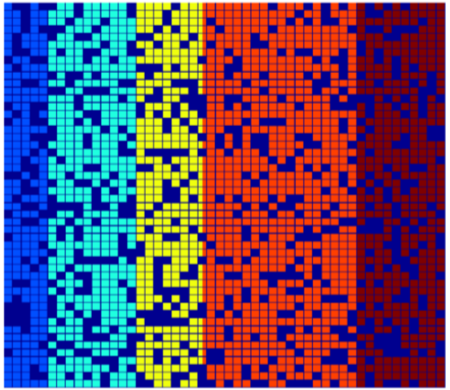
Dimension Reduction Techniques



Dr. Jean-Charles Bazin

Imaging and Video Group, Disney Research Zürich

ETH lecturer

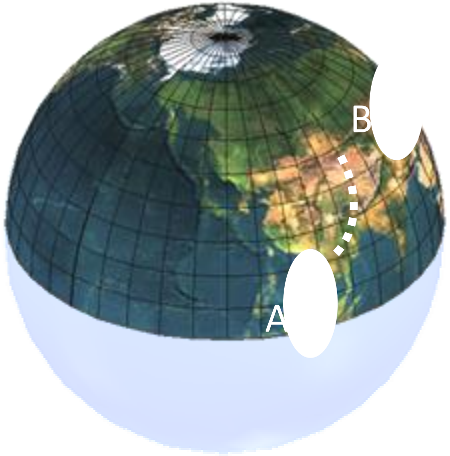
ETH zürich Disney Research

Contents

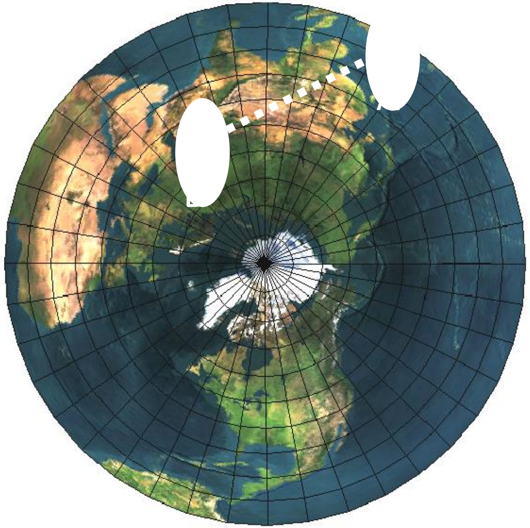
- PCA
- Robust PCA and rank minimization
- Multi-dimensional scaling

Motivation

- How to display high dimensional data in 2D?
  - Find “appropriate” 2D projection



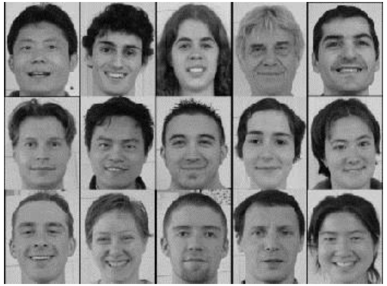
Earth (sphere)



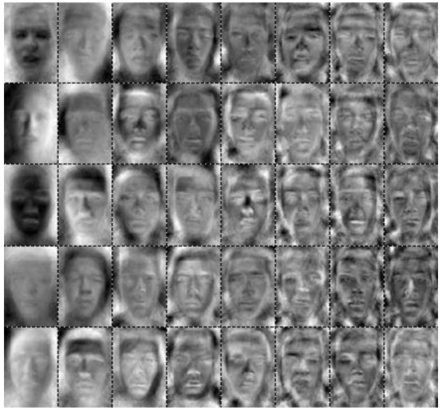
Planar map

Motivation

- Data summarization
  - Reduce required amount of memory, etc…
    - E.g. combination of “basis”
  - Possible to extract useful information



dataset

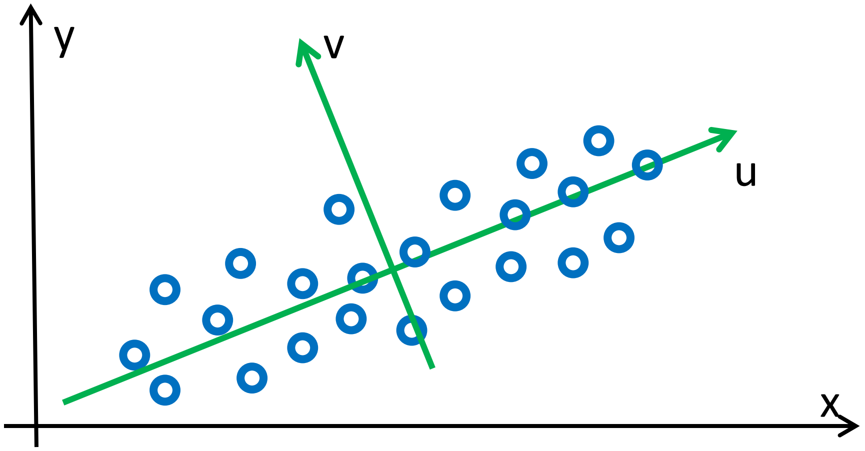


Top eigenfaces

Reconstruction using eigenfaces basis

PCA

- Convert data into linearly uncorrelated variables
  - Uncorrelated variables:
    - covariance=0 (orthogonal)
    - Example of such variables: (O_x, O_y)
  - Why "uncorrelated"? Reduce the redundancy
  - Why "linear"? Simpler to compute



PCA

- **Principal component analysis (PCA)**
- Find a linear mapping/projection that preserves (most of) the information or structure
- Main idea:
  - the first principal component has the largest possible variance
    - Why? accounts for as much of the variability in the data as possible
  - the second principal component:
    - The highest variance possible
    - Must be orthogonal to (i.e., uncorrelated with) the first PC
  - Etc….

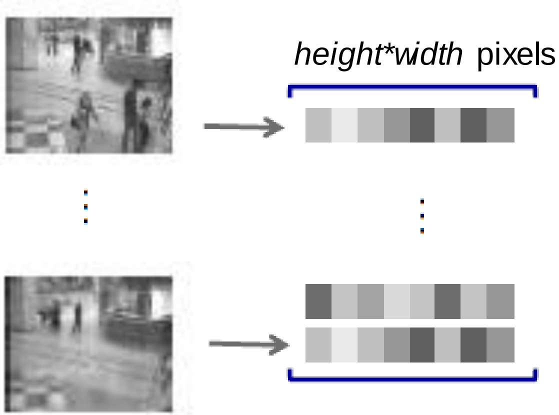
PCA algorithm #1/2

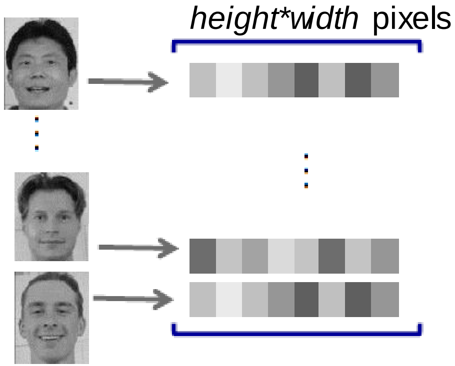
- Data

m variables/sensors/pixels/features

$$X = \begin{bmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{nj} & \dots & x_{nm} \end{bmatrix}$$

n observations/experiments/images





PCA algorithm #1/2

- Data

$$X = \begin{bmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{nj} & \dots & x_{nm} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_j & \dots & \mathbf{X}_m \end{bmatrix}$$

n observations x *m* variables

Let's assume the data has been "centered" (the mean of each column is 0, i.e. the mean of each column has been shifted to zero)

- Linear combination of the *m* variables

$$\mathbf{y} = \mathbf{w}X = w_1\mathbf{X}_1 + w_2\mathbf{X}_2 + \ldots + w_m\mathbf{X}_m$$

$$var(\mathbf{y}) = (\mathbf{w}X)^T(\mathbf{w}X) = \mathbf{w}^T X^T X \mathbf{w} = \mathbf{w}^T \Sigma \mathbf{w}$$

$$\arg \max_{\mathbf{w}, \|\mathbf{w}\|=1} var(\mathbf{y}) = \arg \max_{\mathbf{w}, \|\mathbf{w}\|=1} \mathbf{w}^T \Sigma \mathbf{w} \Rightarrow$$

Eigenvector associated to the highest eigenvalue of Σ (SVD)

See previous lecture

PCA algorithm #2/2

- Second principal component

$$\mathbf{y}_2 = \mathbf{w}_2 X = w_{21}\mathbf{x}_1 + w_{22}\mathbf{x}_2 + \ldots + w_{2m}\mathbf{x}_m$$

$$var(\mathbf{y}_2) = \mathbf{w}_2^T \Sigma \mathbf{w}_2$$

$$\arg\max_{\mathbf{w}_2, \|\mathbf{w}_2\|=1} var(\mathbf{y}_2) \text{ s.t. } \mathbf{w}_1 \cdot \mathbf{w}_2 = 0 \Rightarrow \arg\max_{\mathbf{w}_2, \|\mathbf{w}_2\|=1} \mathbf{w}_2^T \Sigma \mathbf{w}_2 \text{ s.t. } \mathbf{w}_1 \cdot \mathbf{w}_2 = 0$$

$\Rightarrow$ Eigenvector associated to the second highest eigenvalue of Σ (SVD)

Why? Remember that eigenvectors are orthogonal

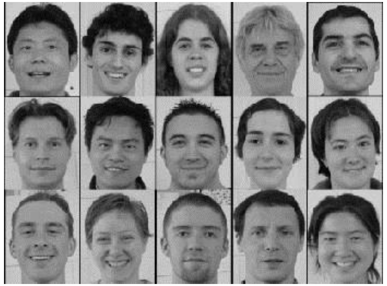
- i-th principal component

$$\mathbf{y}_i = \mathbf{w}_i X = w_{i1}\mathbf{x}_1 + w_{i2}\mathbf{x}_2 + \ldots + w_{im}\mathbf{x}_m$$

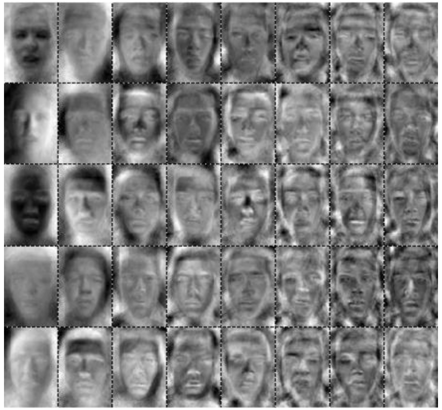
$$\arg\max_{\mathbf{w}_i, \|\mathbf{w}_i\|=1} \mathbf{w}_i^T \Sigma \mathbf{w}_i \text{ s.t. } \mathbf{w}_j \cdot \mathbf{w}_i = 0 \forall j < i$$

$\Rightarrow$ Eigenvector associated to the i-th highest eigenvalue of Σ (SVD)

