Sequential Decomposition of 2D Apparent Motion Fields Based on Low-Rank and Sparse Approximation

Shun Ogawa*, Hiroki Kuhara* and Tomoya Sakai*
* Nagasaki University, Nagasaki, Japan
E-mail: tsakai@cis.nagasaki-u.ac.jp Tel/Fax: +81-95-819-2583

Abstract—**Estimation of camera egomotion and detection/tracking of moving objects in video captured by moving camera are challenging tasks in many applications such as autonomous vehicle control, medical image analysis, video surveillance, compression, stabilization, etc. Camera egomotion and object motion in a scene both induce temporal variation of image intensity, which is represented as a vector field of the so-called optical flow. An egomotional optical-flow field at any moment is composed of a few common linear basis vector fields related to camera's translation and rotation, while the moving objects superpose spatio-temporally local and sparse optical flows. We present an online algorithm of stable principal component pursuit that decomposes frame by frame the apparent motion fields into linearly dependent optical-flow fields and sparse optical flows for a given sequence of images. We experimentally show that the former captures the egomotion, and the latter detects the moving objects.**

I. Introduction

This paper presents a sparsity-aware video analyzing technique for perceiving motion of objects through a moving camera. Apparent motion in a video sequence is usually induced by either or both of the camera movement (a.k.a. egomotion) and motion of objects in a scene. Unmixing apparent motion into those by egomotion and object motion enables us to estimate the camera's translation and rotation as well as to detect and track the objects. These image processing tasks appear in many applications: vehicle control, visual odometry, medical image analysis, surveillance, video compression, stabilization, and so forth.

The vector field of instantaneous apparent motion computed from successive images is known as optical flow, of which computation schemes have been developed mostly on the basis of mathematical optimization [1], [2], [3], [4], [5]. Unlike the videos taken from fixed cameras, however, it is problematical to perceive object motion in a dynamic scene with camera egomotion. To estimate the optical flow of the scene background caused by the egomotion, one has to eliminate foreground optical flow superposed on the background motion, while the detection of the unknown foreground objects requires object tracking or accurate estimation of the dynamic background. This problem therefore needs alternative or simultaneous estimation of the foreground motion of objects and background motion. Background subtraction [6], [7] is a popular approach to the extraction of foreground objects, but it is hard to apply it to dynamic background. A precise model of camera geometry would be useful for the estimation of background as well as the egomotion [8], [9], [10], [11], [12]. Foreground objects, often detected as outliers of the background, are tracked via prediction and matching [13], [14].

Some literature on a sequence of optical-flow fields [15], [16], [17], [18] suggested that a matrix of two-dimensional optical-flow sequence of a rigid body object has a low-rank structure. We have exploited this low-rank structure using complex-number representation of the optical flow to separate sparse foreground motion from the low-rank background via low-rank and sparse approximation [19]. Decomposition of a matrix into a pair of low-rank and sparse matrices is known as robust principal component analysis (RPCA) [20], and it can be solved by convex optimization techniques called principal component pursuit (PCP) [21], [22], [23], [24], [25], [26], [7]. It would be preferable to perform this separation frame by frame for video analysis. There have also been proposed online algorithms for RPCA [27], [28]. These online algorithms are less time and memory consuming, but their performance has been examined mostly in the application to background subtraction in static scenes. Because they treat only the low-rank and sparse components, video noise and approximation error entirely remain in one of the low-rank and sparse components.

In this paper, we design an online algorithm of stable PCP (SPCP) that computes the low-rank and sparse components of a high-dimensional data stream in a sequential fashion. One of the major differences from the existing online algorithms is simultaneous estimation of the two components while allowing a specific amount of approximation error. After briefly introducing optical flow and low-rank nature of its temporal sequences, we apply our algorithm to the sequential decomposition of two-dimensional optical-flow fields to experimentally confirm its advantages.

II. Online Algorithm of Low-Rank and Sparse Approximation

A. Problem Formulation

Approximation of a matrix $\mathbf{C} \in \mathbb{C}^{M \times N}$ with a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ can be posed as a convex optimization problem known as the stable principal component pursuit (SPCP) [29], [7]:

$$\underset{\mathbf{L},\mathbf{S}}{\text{Minimize}}\ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \text{ subject to } \mathbf{C} - (\mathbf{L}+\mathbf{S}) \in \mathcal{E}. \quad (1)$$

Here, the nuclear norm (a.k.a. the trace norm or the Schatten 1-norm) of $\mathbf{L}$, $\|\mathbf{L}\|_*$, is defined as the sum of the singular values of $\mathbf{L}$. The matrix ℓ_1 norm of $\mathbf{S}$, $\|\mathbf{S}\|_1$, is defined as the sum of the absolute values of the matrix entries. The parameter $\lambda > 0$ balances these norms in the minimization. This minimization promotes sparsity of the singular values of $\mathbf{L}$ and entries of $\mathbf{S}$. $\mathcal{E} \subset \mathbb{C}^{M\times N}$ can be any nonempty convex set of possible approximation error matrices. In this paper, we regard it as $\mathcal{E} = \mathcal{E}_\varepsilon := \{\mathbf{E} \mid \mathbf{E} \in \mathbb{C}^{M\times N}, \|\mathbf{E}\|_F \le \varepsilon\}$ where $\|\cdot\|_F$ denotes the Frobenius norm.

