Panoramic Video Separation with Online Grassmannian Robust Subspace Estimation

Kyle Gilman
University of Michigan
Ann Arbor, MI
kgilman@umich.edu

Laura Balzano
University of Michigan
Ann Arbor, MI
girasole@umich.edu

Abstract

In this work, we propose a new total variation (TV)-regularized robust principal component analysis (RPCA) algorithm for panoramic video data with incremental gradient descent on the Grassmannian. The resulting algorithm has performance competitive with state-of-the-art panoramic RPCA algorithms and can be computed frame-by-frame to separate foreground/background in video with a freely moving camera and heavy sparse noise. We show that our algorithm scales favorably in computation time and memory. Finally we compare foreground detection accuracy and computation time of our method versus several existing methods.

1. Introduction

Video foreground/background separation is of great value to many computer vision algorithms for tasks such as activity or object recognition, target tracking, surveillance [5], or identifying trans-Neptunian objects in highly noisy astronomical data studies [14]. Modern applications such as these include a variety of challenges, ranging from video captured from consumer mobile devices to extremely memory-intensive video datasets. In this work, we focus on the problem of foreground/background separation in panoramic videos, where scalability with respect to memory and computation is a key requirement.

One successful collection of solutions for foreground/background separation in video leverages a problem formulation called robust principal component analysis (RPCA) [28]. RPCA naturally results in both foreground/background separation and foreground recovery. RPCA seeks to recover both a low-rank, incoherent matrix and a sparse corruptions matrix whose sum is the observed data [6]. Such scenarios arise in many practical contexts where video data can be modeled as sparse foreground objects superpositioned on low-rank backgrounds.

Most RPCA work in video separation has thoroughly addressed the case of static video, but modern video, especially from consumer mobile devices, is often captured by a camera undergoing motion, a significant challenge to models that assume a nearly constant background. More recent work on Panoramic RPCA [10] has considered this issue, with the observation that panoramic video can be modeled as snapshots of small pieces of a large static scene with many missing pixels in each frame. However, this work as well as other existing RPCA methods become prohibitively expensive to compute in both time and memory with higher resolution videos and larger ranges of camera motion. The majority of batch algorithms use the singular value decomposition (SVD) to perform RPCA, but the standard SVD requires all of the data to be stored in memory at computation time and can be too slow for many real-time applications [15]. The complexity of computing SVDs or thin SVDs grows quadratically in the number of matrix columns which may become prohibitive with large videos [26].

Finally, few RPCA models are capable of removing sparse noise–or impulse noise–that cannot be distinguished from the foreground, such as in surveillance camera footage with blizzard or rainstorm conditions [27] or in hyperspectral images [25]. Video sequences are also often corrupted with inter-channel correlated impulse noise during the transmission stage, as a result of external effects such as thunderstorms, electric engines, wireless phones etc [24].

We propose a novel RPCA algorithm that can handle panoramic camera motion. Our method is online once we compute the homographic video registration. Our method is also robust to heavy sparse corruptions and can accurately disentangle the noise from foreground objects in the 2017 DAVIS Challenge videos [20]. To the best of our knowledge, our method is the only one that can perform incremental gradient descent on the Grassmann manifold with total-variation (TV) regularization in an online way without using SVDs. We show our method is far more advantageous in computation time and memory than the existing state-of-the-art panoramic RPCA algorithm in [10].

Organization We have a literature review in Section 2, and our model and algorithm are presented in Section 3. Section 4 presents a performance comparison of panoramic RPCA methods in terms of foreground separation and computation time. Finally, Section 5 concludes and discusses opportunities for future work.

2. Previous Work

2.1. RPCA Model in Video Decomposition

Robust PCA algorithms are adept at low-rank-sparse decomposition in difficult problems with high-dimensional and incomplete data. Video background can be thought of as frames with high temporal correlation across the video. Mathematically, in an idealized setting with a completely static background, the matrix of vectorized background video frames can be modeled as a rank-1 matrix $L = b\mathbb{1}_n^T$ where $b \in \mathbb{R}^m$ is the vectorized background frame we wish to recover. While the matrix may not be exactly rank-1 empirically, it is usually very low-rank. We therefore seek to recover the low-rank subspace $U \in \mathbb{R}^{m\times r}$ and the weights $V^T \in \mathbb{R}^{r\times n}$ in a matrix factorization model $L = UV^T$ with $r \ll \min(m, n)$. Any foreground objects in each frame will appear as sparse corruptions in vectorized form added to the background frame. The observed video frame matrix $X \in \mathbb{R}^{m\times n}$ is then $X = L + S$ for some sparse matrix S.

An abundance of research has developed algorithms capable of decomposing video where the background and camera are nearly static. The work in [8] proposed Principal Component Pursuit (PCP)–a classical batch RPCA algorithm that performs singular value shrinkage on the low-rank matrix component. Other works have followed to further constrain the sparse foreground based on *a priori* information. The authors of the Grassmannian Online Subspace Updates with Structured-sparsity (GOSUS) algorithm [29] enforce the foreground objects to belong to superpixels, enhancing the cohesiveness and smoothness of foreground objects. However, the method is expensive to compute, requires a GPU solver, is slow to train, cannot separate the foreground from video corruptions like shotgun noise, and cannot handle missing data.

The authors in [9] proposed to separate background from moving objects using TV-based regularization. It demonstrated TV-based models can effectively distinguish foreground, which should be smooth and spatially cohesive in image space, from sparse corruptions like snow and rain in poor weather conditions. Their method, called TVRPCA, composes the video as a summation of a low-rank component, a sparse TV-regularized foreground, and dense and sparse noise corruptions. TVRPCA is also a batch algorithm that uses the SVD for singular value shrinkage.

