Robust PCA: Optimization of the Robust Reconstruction Error over the Stiefel Manifold

Anastasia Podosinnikova[1], Simon Setzer[2], and Matthias Hein[2]

[1]INRIA – Sierra Project-Team, École Normale Supérieure, Paris, France
[2]Computer Science Department, Saarland University, Saarbrücken, Germany

Abstract. It is well known that Principal Component Analysis (PCA) is strongly affected by outliers and a lot of effort has been put into robustification of PCA. In this paper we present a new algorithm for robust PCA minimizing the trimmed reconstruction error. By directly minimizing over the Stiefel manifold, we avoid deflation as often used by projection pursuit methods. In distinction to other methods for robust PCA, our method has no free parameter and is computationally very efficient. We illustrate the performance on various datasets including an application to background modeling and subtraction. Our method performs better or similar to current state-of-the-art methods while being faster.

1 Introduction

PCA is probably the most common tool for exploratory data analysis, dimensionality reduction and clustering, e.g., [11]. It can either be seen as finding the best low-dimensional subspace approximating the data or as finding the subspace of highest variance. However, due to the fact that the variance is not robust, PCA can be strongly influenced by outliers. Indeed, even one outlier can change the principal components (PCs) drastically. This phenomenon motivates the development of robust PCA methods which recover the PCs of the uncontaminated data. This problem received a lot of attention in the statistical community and recently became a problem of high interest in machine learning.

In the statistical community, two main approaches to robust PCA have been proposed. The first one is based on the robust estimation of the covariance matrix, e.g., [5], [10]. Indeed, having found a robust covariance matrix one can determine robust PCs by performing the eigenvalue decomposition of this matrix. However, it has been shown that robust covariance matrix estimators with desirable properties, such as positive semidefiniteness and affine equivariance, have a breakdown point[1] upper bounded by the inverse of the dimensionality [5]. The second approach is the so called projection-pursuit [9], [13], where one maximizes a robust scale measure, instead of the standard deviation, over all possible directions. Although, these methods have the best possible breakdown point of 0.5,

[1] The breakdown point [10] of a statistical estimator is informally speaking the fraction of points which can be arbitrarily changed and the estimator is still well defined.

they lead to non-convex, typically, non-smooth problems and current state-of-the-art are greedy search algorithms [4], which show poor performance in high dimensions. Another disadvantage is that robust PCs are computed one by one using deflation techniques [14], which often leads to poor results for higher PCs.

In the machine learning and computer vision communities, matrix factorization approaches to robust PCA were mostly considered, where one looks for a decomposition of a data matrix into a low-rank part and a sparse part, e.g., [3], [15], [16], [22]. The sparse part is either assumed to be scattered uniformly [3] or it is assumed to be row-wise sparse corresponding to the model where an entire observation is corrupted and discarded. While some of these methods have strong theoretical guarantees, in practice, they depend on a regularization parameter which is non-trivial to choose as robust PCA is an unsupervised problem and default choices, e.g., [3], [16], often do not perform well as we discuss in Section 4. Furthermore, most of these methods are slow as they have to compute the SVD of a matrix of the size of the data matrix at each iteration.

As we discuss in Section 2, our formulation of robust PCA is based on the minimization of a robust version of the reconstruction error over the Stiefel manifold, which induces orthogonality of robust PCs. This formulation has multiple advantages. First, it has the maximal possible breakdown point of 0.5 and the interpretation of the objective is very simple and requires no parameter tuning in the default setting. In Section 3, we propose a new fast TRPCA algorithm for this optimization problem. Our algorithm computes both orthogonal PCs and a robust center, hence, avoiding the deflation procedure and preliminary robust centering of data. While our motivation is similar to the one of [15], our optimization scheme is completely different. In particular, our formulation requires no additional parameter.

2 Robust PCA

Notation. All vectors are column vectors and $I_p \in \mathbb{R}^{p\times p}$ denotes the identity matrix. We are given data $X \in \mathbb{R}^{n\times p}$ with n observations in $\mathbb{R}^p$ (rows correspond to data points). We assume that the data contains t true observations $T \in \mathbb{R}^{t\times p}$ and $n-t$ outliers $O \in \mathbb{R}^{n-t\times p}$ such that $X = T \cup O$ and $T \cap O \neq \varnothing$. To be able to distinguish true data from outliers, we require the standard in robust statistics assumption, that is $t \geq \lceil \frac{n}{2} \rceil$. The Stiefel manifold is denoted as $\mathcal{S}_k = \left\{U \in \mathbb{R}^{p\times k} \mid U^\top U = I\right\}$ (the set of orthonormal k-frames in $\mathbb{R}^p$).

PCA. Standard PCA [11] has two main interpretations. One can either see it as finding the k-dimensional subspace of maximum variance in the data or the k-dimensional affine subspace with minimal reconstruction error. In this paper we are focusing on the second interpretation. Given data $X \in \mathbb{R}^{n\times p}$, the goal is to find the offset $m \in \mathbb{R}^p$ and k principal components $(u_1, \ldots, u_k) = U \in \mathcal{S}_k$, which describe $\mathcal{A}(m, U) = \left\{z \in \mathbb{R}^p \;\middle|\; z = m + \sum_{j=1}^k s_j u_j,\ s_j \in \mathbb{R}\right\}$, the k-dimensional

affine subspace, so that they minimize the reconstruction error

$$\left\{\hat{m}, \hat{U}\right\} = \underset{m \in \mathbb{R}^p,\, U \in \mathcal{S}_k,\, z_i \in \mathcal{A}(m,U)}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \|z_i - x_i\|_2^2 \,. \tag{1}$$

It is well known that $\hat{m} = \frac{1}{n} \sum_{i=1}^{n} x_i$, and the optimal matrix $\hat{U} \in \mathcal{S}_k$ is generated by the top k eigenvectors of the empirical covariance matrix. As $U \in \mathcal{S}_k$ is an orthogonal projection, an equivalent formulation of (1) is given by

$$\left\{\hat{m}, \hat{U}\right\} = \underset{m \in \mathbb{R}^p,\, U \in \mathcal{S}_k}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left\| \left(UU^\top - I\right)(x_i - m) \right\|_2^2 . \tag{2}$$

Robust PCA. When the data X does not contain outliers ($X = T$), we refer to the outcome of standard PCA, e.g., (2), computed for the true data T as $\{\hat{m}_T, \hat{U}_T\}$. When there are some outliers in the data X, i.e. $X = T \cup O$, the result $\{\hat{m}, \hat{U}\}$ of PCA can be significantly different from $\{\hat{m}_T, \hat{U}_T\}$ computed for the true data T. The reason is the non-robust squared ℓ_2-norm involved in the formulation, e.g., [5], [10]. It is well known that PCA has a breakdown point of zero, that is a single outlier can already distort the components arbitrarily. As outliers are frequently present in applications, robust versions of PCA are crucial for data analysis with the goal of recovering the true PCA solution $\{\hat{m}_T, \hat{U}_T\}$ from the contaminated data X.

As opposed to standard PCA, robust formulations of PCA based on the maximization of the variance (the projection-pursuit approach as extension of (1)), eigenvectors of the empirical covariance matrix (construction of a robust covariance matrix), or the minimization of the reconstruction error (as extension of (2)) are not equivalent. Hence, there is no universal approach to robust PCA and the choice can depend on applications and assumptions on outliers. Moreover, due to the inherited non-convexity of standard PCA, they lead to NP-hard problems. The known approaches for robust PCA either follow to some extent greedy/locally optimal optimization techniques, e.g., [4], [13], [19], [21], or compute convex relaxations, e.g., [3], [15], [16], [22].

In this paper we aim at a method for robust PCA based on the minimization of a robust version of the reconstruction error and adopt the classical outlier model where entire observations (corresponding to rows in the data matrix X) correspond to outliers. In order to introduce the trimmed reconstruction error estimator for robust PCA, we employ the analogy with the least trimmed squares estimator [17] for robust regression. We denote by $r_i(m, U) = \left\| \left(UU^\top - I\right)(x_i - m) \right\|_2^2$ the reconstruction error of observation x_i for the given affine subspace parameterized by (m, U). Then the trimmed reconstruction error is defined to be the sum of the t-smallest reconstruction errors $r_i(m, U)$,

$$R(m, U) = \frac{1}{t} \sum_{i=1}^{t} r_{(i)}(m, U), \tag{3}$$

where $r_{(1)}(m, U) \leq \cdots \leq r_{(n)}(m, U)$ are in nondecreasing order and t, with $\left\lceil \frac{n}{2} \right\rceil \leq t \leq n$, should be a lower bound on the number of true examples T. If

such an estimate is not available as it is common in unsupervised learning, one can set by default $t = \lceil \frac{n}{2} \rceil$. With the latter choice it is straightforward to see that the corresponding PCA estimator has the maximum possible breakdown point of 0.5, that is up to 50% of the data points can be arbitrarily corrupted. With the default choice our method has no free parameter except the rank k.