Consider now an online algorithm for solving Eq. (1). Suppose that we have already computed the j-th columns ($j = 1,\dots,n-1 < N$) of $\mathbf{L}$ and $\mathbf{S}$ respectively as $\mathbf{l}^{(j)}$ and $\mathbf{s}^{(j)}$ from the columns $\mathbf{c}^{(j)}$ of $\mathbf{C}$. To compute $\mathbf{l}^{(n)}$ and $\mathbf{s}^{(n)}$ from a given vector $\mathbf{c}^{(n)}$, we make three assumptions on $\mathbf{L}$ and $\mathbf{S}$: first that the subspace $\mathcal{S}_{n-1} := \mathrm{span}\{\mathbf{l}^{(1)},\dots,\mathbf{l}^{(n-1)}\}$ well approximates the one spanned by all the columns of $\mathbf{L}$, second that $\mathbf{s}^{(n)}$ is sparse, and third that the approximation error, $\mathbf{c}^{(n)} - (\mathbf{l}^{(n)} + \mathbf{s}^{(n)})$, remains bounded (typically in ℓ_2 norm). Let $\mathbf{U}^{(n-1)} \in \mathbb{C}^{M\times r}$ be a matrix of orthonormal basis vectors for $\mathcal{S}_{n-1}$, i.e., ${\mathbf{U}^{(n-1)}}^\top \mathbf{U}^{(n-1)} = \mathbf{I}$, and $\mathbf{P}^{(n-1)} := \mathbf{U}^{(n-1)}{\mathbf{U}^{(n-1)}}^\top$ is the projection matrix onto the subspace $\mathcal{S}_{n-1}$ [1]. Regarding $\mathbf{U}^{(n-1)}$ as an accurate estimate of $\mathbf{U}^{(n)}$, we have $\|\,\mathbf{l} - \mathbf{P}^{(n-1)}\mathbf{l}\,\|_2 \ll \|\mathbf{l}\|_2$, $\forall \mathbf{l} \in \mathcal{S}_n$. On the basis of the above, we formulate the problem of finding $\mathbf{l}^{(n)}$ and $\mathbf{s}^{(n)}$ for given $\mathbf{c}^{(n)}$ and $\mathbf{U}^{(n-1)}$ as

$$\underset{(\mathbf{l},\mathbf{s})}{\text{Minimize}}\ \frac{1}{2}\|(\mathbf{I} - \mathbf{P}^{(n-1)})\mathbf{l}\|_2^2 + \lambda_s\|\mathbf{s}\|_1 + h(\mathbf{l}+\mathbf{s}|\mathbf{c}^{(n)}) \quad (2)$$

where $\mathbf{P}^{(n-1)} := \mathbf{U}^{(n-1)}{\mathbf{U}^{(n-1)}}^\top$ and

$$h(\mathbf{v}|\mathbf{c}^{(n)}) = \begin{cases} 0 & \text{if } \|\mathbf{v} - \mathbf{c}^{(n)}\|_2 \le \delta \\ \infty & \text{otherwise.} \end{cases} \quad (3)$$

A new basis matrix $\mathbf{U}^{(n)}$ also needs to be estimated so as to span the new subspace $\mathcal{S}_n$.

B. Algorithm Derivation

Letting $\mathbf{x} = [\mathbf{l}^\top, \mathbf{s}^\top]^\top$ and $\mathbf{z} = [\mathbf{z}_{l+s}^\top, \mathbf{z}_l^\top, \mathbf{z}_s^\top]^\top$, we can rewrite Eq (2) as

$$\underset{(\mathbf{x},\mathbf{z})}{\text{Minimize}}\ h(\mathbf{z}_{l+s}|\mathbf{c}^{(n)}) + \frac{1}{2}\|(\mathbf{I} - \mathbf{P}^{(n-1)})\mathbf{z}_l\|_2^2 + g_s(\mathbf{z}_s)$$

$$\text{subject to } \mathbf{z} = \mathbf{A}\mathbf{x} \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix}. \quad (4)$$

Here, the regularizer $g_s(\mathbf{z}_s)$ is $\lambda_s\|\mathbf{z}_s\|_1$ or any sparsity-inducing norm whose proximity operator is efficiently computable.