Application - eigenfaces



dataset



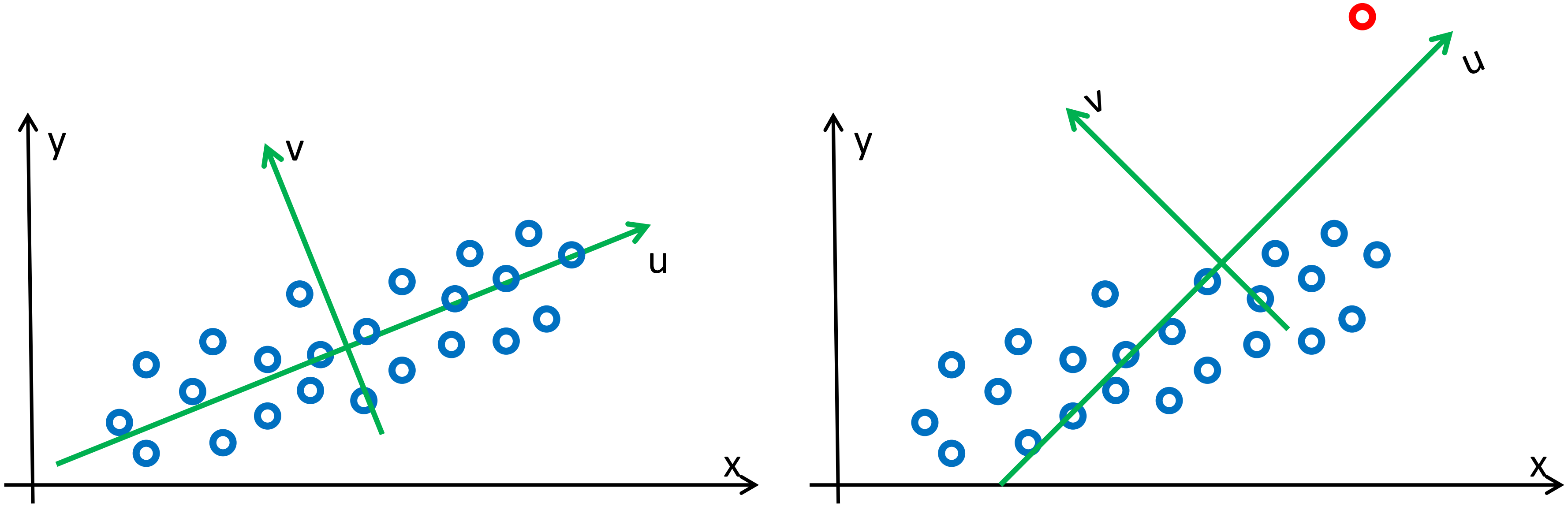
Top eigenfaces

Reconstruction using eigenfaces basis

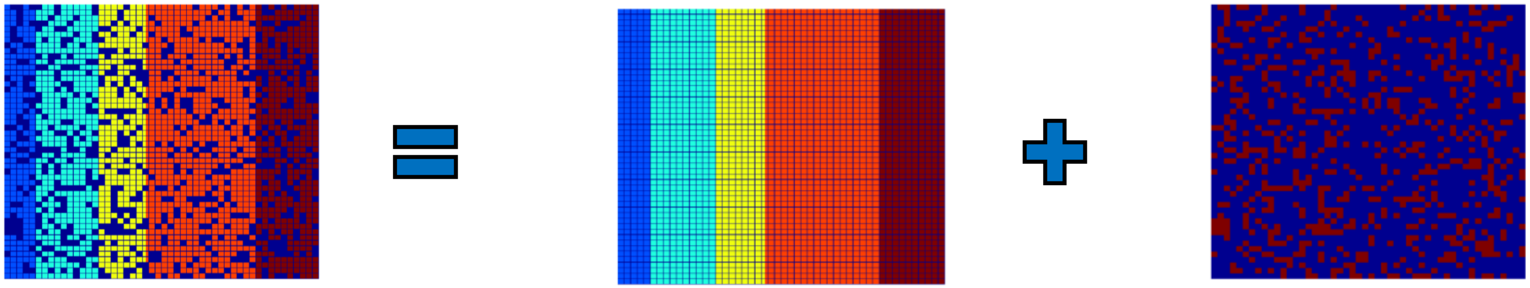
- "Eigenfaces": eigenvectors for face dataset
- Any face is a linear combination of the eigenfaces
  - E.g. average face + 15% of eigenface 1, -9% of eigenface 2, and +37% of eigenface 3
- A limited number of eigenfaces is usually sufficient to approximate/represent faces

Robust PCA

- Missing information, outliers, etc…

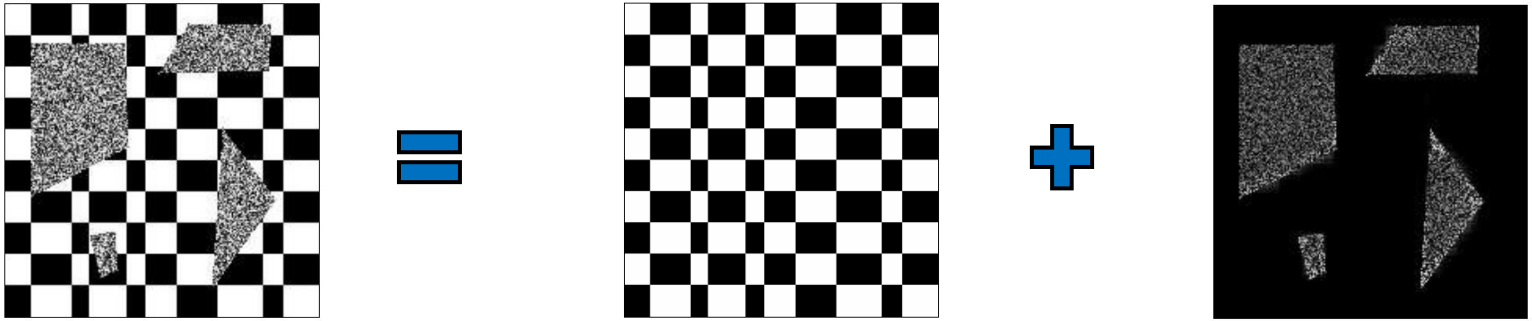


Robust PCA



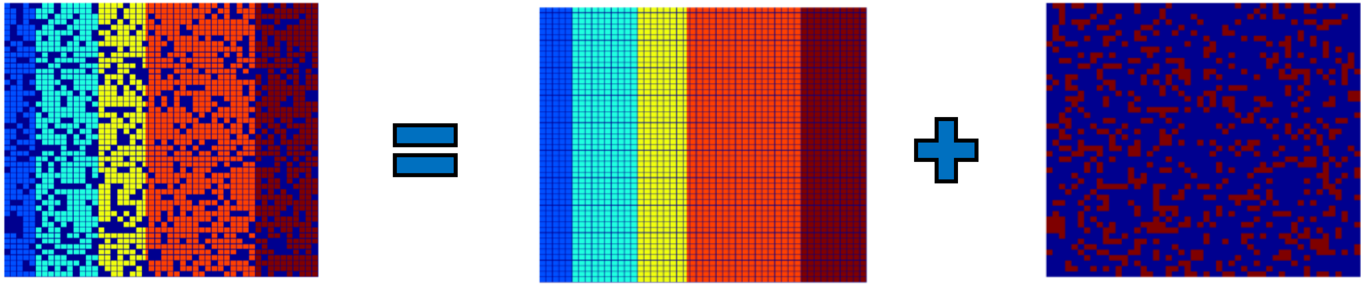
Matrix of corrupted observations | *Underlying low-rank matrix* | *Sparse error matrix*

- Remember the definition of rank?
- Why sparse?



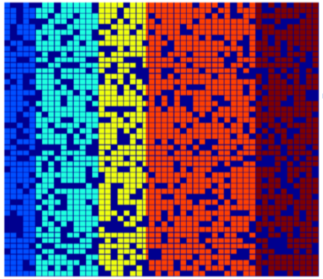
From http://perception.csl.illinois.edu/matrix-rank/

Robust PCA



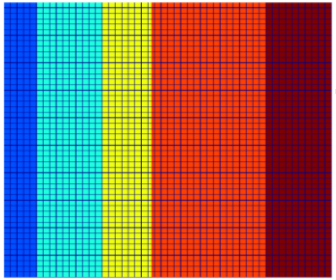
D (observation) **A (low-rank)** **E (sparse)**

- Problem: Given $D = A + E$, find A and E
- Target:
  - $rank(A)$ must be small
  - E must contain many 0s, i.e. $\|E\|_0 = \#\{E_{ij} \neq 0\}$ must be small



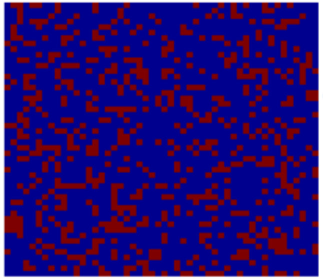
D (observation)

$$= $$



A (low-rank)

$$+$$



E (sparse)

$$O = \begin{bmatrix} 1 & 1 & 3 & 6 & 6 & 6 \\ 1 & 1 & 3 & 6 & 6 & 6 \\ 1 & 1 & 3 & 6 & 6 & 6 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 3 & 6 & 6 & 6 \\ 1 & 1 & 3 & 6 & 6 & 6 \\ 1 & 1 & 3 & 6 & 6 & 6 \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rank(A)=? $\|E\|_0 = ?$

$$A = \begin{bmatrix} -1 & -1 & 1 & 4 & 4 & 4 \\ -1 & -1 & 1 & 4 & 4 & 4 \\ -1 & -1 & 1 & 4 & 4 & 4 \end{bmatrix} \quad E = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

rank(A)=? $\|E\|_0 = ?$

$$O = \begin{bmatrix} 1 & 1 & 3 & 7 & 6 & 6 \\ 1 & 2 & 3 & 6 & 8 & 7 \\ 1 & 1 & 3 & 6 & 6 & 6 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 3 & 6 & 6 & 6 \\ 1 & 1 & 3 & 6 & 6 & 6 \\ 1 & 1 & 3 & 6 & 6 & 6 \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

rank(A)=? $\|E\|_0 = ?$

Robust PCA - formulation

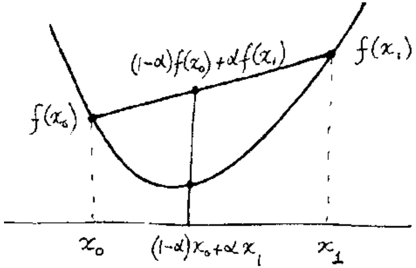
- Original mathematical formulation

$$\boxed{\arg\min_{A,E} rank(A) + \gamma \|E\|_0 \text{ s.t. } A + E = D}$$

$$rank(A) = \#\{\sigma_i(A) \neq 0\} \qquad \|E\|_0 = \#\{E_{ij} \neq 0\}$$

  - But… NP hard and not convex

Is rank convex? Show
Is L0 convex? Show



$$f((1-\alpha)x_0 + \alpha x_1) \leq (1-\alpha)f(x_0) + \alpha f(x_1)$$

J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In NIPS, 2009
E. Cand`es, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? Journal of the ACM, 2011.

Robust PCA - formulation

- Convex reformulation

$$\boxed{\arg\min_{A,E} rank(A) + \gamma \|E\|_0 \text{ s.t. } A + E = D}$$

$$rank(A) = \#\{\sigma_i(A) \neq 0\}$$

$$\|E\|_0 = \#\{E_{ij} \neq 0\}$$

$$\boxed{\arg\min_{A,E} \|A\|_* + \gamma \|E\|_1 \text{ s.t. } A + E = D}$$

Same solution
under certain cases!

"nuclear norm" $\|A\|_* = \sum_i \sigma_i(A)$

$$\|E\|_1 = \sum_{ij} |E_{ij}|$$

- Norms are convex, sum of convex functions is convex

➔ So the problem is convex

J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In NIPS, 2009
E. Cand`es, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? Journal of the ACM, 2011.

