Supplementary Material:
Fast Randomized Singular Value Thresholding for Nuclear Norm Minimization

Tae-Hyun Oh
KAIST
thoh.kaist.ac.kr@gmail.com

Yasuyuki Matsushita
Microsoft Research Asia / Osaka University
yasumat@ist.osaka-u.ac.jp

Yu-Wing Tai
KAIST
yuwing@gmail.com

In So Kweon
KAIST
iskweon77@kaist.ac.kr

Supplementary Material

In this supplementary material, we prove Proposition 1 in detail. In addition, we present additional experimental results of synthetic dataset and real-world applications that do not appear in the main paper due to space limitation. Movie clips for weather artifact removal and face stabilization are provided in the multimedia supplementary material. All the parameters specifically not mentioned here in this material are same with in the main paper or the referred papers.

1. Summary of Notations

$\pi_\Psi(\cdot)$	Orthogonal projection operator with a map (index set) Ψ
$\mathbb{S}_\tau(\cdot)$	SVT operator with the parameter τ
$\mathcal{S}_\tau(\cdot)$	Soft shrinkage operator [10] with the parameter τ
$\sigma_i(\cdot)$	i-th singular value of a matrix
$\mathbf{\Omega}$	A standard Gaussian random matrix
$\mathbf{Q}$	Orthonormal column matrix
$\mathbf{O}$	Observation matrix (Input)
$\mathbf{L}, \mathcal{L}$	Low-rank optimization related matrix or tensor
$\mathbf{S}, \mathcal{E}$	Sparse optimization related matrix or tensor
k	Parameter for the target rank
p	Parameter for over-sampling rate
l	Sampling (predicted) rate ($l = k + p$)
r	True rank of the target matrix $\mathbf{A}$
s	Real sampling rate ($s = \min(l, r)$)
η	Number of the power iteration
b	Parameter for the maximum rank bound
$\mathbf{v}$	Parameter vector for the polynomial error bound ($\mathbf{v} = \{k, p\}$ without the power iteration, or $\mathbf{v} = \{k, p, \eta\}$ with it)

2. Proof of Proposition 1.

Proposition 1. *Let $\mathbf{A} = \mathbf{QB} \in \mathbb{R}^{m \times n}$, where $\mathbf{Q} \in \mathbb{R}^{m \times n}$ has orthonormal columns. Then, the following equality holds:*

$$\mathbb{S}_\tau(\mathbf{A}) = \mathbf{Q}\,\mathbb{S}_\tau(\mathbf{B}),$$

where $\mathbb{S}_\tau(\cdot)$ is the SVT operator.

Proof. When $m = n$ and $\mathbf{Q}$ is an orthonormal matrix, the equality obviously holds by unitary invariant property of norms.

For $m > n$, we prove the equality for the orthonormal column matrix. Consider an arbitrary matrix $\mathbf{Z} = \mathbf{U_Z \Sigma_Z V_Z^\top} \in \mathbb{R}^{n\times n}$, where $\mathbf{U_Z \Sigma_Z V_Z^\top}$ is its SVD, then

$$\begin{aligned}
\mathbf{Q}\cdot \mathbb{S}_\tau(\mathbf{B}) &= \mathbf{Q}\cdot \operatorname*{argmin}_{\mathbf{Z}} \left(\tau\|\mathbf{Z}\|_* + \frac{1}{2}\|\mathbf{Z}-\mathbf{B}\|_F^2\right)\\
&= \operatorname*{argmin}_{\mathbf{QZ}=\mathbf{X}} \left(\tau\|\mathbf{Z}\|_* + \frac{1}{2}\|\mathbf{Z}-\mathbf{B}\|_F^2\right)\\
&= \operatorname*{argmin}_{\mathbf{X}} \tau\|\mathbf{Q}^\top\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{Q}(\mathbf{Z}-\mathbf{B})\|_F^2\\
&\text{(since } \mathbf{Q}^\top\mathbf{Q} = \mathtt{I}\text{, and the unitary invariant norm property)}\\
&= \operatorname*{argmin}_{\mathbf{X}} \tau\|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X}-\mathbf{QB}\|_F^2\\
&\text{(since } \|\mathbf{Q}^\top\mathbf{X}\|_* = \|\mathbf{Z}\|_* = \|\mathbf{QZ}\|_* = \|\mathbf{X}\|_*)\\
&= \mathbb{S}_\tau(\mathbf{QB}) = \mathbb{S}_\tau(\mathbf{A}).
\end{aligned}$$

□

3. Experiment with Synthetic Data

The following lists comparison methods. All the parallelization techniques are not used and turned off for the fair comparison.