2.2. RPCA in Moving Camera Settings

Low-rank plus sparse separation becomes difficult with a freely-moving camera, as background is no longer static and cannot be modeled with a simple low-rank projection. A common solution is to embed a global motion compensation model into the matrix decomposition optimization problem, jointly solving for a transformation matrix containing the global motion of the camera along with the sparse component and low-rank background aligned under the transformation [30]. The work in [22] proposed a fully incremental PCP algorithm for video background modeling under camera jitter, and the work in [23] expounded upon this algorithm to better handle panning and camera motions with newly observed frames. However, in general these methods can only model for either small 2-D camera jitter or slow 2-D camera motion.

A far more challenging problem arises with cameras undergoing rapid perspective motion. Researchers working on the DAVIS Challenge [20] dataset seek to segment foreground objects in a large, diverse set of short, high-resolution RGB videos where the camera undergoes large degrees of motion. In the paper that inspired our work, Moore, Gao, & Nadakuditi [10] showed a classic computer vision technique to re-register the frames into a common reference perspective where RPCA can be applied. Many of the videos in the DAVIS Challenge undergo perspective camera motion limited to eight degrees of freedom. Given correspondence points between frames, a homographic transformation between pairs of frames can be estimated. This clever preprocessing step allows RPCA to decompose the frames into a panoramic background component that spans the entire field of view. Unfortunately, this creates even higher-dimensional data when each transformed frame in the common reference perspective is vectorized. It also creates large numbers of unobserved pixels resulting from the partially overlapping views of the registered frames. This panoramic robust PCA (PRPCA) problem is the perfect storm of extremely high-dimensional and incomplete data.

The work in [10] poses the video decomposition as a type of algorithm similar to TVRPCA. Their formulation is more advanced because it uses the OptShrink algorithm [19, 18] to update the low-rank subspace (which has been shown to be superior to singular value shrinkage algorithms) while separating the foreground from video corruptions like sparse and dense noise.

2.3. Online Grassmannian Subspace Tracking

The GRASTA algorithm by He et al. [15] models the background as a subspace on the Grassmann manifold and develops an iterative algorithm for tracking the low-rank subspace. GRASTA uses the natural ℓ_1-norm cost function for data corrupted by sparse outliers, and operates only one

data vector at a time, making it faster than other state-of-the-art algorithms and amenable to streaming and real-time applications [15]. The algorithm called t-GRASTA [16] extended online video background separation to video with severe camera jitter. GRASTA and t-GRASTA use explicit computations for the Grassmannian geodesics and the gradient of a function defined on the Grassmannian manifold in the work of Edelman, Arias and Smith [11]. We will exploit a very similar Grassmannian update in our proposed methods.

GRASTA operates under the rank-sparsity model which assumes the foreground is sparse and its entries are distributed in a uniformly random pattern. This model works well in most instances, but it could further benefit from *a priori* knowledge that the foreground objects are smooth and spatially cohesive in image space. This is especially complicated if the video is heavily corrupted by sparse noise. The rank-sparsity model is incapable of distinguishing between a sparse signal of interest and sparse corruptions, and foreground recovery is poor. We will show that our proposed algorithm inspired by GRASTA that we call PanGAEA (Panoramic Grassmannian Augmented Estimation Algorithm) not only achieves better foreground segmentations in clean video, but is also adept at handling sparse corruptions.

3. Methods

Our contribution is a novel Grassmannian descent algorithm that can handle missing data in panoramic video, operate orders of magnitude faster than batch methods, and can update its estimates with single streaming vectors in an online setting. We use the same panoramic mosaicking and preprocessing procedure as the authors in [10]. Although the algorithm will be a batch method because of the homography registration, our Grassmannian algorithm still updates it estimates one data vector at a time in an online fashion.

3.1. Registering two frames with a homography [10]

Given a point $p = [x, y, 1]$ in a frame and its corresponding point $\tilde{p} = [\tilde{x}, \tilde{y}, 1]$ in another frame, under the planar surface model, the points are related via the projective transformation

$$\kappa \tilde{p} = H^T p$$

for some arbitrary nonzero scaling constant κ and $H \in \mathbb{R}^{3\times 3}$ with $H_{33} = 1$. The homography matrix H has eight unknown degrees of freedom we can estimate by minimizing

$$\min_h \|Ah\|^2 \quad \text{s.t.} \quad h_9 = 1, \tag{1}$$

where $h = \text{vec}(H)$, and given c correspondences $\{p_i \mapsto \tilde{p}_i\}_{i=1}^c$, $A^T = [A_1^T, \ldots, A_c^T]$ where

$$A_i = \begin{bmatrix} 0 & p_i^T & -\tilde{y}_i p_i^T \\ p_i^T & 0 & -\tilde{x}_i p_i^T \end{bmatrix} \in \mathbb{R}^{2\times 9}$$

To make the least squares problem well-conditioned, a minimum of four correspondence points is required, where each correspondence pair gives two independent linear equations and eight are needed to recover the eight unknown degrees of freedom. The solution to Eq. (1) is the rightmost singular vector of A scaled so the last element is 1. This vector best approximates the vector in the null space of A to minimize the objective in Eq. (1).

The correspondence points are also unknown, and we can use any popular computer vision feature algorithm, e.g. SIFT [17] or SURF [3], to find them and use RANSAC [13] to robustly estimate the H with the best objective value in (1). Usually 10 correspondence points are best in each iteration of RANSAC to ensure a well-conditioned A.