Robust PCA - solving

- Objective function

$$\boxed{\arg\min_{A,E} \|A\|_* + \gamma\|E\|_1 \text{ s.t. } A + E = D}$$

- Convex but how to solve it?
  - Methods for general convex function
  - Scalability issue

- Special structure
  - Soft-thresholding

Algorithms	Accuracy	Rank	||E||_0	# iterations	time (sec)
IT	5.99e-006	50	101,268	8,550	119,370.3
DUAL	8.65e-006	50	100,024	822	1,855.4
APG	5.85e-006	50	100,347	134	1,468.9
APG_P	5.91e-006	50	100,347	134	82.7
$EALM_P$	2.07e-007	50	100,014	34	37.5
$IALM_P$	3.83e-007	50	99,996	23	11.8

Background

$$\arg\min_{A,E} \|A\|_* + \gamma\|E\|_1 \text{ s.t. } A+E=D$$

- Constrained optimization system:

$$\arg\min_x f(x) \text{ s.t. } h(x)=c$$

- Lagrange function

Lagrange multiplier

$$L(x,\lambda) = f(x) + \lambda\,(h(x)-c)$$

- Generalization: $\arg\min_{x_1,\ldots,x_n} f(x_1,\ldots,x_n)$

n dimensions

$$\left.\begin{aligned} \text{s.t. } h_1(x_1,\ldots,x_n) &= c_1 \\ h_2(x_1,\ldots,x_n) &= c_2 \\ &\vdots \\ h_m(x_1,\ldots,x_n) &= c_m \end{aligned}\right\}$$

m constraints

$$\Rightarrow L(x_1,\ldots,x_n,\lambda_1,\ldots,\lambda_m) = f(x_1,\ldots,x_n) + \sum_{k=1}^{m} \lambda_k\,(h_k(x_1,\ldots,x_m)-c_k)$$

Background

- Method of Lagrange multipliers – example#1

$$\begin{aligned} &\max\ f(x,y) = x + y \\ &\text{s.t. } x^2 + y^2 = 1 \end{aligned}$$

$$\begin{aligned} \mathcal{L}(x,y,\lambda) &= f(x,y) + \lambda(g(x,y) - c) \\ &= x + y + \lambda(x^2 + y^2 - 1) \end{aligned}$$

$$\begin{aligned} \nabla_{x,y,\lambda}\mathcal{L}(x,y,\lambda) &= \left(\frac{\partial\mathcal{L}}{\partial x}, \frac{\partial\mathcal{L}}{\partial y}, \frac{\partial\mathcal{L}}{\partial\lambda}\right) \\ &= (1 + 2\lambda x, 1 + 2\lambda y, x^2 + y^2 - 1) \end{aligned}$$

$$\nabla_{x,y,\lambda}\mathcal{L}(x,y,\lambda) = 0 \quad \Leftrightarrow \quad \begin{cases} 1 + 2\lambda x = 0 \\ 1 + 2\lambda y = 0 \\ x^2 + y^2 - 1 = 0 \end{cases}$$

$$x = y = -\frac{1}{2\lambda}, \qquad \lambda \neq 0.$$

$$\frac{1}{4\lambda^2} + \frac{1}{4\lambda^2} - 1 = 0, \qquad \lambda = \mp\frac{1}{\sqrt{2}},$$

$$\left(\tfrac{\sqrt{2}}{2}, \tfrac{\sqrt{2}}{2}\right), \qquad \left(-\tfrac{\sqrt{2}}{2}, -\tfrac{\sqrt{2}}{2}\right).$$

$$f\left(\tfrac{\sqrt{2}}{2}, \tfrac{\sqrt{2}}{2}\right) = \sqrt{2}, \qquad f\left(-\tfrac{\sqrt{2}}{2}, -\tfrac{\sqrt{2}}{2}\right) = -\sqrt{2}.$$

From https://en.wikipedia.org/wiki/Lagrange_multiplier

Penalty method and ALM

- Constrained optimization system:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } h_i(\mathbf{x}) = 0 \; \forall i \in I$$

- Penalty method: $\arg\min_{\mathbf{x}} f(\mathbf{x}) + \mu \sum_{i \in I} h_i(\mathbf{x})^2$

penalty parameter
(scalar sufficiently large)

- from hard to soft constraints
- at each iteration the penalty parameter μ is increased
- see also the barrier method

- Augmented Lagrangian Method (ALM)
  - introduced by [Hestenes'69] and [Powell'69]

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\mu}{2} \sum_{i \in I} h_i(\mathbf{x})^2 + \sum_{i \in I} \lambda_i h_i(\mathbf{x})$$

$$= \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\mu}{2} ||h(\mathbf{x})||^2 + \lambda^T h(\mathbf{x})$$

M. J. D. Powell, "A method for nonlinear constraints in minimization problems, in Optimization",1969
M. R. Hestenes, "Multiplier and gradient methods", J. Optim. Theory Appl., 1969

Penalty method and ALM

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } h_i(\mathbf{x}) = 0 \ \forall i \in I$$

$$f(\mathbf{x}) + \mu \sum_{i\in I} h_i(\mathbf{x})^2$$

Penalty method

$$f(\mathbf{x}) + \sum_{i\in I} \lambda_i h_i(\mathbf{x})$$

Lagrangian method

$$f(\mathbf{x}) + \frac{\mu}{2} \sum_{i\in I} h_i(\mathbf{x})^2 + \sum_{i\in I} \lambda_i h_i(\mathbf{x})$$

Augmented Lagrangian method

- Notes
  - ALM is a combination of penalty and Lagrangian methods (“the best of both worlds”)
  - Think about ALM vs penalty method:
    - In ALM, μ needs not go to infinity to solve the original problem (thanks to the Lagrange multiplier), which avoids ill-conditioning
  - Think about ALM vs LM:
    - e.g. a stationary point of the Lagrangian function might not be the optimizer of the original system
    - ΔL=0 is a necessary but not sufficient condition (see KKT)

See proof and details in “Constrained Optimization and Lagrange Multiplied Methods” by Dimitri Bertsekas
http://www.mit.edu/~dimitrib/Constrained-Opt.pdf

“an introduction to optimization”, Edwin K. P. Chong and Stanislaw H. Zak, 2001

Background

- Constrained optimization system:

$$\arg\min_X f(X) \text{ s.t. } h(X) = 0 \qquad h(X) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

- Augmented Lagrangian function for matrix X

$$f(\mathbf{x}) + \frac{\mu}{2}\|h(\mathbf{x})\|^2 + \lambda^T h(\mathbf{x})$$

matrix

$$f(X) \qquad \frac{\mu}{2}\|h(X)\|_F^2 \qquad < Y, h(X) >$$

Y contains all the λ (one per element of h(X))

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}$$

$$< A, B > = trace(A^T B) = \sum_{i,j} A_{i,j} B_{i,j}$$

$$L(X, Y, \mu) = f(X) + \frac{\mu}{2}\|h(X)\|_F^2 + < Y, h(X) >$$

Background

- Solving

$$L(X,Y,\mu) = f(X) + < Y, h(X) > + \frac{\mu}{2}\|h(X)\|_F^2$$

Algorithm 3 (General Method of Augmented Lagrange Multiplier)

1: $\rho \geq 1$.
2: **while** not converged **do**
3: Solve $X_{k+1} = \arg\min_X L(X, Y_k, \mu_k)$.
4: $Y_{k+1} = Y_k + \mu_k h(X_{k+1})$;
5: $\mu_{k+1} = \rho\mu_k$.
6: **end while**
Output: X_k.

[Lin 2009]: Z. Lin, M. Chen, and Y. Ma. “The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices”. Technical report, 2009

Lagrangian

$$\boxed{\arg\min_{A,E} \|A\|_* + \gamma\|E\|_1 \text{ s.t. } A+E=D}$$

$$\arg\min_{X} f(X) \text{ s.t. } h(X)=0$$

$$X=(A,E) \qquad f(X)=\|A\|_*+\gamma\|E\|_1 \qquad h(X)=D-A-E$$

$$\boxed{L(A,E,Z,\mu)=\|A\|_*+\gamma\|E\|_1+<Z,D-A-E>+\frac{\mu}{2}\|D-A-E\|_F^2}$$

[Lin 2009]: Z. Lin, M. Chen, and Y. Ma. “The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices”. Technical report, 2009

Alternating solving

$$\boxed{L(A,E,Z,\mu) = \|A\|_* + \gamma\|E\|_1 + < Z, D-A-E > + \frac{\mu}{2}\|D-A-E\|_F^2}$$

$$\begin{aligned} A^* &= \arg\min_A L(A, E_k, Z_k, \mu_k) \\ &= \arg\min_A \|A\|_* + < Z_k, D-A-E_k > + \frac{\mu_k}{2}\|D-A-E_k\|_F^2 \\ &= \arg\min_A \mu_k^{-1}\|A\|_* + \frac{1}{2}\|A-(D-E_k+\mu^{-1}Z_k)\|_F^2 \end{aligned}$$

$$\begin{aligned} E^* &= \arg\min_E L(A_k, E, Z_k, \mu_k) \\ &= \arg\min_E \gamma\|E\|_1 + < Z_k, D-A-E_k > + \frac{\mu_k}{2}\|D-A_k-E\|_F^2 \\ &= \arg\min_E \gamma\mu_k^{-1}\|E\|_1 + \frac{1}{2}\|E_k-(D-A_k+\mu_k^{-1}Z_k)\|_F^2 \end{aligned}$$

How to solve A and E?
First let’s have a look at thresholding operators

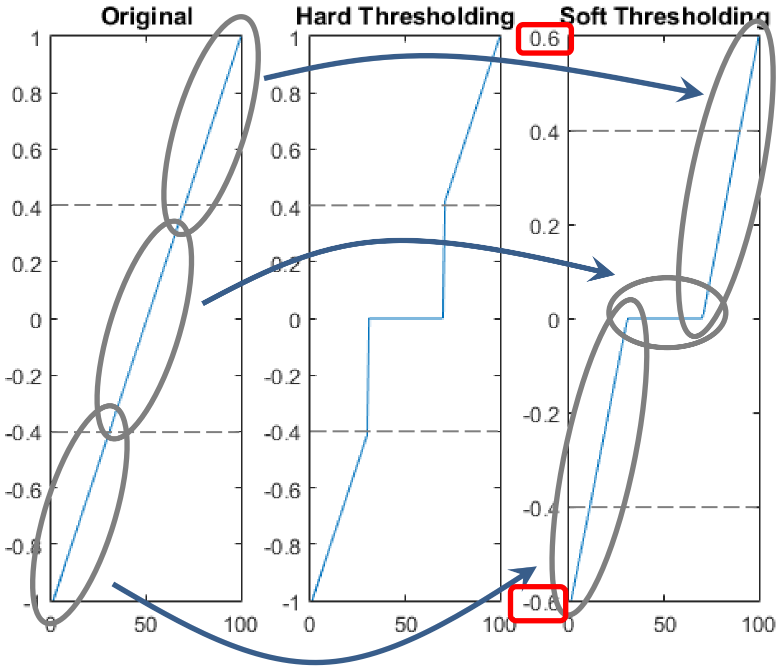
Hard- and Soft-Thresholding

- Hard-thresholding

$$H_\gamma[x] = \begin{cases} x & \text{if } |x| \geq \gamma \\ 0 & \text{otherwise} \end{cases}$$

- Soft-thresholding

$$S_\gamma[x] = \begin{cases} x - \gamma & \text{if } x \geq \gamma \\ x + \gamma & \text{if } x < -\gamma \\ 0 & \text{otherwise} \end{cases}$$



Hard- and Soft-Thresholding

Several equivalent notations

- Hard-thresholding

$$H_\gamma[x] = \begin{cases} x & \text{if } |x| \geq \gamma \\ 0 & \text{otherwise} \end{cases} \iff H_\gamma[x] = x \cdot I(|x| \geq \gamma)$$

- Soft-thresholding

$$S_\gamma[x] = \begin{cases} x - \gamma & \text{if } x \geq \gamma \\ x + \gamma & \text{if } x < -\gamma \\ 0 & \text{otherwise} \end{cases} \iff S_\gamma[x] = sign(x) \cdot (|x| - \gamma) \cdot I(|x| \geq \gamma)$$

$$I(a) = \begin{cases} 1 & \text{if } a \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

$$\iff S_\gamma[x] = sign(x).(|x| - \gamma)_+$$

$$(t)_+ = \begin{cases} t & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\iff (t)_+ = \max(0, t)$$

Solving

- Soft-thresholding (shrinkage) operator

$$\arg\min_X \frac{1}{2}\|X-Y\|_F^2 + \gamma\|X\|_* = \mathbb{S}_\gamma(Y) = US_\gamma[D]V^T$$

See proof in [Cai 2010], Theorem 1

$$Y = UDV^T \quad \text{and} \quad S_\gamma[x] = \begin{cases} x-\gamma & \text{if } x \geq \gamma \\ x+\gamma & \text{if } x < -\gamma \\ 0 & \text{otherwise} \end{cases}$$

(by SVD)

Applied element-wise (i.e. on each singular value)

In practice, singular values are positive, and ɣ also. Therefore, in our case,

$$S_\gamma[x] = \max(0, x-\gamma)$$

[Cai 2010]: J.-F. Cai, E. J. Candes, and Z. Shen. “A singular value thresholding algorithm for matrix completion “, SIAM Journal on Optimization, 2010.

