Computer Vision: Lecture 10

Carl Olsson

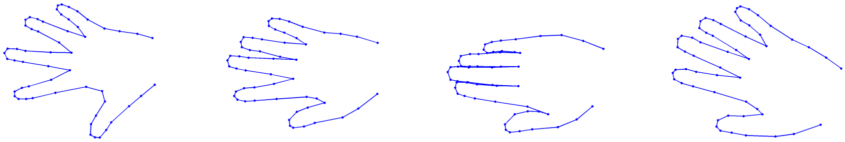
2019-02-20

Todays Lecture

Low rank models

- Factorization
- Low Rank Approximation
- Affine Cameras
- The Missing data problem
- Non-rigid SfM

Dynamic Scenes



Four (out of 40) images of a deformable model with 56 tracked point.

- Point movements are not independent
- No explicit model
- Extract linear model from observed data

Dynamic Scenes

(x_{ij}, y_{ij}) - coordinates of point j in image i.
Measurement matrix

$$M = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ y_{11} & y_{12} & y_{13} & \cdots & y_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ y_{21} & y_{22} & y_{23} & \cdots & y_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \\ y_{m1} & y_{m2} & y_{m3} & \cdots & y_{mn} \end{pmatrix}.$$

Column = point in $\mathbb{R}^{2m}$.
Row = point in $\mathbb{R}^{n}$.

Linear Basis Assumption

Assumption: Each column can be written as a linear combination of a few basis elements $B_1, B_2, ..., B_r$.

Alternatively: The columns of M belong to a r-dimensional ($r << 2m$) subspace spanned by the basis elements $B_1, B_2, ..., B_r$

From linear algebra:

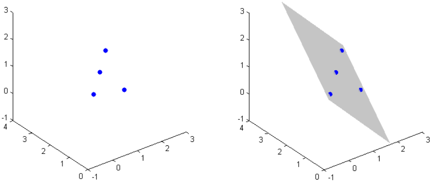
- The **column space** of M consists of all linear combinations of columns in M.
- The **row space** of M consists of all linear combinations of rows in M.
- The **rank** of M is the dimension of the row and column spaces.

Exercise 1

Show that the columns $B_1 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$ and $B_2 = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$ form a basis for the column space of

$$M = \begin{pmatrix} 1 & 2 & 2 & 0 \\ 2 & 3 & 2 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix},$$

and determine its rank.



Exercise 2

Find a 3×2 matrix B and a 2×4 matrix C such that $M = BC^T$ and determine a basis for the row space of M.

- $M = BC^T$ - factorization
- B, C - factors
- B - basis for column space
- C - basis for row space
- $rank(M)$ = number of basis elements (in C and B).
- Factorization not unique:

$$M = BC^T = BHH^{-1}C^T = \tilde{B}\tilde{C}^T.$$

Exercise 2

Find a 3×2 matrix B and a 2×4 matrix C such that $M = BC^T$ and determine a basis for the row space of M.

- $M = BC^T$ - factorization
- B, C - factors
- B - basis for column space
- C - basis for row space
- $rank(M) =$ number of basis elements (in C and B).
- Factorization not unique:

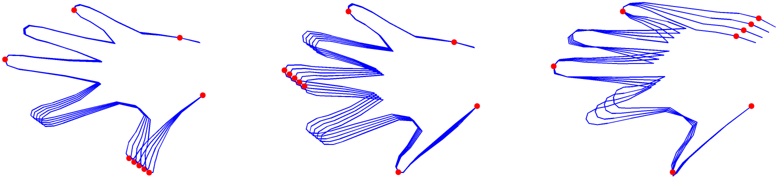
$$M = BC^T = BHH^{-1}C^T = \tilde{B}\tilde{C}^T.$$

Generating New Shapes

Assumption all hand shapes are in the subspace spanned by C.

$(x_1, y_1), (x_2, y_2), \ldots$ unknown point coordinates
B_{new} - 2×5 matrix of parameters

$$\begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ y_1 & y_2 & \ldots & y_n \end{pmatrix} = f(B_{\text{new}}) := B_{\text{new}} C^T,$$



Low Rank Approximation

M is usually not low rank due to noise.
Remove noise before factorization.
ML estimate under Gaussian noise:

$$\min_{\text{rank}(X)=r} \|X - M\|_F^2,$$

Low Rank Approximation

Ekhart-Young 1936: If $\text{rank}(M) = k > r$ and M has the SVD $M = USV^T$ with

$$S = \begin{bmatrix} \text{diag}(\sigma_1, \sigma_2, ..., \sigma_k) & 0 \\ 0 & 0 \end{bmatrix},$$

then the solution to is given by $X = US_rV^T$ where

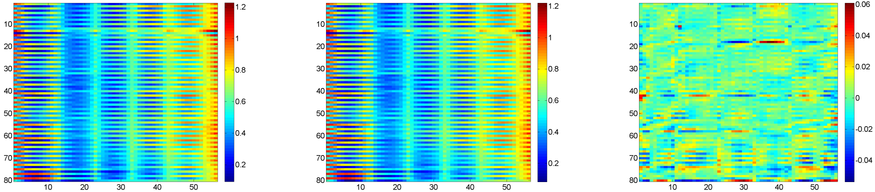
$$S_r = \begin{bmatrix} \text{diag}(\sigma_1, \sigma_2, ..., \sigma_r, 0, 0, ...) & 0 \\ 0 & 0 \end{bmatrix}.$$

Only the first r columns of U and V affect the product US_rV^T.

