Shifted Subspaces Tracking on Sparse Outlier for Motion Segmentation

Tianyi Zhou and Dacheng Tao
Centre for Quantum Computation & Intelligent Systems
University of Technology Sydney, NSW 2007, Australia
tianyi.david.zhou@gmail.com; dacheng.tao@uts.edu.au

Abstract

In low-rank & sparse matrix decomposition, the entries of the sparse part are often assumed to be i.i.d. sampled from a random distribution. But the structure of sparse part, as the central interest of many problems, has been rarely studied. One motivating problem is tracking multiple sparse object flows (motions) in video. We introduce "shifted subspaces tracking (SST)" to segment the motions and recover their trajectories by exploring the low-rank property of background and the shifted subspace property of each motion. SST is composed of two steps, background modeling and flow tracking. In step 1, we propose "semi-soft GoDec" to separate all the motions from the low-rank background L as a sparse outlier S. Its soft-thresholding in updating S significantly speeds up GoDec and facilitates the parameter tuning. In step 2, we update X as S obtained in step 1 and develop "SST algorithm" further decomposing X as $X = \sum_{i=1}^{k} L(i) \circ \tau(i) + S + G$, wherein $L(i)$ is a low-rank matrix storing the i^{th} flow after transformation $\tau(i)$. SST algorithm solves k sub-problems in sequel by alternating minimization, each of which recovers one $L(i)$ and its $\tau(i)$ by randomized method. Sparsity of $L(i)$ and between-frame affinity are leveraged to save computations. We justify the effectiveness of SST on surveillance video sequences.

1 Introduction

In video sequences, an object flow is composed of multiple motions or moving objects with the identical trajectory. Analyzing object flows is more frequently preferred than analyzing the motion of a single object, because the flows can provide more semantic clues for the crowd behavior [Ali and Shah, 2007]. Tracking multiple object flows and motion segmentation [Wu *et al.*, 2011][Galasso *et al.*, 2012] in complex scenes is a vital and challenging problem in a variety of computer vision tasks such as surveillance, robotics, augmented reality, medical imaging and human-computer interactions. The challenges mainly come from the complex background, occlusion, illumination variation, noise, overlapping, and intertwined trajectories of different flows [Fragkiadaki and Shi, 2011]. Some of these problems have been well studied in the previous research works, but some are typical features of the flow tracking or motion segmentation and thus have not been fully tackled before. In this paper, we construct a model that considers all the miscellaneous problems above, reduce the motion detection, tracking and segmentation to a simple matrix factorization, and then develop an efficient optimization algorithm to achieve the factorization result.

1.1 Related Work

To begin with, we can roughly categorize existing tracking approaches into two groups, i.e., generative model [Doucet *et al.*, 2001][Comaniciu *et al.*, 2003] and discriminative method [Hess and Fern, 2009][Hong *et al.*, 2012]. Generative model formulates tracking as estimation of the state within a time series space state space model, and search the regions of the highest likelihood. Early works such as Kalman filter and its variants have been demonstrated to be optimal for linear Gaussian model. Another representative generative model for tracking is particle filter [Doucet *et al.*, 2001], which approximates the posterior distribution of the state space by Monte Carlo integration. Single appearance model or multiple appearance models [Yu *et al.*, 2008] are used in these generative models. Different from generative models, discriminative classifies cast the tracking problem into a classification task whose goal is to distinguish the target object from the background. In the training stage, each pixel is represented by a feature vector, and the pixels belonging to the same target object is assigned to the same class. Hybrid approaches [Yu *et al.*, 2008] combining both generative models and discriminative classifiers have also been proposed for solving visual tracking under significant occlusions.

Recent advances in matrix completion [Candès and Recht, 2008][Ji and Ye, 2009] and robust principle component analysis (RPCA) [Chandrasekaran *et al.*, 2009][Candès *et al.*, 2009][Chen *et al.*, 2010][F. Nie, 2011] lead us to a new perspective for analyzing large-scale video data, which is exploring the inherent low-rank structures of background and motions. As an extension of matrix completion who recovers a low-rank matrix from a small portion of its entries, RPCA exactly recovers a low-rank matrix L from collected data matrix $X = L + S$, where S is sparse outlier of large magnitude on random support. RPCA decomposes X by minimizing the convex surrogates of L's rank and S's cardinality, i.e., the

trace norm of L and the ℓ_1 norm of S,

$$\begin{array}{ll}\min\limits_{L,S} & \|L\|_* + \lambda\|S\|_1 \\ s.t. & X = L + S.\end{array} \tag{1}$$

Above convex optimization ensures exact or adequately precise recovery of L and S under mild identifiability conditions such as rank-sparsity coherence, on the premise that exact decomposition $X = L + S$ does exist. However, this premise cannot be always fulfilled for data collected from real applications. Moreover, the rank and cardinality cannot be fixed within a small range while the error is still guaranteed to be small. This uncertainty brings unpredictable computational costs in optimization.

