Random Consensus Robust PCA
Supplementary Material

Daniel Pimentel-Alarcón & Robert Nowak
University *of* Wisconsin-Madison

A Proof of Theorem 1

In this section we give the proof of Theorem 1. Recall that $\mathbf{\Omega}$ is a $d \times (d-r)$ matrix, and that $\boldsymbol{\omega}_i \subset \{1,\dots,d\}$ indexes the $r+1$ nonzero entries in the i^{th} column of $\mathbf{\Omega}$. Each $\boldsymbol{\omega}_i$ indicates the coordinates of a projection of U that we aim to identify. Since R2PCA selects a matrix $\mathbf{\Omega}$ satisfying (i), Lemma 1 implies that if we find the projections of U onto the coordinates indicated in $\mathbf{\Omega}$, then we can reconstruct U from these projections. Hence we need to show that under the assumptions of Theorem 1, R2PCA can find the projections of U onto the $\boldsymbol{\omega}_i$'s in $\mathbf{\Omega}$. To this end, we will show that R2PCA can potentially find the projection onto *any* $\boldsymbol{\omega} \subset \{1,2,\dots,d\}$ with exactly $r+1$ elements.

So let $\boldsymbol{\omega}$ be given. As discussed in Section 2, finding the projection $U_{\boldsymbol{\omega}}$ equates to finding $r+1$ uncorrupted columns in $\mathbf{M}_{\boldsymbol{\omega}}$. So R2PCA can potentially find $U_{\boldsymbol{\omega}}$ as long as there are $r+1$ uncorrupted columns in $\mathbf{M}_{\boldsymbol{\omega}}$. Since $\mathbf{M}_{\boldsymbol{\omega}}$ only contains $r+1$ rows, **A4** implies that there are at most $\frac{(n-r)(r+1)}{2(r+1)^{\alpha}}$ corrupted entries in $\mathbf{M}_{\boldsymbol{\omega}}$. In the worst-case scenario, each of these corrupted entries is located in a different column. It follows that $\mathbf{M}_{\boldsymbol{\omega}}$ has at most $\frac{n-r}{2(r+1)^{\alpha-1}}$ corrupted columns. Then

$$\mathsf{P}\big(i^{\text{th}} \text{ column in } \mathbf{M}'_{\boldsymbol{\omega}} \text{ is uncorrupted}\big) \geq 1 - \frac{1}{2(r+1)^{\alpha-1}},$$

which corresponds to the case where the first r columns in $\mathbf{M}'_{\boldsymbol{\omega}}$ are uncorrupted, whence the ratio of uncorrupted columns ($(n-r) - \frac{n-r}{2(r+1)^{\alpha-1}}$) versus total remaining columns $(n-r)$ is smallest. It follows that

$$\begin{aligned}
&\mathsf{P}\big(\text{all columns in } \mathbf{M}'_{\boldsymbol{\omega}} \text{ are uncorrupted}\big) \\
&\quad\geq \left(1 - \frac{1}{2(r+1)^{\alpha-1}}\right)^{r+1} \\
&\quad= \left(1 - \frac{1/2}{(r+1)^{\alpha-1}}\right)^{(r+1)^{1+(\alpha-1)-(\alpha-1)}} \\
&\quad= \left(1 - \frac{1/2}{(r+1)^{\alpha-1}}\right)^{(r+1)^{(\alpha-1)}(r+1)^{2-\alpha}} \\
&\quad= \left(\left(1 - \frac{1/2}{(r+1)^{\alpha-1}}\right)^{(r+1)^{(\alpha-1)}}\right)^{(r+1)^{2-\alpha}} \\
&\quad\geq (1/2)^{(r+1)^{2-\alpha}}. \qquad (1)
\end{aligned}$$

This implies that on expectation, R2PCA will require at most $2^{(r+1)^{2-\alpha}}$ iterations to find a set of $r+1$ uncorrupted columns in $\mathbf{M}_{\boldsymbol{\omega}}$. This is true for every $\boldsymbol{\omega}$. Since R2PCA only searches over the $\boldsymbol{\omega}_i$'s in $\mathbf{\Omega}$, and since $\mathbf{\Omega}$ has exactly $d-r$ columns, it follows that on expectation, R2PCA will require at most $(d-r)2^{(r+1)^{2-\alpha}}$ iterations to find the projections of U onto the canonical coordinates indicated by $\mathbf{\Omega}$. Since $\mathbf{\Omega}$ satisfies condition (i), we know by Lemma 1 that U is given by $\ker \mathbf{A}^{\mathsf{T}}$.