An efficient algorithm for solving Eq. (4) can be derived from the so-called alternating direction method of multipliers (ADMM) [30], [31], which iterates updating steps for $\mathbf{x}$, $\mathbf{z}$, and a dual variable $\mathbf{y}$. We show the detailed steps in Algorithm 1. To obtain $\mathbf{l}^{(n)}$ and $\mathbf{s}^{(n)}$, use Algorithm 1 as

$$(\mathbf{l}^{(n)}, \mathbf{s}^{(n)}) \leftarrow \textsc{ColSPCP}(\mathbf{c}^{(n)}, \mathbf{U}^{(n-1)}, \lambda_s, \delta, \rho). \quad (5)$$

In Algorithm 1, Steps 3 and 4 calculate $\mathbf{x}$ as the least squares solution $\arg\min_{\mathbf{x}} \|\mathbf{y} + \mathbf{A}\mathbf{x} - \mathbf{z}\|_2^2 = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top(\mathbf{z} - \mathbf{y})$. Steps 6 to 8 update $\mathbf{z}$ with the so-called proximal points for given prox-centers $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$. Step 6 computes $\mathbf{z}_{l+s}$ as the projection of $\mathbf{q}_1$ onto the convex set $\{\mathbf{v} \mid \|\mathbf{v} - \mathbf{c}\|_2^2 \le \delta\}$. Step 7 obtains $\mathbf{z}_l$ as the proximal point of the ℓ_2 distance from $\mathcal{S}_{n-1}$ for the prox-center $\mathbf{q}_2$. When $g_s(\mathbf{v}) = \lambda_s\|\mathbf{v}\|_1$, Step 8 becomes $\mathbf{z}_s \leftarrow \mathrm{soft}(\mathbf{q}_3, \lambda_s/\rho)$ where the soft thresholding operation for a complex value $\chi \in \mathbb{C}$ is defined as

$$\mathrm{soft}(\chi,\theta) := \exp(\sqrt{-1}\arg\chi)\max(|\chi| - \theta, 0), \quad (6)$$

and it works element-wise on vectors.

For the estimation of the basis matrix, we can employ the incremental singular value decomposition [32]. Let $\boldsymbol{\kappa}^{(n-1)} \in \mathbb{R}_+^r$ and $\mathbf{U}^{(n-1)}$ are respectively a vector of the singular values and a matrix of the left singular vectors of $[\mathbf{l}^{(1)},\dots,\mathbf{l}^{(n-1)}] \in \mathbb{C}^{M\times(n-1)}$. Then, the new basis matrix $\mathbf{U}^{(n)}$ and singular values $\boldsymbol{\kappa}^{(n)}$ are obtained by Algorithm 2 as

$$(\mathbf{U}^{(n)}, \boldsymbol{\kappa}^{(n)}) \leftarrow \textsc{UpdateBasis}(\mathbf{U}^{(n-1)}, \boldsymbol{\kappa}^{(n-1)}, \mathbf{l}^{(n)}). \quad (7)$$

Step 5 performs the singular value decompositon of $\mathbf{B} \in \mathbb{C}^{(r+1)\times(r+1)}$ to obtain $\mathbf{U}_B$, diagonal matrix $\mathbf{K}_B = \mathrm{diag}(\boldsymbol{\kappa}_B)$, and $\mathbf{V}_B$ such that $\mathbf{U}_B\mathbf{K}_B\mathbf{V}_B^\top = \mathbf{B}$ and $\mathbf{U}_B^\top\mathbf{U}_B = \mathbf{V}_B^\top\mathbf{V}_B = \mathbf{I}$. Step 7 and 8 truncate the rank to $r_{\max}$.

C. Variant Algorithms

a) Adaptive Estimation: If the subspace $\mathcal{S}_n$ changes fast with respect to n, then $\mathbf{l}^{(n)}$ obtained by Eq. (5) is inaccurate because the basis $\mathbf{U}^{(n-1)}$ cannot approximate the low-rank component in $\mathcal{S}_n$. A heuristic improvement is to adjust $\mathbf{U}$ using $\mathbf{l}$ during the estimation, i.e., , insert

$$(\mathbf{U}, \boldsymbol{\kappa}) \leftarrow \textsc{UpdateBasis}(\mathbf{U}^{(n-1)}, \boldsymbol{\kappa}^{(n-1)}, \mathbf{l}) \quad (8)$$

before Step 7 in Algorithm 1.

b) Debiasing: Since we allow the approximation error by δ, the ℓ_1 regularizer might choose $\mathbf{z}_s$ with smaller ℓ_1 norm at the sacrifice of acceptable approximation error. The soft thresholding operation, appeared in Algorithm 1 as the proximity operator of ℓ_1 regularizer $g_s(\mathbf{z}_s)$, shrinks every entry of the given vector to select nonzero entries of $\mathbf{z}_s$. This results in underestimate of nonzero entries. To remedy the underestimation, Fan and Li [33] proposed a non-concave sparse regularizer referred to as the smoothly clipped absolute deviation (SCAD). The proximity operator of the SCAD regularizer can be described as a thresholding operation.

$$\mathrm{SCAD}(\chi,\theta) = \begin{cases} \mathrm{soft}(\chi,\theta) & \text{if } |\chi| \le 2\theta, \\ \dfrac{(a-1)\chi - \mathrm{sign}(\chi)a\theta}{a-2} & \text{if } 2\theta < |\chi| \le a\theta, \\ \chi & \text{otherwise} \end{cases} \quad (9)$$

[1]The superscript $\top$ for a complex matrix indicates the conjugate transpose operation.

