Lecture 10: Dimensional Reduction II & Robust Fitting and Matching I

[AIX7021] Computer Vision

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Tentative schedule

week	topic	date
1	Introduction and Basics	09.01
2	Image Process I	09.08
3	Image Process II	09.15
4	(휴강)	09.22
5	Feature Detection and Matching I	09.29
6	Feature Detection and Matching II	10.06
7	Clustering and Segmentation I & II	10.13 & 10.16
8	**Mid-Term Exam**	**10.20**
9	Mid-Term Solution	10.27
10	Dimensional Reduction I	11.03
11	Dimensional Reduction II & Robust Fitting and Matching I	11.10
12	Robust Fitting and Matching II & Object Recognition	11.17
13	Motion and Tracking	11.24
14	Image Classification	12.01
15	**Final exam**	**12.08**

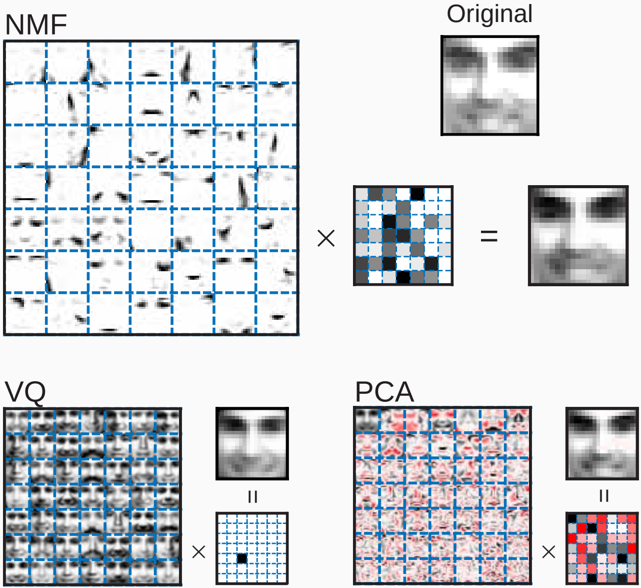
Matrix Factorization for Dimensional Reduction

Matrix factorization

- Principal component analysis (PCA): orthogonal property
- Vector quantization (VQ): unary property
- Non-negative matrix factorization (NMF): non-negativity

$$
\begin{aligned}
\text{PCA: } V \approx WH & \qquad \text{where } W^T W = I \\
\text{VQ: } V \approx WH & \qquad \text{where } H \text{ consists of unary vectors} \\
\text{NMF: } V \approx WH & \qquad \text{where } W_{ij} \geq 0, H_{ij} \geq 0
\end{aligned}
$$

Matrix factorization and face image compression

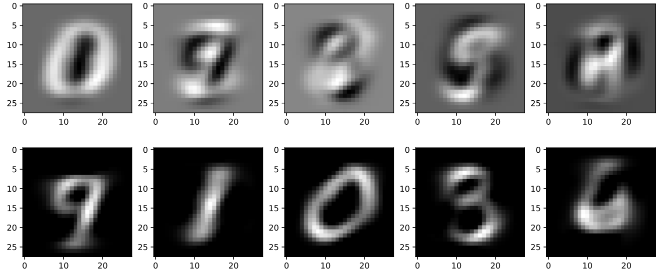


Nonnegative matrix factorization (NMF)

Nonnegative matrix factorization (NMF)

$$\begin{aligned} \text{given} \quad & A \in \Re^{n \times m} \\ \text{find} \quad & (W, H) = \underset{(W,H)}{\arg\min} \|A - WH\|_F^2 \\ \text{s.t.} \quad & W \in \Re_+^{n \times k} \text{ and } H \in \Re_+^{k \times m} \ (k \leq \mathrm{rank}(A)) \end{aligned}$$

PCA vs. NMF for MNIST dataset



Robust Fitting and Matching

After feature detection

Now, we can detect features from images.

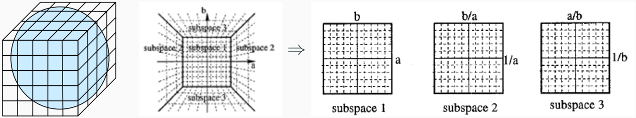
- Features: edge, corner, and other interest points
- Feature descriptors: template, histogram, SIFT, HOG, etc.

What can we do with detected features?

- We may need to model a geometric or semantic notion with the features in the image.
- Many problems are actually fitting problems to the given model.
- Note: Features can also be used to model the appearance (or something else) of the object in the image.

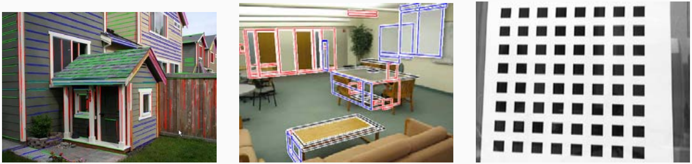
We are going to discuss fitting problem in this class!

