Fast Automatic Background Extraction via Robust PCA

Ivan Papusha*

June 6, 2011

1 Introduction

Recent years have seen an explosion of interest in applications of sparse signal recovery and low rank matrix completion, due in part to the compelling use of the nuclear norm as a convex proxy for matrix rank. In some cases, minimizing the nuclear norm is equivalent to minimizing the rank of a matrix, and can lead to exact recovery of the underlying rank structure, see [Faz02, RFP10] for background. Even when exact recovery is not possible, the results often lead to practically useful starting points of analysis.

Classical Principal Component Analysis (PCA) is one of very few rank-constrained problems that can be solved exactly (via the singular value decomposition). In classical PCA, one would like to find the closest low-rank matrix (in spectral norm) to a given matrix. Because it assumes the corresponding noise is small, however, classical PCA fails in cases where a matrix is only a few sparse terms away from being low-rank, especially if the sparse terms have large magnitude.

Robust PCA naturally addresses this failure of classical PCA by applying ℓ_1-regularization on the matrix entries and using the nuclear norm as a global underestimator of rank. Hence it is robust to grossly corrupted observations of the underlying low-rank matrix. Under certain conditions, Robust PCA results in exact recovery [CLMW09]. Robust PCA has a number of existing applications in video surveillance, face recognition, semantic indexing, and collaborative filtering. For example, it can be used to automatically remove illumination from multiple images of a face under different illumination conditions. However, the technique requires solving a large optimization problem, with an objective that can be computationally intensive to evaluate.

This work explores several large-scale methods for solving the Robust PCA problem via Principal Component Pursuit (PCP), a convex optimization problem, and demonstrates its use in background extraction tasks in video and point-cloud LIDAR data. The idea is to stack the frames of data into a matrix, and then to find a low-rank approximation to the matrix with sparse error. The low-rank part of the answer corresponds to the (static) background parts of the video or 3D structure, and the sparse part to the (dynamic) foreground parts.

*Stanford Electrical Engineering department, Stanford, CA

One can then extract, classify, and track the sparse components, even in a multimodal setting. Robust PCA can also be used as a drop-in for classical PCA in *e.g.*, machine learning applications, whenever the data contain many outliers.

2 Previous work

Solving the Robust PCA problem by PCP was studied by [CLMW09, RFP10], who give conditions on exact recovery. Lin et al. [LCM09] and Arvind et al. [GLW$^+$09] studied fast algorithms based on the Alternating Direction Method of Multipliers (ADMM) and Nesterov-style Accelerated Proximal Gradient (APG), see [BPC$^+$10, Nes05, Nes07, BT09, Tse08] for background. Interior point and SDP algorithms for dealing with the matrix norm were considered by [LV09, RFP10]. Iterative reweighted least squares is an alternate approach [MF10]. Peng et al. [PGW$^+$10] give an extension to Robust PCA, called RASL, in the context of simultaneous image alignment and low-rank approximation. The Fast SVD algorithm [DFK$^+$04, DKM06] with Lanczos-style iteration allows for approximate singular value soft thresholding in the algorithm we propose. Dynamic object extraction from 3D point cloud data has been studied in [TSS08, STS08]. Typical RANSAC-based applications can be found in [FRS07] and [SRB08].

3 Robust PCA

To set up the problem, we let $M \in \mathbf{R}^{m\times n}$ be a data matrix, where the j-th column of M contains the (stacked) point coordinates from frame j, depending on the application. See §5.2 and §5.3 for how to generate M in a particular application. We are interested in writing $M = L + S$ as a sum of of a low-rank matrix L and a sparse matrix S. The Robust PCA problem is to solve

$$\begin{array}{ll} \text{minimize} & \mathbf{rank}(L) + \lambda\,\mathbf{card}(S) \\ \text{subject to} & L + S = M, \end{array} \tag{1}$$

over the (matrix) variables $L, S \in \mathbf{R}^{m\times n}$, where the parameter $\lambda > 0$ trades off rank against sparsity of the approximation. As it stands, the original problem is not convex, however, we can consider a convex relaxation, known as *Principal Component Pursuit (PCP)*,