The minimization of the trimmed reconstruction error (3) leads then to a simple and intuitive formulation of robust PCA

$$\{m^*, U^*\} = \operatorname*{arg\,min}_{m \in \mathbb{R}^p,\, U \in \mathcal{S}_k} R(m, U) = \operatorname*{arg\,min}_{m \in \mathbb{R}^p,\, U \in \mathcal{S}_k} \frac{1}{t} \sum_{i=1}^{t} r_{(i)}(m, U). \tag{4}$$

Note that the estimation of the subspace U and the center m is done jointly. This is in contrast to [3], [4], [13], [16], [21], [22], where the data has to be centered by a separate robust method which can lead to quite large errors in the estimation of the true PCA components. The same criterion (4) has been proposed by [15], see also [23] for a slightly different version. While both papers state that the direct minimization of (4) would be desirable, [15] solve a relaxation of (4) into a convex problem while [23] smooth the problem and employ deterministic annealing. Both approaches introduce an additional regularization parameter controlling the number of outliers. It is non-trivial to choose this parameter.

3 TRPCA: Minimizing Trimmed Reconstruction Error on the Stiefel Manifold

In this section, we introduce TRPCA, our algorithm for the minimization of the trimmed reconstruction error (4). We first reformulate the objective of (4) as it is neither convex, nor concave, nor smooth, even if m is fixed. While the resulting optimization problem is still non-convex, we propose an efficient optimization scheme on the Stiefel manifold with monotonically decreasing objective. Note that all proofs of this section can be found in the supplementary material [18].

3.1 Reformulation and First Properties

The reformulation of (4) is based on the following simple identity. Let $\widetilde{x}_i = x_i - m$ and $U \in \mathcal{S}_k$, then

$$r_i(m, U) = \left\| \left(UU^\top - I\right)(x_i - m) \right\|_2^2 = -\left\| U^\top \widetilde{x}_i \right\|_2^2 + \|\widetilde{x}_i\|_2^2 := \widetilde{r}_i(m, U). \tag{5}$$

The equality holds only on the Stiefel manifold. Let $\widetilde{r}_{(1)}(m, U) \leq \ldots \leq \widetilde{r}_{(n)}(m, U)$, then we get the alternative formulation of (4),

$$\{m^*, U^*\} = \operatorname*{arg\,min}_{m \in \mathbb{R}^p,\, U \in \mathcal{S}} \widetilde{R}(m, U) = \frac{1}{t} \sum_{i=1}^{t} \widetilde{r}_i(m, U). \tag{6}$$

While (6) is still non-convex, we show in the next proposition that for fixed m the function $\widetilde{R}(m, U)$ is concave on $\mathbb{R}^{p \times k}$. This will allow us to employ a simple optimization technique based on linearization of this concave function.

Proposition 1. *For fixed $m \in \mathbb{R}^p$ the function $\widetilde{R}(m,U) : \mathbb{R}^{p\times k} \to \mathbb{R}$ defined in (6) is concave in U.*

Proof. We have $\widetilde{r}_i(m,U) = -\left\|U^\top \widetilde{x}_i\right\|_2^2 + \|\widetilde{x}_i\|_2^2$. As $\left\|U^\top \widetilde{x}_i\right\|^2$ is convex, we deduce that $\widetilde{r}_i(m,U)$ is concave in U. The sum of the t smallest concave functions out of $n \geq t$ concave functions is concave, as it can be seen as the pointwise minimum of all possible $\binom{n}{t}$ sums of t of the concave functions, e.g., [2].

The iterative scheme uses a linearization of $\widetilde{R}(m,U)$ in U. For that we need to characterize the superdifferential of the concave function $\widetilde{R}(m,U)$.

Proposition 2. *Let m be fixed. The superdifferential $\partial\widetilde{R}(m,U)$ of $\widetilde{R}(m,U) : \mathbb{R}^{p\times k} \to \mathbb{R}$ is given as*

$$\partial\widetilde{R}(m,U) = \Big\{ \sum_{i\in I} \alpha_i (x_i - m)(x_i - m)^\top U \,\Big|\, \sum_{i=1}^n \alpha_i = t,\ 0 \leq \alpha_i \leq 1 \Big\}, \quad (7)$$

where $I = \{i \,|\, \widetilde{r}_i(m,U) \leq \widetilde{r}_{(t)}(m,U)\}$ with $\widetilde{r}_{(1)}(m,U) \leq \ldots \leq \widetilde{r}_{(n)}(m,U)$.

Proof. We reduce it to a well known case. We can write $\widetilde{R}(m,U)$ as

$$\widetilde{R}(m,U) = \min_{0\leq\alpha_i\leq 1,\ i=1,\ldots,n,\ \sum_{i=1}^n \alpha_i = t} \ \sum_{i=1}^n \alpha_i \widetilde{r}_i(m,U), \quad (8)$$

that is a minimum of a parameterized set of concave functions. As the parameter set is compact and continuous (see Theorem 4.4.2 in [7]), we have

$$\partial\widetilde{R}(m,U) = \operatorname{conv}\Big(\bigcup_{\alpha^j \in I(U)} \partial\big(\sum_{i=1}^n \alpha_i^j \widetilde{r}_i(m,U) \big) \Big) = \operatorname{conv}\Big(\bigcup_{\alpha^j \in I(U)} \sum_{i=1}^n \alpha_i^j \partial\widetilde{r}_i(m,U) \Big), \quad (9)$$

where $I(U) = \{\alpha \,|\, \sum_{i=1}^n \alpha_i \widetilde{r}_i(m,U) = \widetilde{R}(m,U),\ \sum_{i=1}^n \alpha_i = t,\ 0 \leq \alpha_i \leq 1,\ i = 1,\ldots,n\}$ and $\operatorname{conv}(S)$ denotes the convex hull of S. Finally, using that $\widetilde{r}_i(m,U)$ is differentiable with $\partial\widetilde{r}_i(m,U) = \{(x_i - m)(x_i - m)^\top U\}$ yields the result.

3.2 Minimization Algorithm

Algorithm 1 for the minimization of (6) is based on block-coordinate descent in m and U. For the minimization in U we use that $\widetilde{R}(m,U)$ is concave for fixed m. Let $G \in \partial\widetilde{R}(m,U^k)$, then by definition of the supergradient of a concave function,

$$\widetilde{R}\left(m, U^{k+1}\right) \leq \widetilde{R}\left(m, U^k\right) + \left\langle G, U^{k+1} - U^k \right\rangle. \quad (10)$$

The minimization of the linear upper bound on the Stiefel manifold can be done in closed form, see Lemma 1 below. For that we use a modified version of a result of [12]. Before giving the proof, we introduce the polar decomposition of a

matrix $G \in \mathbb{R}^{p\times k}$ which is defined to be $G = QP$, where $Q \in \mathcal{S}$ is an orthonormal matrix of size $p \times k$ and P is a symmetric positive semidefinite matrix of size $k \times k$. We denote the factor Q of G by $Polar(G)$. The polar can be computed in $\mathcal{O}(pk^2)$ for $p \geq k$ [12] as $Polar(G) = UV^\top$ (see Theorem 7.3.2. in [8]) using the SVD of G, $G = U\Sigma V^\top$. However, faster methods have been proposed, see [6], which do not even require the computation of the SVD.

Lemma 1. *Let $G \in \mathbb{R}^{p\times k}$, with $k \leq p$, and denote by $\sigma_i(G)$, $i = 1, \ldots, k$, the singular values of G. Then $\min_{U\in\mathcal{S}_k} \langle G, U\rangle = -\sum_{i=1}^k \sigma_i(G)$, with minimizer $U^* = -Polar(G)$. If G is of full rank, then $Polar(G) = G(G^\top G)^{-1/2}$.*

Proof. Let $G = U\Sigma V^\top$ be the SVD of G, that is $U \in O(p)$, $V \in O(k)$, where $O(m)$ denotes the set of orthogonal matrices in $\mathbb{R}^m$,

$$\min_{O\in\mathcal{S}_k} \langle G, O\rangle = \min_{O\in\mathcal{S}_k} \left\langle \Sigma, U^\top OV\right\rangle = \min_{W\in\mathcal{S}_k} \sum_{i=1}^k \sigma_i(G)W_{ii} \geq -\sum_{i=1}^k \sigma_i(G). \quad (11)$$

The lower bound is realized by $-UV^\top \in \mathcal{S}_k$ which is equal to $-Polar(G)$. We have, $-\left\langle U\Sigma V^\top, UV^\top\right\rangle = -\text{trace}(\Sigma) = -\sum_{i=1}^k \sigma(G)_i$. The final statement follows from the proof of Theorem 7.3.2. in [8].

