb{1} = 0$, the fitting terms in the objective function can be written

$$\sum_{i=1}^{b} \|X_i + \bar{x}_i\mathbb{1}^T - M_i\|^2 = \tag{37}$$

$$\sum_{i=1}^{b} \|X_i - (M_i - \bar{m}_i\mathbb{1}^T)\|_F^2 + k_i\|\bar{x}_i - \bar{m}_i\|_2^2, \tag{38}$$

where $\bar{m}_i$ is the row mean of M_i and k_i is equal to the number of columns in block M_i. To ensure consistency between shared variables, we penalize the differences by adding to the objective

$$\alpha \sum_{i=1}^{b} \|\mathcal{P}_i(X) - (X_i + \bar{x}_i\mathbb{1}^T)\|_F^2,$$

where α is a weighting factor, X is the approximation of the measurement matrix M and $\mathcal{P}_i(X)$ retrieves block i in X. In summary, we have the following optimization problem

$$\min_{\hat{X}, X, \bar{x}_i} \; r(\hat{X}) + \sum_{i=1}^{b} \Big(\|X_i - (M_i - \bar{m}_i\mathbb{1}^T)\|_F^2 + \tag{39}$$

$$k_i\|\bar{x}_i - \bar{m}_i\|_2^2 + \alpha\|\mathcal{P}_i(X) - (X_i + \bar{x}_i\mathbb{1}^T)\|_F^2 \Big) \tag{40}$$

$$\textit{s.t.}\; X_i\mathbb{1} = 0 \quad i = 1, \ldots, b. \tag{41}$$

We have already derived the convex envelope for the terms on the first row (39) and the terms on the second row (40) are convex from the start. To minimize (39) we use the convex envelope and introduce auxiallary variables Z_i, which results in

$$\min_{\hat{X}, X, \bar{x}_i, \hat{Z}} \; f^{**}(\hat{X}) +$$

$$\sum_{i=1}^{b} \Big(k_i\|\bar{x}_i - \bar{m}_i\|_2^2 + \alpha\|\mathcal{P}_i(X) - Z_i - \bar{x}_i\mathbb{1}^T\|_F^2 \Big) \tag{42}$$

$$\textit{s.t.}\; X_i = Z_i, \quad Z_i\mathbb{1} = 0, \quad i = 1, \ldots, b, \tag{43}$$

and in turn, this leads to the ADMM formulation

$$\min_{\hat{X}, X, \bar{x}_i, \hat{Z}} \; f^{**}(\hat{X}) + \rho\|\hat{X} - \hat{Z} + \hat{\Lambda}\|^2 - \rho\|\hat{\Lambda}\|^2 +$$

$$\sum_{i=1}^{b} \Big(k_i\|\bar{x}_i - \bar{m}_i\|_2^2 + \alpha\|\mathcal{P}_i(X) - Z_i - \bar{x}_i\mathbb{1}^T\|_F^2 \Big) \tag{44}$$

$$\textit{s.t.}\; Z_i\mathbb{1} = 0, \quad i = 1, \ldots, b. \tag{45}$$

We get one part which depends on $\hat{X}$,

$$\min_{\hat{X}} f^{**}(\hat{X}) + \rho\|\hat{X} - \hat{Z} + \hat{\Lambda}\|_F, \tag{46}$$

which is precisely the proximal operator we have seen before. To minimize with respect to the other variables $\hat{Z}$, $\bar{x}_i$ and X is now straightforward. Keeping the other variables fixed and solving for one we get the following updates:

$$X^{t+1} = \arg\min_{X^t} \alpha \sum_{i=1}^{b} \|\mathcal{P}_i(X^t) - Z_i^t - \bar{x}_i^t\mathbb{1}^T\|_F^2 \tag{47}$$

which is a separable least squares problem,

$$Z_i^{t+1} = \arg\min_{Z_i^t} \alpha\|\mathcal{P}_i(X^{t+1}) - Z_i^t - \bar{x}_i^t\mathbb{1}^T\|_F^2 +$$

$$\rho\|X_i^t - Z_i^t + \Lambda_i^t\|_F^2 \quad i = 1, \ldots, b, \tag{48}$$

since all blocks are independent in the minimization,

$$\bar{x}_i^{t+1} = \arg\min_{\bar{x}_i^t} k_i\|\bar{x}_i^t - \bar{m}_i\|_2^2 +$$

$$\alpha\|\mathcal{P}_i(X^{t+1}) - Z_i^{t+1} - \bar{x}_i^t\mathbb{1}^T\|_F^2, \tag{49}$$

since we can minimize each $\bar{x}_i$ separately. To find $\hat{X}^{t+1}$ we use the proximal operator we have deduced:

$$\hat{X}^{t+1} = \text{prox}_{f^{**}}(\hat{Z}^{t+1} - \hat{\Lambda}^{t+1}). \tag{50}$$