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda\|S\|_1 \\ \text{subject to} & L + S = M, \end{array} \tag{2}$$

where $\|L\|_* = \sum_i \sigma_i(L)$ is the sum of the singular values of the matrix, also known as the *nuclear* (or *Ky Fan*) norm, and $\|S\|_1 = \sum_{ij} |S_{ij}|$ is the ℓ_1-norm of a matrix thought as a vector. The variables are $L, S \in \mathbf{R}^{m\times n}$, while M encodes the problem data. For the purposes of this work, the regularization parameter is fixed $\lambda = 1/\sqrt{\max(m,n)}$. In practice, this parameter choice works very well for the applications we consider; justification for this parameter selection can be found in [CLMW09].

Robust PCA is in contrast to classical PCA in the sense that the latter solves the (non-convex) problem,

$$\begin{array}{ll} \text{minimize} & \|M - L\|_2 \\ \text{subject to} & \mathbf{rank}(L) \leq k, \end{array}$$

with variable L, problem data M, and integer parameter $k \geq 1$. In classical PCA, M is approximated as a rank-k matrix with a small ℓ_2-error. Classical PCA is not robust to gross componentwise errors in the same way that Robust PCA is, since the former places a large penalty on large errors, while the latter allows for a few large errors while making most errors small. It is precisely the low-rank part and the small sparse errors that we are interested in when we talk about differentiating background from foreground.

4 Algorithms

4.1 Interior point methods

The nuclear norm in the PCP problem admits representation as a semidefinite program via the semidefinite embedding lemma [Faz02, §3.2]. The original non-convex problem 1 has the equivalent formulation,

$$\begin{array}{ll} \text{minimize} & (1/2)\,\mathbf{rank}(\mathbf{diag}(Y, Z)) + \lambda\|S\|_1 \\ \text{subject to} & \begin{bmatrix} Y & L \\ L^T & Z \end{bmatrix} \succeq 0, \quad L + S = M \end{array}$$

over the variables $L, S \in \mathbf{R}^{m\times n}$, and symmetric matrices $Y \in \mathbf{S}^m$, $Z \in \mathbf{S}^n$. Here $\succeq$ corresponds to matrix inequality with respect to the positive semidefinite cone. Since $Y, Z \succeq 0$, taking the nuclear norm of a block-diagonal, positive semidefinite matrix is the same as taking the trace. Therefore, the PCP problem is equivalent to a convex relaxation of the above,

$$\begin{array}{ll} \text{minimize} & (1/2)\,(\mathbf{Tr}(Y) + \mathbf{Tr}(Z)) + \lambda\|S\|_1 \\ \text{subject to} & \begin{bmatrix} Y & L \\ L^T & Z \end{bmatrix} \succeq 0, \quad L + S = M, \end{array}$$

which is an SDP, for which a general-purpose interior point solver like SeDuMi or SDPT3 can be used. Such methods usually cannot be used for problem sizes m, n larger than 50 or so, due to the large computational burden involved in inverting the KKT system. Since we are looking to apply PCP on matrices of size $m, n \sim 10^3$ or more, a general purpose interior-point method is intractable on current computers.

4.2 Subgradient methods

To get around the data size restriction, we can run a subgradient to solve the PCP problem, with the equality constraint $L + S = M$ eliminated,

$$\text{minimize} \quad \|L\|_* + \lambda\|M - L\|_1,$$

over the single variable L. The optimal sparse term $S^\star$ is recovered from the optimal low-rank term $L^\star$ by setting $S^\star = M - L^\star$. To find a subgradient, we use the fact that $\|\cdot\|_*$ for matrices is the dual norm of the ℓ_2- or spectral norm. That is, for any matrix $Z \in \mathbf{R}^{m\times n}$,

$$\|Z\|_* = \sup\{\mathbf{Tr}(Z^T X) \mid \|X\|_2 \leq 1\}.$$

If $Z = U\Sigma V^T$ is the singular value decomposition of Z, then UV^T is a subgradient of $\|Z\|_*$. This observation leads to an iterative method for solving the Principal Component Pursuit problem by calculating subgradients at each step. The stepsize α_k is chosen to be square summable but not summable,

