VIDEO BACKGROUND SUBTRACTION USING SEMI-SUPERVISED ROBUST MATRIX COMPLETION

Hassan Mansour, Anthony Vetro

Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139, USA
mansour@merl.com, vetro@merl.com

ABSTRACT

We propose a factorized robust matrix completion (FRMC) algorithm with global motion compensation to solve the video background subtraction problem. The algorithm decomposes a sequence of video frames into the sum of a low rank background component and a sparse motion component. The algorithm alternates between the solution of each component following a Pareto curve trajectory for each subproblem. For videos with moving background, we utilize the motion vectors extracted from the coded video bitstream to compensate for the change in the camera perspective. Performance evaluations show that our approach is faster than state-of-the-art solvers and results in highly accurate motion segmentation.

Index Terms— Motion segmentation, foreground / background separation, robust matrix completion, robust principal component analysis, global motion estimation.

1. INTRODUCTION

Background subtraction is the problem is of finding moving objects in video that are independent of the background scene. The segmentation of moving objects helps in analyzing the trajectory of moving targets and in improving the performance of object detection and classification.

Motion segmentation algorithms can be classified into algebraic decomposition techniques [1–4] and statistical motion flow techniques [5–8]. Algebraic approaches generally model the background scene as a low dimensional subspace. The moving objects are then separated as the error terms that live in the orthogonal complement of the background subspace. When the camera is stationary, the low dimensional subspace is low rank and algorithms such as robust principle component analysis (RPCA) have been shown to successfully segment the foreground from the background [1, 2, 9]. When the camera is moving, the low rank structure no longer holds and adaptive subspace estimation techniques are used to track the background subspace [4, 10]. Statistical motion flow techniques generally utilize Gaussian mixture models of image plane motion to capture the trajectories of moving objects.

In this paper, we follow an algebraic approach to solving the background subtraction problem using robust matrix completion. We develop in section 2 a fast and memory efficient algorithm that decomposes a group of video pictures (GOP) into a low rank component corresponding to the background and a sparse component corresponding to the moving objects in the scene. The algorithm uses factorized matrix decomposition with a pre-specified rank to compute the low rank component similar to the approach adopted by Aravkin et al. [11]. We then adopt a block coordinate descent approach using spectral projected gradient steps to alternate between the solution of the low rank component and the sparse component all the while traversing the updated Pareto curve of each subproblem. In scenes that exhibit camera motion, we first extract the motion vectors from the coded video bitstream and fit the global motion of every frame to a parametric perspective model with 8 parameters described in section 3. We then align the frames to match the perspective of the first frame in the GOP and use our factorized robust matrix completion algorithm to fill in the background pixels that are missing from the individual video frames in the GOP. In sections 4 and 5, we demonstrate how our algorithm can be run in batch mode and online and how it does not require any training step to learn the initial subspace.

2. FACTORIZED ROBUST MATRIX COMPLETION

In this section, we describe the factorized robust matrix completion (FRMC) algorithm we use to solve background subtraction problem.

Suppose that we are given a data matrix $Y \in \mathbb{R}^{m\times n}$ that is composed of the sum of a low rank component X_0 and a sparse component S_0, such that, $Y = X_0 + S_0$. Let $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ be a restriction operator that selects a subset Ω of size p of the mn samples in Y. We define the robust matrix completion problem as the problem of finding X_0 and S_0 from incomplete measurements $b = \mathcal{A}(Y)$. Several works in the literature [1,2] address this problem by formulating it as the multi objective minimization problem

$$\min_{X,S} \|X\|_* + \lambda\|S\|_1 \text{ subject to } b = \mathcal{A}(X+S), \qquad (1)$$

where λ is a positive weighting parameter. When the entries of Y are fully observed, i.e., $\mathcal{A}$ is an identity matrix, the problem is known as the robust principal component analysis (RPCA) problem and the optimization problem (1) is referred to as principal component pursuit. Moreover, it was shown in [2] that a choice of $\lambda = \hat{n}^{-1/2}$, $\hat{n} := \max\{m, n\}$, is sufficient to guarantee the recovery of X_0 and S_0 with high probability when the rank$(X_0) \leq C\hat{n}\left(\log\hat{n}\right)^{-2}$ for some constant C that depends on the coherence of the subspace of X_0.