Now that U is known, let us show that R2PCA can recover $\mathbf{L}$. Let $\mathbf{U}$ be an arbitrary basis of U. We will show that R2PCA can determine the matrix $\mathbf{\Theta}$ containing the coefficients of $\mathbf{L}$ in this basis, such that in the end, $\mathbf{L}$ will be given by $\mathbf{U\Theta}$. To this end, let $\mathbf{m}$ be a column in $\mathbf{M}$. Observe that R2PCA can potentially find the coefficients of the corresponding column of $\mathbf{L}$ as long as there is a set $\boldsymbol{\omega} \subset \{1,2,\dots,d\}$ with $r+1$ elements such that $\mathbf{m}_{\boldsymbol{\omega}} \in U_{\boldsymbol{\omega}}$. **A1-A3** imply that with probability 1, this will be the case if and only there are at least $r+1$ uncorrupted entries in $\mathbf{m}$. By **A4**, there are at most $\frac{d-r}{2(r+1)^{\alpha-1}}$ corrupted entries in $\mathbf{m}$. It follows that

$$\mathsf{P}\big(i^{\text{th}} \text{ entry in } \mathbf{m}_{\boldsymbol{\omega}} \text{ is uncorrupted}\big) \geq 1 - \frac{1}{2(r+1)^{\alpha-1}},$$

which corresponds to the case where the first r entries in $\mathbf{m}_{\boldsymbol{\omega}}$ are uncorrupted, whence the ratio of uncorrupted entries $((d-r) - \frac{d-r}{2(r+1)^{\alpha-1}})$ versus total remaining entries $(d-r)$ is smallest. It follows that

$$\begin{aligned} &\mathsf{P}\big(\text{all entries in } \mathbf{m}_{\boldsymbol{\omega}} \text{ are uncorrupted}\big) \\ &\qquad \geq \left(1 - \frac{1}{2(r+1)^{\alpha-1}}\right)^{r+1} \geq (1/2)^{(r+1)^{2-\alpha}}, \end{aligned}$$

where the last inequality follows by the same arithmetic manipulations as in (1). This implies that on expectation, R2PCA will require at most $2^{(r+1)^{2-\alpha}}$ iterations to find a set of $r+1$ uncorrupted entries in $\mathbf{m}$. This is true for every $\mathbf{m}$. Since $\mathbf{M}$ has n columns, it follows that on expectation, R2PCA will require at most $n2^{(r+1)^{2-\alpha}}$ iterations to recover $\mathbf{L}$. Once $\mathbf{L}$ is known, $\mathbf{S}$ can be trivially recovered as $\mathbf{S} = \mathbf{M} - \mathbf{L}$. This shows that on expectation, R2PCA will require at most $(d+n-r)2^{(r+1)^{2-\alpha}}$ iterations to recover U, $\mathbf{L}$ and $\mathbf{S}$ from $\mathbf{M}$. □

B Noisy Variant

In Section 5 we described a noisy variant of R2PCA. This variant iteratively selects matrices $\mathbf{M}'_{\boldsymbol{\kappa}} \in \mathbb{R}^{k \times k}$ formed with k rows of k columns of $\mathbf{M}$, and verifies the $(r+1)^{\text{th}}$ singular value of $\mathbf{M}'_{\boldsymbol{\kappa}}$. If this singular value is within the noise level, Algorithm 1 will consider $\mathbf{M}'_{\boldsymbol{\kappa}}$ uncorrupted, and use it to estimate projections of U. Otherwise Algorithm 1 will discard $\mathbf{M}'_{\boldsymbol{\kappa}}$ and keep looking. This process is repeated until there are enough projections to recover U. Once U is estimated, Algorithm 1 proceeds to estimate the coefficients of $\mathbf{L}$ using k entries per column of $\mathbf{M}$. If these entries agree with the estimated subspace U, they will be considered uncorrupted, and used to estimate the coefficient of the corresponding column of $\mathbf{L}$. Otherwise, Algorithm 1 will discard these entries, and select an other k. This process is repeated until we recover all the coefficients of $\mathbf{L}$. This noisy variant of R2PCA is summarized in Algorithm 1.