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=1}^{\infty} \alpha_k = \infty.$$

Algorithm: Subgradient Method

Initialize: $L^{(1)} = \mathbf{1}\mathbf{1}^T$

For $k = 1, 2, \ldots$

1. Find the singular value decomposition: $L^{(k)} = U^{(k)}\Sigma^{(k)}V^{(k)T}$
2. Calculate a subgradient: $G^{(k)} = U^{(k)}V^{(k)T} - \lambda\,\mathbf{sign}(M - L^{(k)})$
3. Update: $L^{(k+1)} = L^{(k)} - \alpha_k G^{(k)}$

A problem with the subgradient method, besides slow convergence, is the costly full SVD executed at every step. One can approximate the SVD by a rank-k approximation $k < \min(m, n)$, see [DFK+04, DKM06] for randomized methods of approximating the SVD that lead to a stochastic subgradient method. Because approximating the SVD by these methods leads to a biased subgradient estimate, however, the method is no longer guaranteed to converge to the optimal value. In practice, we saw reasonable convergence to a (suboptimal) answer using just the top singular vectors.

4.3 Alternating direction method of multipliers

As posed, the PCP problem admits natural decomposition into Alternating Direction Method of Multipliers (ADMM) form, see [BPC+10] for background. The augmented Lagrangian for the equality constrained PCP problem with penalty parameter $\rho > 0$ is

$$\begin{aligned}\mathcal{L}_\rho(L, S, Y) &= \|L\|_* + \lambda\|S\|_1 + \mathbf{Tr}(Y^T(L + S - M)) + (\rho/2)\|L + S - M\|_F^2 \\ &= \|L\|_* + \lambda\|S\|_1 + (\rho/2)\|L + S - M + W\|_F^2 + \text{const},\end{aligned}$$

where $Y = \rho W \in \mathbf{R}^{m\times n}$ is a dual variable. The scaled dual form of ADMM follows.

Algorithm: ADMM

Initialize: $L^{(1)} = \mathbf{1}\mathbf{1}^T$, $S^{(1)} = M - L^{(1)}$, and $W^{(1)} = (1/\rho)\mathbf{1}\mathbf{1}^T$

For $k = 1, 2, \ldots$

1. $L^{(k+1)} := \operatorname{argmin}_L \left(\|L\|_* + (\rho/2)\|L + S^{(k)} - M + W^{(k)}\|_F^2 \right)$
2. $S^{(k+1)} := \operatorname{argmin}_S \left(\lambda\|S\|_1 + (\rho/2)\|L^{(k+1)} + S - M + W^{(k)}\|_F^2 \right)$
3. $W^{(k+1)} := W^{(k)} + (L^{(k+1)} + S^{(k+1)} - M)$ (dual update)

Each of the minimizations has a closed form in terms of the soft thresholding operator, S_κ, $\kappa > 0$, where

$$S_\kappa(x) = \begin{cases} x - \kappa & x > \kappa \\ 0 & |x| \leq \kappa \\ x + \kappa & x < -\kappa. \end{cases}$$

The minimization in step 1 corresponds to a proximal operator of the nuclear norm evaluated at $M - S^{(k)} - W^{(k)}$. The prox is calculated by soft thresholding on the singular values of the matrix. That is, if $\sigma_1, \ldots, \sigma_r$ are the singular values of $M - S^{(k)} - W^{(k)}$, then the L-update, in terms of the SVD, is

$$L^{(k+1)} := U\,\mathbf{diag}(S_{1/\rho}(\sigma_1), \ldots, S_{1/\rho}(\sigma_r))V^T, \quad U\Sigma V^T = M - S^{(k)} - W^{(k)}.$$

