Side Information in Robust Principal Component Analysis: Algorithms and Applications

Supplementary Material

1 Parameter calibration

In order to tune the algorithmic parameters, we first conduct a benchmark experiment as follows: a low-rank matrix $\mathbf{L}_0$ is generated from $\mathbf{L}_0 = \mathbf{J}\mathbf{K}^T$, where $\mathbf{J}, \mathbf{K} \in \mathbb{R}^{200\times 10}$ have entries from a $\mathcal{N}(0, 0.005)$ distribution; a 200×200 sparse matrix $\mathbf{S}_0$ is generated by randomly setting $38,000$ entries to zero with others taking values of ± 1 with equal probability.

If $\mathbf{X}$ is set as the left-singular vectors of $\mathbf{L}_0$ and $\mathbf{Y}$ is set as the right-singular vectors of $\mathbf{L}_0$, then a scaling ratio $\alpha = 1.1$, a tolerance threshold $\epsilon = 10^{-7}$ and a maximum step size $\mu = 10^{18}$ to avoid ill-conditioning can bring PCP, RAPS, PCPF to convergence with a recovered $\mathbf{L}$ of rank 10, a recovered $\mathbf{S}$ of sparsity 5% and an accuracy $\|\mathbf{L} - \mathbf{L}_0\|_F / \|\mathbf{L}_0\|_F$ on the order of 10^{-6}. Hereafter, we will adopt these parameter settings for PCP, RAPS, PCPF and will apply them to PCPS and PCPSF as well. PSSV also uses these parameter settings as done similarly in [19].

For RPCAG and FRPCAG, the graphs are built using k-nearest neighbors. Using Euclidean distances, each sample is connected to 10 nearest neighbors with weight $e^{-\frac{s^2}{\sigma^2}}$, where s is the Euclidean distance between the two samples and σ is the average of s. Weight between unconnected samples is set to 0. Having obtained such weight matrix $\mathbf{A}$, we can calculate the normalised graph Laplacian $\mathbf{\Phi} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is the diagonal degree matrix. The tolerance threshold for RPCAG and FRPCAG are all set to $\epsilon = 10^{-7}$ for reasons of consistency. We choose $\lambda = 1/\sqrt{\max(n_1, n_2)}$ for a general matrix of dimension $n_1 \times n_2$ as suggested in [23,24]. For simulation experiments, γ in RPCAG is given by the minimiser (at $\gamma = 0.2$) of $\frac{\|\mathbf{L}-\mathbf{L}_0\|_F}{\|\mathbf{L}_0\|_F}$ on the benchmark problem (Figure 1). And for real-world datasets, γ is set to 10 following [23]. For FRPCAG, we take $\gamma = \gamma_1 = \gamma_2$ which is searched over $[0.01, 10]$ on the benchmark problem (Figure 2). The resulting minimiser (at $\gamma = 7.3$) of $\frac{\|\mathbf{L}-\mathbf{L}_0\|_F}{\|\mathbf{L}_0\|_F}$ is used in both simulation and real-world experiments.

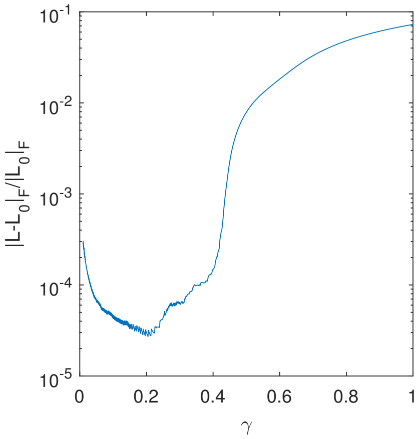

Figure 1: Relative error ($\frac{\|\mathbf{L}-\mathbf{L}_0\|_F}{\|\mathbf{L}_0\|_F}$) of RPCAG for $\gamma \in [0.01, 1]$.

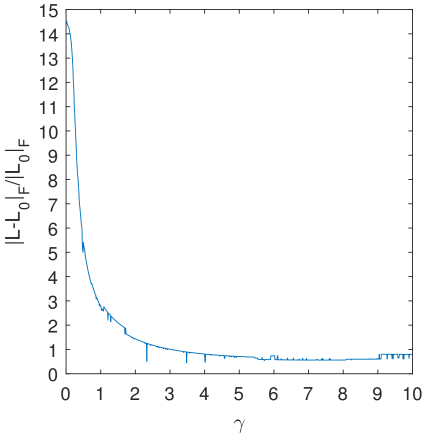

Figure 2: Relative error ($\frac{\|\mathbf{L}-\mathbf{L}_0\|_F}{\|\mathbf{L}_0\|_F}$) of FRPCAG for $\gamma \in [0.01, 10]$.