Algorithm 1 $(\mathbf{l}, \mathbf{s}) \leftarrow$ColSPCP$(\mathbf{c}, \mathbf{U}, \lambda_s, \delta, \rho, \mathrm{prox}_{g_s})$

Input: $\mathbf{c}$, $\mathbf{U}$, $\lambda_s > 0$, $\delta \geq 0$, $\rho > 0$, and
$\mathrm{prox}_{g_s}(\boldsymbol{\chi}, \theta) = \mathrm{soft}(\boldsymbol{\chi}, \theta)$ by default,
Output: $\mathbf{l}$ and $\mathbf{s}$.

1 Initialize $\mathbf{z}_{l+s}$, $\mathbf{z}_l$, $\mathbf{z}_s$, and $\mathbf{y}$;

2 **while** not converge **do**

3 $$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{z}_{l+s} \\ \mathbf{z}_l \\ \mathbf{z}_s \end{bmatrix} - \mathbf{y};$$

4 $$\begin{bmatrix} \mathbf{l} \\ \mathbf{s} \end{bmatrix} \leftarrow \frac{1}{3} \begin{bmatrix} \mathbf{u}_1 + 2\mathbf{u}_2 - \mathbf{u}_3 \\ \mathbf{u}_1 - \mathbf{u}_2 + 2\mathbf{u}_3 \end{bmatrix};$$

5 $$\begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{l} + \mathbf{s} \\ \mathbf{l} \\ \mathbf{s} \end{bmatrix} + \mathbf{y};$$

6 $$\mathbf{z}_{l+s} \leftarrow \arg\min_{\mathbf{v}} h(\mathbf{v}|\mathbf{c}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{q}_1\|_2^2 = \begin{cases} \mathbf{q}_1 & \text{if } \|\mathbf{q}_1 - \mathbf{c}\|_2 \leq \delta \\ \mathbf{c} + \delta \dfrac{\mathbf{q}_1 - \mathbf{c}}{\|\mathbf{q}_1 - \mathbf{c}\|_2} & \text{otherwise;} \end{cases}$$

7 $$\mathbf{z}_l \leftarrow \arg\min_{\mathbf{v}} \frac{1}{2}\|(\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\mathbf{v}\|_2^2 + \frac{\rho}{2}\|\mathbf{v} - \mathbf{q}_2\|_2^2 = \frac{\rho \mathbf{q}_2 + \mathbf{U}(\mathbf{U}^\top \mathbf{q}_2)}{\rho + 1};$$

8 $$\mathbf{z}_s \leftarrow \arg\min_{\mathbf{v}} g_s(\mathbf{v}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{q}_3\|_2^2 = \mathrm{prox}_{g_s}(\mathbf{q}_3, \lambda_s/\rho);$$

9 $$\mathbf{y} \leftarrow \mathbf{y} + \begin{bmatrix} \mathbf{l} + \mathbf{s} - \mathbf{z}_{l+s} \\ \mathbf{l} - \mathbf{z}_l \\ \mathbf{s} - \mathbf{z}_s \end{bmatrix};$$

10 **end while**

Algorithm 2 $(\mathbf{U}_{\text{new}}, \boldsymbol{\kappa}_{\text{new}}) \leftarrow$UpdateBasis$(\mathbf{U}, \boldsymbol{\kappa}, \mathbf{l}, r_{\max})$

Input: $\mathbf{U} \in \mathbb{C}^{M \times r}$, $\boldsymbol{\kappa} \in \mathbb{R}_+^r$, $\mathbf{l} \in \mathbb{C}^M$, and $r_{\max}$,
Output: $\mathbf{U}_{\text{new}}$ and $\boldsymbol{\kappa}_{\text{new}}$.

1 $\boldsymbol{\eta} \leftarrow \mathbf{U}^\top \mathbf{l}$;

2 $\mathbf{p} \leftarrow \mathbf{l} - \mathbf{U}\boldsymbol{\eta}$;

3 $p \leftarrow \|\mathbf{p}\|_2$;

4 $\mathbf{B} \leftarrow \begin{bmatrix} \mathrm{diag}(\boldsymbol{\kappa}) & \boldsymbol{\eta} \\ \mathbf{0}^\top & p \end{bmatrix}$;

5 $(\mathbf{U}_B, \mathrm{diag}(\boldsymbol{\kappa}_B), \mathbf{V}_B) \leftarrow \mathrm{svd}(\mathbf{B})$;
(singular value decomposition of $\mathbf{B}$)

6 $\mathbf{U}_{\text{new}} \leftarrow [\mathbf{U}, \frac{\mathbf{p}}{p}]\mathbf{U}_B$;