Solving

$$A^* = \arg\min_A \mu_k^{-1} \|A\|_* + \frac{1}{2}\|A - (D - E_k + \mu^{-1} Z_k\|_F^2$$ strictly convex

$$\arg\min_X \frac{1}{2}\|X - Y\|_F^2 + \gamma\|X\|_* = \mathbb{S}_\gamma(Y) = U S_\gamma[D] V^T$$ [Cai 2010]

where $Y = UDV^T$ (by SVD) and $S_\gamma[x] = \begin{cases} x - \gamma & \text{if } x \geq \gamma \\ x + \gamma & \text{if } x < -\gamma \\ 0 & \text{otherwise} \end{cases}$

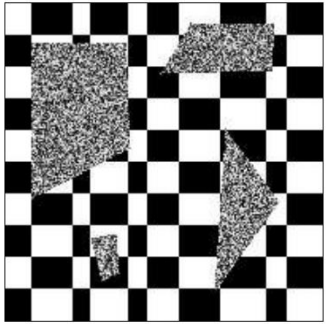
$$E^* = \arg\min_E \gamma\mu_k^{-1} \|E\|_1 + \frac{1}{2}\|E_k - (D - A_k + \mu_k^{-1} Z_k\|_F^2$$

$$\arg\min_X \frac{1}{2}\|X - Y\|_F^2 + \gamma\|X\|_1 = S_\gamma(Y)$$ [Hale 2008]

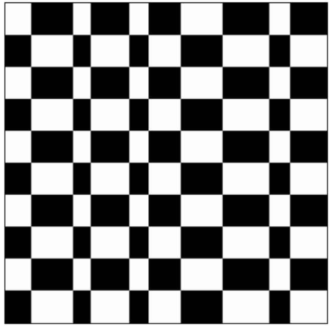
[Cai 2010]: J.-F. Cai, E. J. Candes, and Z. Shen. “A singular value thresholding algorithm for matrix completion “, SIAM Journal on Optimization, 2010.
[Hale 2008]: E. T. Hale, W. Yin, and Y. Zhang. “Fixed-point continuation for l1-minimization: Methodology and convergence”. SIAM Journal on Optimization, 2008
[Lin 2009]: Z. Lin, M. Chen, and Y. Ma. “The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices”. Technical report, 2009

Repairing Low-rank Textures

D (observation)

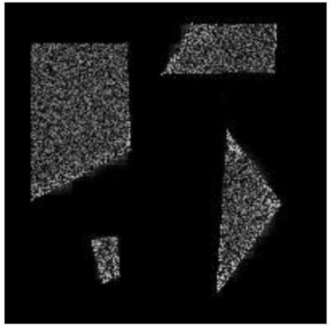


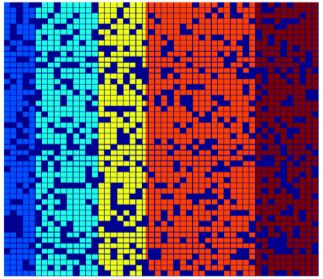
A (low-rank)



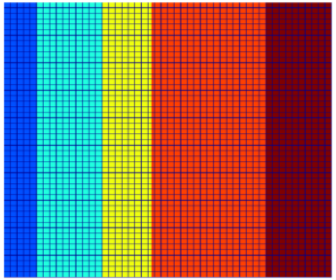
rank(A)=?

E (sparse corruptions)

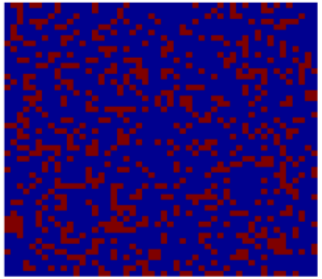




D (observation)



A (low-rank)

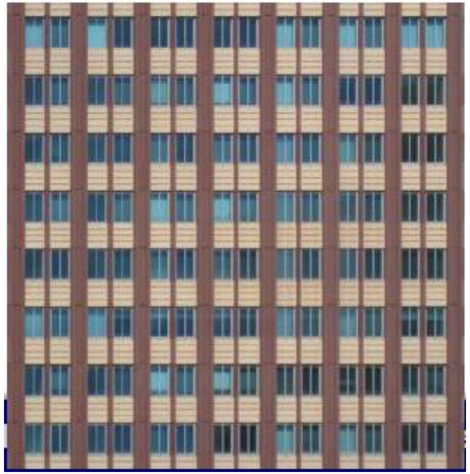


E (sparse)

From ECCV'12 short course "Sparse Representation and Low-Rank Representation in Computer Vision", by Yi Ma, John Wright, and Allen Y. Yang.
http://perception.csl.illinois.edu/matrix-rank/references.html

Repairing Low-rank Textures

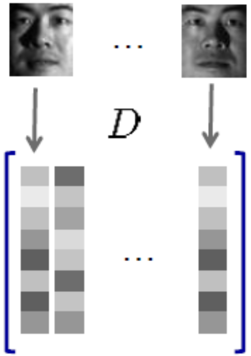
D (observation)



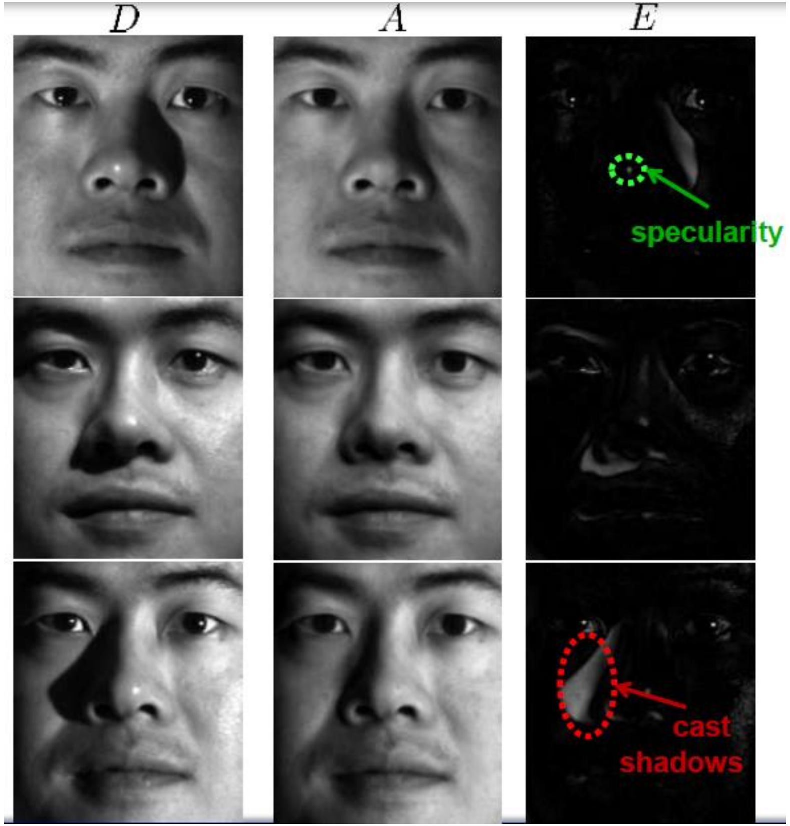
A (low-rank)

Repairing Multiple Correlated Images

58 images of one person
under varying lighting:

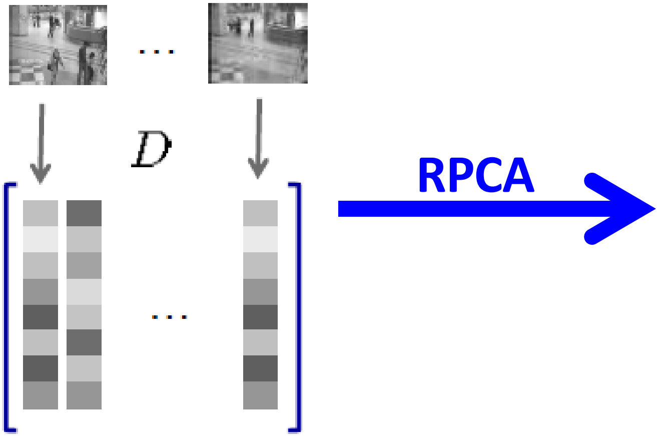


RPCA



Background modeling from video

- Surveillance video
- 200 frames,
- 144 x 172 pixels,
- significant foreground motion



Video D (observation) | *Low-rank approx. A* | *Sparse error E*

RPCA - rank

$$\min_{\mathbf{A},\mathbf{E}} \ \mathrm{rank}(\mathbf{A}) + \lambda\|\mathbf{E}\|_0, \quad \text{s.t. } \mathbf{O} = \mathbf{A} + \mathbf{E}$$

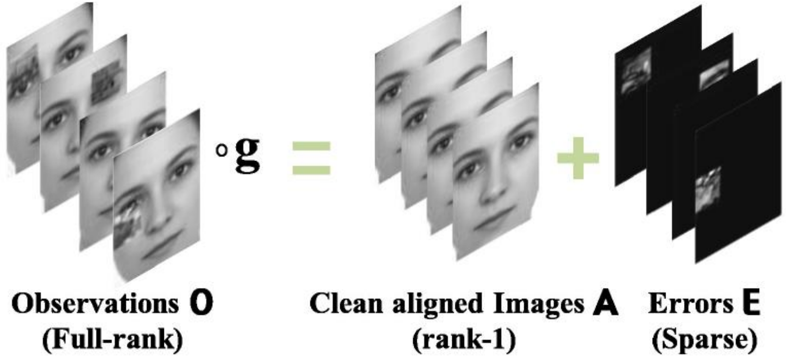
$$\arg\min_{\mathbf{A},\mathbf{E}} \ \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1, \quad \text{s.t. } \mathbf{O} = \mathbf{A} + \mathbf{E}$$

What if you know the rank **N** of **A** in advance?
e.g. **N** = 1 for background subtraction, **N** = 3 for photometric stereo

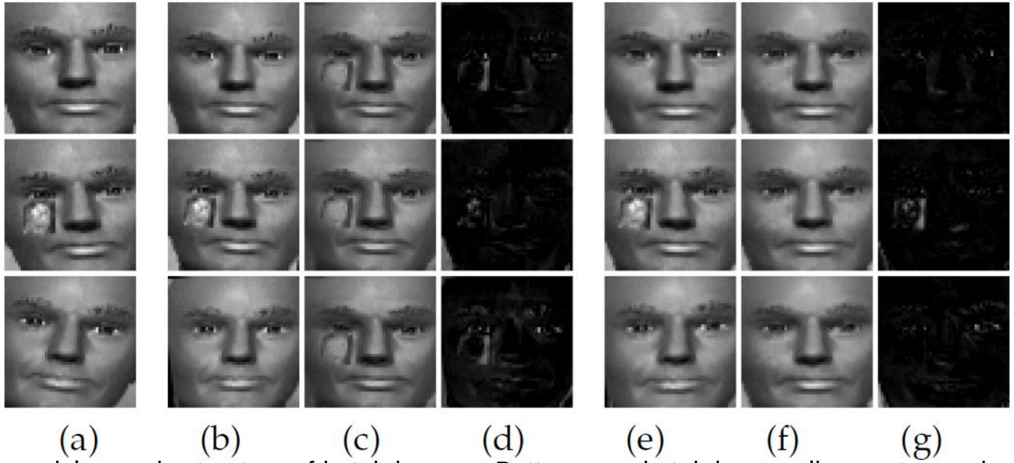
$$\arg\min_{A,E} \|E\|_0 \text{ s.t. } A + E = D \text{ and } rank(A) = N$$??

$$\arg\min_{\mathbf{A},\mathbf{E}} \sum_{i=N+1}^{\min(m,n)} \sigma_i(\mathbf{A}) + \lambda\|\mathbf{E}\|_1, \quad \text{s.t. } \mathbf{O} = \mathbf{A} + \mathbf{E}$$

"Partial Sum Minimization of Singular Values in RPCA for Low-Level Vision" by T.-H. Oh, H. Kim, Y.-W. Tai, J.-C. Bazin, I. S. Kweon, ICCV, 2013



$$\arg\min_{\mathbf{A},\mathbf{E},\mathbf{g}} \sum_{i=N+1}^{\min(m,n)} \sigma_i(\mathbf{A}) + \lambda \|\mathbf{E}\|_1, \ \ \text{s.t} \ \ \mathbf{O} \circ \mathbf{g} = \mathbf{A} + \mathbf{E}.$$



Top row: Illustration of the transformed low-rank structure of batch images. Bottom row: batch image alignment experiments. (a) Three input images. (b-d) The aligned, low-rank, sparse results from Peng et al. [28]. (e-g) The aligned, low-rank, sparse results from the proposed method.