3.1.1 Homographic video registration

The PRPCA problem registers each of the frames $F_1, \ldots, F_n \in \mathbb{R}^{a\times b}$ to the common reference. Like [10], we choose the middle frame $F_{\tilde{k}}$ as the "anchor" frame, or the common reference, where $\tilde{k} = \lfloor n/2 \rfloor$. Each frame is highly correlated with the frame preceding and following it, so we can accurately estimate the homographies $H_k := H_{k\mapsto k+1}$ between frames k and $k+1$. Let $\mathcal{H}_k := \mathcal{H}_{k+1}$ denote the linear transformation between all points in frames k and $k+1$. Each transformed frame $\tilde{F}_k \in \mathbb{R}^{\tilde{a}\times\tilde{b}}$, where $\tilde{a}$ and $\tilde{b}$ are the height and width of the region defined by the union of the registered frame extents, can be computed with respect to the anchor frame by

$$\tilde{F}_k = \begin{cases} (\mathcal{H}_{\tilde{k}-1} \circ \mathcal{H}_{\tilde{k}-2} \circ \ldots \circ \mathcal{H}_k)(F_k) & k < \tilde{k} \\ F_k & k = \tilde{k} \\ (\mathcal{H}_{\tilde{k}}^{-1} \circ \mathcal{H}_{\tilde{k}+1}^{-1} \circ \ldots \circ \mathcal{H}_{k-1}^{-1})(F_k) & k > \tilde{k} \end{cases} \tag{2}$$

We then construct our data matrix $X \in \mathbb{R}^{m\times n}$ for the RPCA problem where $m = \tilde{a}\tilde{b}$ as

$$X = [\text{vec}(\tilde{F}_1) \ldots \text{vec}(\tilde{F}_p)] \tag{3}$$

As an example, we illustrate the homographic frame registration result for "Horsejump-High" from the DAVIS Challenge [20] in Fig. 1. The horse and jockey jump over the gate and gallop towards the red gate seen at the right. Here, each frame has been transformed to a global coordinate system in reference to the video's anchor frame and overlain in reverse sequence.

Following panoramic transformation, the moving camera video data is expressed as a static space-time matrix where each row corresponds to a fixed point in space and where missing matrix entries are unobserved pixels of the

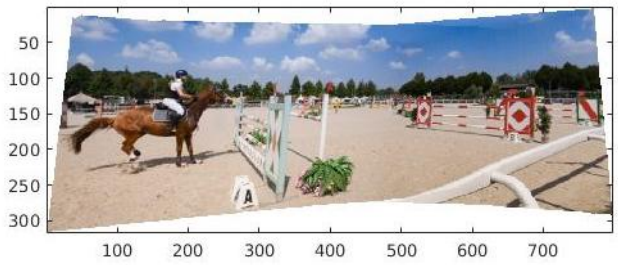


Figure 1. Frames of the video "Horsejump-high" registered in panoramic mosaic.

panoramic mosaic scene [10]. We then perform RPCA on the registered frames matrix with our fast Grassmannian stochastic gradient descent algorithm.

3.2. Model and Algorithms

We propose our algorithm called PanGAEA (Panoramic Grassmannian Augmented Estimation Algorithm) that adheres to rank-sparsity theory for well-posed separability while regularizing the foreground with TV smoothing. This should not only improve segmentation generally but also make the segmentation robust to sparse noise.

PanGAEA is motivated from TVRPCA [9] and GRASTA [15] to obtain fast video separation using iterative Grassmannian descent with TV-regularization of the foreground vector in the objective function. We first model the batch problem using all n frames of the video $X \in \mathbb{R}^{m\times n}$ for vectorized frames of ambient dimension m:

$$\begin{aligned} \min_{U,W,S,E} \quad & \text{TV}(S) + \beta_S\|S\|_1 + \|E\|_1 \\ \text{s.t. } & \mathcal{A}_\Omega(X) = \mathcal{A}_\Omega(UW + S + E) \\ & U^TU = I \end{aligned} \tag{4}$$

Above, $\|Y\|_1 = \sum_{i,j}|Y_{ij}|$ for some $m \times n$ matrix Y. The linear operator $\mathcal{A}_\Omega(\cdot)$ extracts the pixels observed in the panorama mosaic scene on the set $\Omega \subset \{1,\dots,m\} \times \{1,\dots,n\}$. We assume all original frame pixels are observed. $U \in \mathbb{R}^{m\times r}$ is the orthonormal matrix whose columns span the rank-r subspace from which the background frames approximately lie in. We say that U is a point on the Grassmann manifold of subspaces, denoted $U \sim \mathcal{G}(m,r)$, which is the set of all subspaces of dimension r in $\mathbb{R}^m$. The Grassmannian is a compact Riemannian manifold, and its geodesics can be explicitly computed [1].

The matrix $W \in \mathbb{R}^{r\times n}$ is the weights matrix. The matrix $S \in \mathbb{R}^{m\times n}$ captures the sparse foreground objects, and $E \in \mathbb{R}^{m\times n}$ models sparse corruptive noise. The hyperparameter β_S balances the smoothness of the foreground signal with the sparsity of the noise. Here $\text{TV}(S) = \|WCS\|_1$, where C is the block-circulant first-order differences matrix formed by

$$C \in \mathbb{R}^{2m\times m} = \begin{bmatrix} I_N \otimes D_M \\ D_N \otimes I_M \end{bmatrix} \tag{5}$$

Here, D_M is the $M \times M$ first-order differences matrix. Recall that M, N are the dimensions of the registered frames in the common reference. W is the square, diagonal matrix of weights whose diagonal d has zeros on the indices corresponding to the circulant boundaries and ones otherwise.