7 $\mathbf{U}_{\text{new}} \leftarrow \mathbf{U}_{\text{new}}(:, 1\!:\!\min(r+1, r_{\max}))$;

8 $\boldsymbol{\kappa}_{\text{new}} \leftarrow \boldsymbol{\kappa}_B(1\!:\!\min(r+1, r_{\max}))$;

Here, $\mathrm{sign}(\chi) = \exp(\sqrt{-1} \arg \chi)$ for $\chi \in \mathbb{C}$. The constant $a > 2$ is suggested to set to 3.7. SCAD shall work element-wise on vectors. Large entries in $\mathbf{z}_s$ are free from shrinkage because $\mathrm{SCAD}(\chi, \theta)$ acts as hard thresholding for $|\chi| > a\theta$. After the basis update as in Eq. (7), we recommend rerunning

$$(\mathbf{l}^{(n)}, \mathbf{s}^{(n)}) \leftarrow \text{ColSPCP}(\mathbf{c}^{(n)}, \mathbf{U}^{(n)}, \lambda_s, \delta, \rho, \text{SCAD}) \quad (10)$$

using $\mathbf{s}^{(n)}$ obtained by Eq. (5) as the initial guess of $\mathbf{z}_s$ at Step 1 in Algorithm 1 to remove the underestimation bias.

III. Sequence of 2D Apparent Motion Fields

A. Optical Flow

Optical flow of a time-varying two-dimensional image is a vector field describing displacement of the image intensity in the manner of continuum machanics as

$$\mathbf{d}(x, y, t) = \begin{bmatrix} d_x(x, y, t) \\ d_y(x, y, t) \end{bmatrix} := \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}. \quad (11)$$

Here, $[x, y]^\top$ denotes the position in the two-dimensional image domain, and t refers to time. Given a spatio-temporal image $I(x, y, t)$, we have an equation of continuity where the image intensity I is assumed as conserved quantity.

$$\frac{\partial I}{\partial t} + \nabla^\top (I\mathbf{d}) = 0 \quad (12)$$

This continuity equation is rewritten as

$$\frac{\partial I}{\partial t} + \mathbf{d}^\top \nabla I + I \nabla^\top \mathbf{d} = \frac{DI}{Dt} + I \,\mathrm{div}\, \mathbf{d} = 0 \quad (13)$$

where $\mathrm{div}\, \mathbf{d} := \nabla^\top \mathbf{d} = \partial d_x/\partial x + \partial d_y/\partial y$ is the divergence of the optical-flow field $\mathbf{d}$ at a fixed time t. The convective derivative of an image, DI/Dt, is defined as the time derivative of the image intensity as seen in a volume which is moving at the same rate as the flow, i.e.,

$$\begin{aligned} \frac{DI}{Dt} &:= \lim_{\Delta t \to 0} \frac{I(x + d_x \Delta t, y + d_y \Delta t, t + \Delta t) - I(x, y, t)}{\Delta t} \\ &= \frac{\partial I}{\partial t} + \mathbf{d}^\top \nabla I. \end{aligned} \quad (14)$$

In computer vision, optical flow is often assumed or approximated as incompressible. That is, the image intensity is considered to remain constant through time and displacement. The incompressible property, equivalent to a divergence free condition, reduces Eq. (13) to the so-called optical flow constraint (OFC):

$$\frac{DI}{Dt} = \frac{\partial I}{\partial t} + \mathbf{d}^\top \nabla I = 0. \quad (15)$$

Equation (15) is not enough to uniquely determine the vector $\mathbf{d}$. To resolve this underdetermined issue, computation of optical flow needs to impose some known structure or characteristics on the flow, such as smoothness over the image domain [1], piecewise constant [2], minimal total variation [34], [35] and so on.

Given a sequence of discrete images $I(x_m, y_m, t_n)$ ($m \in \{1, \ldots, M\}$ and $n \in \{1, \ldots, N+1\}$), one can compute a pixel-wise dense optical-flow field at each time t_n from the subsequent images at t_n and t_{n+1}. See [36], [37], [38], [5] for the review of existing methods for optical flow computation.

B. Low-Dimensional Nature of Optical-Flow Sequence

An optical camera observes three dimensional motion in the world, projecting them to apparent motion of a two-dimensional image. Temporal variation of the image taken by such a camera moving in a static scene depends on the camera's translational and rotational velocities, $\boldsymbol{\tau}(t) \in \mathbb{R}^3$ [m/s] and $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ [rad/s], respectively. For the pinhole camera model, one can derive the displacement by this camera egomotion, $\mathbf{d}_{\text{ego}}$, at a point $[x, y]^\top$ on the image plane corresponding to a 3D point $[X, Y, Z]^\top$ in the camera coordinate system [8].

$$\mathbf{d}_{\text{ego}} := \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\boldsymbol{\tau} + \frac{1}{f}\begin{bmatrix} xy & -(f^2+x^2) & fy \\ f^2+y^2 & -xy & -fx \end{bmatrix}\boldsymbol{\omega} \quad (16)$$

Here, f is the focal length, and a 3D point is projected to $[x, y]^\top = f/Z \cdot [X, Y]^\top$ on the image plane.

