Robust Principal Component Analysis?

Emmanuel Candès

IPAM's 10th Anniversary Conference, UCLA, November 2010

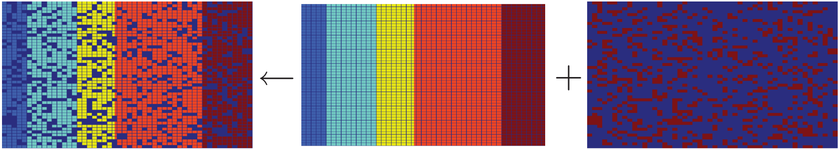
Collaborators

- Xiaodong Li (Stanford)
- Yi Ma (Microsoft Research Asia & UIUC)
- John Wright (Microsoft Research Asia)

Agenda

- A separation problem
- Computer vision applications

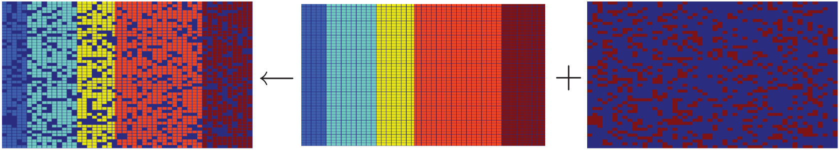
The separation problem



$$M = L_0 + S_0$$

- M: data matrix (observed)
- L_0: low-rank (unobserved)
- S_0: sparse (unobserved)

The separation problem



$$M = L_0 + S_0$$

- M: data matrix (observed)
- L_0: low-rank (unobserved)
- S_0: sparse (unobserved)

Problem: can we recover L_0 and S_0 accurately?

Seems daunting but solution would be really great!

Motivation

Classical PCA

$$M = L_0 + N_0$$

- L_0: low-rank (unobserved)
- N_0: (small) perturbation

Classical PCA

$$M = L_0 + N_0$$

- L_0: low-rank (unobserved)
- N_0: (small) perturbation

Dimensionality reduction (Schmidt 1907, Hotelling 1933)

$$\begin{array}{ll} \text{minimize} & \|M - L\| \\ \text{subject to} & \text{rank}(L) \le k \end{array}$$

Classical PCA

$$M = L_0 + N_0$$

- L_0: low-rank (unobserved)
- N_0: (small) perturbation

Dimensionality reduction (Schmidt 1907, Hotelling 1933)

$$\begin{array}{ll} \text{minimize} & \|M - L\| \\ \text{subject to} & \text{rank}(L) \le k \end{array}$$

Solution given by truncated SVD

$$M = U\Sigma V^* = \sum_i \sigma_i u_i v_i^* \quad \Rightarrow \quad L = \sum_{i \le k} \sigma_i u_i v_i^*$$

Classical PCA

$$M = L_0 + N_0$$

- L_0: low-rank (unobserved)
- N_0: (small) perturbation

Dimensionality reduction (Schmidt 1907, Hotelling 1933)

$$\begin{array}{ll} \text{minimize} & \|M - L\| \\ \text{subject to} & \text{rank}(L) \le k \end{array}$$

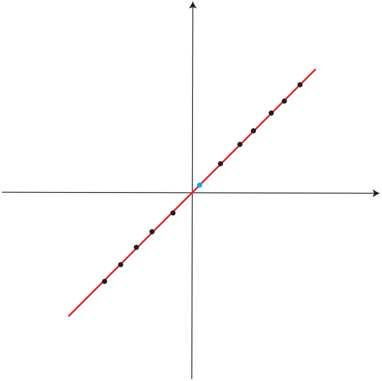
Solution given by truncated SVD

$$M = U\Sigma V^* = \sum_i \sigma_i u_i v_i^* \quad \Rightarrow \quad L = \sum_{i \le k} \sigma_i u_i v_i^*$$

Fundamental statistical tool: enormous impact

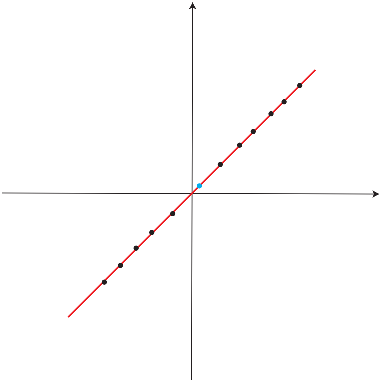
PCA and corruptions/outliers

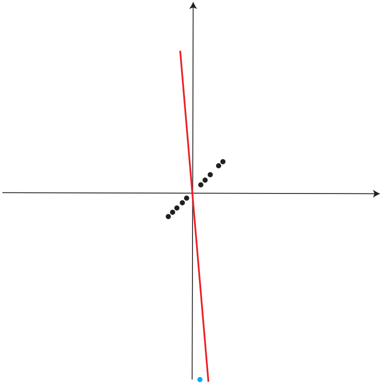
PCA: very sensitive to outliers



PCA and corruptions/outliers

PCA: very sensitive to outliers





Breaks down with one (badly) corrupted data point

Robust PCA

Gross errors frequently occur in many applications

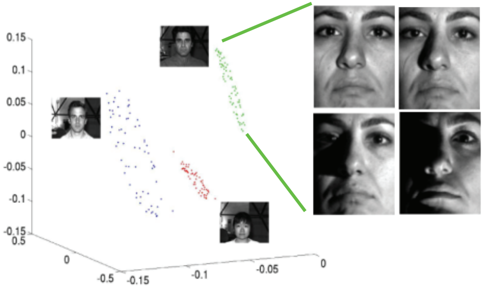
- Image processing
- Web data analysis
- Bioinformatics
- ...

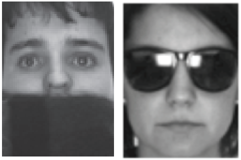
- Occlusions
- Malicious tampering
- Sensor failures
- ...

Important to make PCA robust

- Influence function techniques: Huber; De La Torre and Black
- Multivariate trimming: Gnanadesikan and Kettenring
- Alternating minimization: Ke and Kanade
- Random sampling techniques: Fischler and Bolles
- ...

Occlusions in computer vision





An interesting separation problem

Recover low-rank L_0 and sparse S_0 from

$$M = L_0 + S_0$$

Many applications other than robust PCA: informative component may be

- L_0 (RPCA)
- S_0 (examples to follow)

Video surveillance

Sequence of video frames with a static background

Problem: detect any activity in the foreground

Video surveillance

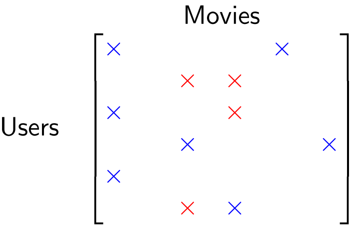
Sequence of video frames with a static background

Problem: detect any activity in the foreground

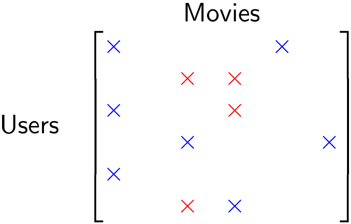
$$M = L_0 + S_0$$

This is a separation problem!

Ranking and collaborative filtering



Ranking and collaborative filtering



$$M = L_0 + S_0$$

- Available data $M_{ij} : (i, j) \in \Omega_{\text{obs}}$
- L_0 : all users' ratings (what we would like to know)
- S_0 : ratings that have been tampered with

Other applications

- Face recognition
- System identification
- Quantum-state tomography (Gross)
- Graphical modeling with latent variables (Chandrasekaran, Parrilo, Willsky)

Theoretical aspects

Principal Component Pursuit (PCP)

$$M = L_0 + S_0$$

- L_0 unknown (rank unknown)
- S_0 unknown ($\#$ of entries $\neq 0$, locations, magnitudes all unknown)

Principal Component Pursuit (PCP)

$$M = L_0 + S_0$$

- L_0 unknown (rank unknown)
- S_0 unknown ($\#$ of entries $\neq 0$, locations, magnitudes all unknown)

Recovery via (convex) PCP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & L + S = M \end{array}$$

See also Chandrasekaran, Sanghavi, Parrilo, Willsky ('09)

- nuclear norm: $\|L\|_* = \sum_i \sigma_i(L)$ (sum of sing. values)
- ℓ_1 norm: $\|S\|_1 = \sum_{ij} |S_{ij}|$ (sum of abs. values)

Principal Component Pursuit (PCP)

$$M = L_0 + S_0$$

- L_0 unknown (rank unknown)
- S_0 unknown ($\#$ of entries $\neq 0$, locations, magnitudes all unknown)

Recovery via (convex) PCP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & L + S = M \end{array}$$

See also Chandrasekaran, Sanghavi, Parrilo, Willsky ('09)

- nuclear norm: $\|L\|_* = \sum_i \sigma_i(L)$ (sum of sing. values)
- ℓ_1 norm: $\|S\|_1 = \sum_{ij} |S_{ij}|$ (sum of abs. values)

- Nuclear norm heuristics introduced in 90's
- ℓ_1 norm heuristics introduced in 50's

Surprise

$$M = L_0 + S_0$$

- L_0 unknown
- S_0 unknown

Recovery via

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & L + S = M \end{array}$$

Surprise

$$M = L_0 + S_0$$

- L_0 unknown
- S_0 unknown

Recovery via

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & L + S = M \end{array}$$

Under broad conditions, solution $(\hat{L}, \hat{S})$ obeys

$$\hat{L} = L_0, \quad \hat{S} = S_0!$$

When does separation make sense?