Algorithm 1 TRPCA

Input: X, t, d, $U^0 \in \mathcal{S}$, and m^0 median of X, tolerance ε
Output: robust center m^k and robust PCs U^k
repeat for $k = 1, 2, \ldots$
  Center data $\widetilde{X}^k = \left\{\widetilde{x}_i^k = x_i - m^k,\ i = 1, \ldots, n\right\}$
  Compute supergradient $\mathcal{G}(U^k)$ of $\widetilde{R}(m^k, U^k)$ for fixed m^k
  Update $U^{k+1} = -Polar\left(\mathcal{G}(U^k)\right)$
  Update $m^{k+1} = \frac{1}{t}\sum_{i\in\mathcal{I}^{k'}} x_i$, where $\mathcal{I}^{k'}$ are the indices of the t smallest $\widetilde{r}_i(m^k, U^{k+1})$, $i = 1, \ldots, n$
until relative descent below ε

Given that U is fixed, the center m can be updated simply as the mean of the points realizing the current objective of (6), that is the points realizing the t-smallest reconstruction error. Finally, although the objective of (6) is neither convex nor concave in m, we prove monotonic descent of Algorithm 1.

Theorem 1. *The following holds for Algorithm 1. At every iteration, either $\widetilde{R}(m^{k+1}, U^{k+1}) < \widetilde{R}(m^k, U^k)$ or the algorithm terminates.*

Proof. Let m^k be fixed and $G(U^k) \in \partial\widetilde{R}(m, U^k)$, then from (10) we have

$$\widetilde{R}(m^k, U) \leq \widetilde{R}(m, U^k) - \left\langle G(U^k), U^k\right\rangle + \left\langle G(U^k), U\right\rangle. \quad (12)$$

The minimizer $U^{k+1} = \underset{U \in \mathcal{S}_k}{\arg\min} \left\langle G(U^k), U \right\rangle$, over the Stiefel manifold can be computed via Lemma 1 as $U^{k+1} = -Polar(G(U^k))$. Thus we get immediately,

$$\widetilde{R}(m^k, U^{k+1}) \leq \widetilde{R}(m^k, U^k).$$

After the update of U^{k+1} we compute $\mathcal{I}^{k'}$ which are the indices of the t smallest $\widetilde{r}_i(m^k, U^{k+1})$, $i = 1, \ldots, n$. If there are ties, then they are broken randomly. For fixed U^{k+1} and fixed $\mathcal{I}^{k'}$ the minimizer of the objective

$$\sum_{i \in \mathcal{I}^{k'}} - \left\| (U^{k+1})^\top (x_i - m) \right\|_2^2 + \left\| x_i - m \right\|_2^2, \tag{13}$$

is given by $m^{k+1} = \frac{1}{t} \sum_{i \in \mathcal{I}^{k'}} x_i$, which yields, $\sum_{i \in \mathcal{I}^{k'}} \widetilde{r}_i(m^{k+1}, U^{k+1}) \leq \widetilde{R}(m^k, U^{k+1})$. After the computation of m^{k+1}, $\mathcal{I}^{k'}$ need no longer correspond to the t smallest reconstruction errors $\widetilde{r}_i(m^{k+1}, U^{k+1})$. However, taking the t smallest ones only further reduces the objective, $\widetilde{R}(m^{k+1}, U^{k+1}) \leq \sum_{i \in \mathcal{I}^{k'}} \widetilde{r}_i(m^{k+1}, U^{k+1})$. This yields finally the result, $\widetilde{R}(m^{k+1}, U^{k+1}) \leq \widetilde{R}(m^k, U^k)$.

The objective is non-smooth and neither convex nor concave. The Stiefel manifold is a non-convex constraint set. These facts make the formulation of critical points conditions challenging. Thus, while potentially stronger convergence results like convergence to a critical point are appealing, they are currently out of reach. However, as we will see in Section 4, Algorithm 1 yields good empirical results, even beating state-of-the-art methods based on convex relaxations or other non-convex formulations.

3.3 Complexity and Discussion

The computational cost of each iteration of Algorithm 1 is dominated by $\mathcal{O}(pk^2)$ for computing the polar and $\mathcal{O}(pkn)$ for a supergradient of $\widetilde{R}(m, U)$ and, thus, has total cost $\mathcal{O}(pk(k+n))$. We compare this to the cost of the proximal method in [3], [20] for minimizing $\min_{X=A+E} \|A\|_* + \lambda \|E\|_1$. In each iteration, the dominating cost is $\mathcal{O}(\min\{pn^2, np^2\})$ for the SVD of a matrix of size $p \times n$. If the natural condition $k \ll \min\{p, n\}$ holds, we observe that the computational cost of TRPCA is significantly better. Thus even though we do 10 random restarts with different starting vectors, our TRPCA is still faster than all competing methods, which can also be seen from the runtimes in Table 1.

In [15], a relaxed version of the trimmed reconstruction error is minimized:

$$\min_{m \in \mathbb{R}^p,\, U \in S_k,\, s \in \mathbb{R}^k} \left\| X - \mathbf{1}_n m^\top - Us - O \right\|_F^2 + \lambda \left\| O \right\|_{2,1}, \tag{14}$$

where $\|O\|_{2,1}$ is added in order to enforce row-wise sparsity of O. The optimization is done via an alternating scheme. However, the disadvantage of this formulation is that it is difficult to adjust the number of outliers via the choice of λ and thus requires multiple runs of the algorithm to find a suitable range, whereas in our formulation the number of outliers $n-t$ can be directly controlled by the user or t can be set to the default value $\lceil \frac{n}{2} \rceil$.

4 Experiments

We compare our TRPCA (the code is available for download at [18]) algorithm with the following robust PCA methods: ORPCA [15], LLD[2] [16], HRPCA [21], standard PCA, and true PCA on the true data T (ground truth). For background subtraction, we also compare our algorithm with PCP [3] and RPCA [19], although the latter two algorithms are developed for a different outlier model.

To get the best performance of LLD and ORPCA, we run both algorithms with different values of the regularization parameters to set the number of zero rows (observations) in the outlier matrix equal to $\tilde{t}$ (which increases runtime significantly). The HRPCA algorithm has the same parameter t as our method.

We write (0.5) in front of an algorithm name if the default value $\tilde{t} = \lceil \frac{n}{2} \rceil$ is used, otherwise, we use the ground truth information $\tilde{t} = |T|$. As performance measure we use the reconstruction error relative to the reconstruction error of the true data (which is achieved by PCA on the true data only):

$$\mathrm{tre}(U, m) = \frac{1}{t} \sum\nolimits_{\{i \mid x_i \in T\}} r_i(m, U) - r_i(\hat{m}_T, \hat{U}_T), \tag{15}$$

where $\{\hat{m}_T, \hat{U}_T\}$ is the true PCA of T and it holds that $tre(U, m) \geq 0$. The smaller $tre(U, m)$, i.e., the closer the estimates $\{m, U\}$ to $\{\hat{m}_T, \hat{U}_T\}$, the better. We choose datasets which are computationally feasible for all methods.

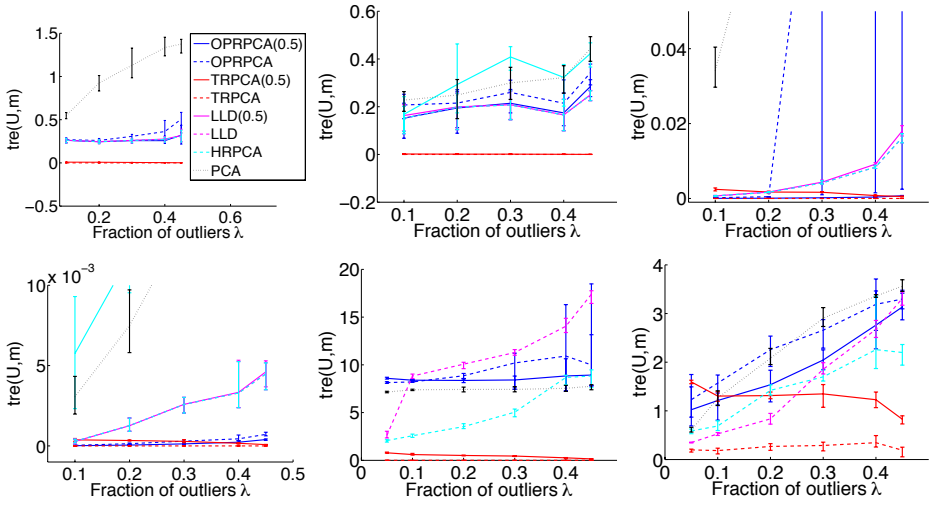

Fig. 1. First row left to right: 1) Data1, $p = 100$, $\sigma_o = 2$; 2) Data1, $p = 20$, $\sigma_o = 2$; 3) Data2, $p = 100$, $\sigma_o = 0.35$; Second row left to right: 1) Data2, $p = 20$, $\sigma_o = 0.35$; 2) USPS10, $k = 1$; 3) USPS10, $k = 10$.