Equation (16) implies that the egomotional optical flow $\mathbf{d}_{\text{ego}}$ is a linear combination of the six column vectors of the matrices, controlled by the camera velocities as their coefficients. We regard these column vectors as the basis vectors of the optical flow. The vector field of optical flow on the image plane can likewise be viewed as a linear combination of the corresponding six basis optical-flow fields.

Let us denote a two-dimensional optical-flow vector $\mathbf{d} = [d_x, d_y]^\top \in \mathbb{R}^2$ as a complex number

$$d := d_x + \sqrt{-1}d_y \in \mathbb{C}. \quad (17)$$

Then, an optical-flow field at each time t_n computed from a sequence of discrete images of M pixels is represented as a M-dimensional complex vector

$$\mathbf{c}^{(n)} := [d(x_1, y_1, t_n), \ldots, d(x_M, y_M, t_n)]^\top \in \mathbb{C}^M. \quad (18)$$

This complex-number representation reveals low-dimensional nature of a sequence of optical-flow fields induced by egomotion in a static scene. Since an optical-flow field is a linear combination of six basis optical-flow fields, the M-dimensional vector $\mathbf{c}^{(n)} \in \mathbb{C}^M$ is composed of six M-dimensional complex vectors of them as well. If the basis vectors are time invariant, the time sequence $\{\mathbf{c}^{(n)}\}$ resides in the six-dimensional subspace of $\mathbb{C}^M$ spanned by the M-dimensional complex vectors. Stacking the time sequence into a complex matrix as $\mathbf{C} := [\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(N)}] \in \mathbb{C}^{M \times N}$, we have rank $\mathbf{C} \leq 6$. It was suggested that the optical flow of a central panoramic camera also has the very similar nature of rank-six constraint [18].

In general, the basis vectors are not time invariant: the depth Z depends on relative position between the moving camera and scene. Nevertheless, one can expect that optical-flow fields of the background can be approximately expressed as a linear combination of a very small number of vector fields in a short time window. When there exist arbitrarily moving foreground objects, their optical-flow field is superposed onto the background, which disrupts the low-dimensional nature of optical-flow sequence. Fourtunately, such foreground objects can be assumed to occupy small image regions in many cases. As a consequence, the complex matrix of optical-flow sequence, $\mathbf{C}$, can be approximated as a sum of low-rank and sparse matrices. The low-rank matrix tells us the apparent background motion induced by camera egomotion, and the support of the sparse matrix are associated with the pixels of moving objects.

IV. Experimental Examples

We demonstrate our online algorithm of low-rank and sparse approximation on a synthetic driving sequence in SET 2 of EISATS [39]. We resize its dense optical-flow fields to M=160×120=19,200 pixel resolution, and express them as a sequence of complex vectors $\mathbf{c}^{(n)}$ ($n = 1, \ldots, 395$).

Figure 1 shows examples of the decomposition results. We set $\lambda_s = 2$, $\delta = 0.02\|\mathbf{c}^{(n)}\|_2$, $\rho = 1$, and $r_{\text{max}} = 12$. Our algorithm ColSPCP with the adaptive estimation and debiasing described in II-C renders the optical-flow fields of egomotion and object motion as shown in Fig. 1-(i) and (iii). It runs on a standard laptop at about five fps with MATLAB-only implementation, and six times faster than ReProCS [28]. As suggtested in I, existing online algorithms for RPCA are noise intolerant. For example, ReProCS first estimates the sparse component $\mathbf{s}^{(n)}$ using $\mathbf{c}^n$ and $\mathbf{U}^{(n-1)}$, and then computes the low-rank component simply as $\mathbf{l}^{(n)} := \mathbf{c}^{(n)} - \mathbf{s}^{(n)}$. The noise in $\mathbf{c}^{(n)}$ and approximation error remain in $\mathbf{l}^{(n)}$, and its following procedure using $\mathbf{l}^{(n)}$ can contaminate the results. This contamination is observed in Fig. 1-(ii) and (iv) when the depth Z changes rapidly neaby the horizion. Since our algorithm ColSPCP simultaneously estimates $\mathbf{l}^{(n)}$ and $\mathbf{s}^{(n)}$ taking the approximation error into account, and it is capable of adjusting the basis matrix during the estimation, the resulting two components are stable against the approximation error and changes of subspace. At the $n = 101$st frame, an oncoming car is clearly found in $\mathbf{s}^{(101)}$ by ColSPCP with an F-score of 0.717 (0.705 precision and 0.731 recall). The F-score of $\mathbf{s}^{(101)}$ by ReProCS is 0.233 (0.135 precision and 0.85 recall) for reference.