M cannot be low-rank and sparse

$$\boldsymbol{M} = \boldsymbol{e}_1 \boldsymbol{e}_n^* = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Low-rank component cannot be sparse

$$L_0 \in \mathbb{R}^{n\times n} = U\Sigma V^* = \sum_{1\le i\le r} \sigma_i u_i v_i^* \quad r = \mathsf{rank}(L_0)$$

Coherence condition (C. and Recht, '08): $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$

$$\|U^* e_i\|^2 \le \frac{\mu r}{n} \quad \|V^* e_i\|^2 \le \frac{\mu r}{n}$$

and

$$|UV^*|_{ij}^2 \le \frac{\mu r}{n^2}$$

Roughly: singular vectors (PC's) are not sparse/spiky

What if the sparse component has low-rank?

Example: first column of S_0 is that of L_0

$$S_0 = \begin{bmatrix} * & 0 & \cdots & 0 & 0 \\ * & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & 0 & \cdots & 0 & 0 \end{bmatrix} \quad \Rightarrow \quad M_0 = L_0 - S_0 = \begin{bmatrix} 0 & * & \cdots & * & * \\ 0 & * & \cdots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & \cdots & * & * \end{bmatrix}$$

What if the sparse component has low-rank?

Example: first column of S_0 is that of L_0

$$S_0 = \begin{bmatrix} * & 0 & \cdots & 0 & 0 \\ * & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ * & 0 & \cdots & 0 & 0 \end{bmatrix} \quad \Rightarrow \quad M_0 = L_0 - S_0 = \begin{bmatrix} 0 & * & \cdots & * & * \\ 0 & * & \cdots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & \cdots & * & * \end{bmatrix}$$

Sparsity pattern will be assumed (uniform) random

Main result: $M = L_0 + S_0$

Theorem

- L_0 *is* $n \times n$ *of rank*$(L_0) \leq \rho_r n\, \mu^{-1} (\log n)^{-2}$
- S_0 *is* $n \times n$*, random sparsity pattern of cardinality* $m \leq \rho_s n^2$

Then with probability $1 - O(n^{-10})$*, PCP with* $\lambda = 1/\sqrt{n}$ *is exact:*

$$\hat{L} = L_0, \quad \hat{S} = S_0$$

Same conclusion for rectangular matrices with $\lambda = 1/\sqrt{\textit{max dim}}$

- Exact
  - whatever the magnitudes of L_0!
  - whatever the magnitudes of S_0!
- No tuning parameter!

Can achieve stronger probabilities of success, e. g. $1 - O(n^{-\beta})$, $\beta > 0$

Connections with matrix completion (MC)

Recover a (low-rank) matrix from a subset of its entries

- C. and Recht ('08)
- C. and Tao ('09)
- Keshavan, Montanari and Oh ('09)
- Mazumder, Hastie and Tibshirani ('09)
- Different problem: Recht, Fazel and Parrilo ('07)
- Many others

$$\begin{bmatrix} \times & ? & ? & ? & \times & ? \\ ? & ? & \times & \times & ? & ? \\ \times & ? & ? & \times & ? & ? \\ ? & ? & \times & ? & ? & \times \\ \times & ? & ? & ? & ? & ? \\ ? & ? & \times & \times & ? & ? \end{bmatrix}$$

Connections with matrix completion (MC)

$$\begin{array}{ll} \text{minimize} & \|L\|_* \\ \text{subject to} & L_{ij} = L^0_{ij} \ (i,j) \in \Omega_{\textsf{obs}} \end{array}$$

$$\begin{bmatrix} \times & ? & ? & ? & \times & ? \\ ? & ? & \times & \times & ? & ? \\ \times & ? & ? & \times & ? & ? \\ ? & ? & \times & ? & ? & \times \\ \times & ? & ? & ? & ? & ? \\ ? & ? & \times & \times & ? & ? \end{bmatrix}$$

Connections with matrix completion (MC)

$$\begin{array}{ll} \text{minimize} & \|L\|_* \\ \text{subject to} & L_{ij} = L^0_{ij} \ (i,j) \in \Omega_{\text{obs}} \end{array}$$

$$\begin{bmatrix} \times & ? & ? & ? & \times & ? \\ ? & ? & \times & \times & ? & ? \\ \times & ? & ? & \times & ? & ? \\ ? & ? & \times & ? & ? & \times \\ \times & ? & ? & ? & ? & ? \\ ? & ? & \times & \times & ? & ? \end{bmatrix}$$

Theorem (C. and Tao '09 improving C. and Recht '08)

- *rank*$(L_0) = r$ *and* L_0 *as before*
- Ω_{obs} *random set of size* m

Solution to SDP is exact with probability at least $1 - n^{-10}$ *if*

$$m \gtrsim \mu\, nr \log^a n \quad a \leq 6$$

Gross' near-optimal improvement

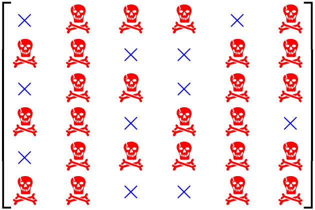
$$m \gtrsim \mu\, nr \log^2 n$$

Connections with matrix completion (MC)

Missing vs. corrupted data

$$\begin{bmatrix} \times & ? & ? & ? & \times & ? \\ ? & ? & \times & \times & ? & ? \\ \times & ? & ? & \times & ? & ? \\ ? & ? & \times & ? & ? & \times \\ \times & ? & ? & ? & ? & ? \\ ? & ? & \times & \times & ? & ? \end{bmatrix}$$

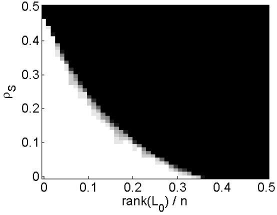
MC: missing



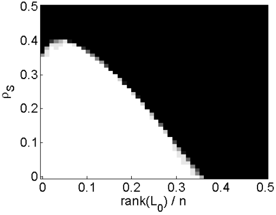
RPCA: corrupted

<u>Harder</u> to detect and correct than to fill in

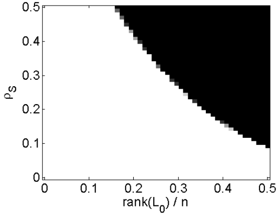
Phase transitions in probability of success



(a) PCP, Random Signs



(b) PCP, Coherent Signs



(c) Matrix Completion

$L_0 = XY^*$ is a product of independent $n \times r$ i.i.d. $\mathcal{N}(0, 1/n)$ matrices

Contemporary result: Chandrasekaran, Sanghavi, Parrilo, Willsky ('09)

Deterministic conditions for PCP to succeed

- $T(L_0)$: span of all matrices with row space included in that of L_0 or with col. space included in that of L_0

$$\xi(L_0) = \sup_{N \in T(L_0): \|N\| \leq 1} \|N\|_\infty$$

- $\Omega(S_0)$: span of all matrices with support included in that of S_0

$$\nu(S_0) = \sup_{N \in \Omega(S_0): \|N\|_\infty \leq 1} \|N\|$$

Contemporary result: Chandrasekaran, Sanghavi, Parrilo, Willsky ('09)

Deterministic conditions for PCP to succeed

- $T(L_0)$: span of all matrices with row space included in that of L_0 or with col. space included in that of L_0

$$\xi(L_0) = \sup_{N \in T(L_0): \|N\| \le 1} \|N\|_\infty$$

- $\Omega(S_0)$: span of all matrices with support included in that of S_0

$$\nu(S_0) = \sup_{N \in \Omega(S_0): \|N\|_\infty \le 1} \|N\|$$

Then PCP succeeds *for some* λ if

$$\xi(L_0)\, \nu(S_0) \le 1/6$$

Comparison for random sparsity patterns

Corollary: correct recovery if

$$\text{max number of corruptions per col.} \times \sqrt{\mu r/n} < 1/12$$

so fraction of corrupted entries must obey

$$\rho_s \leq \frac{1}{12}\sqrt{\frac{1}{\mu n r}}$$

Accommodate only vanishing fractions – even for rank-1 matrices

Comparison for random sparsity patterns

Corollary: correct recovery if

$$\text{max number of corruptions per col. } \times \sqrt{\mu r/n} < 1/12$$

so fraction of corrupted entries must obey

$$\rho_s \leq \frac{1}{12}\sqrt{\frac{1}{\mu n r}}$$

Accommodate only vanishing fractions – even for rank-1 matrices

Significant differences

- models, proofs: not much in common
- selection of λ

Matrix completion from grossly corrupted data

Entries may be both corrupted and missing

Matrix completion from grossly corrupted data

Entries may be both corrupted and missing

$$\text{(PCP)} \qquad \begin{array}{ll} \text{minimize} & \|L\|_* + \lambda\|S\|_1 \\ \text{subject to} & L_{ij} + S_{ij} = M_{ij}, \ (i,j) \in \Omega_{\text{obs}} \end{array}$$

Ω_{obs} locations of observed entries

Matrix completion from grossly corrupted data

Entries may be both corrupted and missing

$$\text{(PCP)} \quad \begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & L_{ij} + S_{ij} = M_{ij}, \ (i,j) \in \Omega_{\text{obs}} \end{array}$$

Ω_{obs} locations of observed entries

Theorem

- *L_0 is $n \times n$ as before, rank$(L_0) \le \rho_r n \, \mu^{-1} (\log n)^{-2}$*
- *Ω_{obs} random set of size[a] $m = 0.1n^2$*
- *each observed entry is corrupted with probability $\tau \le \tau_s$*

Then with probability $1 - O(n^{-10})$, PCP with $\lambda = 1/\sqrt{0.1n}$ is exact:

$$\hat{L} = L_0$$

Same conclusion for rectangular matrices with $\lambda = 1/\sqrt{0.1 \text{max dim}}$

[a] missing fraction is arbitrary

Simultaneous completion and correction!