[2] Note, that the LLD algorithm [16] and the OPRPCA algorithm [22] are equivalent.

4.1 Synthetic Data Sets

We sample uniformly at random a subspace of dimension k spanned by $U \in S_k$ and generate the true data $T \in \mathbb{R}^{t\times p}$ as $T = AU^\top + E$ where the entries of $A \in \mathbb{R}^{t\times k}$ are sampled uniformly on $[-1, 1]$ and the noise $E \in \mathbb{R}^{t\times p}$ has Gaussian entries distributed as $\mathcal{N}(0, \sigma_T)$. We consider two types of outliers: (Data1) the outliers $O \in \mathbb{R}^{o\times p}$ are uniform samples from $[0, \sigma_o]^p$, (Data2) the outliers are samples from a random half-space, let w be sampled uniformly at random from the unit sphere and let $x \sim \mathcal{N}(0, \sigma_0 \mathbb{1})$ then an outlier $o_i \in \mathbb{R}^p$ is generated as $o_i = x - \max\{\langle x, w\rangle, 0\}w$. For Data2, we also downscale true data by 0.5 factor. We always set $n = t + o = 200$, $k = 5$, and $\sigma_T = 0.05$ and construct data sets for different fractions of outliers $\lambda = \frac{o}{t+o} \in \{0.1, 0.2, 0.3, 0.4, 0.45\}$. For every λ we sample 5 data sets and report mean and standard deviation of the relative true reconstruction error $\mathrm{tre}(U, m)$.

4.2 Partially Synthetic Data Set

We use USPS, a dataset of 16×16 images of handwritten digits. We use digits 1 as true observations T and digits 0 as outliers O and mix them in different proportions. We refer to this data set as USPS10 and the results can be found in Fig. 1. Another similar experiment is on the MNIST data set of 28×28 images of handwritten digits. We use digits 1 (or 7) as true observations T and all other digits $0, 2, 3, \ldots, 9$ as outliers O (each taken in equal proportion). We mix true data and outliers in different proportions and the results can be found in Fig. 2 (or Fig. 3), where we excluded LLD due to its low computational time, see Tab. 1. We notice that TRPCA algorithm with the parameter value $\tilde{t} = t$ (ground truth information) performs almost perfectly and outperforms all other methods, while the default version of TRPCA with parameter $\tilde{t} = \lceil \frac{n}{2} \rceil$ shows slightly worse performance. The fact that TRPCA estimates simultaneously the robust center m influences positively the overall performance of the algorithm, see, e.g., the experiments for background subtraction and modeling in Section 4.3 and additional ones in the supplementary material. That is Fig. 6-17.

4.3 Background Modeling and Subtraction

In [19] and [3] robust PCA has been proposed as a method for background modeling and subtraction. While we are not claiming that robust PCA is the best method to do this, it is an interesting test for robust PCA. The data X are the image frames of a video sequence. The idea is that slight change in the background leads to a low-rank variation of the data whereas the foreground changes cannot be modeled by this and can be considered as outliers. Thus with the estimates m^* and U^* of the robust PCA methods, the solution of the background subtraction and modeling problem is given as

$$x_i^b = m^* + U^*(U^*)^\top(x_i - m^*) \tag{16}$$

where x_i^b is the background of frame i and its foreground is simply $x_i^f = x_i - x_i^b$.

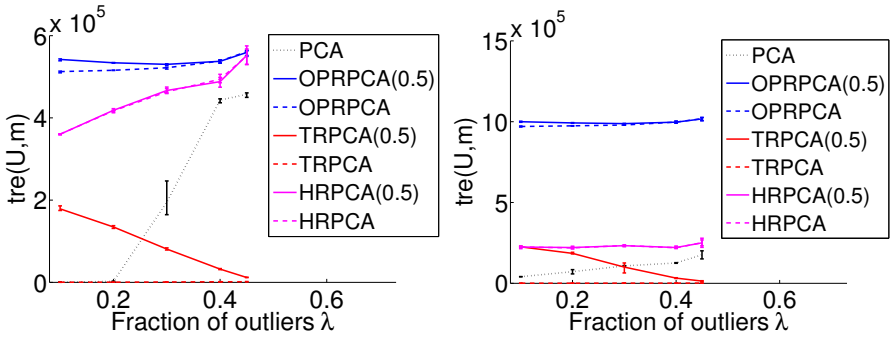

Fig. 2. Experiment on the MNIST data set with digits 1 as true observations T and all other digits $0, 2, 3, \ldots, 9$ as outliers. Number of recovered PCs is $k = 1$ (left) and $k = 5$ (right).

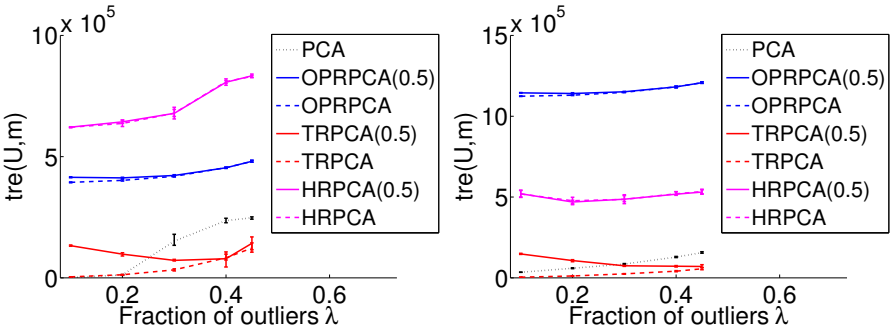

Fig. 3. Experiment on the MNIST data set with digits 7 as true observations T and all other digits $0, 2, 3, \ldots, 9$ as outliers. Number of recovered PCs is $k = 1$ (left) and $k = 5$ (right).

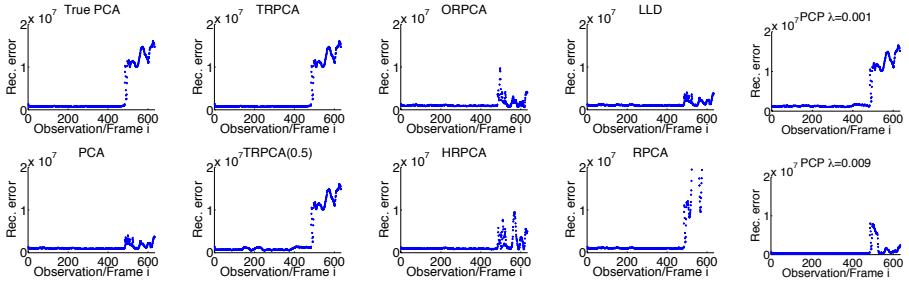

Fig. 4. Reconstruction errors, i.e., $||(x_i - m^*) - U^* (U^*)^\top (x_i - m^*)||_2^2$, on the y-axis, for each frame on the x-axes for $k = 10$. Note that the person is visible in the scene from frame 481 until the end. We consider the background images as true data and, thus, the reconstruction error should be high after frame 481 (when the person enters).

We experimentally compare the performance of all robust PCA methods on the water surface data set [1], which has moving water in its background. We choose this dataset of $n = 633$ frames each of size $p = 128 \times 160 = 20480$ as it is computationally feasible for all the methods. In Fig. 5, we show the background subtraction results of several robust PCA algorithms. We optimized the value λ for PCP of [3], [20] by hand to obtain a good decomposition, see the bottom right pictures of Fig. 5. How crucial the choice of λ is for this method can be seen from the bottom right pictures. Note that the reconstruction error of both the default version of TRPCA and TRPCA(0.5) with ground truth information provide almost perfect reconstruction errors with respect to the true data, cf., Fig. 4. Hence, TRPCA is the only method which recovers the foreground and background without mistakes. We refer to the supplementary material for more explanations regarding this experiment as well as results for another background subtraction data set. The runtimes of all methods for the water surface data set are presented in Table 1, which shows that TRPCA is the fastest of all methods.