We rewrite the problem in Eq. (4) in terms of each frame, or column, at time instance t where each column is observed on the set $\Omega_t \subset \{1,\dots,m\}$ for $n = T$ columns:

$$\begin{aligned} \min_{s_t,e_t,U,w_t} \sum_{t=1}^{T} & \|WC\chi_{\Omega_t}(s_{\Omega_t})\|_1 + \beta_S\|s_{\Omega_t}\|_1 + \|e_{\Omega_t}\|_1 \\ \text{s.t. } & x_{\Omega_t} = U_{\Omega_t}w_t + s_{\Omega_t} + e_{\Omega_t} \\ & U^TU = I \end{aligned}$$

Here, U_{Ω_t} denotes the $|\Omega_t| \times r$ submatrix formed by extracting the rows indexed observed on Ω_t, and similarly for $x_{\Omega_t}, s_{\Omega_t}$, and e_{Ω_t}. We also denote the linear operator $\chi_{\Omega_t}(\cdot)$ which zero pads a vector argument of length $|\Omega_t|$ to dimension m on the indices in the complement of Ω_t. We then minimize this objective function with a stochastic gradient descent procedure for each time instance t:

$$\begin{aligned} \min_{s_t,e_t,U,w_t} & \|WC\chi_{\Omega_t}(s_{\Omega_t})\|_1 + \beta_S\|s_{\Omega_t}\|_1 + \|e_{\Omega_t}\|_1 \\ \text{s.t. } & x_{\Omega_t} = U_{\Omega_t}w_t + s_{\Omega_t} + e_{\Omega_t} \\ & U^TU = I \end{aligned} \tag{6}$$

Note that we have enforced the foreground in s_t to be TV-smooth in image space but also sparse. While Moore, Gao, & Nadakuditi [10] argue the sparsity constraint is over-restrictive, we found it was actually necessary in our model to be accountable to rank-sparsity theory and achieve any kind of acceptable separation. The two regularizers work in concert to separate foreground objects that are recoverable in the RPCA sense, but also conform to our heuristical understanding of how foreground objects should appear and behave in video.

To make the terms in the objective function of (6) separable in each variable and compatible with the ADMM model, we can rewrite the problem using linear constraints as

$$\begin{aligned} \min_{z_t,s_t,\xi_t,e_t,U,w_t} & \|Wz_t\|_1 + \beta_S\|\xi_{\Omega_t}\|_1 + \|e_{\Omega_t}\|_1 \\ \text{s.t. } & z_t = C\chi_{\Omega_t}(s_{\Omega_t}) \\ & x_{\Omega_t} = U_{\Omega_t}w_t + \xi_{\Omega_t} + e_{\Omega_t} \\ & \xi_{\Omega_t} = s_{\Omega_t} \\ & U^TU = I \end{aligned} \tag{7}$$

with $z_t \in \mathbb{R}^{2m}, x_t, s_t, \xi_t, e_t \in \mathbb{R}^m$, and $w_t \in \mathbb{R}^r$. The problem is nonconvex because of the coupling between

U and w_t and because U lies on the Grassmann manifold. First, we form the augmented Lagrangian and optimize by block-coordinate descent. We alternate by holding U fixed and solving for the variables z_t, s_t, ξ_t, e_t, and w_t with ADMM; then, holding all variables fixed except for U, our algorithm takes a geodesic step along the manifold in the direction of the negative gradient of the augmented Lagrangian.

From Eq. (7), we form the augmented Lagrangian with the dual variables of appropriate dimensions $\lambda_{1t}, \lambda_{2t}$, and λ_{3t} at time t. After completing the square and ignoring constant terms, the augmented Lagrangian becomes

$$\begin{aligned}&\mathcal{L}(U, z_t, s_t, \xi_t, e_t, w_t, \lambda_{1,2,3_t}) = \\ &\|Wz_t\|_1 + \beta_S\|\xi_{\Omega_t}\|_1 + \|e_{\Omega_t}\|_1 \\ &+ \frac{\rho_1}{2}\|C\chi_{\Omega_t}(s_{\Omega_t}) - z_t + \frac{\lambda_{1_t}}{\rho_1}\|_2^2 + \frac{\rho_2}{2}\|\xi_{\Omega_t} - s_{\Omega_t} + \frac{\lambda_{2_t}}{\rho_2}\|_2^2 \\ &+ \frac{\rho_3}{2}\|U_{\Omega_t}w_t + \xi_{\Omega_t} + e_{\Omega_t} - x_{\Omega_t} + \frac{\lambda_{3_t}}{\rho_3}\|_2^2 \end{aligned} \tag{8}$$

The smoothing penalties ρ are user-defined, and we will assume all three penalties are equal to 1.8, which works well in practice.

3.2.1 Updates of the principal weights, sparse vectors, surrogate variables, and dual variables with ADMM

Given an estimate of the subspace $\hat{U}$, the problem in (7) is a constrained convex optimization problem with strong duality [7]. Given the partial observation x_{Ω_t} and the observed entries indices Ω_t, the optimal $(z_t^*, s_t^*, \xi_t^*, e_t^*, w_t^*, \lambda_{1,2,3_t}^*)$ in Eq. (6) can be found by minimizing the augmented Lagrangian in Eq. (8) with respect to these variables:

$$\begin{aligned}&(z_t^*, s_t^*, \xi_t^*, e_t^*, w_t^*, \lambda_{1,2,3_t}^*) = \\ &\qquad \underset{z_t, s_t, \xi_t, e_t, w_t, \lambda_{1,2,3_t}}{\operatorname{argmin}} \mathcal{L}(\hat{U}, z_t, s_t, \xi_t, e_t, w_t, \lambda_{1,2,3_t}) \end{aligned} \tag{9}$$

We efficiently update each variable in Eq. (9) with ADMM in an alternating fashion, yielding the updates given in Steps (5) and (6) of Algorithm 2.

We note that $z = \mathrm{S}_\beta(y) = \mathrm{sign}(y) \odot \max(|y| - \beta, 0)$ in Algorithm 2 is the elementwise soft-thresholding operator of argument vector $y \in \mathbb{R}^d$ for some positive constant β that yields the vector $z \in \mathbb{R}^d$ [4, 12].