A cute thing

If no corruption $\rightarrow$ MC problem

- MC: perfect recovery via

$$\begin{array}{ll} \text{minimize} & \|L\|_* \\ \text{subject to} & L_{ij} = L^0_{ij},\ (i,j) \in \Omega_{\text{obs}} \end{array}$$

- PCP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \frac{1}{\sqrt{n}} \|S\|_1 \\ \text{subject to} & L_{ij} + S_{ij} = L^0_{ij},\ (i,j) \in \Omega_{\text{obs}} \end{array}$$

A cute thing

If no corruption $\rightarrow$ MC problem

- MC: perfect recovery via

$$\begin{array}{ll} \text{minimize} & \|L\|_* \\ \text{subject to} & L_{ij} = L^0_{ij}, \ (i,j) \in \Omega_{\text{obs}} \end{array}$$

- PCP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \frac{1}{\sqrt{n}}\|S\|_1 \\ \text{subject to} & L_{ij} + S_{ij} = L^0_{ij}, \ (i,j) \in \Omega_{\text{obs}} \end{array}$$

Same answer! $\hat{S} = 0$

Methods of Proof

Find a dual variable certifying that (L_0, S_0) is solution to PCP

Existence is a deep question in probability theory

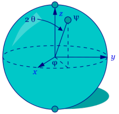
Methods of Proof

Find a dual variable certifying that (L_0, S_0) is solution to PCP

Existence is a deep question in probability theory

- Tools from Banach space theory (Rudelson's lemma, concentration of measure, noncommutative Khintchine inequality, ...)
- Arsenal of techniques developed for matrix completion (C. and Recht, 08)
- Important role played by Gross' golfing scheme ('09)

Quantum-state tomography



- k spin-$1/2$ system in an *unknown* quantum state $M \in \mathbb{C}^{n\times n}$ (density matrix)

$$n = 2^k, \quad \text{trace}(M) = 1, \quad M \succcurlyeq 0$$

- Quantum measurements (data)

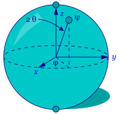
$$\mathbb{E}[\text{measurement with observable } A_j] = \langle A_j, M\rangle = \text{trace}(A_j^* M)$$

e.g. $\{A_j\}$: tensor Pauli matrices

Q? Can we reduce # measurements by using the structure of special classes of quantum states?

- pure state $\longrightarrow$ rank$(M) = 1$
- interesting mixed states $\longrightarrow$ (approx) low rank

Quantum-state tomography



- k spin-$1/2$ system in an *unknown* quantum state $M \in \mathbb{C}^{n\times n}$ (density matrix)

$$n = 2^k, \quad \text{trace}(M) = 1, \quad M \succcurlyeq 0$$

- Quantum measurements (data)

$$\mathbb{E}[\text{measurement with observable } A_j] = \langle A_j, M\rangle = \text{trace}(A_j^* M)$$

e.g. $\{A_j\}$: tensor Pauli matrices

Q? Can we reduce # measurements by using the structure of special classes of quantum states?

- pure state $\longrightarrow$ $\text{rank}(M) = 1$
- interesting mixed states $\longrightarrow$ (approx) low rank

A. Yes. Sample in proportion to the rank of the quantum state (Gross 09)

Computational aspects and simulations

Computational issues

Wish to solve the SDP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & L + S = M \end{array}$$

- Off-the-shelf algorithms (SDPT3, SeDuMi) need $n < 80, 100$
- Customized IPMs don't do much better

Have developed a simple and scalable algorithm via the Alternating Direction Method of Multipliers (ADMM)

Empirical performance II

n	$\text{rank}(L_0)$	$\|S_0\|_0$	$\text{rank}(\hat{L})$	$\|\hat{S}\|_0$	$\frac{\|\hat{L}-L_0\|_F}{\|L_0\|_F}$	# SVD	Time(s)
500	25	12,500	25	12,500	1.1×10^{-6}	16	2.9
1,000	50	50,000	50	50,000	1.2×10^{-6}	16	12.4
2,000	100	200,000	100	200,000	1.2×10^{-6}	16	61.8
3,000	250	450,000	250	450,000	2.3×10^{-6}	15	185.2

rank$(L_0) = 0.05 \times n$, $\|S_0\|_0 = 0.05 \times n^2$.

n	$\text{rank}(L_0)$	$\|S_0\|_0$	$\text{rank}(\hat{L})$	$\|\hat{S}\|_0$	$\frac{\|\hat{L}-L_0\|_F}{\|L_0\|_F}$	# SVD	Time(s)
500	25	25,000	25	25,000	1.2×10^{-6}	17	4.0
1,000	50	100,000	50	100,000	2.4×10^{-6}	16	13.7
2,000	100	400,000	100	400,000	2.4×10^{-6}	16	64.5
3,000	150	900,000	150	900,000	2.5×10^{-6}	16	191.0

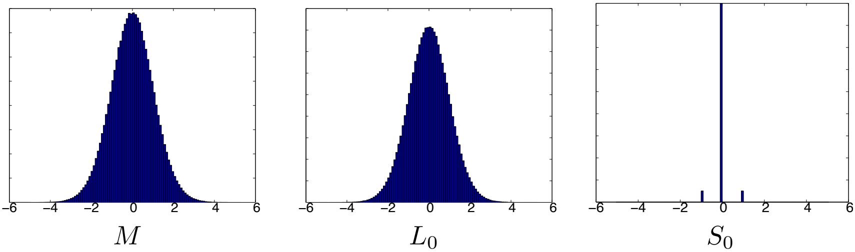
rank$(L_0) = 0.05 \times n$, $\|S_0\|_0 = 0.10 \times n^2$.

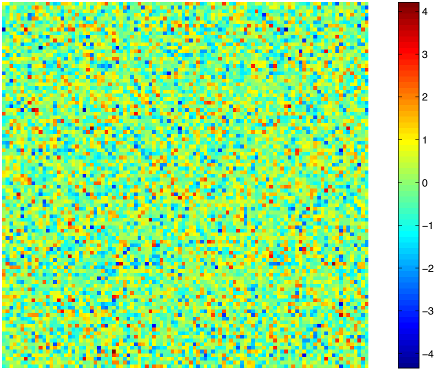
Computational cost higher than classical PCA but not by a large factor!

Empirical performance: Chiara's example

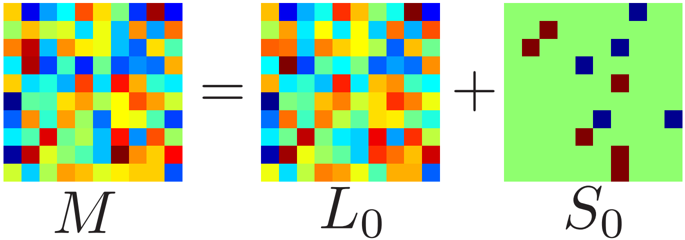
Rank-r matrix $L_0 = \frac{1}{\sqrt{r}} X_{n \times r} Y_{r \times n}$*:* X, Y independent $\mathcal{N}(0,1)$ entries

Sparse component S_0*:* random support + indep. symmetric ± 1 Bernoullis

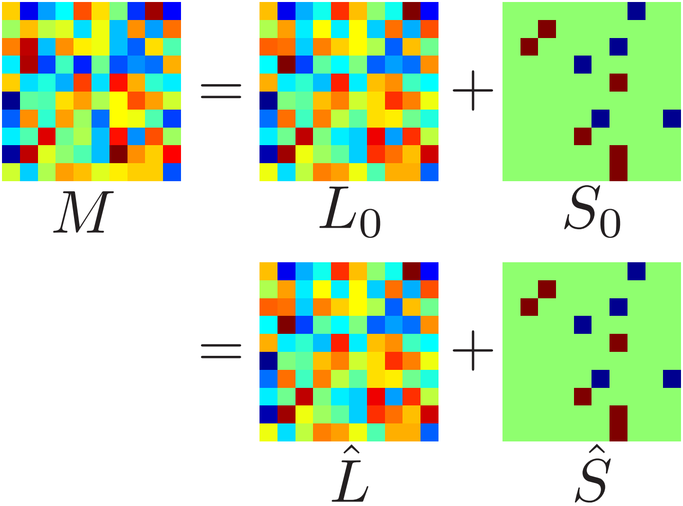


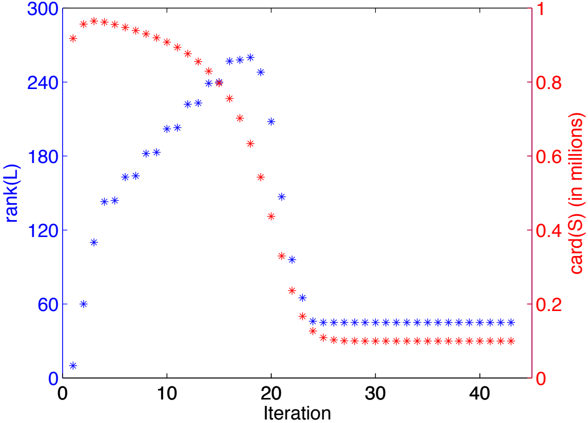


M



M
=
L_0
+
S_0


M
$=$
L_0
$+$
S_0
$=$
$\hat{L}$
$+$
$\hat{S}$



300
240
180
120
60
0
rank(L)
1
0.8
0.6
0.4
0.2
0
card(S) (in millions)
0
10
20
30
40
Iteration

Some applications

- Many applications
- Today, applications in computer vision

Application to video surveillance

Sequence of 200 video frames (144×172 pixels) with a static background

Problem: detect any activity in the foreground

Background modeling from surveillance video, I

(a) Original (b) Low-rank $\hat{L}$ (c) Sparse $\hat{S}$ (d) Low-rank $\hat{L}$ (e) Sparse $\hat{S}$

PCP — Alternating minimization

Alternating minimization of an M-estimator (De La Torre and Black, '03)