RPCA has been successfully applied to background modeling in video surveillance [Candès *et al.*, 2009], where the backgrounds in all frames compose the rows of low-rank matrix L, whilst the moving objects or motions are captured by the sparse outlier S. However, most existing methods simply treat S as random noise and lack further study of the obtained sparse outlier, which in fact contains substantial rich information about the motions of object flows. In addition, the extra dense noise caused by camera lens or environment illumination changes make the exact decomposition assumption $X = L + S$ cannot be satisfied in practices. Hence several recent approaches [Zhou *et al.*, 2010][Hsu *et al.*, 2011][Zhou and Tao, 2013] aim to obtain the approximated RPCA decomposition $X = L + S + G$, where G is the dense noise. Since most RPCA algorithms rely on repeating time consuming singular value thresholding, randomized method [Zhou and Tao, 2012] was introduced for acceleration.

In order to overcome the shortcomings of existing RPCA approaches, a novel method GoDec [Zhou and Tao, 2011] is proposed. GoDec imposes hard constraints to the rank of L and the cardinality of S [Xiong *et al.*, 2010], and additionally accelerates the noisy decomposition with the help of bilateral random projections (BRP) based low-rank approximation [Zhou and Tao, 2012] and controllable rank of L.

$$\begin{array}{ll}\min\limits_{L,S} & \|X - L - S\|_F^2 \\ s.t. & \mathrm{rank}(L) \le r, \mathrm{card}(S) \le k.\end{array} \tag{2}$$

Although linear convergence of GoDec can be proved by the framework "alternating projections on two manifolds" [Lewis and Malick, 2008], the upper bound for cardinality of S has to be carefully chosen in GoDec. Otherwise some portion of S's entries will be contaminated or lost. Moreover, the hard thresholding towards S requires sorting all its entries' magnitudes and thus is time consuming.

1.2 Overview of SST

A significant open problem left by the mentioned RPCA approaches is how to further analyze the rich structure of the sparse part. In many practical applications, the sparse part has structures that can contribute more useful information than the low-rank part. An example in point is that the sparse part of video sequence data is comprised of the motions of multiple object flows. In this paper, we consider decomposing the sparse part as the sum of several shifted low-rank matrices, each of which corresponds to objects sharing one motion trajectory. This can be viewed as a novel structured sparsity. It is also worthwhile to point out that although the main focus of this paper is object flow tracking and motion segmentation, the SST framework allows other forms of nonlinear transformation and thus can be directly applied to other problems.

We develop an efficient unsupervised framework to detect, track and segment multiple motions in complex scenes via solving a matrix factorization model. This framework invokes a sequence of matrix decompositions as subroutines, which can be summarized in two steps, i.e., background modeling and flow tracking. For the first step, we proposes "semi-soft GoDec" which replaces the cardinality constraint in GoDec with an ℓ_1 penalty. This small change greatly shortens the computational time and facilitates the parameter tuning. For the second step, as the main contribution of this paper, we present a novel insight that each flow can be depicted by a low rank matrix after certain geometric transformation sequence to video frames (which are stored as rows of a matrix), and develop a matrix factorization method to recover both the low-rank matrices and the transformation sequences. If we treat the sparse outliers attained by semi-soft GoDec as the new data matrix X, our flow tracking approach "shifted subspaces tracking (SST)" decomposes X as $X = \sum_{i=1}^{k} L(i) \circ \tau(i) + S + G$, where $L(i)$ stands for a low-rank matrix and $\tau(i)$ stands for a transformation sequence, both correspond to the motion of the i^{th} object flow. SST reduces the matrix factorization to a sequence of alternating optimizations in a similar manner with semi-soft GoDec, and efficiently solves them by taking advantages of the between-frame affinity and the motion sparsity of $L(i)$. Besides, BRP [Zhou and Tao, 2012] is invoked to speed up the update of $L(i)$. The low-rank patterns, together with their transformation sequences, reveals unexplored structure of the sparse part and rich information of segmented motion in complex scenes.