Similarly, step 2 is computed by (elementwise) soft thresholding, with i,j-entry of the matrix $S^{(k+1)}$ given by the update,

$$S_{ij}^{(k+1)} := S_{\lambda/\rho}\left(M_{ij} - L_{ij}^{(k+1)} - W_{ij}^{(k)}\right), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

Given an upper bound on the rank of the matrix, ADMM combined with a fast SVD routine (Lanczos iteration) and adaptive stepsize (see [BPC+10, §3.4.1]) yields a state of the art fast method for solving PCP.

4.4 Nesterov accelerated methods

To accelerate convergence of the subgradient method, a Nesterov-style proximal gradient algorithm can be used, see [Tse08] for a review of current methods. With careful tuning, accelerated algorithm can have theoretical $\mathcal{O}(1/k^2)$ convergence, where k is the number of iterations. If we define $f : \mathbf{R}^{m\times n} \to \mathbf{R}$ as the objective

$$f(L) = \|L\|_* + \lambda\|M - L\|_1,$$

we can then define a "Nesterized" proximal point algorithm for minimizing the nonsmooth convex f. Here the stepsize parameters are given by $\theta_k = 2/(k+1)$, and $t_k = t$ is held constant for all iterations.

Algorithm: (Accelerated) Proximal Point

Initialize: $x^{(0)} = y^{(0)} \in \mathbf{dom}\, f$

For $k = 1, 2, \ldots$

1. $x^{(k)} := \mathbf{prox}_{t_k f}(y^{(k-1)}) = \mathrm{argmin}_u \left(f(u) + (1/2t_k)\|u - y^{(k-1)}\|_2^2 \right)$
2. $v^{(k)} := x^{(k-1)} + (1/\theta_k)(x^{(k)} - x^{(k-1)})$
3. $y^{(k)} := (1 - \theta_{k+1})x^{(k)} + \theta_{k+1} v^{(k)}$

The ℓ_2-norm in step 1 becomes the Frobenius norm for the matrix case. Unfortunately, the accelerated proximal point algorithm does not fare as well as we might have hoped for the PCP problem; poor convergence has been seen in [LCM09], as well, although adaptive restarting might work well [O'D11].

5 Results

5.1 Small example

As a toy example, we solve the PCP problem for a small $M = L_0 + S_0 \in \mathbf{R}^{30 \times 10}$, where we picked a random L_0 with $\mathbf{rank}(L_0) = 3$ and random S_0 with 10% nonzero entries. The optimal objective value $f^\star$ was calculated using 53 iterations of an interior-point method. Figure 1 shows typical convergence results. Both ADMM and Accelerated Proximal Gradient (APG) reach a good initial accuracy within about 100 iterations, however, ADMM and the subgradient method eventually reach better precision than APG.

Such convergence curves are typical for larger data sets as well. The inability of APG to give good accuracy lies in part due to the lack of restarting, which has been shown to perform well in other applications. Furthermore, different choices for the stepsize parameter θ_k can lead to better convergence in APG.

5.2 Aligned video

Video fits nicely within the Robust PCA framework, assuming the video frames are aligned to one another, the action in the foreground is continuous, and the background constitutes most of the static data. Note that in particular, these assumptions disallow sudden cuts to a different scene or scenes with camera movement.

Here, we stack the j-th frame of the video as the j-th column of M. PCP deals well with both color and grayscale video. Using color increases the vertical dimension m by a factor of 3 from grayscale, increasing the cost of each iteration. Results of PCP applied to a video of tennis point are presented in Figure 2.