Background modeling from surveillance video, II

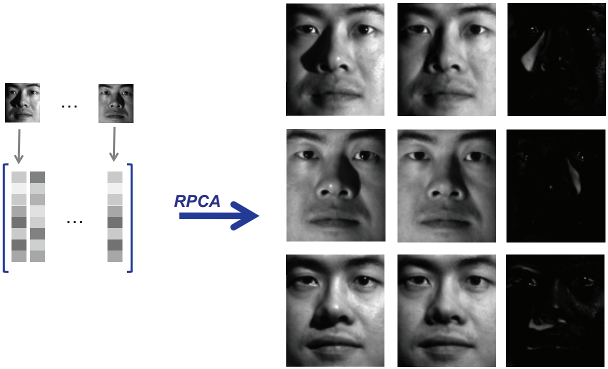
(a) Original (b) Low-rank $\hat{L}$ (c) Sparse $\hat{S}$ (d) Low-rank $\hat{L}$ (e) Sparse $\hat{S}$

PCP Alternating minimization

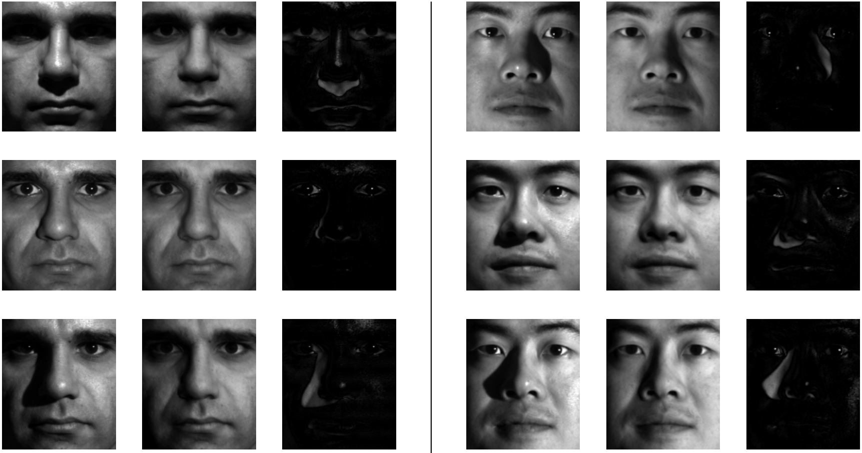
Three frames from a 250 frame sequence taken in a lobby, with varying illumination (Li et al., '04).

Removing shadows and specularities from face images

Sequence of 58 images (192×168) under different illumination conditions



Removing shadows and specularities from face images



(a) M (b) $\hat{L}$ (c) $\hat{S}$ | (a) M (b) $\hat{L}$ (c) $\hat{S}$

Corrections of specularities in the eyes, shadows, brightness saturation, ...

APPLICATIONS – *Repairing vintage movies*

Original D

Corruptions

Repaired A

Frame 1

480×620 pixels

APPLICATIONS – *Repairing vintage movies*

Original D

Corruptions

Repaired A

Frame 2

APPLICATIONS – *Repairing vintage movies*

Original D

Corruptions

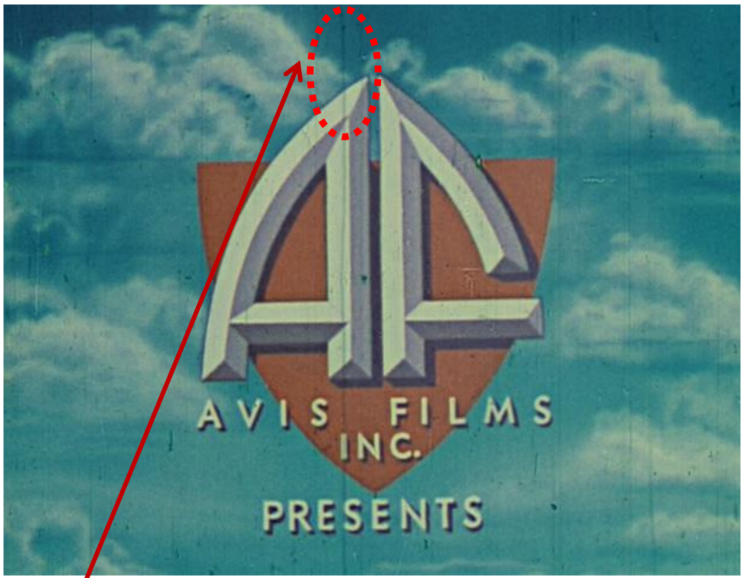
Repaired A

Frame 3

APPLICATIONS – *Repairing vintage movies*

Original D

Repaired A

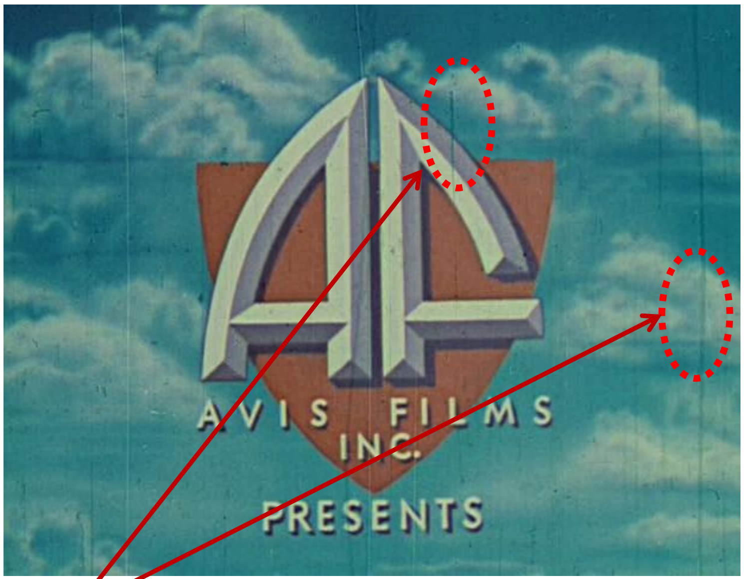


Corruptions

Frame 4

APPLICATIONS – *Repairing vintage movies*

Original D



Corruptions

Repaired A

Frame 5

APPLICATIONS – *Repairing vintage movies*

Original D

Repaired A

Corruptions

Frame 6

Original D

Repaired A

Corruptions

Frame 7

Robust batch image alignment (Ma et al.)

- *Input*: $\boldsymbol{M}$ corrupted and misaligned batch of images (data)
- *Output*: $\boldsymbol{L}$ aligned low-rank images; $\boldsymbol{S}$ sparse errors

$$\text{(Model)} \qquad \boldsymbol{M} \circ \tau = \boldsymbol{L_0} + \boldsymbol{S_0}$$

τ: parametric deformation (rigid, affine, projective)

Robust batch image alignment (Ma et al.)

- *Input*: $\boldsymbol{M}$ corrupted and misaligned batch of images (data)
- *Output*: $\boldsymbol{L}$ aligned low-rank images; $\boldsymbol{S}$ sparse errors

$$\text{(Model)} \qquad \boldsymbol{M} \circ \tau = \boldsymbol{L_0} + \boldsymbol{S_0}$$

τ: parametric deformation (rigid, affine, projective)

Bootstrap: find $\boldsymbol{L}$ and $\boldsymbol{S}$ and τ solution to

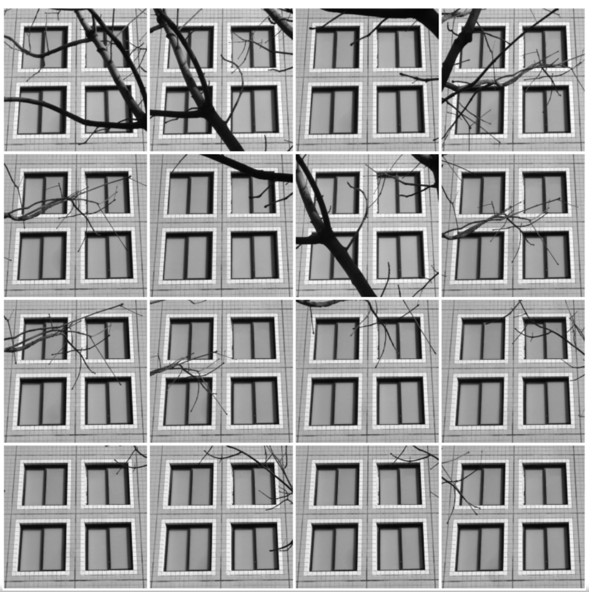
$$\min \|\boldsymbol{L}\|_1 + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{L}\|_1 + \lambda \|\boldsymbol{S}\|_1 = \boldsymbol{M} \circ \tau$$