2 Problem Setup

We consider the problem of tracking object flows and segmenting their motions from the raw video data. Given a data matrix $X \in \mathbb{R}^{n \times p}$ that stores a video sequence of n frames, each of which has $w \times h = p$ pixels and reshaped as a row vector in X, the goal of SST framework is to separate the motions of the object flows, recover both their low-rank patterns and geometric transformation sequences. This task is decomposed as two steps in SST, i.e., background modeling that separating all the moving objects from the static background, and flow tracking that recovers the information of each motion. In this paper, $\cdot_i$ stands for the i^{th} entry of a vector or the i^{th} row of a matrix, while $\cdot_{i,j}$ signifies the entry at the i^{th} row and the j^{th} column of a matrix.

2.1 Background modeling

Although previous RPCA approaches provides several effective matrix decomposition formulations for background modeling, some problems still arise in real applications. In SST, we consider the low-rank and sparse matrix decomposition in noisy case, which has been adopted by several recent models

due to its robustness and adaptiveness to the real data,

$$X = L + S + G, \mathrm{rank}(L) \leq r, \mathrm{card}(S) \leq s, \quad (3)$$

where L describes the background and S denotes the motions.

In SST, a new formulation for background modeling is adopted to overcome the shortcomings of both RPCA and GoDec,

$$\begin{array}{ll} \min\limits_{L,S} & \|X - L - S\|_F^2 + \lambda\|S\|_1 \\ s.t. & \mathrm{rank}\,(L) \leq r. \end{array} \quad (4)$$

Compared with RPCA, (4) does not require the exact decomposition $X = L + S$. Furthermore, the rank of L is controllable and the update of L can be randomized as in GoDec, thus the time cost can be largely decreased. Compared with GoDec, (4) replaces the "hard" cardinality constraint with a "soft" ℓ_1 regularization. Tuning soft threshold λ is much easier than determining s in (3), because the resulting decomposition error is more robust to the change of λ. Later, we will develop "semi-soft GoDec" to solve the optimization (4).

2.2 Flow tracking

After obtaining the sparse outliers storing motions by background modeling, SST treats the sparse matrix S as the new data matrix X, and decomposes it as $X = \sum_{i=1}^{k} \tilde{L}(i) + S + G$, wherein $\tilde{L}(i)$ denotes the i^{th} object flow, S stands for the sparse outliers and G stands for the Gaussian noise.

The matrix decomposition model for flow tracking in SST is based on an observation to the implicit structures of the sparse matrix $\tilde{L}(i)$. If the trajectory of the object flow $\tilde{L}(i)$ is known and each frame (row) in $\tilde{L}(i)$ is shifted to the position of a reference frame, due to the limited number of poses for the same object flow in different frames, it is reasonable to assume that the rows of the shifted $\tilde{L}(i)$ exist in a subspace. In other words, $\tilde{L}(i)$ after inverse geometric transformation is low-rank. Hence the sparse motion matrix $\tilde{L}(i)$ has the following structured representation

$$\tilde{L}(i) = \begin{bmatrix} L(i)_1 \circ \tau(i)_1 \\ \vdots \\ L(i)_n \circ \tau(i)_n \end{bmatrix} = L(i) \circ \tau(i). \quad (5)$$

The invertible transformation $\tau(i)_j : \mathbb{R}^2 \to \mathbb{R}^2$ denotes the 2-D geometric transformation (to the reference frame) associated with the i^{th} object flow in the j^{th} frame, which is represented by $L(i)_j$. To be specific, the j^{th} row in $\tilde{L}(i)$ is $L(i)_j$ after certain permutation of its entries. The permutation results from applying the nonlinear transformation $\tau(i)_j$ to each nonzero pixel in $L(i)_j$ such that,

$$\tau(i)_j(x, y) = (u, v), \quad (6)$$

where $\tau(i)_j$ could be one of the five geometric transformations [Prince, 2011], i.e., translation, Euclidean, similarity, affine and homography, which are able to be represented by 2, 3, 4, 6 and 9 free parameters, respectively. For example, affine transformation is defined as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \rho\cos\theta & \rho\sin\theta \\ -\rho\sin\theta & \rho\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \quad (7)$$

wherein θ is the rotation angle, t_x and t_y are the two translations and ρ is the scaling ratio. It is worth to point out that $\tau(i)_j$ can be any other transformation beyond the geometric group. So SST can be applied to sparse structure in other applications if parametric form of $\tau(i)_j$ is known. We define the nonlinear operator $\circ$ as

$$\begin{aligned} \tilde{L}(i)_{j,u+(v-1)h} &= (L(i)_j \circ \tau(i)_j)_{u+(v-1)h} \\ &= L(i)_{j,x+(y-1)h}. \end{aligned} \quad (8)$$