To find λ and κ in PCPS, a parameter sweep in the $\kappa - \lambda$ space for perfect side information ($\mathbf{W} = \mathbf{L}_0$) is shown in Figure 3 (a) and for observation as side information ($\mathbf{W} = \mathbf{M}$) in Figure 3 (b) to impart a lower bound and a upper bound respectively. It can be easily seen that $\lambda = 1/\sqrt{200}$ from PCP works well in both cases. Conversely, κ depends on the quality of the side information. At $\lambda = 1/\sqrt{200}$, the minimiser of $\frac{\|\mathbf{L}-\mathbf{L}_0\|_F}{\|\mathbf{L}_0\|_F}$ occurs at $\kappa = 0.2$ for noisy side information. This value of κ together with $\lambda = 1/\sqrt{200}$ is used in simulation experiments for both PCPS and PCPSF. For public video sequences, increasing the value of κ to 0.5 can produce visual results that are noticeable to the naked eye.

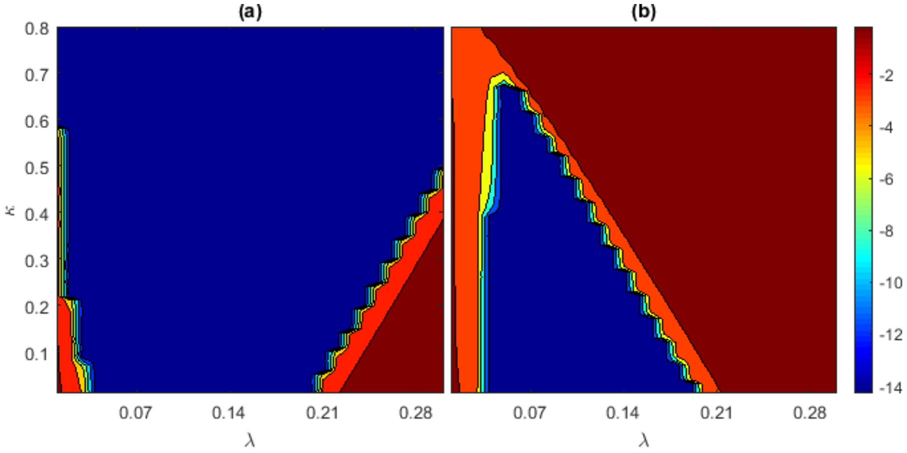

Figure 3: Relative error ($\frac{\|\mathbf{L}-\mathbf{L}_0\|_F}{\|\mathbf{L}_0\|_F}$) of PCPS: **(a)** when side information is perfect; **(b)** when side information is the observation.

2 Simulation Results

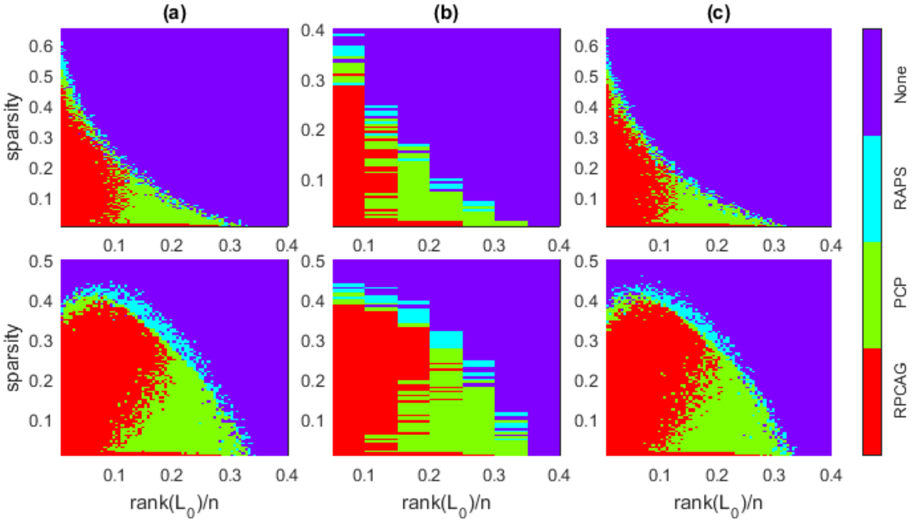

Figure 4: Domains of recovery by various algorithms: random signs in row **I** and coherent signs in row **II**. **(a)** for entry-wise corruptions, **(b)** for deficient ranks and **(c)** for distorted singular values.

A direct comparison of RAPS, RPCAG and PCP from simulation studies is presented in Figure 4. Simulation results for PSSV are shown in Figure 5.