$\text{SVD}_{\text{Matlab 2010a}}$	Built-in `SVD(·, 'econ')` is based on Intel MKL 10.2.2 and LAPACK 3.2.1.
$\text{SVD}_{\text{Matlab 2014a}}$	Built-in `SVD(·, 'econ')` is based on LAPACK 3.4.1 in Intel MKL 11.0.5.
LTSVD*	The Matlab implementation by Ma *et al.* [25] is used.
Lanczos	The implementation optimized by Lin *et al.* [16] is used. It is implemented by Matlab and Mex to use LAPACK (ver. 3.4.1 in Intel MKL 11.0.5 is used).
BLWS	The Matlab implementation by Lin *et al.* [19] is used.
Mu *et al.*	The Matlab implementation provided by Mu *et al.* [26] is used. All the parameters is directly used as suggested.
FSVT ***	We implement FSVT by Matlab and Mex based on Intel MKL 11.0.5, and the speed-up gain is checked and consistent with the reported results in Cai *et al.* [2]. All the parameters are directly used as suggested.
RSVD ***	We implement RSVD [11] based on LAPACK 3.4.1 in Intel MKL 11.0.5. by referring to RedSVD**.
FRSVT ***	Our Matlab implementation based on Intel MKL 11.0.5. Partially, we implement Mex functions for QR–CP (`dgeqp3`) and Polar Decomposition with BLAS and LAPACK 3.4.1 routines.

* LTSVD: In the implementation of [25], the leverage score estimation suggested by the original paper [5] is omitted, which is required for the importance sampling. So we implement it with $O(mn)$ complexity.
** RedSVD (`http://code.google.com/p/redsvd/`): Since RedSVD is implemented based on Eigen 3.0-b1, without the power iteration, so the original implementation is not fast and inaccurate. Thus, we re-implement RSVD [11] by just referring to RedSVD (2-sides random projection).
*** For FSVT, RSVD and FRSVT implementation, we utilize Armadillo library [33] as a high level wrapper for LAPACK and BLAS of Intel MKL in our Mex implementation part. In the Mex functions of Lanczos, Lin *et al.* [16] directly call LAPACK routines without any external library.

3.1. Single SVT test

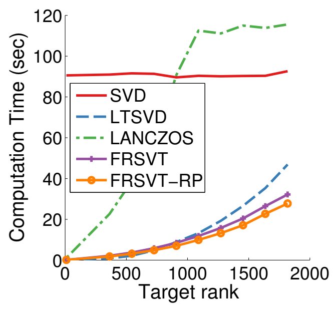
The comparisons are shown in Figs. 5 and 6.

Parameters for Single SVT test

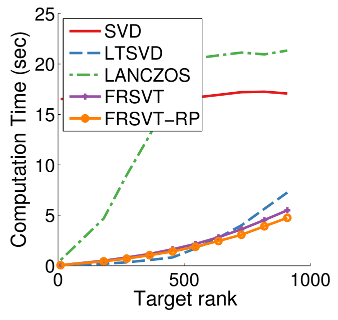
– Gaussian random matrix: drawn from $N(0, 1)$.
– $\tau = 1/\sqrt{\|\mathbf{A}\|_2}$.
– The number of repeats: 3 times.
– Over-sampling rate p:

	Lanczos	LTSVD	FRSVT
‘ p	2	$2k$	5

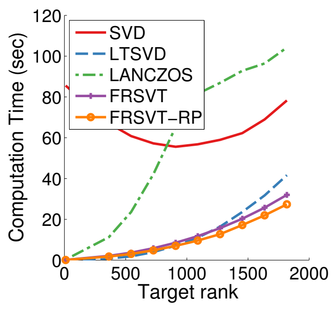
– Power iteration $q = 2$.



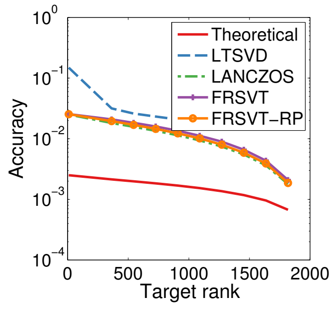
(a) Full rank 2000×2000.



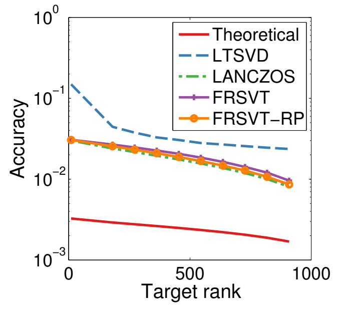
(b) Full rank 2000×1000.



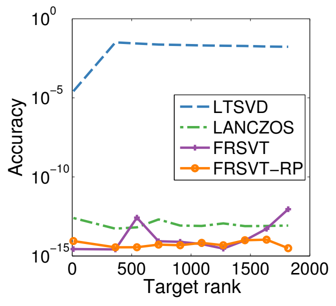
(c) Low rank 2000×2000.