One of the main drawbacks of problem (1) is that it requires the computation of full (or partial) singular value decompositions of X in every iteration of the algorithm, which could become prohibitively expensive when the dimensions are large. To overcome this problem, we adopt a proxy for the nuclear norm of a rank-r matrix X defined by the following factorization from Lemma 8 in [12]

$$\|X\|_* = \inf_{L\in\mathbb{R}^{m,r}, R\in\mathbb{R}^{n,r}} \frac{1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right) \text{ s.t. } LR^T = X. \quad (2)$$

The nuclear norm proxy has recently been used in standard nuclear norm minimization algorithms [11,13] that scale to very large matrix

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

completion problems. Moreover, it was shown in Corollary 4.2 of [11] that when the factors L and R have a rank greater than or equal to the true rank of X, then a spectral projected gradient algorithm utilizing the nuclear norm proxy is guaranteed to converge to the solution of the corresponding nuclear norm minimization problem.

Our FRMC algorithm extends the approach of [11] to the robust matrix completion problem by alternating between the solutions of two subproblems as shown in Algorithm 1:

Algorithm 1 Factorized Robust Matrix Completion (FRMC)

Input $b = \mathcal{A}(X_0 + S_0)$, tolerance σ
Output L, R, S
Initialize $S = 0$, L, R random Gaussian entries, $\tau_S = \|S\|_1$, $\tau_X = \frac{1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right)$
while $\|b - \mathcal{A}(X + S)\|_F \leq \sigma$ **do**
  Solve for low rank component:
  $b_{LR} = b - \mathcal{A}(S)$
  $M = b_{LR} - \mathcal{A}(LR^T)$
  $\tau_X = \tau_X + \left(\|M\|_F - \sigma\right) / \|\mathcal{A}^T(M)\|_2$

$$\min_{L,R} \|b_{LR} - \mathcal{A}(LR^T)\|_F \text{ s.t. } \|L\|_F^2 + \|R\|_F^2 \leq 2\tau_X \quad (3)$$

  Solve for sparse component:
  $b_S = b - \mathcal{A}(LR^T)$
  $M = b_S - \mathcal{A}(S)$
  $\tau_S = \tau_S + \left(\|M\|_F^2 - \sigma\|M\|_F\right) / \|\mathcal{A}^T(M)\|_\infty$

$$\min_{S} \|b_S - \mathcal{A}(S)\|_F \text{ s.t. } \|S\|_1 \leq \tau_S \quad (4)$$

end while

Each subproblem in the FRMC algorithm is a LASSO problem that we solve using spectral projected gradient iterations. The rationale behind this approach follows from the work of van den Berg and Friedlander [14] on the SPGL1 solver for the basis pursuit denoise problem. For every fixed sparse component S with $b_{LR} = b - \mathcal{A}(S)$, the sequence of iterates $X_\tau = LR^T$, where (L, R) is the solution to (3), are samples on the Pareto curve of the nuclear norm minimization problem

$$\min_X \|X\|_* \text{ s.t. } \|b_{LR} - \mathcal{A}(X)\|_F \leq \sigma.$$

Moreover, the update rule of τ_X is a Newton root finding step of the problem $\phi(\tau_X) = \sigma$, where

$$\phi(\tau_X) = \min_{L,R} \|b_{LR} - \mathcal{A}(LR^T)\|_F \text{ s.t. } \frac{1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right) \leq \tau_X.$$

Consequently, the FRMC algorithm switches between traversing the Pareto curves of the nuclear norm minimization problem and the ℓ_1 norm minimization problem. For every subproblem, the Pareto curve is updated to the new value of S and the function $\phi(\tau_X)$ is minimized by following a Newton step on the new Pareto curve. The same analysis applies to S when L and R are fixed. Therefore, the algorithm guarantees that the exit condition $\|b - \mathcal{A}(X + S)\|_F \leq \sigma$ will be satisfied.

We note that our framework in Algorithm 1 bares similarity to the SpaRCS algorithm [9]. The main difference is that SparCS follows a greedy approach that iteratively estimates the low rank subspace of X as well as the support of S followed by truncated SVD and least squares inversion to compute estimates for X and S.

3. GLOBAL MOTION PARAMETRIZATION

In videos where the camera itself is moving, applying the FRMC algorithm directly to the video frames fails in segmenting the correct motion since the background itself is non-stationary. A non-stationary background does not live in a low rank subspace, therefore, we can only expect the algorithm to fail. Therefore, we first estimate the global motion parameters in the video in order to compensate for the camera motion. We then align the background and apply the FRMC algorithm to segment the moving objects.