Algorithm 1: Random Robust PCA (R2PCA, noisy variant)

Input: Data $\mathbf{M} \in \mathbb{R}^{d \times n}$, rank r,
matrix $\mathbf{\Omega} \in \{0,1\}^{d \times (d-r)}$ satisfying (i),
parameter $k \in \mathbb{N}$.
PART 1: Estimate $\mathbf{U}$
  for $i = 1, 2, \dots, d-r$ **do**
    $\boldsymbol{\omega}_i$ = indices of the $r+1$ nonzero rows of
      the i^{th} column in $\mathbf{\Omega}$.
    $\boldsymbol{\kappa}_i$ = subset of $\{1, \dots, d\}$ containing $\boldsymbol{\omega}_i$
      and $k-r+1$ other rows selected randomly. **repeat**
      $\mathbf{M}'_{\boldsymbol{\kappa}_i} \in \mathbb{R}^{k \times k}$ = k columns of $\mathbf{M}_{\boldsymbol{\kappa}_i}$,
        selected randomly.
    until $(r+1)^{\text{th}}$ singular value of $\mathbf{M}'_{\boldsymbol{\kappa}_i}$
      is within the noise level.
    $\mathbf{V}_{\boldsymbol{\kappa}_i} \in \mathbb{R}^{k \times r}$ = r leading singular vectors
      of $\mathbf{M}'_{\boldsymbol{\kappa}_i}$.
    $\boldsymbol{\upsilon}_i$ = subset of $\boldsymbol{\kappa}_i$ with exactly r elements,
      selected randomly.
    for each $j \in \boldsymbol{\kappa}_i \backslash \boldsymbol{\upsilon}_i$ **do**
      $\boldsymbol{\omega}_{ij} := \boldsymbol{\upsilon}_i \cup j$.
      $\mathbf{a}_{\boldsymbol{\omega}_{ij}} \in \mathbb{R}^{r+1}$ = nonzero vector
        in $\ker \mathbf{V}^{\mathsf{T}}_{\boldsymbol{\omega}_{ij}}$.
      $\mathbf{a}_{ij} \in \mathbb{R}^d$ = vector with $\mathbf{a}_{\boldsymbol{\omega}_{ij}}$ in the
        locations of $\boldsymbol{\omega}_{ij}$, and zeros
        elsewhere.
      Insert $\mathbf{a}_{ij}$ into $\mathbf{A}$.
  $\hat{\mathbf{U}} \in \mathbb{R}^{d \times r}$ = basis of $\ker \mathbf{A}^{\mathsf{T}}$.
PART 2: Estimate $\mathbf{\Theta}$
  for each column $\mathbf{m}$ in $\mathbf{M}$ **do**
    repeat
      $\boldsymbol{\kappa}$ = subset of $\{1, \dots, d\}$ with k
        elements, selected randomly.
    until $\mathbf{m}_{\boldsymbol{\kappa}}$ is close to span$\{\hat{\mathbf{U}}_{\boldsymbol{\kappa}}\}$
      (within the noise level).
    $\hat{\boldsymbol{\theta}} = (\hat{\mathbf{U}}^{\mathsf{T}}_{\boldsymbol{\kappa}} \hat{\mathbf{U}}_{\boldsymbol{\kappa}})^{-1} \hat{\mathbf{U}}^{\mathsf{T}}_{\boldsymbol{\kappa}} \mathbf{m}_{\boldsymbol{\kappa}}$.
    Insert $\hat{\boldsymbol{\theta}}$ into $\hat{\mathbf{\Theta}}$.
Output: $\hat{\mathbf{U}}$, $\hat{\mathbf{L}} = \hat{\mathbf{U}}\hat{\mathbf{\Theta}}$, $\hat{\mathbf{S}} = \mathbf{M} - \hat{\mathbf{L}}$.

C Additional Results

Microscopy Segmentation In Section 6 we gave three examples of the background segmentation that we obtained for three microscopy videos from the Internet. Figure 1 shows more results.

Wallflower and I2R Datasets. To complement the real data experiments in Section 6, we also ran R2PCA and RPCA-ALM on the Wallflower [23] and the I2R [24] datasets. The results are summarized in Figure 2. We point out that many cases of the Wallflower and the I2R datasets have low coherence. In these cases, the performance of R2PCA and RPCA-ALM is very similar. Consistent with our theory, the advantage of R2PCA becomes more evident in highly coherent cases, like our microscopy and astronomy experiments.