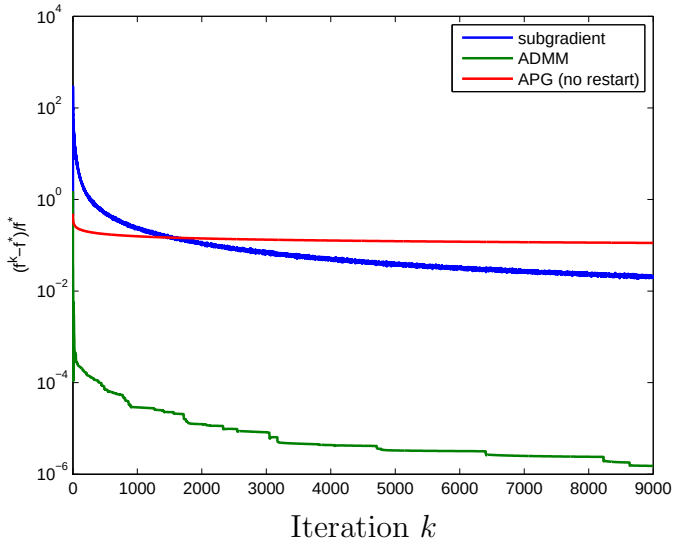

Figure 1: Convergence of a Subgradient, ADMM, and Accelerated Proximal Gradient (APG) algorithms for a small example data set $M \in \mathbf{R}^{30\times 10}$.

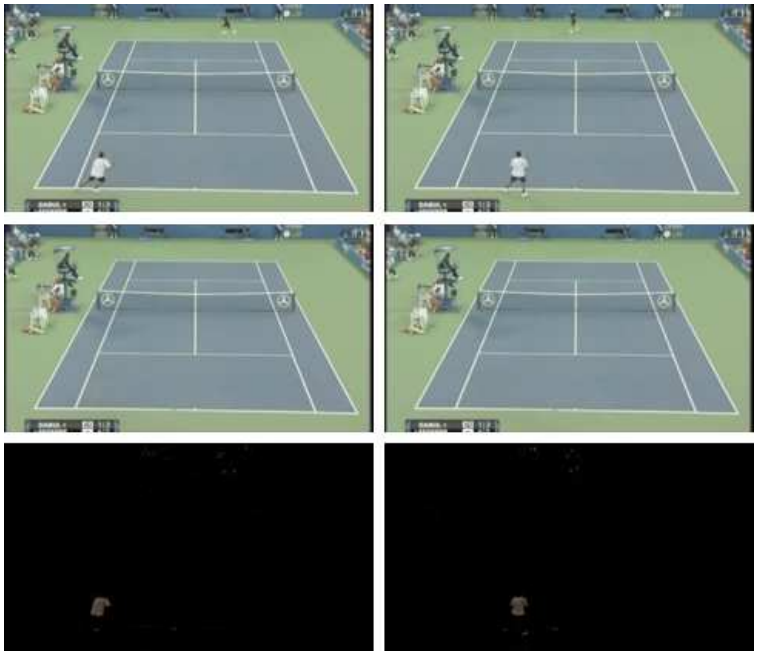

Figure 2: Frames 5 (left) and 20 (right) of a video of two tennis players. Here the video contains 100 frames, each of which has $250 \times 135 \times 3$ (color) pixels, implying $M \in \mathbf{R}^{97200\times 100}$. Visualization of the original frame (top), the low-rank part L (middle), and the sparse part S (bottom) for each frame has allowed the extraction of the court (background) and the moving players (foreground), as well as the ball, within ~ 40 iterations of ADMM (~ 3 minutes).

5.3 LIDAR range data

Robust PCA can be used in video to relate background (low rank) to the foreground (sparse) parts of a sequence of images. The same technique be used to relate the background to the foreground of 3D laser range scan data. Here, a LIDAR scanner atop Stanford's *Junior* autonomous car was used to gather a 3D point cloud "video" of an intersection on the Stanford campus. In addition to (relatively) static buildings, trees, and bushes, the data exhibit moving shapes that correspond to pedestrians and bicyclists as they enter and exit the scanner's field of view. See Figure 3 for PCP applied to this data modality.