APPLICATIONS – *2D image matching and 3D modeling*

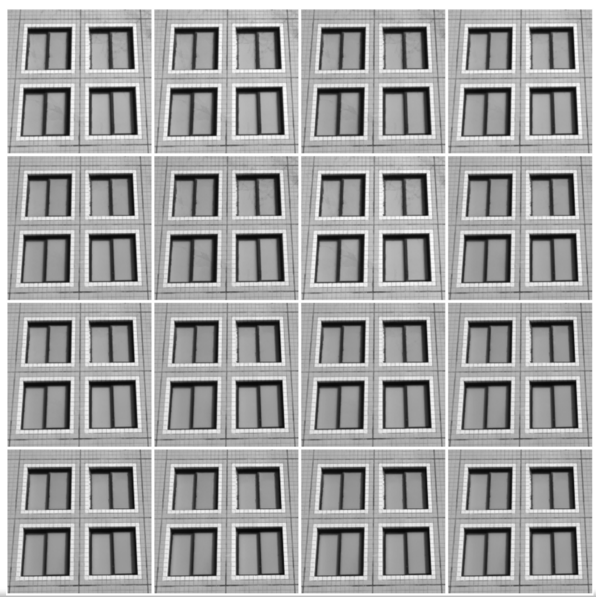
D

$\tau \in$ 2D homographies

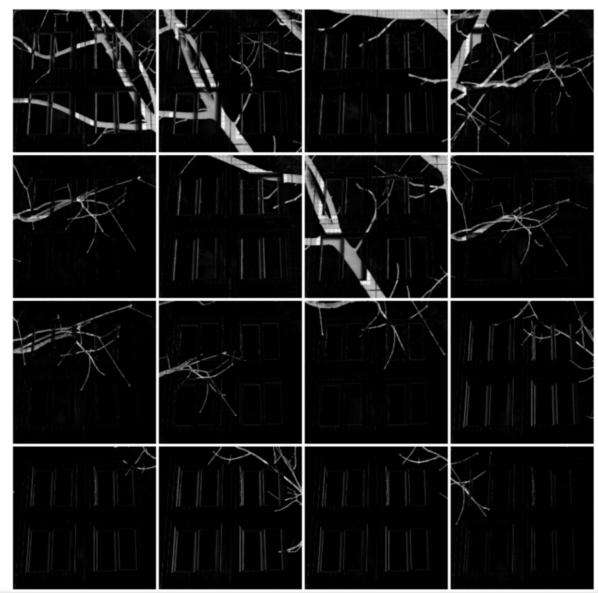
$D \circ \tau$



A



E



Peng, Ganesh, Wright, Ma, to appear CVPR'10

APPLICATIONS – *Video stabilization and enhancement*

Shaky video (D)
vs.
Aligned video ($D \circ \tau$)

Clean video (A)

Error video (E)

APPLICATIONS – *Aligning handwritten digits*

D

Learned-Miller PAMI'06

Vedaldi CVPR'08

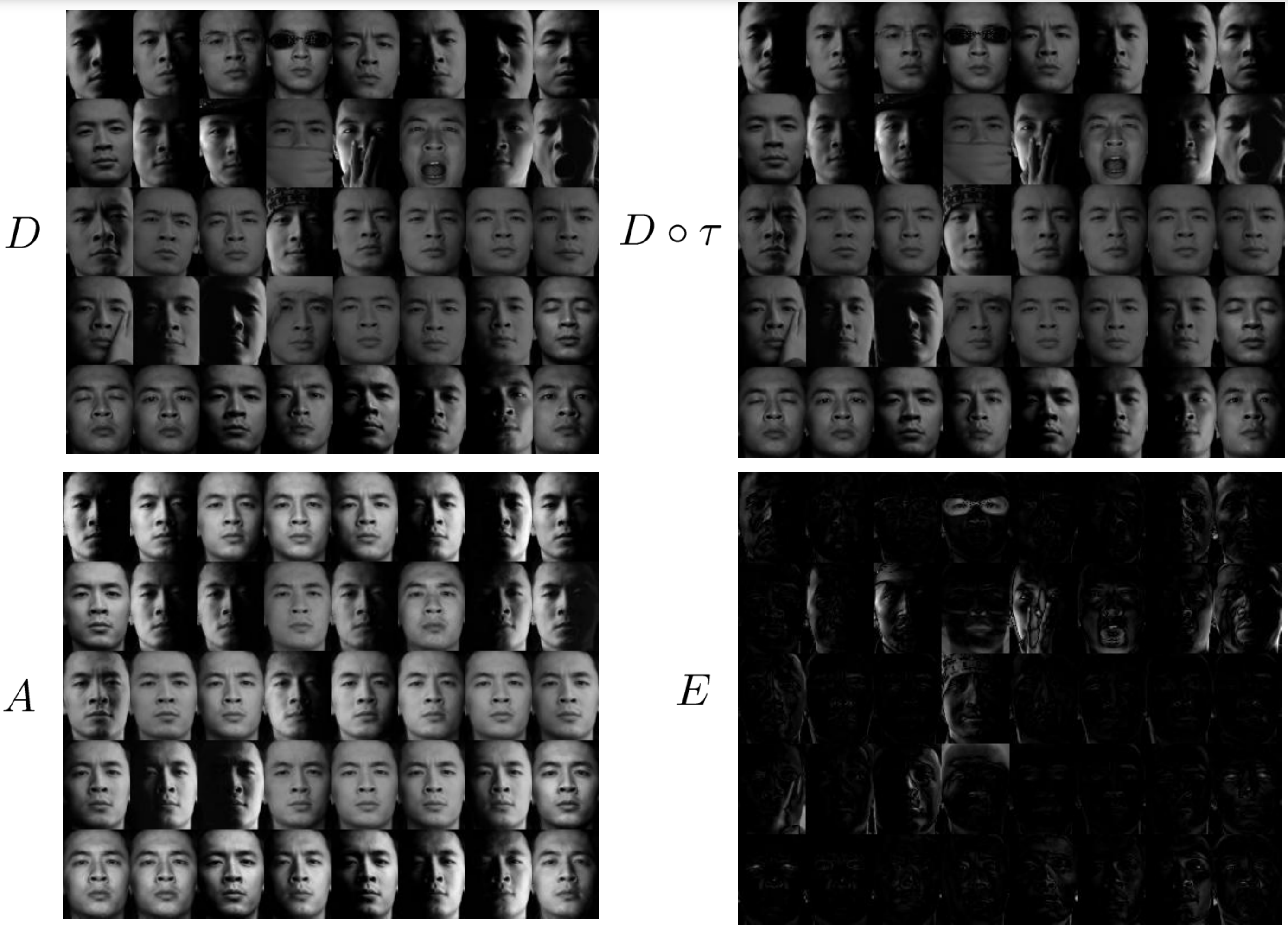
$D \circ \tau$

A

E

Peng, Ganesh, Wright, Ma, submitted to CVPR'10.

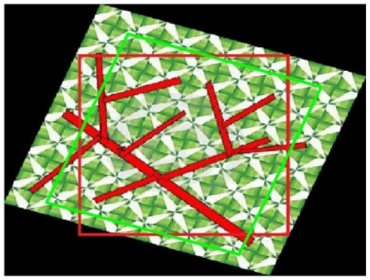
APPLICATIONS – *Simultaneous Alignment and Repair*



Peng, Ganesh, Wright, Ma, submitted to CVPR'10.

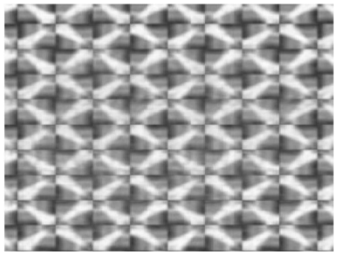
Transform Invariant Low-rank Textures (TILT)

D– corrupted & deformed observation



$\circ\, \tau \;=$

A– rectified low-rank textures



$+$

E– sparse errors

Problem: Given $D \circ \tau = A_0 + E_0$, recover τ , A_0 and E_0.

Parametric deformations (affine, projective…)

Low-rank component

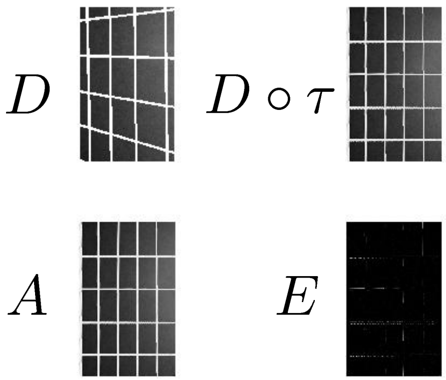
Sparse component

Solution: iteratively estimate the deformation and low-rank texture:

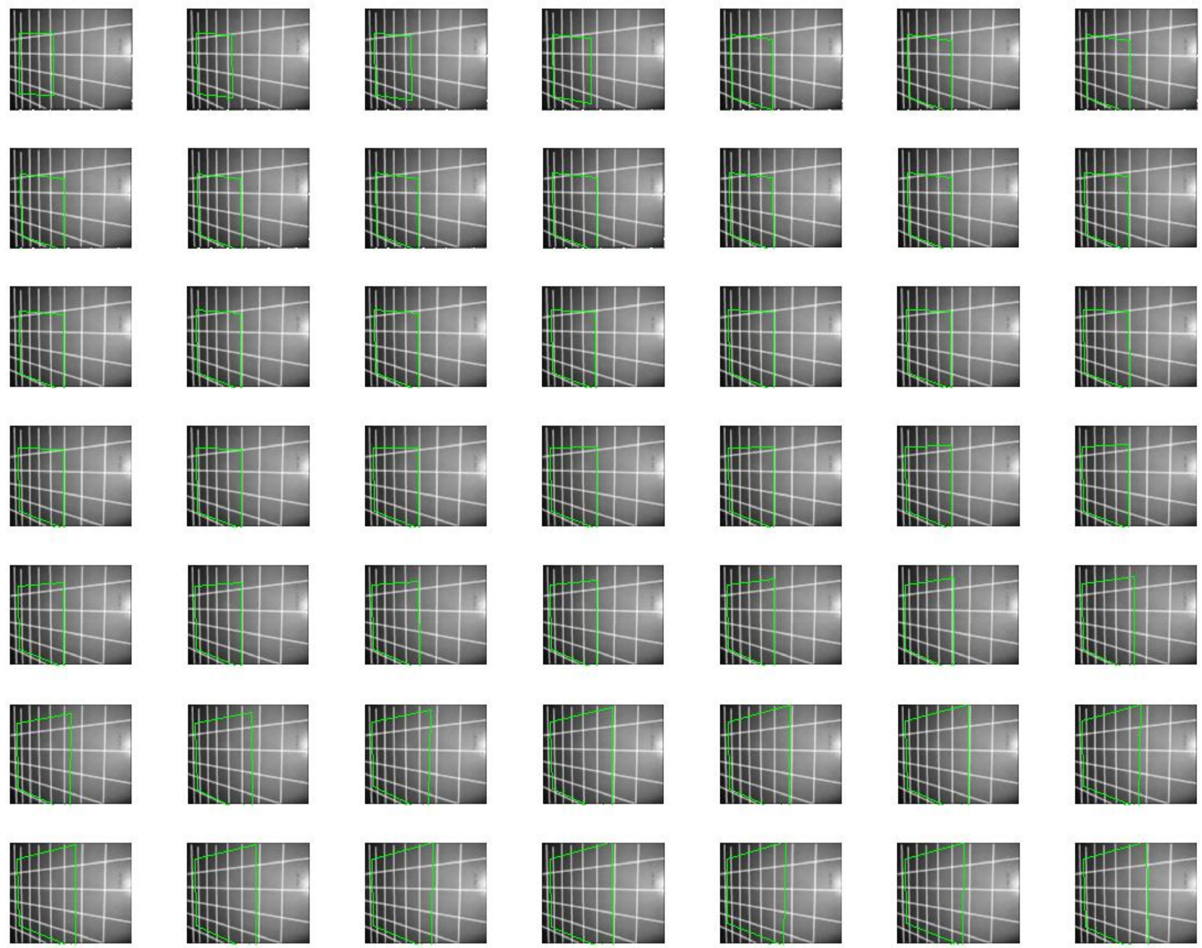
Iterate:

$$\min \; \|A\|_* + \lambda \|E\|_1 \quad \text{subj} \quad A + E = D \circ \tau_k + J \Delta\tau$$