The matrix-vector product $C\chi_{\Omega_t}(s_{\Omega_t})$ in Algorithm 2 can be efficiently computed by taking the first order differences of only the observed pixels in the frame. We also assume above that the matrix $U_{\Omega_t}^T U_{\Omega_t}$ is always invertible, which has been shown to be guaranteed if $|\Omega_t|$ is large enough [2].

The derived update of s_{Ω_t} originally involves the inverse of a very large matrix $(I + C^T C) \in \mathbb{R}^{m \times m}$, assuming all ρ's are equal. Computing the inverse is prohibitive for our applications in video where m is usually very large. Fortunately, the matrix has block-circulant structure, and it can be shown there is a fast and efficient update that does not involve difficult matrix inverses [21]:

$$s_{\Omega_t}^{k+1} = \mathcal{A}_{\Omega_t}\left(\mathcal{F}_2^{-1}\left(\frac{\mathcal{F}_2(\rho_1 C^T(z_t^k - \lambda_{1_t}^k/\rho_1) + \rho_3 r_t^{k+1})}{1 + \rho_1 \mathcal{F}_2(c)}\right)\right), \tag{10}$$

where $\mathcal{F}_2 : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ (again, M, N are the dimensions of the registered frames in the common reference, and where the ambient dimension of the video data becomes $m = MN$) denotes the operator that reshapes its input into an $M \times N$ matrix, computes the 2D fast Fourier transform, and vectorizes the result. The operator $\mathcal{A}_{\Omega_t}$ extracts the observed vector indices. The vector $c = C^T C[:, 1]$ denotes the first column of the matrix $C^T C$. This denominator is a constant and can be precomputed. The total computational complexity of one update is dominated by the Fourier updates at $O(m \log(m))$. ADMM empirically converges very quickly, usually within a few tens of iterations. In our algorithm, we found 50 iterations to work well for high-dimensional video to where the Karush-Kuhn Tucker (KKT) conditions are met within precision of some small ϵ.

3.2.2 Grassmannian geodesic gradient step

The partial derivative of the augmented Lagrangian with respect to the components of U given estimates of the other variables from ADMM is

$$\frac{\partial \mathcal{L}}{\partial U} = \chi_{\Omega_t}(\lambda_{3_t}^* + \rho_3(U_{\Omega_t} w_t^* + \xi_{\Omega_t}^* + e_{\Omega_t}^* - x_{\Omega_t}) w_t^{*T}) \tag{11}$$

From the work of [11], the gradient of the augmented Lagrangian on the Grassmannian is

$$\nabla \mathcal{L} = (I - UU^T)\frac{\partial \mathcal{L}}{\partial U} = \Gamma w_t^{*T}, \tag{12}$$

where

$$\begin{aligned}\Gamma_1 &= \lambda_{3_t}^* + \rho_3(U_{\Omega_t} w_t^* + \xi_{\Omega_t}^* + e_{\Omega_t}^* - x_{\Omega_t}) \\ \Gamma_2 &= U_{\Omega_t}^T \Gamma_1 \\ \Gamma &= \chi_{\Omega_t}\Gamma_1 - U\Gamma_2 \end{aligned} \tag{13}$$

It is easily verified that $\nabla \mathcal{L}$ is a rank-1 matrix with a trivial SVD whose only nonzero singular value is $\sigma = \|\Gamma\|\|w_t^*\|$ with left and right singular vectors $\Gamma/\|\Gamma\|$ and

$w_t^*/\|w_t^*\|$ respectively. From [11], the gradient step on the Grassmann manifold for some positive length η in the direction of $-\nabla L$ is

$$U_{t+1}(\eta) = U_t + \left((\cos(\eta\sigma) - 1)\frac{U_t w_t^*}{\|w_t^*\|} - \sin(\eta\sigma)\frac{\Gamma}{\|\Gamma\|} \right) \frac{w_t^{*T}}{\|w_t^*\|} \tag{14}$$

PanGAEA is fully summarized in Algorithm 1.

3.2.3 Complexity Analysis

The total cost of PanGAEA is $O(|\Omega|r^3 + Km\log(m) + K|\Omega|r + mr^2)$. Algorithm 1 costs $O(|\Omega|r^3 + |\Omega|r + mr^2)$ flops like GRASTA. The w_t and soft-thresholding updates in the ADMM solver in Algorithm 2 are simple linear algebraic computations and require at most $O(K|\Omega|r)$ flops. A notable advantage of PanGAEA is its savings in these updates from operating on dimensions $|\Omega|$, the number of observed pixels, compared to the full ambient dimension m. The update for s_t is the most costly in the ADMM solver, requiring $O(Km\log(m))$. PanGAEA also avoids computing SVDs, a cost which grows quadratically in the number of video frames. PanGAEA relies on simple, efficient linear algebra operations with linear complexity in the data dimensions, is constant in memory use, and is numerically stable by maintaining orthonormality on the Grassmann manifold.

Algorithm 1 Algorithm for PanGAEA

Input: A $m \times r$ orthonormal matrix U_0. A sequence of corrupted vectors x_t, each vector observed in entries $\Omega_t \subset \{1, ..., m\}$. Step size $\eta > 0$. Regularizer $\beta_S > 0$. Augmented Lagrangian penalty ρ.

Output: U and $w_t, s_{\Omega_t}, e_{\Omega_t}$ at time t.