(d) Full rank 2000×2000.

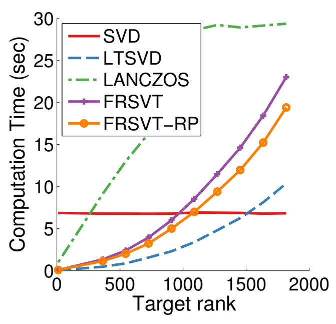


(e) Full rank 2000×1000.

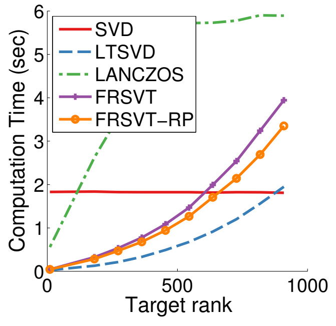


(f) Low rank 2000×2000.

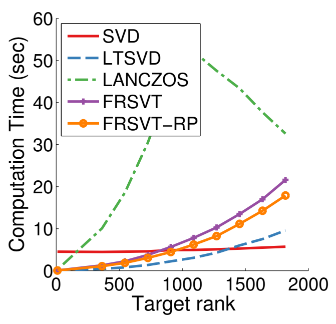
Figure 5. SVT comparisons according to SVDs on Matlab 2010a. [Top] X-axis: Target rank, Y-axis: Elapsed time (Sec). [Bottom] X-axis: Target rank, Y-axis: $\|\mathbf{A}^* - \hat{\mathbf{A}}_k\|_F / \|\mathbf{A}^*\|_F$, where $\mathbf{A}^* = \mathbb{S}_\tau(\mathbf{D}_{GT})$, $\hat{\mathbf{A}}_k = \mathbb{S}_\tau(\mathbf{D}_k)$, $\mathbf{D}_{GT}$ is the input data, and $\mathbf{D}_k$ is approximated by each method. For the low-rank matrix tests, we generate input data matrices of which rank corresponds to the target rank.



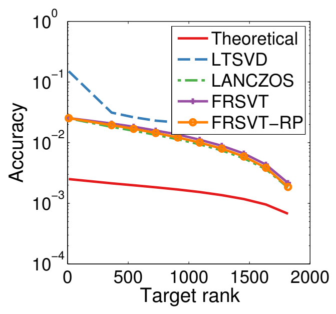
(a) Full rank 2000 × 2000.



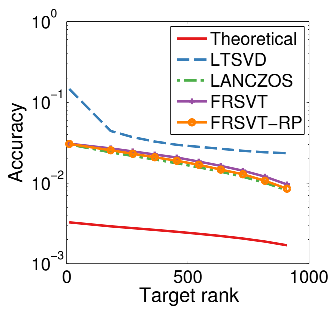
(b) Full rank 2000 × 1000.



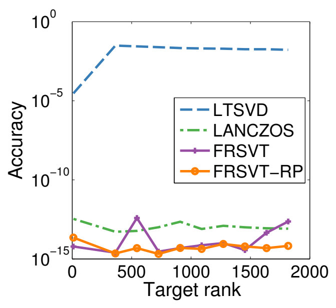
(c) Low rank 2000 × 2000.



(d) Full rank 2000 × 2000.



(e) Full rank 2000 × 1000.



(f) Low rank 2000 × 2000.

Figure 6. SVT comparisons according to SVDs on Matlab 2014a. [Top] X-axis: Target rank, Y-axis: Elapsed time (Sec). [Bottom] X-axis: Target rank, Y-axis: $\|\mathbf{A}^* - \hat{\mathbf{A}}_k\|_F / \|\mathbf{A}^*\|_F$, where $\mathbf{A}^* = \mathbb{S}_\tau(\mathbf{D}_{GT})$, $\hat{\mathbf{A}}_k = \mathbb{S}_\tau(\mathbf{D}_k)$, $\mathbf{D}_{GT}$ is the input data, and $\mathbf{D}_k$ is approximated by each method. For the low-rank matrix tests, we generate input data matrices of which rank corresponds to the target rank.

3.2. RPCA test

The comparisons on RPCA application are shown in Figs. 7, 8 and 9.

Parameters for RPCA test

– $\lambda = \frac{1}{\sqrt{\max(m,n)}}$

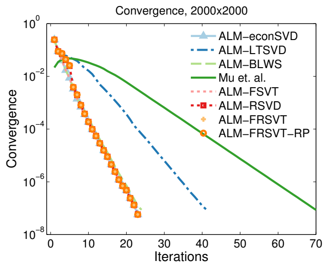
– All other parameters for FRSVT are same as mentioned in Sec. 3 and with the below table.