Time-lapse videos

"Partial Sum Minimization of Singular Values in RPCA for Low-Level Vision" by T.-H. Oh, H. Kim, Y.-W. Tai, J.-C. Bazin, I. S. Kweon, ICCV, 2013

Time-lapse videos

Source

Long-Term Motion

Short-Term Motion

Source

Long-Term Motion

Short-Term Motion

"Partial Sum Minimization of Singular Values in RPCA for Low-Level Vision" by T.-H. Oh, H. Kim, Y.-W. Tai, J.-C. Bazin, I. S. Kweon, ICCV, 2013

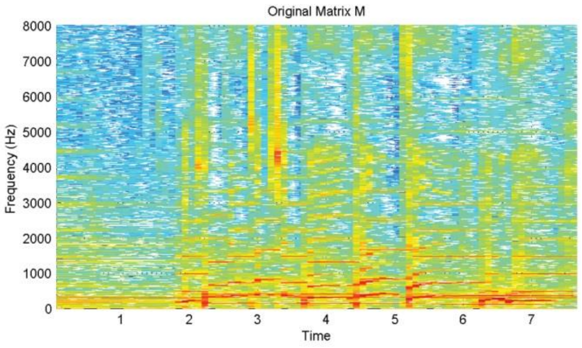
Applications – completion/inpainting

(a) PSNR = 28.41 (b) PSNR = 20.69 (c) PSNR = 24.28 (d) PSNR = 31.23 (e) PSNR = 24.32 (f) PSNR = 44.73 (g) PSNR = 24.99

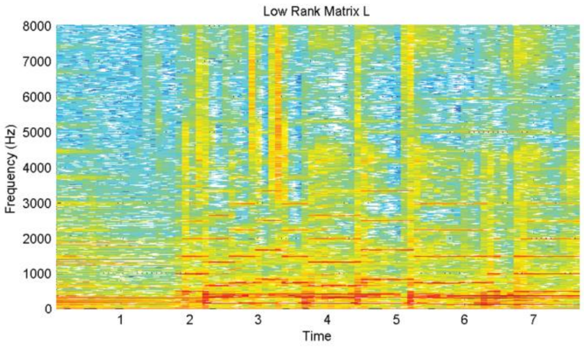
Figure 6. The original images are listed in the first row and the corresponding masked images are listed below. The last row shows the results of TNNR-APGL.

Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, Xiaofei He: Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization, TPAMI, 2013
Debing Zhang, Yao Hu, Jieping Ye, Xuelong Li, Xiaofei He, “Matrix Completion by Truncated Nuclear Norm Regularization“, CVPR 2012

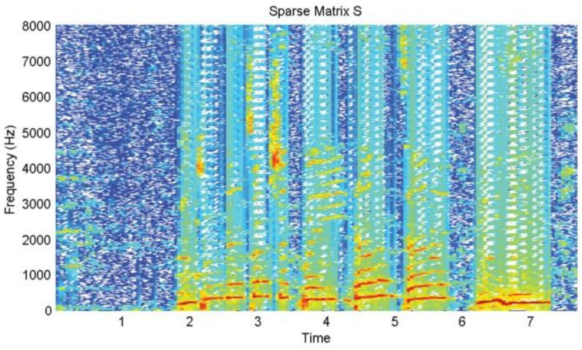
Applications - audio



(a) Original Matrix M



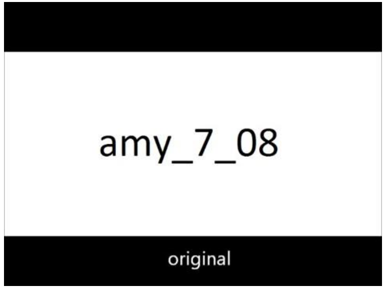
(b) Low-Rank Matrix L



(c) Sparse Matrix M

Goal: Separating singing voices from music accompaniment
- **low-rank** : music accompaniment because of its repetition structure
- singing voices can be regarded as **relatively sparse** within songs

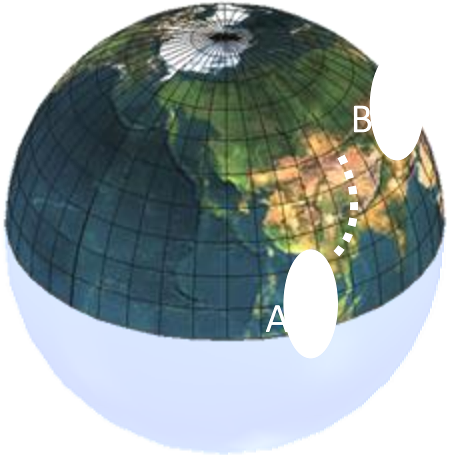
$$\begin{aligned} &\text{minimize} && ||L||_* + \lambda ||S||_1 \\ &\text{subject to} && L + S = M \end{aligned}$$



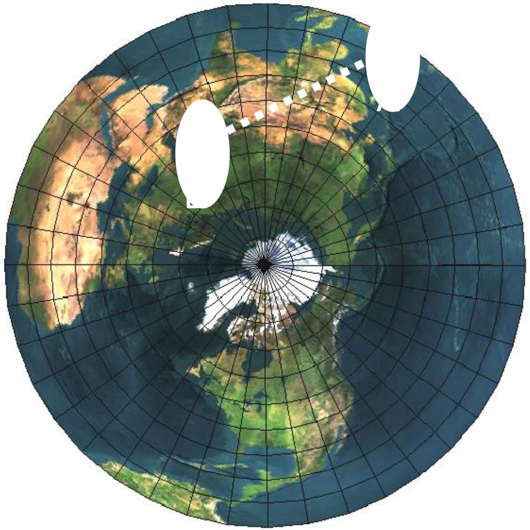
Huang, P-S., Chen, S.D., Smaragdis, P., and Hasegawa-Johnson, M. Singing-voice separation from monaural recordings using robust principal component analysis. In ICASSP, 2012
https://sites.google.com/site/singingvoiceseparationrpca/
Follw-up work: https://sites.google.com/site/deeplearningsourceseparation/

Map projection

- Represent the 3D surface of the Earth in a 2D map by projection



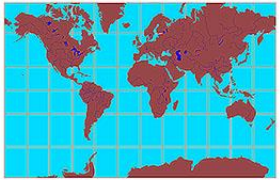
Earth (sphere)



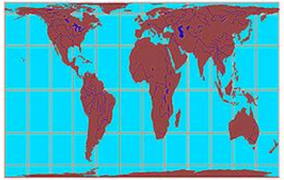
Planar map

Map projection

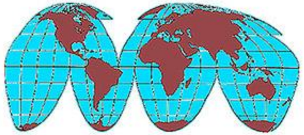
- Unfortunately, impossible to create a distance-preserving planar map of the Earth without distortion!
- Carl F. Gauss proved that a sphere cannot be represented on a plane without distortion
- So, how to do in practice?
  - different map projections have been invented to preserve **some** properties



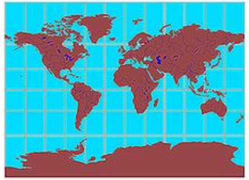
Mercator Projection



Gall-Peters Projection



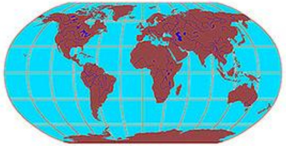
Goode's Homolosine Equal-area Projection



Miller Cylindrical Projection

Mollweide Projection

Sinusoidal Equal-Area Projection

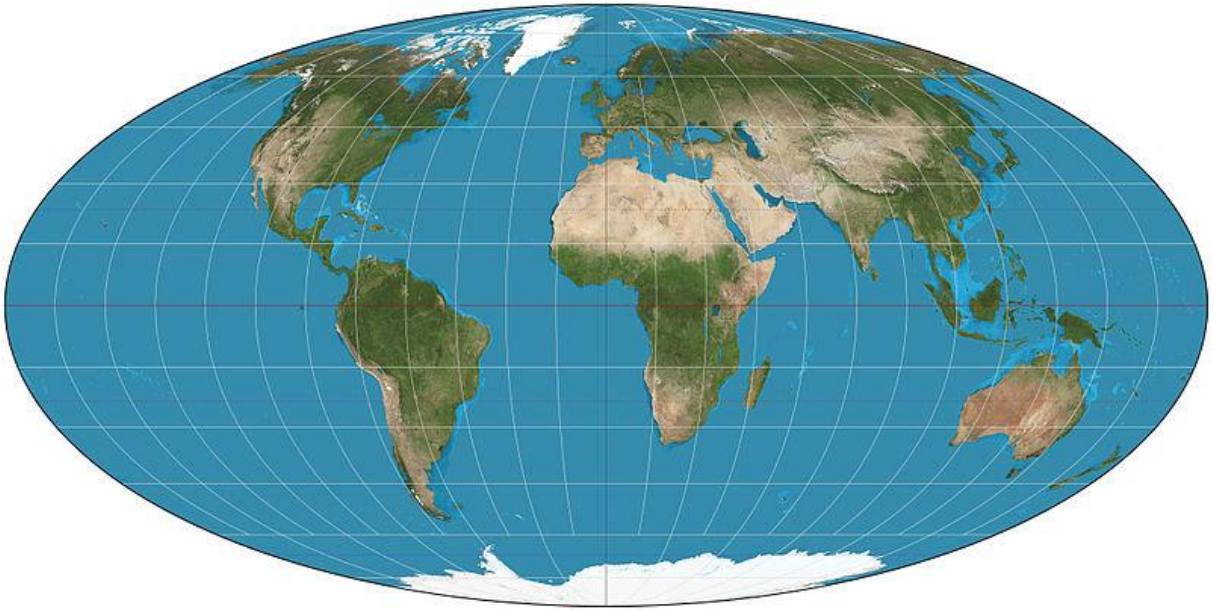


Robinson Projection

Map projection

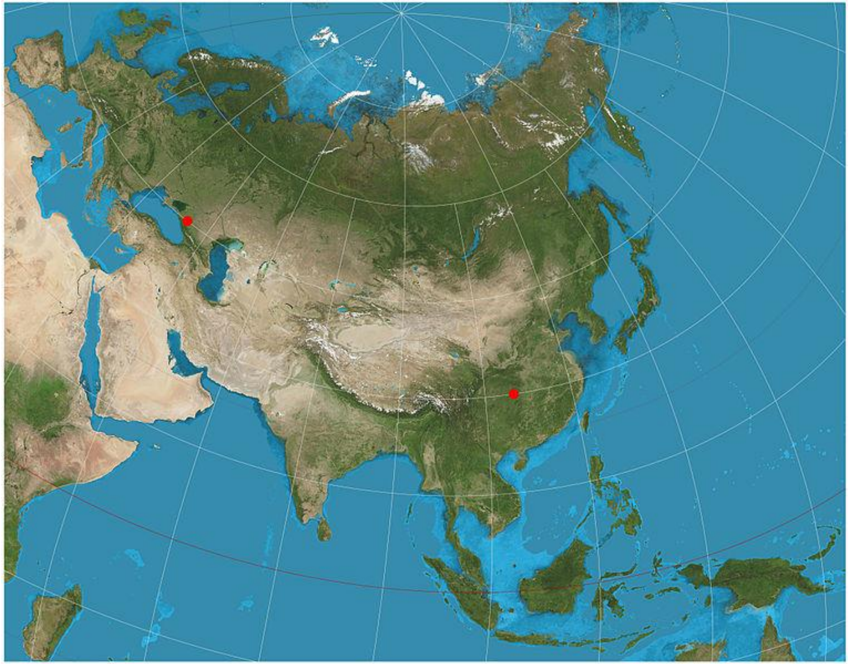
- Popular properties:
  - Preserving area
  - Preserving distance
    - possible only between one or two points and every other point
  - Preserving direction
    - possible only from one or two points to every other point
  - Preserving shape locally