Form $C = \begin{bmatrix} I_N \otimes D_M & D_N \otimes I_M \end{bmatrix}^T$
Compute $c = C^T C[:, 1]$
Compute $\phi = 1 + \rho\mathcal{F}_2(c)$
for $t = 0$ to T **do**
  Extract U_{Ω_t} from U: $U_{\Omega_t} = \mathcal{A}_{\Omega_t}(U)$
  Estimate $w_t^*, s_{\Omega_t}^*, \xi_{\Omega_t}^*, e_{\Omega_t}^*, \lambda_{3t}^*$ via Algorithm 2.
  Compute Γ by Eq. (13).
  Update the subspace with Eq. (14).
end for
return U_{t+1} and $w_t, s_{\Omega_t}, e_{\Omega_t}, \quad \forall t = 0, \ldots, T$

4. Experiments & Evaluation

Next we show experimental results of our algorithms on three RGB videos from the 2017 DAVIS Challenge [20] compared to RPCA with OptShrink [8, 10], GRASTA [15], and PRPCA [10]. The DAVIS Challenge provides ground-truth binary masks of the foreground objects in each video frame for 60 training videos. We show the results of each algorithm on three of those videos, "Tennis," "Paragliding,"

Algorithm 2 ADMM Solver for PanGAEA

Input: A $|\Omega_t| \times r$ orthonormal matrix U_{Ω_t}. A sequence of corrupted vectors x_t, each vector observed in entries $\Omega_t \subset \{1, ..., m\}$. Augmented Lagrangian penalty ρ. Parameter μ_0. $\phi = 1 + \rho\mathcal{F}_2(c)$. Tolerance ϵ.

Output: $w_t, s_{\Omega_t}, e_{\Omega_t}$ at time t.

Precompute $P = (U_{\Omega_t}^T U_{\Omega_t})^{-1} {U_{\Omega_t}}^T$
$\mu^k = \mu_0$
for $k = 0$ to K or until convergence **do**
  Update principal weights:
  $w^{k+1} = P(x_{\Omega_t} - \xi_{\Omega_t}^k - e_{\Omega_t}^k - \lambda_3^k/\mu^k)$
  Update foreground sparse vector:
  $r^{k+1} = \chi_{\Omega_t}(\xi_{\Omega_t}^k + \lambda_2^k/\mu^k)$
  $s_{\Omega_t}^{k+1} = \mathcal{A}_{\Omega_t}(\mathcal{F}_2^{-1}(\frac{\mathcal{F}_2(\mu^k C^T(z^k - \lambda_1^k/\mu^k) + \mu^k r^{k+1})}{\phi}))$
  Update soft-thresholded variables:
  $h^k = x_{\Omega_t} - U_{\Omega_t} w^{k+1} - \lambda_3^k/\mu^k$
  $\xi_{\Omega_t}^{k+1} = \frac{1}{2}\mathrm{S}_{\beta_S/\mu^k}(h^k - e_{\Omega_t}^k + s_{\Omega_t}^{k+1} - \lambda_2^k/\mu^k)$
  $e_{\Omega_t}^{k+1} = \mathrm{S}_{1/\mu^k}(h^k - \xi_{\Omega_t}^{k+1})$
  $z_t^{k+1} = \mathrm{S}_{d./\mu^k}(C\chi_{\Omega_t}(s_{\Omega_t}^{k+1}) + \lambda_1^k/\mu^k)$
  Update the residuals of the linear equality constraints:
  $y_1^k = C\chi_{\Omega_t}(s_{\Omega_t}^{k+1}) - z_t^{k+1}$
  $y_2^k = U_{\Omega_t} w^{k+1} + \xi_{\Omega_t}^{k+1} + e_{\Omega_t}^{k+1} - x_{\Omega_t}$
  $y_3^k = \xi_{\Omega_t}^{k+1} - s_{\Omega_t}^{k+1}$
  Update the dual variables
  $\lambda_1^{k+1} = \lambda_1^k + \mu^k y_1^k$
  $\lambda_2^{k+1} = \lambda_2^k + \mu^k y_2^k$
  $\lambda_3^{k+1} = \lambda_3^k + \mu^k y_3^k$
  Update the ADMM penalty $\mu^{k+1} = \rho\mu^k$
  if $\max\{\|y_1\|_2, \|y_2\|_2, \|y_3\|_2\} \leq \epsilon$ **then**
    Converge and break the loop
  end if
end for
return $w_t^* = w^{k+1}, s_{\Omega_t}^* = s_{\Omega_t}^{k+1}, e_{\Omega_t}^* = e_{\Omega_t}^{k+1}$

and "Horsejump-High," with and without sparse additive noise. We compare performance with receiver operating curves (ROC), area under the curve (AUC), computation time, and mean peak signal-to-noise ratios (PSNR) when sparse noise is added. We also show frames from the recovered videos.

Both Grassmannian algorithms (PanGAEA and GRASTA) are specified to learn a rank-1 subspace. We found that $\beta_S = 0.5$ worked well in PanGAEA. We run PanGAEA for 7 epochs, randomly shuffling the frames and diminishing the step size each epoch. We run GRASTA for 10 epochs with diminishing step size and random frame order. PRPCA is computed with the code provided and hyperparameters suggested by the authors in [10]. We test each algorithm with "clean" video–i.e. video with no sparse corruptions–and noisy data with 20% shotgun

noise, a challenging scenario where most RPCA algorithms should perform poorly to separate foreground objects from the sparse noise.

Table 1 shows that PanGAEA is competitive on area under the curve (AUC). Table 2 shows similar performance on PSNR. Most importantly, Chart 1 shows that PanGAEA is significantly faster than PRPCA while still achieving competitive performance. It is still slower than GRASTA and RPCA, but its separation performance overall is significantly more accurate.