Table 1. Runtimes for the water surface data set for the algorithms described in Section 4. For TRPCA/TRPCA(0.5) we report the average time of one initialization (in practice, $5 - 10$ random restarts are sufficient). For PCP we report the runtime for the employed parameter $\lambda = 0.001$. For all others methods, it is the time of one full run of the algorithm including the search for regularization parameters.

	trpca	trpca(.5)	orpca	orpca(.5)	hrpca	hrpca(.5)	lld	rpca	pcp($\lambda = 0.001$)
$k = 1$	7	13	3659	3450	45990	48603	–	1078	–
$k = 3$	99	61	8151	13852	50491	56090	–	730	–
$k = 5$	64	78	2797	3726	72009	77344	232667	3615	875
$k = 7$	114	62	4138	3153	67174	90931	–	4230	–
$k = 9$	119	92	6371	8508	96954	106782	–	4113	–

5 Conclusion

We have presented a new method for robust PCA based on the trimmed reconstruction error. Our efficient algorithm, using fast descent on the Stiefel manifold, works in the default setting ($t = \lceil \frac{n}{2} \rceil$) without any free parameters and is significantly faster than other competing methods. In all experiments TRPCA performs better or at least similar to other robust PCA methods, in particular, TRPCA solves challenging background subtraction tasks.

Acknowledgements. M.H. has been partially supported by the ERC Starting Grant NOLEPRO and M.H. and S.S. have been partially supported by the DFG Priority Program 1324, "Extraction of quantifiable information from complex systems".

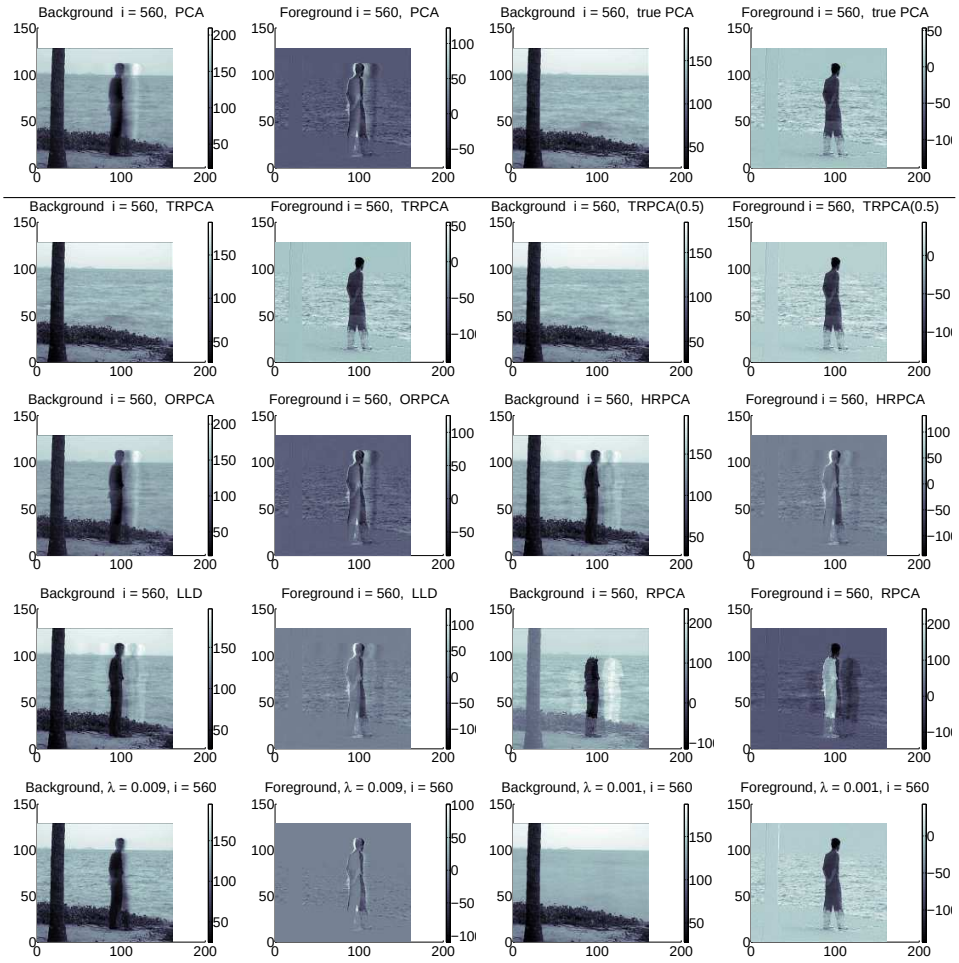

Fig. 5. Backgrounds and foreground for frame $i = 560$ of the water surface data set. The last row corresponds to the PCP algorithm with values of λ set by hand

References

1. Bouwmans, T.: Recent advanced statistical background modeling for foreground detection: A systematic survey. Recent Patents on Computer Science 4(3), 147–176 (2011)
2. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
3. Candès, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM 58(3) (2011)
4. Croux, C., Pilzmoser, P., Oliveira, M.R.: Algorithms for projection–pursuit robust principal component analysis. Chemometrics and Intelligent Laboratory Systems 87, 218–225 (2007)
5. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust Statistics. The Approach Based on Influence Functions. John Wiley and Sons, New York (1986)
6. Higham, N.J., Schreiber, R.S.: Fast polar decomposition of an arbitrary matrix. SIAM Journal on Scientific Computing 11(4), 648–655 (1990)
7. Hiriart-Urruty, J.B., Lemaréchal: Fundamentals of Convex Analysis. Springer, Berlin (2001)
8. Horn, R., Johnson, C.: Matrix Analysis. Cambridge University Press, Cambridge (1990)
9. Huber, P.J.: Projection pursuit. Annals of Statistics 13(2), 435–475 (1985)
10. Huber, P., Ronchetti, E.: Robust Statistics. John Wiley and Sons, New York, 2nd edn. (2009)
11. Jolliffe, I.: Principal Component Analysis. Springer, New York, 2nd edn. (2002)
12. Journée, M., Nesterov, Y., Richtárik, P., Sepulchre, R.: Generalized power method for sparse principal component analysis. Journal of Machine Learning Research 1(1), 517–553 (2010)
13. Li, G., Chen, Z.: Projection–pursuit approach to robust dispersion matrices and principal components: Primary theory and Monte Carlo. Journal of the American Statistical Association 80(391), 759–766 (1985)
14. Mackey, L.: Deflation methods for sparse PCA. In: 24th Conference on Neural Information Processing Systems. pp. 1017–1024 (2009)
15. Mateos, G., Giannakis, G.: Robust PCA as bilinear decomposition with outlier-sparsity regularization. IEEE Transactions on Signal Processing 60(10), 5176–5190 (2012)
16. McCoy, M., Tropp, J.A.: Two proposals for robust PCA using semidefinite programming. Electronic Journal of Statistics 5, 1123–1160 (2011)
17. Rousseeuw, P.J.: Least median of squares regression. Journal of the American Statistical Association 79(388), 871–880 (1984)
18. Supplementary material. http://www.ml.uni-saarland.de/code/trpca/trpca.html
19. De la Torre, F., Black, M.: Robust principal component analysis for computer vision. In: 8th IEEE International Conference on Computer Vision. pp. 362–369 (2001)
20. Wright, J., Peng, Y., Ma, Y., Ganesh, A., Rao, S.: Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In: 24th Conference on Neural Information Processing Systems. pp. 2080–2088 (2009)
21. Xu, H., Caramanis, C., Mannor, S.: Outlier-robust PCA: the high dimensional case. IEEE Transactions on Information Theory 59(1), 546–572 (2013)

22. Xu, H., Caramanis, C., Sanghavi, S.: Robust PCA via outlier pursuit. IEEE Transactions on Information Theory 58(5), 3047–3064 (2012)
23. Xu, L., Yuille, A.L.: Robust principal component analysis by self-organizing rules based on statistical physics approach. IEEE Transactions on Neural Networks 6, 131–143 (1995)

6 Supplementary material: Experiments

In this supplementary material we present additional illustrations of the background subtraction experiments in Fig. 4-15. We consider the water surface data set and the moved object[3] data set. For both data sets the frames where no person is present represent the true data T (background) and frames where the person is present are considered as outliers O.