V. Conclusions

The stable principal component pursuit enables an unsupervised video analysis. Its online algorithm provides apparent motion fields by camera egomotion and object motion respectively as the low-rank and sparse components at each video frame. Our algorithm ColSPCP is time and memory efficient, and tolerant against the error in the low-rank and sparse approximation. There is much room for improvement and extension. Support estimation or prediction of the sparse solution for the next frame acts as multiple object tracking and improves the detection performance. A new method for optical flow computation with low-rank and sparse regulariztion could be designed on the basis of ColSPCP algorithm by evaluating the optical flow constraint in h.

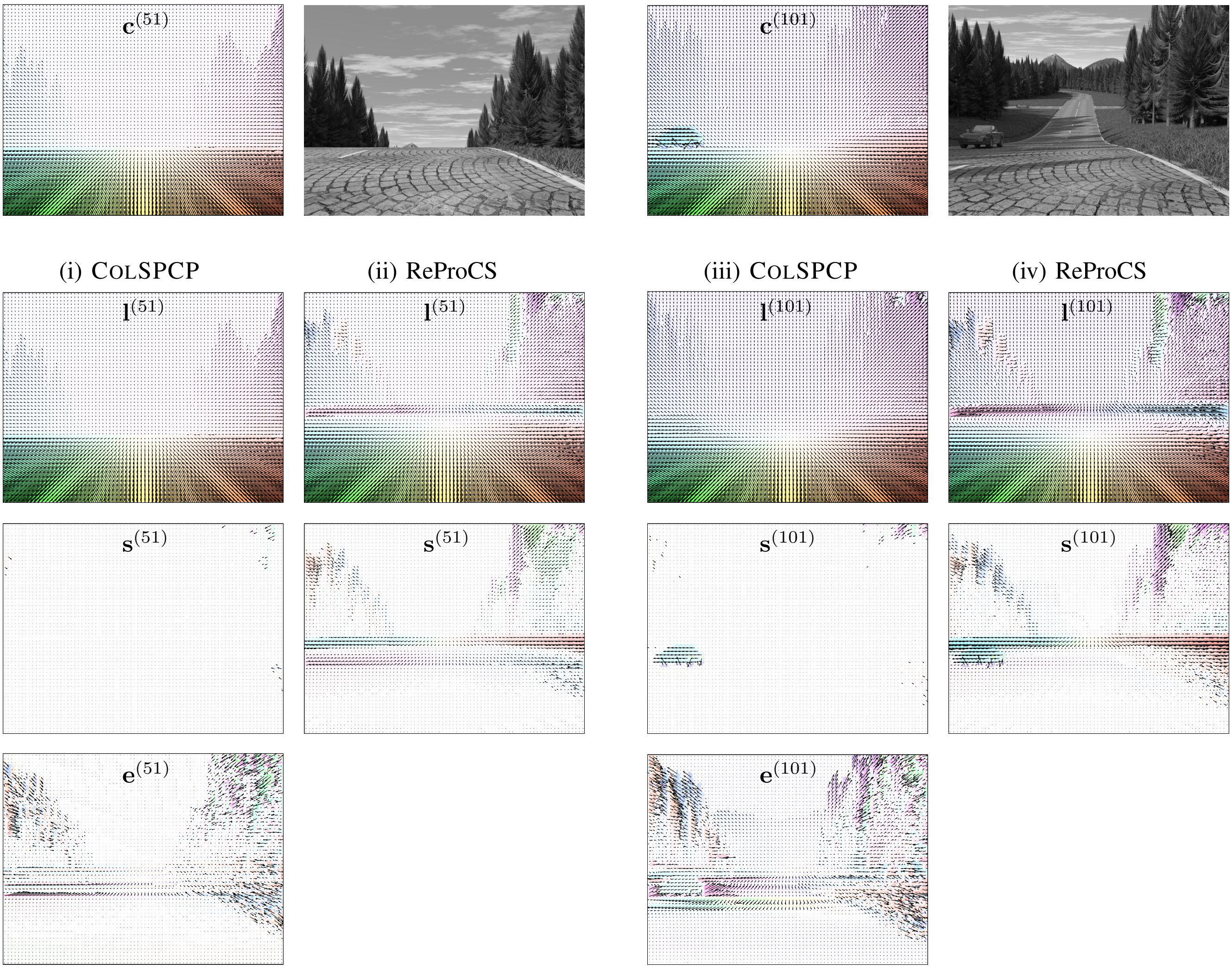

Fig. 1. Decompositon of optical-flow fields of a driving sequence (SET 2 of EISATS [39]). First row: optical-flow fields and images at $n = 51$st and 101st video frames, respectively. Second and third rows: low-rank and sparse components. Fourth row: approximation error components $\mathbf{e}^{(n)} = \mathbf{c}^{(n)} - (\mathbf{l}^{(n)} + \mathbf{s}^{(n)})$ (magnified by ten). Columns (i) and (iii) are the results by ColSPCP with adaptive estimation and debiasing (ours), and (ii) and (iv) are those by ReProCS [28].

References

[1] Berthold K.P. Horn and Brian G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[2] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Imaging Understanding Workshop*, 1981, pp. 121–130.