Therefore, the flow tracking in SST aims at decomposing the sparse matrix X (S obtained in the background modeling) as

$$\begin{array}{l} X = \sum\limits_{i=1}^{k} L(i) \circ \tau(i) + S + G, \\ \mathrm{rank}\,(L(i)) \leq r_i, \mathrm{card}(S) \leq s. \end{array} \quad (9)$$

In SST, we iteratively invoke k times of the following matrix decomposition to greedily construct the decomposition in (9):

$$X = L \circ \tau + S + G, \mathrm{rank}\,(L) \leq r, \mathrm{card}(S) \leq s. \quad (10)$$

In each time of the matrix decomposition above, the data matrix X is S obtained by former decomposition. In order to save the computation and facilitate the parameter tuning, we cast the decomposition (10) into an optimization similar to (4),

$$\begin{array}{ll} \min\limits_{L,\tau,S} & \|X - L \circ \tau - S\|_F^2 + \lambda\|S\|_1 \\ s.t. & \mathrm{rank}\,(L) \leq r, \end{array} \quad (11)$$

To summarize, the optimization scheme of SST framework is given in Algorithm 1.

Algorithm 1 SST framework

Input: X, r, λ, r_i, $\lambda_i (i = 1, \cdots, k)$,
Output: L, $L(i), \tau(i) (i = 1, \cdots, k)$, S
$(L^\star, S^\star) = \arg\min\limits_{\mathrm{rank}(L) \leq r, S} \|X - L - S\|_F^2 + \lambda\|S\|_1$.
$X := S^\star, L := L^\star$.
for $i = 1 \to k$ **do**
  $(L^\star, \tau^\star, S^\star) =$
  $\arg\min\limits_{\mathrm{rank}(L) \leq r_i, \tau, S} \|X - L \circ \tau - S\|_F^2 + \lambda_i\|S\|_1$.
  $X := S^\star, L(i) := L^\star, \tau(i) = \tau^\star$;
end for
$S := X$.

3 SST framework

In this section, we develop efficient algorithms to solve the two types of minimization in Algorithm 1, both of which can be addressed by alternating minimization. In our algorithm, we speed up the update of low-rank part L in (4) and the low-rank motion patterns $L(i)$ in (11) by bilateral random projection based low-rank approximation [Zhou and Tao, 2012]. In ST algorithm for flow tracking and motion segmentation, piece-wise linear approximation method is used to approximate the nonlinear operator $\circ$ when updating τ. Both the sparsity of $L(i)$ and the between-frame affinity are leveraged to save computations. We also establish a rule to update the reference frame, which decides the uniqueness of $\tau(i)$ and avoids missing any object flow.

3.1 Semi-Soft GoDec for Background Modeling

We propose semi-soft GoDec for optimization (4), which can be solved by alternatively solving the following two subproblems until convergence:

$$\begin{cases} L^t = \arg\min\limits_{\mathrm{rank}(L)\le r} \|X - L - S^{t-1}\|_F^2 \\ S^t = \arg\min\limits_{S} \|X - L^t - S\|_F^2 + \lambda\|S\|_1 \end{cases} \tag{12}$$

The subproblems have global solutions L^t and S^t that can be obtained via closed-forms.

In particular, the two subproblems in (12) can be solved by updating L^t via singular value hard thresholding of $X - S^{t-1}$ and updating S^t via soft thresholding of $X - L^t$, respectively.

$$\begin{cases} L^t = \sum\limits_{i=1}^{r} \lambda_i U_i V_i^T, \mathrm{svd}\left(X - S^{t-1}\right) = U\Lambda V^T \\ S^t = \mathcal{P}_\lambda\left(X - L^t\right), \mathcal{P}_\lambda(x) = \mathrm{sign}(x)\max\left(|x| - \lambda, 0\right) \end{cases}$$

According to the bilateral random projection (BRP) based low-rank approximation [Zhou and Tao, 2012], the costly truncated SVDs in updating L^t can be approximated by

$$\mathrm{BRP}(L^t) = Y_1\left(A_2^T Y_1\right)^{-1} Y_2^T \tag{13}$$

or its power scheme modification, wherein $Y_1 \in \mathbb{R}^{n\times r}$ and $Y_2 \in \mathbb{R}^{p\times r}$ are the left and right random projections and A_2 is the left random projection matrix. The obtained semi-soft GoDec algorithm is given in Algorithm 2.