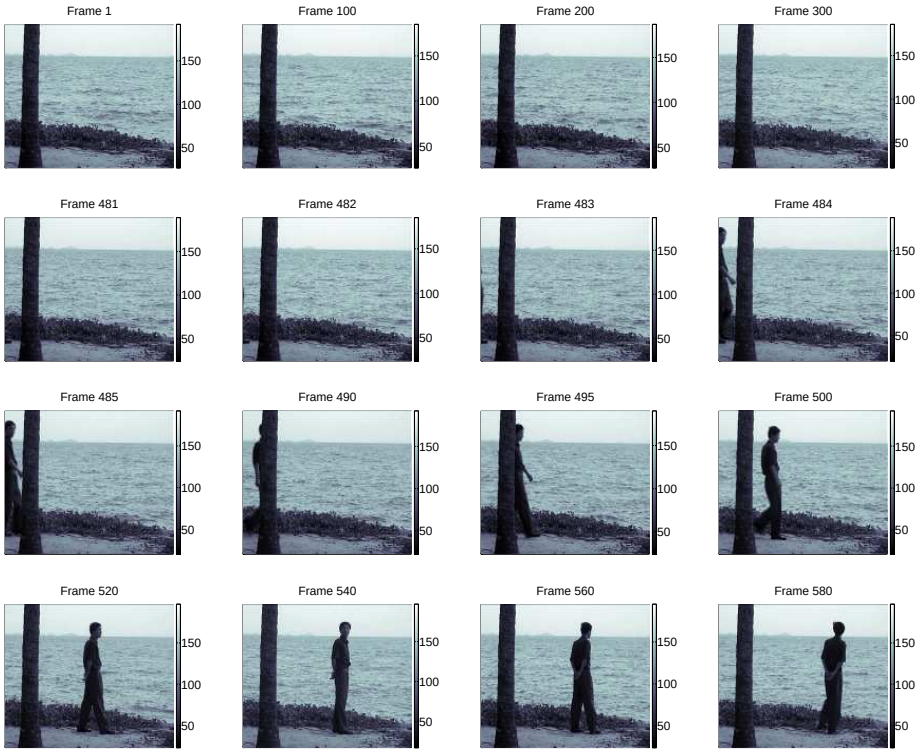

Fig. 6. Examples of the original frames of the water surface data set. Frames from 1 to 481 contain only background (true data) with a moving water surface. The person (considered as outlier) enters the scene in frame 482 and is present up to the last frame 633

[3] See http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

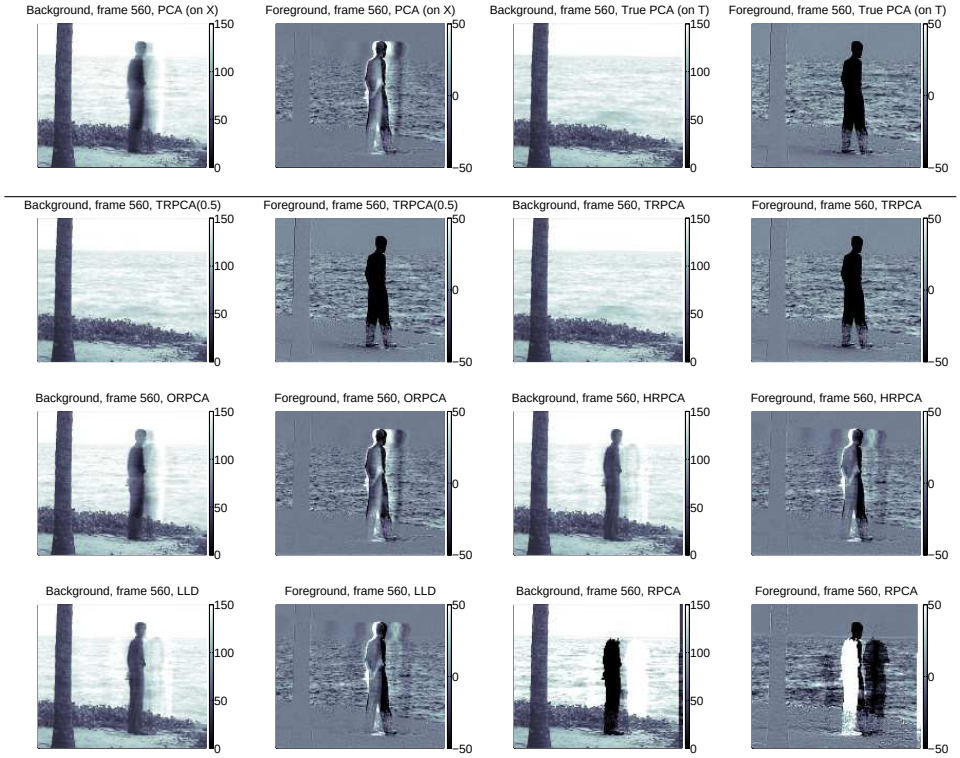

Fig. 7. Background x_i^b and foreground x_i^f recovered with different methods, using (16), of frame 560 of the water surface data set, number of components $k = 10$. These images correspond to the one of Fig. 5, but the scaling has been changed for better visibility. Namely, all backgrounds/foreground images are rescaled so that the maximum and minimum pixel values are the same (please, note the numbers on the color bar); results for PCP can be found in Fig. 8

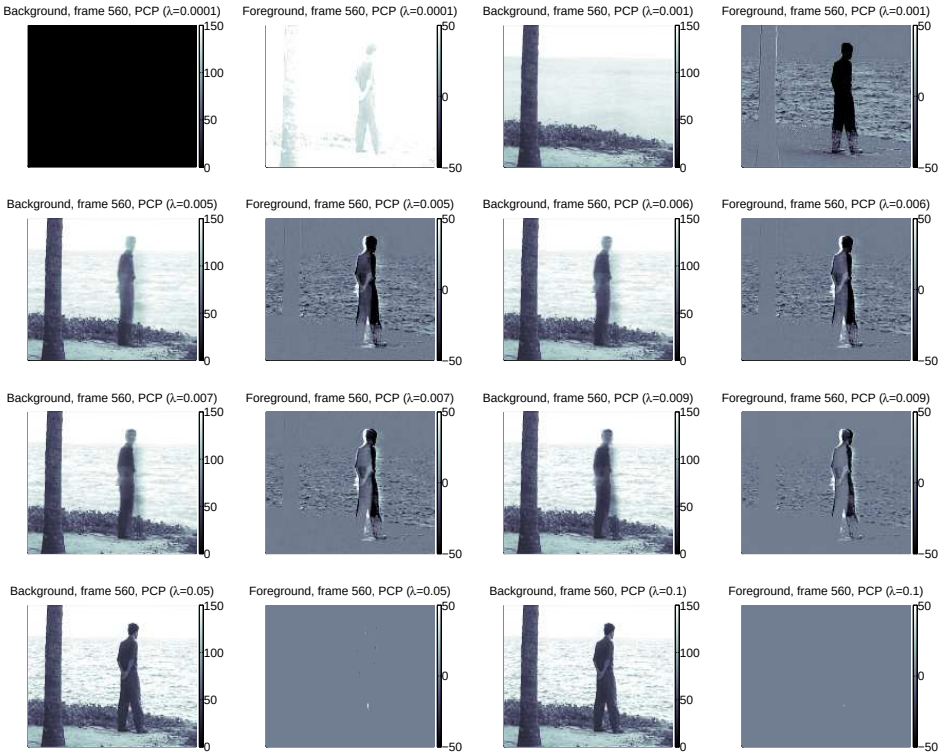

Fig. 8. Background x_i^b and foreground x_i^f recovered, using (16), of frame 560 of the water surface data set with PCP using different regularization parameters. See similar results for other methods in previous Fig. 7

Fig. 9. Examples of the original frames of the moved object data set. Frames from 1 to 637, from 892 to 1389, from 1503 to 1744 (end) contain only background (true data). The Person (outlier) is visible in the scene from frame 638 to 891 and from frame 1390 to 1502. We refer to frames 0 to 892 in the following as the reduced moved object data set

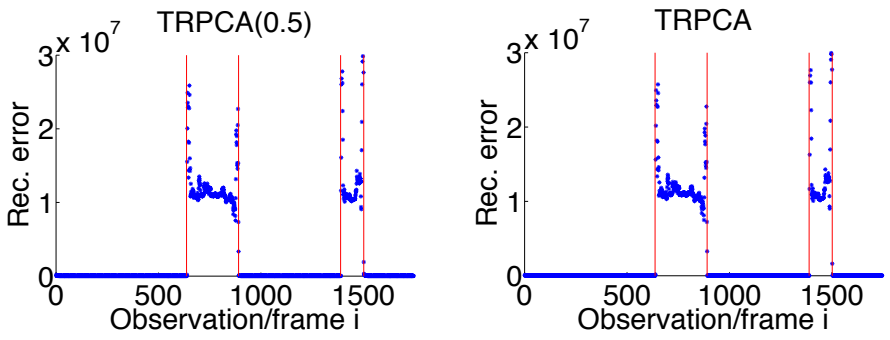

Fig. 10. The reconstruction error of TPRCA/TRPCA(0.5), by analogy with Fig. 4, for the **full** moved object data set. The red vertical lines correspond to frames where the person enters/leaves the scene. We do not perform this experiment on the full datset for all other methods given their high runtimes (see Table 1) and instead proceed with the reduced dataset (see figures below).
Please note also that there is a small change in the background between frames from 1 to 637 (B1) and frames from 892 to 1389 (B2). Thus the robust PCA components will capture this difference. This is not a problem for outlier detection (as we can see from the reconstruction errors of our method above) as this change is still small compared to the variation when the person enters the scene but it disturbs the foreground/background detection of all methods. An offline method could detect the scenes with small reconstruction error and do the background/foreground decomposition for each segment separately. The other option would be to use an online estimation procedure of robust components and center. We do not pursue these directions in this paper as the main purpose of these experiments is an illustration of the differences of the various robust PCA methods in the literature