Map projection - example



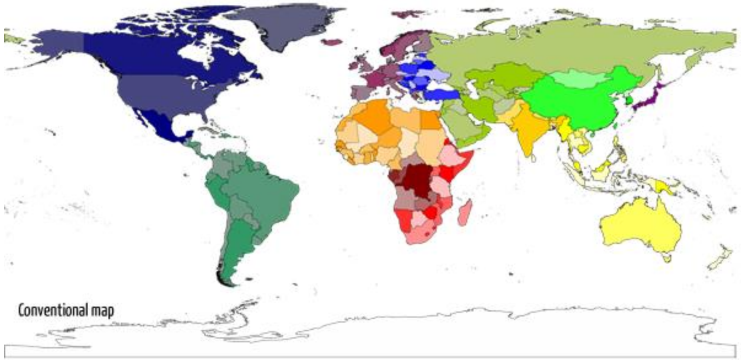
The equal-area Mollweide projection

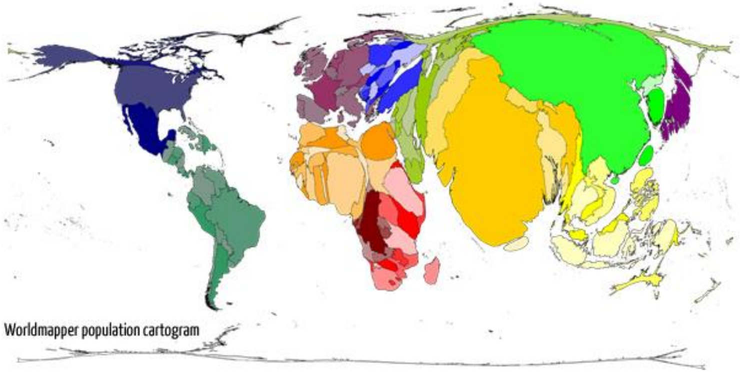
Map projection - example



A two-point equidistant projection of Asia

Map projection - example





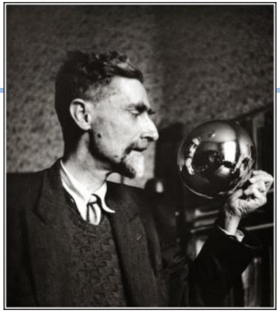
Map projection for non-geometric metrics

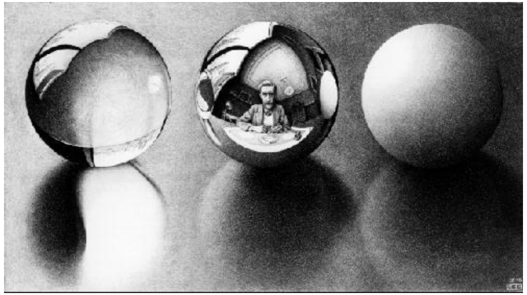
Wide field of view mapping

Omnidirectional cameras and acquired images. From left to right: Nikon Coolpix digital camera and Nikon FC-E8 fish-eye lens; Canon EOS-1Ds and Sigma 8mm-f4-EX fish-eye lens; perspective camera and hyperbolic mirror (catadioptric system); orthographic camera and parabolic mirror (catadioptric system).

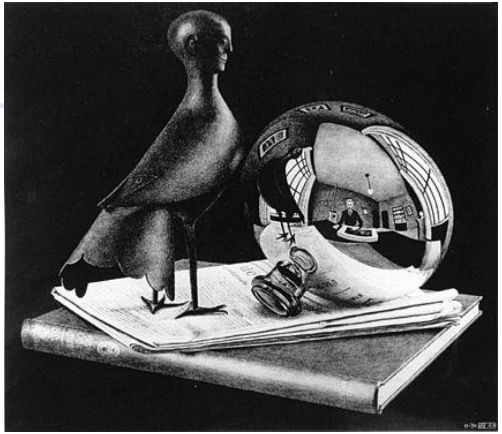
B. Micusik and T. Pajdla,"Structure from motion with wide circular field of view cameras", PAMI, 2006

Escher's work



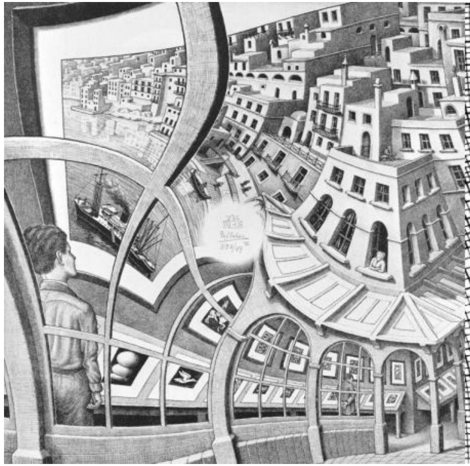


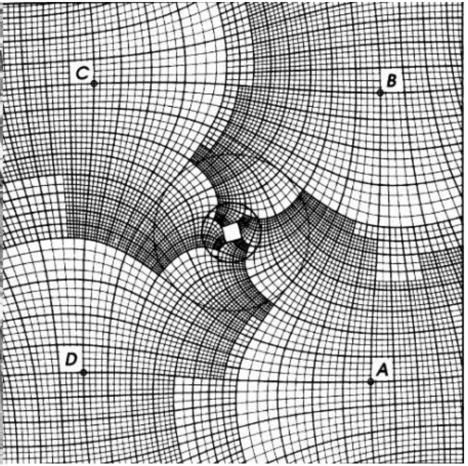
Three Spheres II



Still Life with Spherical Mirror

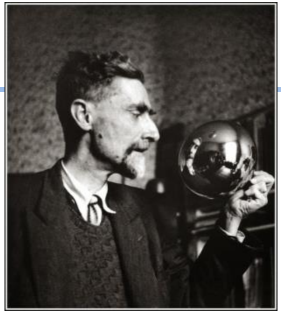
"Hand with Reflecting Sphere" (also known as "Self-Portrait in Spherical Mirror")





"Print Gallery" ("Prententententoonstelling" in Dutch)
And its grid transformation

Mirrors



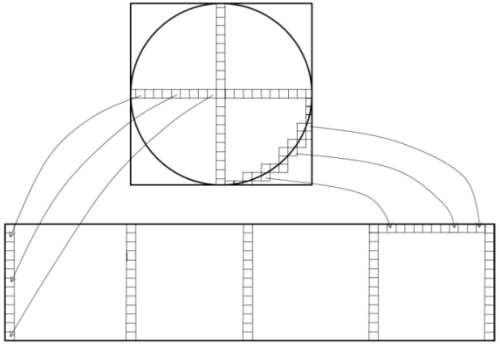
https://www.pinterest.com/pin/19217
7109069533097/

Inflatable Mirror Ball
http://welcome.global-marcom.com/?p=2033

Cloud Gate, in Chicago
http://en.wikipedia.org/wiki/Cloud_Gate

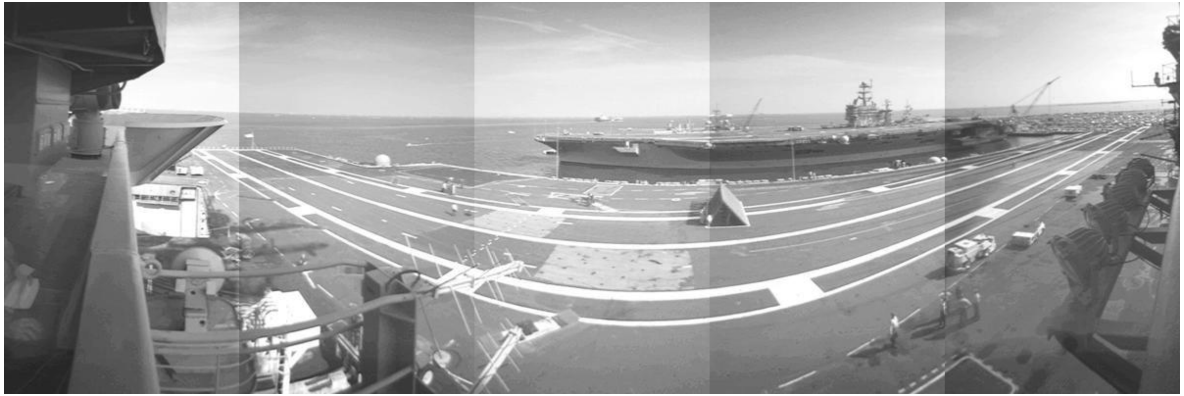
http://www.paulbohman.com/blog/2010/05/r
eflections-with-grace/

Wide field of view mapping



"catadioptric vision for robotic applications", J.C. Bazin and I.S. Kweon, 2011

Wide FOV mapping - planar



Planar rectification of a panoramic image (top) into a planar view (bottom). From www.fullview.com.

Wide FOV mapping - planar

Examples of partial perspective views generated from a single catadioptric image.

S. K. Nayar, “Omnidirectional video camera”, In Proceedings of the 1997 DARPA Image Understanding Workshop

Wide FOV mapping - hybrid

input wide-angle image, with the constrained lines used to compute the unwarping transformation

perspective

mercator

stereographic

hybrid approach

R. Carroll, M. Agrawal, and A. Agarwala,“Optimizing content-preserving projections for wide-angle images”, TOG, 2009

Wide FOV mapping - hybrid

input wide-angle image, with the constrained lines used to compute the unwarping transformation

perspective

mercator

stereographic

hybrid approach

Wide FOV mapping - hybrid

input wide-angle image, with the constrained lines used to compute the unwarping transformation

perspective

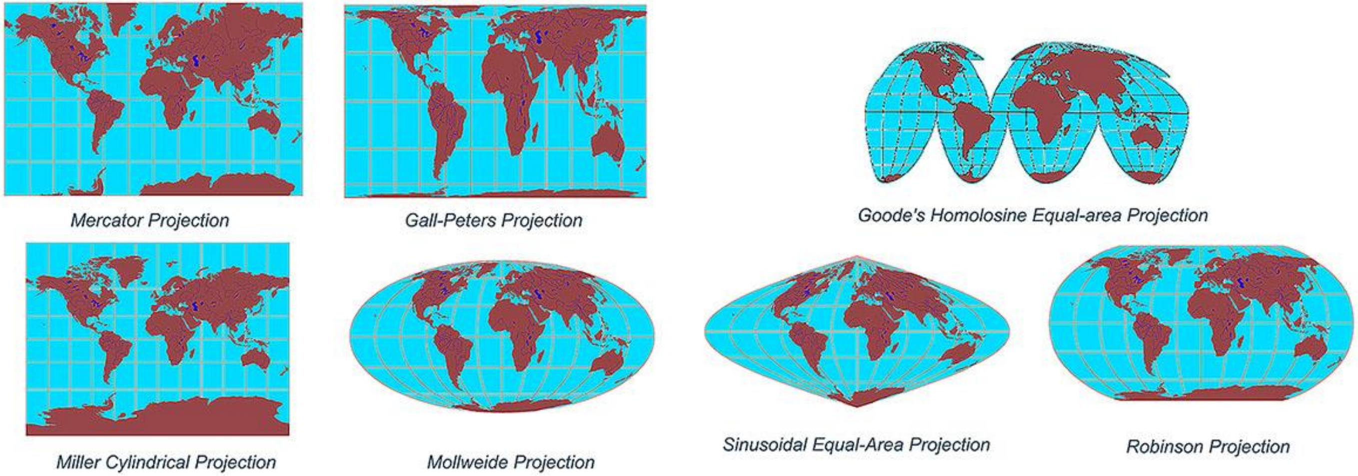
mercator

stereographic

hybrid approach

Map projection

- Unfortunately, impossible to create a distance-preserving planar map of the Earth without distortion!
- Carl F. Gauss proved that a sphere cannot be represented on a plane without distortion
- So, how to do in practice?
  - different map projections have been invented to preserve **some** properties