Table 4. Parameter settings for convergence evaluation of RPCA. (p: Over-sampling rate, AP: Adaptive rank prediction, PI: The number of power iterations η.)

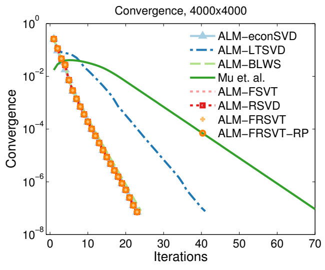
Methods	p	AP	PI
iALM$_{\text{econSVD}}$	–	×	–
iALM$_{\text{LTSVD}}$	+5	◯	–
iALM$_{\text{BLWS}}$	–	◯	–
Mu *et al.* [26]	–	×	–
iALM$_{\text{FSVT}}$	–	×	–
iALM$_{\text{RSVD}}$	+5	◯	2
iALM$_{\text{FRSVT}}$	+2	◯	2
iALM$_{\text{FRSVT-RP}}$	+2	◯	2

Figure 7. Quantitative comparisons on RPCA application. For Mu *et al*. [26], notice that we use the implementation and parameters provided by the authors. Since Mu *et al*. is based on the exact ALM algorithm [17], while other methods are based on the inexact ALM [17], we report the times for a single outer iteration of Mu *et al*. as 'Avg. time for a single iteration'. In addition, the highly stochastic property of the randomly projected nuclear norm makes the algorithm [26] very sensitive to data and does not preserve the singular values and sparsity structure of the matrix. This is consistent results with an another stochastic method, LTSVD (See and compare $\|\mathbf{S}\|_0$ of LTSVD and Mu *et al*.).

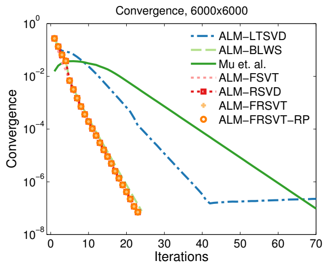
	iALM$_{\text{econSVD}}$ Matlab 2010a	iALM$_{\text{econSVD}}$ Matlab 2014a	iALM$_{\text{LTSVD}}$	iALM$_{\text{BLWS}}$	Mu *et al.*	iALM$_{\text{FSVT}}$	iALM$_{\text{RSVD}}$	iALM$_{\text{FRSVT-RP}}$
	$\|\mathbf{S}\|_0$							
10000×100	99,999		879,415	99,999	999,999	99,999	99,999	99,999
2000×2000	399,997		994,479	399,998	3,998,401	399,997	399,997	399,997
4000×4000	1,599,981		5,607,211	1,599,984	15,995,318	1,599,981	1,599,981	1,599,981
6000×6000	–	3,599,952	35,662,548	3,599,966	35,990,553	3,599,952	3,599,952	3,599,952
	Accuracy							
10000×100	7.34e-05		3.39e-02	2.66e-04	7.65e-01	7.34e-05	7.31e-05	9.33e-05
2000×2000	2.11e-07		7.77e-07	1.03e-06	1.10e+00	2.11e-07	2.11e-07	2.11e-07
4000×4000	1.81e-07		4.59e-07	4.83e-07	1.37e+00	1.81e-07	1.81e-07	1.81e-07
6000×6000	–	1.42e-07	7.76e-05	3.83e-07	1.49e+00	1.42e-07	1.43e-07	1.43e-07
	Total Time							
10000×100	3.217	1.982	1.722	1.317	125.368	2.436	0.970	0.782
2000×2000	2751.910	147.055	11.075	9.808	118.798	348.132	12.497	5.123
4000×4000	21668.280	1068.863	69.547	53.468	759.794	2527.141	80.705	30.274
6000×6000	–	3360.379	203.051	164.752	2385.272	8238.109	149.204	88.132
	Avg. time for a single iteration							
10000×100	0.140	0.086	0.039	0.049	1.687	0.104	0.042	0.034
2000×2000	119.415	6.393	0.274	0.393	1.667	15.189	0.557	0.226
4000×4000	947.908	46.481	1.726	2.185	10.664	110.006	3.623	1.348
6000×6000	–	146.192	4.295	6.848	33.505	358.899	6.681	3.942
	Total no. iteration							
10000×100	23		43	24	73	23	23	23
2000×2000	23		41	24	70	23	23	23
4000×4000	23		41	24	70	23	23	23
6000×6000	–	23	48	24	70	23	23	23