Algorithm 2 Semi-soft GoDec

Input: X, r, λ, ϵ, q
Output: L, S
Initialize: $L^0 := X$, $S^0 := \mathbf{0}$, $t := 0$
while $\|X - L^t - S^t\|_F^2 / \|X\|_F^2 > \epsilon$ **do**
  $t := t + 1$;
  $\tilde{L} = \left[\left(X - S^{t-1}\right)\left(X - S^{t-1}\right)^T\right]^q \left(X - S^{t-1}\right)$;
  $Y_1 = \tilde{L}A_1$, $A_2 = Y_1$;
  $Y_2 = \tilde{L}^T Y_1 = Q_2 R_2$, $Y_1 = \tilde{L}Y_2 = Q_1 R_1$;
  If $\mathrm{rank}\left(A_2^T Y_1\right) < r$ **then** $r := \mathrm{rank}\left(A_2^T Y_1\right)$, go to the first step; **end**;
  $L^t = Q_1\left[R_1\left(A_2^T Y_1\right)^{-1} R_2^T\right]^{1/(2q+1)} Q_2^T$;
  $S^t = \mathcal{P}_\lambda\left(X - L^t\right)$,
  where $\mathcal{P}_\lambda(x) = \mathrm{sign}(x)\max\left(|x| - \lambda, 0\right)$;
end while

Here q is the power parameter. When $q > 0$, the algorithm adopts a power-scheme modification of BRP for improving approximation accuracy [Zhou and Tao, 2011]. When $q = 0$, for dense X, (13) is applied. In the latter case, the QR decomposition of Y_1 and Y_2 in Algorithm 2 are not performed, and L^t is updated as $L^t = Y_1\left(A_2^T Y_1\right)^{-1} Y_2^T$. Actually, $q = 0$ is sufficient to produce satisfying accuracy in most visual applications. The key difference of Semi-soft GoDec comparing to the ordinary GoDec is the soft-thresholding of S, which requires merely np subtractions, while the hard-thresholding of the largest entries in ordinary GoDec needs sorting of np values. This improvement further speedup the computation of background modeling. The linear convergence of L still holds and can be proved by following the same procedure in [Zhou and Tao, 2011].

3.2 SST algorithm for Object Flow Tracking

Flow tracking in SST solves a sequence of optimization problem of type (11). Thus we firstly apply alternating minimization to (11). This results in iterative update of the solutions to the following three subproblems,

$$\begin{cases} \tau^t = \arg\min\limits_{\tau} \|X - L^{t-1}\circ\tau - S^{t-1}\|_F^2; \\ L^t = \arg\min\limits_{\mathrm{rank}(L)\le r} \|X - L\circ\tau^t - S^{t-1}\|_F^2; \\ S^t = \arg\min\limits_{S} \|X - L^t\circ\tau^t - S\|_F^2 + \lambda\|S\|_1. \end{cases} \tag{14}$$

Initialization

Unfortunately, alternating solving the 3 sub-problems might not guarantee to produce a global solution or even a stable one, unless an appropriate initialization is adopted. In particular, in the case when the solutions to $L\circ\tau$ and S in (11) are unique, the pair (L, τ) may not be unique. This is because that we can choose arbitrary frame as the reference frame and transform the object flow in all the other frames of $L\circ\tau$ to their positions in the reference frame, while the low-rank $L\circ\tau$ keeps the same. To avoid such trivial multiple solutions of τ, we pre-define a template frame s and do not update its transformation τ_s during the tracking (w.l.o.g., we fix all the parameters of τ_s to be zeros). Then the object flow in each other frame of $L\circ\tau$ are transformed to the position of the object flow in frame s via the inverse transform of τ, and thus the uniqueness of L and τ can be guaranteed. In this paper, we choose the template frame s as the one with the largest cardinality, which implies that almost all the objects are included in the frame,

$$s = \arg\max_i \mathrm{card}\left(X_i\right). \tag{15}$$

We then set the rows of the low-rank pattern L as the duplicates of X_s, and initialize both the entries of the sparse outlier S as well as the parameters of τ to be zeros,

$$L = [X_s; \cdots; X_s], S = \mathbf{0}, \tau = \overrightarrow{0}. \tag{16}$$

We now start to solve the three subproblems in (14).