Finding vanishing point



(a) (b)

Figure 4.44 Cube map representation for line equations and vanishing points: (a) a cube map surrounding the unit sphere; (b) projecting the half-cube onto three subspaces (Tuytelaars, Van Gool, and Proesmans 1997) © 1997 IEEE.



(a) (b) (c)

Figure 4.45 Real-world vanishing points: (a) architecture (Sinha, Steedly, Szeliski *et al.* 2008), (b) furniture (Mičušìk, Wildenauer, and Košecká 2008) © 2008 IEEE, and (c) calibration patterns (Zhang 2000).

Rectangle detection

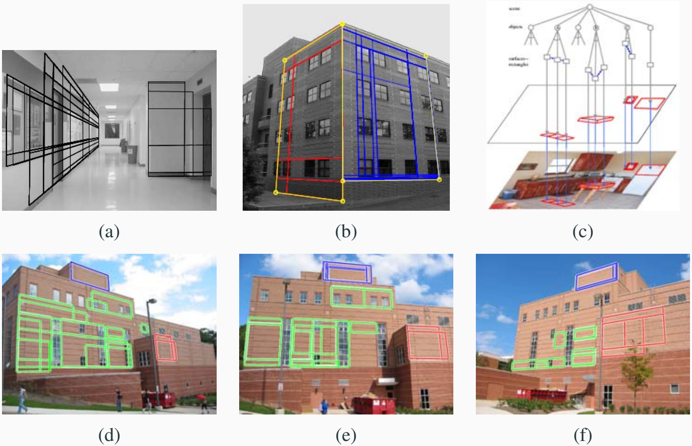
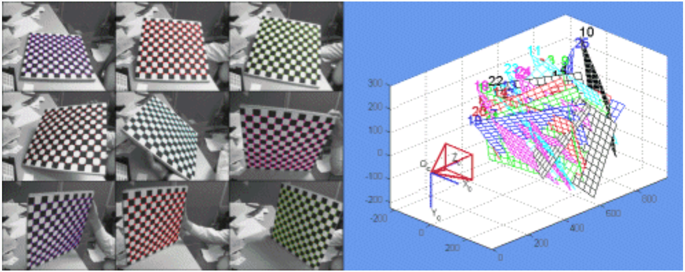
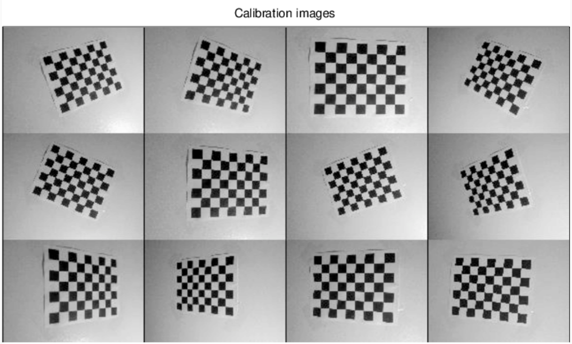


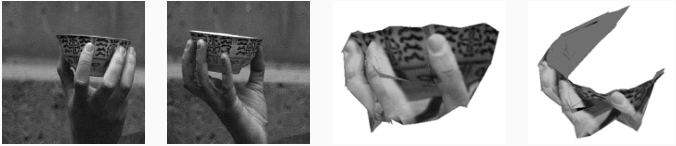
Figure 4.47 Rectangle detection: (a) indoor corridor and (b) building exterior with grouped facades (Košecká and Zhang 2005) © 2005 Elsevier; (c) grammar-based recognition (Han and Zhu 2005) © 2005 IEEE; (d–f) rectangle matching using a plane sweep algorithm (Mičušìk, Wildenauer, and Košecká 2008) © 2008 IEEE.

Camera calibration





3D model extraction



(a) (b) (c) (d)

Figure 7.6 3D teacup model reconstructed from a 240-frame video sequence (Tomasi and Kanade 1992) © 1992 Springer: (a) first frame of video; (b) last frame of video; (c) side view of 3D model; (d) top view of 3D model.