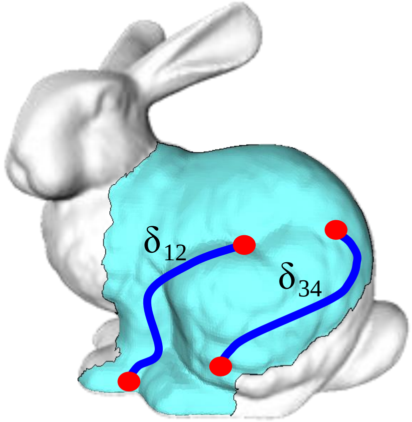


MDS

- Multi-dimensional scaling (MDS)
  - Another classical approach that maps the original high dimensional space to a lower dimensional space
  - Does so by **attempting to preserve pairwise distances/dissimilarities**

- Dissimilarities are distance-like quantities that satisfy the following conditions:

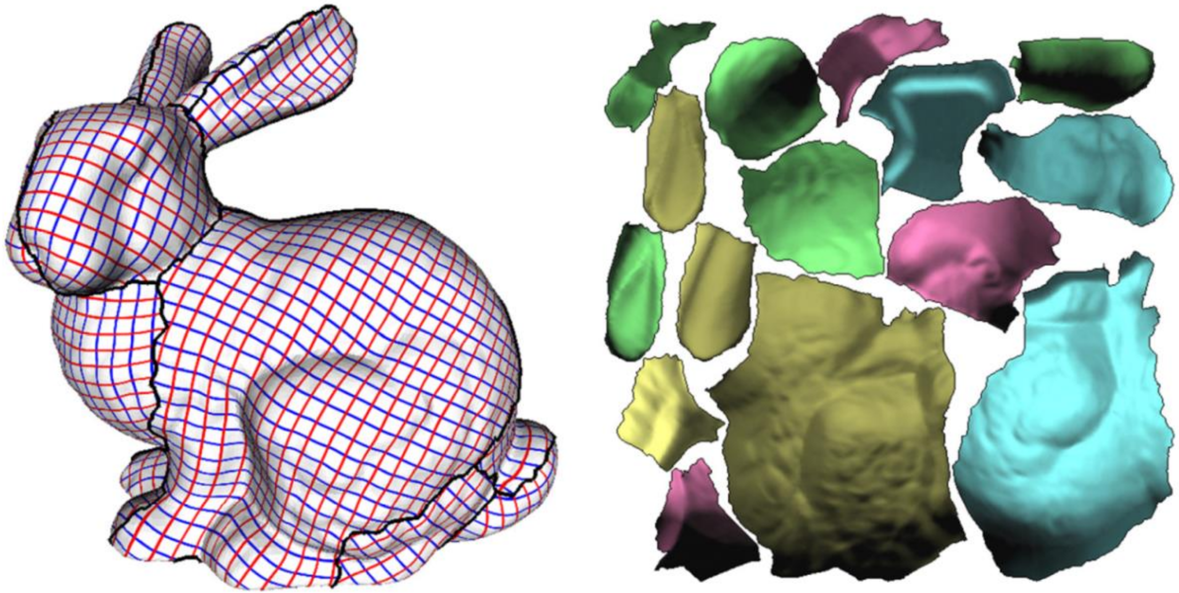
1) $\delta_{ij} \geq 0$
2) $\delta_{ii} = 0$ (self-similarity)
3) $\delta_{ij} = \delta_{ji}$ (symmetry)



- A dissimilarity is **metric** if, in addition, it satisfies

$$\delta_{ij} \leq \delta_{ik} + \delta_{kj} \forall k \quad \text{(triangle inequality)}$$

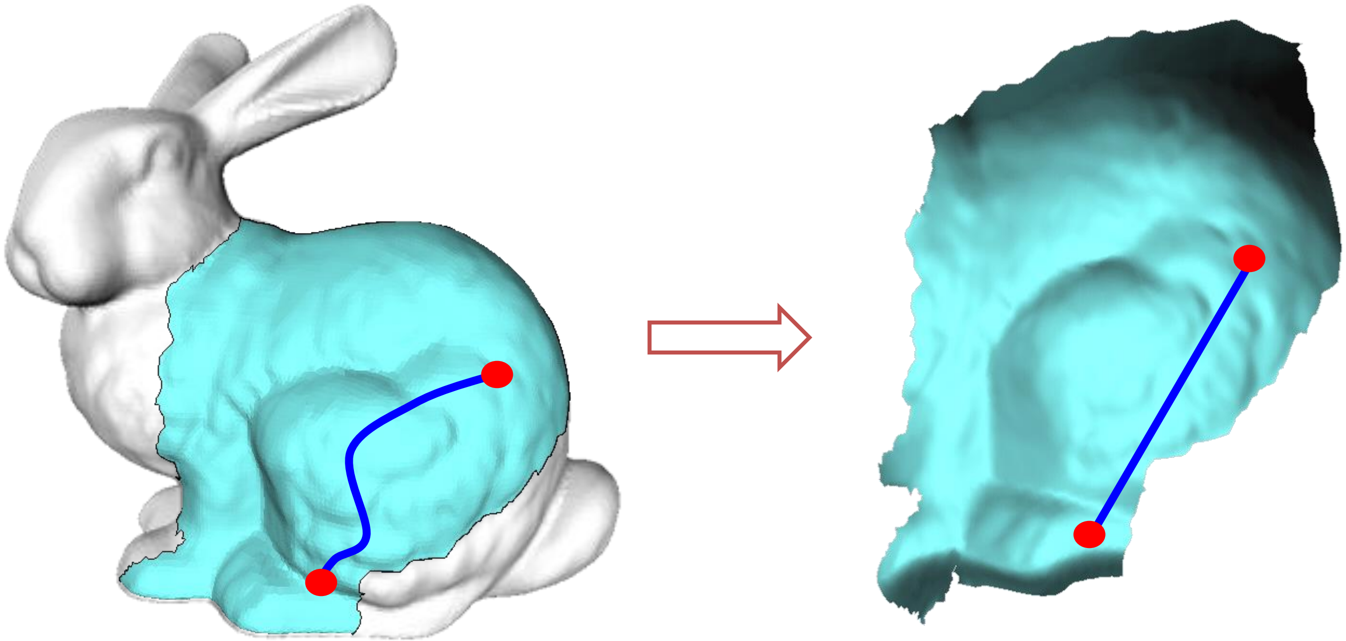
Parameterization



Iso-chart atlas for the Stanford bunny. The model is partitioned into 15 large charts, which can be flattened with lower stretch than previous methods

"Iso-charts: Stretch-driven Mesh Parameterization using Spectral Analysis", Kun Zhou, John Snyder, Baining Guo, Heung-Yeung Shum, SGP, 2004
http://research.microsoft.com/en-us/um/people/johnsny/presentations/isochart.ppt

Goal of Mesh Parametrization



“Iso-charts: Stretch-driven Mesh Parameterization using Spectral Analysis”, Kun Zhou, John Snyder, Baining Guo, Heung-Yeung Shum, SGP, 2004
http://research.microsoft.com/en-us/um/people/johnsny/presentations/isochart.ppt

MDS

- Compute distance/dissimilarity

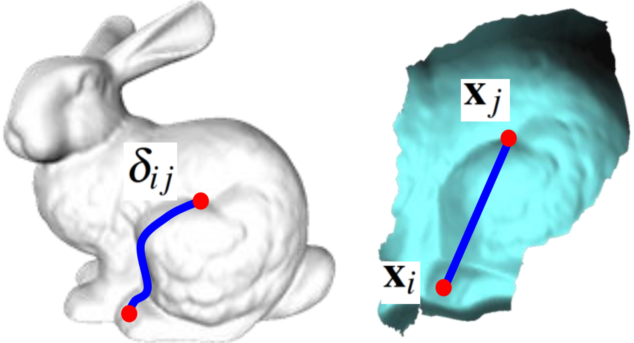
$$\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}$$

- Example: $\mathbf{x}_1 = (3, 1, 4)$ $\quad \mathbf{x}_2 = (3, 5, 7)$

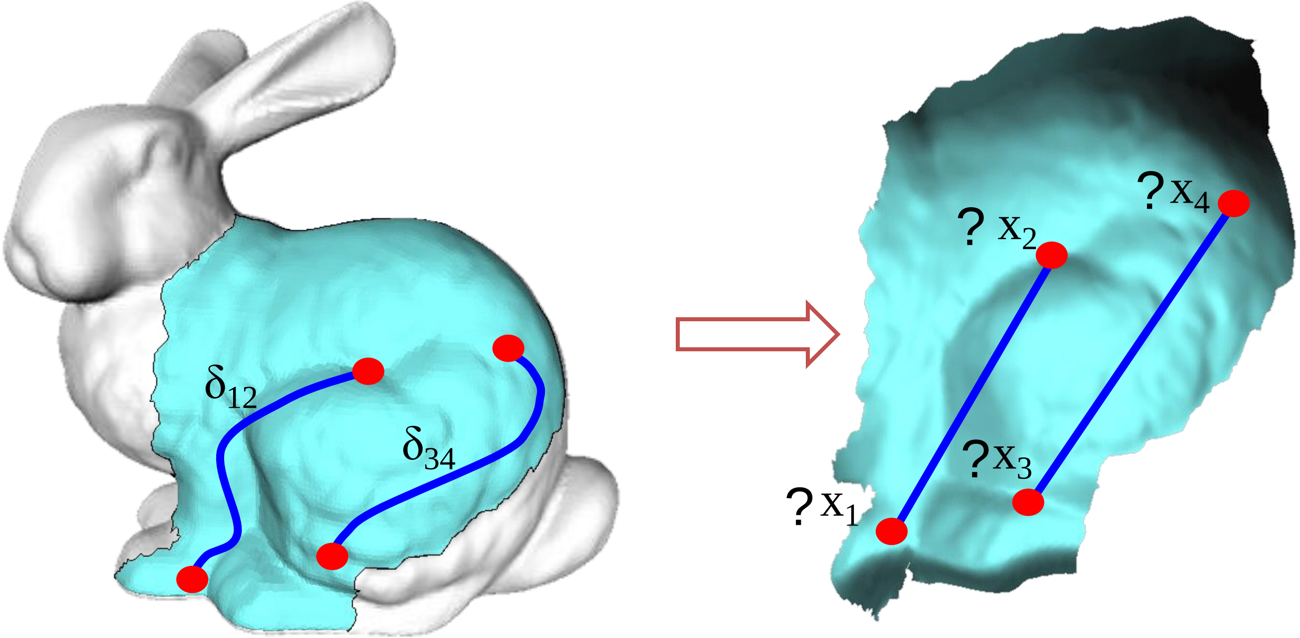
$$\begin{aligned} \|\mathbf{x}_1 - \mathbf{x}_2\|_2 &= \sqrt{(3-3)^2 + (1-5)^2 + (4-7)^2} \\ &= \sqrt{0 + 16 + 9} \\ &= \sqrt{25} = 5 \end{aligned}$$

- Goal:

$$\boxed{\arg\min_{x_1,\ldots,x_n} \sum_{i=1}^{n} \sum_{j>i} (\|\boxed{x_i} - \boxed{x_j}\| - \boxed{\delta_{ij}})^2}$$



MDS



From 3D surface to flat (2D) texturing = dimension reduction

“Iso-charts: Stretch-driven Mesh Parameterization using Spectral Analysis”, Kun Zhou, John Snyder, Baining Guo, Heung-Yeung Shum, SGP, 2004
http://research.microsoft.com/en-us/um/people/johnsny/presentations/isochart.ppt

Classical MDS algorithm

- Compute dissimilarity/distance matrix

$$D = \begin{bmatrix} \delta_{11}^2 & \dots & \delta_{1j}^2 & \dots & \delta_{1n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_{n1}^2 & \dots & \delta_{nj}^2 & \dots & \delta_{nn}^2 \end{bmatrix}$$

- Compute $B = -\frac{1}{2}HDH^T$ where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$

- Eigen decomposition on B: $B = U\Lambda U^T$

- New coordinates are $X = U\Lambda^{1/2}$

$$\tilde{X} = U_k\Lambda_k^{1/2}$$

Proof

- Data points

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{nj} & \dots & x_{nm} \end{bmatrix}$$

- Euclidian distance: $d_{ij}^2 = \sum_{k=1}^{m} (x_{ik} - x_{jk})^2$

- “Scalar distance”: $b_{ij} = \sum_{k=1}^{m} x_{ik} x_{jk} = x_i^T x_j \Rightarrow B = XX^T$

- How to relate Euclidian and scalar distances?