Global motion estimation received a lot of attention from the research community during the development of the MPEG-4 Visual standard [15]. One approach relates the coordinates (x_1, y_1) in a reference image I_1 to the coordinates (x_2, y_2) in a target image I_2 using an 8-parameter homography vector h such that

$$\begin{aligned} x_2 &= \frac{h_0 + h_2 x_1 + h_3 y_1}{1 + h_6 x_1 + h_7 y_1} \\ y_2 &= \frac{h_1 + h_4 x_1 + h_5 y_1}{1 + h_6 x_1 + h_7 y_1}. \end{aligned} \quad (5)$$

Given the homography vector $h = [h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7]^T$ that relates two images, we can warp the perspective of image I_2 to match that of image I_1, thereby aligning the backgrounds of both images. However, estimating h from the raw pixel domain requires finding point-to-point matches between a subset of the pixels of the two images. In order to compute h, we propose to use the horizontal and vertical motion vectors (m_x, m_y) that are readily available from the compressed video bitstream or during the encoding process. Here we assume that motion estimation is performed using the previous video frame as the only reference picture. The motion vectors provide relatively accurate point matches between the two images. Note, however, that we are only interested in matching pixels from the moving background. Therefore, we first compute a 32 bin histogram of each of the motion vectors m_x and m_y. Next, we extract a subset Λ of the indices of pixels whose motion vectors are shared by at least 20% of the pixels in the frame. Our assumption here is that foreground objects correspond to less than 20% of the moving pixels in the image. This threshold may of course vary between different scenes and video sequences. We then use the motion vectors indexed by Λ to estimate the homography parameter vector h by solving the following least squares problem:

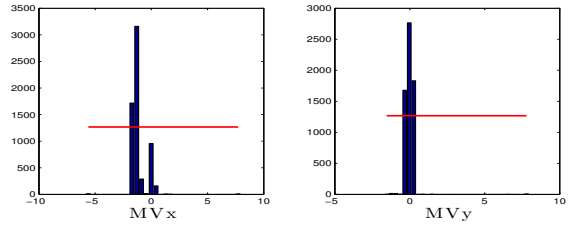


Fig. 1: Histograms of the horizontal (left) and vertical (right) motion vectors of frame 26 of the Bus sequence. The red line indicates the 20% pixel cutoff we use to distinguish between the background and foreground motion.

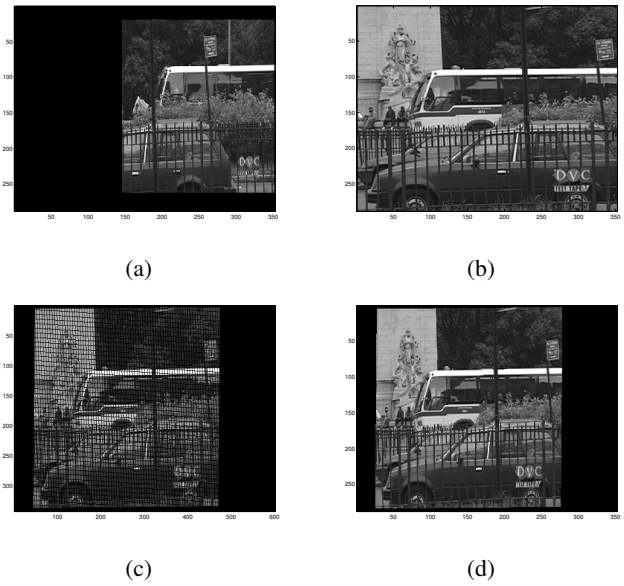
$$h = \arg\min_{\tilde{h}} \left\| p_\Lambda - E\tilde{h} \right\|_2, \quad (6)$$

where $p_\Lambda = \begin{bmatrix} x_{2\Lambda} \\ y_{2\Lambda} \end{bmatrix}$, $x_{2\Lambda} = x_{1\Lambda} + m_{x\Lambda}$, $y_{2\Lambda} = y_{1\Lambda} + m_{x\Lambda}$, and the matrix

$$E = \begin{bmatrix} \mathbf{1} & \mathbf{0} & x_{1\Lambda} & y_{1\Lambda} & \mathbf{0} & \mathbf{0} & -x_{2\Lambda}x_{1\Lambda} & -x_{2\Lambda}y_{1\Lambda} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & x_{1\Lambda} & y_{1\Lambda} & -y_{2\Lambda}x_{1\Lambda} & -y_{2\Lambda}y_{1\Lambda} \end{bmatrix},$$

where the subscript $_\Lambda$ indicates a restriction of the indices to the set Λ. Fig. 1 illustrates the histograms and cutoff thresholds of the vertical and horizontal motion vectors from frame 26 of the Bus sequence. Notice the spike at zero in the Mvx histogram that falls under the 20% threshold. This spike corresponds to the bus in the sequence which appears stationary relative to the camera perspective when in fact it is in motion. Our approach correctly captures the motion of the background and fits the homography parameters to the background pixels alone.