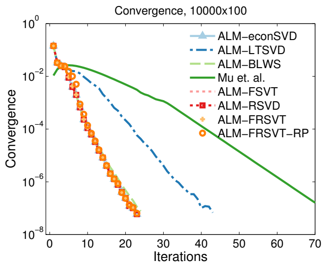
(a) 2000×2000



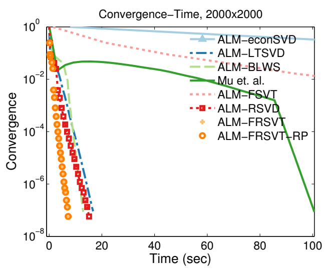
(b) 4000×4000



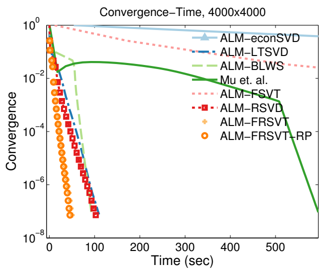
(c) 6000×6000



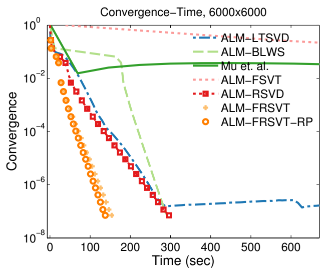
(d) 10000×100



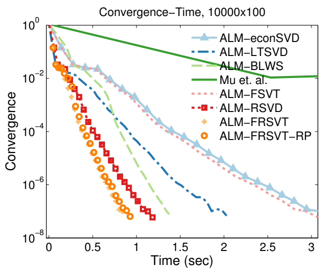
(e) 2000×2000



(f) 4000×4000

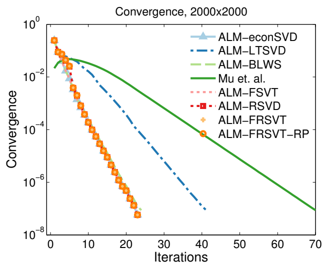


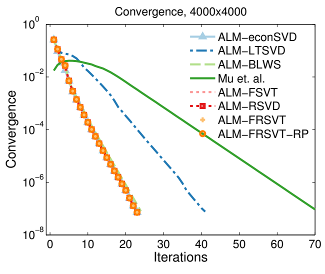
(g) 6000×6000

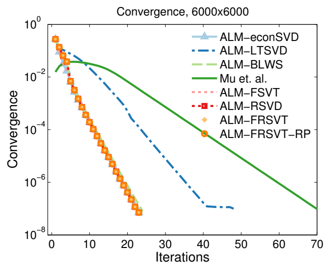


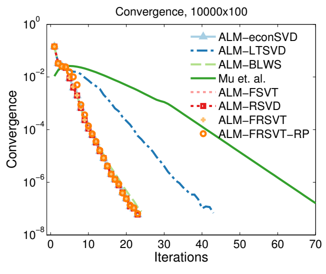
(h) 10000×100

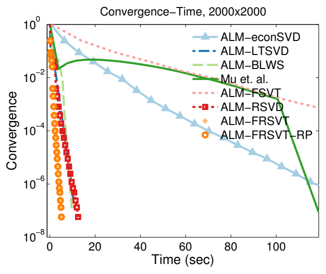
Figure 8. RPCA comparisons on Matlab 2010a. [Top] X-axis: The number of iterations, Y-axis: The stop criterion. [Bottom] X-axis: Elapsed time (Sec), Y-axis: The stop criterion. Except Mu *et al*.is based on the exact ALM method, all the methods are based on the inexact ALM.

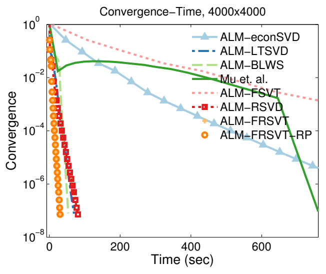

(a) 2000 × 2000

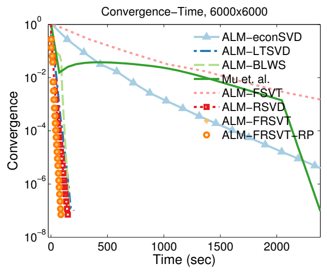

(b) 4000 × 4000


(c) 6000 × 6000


(d) 10000 × 100


(e) 2000 × 2000


(f) 4000 × 4000


(g) 6000 × 6000

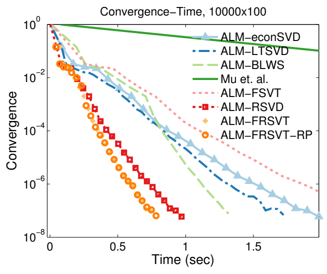

(h) 10000 × 100

Figure 9. RPCA comparisons on Matlab 2014a. [Left] X-axis: The number of iterations, Y-axis: The stop criterion. [Right] X-axis: Elapsed time (Sec), Y-axis: The stop criterion. Except Mu *et al.*is based on the exact ALM method, all the methods are based on the inexact ALM.