Challenges and strategy

Challenges in fitting and matching

- Noises in data
- Outliers
- Missing data

Strategy to **robust** fitting and matching: hope to handle noises, outliers and missing data effectively

- Extract features
- Compute putative matches
- Loop
  - *Hypothesize* transformation T
  - *Verify* transformation and/or *Search* for other matches consistent with T

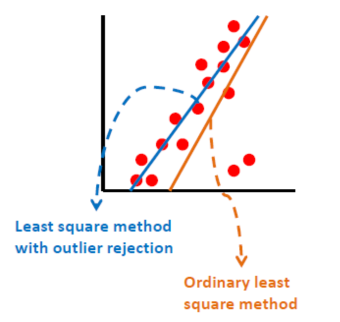
Fitting data with outliers

Effect of outliers

- Ordinary least square methods are very sensitive to noises
- We need to reject the outliers effectively

Handling outliers

- Robust statistics
- RANSAC
- Hough transform



Least Square Method

Ordinary linear least square

Minimize the squared error for given data $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ that should be on $y = \theta_1 x + \theta_2$

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}}_{\boldsymbol{\theta}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} \tag{1}$$

Objective function

$$\min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2 \tag{2}$$

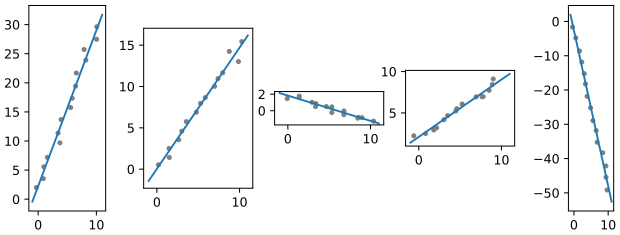
Solution: use pseudo-inverse $\mathbf{X}^\dagger = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2 \tag{3}$$

$$= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4}$$

$$= \mathbf{X}^\dagger\mathbf{y} \tag{5}$$

Ordinary linear least square



Weighted least square

Each sample may have a different weight

Objective function

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} w_i(\theta_1 x_i + \theta_2 - y_i)^2 = \min_{\boldsymbol{\theta}} \|\mathbf{W}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})\|^2 \tag{6}$$

$$\text{where } \mathbf{W} = \mathrm{diag}(\sqrt{w_1}, \sqrt{w_2}, ..., \sqrt{w_n}) \tag{7}$$

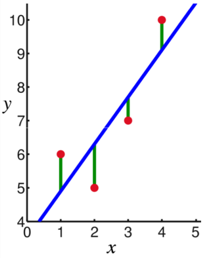
Solution: weighted pseudo-inverse

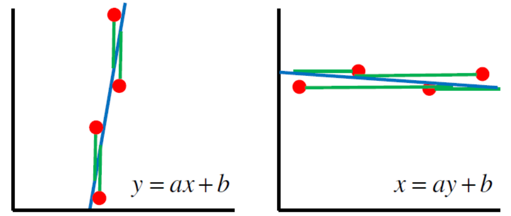
$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{W}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})\|^2 \tag{8}$$

$$= (\mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{y} \tag{9}$$

Limitation of ordinary least square

- Not rotation invariant
- Unable to represent horizontal and vertical lines
- Sensitive to noises when the slope is high

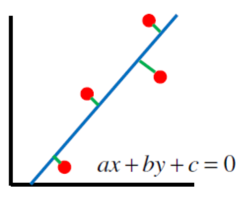




Linear regression with perpendicular distance

= for given data $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ that should be on $\theta_1 x + \theta_2 y + \theta_3 = 0$

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}}_{\boldsymbol{\theta}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{y}} \tag{10}$$



Objective function

$$\min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2 = \min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta}\|^2 \quad \text{trivial solution } \boldsymbol{\theta} = 0 \tag{11}$$

(12)

Total least square

For this reason, we augment this minimization problem with the requirement that $\|\boldsymbol{\theta}\|^2 = 1$ which results in the eigenvalue problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\theta} \quad \text{such that} \quad \|\boldsymbol{\theta}\|^2 = 1 \tag{13}$$

The value of $\boldsymbol{\theta}$ that minimizes this constrained problem is the eigenvector associated with the smallest eigenvalue of $\mathbf{X}^T\mathbf{X}$. This is the same as the last right singular vector of $\mathbf{X}$, since

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{14}$$

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T \tag{15}$$

$$\mathbf{v}_k^T\mathbf{X}^T\mathbf{X}\mathbf{v}_k = \sigma_k^2 \tag{16}$$

which is minimized by selecting the smallest σ_k value.

Appendix

Reference and further reading

- "Chap 8 | Dimensionality Reduction" of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- "Chap 4" of C. Bishop, Pattern Recognition and Machine Learning
- Stephen Boyd, "Lecture 15 | Symmetric matrices, quadratic forms, matrix norm, and SVD" and "Lecture 16 | SVD Applications" of *EE263: Introduction to Linear Dynamical Systems*, Stanford University (2008)
- D. D. Lee and H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature* **401**, 788-791 (1999)
- "Chap 14.2 | Face Recognition" and " A.2 Linear least squares" of R. Szeliski, Computer Vision: Algorithms and Applications
- "Lecture10 | Robust Fitting and Matching" of Bohyung Han, CSED441: Introduction to Computer Vision, POSTECH (2011)