Update of τ

The first subproblem aims at solving the following series of nonlinear equations of τ_j,

$$L_j^{t-1}\circ\tau_j = X_j - S_j^{t-1}, j = 1, \cdots, n. \tag{17}$$

Albeit directly solving the above equation is difficult due to its strong nonlinearity, we can approximate the geometric transformation $L_j^{t-1}\circ\tau_j$ by using piece-wise linear transformations, where each piece corresponds to a small change of τ_j defined by $\Delta\tau_j$. Thus the solution of (17) can be approximated by accumulating a series of $\Delta\tau_j$. This can be viewed as an inner loop included in the update of τ. Thus we have linear approximation

$$L_j^{t-1}\circ(\tau_j + \Delta\tau_j) \approx L_j^{t-1}\circ\tau_j + \Delta\tau_j J_j, \tag{18}$$

where J_j is the Jacobian of $L_j^{t-1} \circ \tau_j$ with respect to the transformation parameters in τ_j. Therefore, by substituting (18) into (17), $\Delta\tau_j$ in each linear piece can be solved as

$$\Delta\tau_j = \left(X_j - S_j^{t-1} - L_j^{t-1} \circ \tau_j\right)(J_j)^{\dagger}. \tag{19}$$

The update of τ_j starts from some initial τ_j, and iteratively solves the overdetermined linear equation (19) with update $\tau_j := \tau_j + \Delta\tau_j$ until the difference between the left hand side and the right hand side of (17) is sufficiently small. It is critical to emphasize that a well selected initial value of τ_j can significantly save computational time. Based on the between-frame affinity, we initialize τ_j by the transformation of its adjacent frame that is closer to the template frame s,

$$\tau_j := \begin{cases} \tau_{j+1}, & j < s; \\ \tau_{j-1}, & j > s. \end{cases} \tag{20}$$

Another important support set constraint, $\mathrm{supp}(L \circ \tau) \subseteq \mathrm{supp}(X)$, needs to be considered in calculating $L_j^{t-1} \circ \tau_j$ during the update of τ. This constraint ensures that the object flows or segmented motions obtained by SST always belong to the sparse part achieved from the background modeling, and thus rules out the noise in background. Hence, suppose the complement set of $\mathrm{supp}(X_j)$ to be $\mathrm{supp}_c(X_j)$, each calculation of $L_j^{t-1} \circ \tau_j$ follows a screening such that,

$$\left(L_j^{t-1} \circ \tau_j\right)_{\mathrm{supp}_c(X_j)} = \overrightarrow{0}. \tag{21}$$

Update of L

The second subproblem has the following global solution that can be updated by BRP based low-rank approximation (13) and its power scheme modification,

$$L^t = \sum_{i=1}^{r} \lambda_i U_i V_i^T, \mathrm{svd}\left(\left(X - S^{t-1}\right) \circ \tau^{-1}\right) = U\Lambda V^T, \tag{22}$$

wherein τ^{-1} denotes the inverse transformation towards τ. The SVDs can be accelerated by BRP based low-rank approximation (4). Another acceleration trick is based on the fact that most columns of $\left(X - S^{t-1}\right) \circ \tau^{-1}$ are nearly all-zeros. This is because the object flow or motion after transformation occupies a very small area of the whole frame. Therefore, The update of L^t can be reduced to low-rank approximation of a submatrix of $\left(X - S^{t-1}\right) \circ \tau^{-1}$ that only includes dense columns. Since the number of dense columns is far less than p, the update of L^t can become much faster.

Update of S

The third subproblem has a global solution that can be obtained via soft-thresholding $\mathcal{P}_\lambda(\cdot)$ similar to the update of S in semi-soft GoDec,

$$S^t = \mathcal{P}_\lambda\left(X - L^t \circ \tau^t\right). \tag{23}$$

A support set constraint $\mathrm{supp}(S) \subseteq \mathrm{supp}(X)$ should be considered in the update of S as well. Hence the above update follows a postprocessing,

$$S^t_{j,\mathrm{supp}_c(X_j)} = \overrightarrow{0}, j = 1, \cdots, n. \tag{24}$$

Note the transformation computation $\circ$ in the update can be accelerated by leveraging the sparsity of the motions. Specifically, the sparsity allows SST to only compute the transformed positions of the nonzero pixels. We summarize the SST algorithm in Algorithm 3.