For $\hat{\Lambda}$ we take a step in the ascent direction, that is,

$$\Lambda_i^{t+1} = \Lambda_i^t + X_i^{t+1} - Z_i^{t+1}, \tag{51}$$

since again, each block is separable.

Experimental results: USPS. To test the denoising method, we use the USPS dataset [15] of handwritten digits. We choose 100 images of each digit and rescale the intensities to lie between $[0, 1]$. The images are perturbed by Gaussian noise with standard deviation $\sigma = 0.3$. We then stack *all* images into one measurement matrix M, and find the $K = 30$ closest neighbours for each image. With this data, we apply our optimization to obtain the new approximate

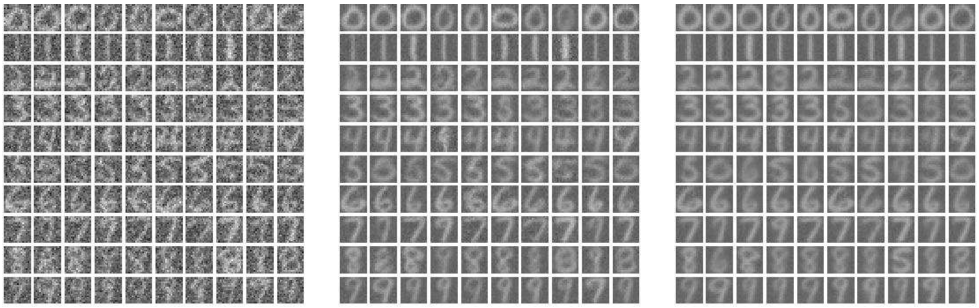

Figure 3: Some results from denoising the USPS digits. From left to right: Input images, our results and the results from Manifold Denoising.

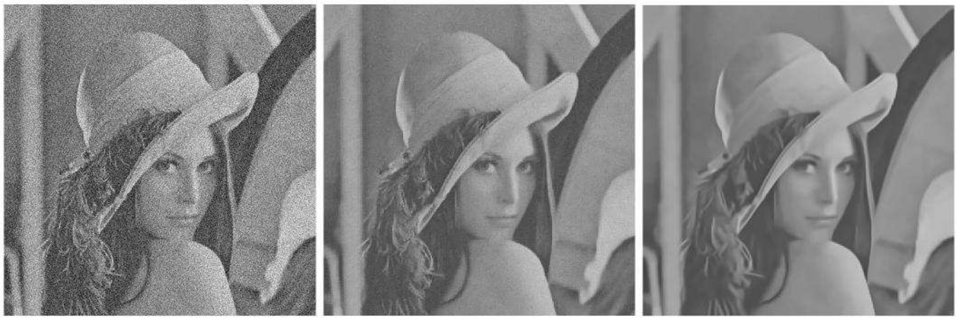

Figure 4: Denoising results of the Lena image. *Left:* Noisy input image. *Middle:* Denoised image using our method. *Right:* Denoising results from BM3D.

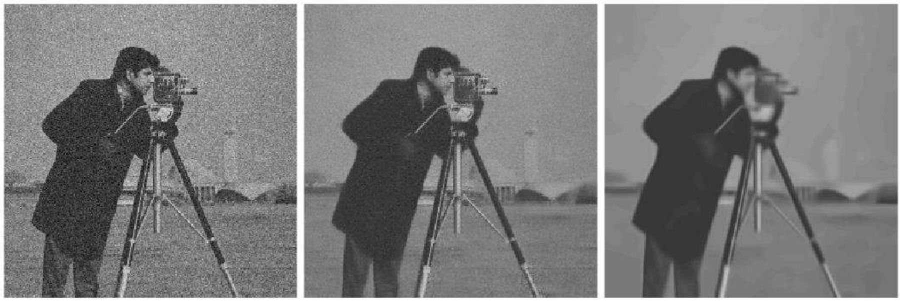

Figure 5: *Left:* Input noisy Cameraman. *Middle:* Our result. Note the preserved details on the camera. *Right:* Result from BM3D.

matrix X of M. Each column in X contains a denoised image corresponding to the noisy image in the same column in M.