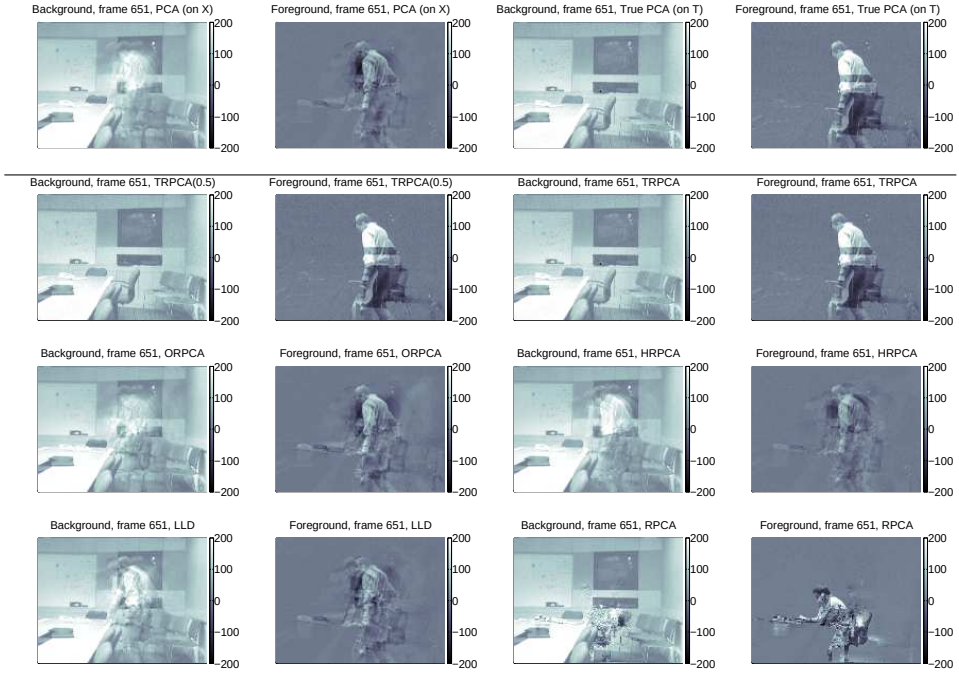

Fig. 11. Extracted background and foreground of frame 651 of the reduced moved object data set. The number of components is $k = 10$ (scaled, compare to unscaled version in Fig. 12)

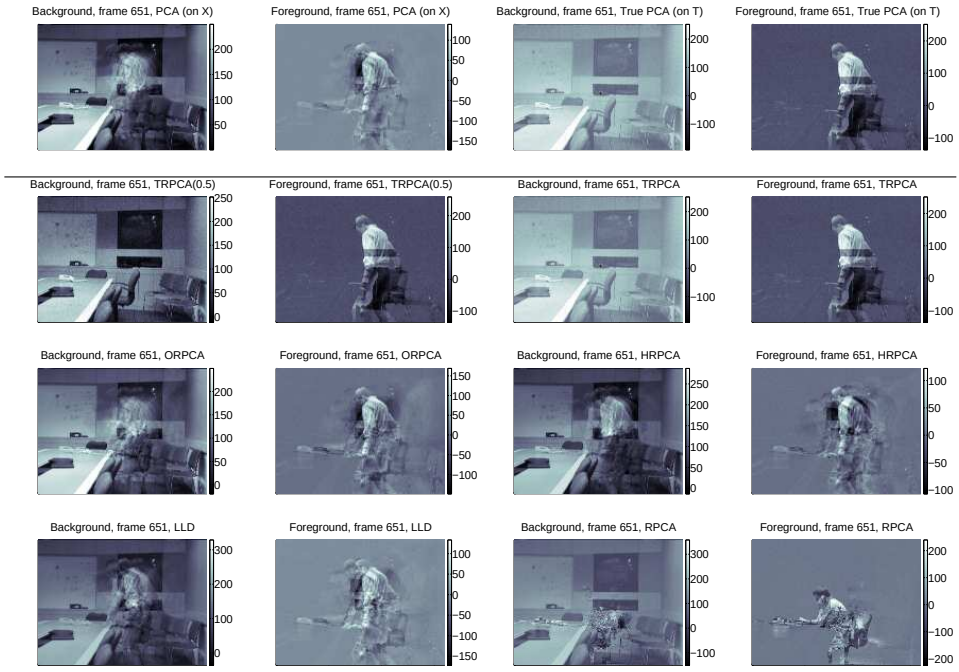

Fig. 12. Extracted background and foreground of frame 651 of the reduced moved object data set. The number of components is $k = 10$ (unscaled, compare to scaled version in Fig. 11)

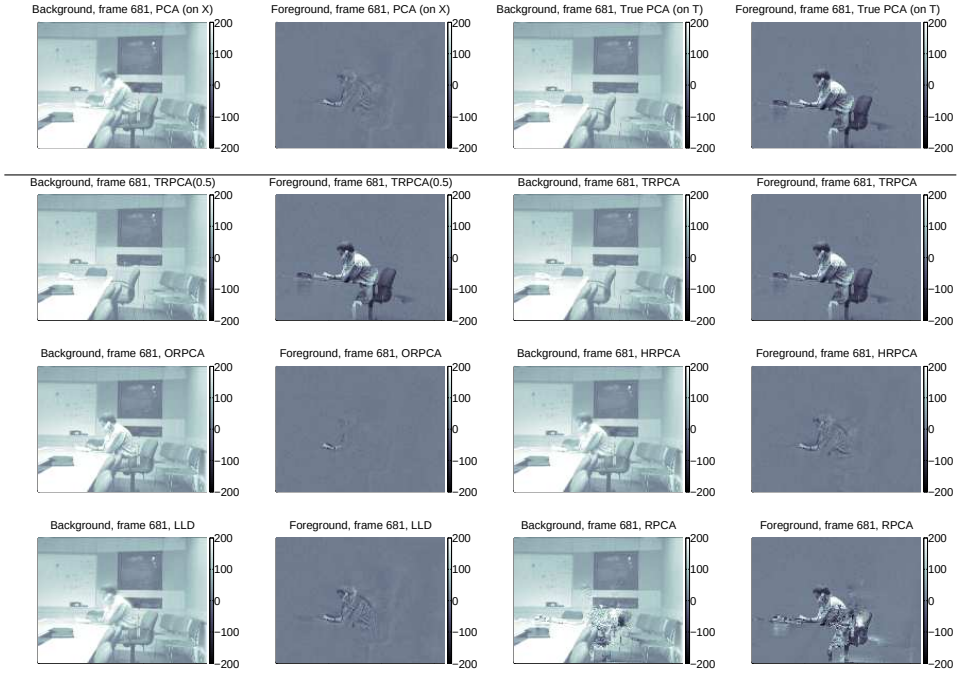

Fig. 13. Extracted background and foreground of frame 681 of the reduced moved object data set. The number of components is $k = 10$ (scaled, compare to unscaled version in Fig. 14)

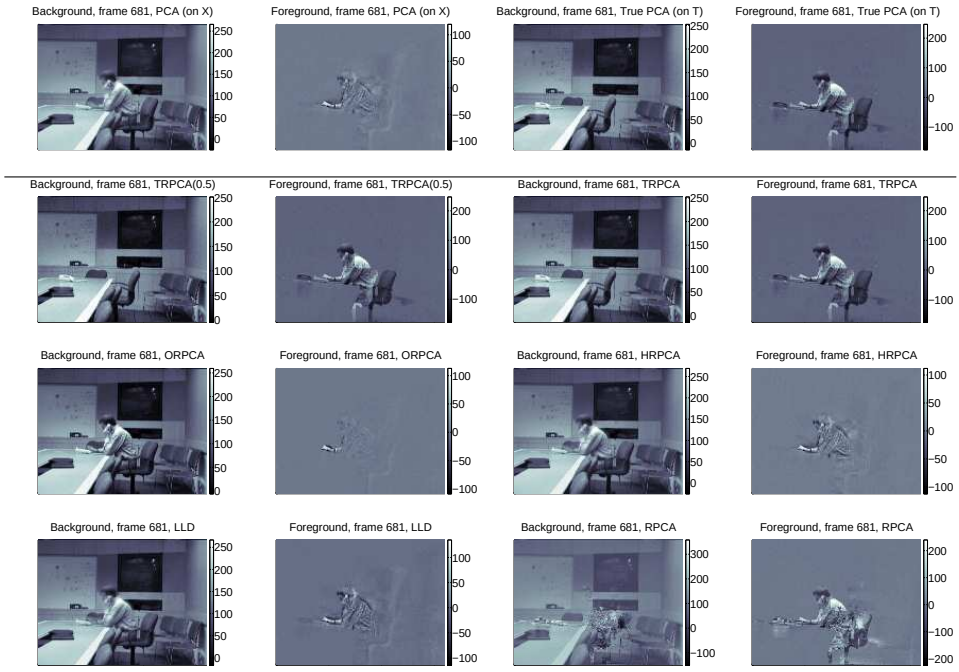

Fig. 14. Extracted background and foreground of frame 681 of the reduced moved object data set. The number of components is $k = 10$ (unscaled, compare to scaled version in Fig. 13)