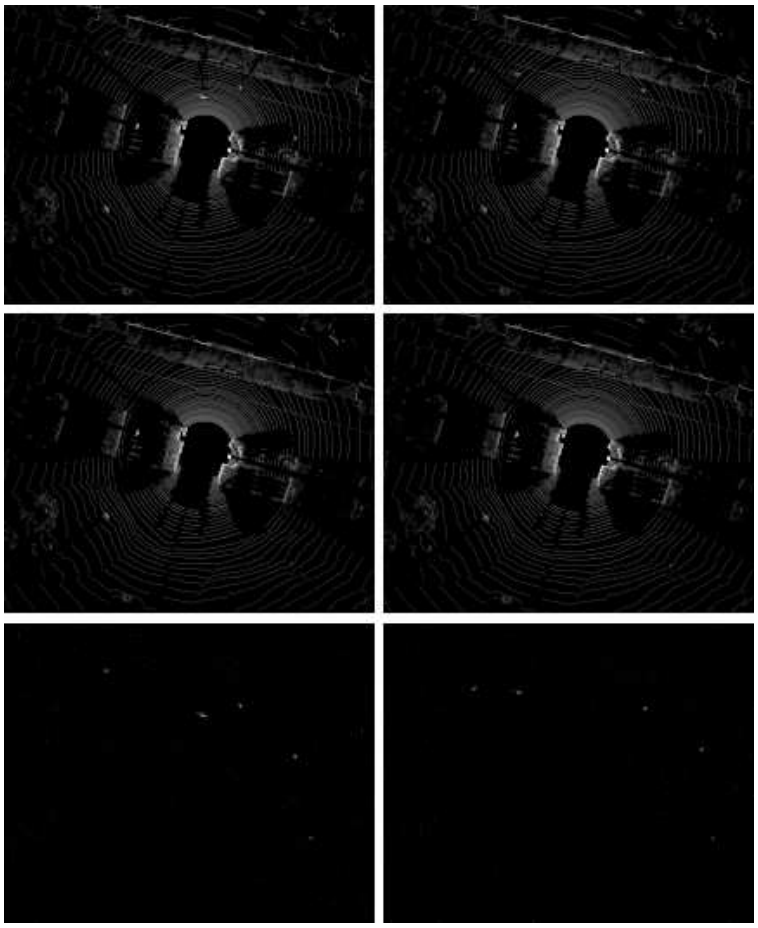

Figure 3: Frames 180 (left) and 200 (right) of LIDAR data, top view. The 200 frames, consisting of a 2D point cloud histogram, are each 214×173 grayscale, implying $M \in \mathbf{R}^{37022 \times 200}$. Visualization of the original frame (top), the low-rank part L (middle), and the sparse part S (bottom) for each frame has allowed the extraction of the surrounding map (background) and the moving bicyclists and pedestrians (foreground) within ~ 30 iterations of ADMM (~ 1 minutes).

A major issue with 3D LIDAR data is point registration. Unlike in raster video (as in the example of the previous section), a 3D point that appears in one frame may not appear

in the same place during the next frame due to the construction of the 3D scanner. To get over this difficulty, we can perform "voxel binning," a rasterization step where we discretize the 3D environment into voxels and associate with each voxel a count of the number of 3D points that appears within its confines. Care must be taken to select the right size for the discretized voxels — too large, and not enough granularity is achieved for the mapping application; too small, and binning reduces back to the 3D pixel problem, as well as increased computational burden due to increased problem dimensions. Certain 3D applications also allow one to compress the voxel data into two dimensions, as in Figure 3.

6 Conclusion

Robust PCA via PCP allows for a general purpose way to extract background from foreground in image-like data modalities, such as video and LIDAR data. The ability to exactly recover a low-rank and sparse approximation to a matrix by solving PCP results in a general-purpose mapping algorithm that depends only on the problem data, and has no parameters to tune. Even when exact recovery is not possible, PCP acts as a great heuristic.