Fig. 2 shows two frames of PanGAEA separation results on "Tennis" from the DAVIS Challenge [20] which has 69 frames, each corrupted with 20% shotgun noise. This is a challenging video with a wide and fast camera pan. To save computation time during testing, we down-sampled the resolution by one-fourth to give a resolution of 120×214. Computing "Tennis" cost PanGAEA 272.65 seconds and PRPCA 2108.40 seconds running both algorithms 150 iterations on a 2.6 GHz Intel Core i7 MacBook Pro. The average time of PanGAEA to cycle over the entire video once was 38.69, seconds whereas the average for PRPCA was 14.55 seconds. However, PRPCA's proximal gradient descent method requires many iterations over the data to obtain acceptable separation results compared to our Grassmannian descent approach which requires far fewer cycles. As the number of video frames grows, we expect this advantage over PRPCA to improve as PRPCA's SVD computations take more time.

One can further improve the computational performance of PanGAEA by subsampling the panoramic-registered frames to rapidly learn the panoramic background spanning the field of view, since our method can robustly estimate the low-rank subspace from partial information in only a few epochs. Then, the sparse components can be estimated by running PanGAEA with full sampling for one pass over the data. We were able to get comparable performance results subsampling only 20% of the pixels in the registered frames for 6 epochs and fully sampling the 7th. For denoising and separating "Tennis", PanGAEA achieved 0.9413 AUC and 20.90 dB PNSR in 193.40 seconds.

Similar results are shown for the videos "Paragliding" and "Horsejump-High" in Fig. 2. The paraglider is quite small and should be difficult to recover in heavy noise. Nevertheless, the TV-regularized algorithms are capable of denoising the separation while their non-augmented counterparts fail. Even with larger foreground objects like the horse and jockey, which begin to encroach on rank-sparsity assumptions, PanGAEA is able to distinguish each component with minimal separability issues.

Our separation results demonstrate PanGAEA's ability to improve segmentation in noiseless regimes and successfully recover foreground in the presence of heavy sparse corruptions using far less total computation time and memory than

Sequence	PanGAEA	PRPCA	GRASTA	RPCA
Fig. 3a	**0.9768**	0.9649	0.9694	0.8488
Fig. 3d	**0.9698**	0.9532	0.8602	0.7621
Fig. 3b	0.9767	0.9793	**0.9870**	0.9618
Fig. 3e	**0.9817**	0.9771	0.9221	0.8824
Fig. 3c	**0.9597**	0.9432	0.9556	0.7755
Fig. 3f	**0.9561**	0.9476	0.8484	0.6608

Table 1. Area Under Curve (AUC) of each algorithm.

Sequence	PanGAEA	PRPCA	GRASTA	RPCA
"Tennis"	21.72	**22.50**	17.79	17.86
"Paragliding"	25.96	**26.33**	18.58	18.63
"Horsejump-High"	21.29	**22.92**	17.13	17.23

Table 2. Mean PSNR (dB) of each algorithm's denoised frames.

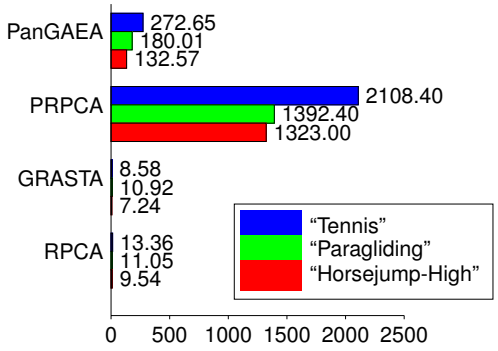


Chart 1. Total computation time (seconds) for each video.

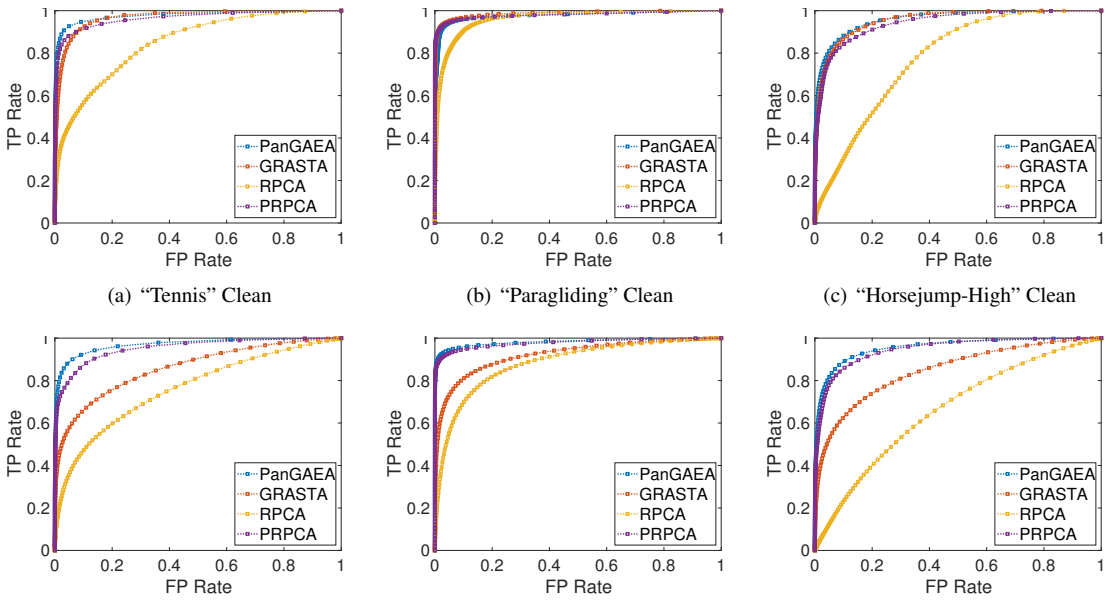
PRPCA. Table 1 shows our method achieves higher area under the ROC than competitor methods. However, it is worth noting the batch methods' denoised frames obtain slightly better peak signal-to-noise (PSNR) with respect to the original frame than the Grassmannian algorithms, as seen in Table 2. In particular, our method experiences more leakage of the sparse foreground component into the sparse noise component than PRPCA. However, it does not seem to significantly affect the foreground detection capability.