Next, we align the pictures relative to the perspective of the first frame in a GOP by sequentially warping the pictures using the coordinates of the previously warped frame $\hat{I}_1$ as reference to warp the coordinates of the next frame I_2 by applying (5). Finally, we note that due to the camera motion, the warped frames $\hat{I}_2$ generally occupy a larger viewing area relative to the reference frame I_2. Consequently, applying a forward map $f : (x_1, y_1) \rightarrow (\hat{x}_2, \hat{y}_2)$ often results in holes in the warped frame. To remedy this problem, we compute the reverse mapping $g : (\hat{x}_2, \hat{y}_2) \rightarrow (x_2, y_2)$ as a function of h and warp the frame to obtain $\hat{I}_2(\hat{x}_2, \hat{y}_2) = I_2(g(\hat{x}_2, \hat{y}_2))$. Fig. 2 illustrates the global motion compensation procedure applied to frame 26 of the Bus sequence.



(a) (b) (c) (d)

Fig. 2: Example of the global motion compensation procedure used to align the backgrounds of images in a GOP. (a) First frame in the GOP aligned and scaled to its relative location. (b) Original frame 26 as input image I_2. (c) Frame 26 warped and aligned as $\hat{I}_2(\hat{x}_2, \hat{y}_2)$, (d) Warped and reverse mapped frame $\hat{I}_2(g(\hat{x}_2, \hat{y}_2))$.

4. BACKGROUND SUBTRACTION VIA FRMC

Once the images are warped and aligned using global motion compensation, we vectorize the images and stack them into a matrix Y of size $m \times n$, where m is the number of pixels in the enlarged GOP frame and n is the number of frames in the GOP. As shown in Fig. 2 (a) and (d), the warped images contain large areas where there are no intensity measurements. Therefore, we construct a restriction operator $\mathcal{A}$ that identifies the pixels that contain intensity values. Applying $\mathcal{A}$ to Y results in a vector $b = \mathcal{A}(Y)$ that only contains the the pixels with intensity values. Our objective then is to compute a low-rank approximation X of Y that should correspond to the background pixels, and a sparse component S that captures the moving objects in the scene, such that, $\|b - \mathcal{A}(X + S)\|_F \leq \sigma$ where σ is a user defined error tolerance.

4.1. FRMC in batch mode

In batch mode, disjoint groups of n video frames are warped and aligned separately into matrices Y_j, where j indicates the GOP number. The FRMC algorithm is then applied to each matrix Y_j resulting in the background matrix X_j and the foreground matrix S_j. Note that every column of S_j is a vectorization of a single video frame containing the moving objects in the GOP.

4.2. FRMC in online mode

In online mode, we first extract a background estimate $X_1 = L_1 R_1^T$ using the first $n > 1$ video frames. The number n can be chosen to satisfy a maximum delay requirement. For every subsequent frame indexed by $i = n + 1, \ldots, N$, we align L_{i-1} with the perspective of frame i to produce $\widehat{L}_{i-1}$ and match the background $X_{i-1} = L_{i-1} R_{i-1}^T(1, \rightarrow)$ to that of the new frame. We then perform a single gradient update of (3) initialized with $(\widehat{L}_{i-1}, R_{i-1}(1, \rightarrow))$, where $R_{i-1}(1, \rightarrow)$ is the first row in the matrix R_{i-1}.

In order to speed up the computation of the sparse component, we replace (4) with a small number of approximate message passing (AMP) [16] iterations

$$\begin{aligned} S^{t+1} &= \eta\left(S^t + \mathcal{A}^T z^t, \gamma\right), \\ z^{t+1} &= b - \mathcal{A}S^{t+1} + \tfrac{\|S^{t+1}\|_0}{mn} z^t \end{aligned} \tag{7}$$

where $z^0 = b$, $S^0 = 0$, η is the soft-thresholding operator

$$\eta(x, \gamma) = \begin{cases} \text{sign}(x)(|x| - \gamma) & |x| \geq \gamma \\ 0 & \text{otherwise} \end{cases},$$

and γ is an adaptive threshold we set equal to the mode of the histogram of S^t.

5. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our approach in separating the background from video sequence with both stationary and moving cameras. We ran the experiments using a pure MATLAB implementation of our algorithm on a 3.2 GHz Intel Core i5 Mac with 16GB RAM. In all of our experiments, we set the rank of the factors L and R equal to one since we found no demonstrable improvement in performance that justifies the increased complexity and memory requirements. Moreover, stationary background scenes are rank one objects.

5.1. Stationary background

For stationary background scenes, we apply the FRMC algorithm directly to the pixel domain, skipping the frame alignment. We test our algorithm on the Shopping Mall video sequence[1]. Fig. 3 com-

[1] Available from:
http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

pares the qualitative separation performance of FRMC to that of the state-of-the-art algorithm GRASTA [3]. The FRMC algorithm completes the recovery 7 to 8 times faster than GRASTA and results in a comparable separation quality. For a quantitative comparison, we plot the ROC curves of the two algorithms in Fig. 4. The curves show that GRASTA achieves a slightly better accuracy than FRMC, however, the computational cost is considerably higher.

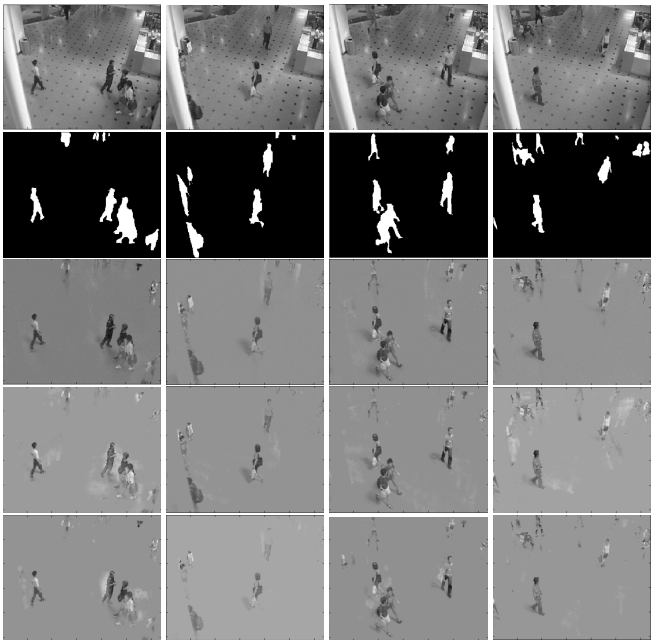


Fig. 3: Background subtraction of four frames from the Shopping Mall sequence. Row one shows the original four frames. Row two shows the ground truth foreground objects. Row three shows the output of the GRASTA algorithm which required 389.7 seconds to complete. Row four shows the output of our FRMC algorithm running in batch mode and completing in 47.1 seconds. Row five shows the output of the FRMC algorithm running in online mode and completing in 55.4 seconds.

5.2. Non-stationary background

For non stationary background sequences, we run our FRMC algorithm with global motion compensation on the reference video sequence Bus composed of 150 CIF resolution (352×288 pixels) frames[2]. The Bus sequence exhibits translation and zooming out. We use the HEVC test model (HM) 11 reference software[3] [17] to encode the sequence and run our FRMC with GME algorithm in batch mode with a batch size of 30 frames. The recovery performance is illustrated in Fig. 5. Notice how the recovered background expands and stretches relative to the original frames in order to cover the translation and zoom of the 30 frame GOP. Notice also how stationary foreground objects are successfully classified as part of the background subspace and are excluded from the segmented moving objects. Finally, we note that we ran the t-GRASTA algorithm [4] that performs an adaptive subspace estimation to capture the variation in the background on the same video sequence. However, t-GRASTA failed completely at segmenting the moving objects.Moreover, statistical motion flow algorithms also struggle with highly non stationary backgrounds.

[2]Available from: http://trace.eas.asu.edu/yuv/

[3]Available from: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

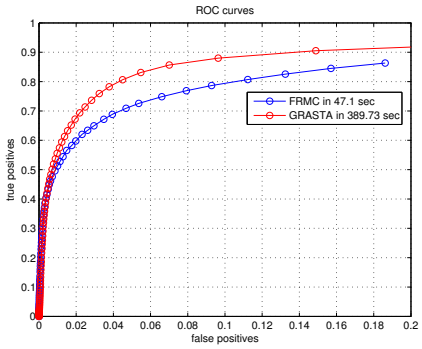


Fig. 4: ROC curves comparing the stationary background subtraction performance between GRASTA and batch-FRMC.