Algorithm 3 SST Algorithm

Input: $X, r_i, \lambda_i (i = 1, \cdots, n), k$
Output: $L_i (i = 1, \cdots, n), S$
for $i = 1 \to k$ **do**
  Initialize: $s = \arg\max_i \mathrm{card}\,(X_i)$,
  $L = [X_s; \cdots; X_s], S = \mathbf{0}, \tau = \overrightarrow{0}$
  while not converge **do**
    for $j = s - 1 : -1 : 1$ **do**
      $\tau_j := \tau_{j+1}$.
      while not converge **do**
        $\tilde{L}_j^{t-1} = L_j^{t-1} \circ \tau_j, \tilde{L}^{t-1}_{j,\mathrm{supp}_c(X_j)} = \overrightarrow{0}$.
        $\tau_j := \tau_j + \left(X_j - S_j^{t-1} - \tilde{L}_j^{t-1}\right)(J_j)^{\dagger}$.
      end while
    end for
    for $j = s + 1 : 1 : n$ **do**
      $\tau_j := \tau_{j-1}$.
      while not converge **do**
        $\tilde{L}_j^{t-1} = L_j^{t-1} \circ \tau_j, \tilde{L}^{t-1}_{j,\mathrm{supp}_c(X_j)} = \overrightarrow{0}$.
        $\tau_j := \tau_j + \left(X_j - S_j^{t-1} - \tilde{L}_j^{t-1}\right)(J_j)^{\dagger}$.
      end while
    end for
    $\tau^t = \tau$.
    $L^t = \mathrm{BRP}\left(\left(X - S^{t-1}\right) \circ \tau^{-1}\right)$.
    $S^t = \mathcal{P}_\lambda\left(X - L^t \circ \tau^t\right), S^t_{j,\mathrm{supp}_c(X_j)} = \overrightarrow{0}$.
  end while
  $X := S^t, L(i) := L^t, \tau(i) = \tau^t$.
end for

4 Experiments on Surveillance Videos

This section justifies both the effectiveness and the efficiency of the SST framework, which includes semi-soft GoDec and SST algorithm as its two steps, via tracking object flows in four surveillance video sequences [1]. We run all the experiments by MATLAB on a server with dual quad-core 3.33 GHz Intel Xeon processors and 32 GB RAM. In the experiments, the type of geometric transformation τ is simply selected as translation. The detection, tracking and segmentation results as well as associated time costs are shown in Figure 1 and Figure 2.

The results show SST can successfully separate the moving objects from the background via semi-soft GoDec, and promisingly recover both the low-rank patterns and the associated geometric transformations for motions of multiple object flows. The detection, tracking and segmentation are seamlessly unified in a matrix factorization framework and achieved with high accuracy. Moreover, it also verifies that SST performs significantly robust on complicated motions in complex scenes. This is attributed to their distinguishing shifted low-rank patterns, because different object flows can hardly share a subspace after the same geometric transformation. Since SST show stable and appealing performance in motion detection, tracking and segmentation for either crowd

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

Figure 1: Background modeling and object flow tracking results of a 50-frame surveillance video sequence from Hall dataset with resolution 144×176.

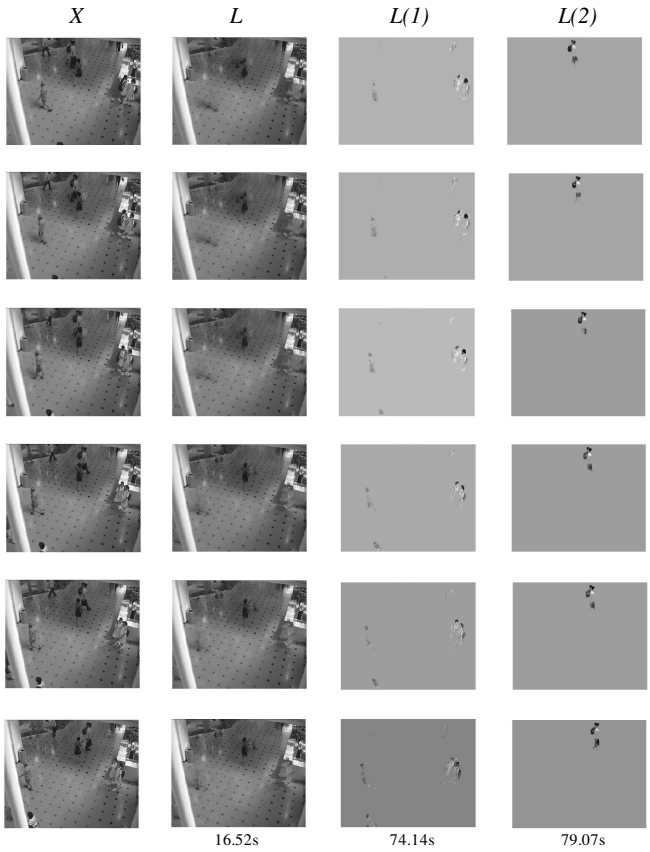

Figure 2: Background modeling and object flow tracking results of a 50-frame surveillance video sequence from Shoppingmall dataset with resolution 256×320.

or individual, it provides a more semantic and intelligent analysis to the video content than existing methods.