Proof

- Property:

$$
\begin{aligned}
b_{ij} &= -\frac{1}{2}\left(d_{ij}^2 - b_{ii} - b_{jj}\right) \\
&= -\frac{1}{2}\left(d_{ij}^2 - \left(\frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 - \frac{1}{n}\sum_{j=1}^{n} b_{jj}\right) - \left(\frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} b_{ii}\right)\right) \\
&= -\frac{1}{2}\left(d_{ij}^2 - \left(\frac{1}{n} d_{i\bullet}^2 - \frac{1}{n} b_\bullet\right) - \left(\frac{1}{n} d_{\bullet j}^2 - \frac{1}{n} b_\bullet\right)\right) \\
&= -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n} d_{i\bullet}^2 + \frac{1}{n} b_\bullet - \frac{1}{n} d_{\bullet j}^2 + \frac{1}{n} b_\bullet\right) \\
&= -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n} d_{i\bullet}^2 - \frac{1}{n} d_{\bullet j}^2 + \frac{2}{n} b_\bullet\right) \\
&= -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n} d_{i\bullet}^2 - \frac{1}{n} d_{\bullet j}^2 + \frac{1}{n^2} d_{\bullet\bullet}\right)
\end{aligned}
$$

Proof

- Property: $b_{ij} = -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}d_{i\bullet}^2 - \frac{1}{n}d_{\bullet j}^2 + \frac{1}{n^2}d_{\bullet\bullet}\right)$
- Matrix form:

$$B = -\frac{1}{2}HDH^T \quad \text{where} \quad H = I - \frac{1}{n}\underbrace{\mathbf{1}\mathbf{1}^T}$$

contains all the d_{ij}^2
nxn

identity matrix
nxn

matrix of 1
nxn

- Property: $B = U\Lambda U^T$

eigenvalues of *B*
(diagonal matrix)

eigenvectors
of *B*

Since B is symmetric (scalar distance is symmetric)

$$B = XX^T \Rightarrow X = U\Lambda^{1/2}$$

Retaining only the first *k* eigenvectors (lower dimensionality) $\Rightarrow \tilde{X} = U_k\Lambda_k^{1/2}$

General case (d_S)

- Find an embedding that distorts the distances the least
- Stress function is a measure of distortion

$$\sigma_2(x_1, ..., x_N; D_S) = \sum_{i>j} |d_{\mathbb{R}^m}(x_i, x_j) - d_S(s_i, s_j)|^2$$

$$\sigma_\infty(x_1, ..., x_N; D_S) = \max_{i,j=1,...,N} |d_{\mathbb{R}^m}(x_i, x_j) - d_S(s_i, s_j)|$$

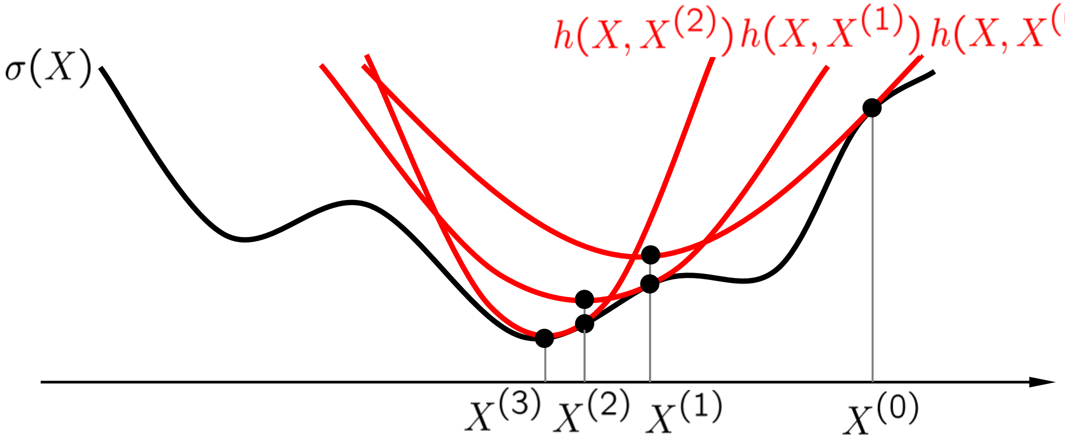
where $x_i = f(s_i)$

- Multidimensional scaling (MDS) problem

$$\boxed{\{x_1^*, ..., x_N^*\} = \underset{x_1,...,x_N}{\operatorname{argmin}} \, \sigma(x_1, ..., x_N)}$$

slides from "Numerical geometry of non-rigid shapes", A. Bronstein, M. Bronstein, R. Kimmel, CVPR'07

Iterative majorization



- Instead of $\sigma(X)$, minimize a convex majorizing function $h(X, Z)$ satisfying

$$h(X, Z) \geq \sigma_2(X), \quad \forall X$$

- above the curve

$$h(Z, Z) = \sigma_2(Z)$$

- touches the curve at the “current” point

- Start with some $X^{(0)}$ and iteratively update

$$\boxed{X^{(k+1)} = \underset{X \in \mathbb{R}^{N \times m}}{\operatorname{argmin}} h(X, X^{(k)})}$$

Matrix expression of the L_2-stress

$$\sigma_2(x_1, \ldots, x_N; D_{\mathcal{S}}) = \sum_{i>j} |d_{\mathbb{R}^m}(x_i, x_j) - d_{\mathcal{S}}(s_i, s_j)|^2$$

$$\boxed{\begin{aligned} \sigma_2(X) &= \sum_{i>j} d_{ij}^2(X) - \sum_{i>j} 2d_{ij}(X) d_{\mathcal{S}}(s_i, s_j) + \sum_{i>j} d_{\mathcal{S}}^2(s_i, s_j) \\ &= \mathrm{tr}(X^\top V X) - 2\mathrm{tr}(X^\top B(X) X) + \sum_{i>j} d_{\mathcal{S}}^2(s_i, s_j) \end{aligned}}$$

- X : a $N \times m$ matrix of coordinates in the embedding space
- V : a $N \times N$ constant matrix with values

$$v_{ij} = \begin{cases} -1 & i \neq j \\ N-1 & i = j \end{cases}$$

- $B(X)$: a $N \times N$ matrix-valued function

$$b_{ij}(X) = \begin{cases} -\frac{d_{\mathcal{S}}(s_i, s_j)}{d_{ij}(X)} & i \neq j, \; d_{ij}(X) \neq 0 \\ 0 & i \neq j, \; d_{ij}(X) \neq 0 \\ -\sum_{k \neq i} b_{ij}(X) & i = j \end{cases}$$

SMACOF algorithm

- Majorize the stress by a convex quadratic function

$$h(X,Z) = \mathrm{tr}(X^\top V X) - 2\underbrace{\mathrm{tr}(X^\top B(Z)Z)}_{\geq \mathrm{tr}(X^\top B(X)X)} + \sum_{i>j} d_{\mathcal{S}}^2(s_i,s_j)$$

- Analytic expression for the minimum of $h(X,Z)$:

$$\nabla_X h(X,Z) = 2VX - 2B(Z)Z = 0$$

$$\Rightarrow X^* = V^\dagger B(Z)Z = \frac{1}{N}B(Z)Z$$

- SMACOF (Scaling by Minimizing a COnvex Function)

$$\boxed{X^{(k+1)} = \frac{1}{N}B(X^{(k)})X^{(k)}}$$

- Comments
  - Guarantees monotonically decreasing sequence of stress values
  - No guarantee of global convergence

Application – texture flattening

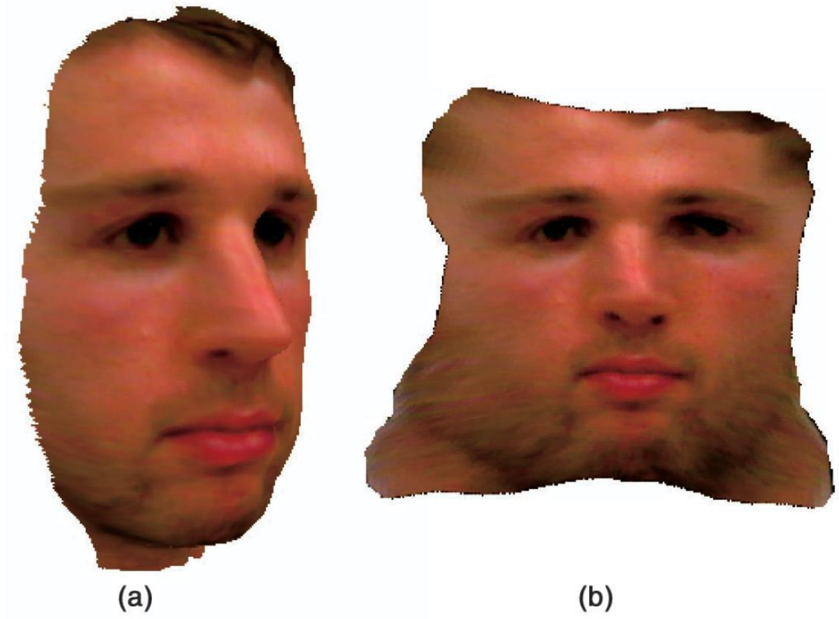


Fig. 3. An example of a face flattening. (a) A 3D reconstruction of a face. (b) The flattened texture image of the face.

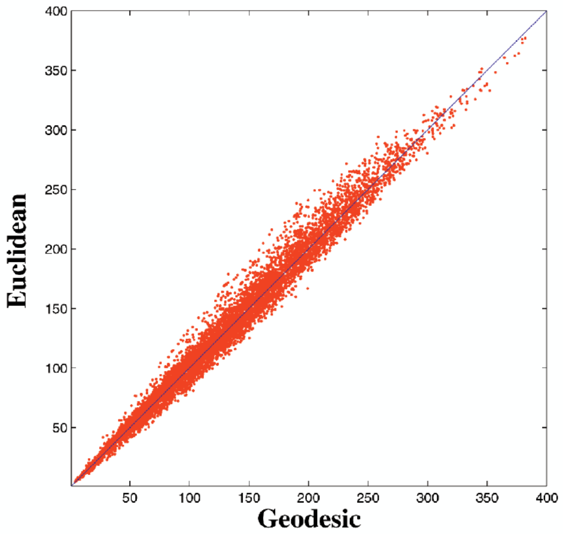
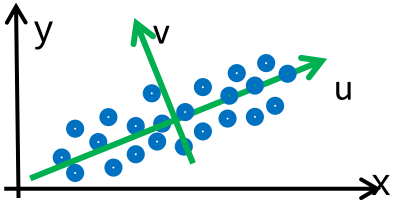


Fig. 5. The geodesic distance on the surface versus the Euclidean distance after flattening. The data corresponds to the face surface shown in Fig. 3. The result approximates the diagonal line, which would have been the geometrically impossible perfect flattening outcome.

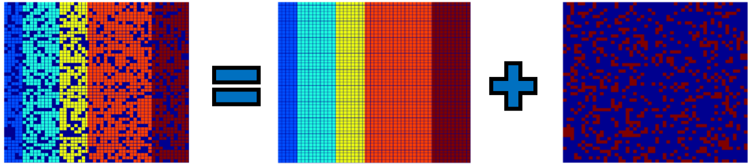
"Texture mapping using surface flattening via multidimensional scaling", G. Zigelman, R. Kimmel, N. Kiryati, TVCG, 2002

Conclusion

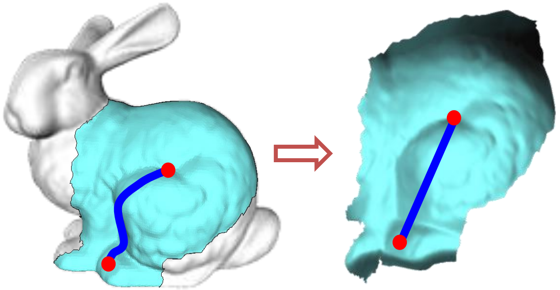
- PCA



- Robust PCA



- MDS



Questions?