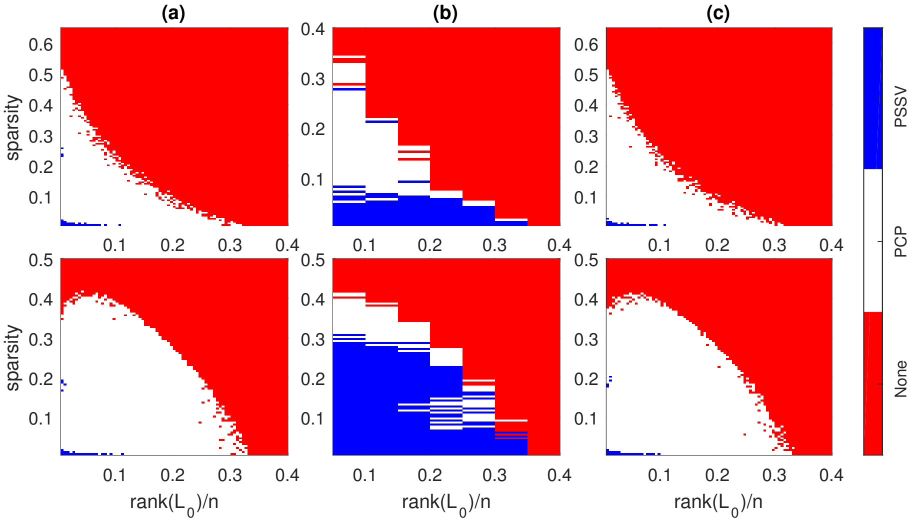

Figure 5: Domains of recovery by PSSV: random signs in row **I** and coherent signs in row **II**. **(a)** for entry-wise corruptions, **(b)** for deficient ranks and **(c)** for distorted singular values.

3 Real-world applications

3.1 Data sources

The datasets used herein are listed below:

The Extended Yale Face Database B: `http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html`.

Performance Evaluation of Tracking and Surveillance Workshop 2006: `http://www.cvg.reading.ac.uk/PETS2006/data.html`.

I2R Dataset: `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`.

The CMU Multi-PIE Face Database: `http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html`.

The Extended Cohn-Kande Dataset (CK+): `http://www.consortium.ri.cmu.edu/ckagree/`.

3.2 Face denoising

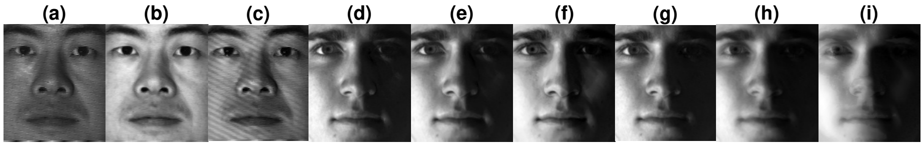

Figure 6: Comparison of face denoising ability: **(a,d)** single-person PSSV; **(b,e)** single-person RPCAG; **(c,f)** single-person FRPCAG; **(g)** multi-person PSSV; **(h)** multi-person RPCAG; and **(i)** multi-person FRPCAG;.

Illustration of face denoising ability of PSSV, RPCAG, FRPCAG is presented in Figure 6. The average running times of different algorithms for a single subject and multiple subjects are summarised in Table 1 [1].

[1] All experiments were performed on a 3.60GHz quad-core computer with 16GB RAM running MATLAB R2016a.

Algorithm	Time	
	Single Subject	Multiple Subjects
K-SVD (X)	9 min	—
K-SVD (Y)	78 min	—
PCP	12s	5 min
PCPS	27s	12 min
PCPF	16s	9 min
PCPSF	19s	8 min
PSSV	13s	5 min
k-NN (X)	7s	4 min
k-NN (Y)	1s	8s
RPCAG	2min	17 min
FRPCAG	8s	1 min

Table 1: Running times of various algorithms.

3.3 Background Subtraction

Recovered images of the background and the foreground from all methods are listed in Figure 7 for Airport and Figure 8 for PETS. The running times of different algorithms for Airport and PETS are summarised in Table 2.