Table 1: CPU seconds of GoDec and semi-soft GoDec (SSGoDec) on 200-frame videos.

Pixels	25344	20480	19200	81920
GoDec	47.38	39.75	36.84	203.72
SSGoDec	8.75	6.52	6.23	24.15

Besides the precise video content analysis, another advantage of SST is its fast speed. In Table 1, we compare semi-soft GoDec with original GoDec on the 4 video sequences used in [Zhou and Tao, 2011] for background modeling task. Semi-soft GoDec can processes a 200 frame video with 144×176 resolution in 9 seconds, which is substantially faster than all the published results of RPCA approaches. In Table 2, we list the corresponding choices of cardinality k in GoDec and soft-threshold λ in semi-soft GoDec. Their resulting relative errors for reconstruction are both within $[10^{-3}, 10^{-2}]$. It shows that for GoDec we should carefully select k, otherwise the noise will leak into the sparse part or some meaningful sparse outlier will be lost. But we can easily tune λ in semi-soft GoDec (we fix it to 8 in all experiments). For flow tracking, we speed up SST algorithm by leveraging the motion sparsity, between-frame affinity, and randomized approximation. Therefore, SST is very competitive in big data problems.

Table 2: Sparsity parameters of GoDec and semi-soft GoDec (SSGoDec) on 200-frame videos.

Pixels	25344	20480	19200	81920
GoDec (k)	310000	65000	540000	1800000
SSGoDec (λ)	8	8	8	8

Acknowledgements

We would like to thank all the anonymous reviewers for their constructive comments on improving this paper. This work is supported by Australian Research Council Discovery Project with number ARC DP-120103730.

References

[Ali and Shah, 2007] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and

stability analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[Candès and Recht, 2008] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2008.

[Candès *et al.*, 2009] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 2009.

[Chandrasekaran *et al.*, 2009] V. Chandrasekaran, S. Sanghavi, S. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. `arXiv: 0906.2220`, 2009.

[Chen *et al.*, 2010] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, 2010.

[Comaniciu *et al.*, 2003] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 2003.

[Doucet *et al.*, 2001] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[F. Nie, 2011] C. Ding D. Luo H. Wang F. Nie, H. Huang. Robust principal component analysis with non-greedy l1-norm maximization. In *International Joint Conference on Artificial Intelligence*, 2011.

[Fragkiadaki and Shi, 2011] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2073–2080, 2011.

[Galasso *et al.*, 2012] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *Asian Conference on Computer Vision*, 2012.

[Hess and Fern, 2009] R. Hess and A. Fern. Discriminatively trained particle filters for complex multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[Hong *et al.*, 2012] Z. Hong, X. Mei, and D. Tao. Dual-force metric learning for robust distracter-resistant tracker. In *European Conference on Computer Vision*, 2012.

[Hsu *et al.*, 2011] D. Hsu, S. Kakade, and T. Zhang. Robust matrix decomposition with sparse corruptions. *IEEE Transactions on Information Theory*, 2011.

[Ji and Ye, 2009] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *The 26th International Conference on Machine Learning (ICML)*, pages 457–464, 2009.

[Lewis and Malick, 2008] A. S. Lewis and J. Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, 2008.

[Prince, 2011] Simon J.D. Prince. *Computer vision: models, learning and inference*. Cambridge University Press, 2011.

[Wu *et al.*, 2011] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *International Conference on Computer Vision*, pages 1419–1426, 2011.

[Xiong *et al.*, 2010] Liang Xiong, Xi Chen, and Jeff Schneider. Direct robust matrix factorization for anomaly detection. In *The 11th International Conference on Data Mining (ICDM)*, 2010.

[Yu *et al.*, 2008] Q. Yu, T. B. Dinh, and G. Medioni. Online tracking and reacquistion using co-trained generative and discriminative trackers. In *European Conference on Computer Vision*, pages 678–691, 2008.

[Zhou and Tao, 2011] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *International Conference on Machine Learning*, 2011.

[Zhou and Tao, 2012] T. Zhou and D. Tao. Bilateral random projections. In *International Symposium on Information Theory*, 2012.

[Zhou and Tao, 2013] T. Zhou and D. Tao. Greedy bilateral sketch, completion & smoothing. In *International Conference on Artificial Intelligence and Statistics*, 2013.

[Zhou *et al.*, 2010] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma. Stable principal component pursuit. In *IEEE International Symposium on Information Theory*, 2010.