In practice, ADMM is the method of choice to solve PCP, although accelerated proximal gradient ("Nesterov") methods can still be improved. A criticism of Robust PCA is that it is not real time. One can imagine an application, however, where a robot sits at the same spot for a minute or two, scanning its dynamic environment. At any particular instance of time, the robot does not have access to the full map because of occlusions or shadows imposed by moving objects. Over the lifetime of the scan, however, the robot will have seen the entire environment. It is exactly in this regime that Robust PCA works extremely well.

In fact, under sparsity assumptions on the objects being scanned, PCP is *guaranteed* to extract the background from the foreground. The background, *i.e.*, the low-rank part of the scan matrix, can be used as a high fidelity map of the environment, while the foreground, *i.e.*, the sparse part of the scan matrix, can further be processed with a classifier. In other words, Robust PCA becomes a *region of interest* finder. Only those parts of the data that correspond the region of interest appear in the foreground. This allows for automatic a posteriori tracking and high fidelity classification of moving objects.

Acknowledgments

The LIDAR data are courtesy of the Stanford Junior Racing team. Thanks go to Emmanuel Candes and Stephen Boyd for helpful suggestions. In addition, thanks go to the optimization team: Brendan O'Donoghue, Matt Kraning, Arezou Keshavarz, Borja Peleato, Eric Chu, Ekine Akuiyibo, Yang Wang, and Neal Parikh for helping me with technical (and nontechnical) tidbits here and there. Final gratitudes go to the rest of the team-quad-roommate-four-squad: Vivek Athalye, Ved Gund, and Geoff Schiebinger for helpful discussions and welcome distractions. Adieu, Stanford EE and the undergrad years, and thanks for all the fish (or something like that).

References

[BPC+10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning (to appear)*, November 2010.

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[CLMW09] Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis. Technical report, Stanford University, 2009.

[DFK+04] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.

[DKM06] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[Faz02] Maryam Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, March 2002.

[FRS07] Daniele Fontanelli, Luigi Ricciato, and Stefano Soatto. A fast ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements. In *IEEE Conference on Automation Science and Engineering*, pages 597–602, September 2007.

[GB11] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, April 2011.

[GLW+09] Arvind Ganesh, Zhouchen Lin, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast algorithms for recovering a corrupted low-rank matrix. In *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, December 2009.

[LCM09] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical Report UILU-ENG-09-2215, UIUC, arXiv:1009.5055v2, November 2009.

[LV09] Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

[MF10] Karthik Mohan and Maryam Fazel. Iterative reweighted least squares for matrix rank minimization. In *Allerton Conference on Communications, Control, and Computing*, September 2010.

[Nes05] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

[Nes07] Yurii Nesterov. Gradient methods for minimizing composite objective function. ECORE Discussion Paper 2007/96, Université catholique de Louvain, `http://www.ecore.be/DPs/dp_1191313936.pdf`, September 2007.

[O'D11] Brendan O'Donoghue. Adaptive restarting for first-order optimization methods. Stanford MATH 301 class project, March 2011.

[PGW+10] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. RASL: Robust Alignment by Sparse and Low-rank decomposition for linearly correlated images. In *CVPR*, 2010.

[RFP10] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[SRB08] Dominik Steinhauser, Oliver Ruepp, and Darius Burschka. Motion segmentation and scene classification from 3d lidar data. In *IEEE Intelligent Vehicles Symposium*, pages 398–403, June 2008.

[STS08] Luciano Spinello, Rudolph Triebel, and Roland Siegwart. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2008.

[Tse08] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM Journal on Optimization (submitted)*, May 2008.

[TSS08] Rudolph Triebel, Luciano Spinello, and Roland Siegwart. Extraction of dynamic objects from 3d point clouds using multimodal detection and tracking. In *Proceedings of The Workshop of the Workshop on 3D-Mapping (IROS)*, 2008.