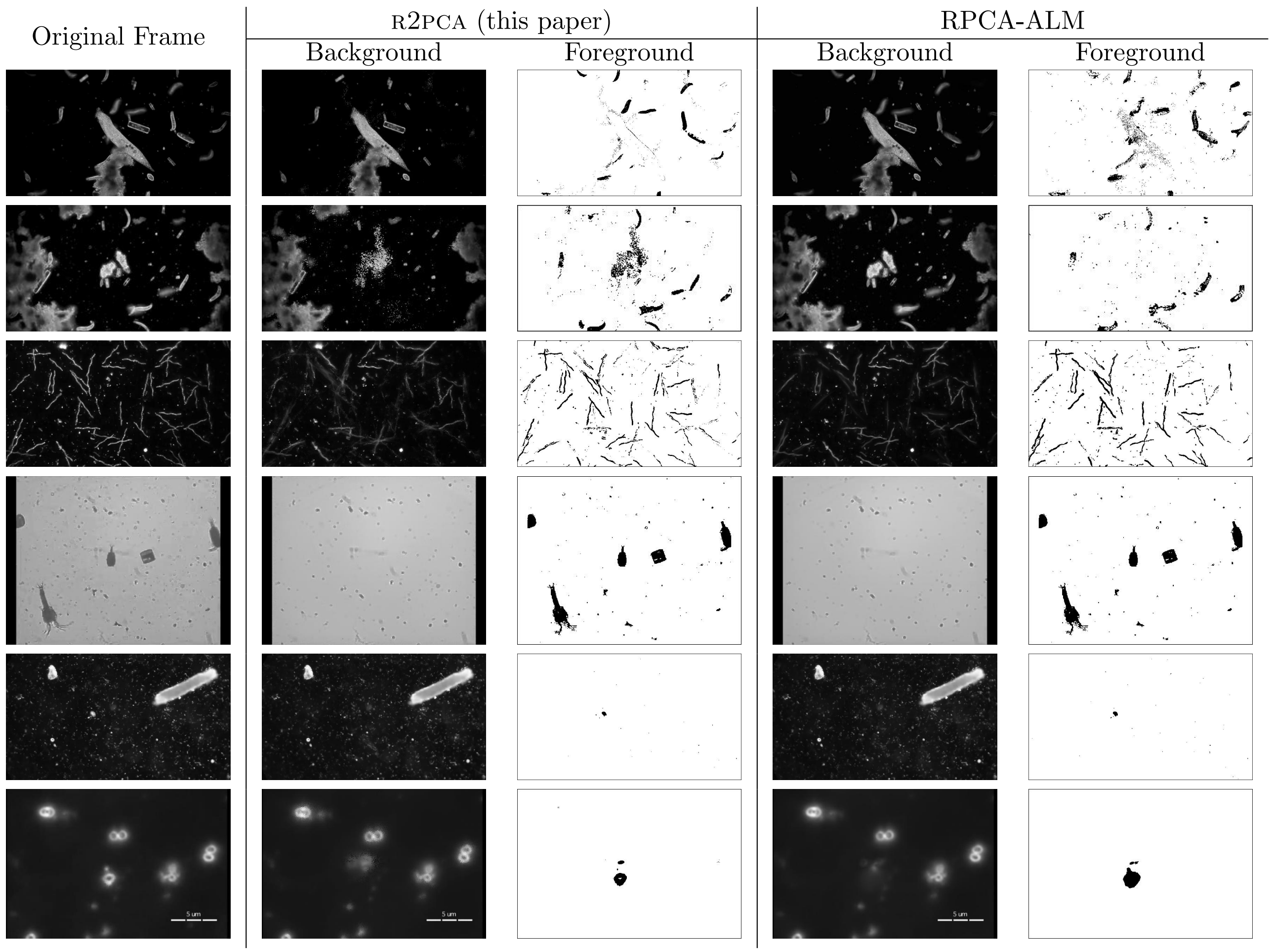

Figure 1: Sparse (foreground) plus low-rank (background) decomposition of some video frames from several microscopy videos from the Internet [25] using R2PCA and RPCA-ALM [18,19]. Notice that the background obtained by RPCA-ALM contains foreground objects, while the background obtained by R2PCA is much cleaner. This is because it these videos the background is mostly dark with a few bright regions (which implies a highly coherent subspace) and the location of the errors is highly correlated (the location of an object in consecutive frames is very similar). In contrast to other optimization methods [5-12,18,19], we make no assumptions about coherence or the distribution of the sparse errors, and so this does not affect our results.