[3] Steven S. Beauchemin and John L. Barron, "The computation of optical flow," *ACM Computing Surveys (CSUR)*, vol. 27, no. 3, pp. 433–466, 1995.

[4] David Fleet and Yair Weiss, "Optical flow estimation," in *Handbook of Mathematical Models in Computer Vision*, pp. 237–257. Springer, 2006.

[5] F. Raudies, "Optic flow," *Scholarpedia*, vol. 8, no. 7, pp. 30724, 2013.

[6] Massimo Piccardi, "Background subtraction techniques: a review," in *Systems, man and cybernetics, 2004 IEEE international conference on*. IEEE, 2004, vol. 4, pp. 3099–3104.

[7] Thierry Bouwmans and El Hadi Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014.

[8] H Christopher Longuet-Higgins and Kvetoslav Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 208, no. 1173, pp. 385–397, 1980.

[9] David Nistér, "Preemptive RANSAC for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, no. 5, pp. 321–329, 2005.

[10] Richard JM Den Hollander and Alan Hanjalic, "A combined RANSAC-Hough transform algorithm for fundamental matrix estimation.," in *British Machine Vision Conference*, 2007, pp. 1–10.

[11] Oliver W Layton, Ennio Mingolla, and N Andrew Browning, "A motion pooling model of visually guided navigation explains human behavior in the presence of independently moving objects," *Journal of vision*, vol. 12, no. 1, pp. 20, 2012.

[12] Yong Xu, Jixiang Dong, Bob Zhang, and Daoyun Xu, "Background modeling methods in video analysis: A review and comparative evaluation," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 43–60, 2016.

[13] Emanuele Trucco and Konstantinos Plakas, "Video tracking: a concise survey," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 2, pp. 520–529, 2006.

[14] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[15] David J Fleet, Michael J Black, Yaser Yacoob, and Allan D Jepson, "Design and use of linear models for image motion analysis," *International Journal of Computer Vision*, vol. 36, no. 3, pp. 171–193, 2000.

[16] Michal Irani, "Multi-frame correspondence estimation using subspace constraints," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 173–194, 2002.

[17] Omid Shakernia, René Vidal, and Shankar Sastry, "Multibody motion estimation and segmentation from multiple central panoramic views," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. IEEE, 2003, vol. 1, pp. 571–576.

[18] René Vidal, "Segmentation of dynamic scenes taken by a moving central panoramic camera," in *Imaging Beyond the Pinhole Camera*, pp. 125–142. Springer, 2006.

[19] Tomoya Sakai and Hiroki Kuhara, "Separating background and foreground optical flow fields by low-rank and sparse regularization," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1523–1527.

[20] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.

[21] John Wright, Arvind Ganesh, Shankar Rao, YiGang Peng, and Yi Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *NIPS*, 2009, pp. 2080–2088.

[22] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky, "Sparse and low-rank matrix decompositions," in *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, 2009, Allerton'09, pp. 962–967.

[23] Silvia Gandy and Isao Yamada, "Convex optimization techniques for the efficient recovery of a sparsely corrupted low-rank matrix," *Journal of Math-for-Industry*, vol. 2, no. 2010B-5, pp. 147–156, 2010.

[24] Min Tao and Xiaoming Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[25] Xiaoming Yuan and Junfeng Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," *Pacific Journal of Optimization*, vol. 9, no. 1, pp. 167–180, 2013.

[26] Necdet Serhat Aybat, Donald Goldfarb, and Shiqian Ma, "Efficient algorithms for robust and stable principal component pursuit problems," *Computational Optimization and Applications*, vol. 58, no. 1, pp. 1–29, 2014.

[27] Jun He, Laura Balzano, and Arthur Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1568–1575.

[28] Han Guo, Chenlu Qiu, and Namrata Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4284–4297, 2014.

[29] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *IEEE ISIT Proceedings*, 2010, pp. 1518–1522.

[30] Daniel Gabay and Bertrand Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers and Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[31] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[32] Matthew Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. ECCV 2002*, 2002, pp. 707–720.

[33] Jianqing Fan and Runze Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[34] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Proceedings of the 29th DAGM Conference on Pattern Recognition*, Berlin, Heidelberg, 2007, pp. 214–223, Springer-Verlag.

[35] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers, "An improved algorithm for TV-L1 optical flow," in *Statistical and Geometrical Approaches to Visual Motion Analysis*, pp. 23–45. Springer-Verlag, Berlin, Heidelberg, 2009.

[36] John L. Barron, David J. Fleet, and Steven Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[37] Edward Castillo, *Optical flow methods for the registration of compressible flow images and images containing large voxel displacements or artifacts*, PhD thesis, Rice University, 2007.

[38] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[39] Tobi Vaudrey, Clemens Rabe, Reinhard Klette, and James Milburn, "Differences between stereo and motion behavior on synthetic and real-world stereo sequences," in *23rd International Conference of Image and Vision Computing New Zealand (IVCNZ '08)*, 2008, pp. 1–6.