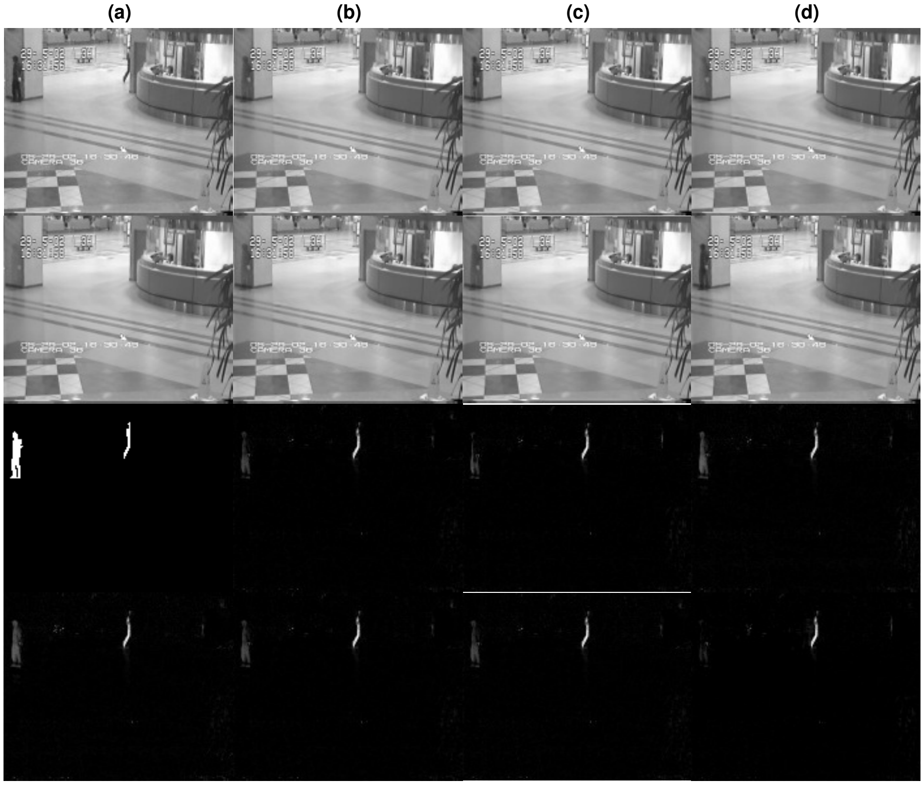

Figure 7: Background subtraction results for Airport : row **I (a)** original image; row **III (a)** ground truth; row **I,III (b)** PCP; row **I,III (c)** PCP (60 frames); **I,III (d)** PCPS (60 frames); row **II,IV (a)** PCPS; row **II,IV (b)** PSSV; row **II,IV (c)** RPCAG; row **II,IV (d)** FRPCAG.

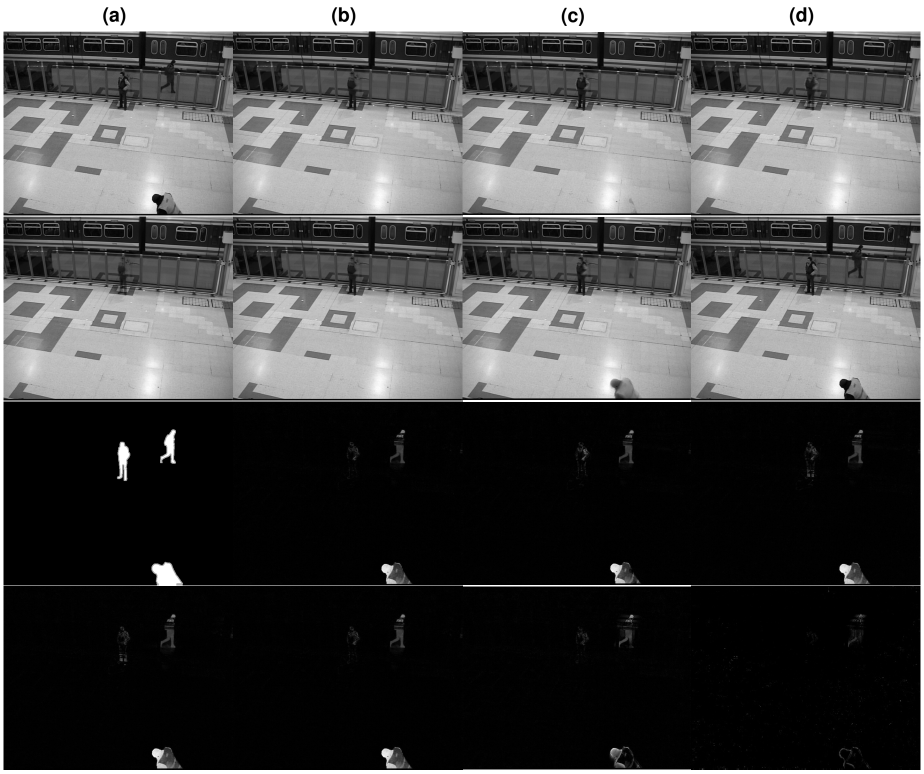

Figure 8: Background subtraction results for PETS : row **I (a)** original image; row **III (a)** ground truth; row **I,III (b)** PCP; row **I,III (c)** PCP (60 frames); **I,III (d)** PCPS (60 frames); row **II,IV (a)** PCPS; row **II,IV (b)** PSSV; row **II,IV (c)** RPCAG; row **II,IV (d)** FRPCAG.

Algorithm	Time	
	Airport	PETS
PCP	52s	17 min
PCPS	2 min	36 min
PSSV	51s	17 min
k-NN (X)	52s	2h
k-NN (Y)	1s	24s
RPCAG	7 min	3h
FRPCAG	11s	34s
PCP (60 frames)	52s	3 min
PCPS (60 frames)	20s	7 min

Table 2: Running times of various algorithms.

4 Derivations

Here we give deviations of the various equivalent subproblems for the algorithm quoted in the text:

$$
\begin{aligned}
&\operatorname*{argmin}_H l(H,E,S,Z,N)\\
={}&\operatorname*{argmin}_H \ ||H||_* + \kappa||E||_* + \lambda||S||_1 + \langle Z, M-S-XHY^T\rangle + \frac{\mu}{2}||M-S-XHY^T||_F^2\\
&+\langle N, H-E-X^TWY\rangle + \frac{\mu}{2}||H-E-X^TWY||_F^2\\
={}&\operatorname*{argmin}_H \ ||H||_* + \langle Z, M-S-XHY^T\rangle + \frac{\mu}{2}||M-S-XHY^T||_F^2\\
&+\langle N, H-E-X^TWY\rangle + \frac{\mu}{2}||H-E-X^TWY||_F^2\\
={}&\operatorname*{argmin}_H \ ||H||_* + \operatorname{tr}(Z^T(M-S-XHY^T))\\
&+\frac{\mu}{2}\operatorname{tr}((M-S-XHY^T)^T(M-S-XHY^T)) + \operatorname{tr}(N^T(H-E-X^TWY))\\
&+\frac{\mu}{2}\operatorname{tr}((H-E-X^TWY)^T(H-E-X^TWY))\\
={}&\operatorname*{argmin}_H \ ||H||_* - \operatorname{tr}(Z^TXHY^T) + \operatorname{tr}(N^TH)\\
&+\frac{\mu}{2}\operatorname{tr}(YH^TX^TXHY^T - YH^TX^T(M-S) - (M-S)^TXHY^T)\\
&+\frac{\mu}{2}\operatorname{tr}((H-E-X^TWY)^TX^TX(H-E-X^TWY)Y^TY)\\
={}&\operatorname*{argmin}_H \ ||H||_* + \mu\operatorname{tr}(-\frac{1}{\mu}Z^TXHY^T) + \mu\operatorname{tr}(\frac{1}{\mu}N^TX^TXHY^TY)\\
&+\frac{\mu}{2}\operatorname{tr}(YH^TX^TXHY^T - YH^TX^T(M-S) - (M-S)^TXHY^T)\\
&+\frac{\mu}{2}\operatorname{tr}(YH^TX^TXHY^T - YH^TX^TX(E+X^TWY)Y^T\\
&-Y(E+X^TWY)^TX^TXHY^T)\\
={}&\operatorname*{argmin}_H \ ||H||_* + \mu\operatorname{tr}(YH^TX^TXHY^T - \frac{1}{2}YH^TX^T(M-S) - \frac{1}{2}(M-S)^TXHY^T\\
&-\frac{1}{2}YH^TX^TX(E+X^TWY)Y^T - \frac{1}{2}Y(E+X^TWY)^TX^TXHY^T\\
&-\frac{1}{2\mu}YH^TX^TZ - \frac{1}{2\mu}Z^TXHY^T + \frac{1}{2\mu}YH^TX^TXNY^T + \frac{1}{2\mu}YN^TX^TXHY^T)\\
={}&\operatorname*{argmin}_H \ ||H||_* + \mu\operatorname{tr}((\frac{1}{2}(M-S+XEY^T+W+\frac{1}{\mu}(Z-XNY^T)) - XHY^T)^T\\
&(\frac{1}{2}(M-S+XEY^T+W+\frac{1}{\mu}(Z-XNY^T)) - XHY^T))\\
={}&\operatorname*{argmin}_H \ ||H||_* + \mu||\frac{1}{2}(M-S+W+\frac{1}{\mu}Z+X(E-\frac{1}{\mu}N)Y^T) - XHY^T||_F^2
\end{aligned}
$$

$$
\begin{aligned}
&\operatorname*{argmin}_E l(H, E, S, Z, N) \\
=&\operatorname*{argmin}_E \;\; ||H||_* + \kappa||E||_* + \lambda||S||_1 + \langle Z, M - S - XHY^T\rangle + \frac{\mu}{2}||M - S - XHY^T||_F^2 \\
&+ \langle N, H - E - X^TWY\rangle + \frac{\mu}{2}||H - E - X^TWY||_F^2 \\
=&\operatorname*{argmin}_E \;\; \kappa||E||_* + \langle N, H - E - X^TWY\rangle + \frac{\mu}{2}||H - E - X^TWY||_F^2 \\
=&\operatorname*{argmin}_E \;\; \kappa||E||_* + \operatorname{tr}(N^T(H - E - X^TWY)) \\
&+ \frac{\mu}{2}\operatorname{tr}((H - E - X^TWY)^T(H - E - X^TWY)) \\
=&\operatorname*{argmin}_E \;\; \kappa||E||_* + \frac{\mu}{2}\operatorname{tr}(-\frac{2}{\mu}N^TE) \\
&+ \frac{\mu}{2}\operatorname{tr}(E^TE - E^T(H - X^TWY) - (H - X^TWY)^TE) \\
=&\operatorname*{argmin}_E \;\; \kappa||E||_* \\
&+ \frac{\mu}{2}\operatorname{tr}(E^TE - E^T(H - X^TWY) - (H - X^TWY)^TE - \frac{1}{\mu}E^TN - \frac{1}{\mu}N^TE) \\
=&\operatorname*{argmin}_E \;\; \kappa||E||_* + \frac{\mu}{2}\operatorname{tr}((H - X^TWY + \frac{1}{\mu}N - E)^T(H - X^TWY + \frac{1}{\mu}N - E)) \\
=&\operatorname*{argmin}_E \;\; \kappa||E||_* + \frac{\mu}{2}||H - X^TWY + \frac{1}{\mu}N - E||_F^2
\end{aligned}
$$