For comparison we use the well-known work called Manifold Denoising by [14]. This work uses a different approach where a partial differential equation is solved on a graph created by the data points to obtain a manifold.

The results shown in Table 1 where obtained when adding noise with standard deviation $\sigma = 0.3$ to the USPS dataset. For Manifold Denoising we set the number of neighbors to 6 and re-weighting parameter $\lambda = 1$ and a symmetric graph, since that gave the best results in our experiment.

Experimental results: Single image denoising. Our method for manifold denoising can also be used to denoise a single image. To apply our method, the image is first divided into several patches, and each patch is considered to be one point in $\mathbb{R}^n$. As above all points, or patches, are then stacked into one measurement matrix M. Thereafter,

	Input	Our method	BM3D
Lena	19.9914	28.6064	**29.1560**
Cameraman	19.9883	**26.0663**	24.7322

Table 2: Denoising results from the Cameraman and Lena. BM3D gives a higher PSNR for Lena, but we do better on the Cameraman.

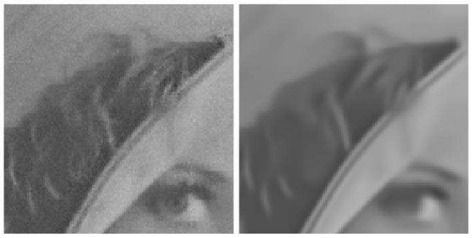

Figure 6: Zooming in, one can see more details in our result (*left*) compared to BM3D's result (*right*). Note that one can see the pupil in the eye of the left image, but not in the right image.

the optimal X is found applying our optimization method and each column in X equals a denoised patch which can be used to rebuild the image.

This was tested on Lena, size 512×512 pixels and the Cameraman, size 256×256 pixels. On both images we added gaussian noise with a standard deviation $\sigma = 0.1$. As can be seen in Figure 4, much of the noise is reduced. To compare our method, we also provide results from state-of-the-art method BM3D [9]. In the closeup of Figure 6, it can be seen that our method keeps details better than BM3D, for example, the pupil is clearly distinguishable in our result but not in BM3D's result. The PSNR on the input data was 19.9914. Our method improved to 28.6064 and BM3D to 29.1560.

To get these results a patch size of 12×12 pixels was used together with an overlap of $2/3$ between two consecutive patches, this results 15876 tangent spaces. The parameter α was set to 1.5 and μ to $75,000$ and the number of neighbors $K = 20$. The optimal blocks X_i had rank 2.

The same approach was also tested on the Cameraman and as can be seen in Figure 5, our method performs well compared to BM3D. Figure 5 shows that our method preserves more details compared to BM3D which smooths out some details. For example the camera is more detailed and the roof on the tower to the right is more preserved. For the Cameraman we used the same parameters as above except that $\mu = 22000$ and the number of tangent spaces was 3844. The optimal blocks X_i had rank 3. The denoising results are summarized in Table 2.

Dataset	Loc. Rank Func. [16]	Our method
Hand	0.474	0.474
Banner	$6.54 \cdot 10^7$	$4.73 \cdot 10^7$
Book	0.121	0.121

Table 3: The error $\sum_{i=1}^{b} \|X_i - M_i\|_F^2$ for the method in [16] and our method. Note that the method in [16] outperforms the nuclear norm relaxation for the same error metric.