4. More Applications

4.1. Robust Subspace Clustering by Low-Rank Representation (LRR)

Detailed computation time comparison is shown in Fig. 10.

Parameters

– All parameters for FRSVT are same as mentioned in Sec. 3.

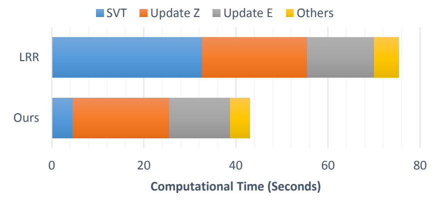


Figure 10. Computation time comparison on the subspace clustering application [20]. Computation times are measured for face clustering on Extended Yale Database B [15].

4.2. Semi-Online Weather Artifact Removal

The computation times with different settings of sliding window sizes are shown in Figs. 11, 12 and 13.

Parameters

– $\lambda = \frac{1}{\sqrt{\max(m,n)}}$

– Target rank for PSVT: rank-1.

– Max rank-l for computing the partial SVD:

$l = 2$, when $n = 3$.

$l = 3$, when $n = 5, 7$ and 10.

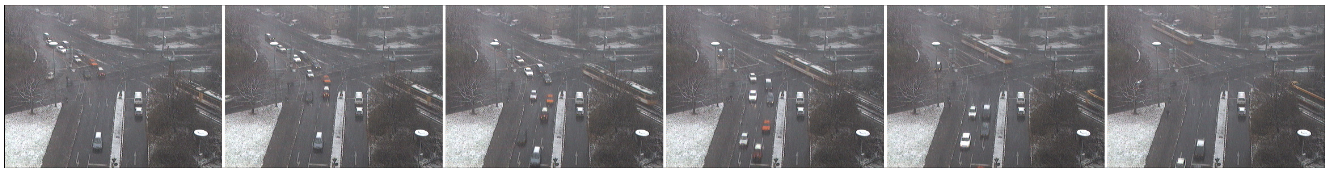
– Power iteration $q = 2$.

Computational Time (ms)

Sliding Window n	3	5	7	10
Elapsed Time	71.5	65.5	68.2	66.2

Figure 11. Computation times of the FRSVT based semi-online weather artifact removal. The elapsed times are measured for a single color channel. Since our algorithm produces n results simultaneously for n images at an execution, we report the time for an single image for a single channel. This results show that, when the number of inputs is small, our weather artifact removal algorithm has linear complexity according to the length of a sliding window, thus the elapsed times for a single image are similar.

(a) Sampled inputs


(b) Low-rank images $\mathbf{L}$ obtained by Oh *et al*. [27] based on SVD

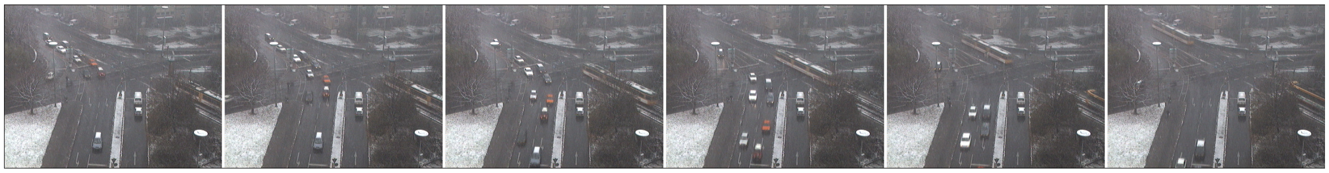

(c) Low-rank images $\mathbf{L}$ obtained by Oh *et al*. [27] based on our FRSVT

Figure 12. Sampled results for the weather artifact removal with the sliding window length 5.