$$
\begin{aligned}
&\operatorname*{argmin}_S l(H,E,S,Z,N)\\
=&\operatorname*{argmin}_S \; ||H||_* + \kappa||E||_* + \lambda||S||_1 + \langle Z, M-S-XHY^T\rangle + \frac{\mu}{2}||M-S-XHY^T||_F^2\\
&+\langle N, H-E-X^TWY\rangle + \frac{\mu}{2}||H-E-X^TWY||_F^2\\
=&\operatorname*{argmin}_S \; \lambda||S||_1 + \langle Z, M-S-XHY^T\rangle + \frac{\mu}{2}||M-S-XHY^T||_F^2\\
=&\operatorname*{argmin}_S \; \lambda||S||_1 + \operatorname{tr}(Z^T(M-S-XHY^T))\\
&+\frac{\mu}{2}\operatorname{tr}((M-S-XHY^T)^T(M-S-XHY^T))\\
=&\operatorname*{argmin}_S \; \lambda||S||_1 + \frac{\mu}{2}\operatorname{tr}(-\frac{2}{\mu}Z^TS)\\
&+\frac{\mu}{2}\operatorname{tr}(S^TS - S^T(M-XHY^T) - (M-XHY^T)^TS)\\
=&\operatorname*{argmin}_S \; \lambda||S||_1\\
&+\frac{\mu}{2}\operatorname{tr}(S^TS - S^T(M-XHY^T) - (M-XHY^T)^TS - \frac{1}{\mu}S^TZ - \frac{1}{\mu}Z^TS)\\
=&\operatorname*{argmin}_S \; \lambda||S||_1 + \frac{\mu}{2}\operatorname{tr}((M-XHY^T+\frac{1}{\mu}Z-S)^T(M-XHY^T+\frac{1}{\mu}Z-S))\\
=&\operatorname*{argmin}_S \; \lambda||S||_1 + \frac{\mu}{2}||M-XHY^T+\frac{1}{\mu}Z-S||_F^2
\end{aligned}
$$

5 Further comments

One might suggest that a potentially better and more direct approach in using the side information is to subtract the side information. That is, do RPCA on $\mathbf{M}' = \mathbf{M} - \mathbf{W}$, where $\mathbf{M}$ is the data and $\mathbf{W}$ is the noisy side information, to obtain $\mathbf{M}' = \mathbf{L}' + \mathbf{S}$ with $\mathbf{L} = \mathbf{L}' + \mathbf{W}$.

We argue that this is not correct for the following reasons:

- The rank of $\mathbf{L}'$ is no smaller than $\mathbf{L}$, which does not make the problem any simpler than the original one.

- When $\mathbf{W}$ is merged into $\mathbf{M}$, the additional information provided by $\mathbf{W}$ is lost and the features can on longer be applied.

- When $\mathbf{W}$ includes full-rank noise on $\mathbf{L}$, $\mathbf{L}'$ is not low-rank anymore. This violates the assumption of RPCA.

To verify our claim, we perform the Airport experiment again, but with different side information than that used in the paper. We collect 200 different frames of relatively clean backgrounds and stack them into the side information $\mathbf{W}$. Comparison of the suggestion with PCPS and PCP is shown in Figure 9, 10 and 11. It is clearly visible that the low-rank structure cannot be recovered by the suggestion and spurious noises are introduced in the segmentation, whereas PCPS works impeccably segmenting accurately the foreground moving objects leaving a clean background.