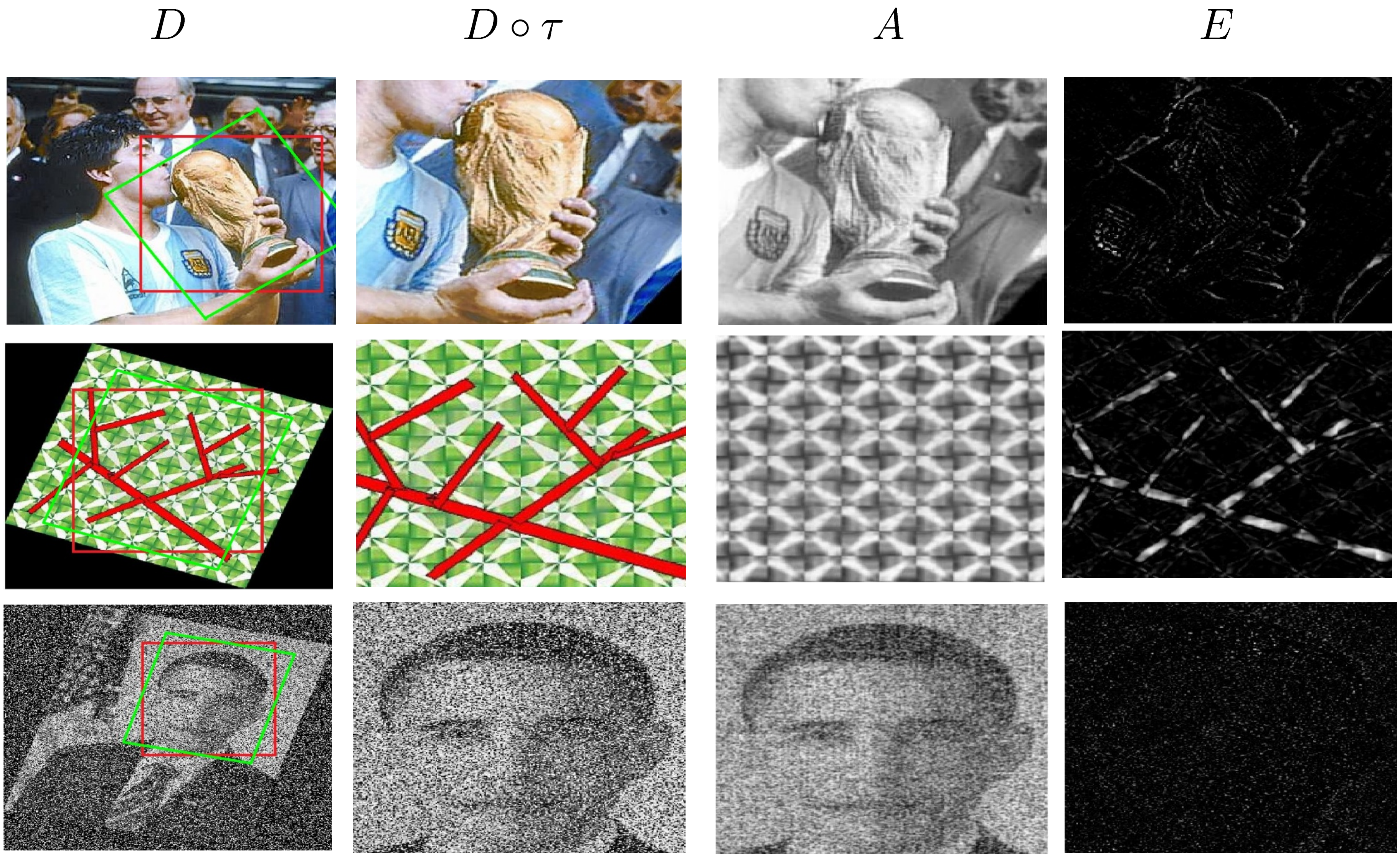
TILT via Iterative RPCA-Like Convex Optimization



Iteration Processes

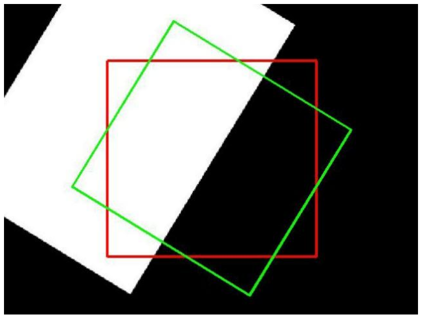


TILT – Robust to Background, Occlusion, and Corruption

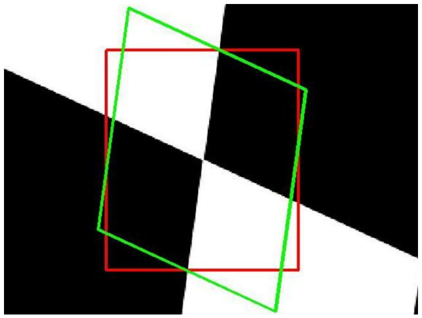


TILT: All Types of Regular Geometric Structures in Images

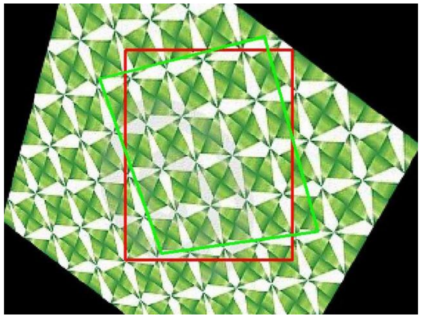
an edge



a corner

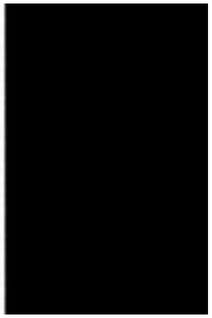


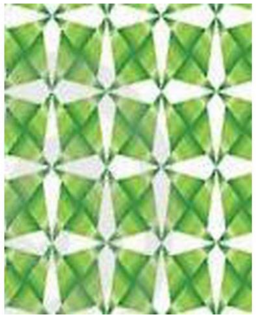
symmetry



regularity

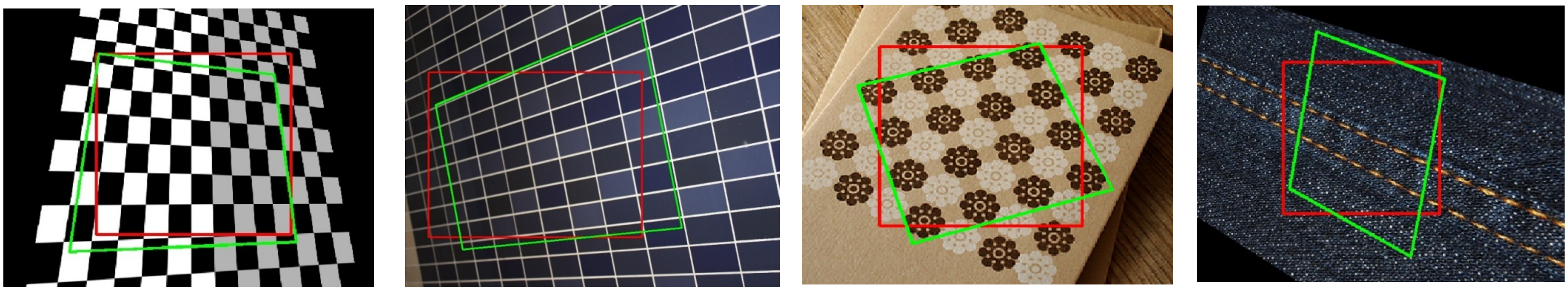
Un-Tilted Low-rank Textures



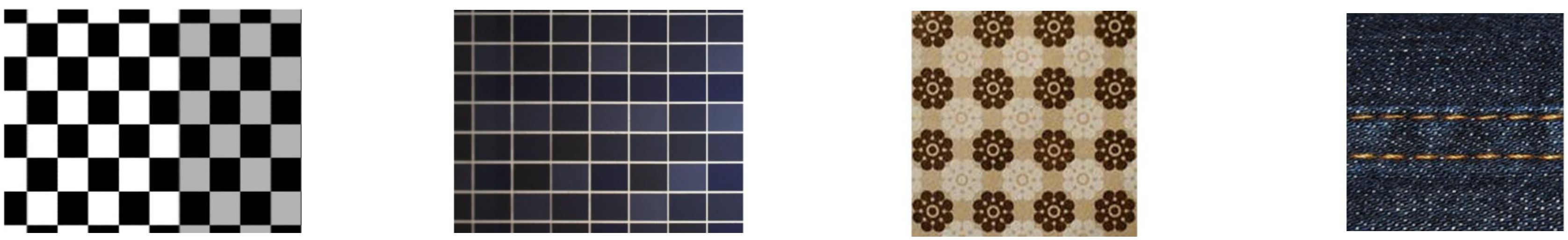


TILT: Examples of Symmetric Patterns and Textures

Input (red window)