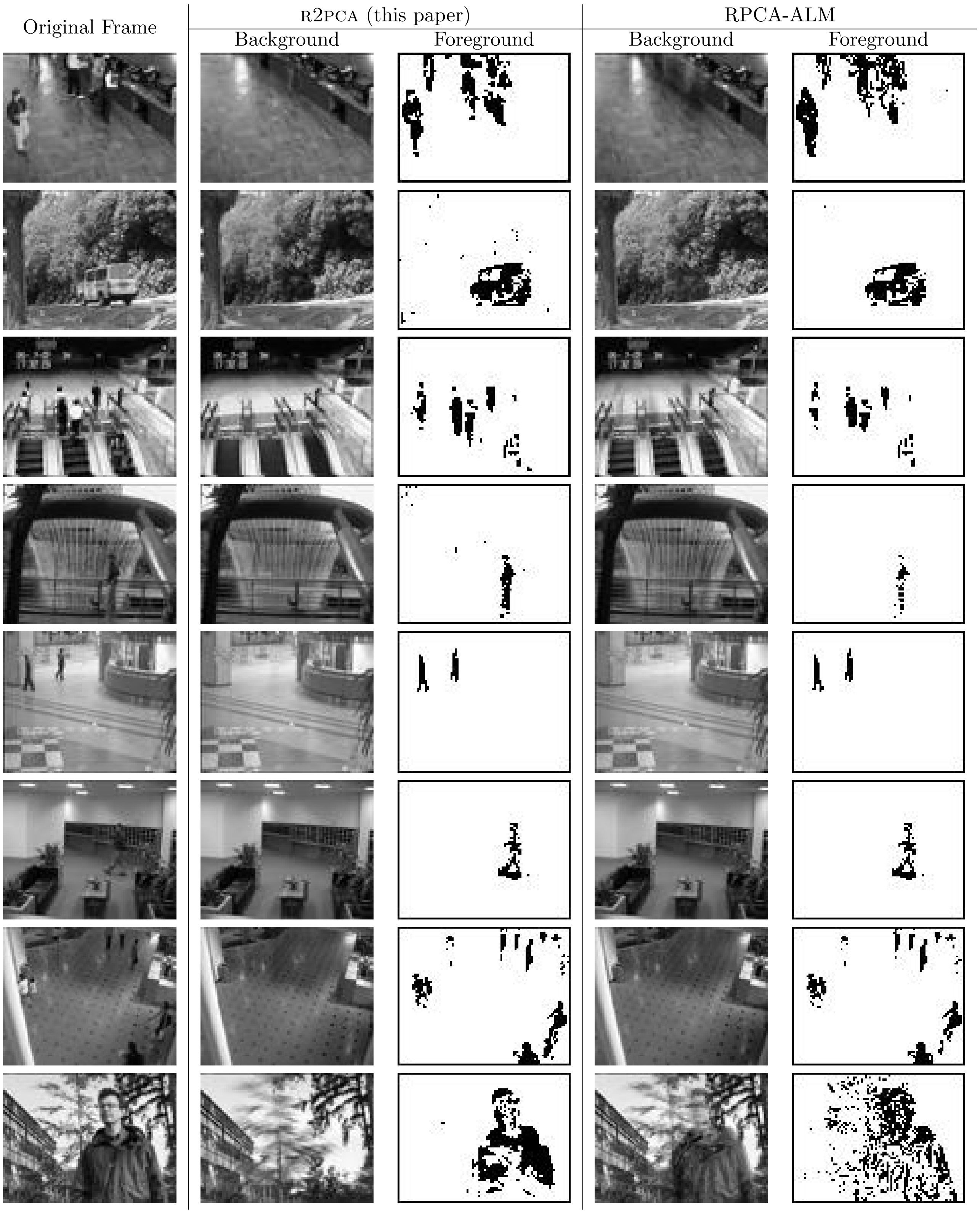

Figure 2: Sparse (foreground) plus low-rank (background) decomposition of some video frames from the Wallflower [23] and I2R [24] datasets using r2pca and RPCA-ALM [18,19].