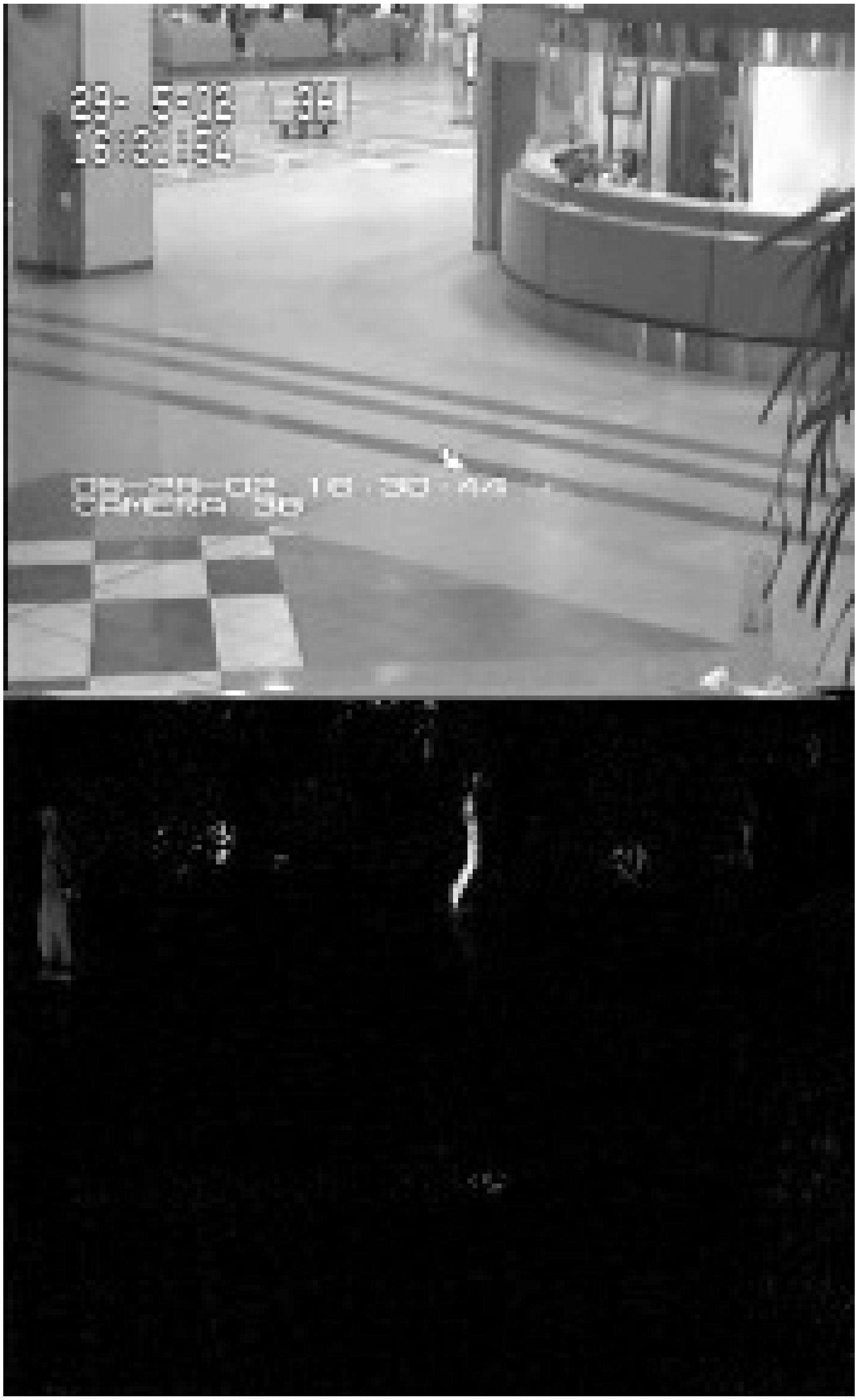

Figure 9: Background subtraction by suggestion: background in row **I** and segmentaion in row **II**.