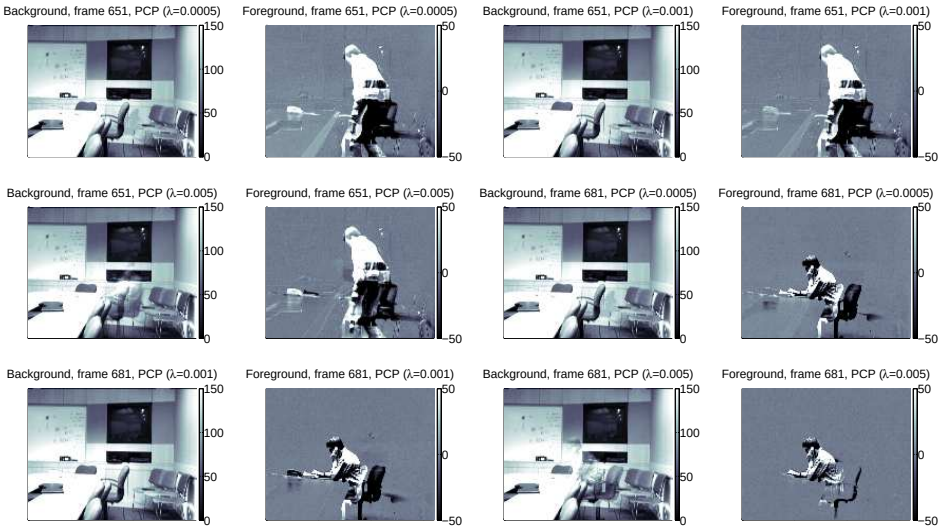

Fig. 15. Extracted background and foreground of frames 651 and 681 of the reduced moved object data set obtained with PCP (scaled, compare to unscaled version in Fig. 16)

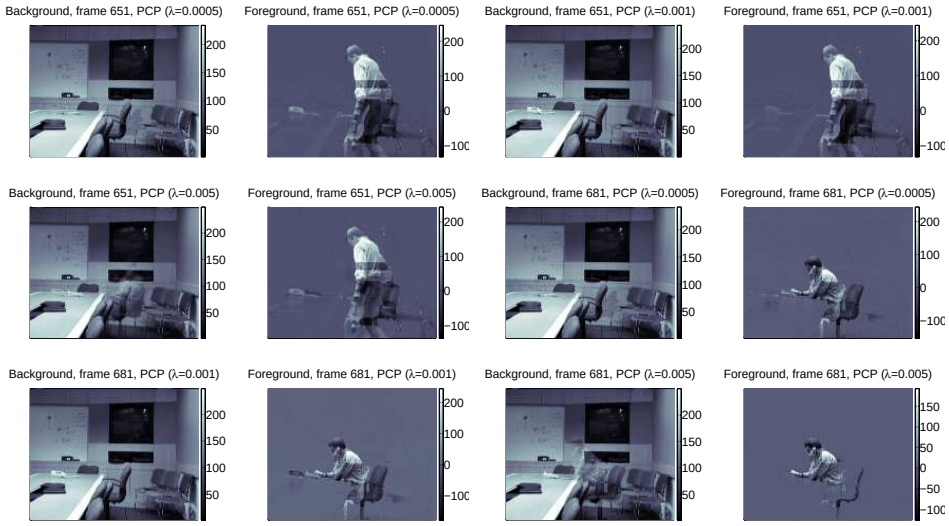

Fig. 16. Extracted background and foreground of frames 651 and 681 of the reduced moved object data set obtained with PCP (unscaled, compare to scaled version in Fig. 15)

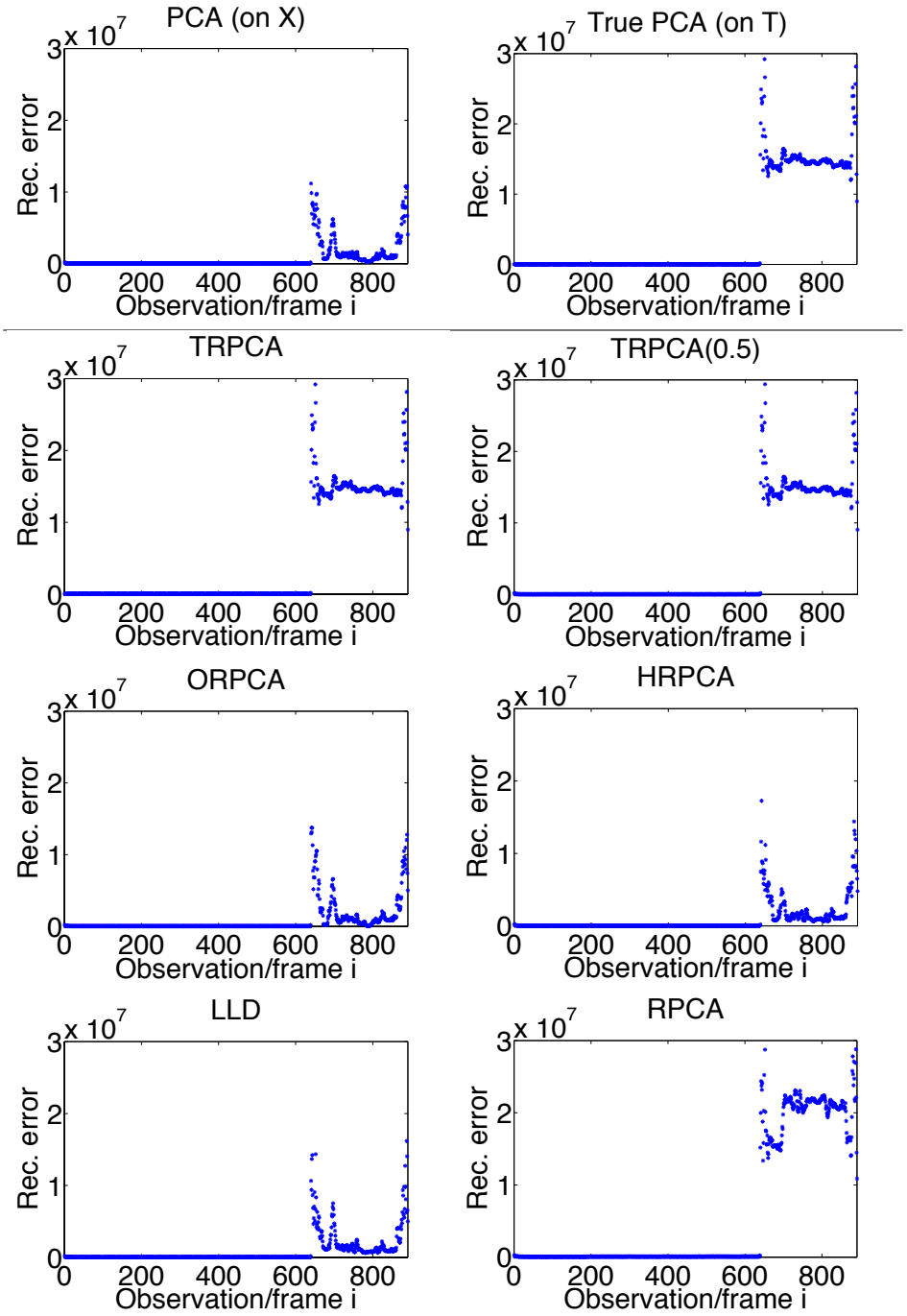

Fig. 17. Reconstruction errors of different methods on the reduced moved object data set (analogous to Fig. 4). One can see that TRPCA/TRPCA(0.5) again recovers the reconstruction errors of the true data almost perfectly as opposed to all other methods. However, note that RPCA does also well in having large reconstruction error for all frames containing the person