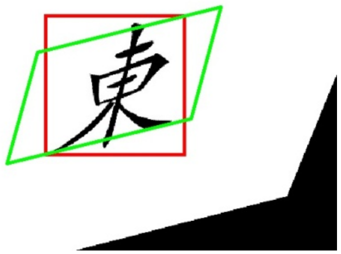


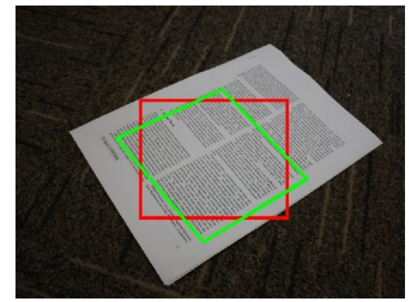
Output (rectified green window)



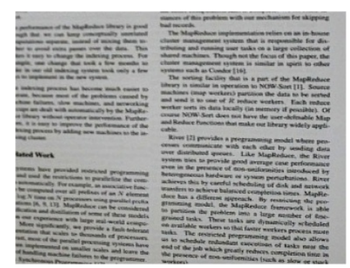
TILT: Examples of Characters, Signs, and Texts

Input (red window)



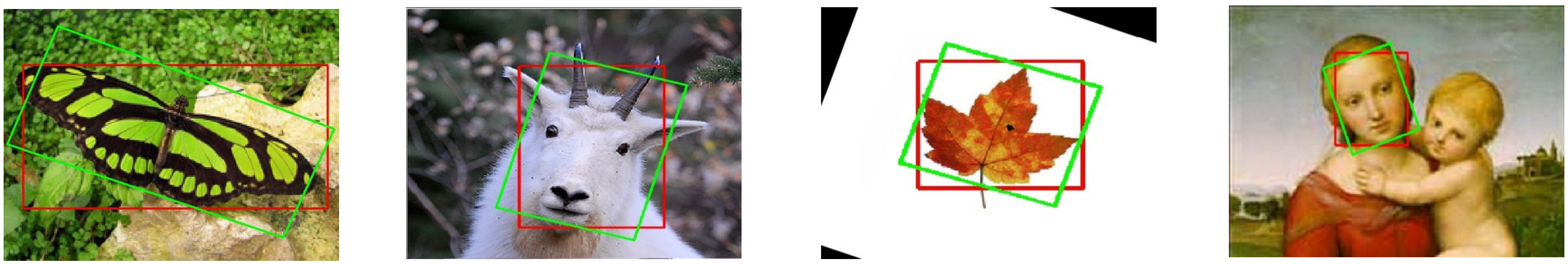


Output (rectified green window)



TILT: Examples of Natural Objects with Bilateral Symmetry

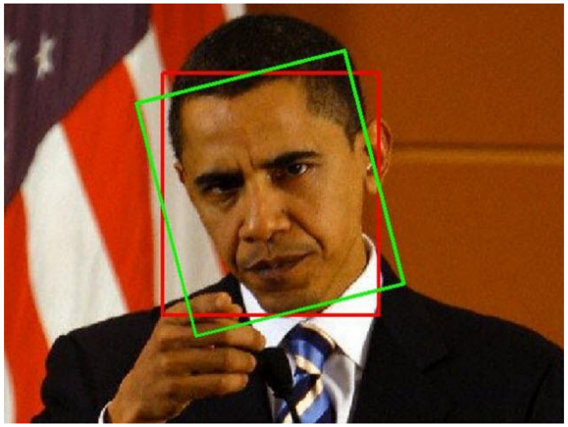
Input (red window)

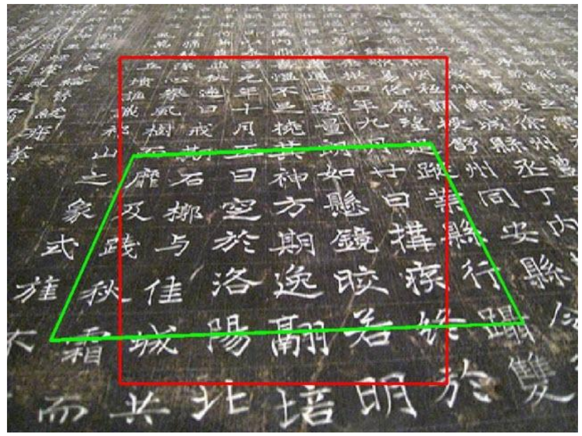


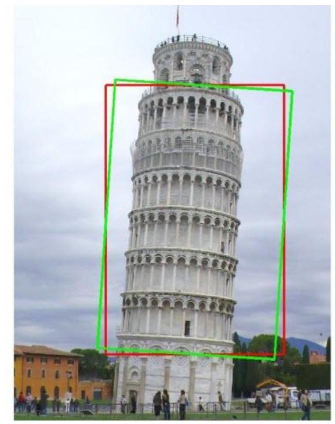
Output (rectified green window)

TILT: More Examples

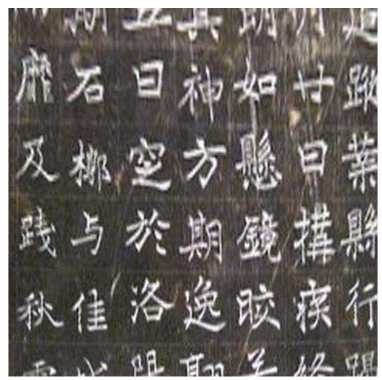
Input (red window)





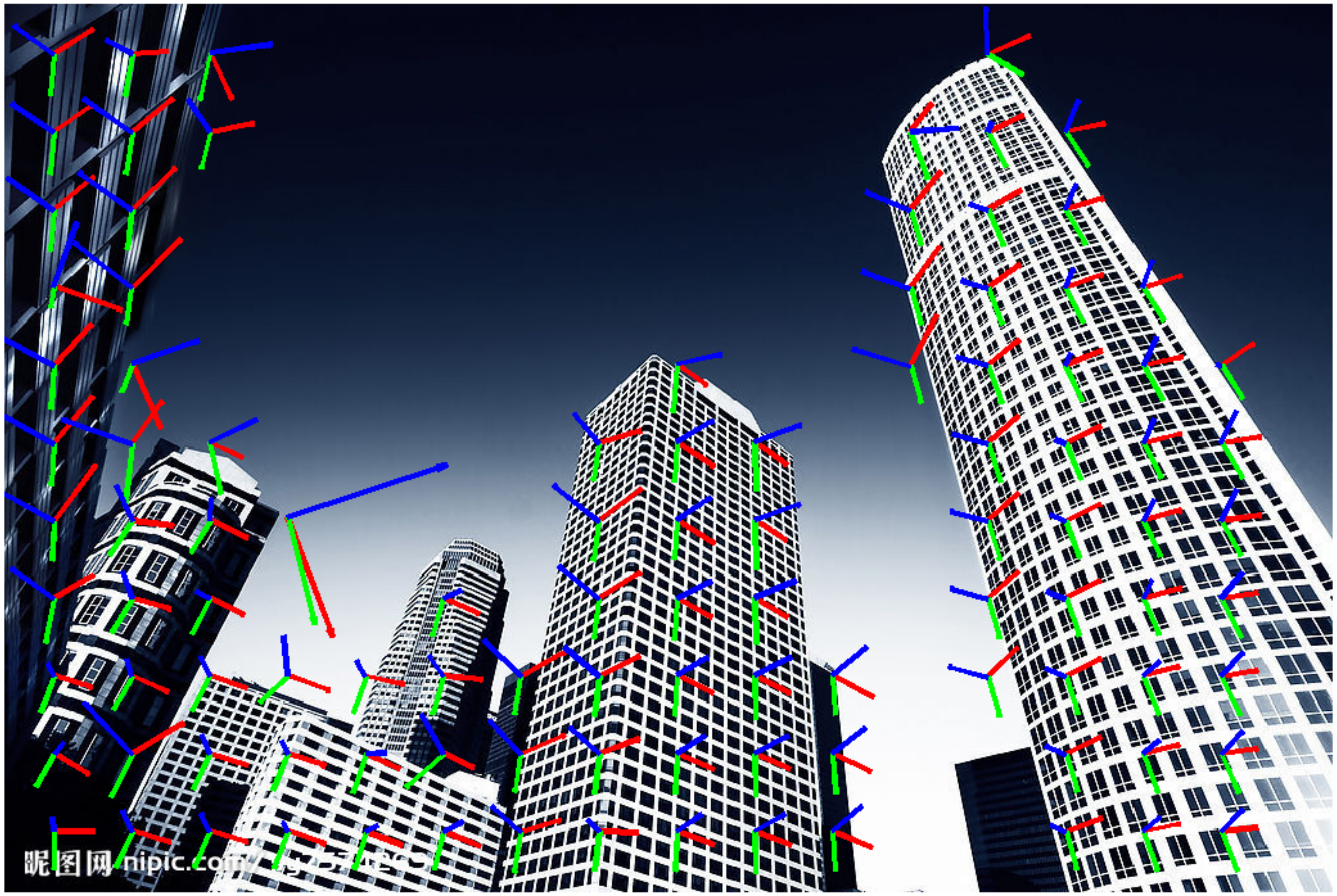


Output (rectified green window)