3.2. Linear Shape Basis Models

Another application we test our framework on is estimation of linear shape models. A common assumption is that a set of tracked image points moving non-rigidly can be described with a small number of basis elements in each frame. If we let $M_f = (m_f^1, m_f^2, \ldots, m_f^N)$ denote the N tracked 2D-points in frame f, we want to find shape basis models $(S_1, S_2, \ldots, S_K)$ — each of size $2 \times N$ — and scalar coefficients $C_{f1}, C_{f2}, \ldots, C_{fK}$ such that the points M_f can be described by

$$M_f = \sum_{k=1}^{K} C_{fk} S_k. \tag{52}$$

Stacking the N points in F frames yields a $2F \times N$ measurement matrix M. Since we want to use as few basis elements as possible, the matrix M should be of low rank. Due to occlusion and tracking failures, not all points will be seen in all frames. This gives a measurement matrix M with missing data. To handle this we create sub-blocks M_i of M, where each M_i has no missing entries, see right of Figure 2. Hence, we have turned the problem into finding low-rank approximations X_i of M_i, where the blocks in X_i share common variables. The objective function we seek to minimize is

$$\begin{aligned} \min\nolimits_{\hat{X},X} \quad & f^{**}(\hat{X}) \\ \textit{s.t.} \quad & P_i(X) = X_i \quad i = 1, \ldots, b, \end{aligned} \tag{53}$$

where $\hat{X}$ is the collection of blocks $(X_1, X_2, \ldots, X_b)$ and $P_i(X)$ retrieves block i from X. The constraint comes from the requirement that the blocks we optimize over shall coincide on the overlap. To minimize this, we use ADMM and our augmented Lagrangian becomes

$$f^{**}(\hat{X}) + \rho\|\hat{X} - \hat{\mathcal{P}}(X) + \hat{\Lambda}\|^2 - \rho\|\hat{\Lambda}\|^2, \tag{54}$$

where $\hat{\mathcal{P}}(X) = (\mathcal{P}_1(X), ..., \mathcal{P}_b(X))$.

In each iteration we perform the following updates:

$$\hat{X}^{t+1} = \arg\min_{\hat{X}^t} f^{**}(\hat{X}^t) + \rho\|\hat{X}^t - \hat{\mathcal{P}}(X^t) + \hat{\Lambda}^t\|^2 \tag{55}$$

$$X^{t+1} = \arg\min_{X^t} \rho\|\hat{X}^{t+1} - \hat{\mathcal{P}}(X^t) + \hat{\Lambda}^t\|^2 \tag{56}$$

$$\Lambda_i^{t+1} = \Lambda_i^t + X_i^{t+1} - \mathcal{P}_i(X^{t+1}). \tag{57}$$

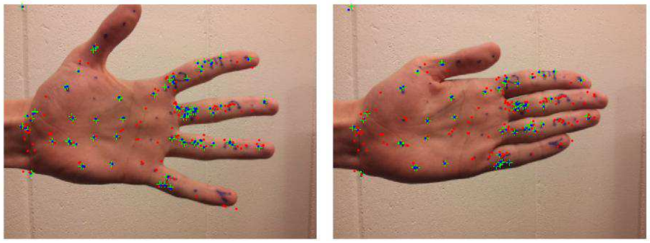

Figure 7: Frames 280 and 371 from the hand experiment. The solution has rank 5.

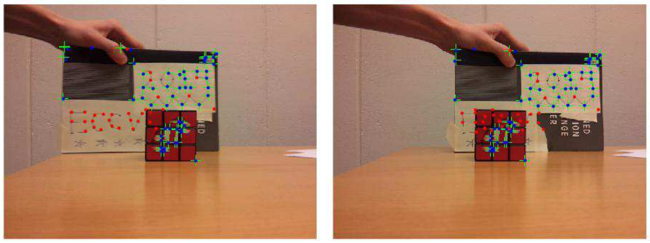

Figure 8: Frames 297 and 337 in the Book sequence. The solution has rank 3.

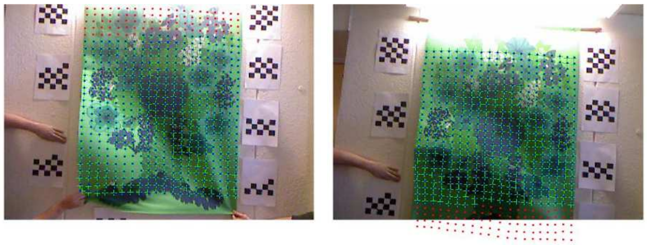

Figure 9: Frames 160 and 250 of the Banner sequence. The solution has rank 9.

When the low-rank approximation is found where only known data has been used, we complete the missing parts in X by applying the same method as in [16].

Experimental results. Our framework has been applied to a number of image sequences obtained from the authors of [16]. The results of the Hand-, Book- and Banner sequences are shown in Figures 7, 8 and 9.