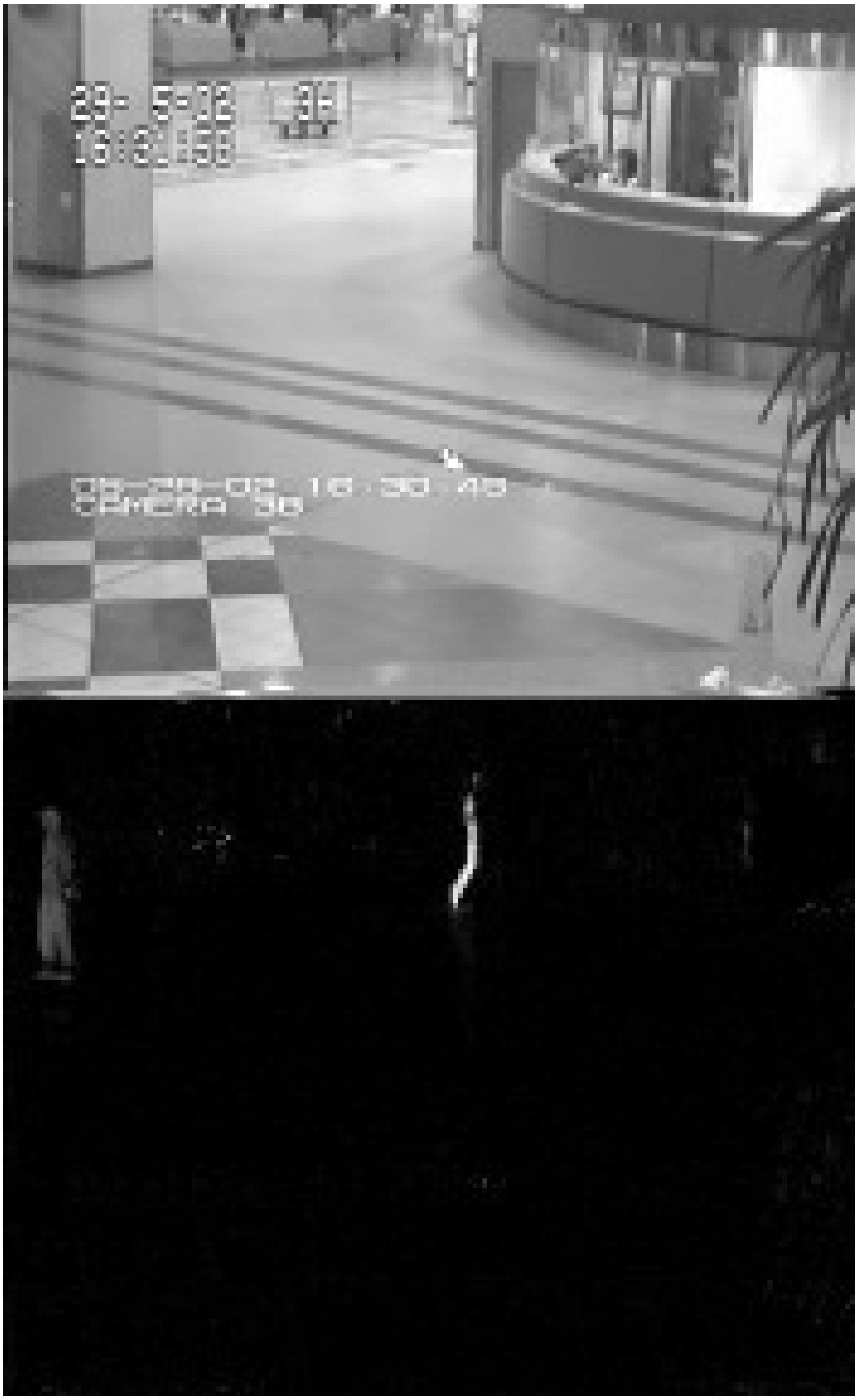

Figure 10: Background subtraction by PCPS: background in row **I** and segmentaion in row **II**.

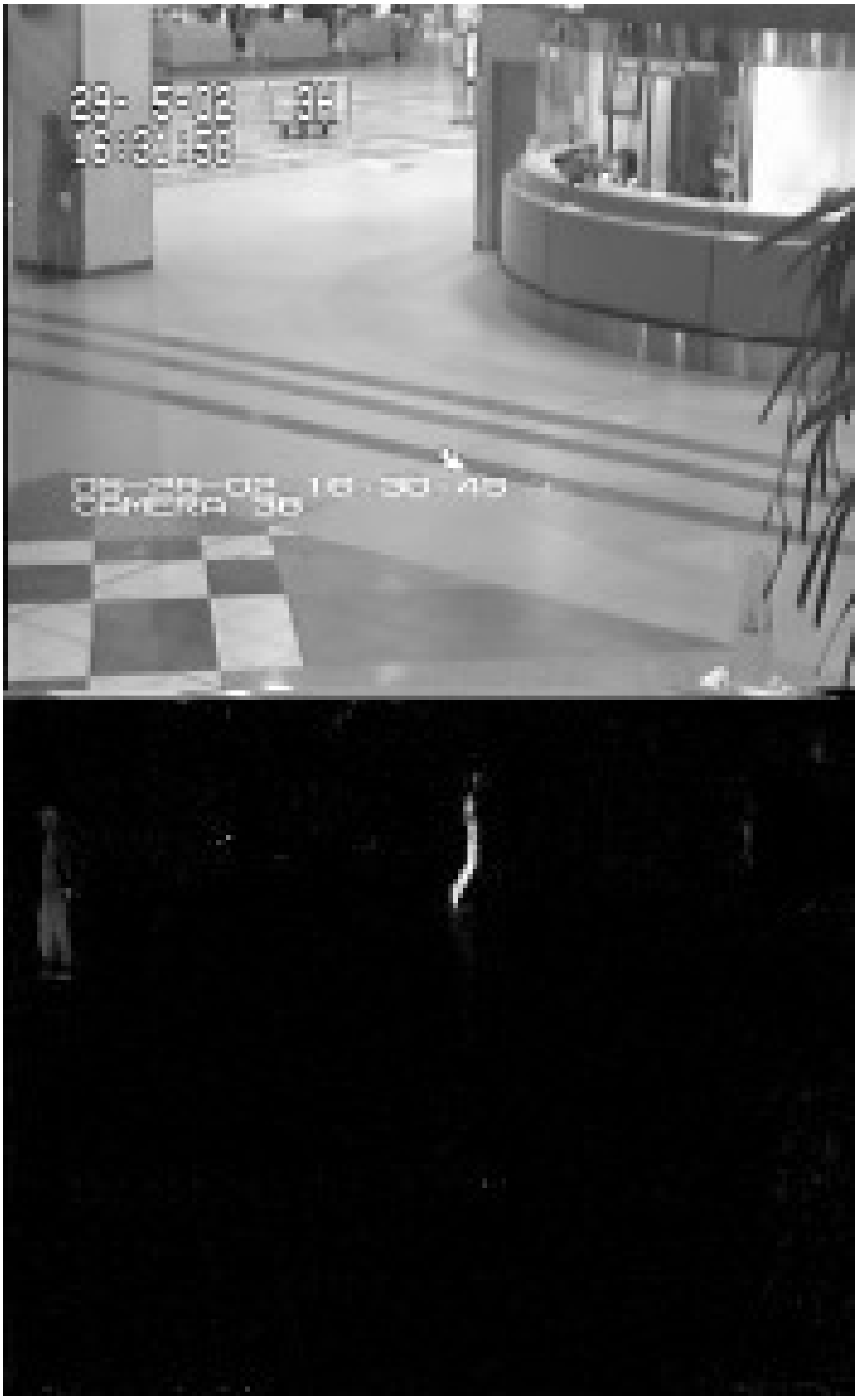

Figure 11: Background subtraction by PCP: background in row **I** and segmentaion in row **II**.