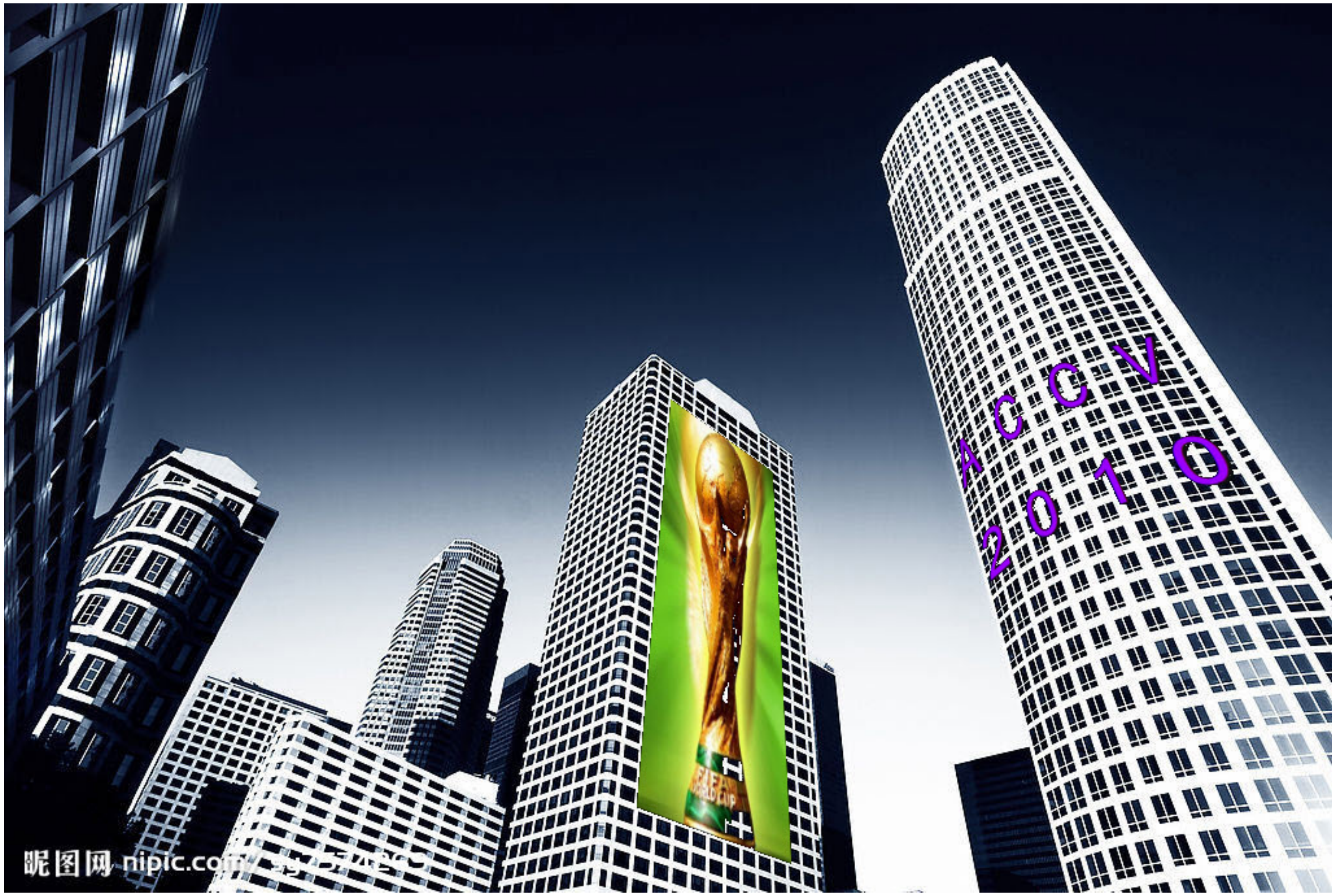


TILT – Local 3D Geometry from Low-rank Textures

Run TILT on a grid of 60x60 windows



TILT – Geometric Image Editing



Extensions

Robustness to noise (same people + Zhou)

- In reality: data matrix = low-rank + sparse + noise

$$M = L_0 + S_0 + Z_0, \quad \|Z_0\|_F \le \delta$$

- Recovery via relaxed PCP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & \|M - (L + S)\|_F \le \delta \end{array}$$

- Reconstruction is stable

$$\frac{1}{n^2}\Big(\|\hat{L} - L_0\|_F^2 + \|\hat{S} - S_0\|_F^2\Big) \le O(\delta^2)$$

Extensions

Robustness to noise (same people + Zhou)

- In reality: data matrix = low-rank + sparse + noise

$$M = L_0 + S_0 + Z_0, \quad \|Z_0\|_F \le \delta$$

- Recovery via relaxed PCP

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda \|S\|_1 \\ \text{subject to} & \|M - (L+S)\|_F \le \delta \end{array}$$

- Reconstruction is stable

$$\frac{1}{n^2}\Big(\|\hat{L} - L_0\|_F^2 + \|\hat{S} - S_0\|_F^2\big) \le O(\delta^2)$$

Dense correction (same people + Ganesh)

- Sparse component S_0 has random signs
- Fraction of nonzero entries in $S_0 \to 1$
- PCP still succeeds with high probability!

Summary

- Principled approach to Robust PCA
- Works well in theory and in practice
- Amenable to large scale problems – early effective algorithms
- Many applications
    - Computer vision
    - Signal processing
    - Data analysis
    - Many more (to come)
- *Interested in what you think!*

E. J. Candès, X. Li, Y. Ma, and J. Wright (2009). *Robust Principal Component Analysis?* Stanford Technical Report

Happy Anniversary!

Long Live IPAM!

Proof via dual certification

Find dual variable Y such that pair $(L_0, S_0; Y)$ obeys KKT optimality conditions

$$M = L_0 + S_0$$

Proof via dual certification

Find dual variable Y such that pair $(L_0, S_0; Y)$ obeys KKT optimality conditions

$$M = L_0 + S_0$$

- T: span of all matrices with row space or col. space included in that of L_0
- Ω: span of all matrices with support included in that of S_0

Proof via dual certification

Find dual variable Y such that pair $(L_0, S_0; Y)$ obeys KKT optimality conditions

$$M = L_0 + S_0$$

- T: span of all matrices with row space <u>or</u> col. space included in that of L_0
- Ω: span of all matrices with support included in that of S_0

Sufficient (and almost necessary) conditions

- $T \cap \Omega = \{0\}$
- There is $W \in T^\perp$ such that

$$\|W\| < 1$$

and $Y = UV^* + W$ obeys

$$\begin{cases} Y_{ij} = \lambda[\mathrm{sgn}(S_0)]_{ij} & (i,j) \in \Omega \\ |Y_{ij}| < \lambda & \text{otherwise} \end{cases}$$

Augmented Lagrangian approach

$$\begin{array}{ll} \text{minimize} & \|L\|_* + \lambda\|S\|_1 + \frac{1}{2\tau}\|M - L - S\|_F^2 \\ \text{subject to} & L + S = M \end{array}$$

Lagrangian

$$\mathcal{L}(L, S; Y) = \|L\|_* + \lambda\|S\|_1 + \frac{1}{\tau}\langle Y, M - L - S\rangle + \frac{1}{2\tau}\|M - L - S\|_F^2$$

Basic algorithm (Usawa): dual gradient ascent

$$\begin{cases} (L_k, S_k) & = \arg\min_{L,S} \mathcal{L}(L, S; Y_{k-1}) \\ Y_k & = Y_{k-1} + \delta_k(M - L_k - S_k) \end{cases}$$

Sequential minimization

Scalar shrinkage: $\mathcal{S}_\tau[x] = \mathsf{sgn}(x)\max(|x| - \tau, 0)$

- Componentwise thresholding $\mathcal{S}_\tau(X)$
- Singular value thresholding $\mathcal{D}_\tau(X)$

$$\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^* \qquad X = U\Sigma V^*$$

Sequential minimization

Scalar shrinkage: $\mathcal{S}_\tau[x] = \mathsf{sgn}(x)\max(|x| - \tau, 0)$

- Componentwise thresholding $\mathcal{S}_\tau(X)$
- Singular value thresholding $\mathcal{D}_\tau(X)$

$$\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^* \qquad X = U\Sigma V^*$$

$$\mathcal{L}(L, S; Y) = \|L\|_* + \lambda\|S\|_1 + \frac{1}{\tau}\langle Y, M - L - S\rangle + \frac{1}{2\tau}\|M - L - S\|_F^2$$

Easy to minimize over L and S separately

$$\arg\min_L \mathcal{L}(L, S, Y) = \mathcal{D}_\tau(M - S + Y)$$
$$\arg\min_S \mathcal{L}(L, S, Y) = \mathcal{S}_{\lambda\tau}(M - L + Y)$$

PCP by alternating directions

initialize: S_0, Y_0 and $\tau > 0$
while not converged

1. $L_k = \mathcal{D}_\tau(M - S_{k-1} + Y_{k-1})$ (shrink singular values)
2. $S_k = \mathcal{S}_{\lambda\tau}(M - L_k + Y_{k-1})$ (shrink scalar entries)
3. $Y_k = Y_{k-1} + (M - L_k - S_k)$

end while
output: L, S

PCP by alternating directions

initialize: S_0, Y_0 and $\tau > 0$
while not converged

1. $L_k = \mathcal{D}_\tau(M - S_{k-1} + Y_{k-1})$ (shrink singular values)
2. $S_k = \mathcal{S}_{\lambda\tau}(M - L_k + Y_{k-1})$ (shrink scalar entries)
3. $Y_k = Y_{k-1} + (M - L_k - S_k)$

end while
output: L, S

All the computational work is in (1)

When iterates L_k have low rank

- Only need to compute few singular values (and vectors) at each step
- Lanczos iterations are very effective