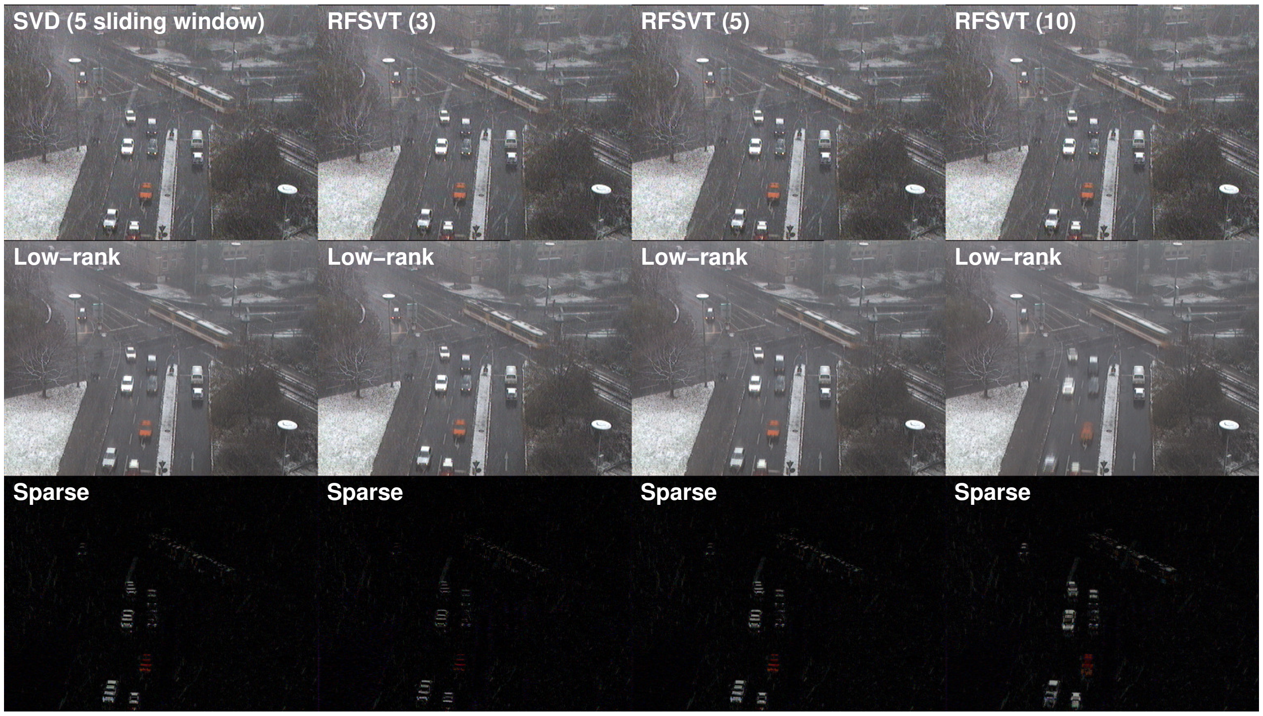


Figure 13. Comparisons according to the sliding window length $\{3, 5, 10\}$. [Top] Input images. [Middle] Low-rank images, $\mathbf{L}$. [Bottom] Absolute of sparse images, $\text{abs}(\mathbf{S})$.

4.3. Simultaneously Alignment and Rectification

The qualitative comparisons are shown in Figs. 14, 15 and 16.

Parameters

– $\alpha = [0.1, 0.1, 0.8]$ for *Digits3, Windows*, and *Al Gore*.
$\alpha = [0.15, 0.15, 0.7]$ for *Gait Challenge*.
– $\lambda = \frac{0.5}{\sqrt{width \cdot height}}$
– Other parameters for FRSVT are same as mentioned in Sec. 3.

Data Generation for Gait Dataset

We use USF Human ID "Gait Challenge" data sets version 1.7. There are 731 gait samples and each sample has 10 sampled gait cycles, and we only use first 300 gait samples. We resize the images into 32×22, so the size of the input tensor is $\mathcal{O} \in \mathbb{R}^{32\times22\times3000}$. The synthetic misalignment generation parameters are described in the below.

Randomly drawn Euclidean (3 DoF) geometric transformations are applied to each gait image. The used parameters are:

– Scale factor: Fixed as 1.
– Angle noise: $N(\frac{10\pi}{180}, (\frac{5\pi}{180})^2)$.
– Translation noise: $N(0, 0.5^2) + 2 \cdot (U(0,1) - 0.5)$.
($N(\mu, \sigma^2)$ denotes Gaussian distribution with the mean μ and the variance σ^2. $U(a, b)$ denotes Uniform distribution defined on $[a, b]$.)

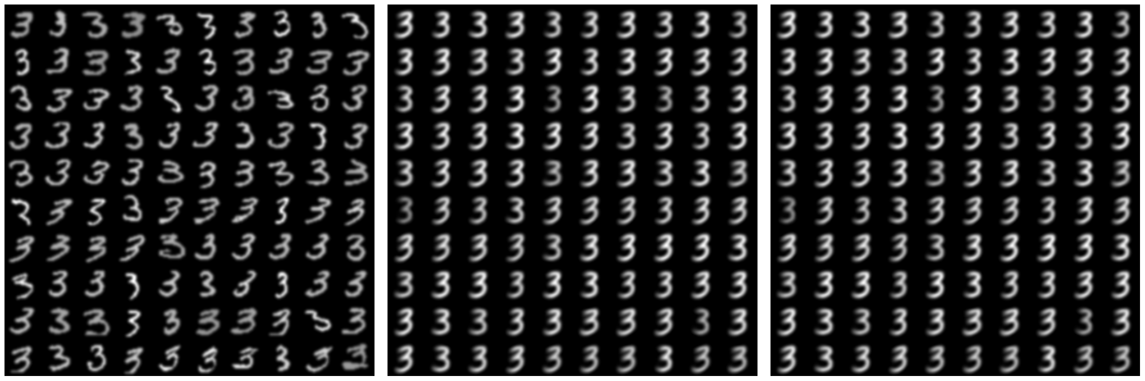

(a) Input (b) SRALT [37] (c) SRALT based on our FRSVT

Figure 14. Simultaneously alignment and rectification comparisons on *Digits3* data [31].