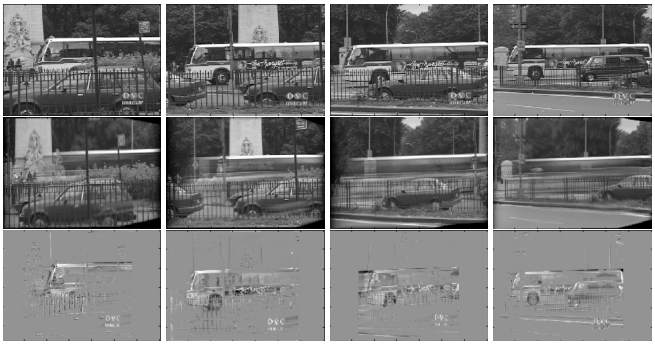


Fig. 5: Background subtraction of four frames from the Bus sequence. Row one shows the original four frames. Row two shows the motion aligned and FRMC separated background relative to a 30 frame GOP. Row three shows the motion aligned and FRMC separated foreground. The total global motion compensation and background subtraction for 150 frames is 19.8 seconds.

6. CONCLUSION AND FUTURE WORK

We proposed a video background subtraction algorithm based on robust matrix completion with global motion compensation. Our FRMC algorithm decomposes a sequence of video frames into the sum of a low rank background component and a sparse motion component. The algorithm alternates between the solution of each component following a Pareto curve trajectory for each subproblem. For videos with moving background, we utilize the motion vectors extracted from the coded video bitstream to compensate for the change in the camera perspective. Performance evaluations show that our approach is faster than state-of-the-art solvers and results in highly accurate motion segmentation for both stationary and non-stationary scenes.. In fact, the algorithm can estimate the perspective parameters, align the frames, and segment CIF resolution videos at a rate that exceeds 10 frames per second using a pure MATLAB implementation. For future work, we plan to analyze the convergence of the FRMC algorithm and improve the matching performance of the global motion compensation inspired by the new work of [18].

7. ACKNOWLEDGEMENTS

The authors would like to thank Shantanu Rane and Dong Tian for their helpful discussions and feedback regarding this work.

8. REFERENCES

[1] J. Wright, A. Ganesh, S. Rao, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems 22*, 2009.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, June 2011.

[3] J. He, L. Balzano, and J. C. S. Lui, "Online robust subspace tracking from partial information," *preprint, http://arxiv.org/abs/1109.3827*, 2011.

[4] J. He, D. Zhang, L. Balzano, and T. Tao, "Iterative grassmannian optimization for robust image alignment," *preprint, http://arxiv.org/abs/1306.0404*, 2013.

[5] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *ICCV*, 1998.

[6] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *ICCV*, 2009.

[7] A. Elqursh and A. M. Elgammal, "Online moving camera background subtraction," in *ECCV*, 2013.

[8] E. Learned-Miller M. Narayana, A. Hanson, "Coherent motion segmentation in moving camera videos using optical flow orientations," in *ICCV*, 2013.

[9] Andrew E. Waters, Aswin C. Sankaranarayanan, and Richard Baraniuk, "Sparcs: Recovering low-rank and sparse matrices from compressive measurements," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, Eds., pp. 1089–1097. 2011.

[10] Chenlu Qiu and Namrata Vaswani, "Reprocs: A missing link between recursive robust pca and recursive sparse recovery in large but correlated noise," *CoRR*, vol. abs/1106.3286, 2011.

[11] A.Y. Aravkin, R. Kumar, H. Mansour, B. Recht, and F.J. Herrmann, "A robust svd free approach to matrix completion, with applications to interpolation of large scale data," *preprint. http://arxiv.org/abs/1302.4886*, 2013.

[12] Nathan Srebro, *Learning with matrix factorizations*, Ph.D. thesis, Cambridge, MA, USA, 2004, AAI0807530.

[13] Benjamin Recht and Christopher Ré, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Mathematical Programming Computation*, 2013.

[14] E. van den Berg and M Friedlander, "Probing the Pareto frontier for basis pursuit solutions.," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[15] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 497–501, 2000.

[16] D. L Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914, 2009.

[17] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, Y. K. Wang, and T. Wiegand, *High Efficiency Video Coding (HEVC) text specification draft 10*, JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Jan. 2013.

[18] A. Abou-Elailah, F. Dufaux, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion of global and local motion estimation for distributed video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 1, pp. 158–172, 2013.