One sees clearly in all sequences that the red and blue points, which are the reconstructed points, obey a motion which is reasonable compared to the input data, green points. The blue points are the reconstructed points which we could track and the red points are the reconstructed positions of points with no measurements available. The found rank for the solution in the Hand sequence is 5, in the Book sequence we get rank 3 and in the Banner sequence we get rank 9. The number of blocks in Hand, Book and Banner was 5,3 and 19.

To compare with [16], we test their method on the same datasets and measure the error $\sum_{i=1}^{b} \|M_i - X_i\|_F^2$ and the results are shown in Table 3. We choose to measure the error on the blocks since that will show if our method differs from [16]. Note that this method was shown to perform better than the nuclear norm relaxation for this application.

As the results in Table 3 show, we do at least as good as they do on these datasets. We investigated the approximated sub-matrices from [16] individually and saw that some sub-matrices had rank 8 and some rank 10. This shows that it is easier to get uniform rank on all sub-matrices using our formulation.

4. Conclusion

In this paper we have derived a novel and general convex framework to approximate low-rank matrices. Our method is suitable in situations where several matrices of the same rank need to be approximated. Our main contribution is the derivation of a strong convex formulation that can be optimized in general frameworks using the proximal operator in an ADMM fashion [4]. One of the advantages of our formulation is that there is a single tuning parameter controlling the trade-of between rank and model fit which is important in manifold estimation where the number of sub-matrices may be well above $10,000$.

Experimental evaluations showed that our method achieves results similar to state-of-the-art approaches on manifold denoising problems and linear shape basis estimation. It should be mentioned that in the case of manifold denoising neighborhoods are computed using noisy patches. Therefore results could potentially be improved by reestimating neighborhoods using cleaned versions. However, to focus the evaluation on our convex formation we have refrained from such heuristics.

5. Acknowledgements

This work has been funded by the Swedish Research Council (grant no. 2012-4213), the Crafoord Foundation and Scientific Research Council, project no. 2012-4215.

References

[1] *The MOSEK optimization toolbox for MATLAB manual.* URL www.mosek.com. 4

[2] R. Angst, C. Zach, and M. Pollefeys. The generalized trace-norm and its application to structure-from-motion problems. In *International Conference on Computer Vision*, 2011. 1

[3] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *Int. J. Comput. Vision*, 72(3):239–257, 2007. 1

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011. 2, 8

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 1

[6] J. Cai, E.J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, 2010. 1

[7] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. 1

[8] T.F. Cootes, G.J. Edwards, and Taylor C.J. Active appearance models. *Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 1

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. BM3D image denoising with shape-adaptive principal component analysis. In *Workshop on Signal Processing with Adaptive Sparse Structured Representations*, 2009. 7

[10] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3): 211–218, 1936. 1

[11] A. Eriksson and A. Hengel. Efficient computation of robust weighted low-rank matrix approximations using the L_1 norm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1681–1690, 2012. 1

[12] R. Garg, A. Roussos, and L. de Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1

[13] R. Garg, A. Roussos, and L. de Agapito. A variational approach to video registration with subspace constraints. *Int. J. Comput. Vision*, 104(3):286–314, 2013. 1

[14] M. Hein and M. Maier. Manifold denoising. In *Advances in Neural Information Processing Systems*, Cambridge, MA, USA, 2007. 6

[15] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):550–554, May 1994. ISSN 0162-8828. doi: 10.1109/34.291440. URL `http://dx.doi.org/10.1109/34.291440`. 5

[16] V. Larsson, C. Olsson, E. Bylow, and F. Kahl. Rank minimization with structured data patterns. In *Eur. Conf. Computer Vision*. 2014. 1, 2, 7, 8

[17] C. Olsson and M. Oskarsson. A convex approach to low rank matrix approximation with missing data. In *Scandinavian Conference on Image Analysis*, 2009. 1

[18] B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, August 2010. 1

[19] R.T. Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970. 3, 4

[20] D. Strelow. General and nested Wiberg minimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1

[21] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999. 4

[22] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Trans. Pattern Analysis and Machine Intelligence*, 31 (2):210–227, 2009. 1

[23] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *Trans. Pattern Analysis and Machine Intelligence*, 30(5):865–877, 2008. 1