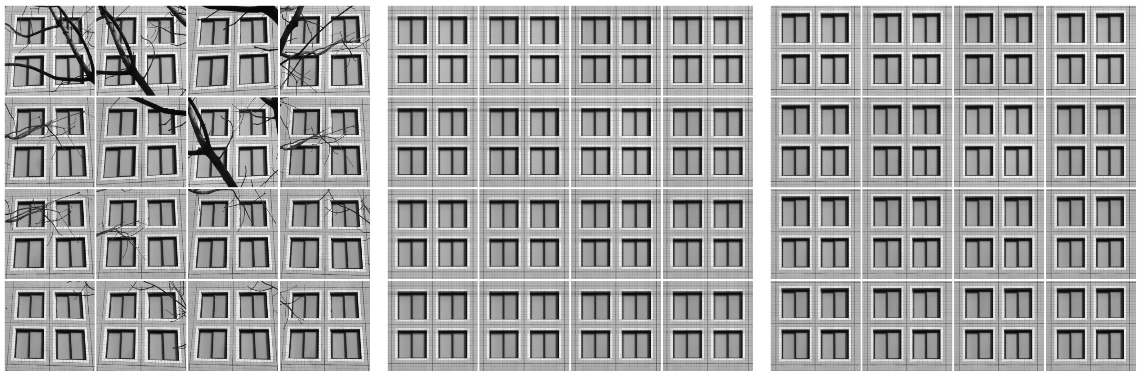

(a) Input (b) SRALT [37] (c) SRALT based on our FRSVT

Figure 15. Simultaneously alignment and rectification comparisons on *Windows* data [31].

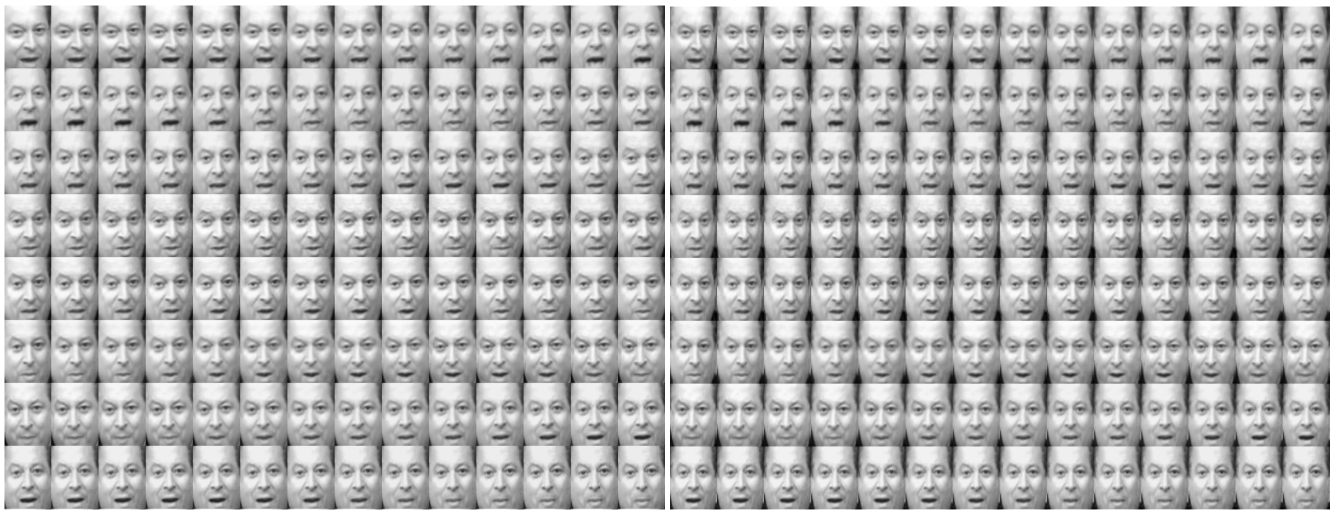

(a) RASL [31] (b) SRALT [37] + Our FRSVT

Figure 16. Simultaneously alignment and rectification comparisons on *Al Gore* data [31].