5. Conclusions and Future Work

In this paper we have presented a novel TV-regularized RPCA algorithm that can estimate subspaces on the Grassmann manifold and perform foreground-background separation in panoramic video. Our algorithm achieves competitive performance with PRPCA in far less computational time by performing first-order gradient descent on the Grassmann manifold. Our optimization method is online by nature and can process data frame-by-frame.

Our future work aims to make the panoramic RPCA problem truly online where the frames do not need to be pre-registered and the geometric transformation between frames is estimated in the objective function on the fly.

Figure 2. PanGAEA separation results on DAVIS Challenge 2017 videos [20]. From top to bottom: Original frames, Corrupted frames with 20% shotgun noise (Observed), Recovered Background, Recovered Sparse Corruptions, Recovered Foreground. Left to right: "Tennis","Paragliding", "Horsejump-High".



(a) "Tennis" Clean (b) "Paragliding" Clean (c) "Horsejump-High" Clean

(d) "Tennis" Corrupted (e) "Paragliding" Corrupted (f) "Horsejump-High" Corrupted

Figure 3. ROC curves for DAVIS Challenge videos. PanGAEA achieves the largest area under its curve in both clean and noisy video and outperforms its competitors.

We also seek an adaptive step size like the one proposed for GRASTA in [15] so that PanGAEA may track time-dynamical subspaces. Combining these goals, we also intend to study developments that can perform separation even with fast and wide camera pans. Also of key interest is making our algorithm robust to dense noise and imputing missing values of the sparse components, since our method can only complete the low-rank background when given partial information.

Acknowledgements: This work was supported by AFOSR YIP award FA9550-19-1-0026, ARO YIP award W911NF1910027, and DARPA grant 16-43-D3M-FP-037. The authors also thank Jeff Fessler for his helpful feedback.

References

[1] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 704–711, Sep. 2010. 4

[2] L. Balzano, B. Recht, and R. Nowak. High-dimensional matched subspace detection when data are missing. In *2010 IEEE International Symposium on Information Theory*, pages 1638–1642, June 2010. 5

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 3

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 693–696, 2009. 5

[5] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo. On the applications of robust PCA in image and video processing. *Proceedings of the IEEE*, 106, 07 2018. 1

[6] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23:1 – 71, 2017. 1

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. 5

[8] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. 2, 6

[9] X. Cao, L. Yang, and X. Guo. Total variation regularized RPCA for irregularly moving object detection under dynamic background. *IEEE transactions on cybernetics*, 46, 04 2015. 2, 4

[10] B. E. Moore, C. Gao, and R. Rao Nadakuditi. Panoramic robust PCA for foreground-background separation on noisy, free-motion camera video. *IEEE Transactions on Computational Imaging*, PP, 12 2017. 1, 2, 3, 4, 6

[11] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, Apr. 1999. 3, 5, 6

[12] J. Fessler. Eecs 551 lecture notes: Chapter 6: Low-rank approximation, 03 2017. 5

[13] R. Fischler and M. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM*, 24:619–638, 01 1981. 3

[14] C. A. G. Gonzalez, O. Absil, P.-A. Absil, M. V. Droogenbroeck, D. Mawet, and J. Surdej. Low-rank plus sparse decomposition for exoplanet detection in direct-imaging ADI sequences the LLSG algorithm. *A&A*, 589, 2016. 1

[15] J. He, L. Balzano, and A. Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1575, June 2012. 1, 2, 3, 4, 6, 8

[16] Jun He, Dejiao Zhang, L. Balzano, and Tao Tao. Iterative online subspace learning for robust image alignment. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8, April 2013. 3

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004. 3

[18] B. E. Moore, R. R. Nadakuditi, and J. A. Fessler. Improved robust PCA using low-rank denoising with optimal singular value shrinkage. In *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pages 13–16, June 2014. 2

[19] R. R. Nadakuditi. Optshrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage. *IEEE Transactions on Information Theory*, 60(5):3002–3018, May 2014. 2

[20] J. Pont-Tuset, S. Caelles, F. Perazzi, A. Montes, K.-K. Maninis, Y. Chen, and L. Van Gool. The 2018 DAVIS challenge on video object segmentation. 03 2018. 1, 2, 3, 6, 7, 8

[21] D. Ren, H. Zhang, D. Zhang, and W. Zuo. Fast total-variation based image restoration based on derivative alternated direction optimization methods. *Neurocomput.*, 170(C):201–212, Dec. 2015. 5

[22] P. Rodríguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision*, 55:1–18, 2015. 2

[23] P. Rodriguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision*, 55(1):1–18, May 2016. 2

[24] P. Rodrguez and B. Wohlberg. Video background modeling under impulse noise. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1041–1045, Oct 2014. 1

[25] S. Tariyal, H. K. Aggarwal, and A. Majumdar. Removing sparse noise from hyperspectral images with sparse and low-rank penalties. *J. Electronic Imaging*, 25:020501, 2016. 1

[26] V. Vasudevan and M. Ramakrishna. A hierarchical singular value decomposition algorithm for low rank matrices. *ArXiv*, abs/1710.02812, 2017. 1

[27] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 393–400, June 2014. 1

[28] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009. 1

[29] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh. Gosus: Grassmannian online subspace updates with structured-sparsity. In *2013 IEEE International Conference on Computer Vision*, pages 3376–3383, Dec 2013. 2

[30] M. Yazdi and T. Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:157 – 177, 2018. 2