A factorization $X = BC^T$ is obtained by letting $B = U'S'_r$ and $C = V'$, where $U' = U(:, 1:r)$, $V' = V(:, 1:r)$, $S'_r = S_r(1:r, 1:r)$

Hand dataset:

M $\quad\quad$ X $\quad\quad$ $M - X$



$$\text{rank}(M) = 56 \qquad \text{rank}(X) = 5$$

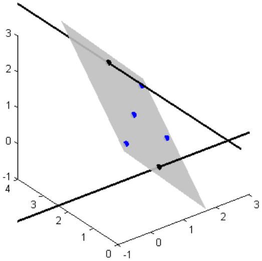
$$80 \cdot 56 = 4480 \qquad (80 + 56) \cdot 5 - 5^2 = 665$$

Exersice 3. (Missing data)

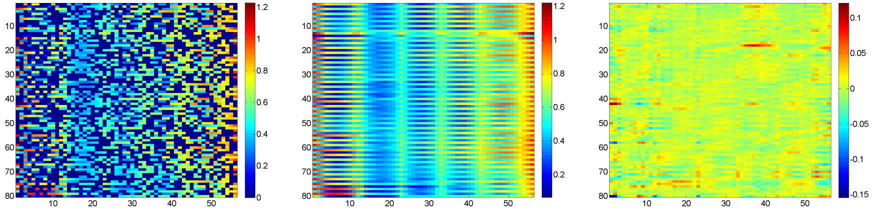
Find the elements m_{15} and m_{26} of

$$M = \begin{pmatrix} 1 & 2 & 2 & 0 & m_{15} & 1 \\ 2 & 3 & 2 & 1 & 1 & m_{26} \\ 1 & 1 & 0 & 1 & 0 & 2 \end{pmatrix},$$

such that $\text{rank}(M) = 2$.

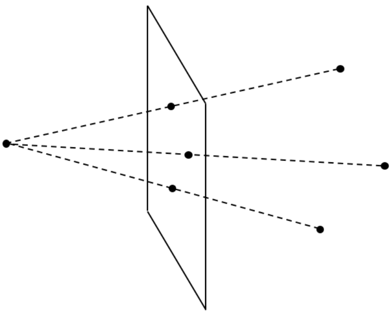


Missing data

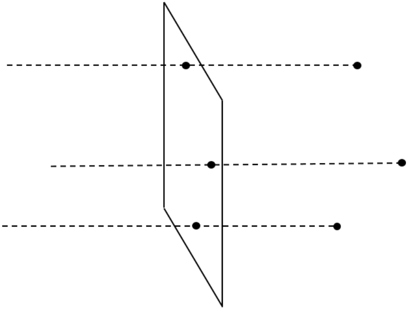


Left - The measurement matrix with roughly 50% missing entries for the hand data set. *Middle* - A rank 5 approximation obtained using local optimization. *Right* - The difference between the true measurement matrix (without missing data) and the obtained rank 5 approximation.

Affine Cameras



Pinhole camera



Affine camera

$$P_{\text{affine}} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}, \quad A - 2 \times 3$$

Projection in regular coordinates

$$x_{ij} = A_i X_j + t_i.$$

Affine Cameras

Solving Structure and Motion via Factorization

Suppose x_{ij} is the projection of X_j in image i. The maximum likelihood solution is obtained by minimizing

$$\sum_{ij} ||x_{ij} - (A_i X_j + t_i)||^2$$

The optimal t_i is given by

$$t_i = \bar{x}_i - A_i \bar{X},$$

where $\bar{X} = \frac{1}{m} \sum_j X_j$ and $\bar{x}_i = \frac{1}{m} \sum_j x_{ij}$.

Affine Cameras

Solving Structure and Motion via Factorization

Changing coordinates, $\tilde{x}_{ij} = x_{ij} - \bar{x}_i$ and $\tilde{X}_i = X_i - \bar{X}$, gives

$$\sum_{ij} ||\tilde{x}_{ij} - A_i \tilde{X}_j||^2.$$

In matrix form

$$|| \underbrace{\begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1m} \\ \tilde{x}_{21} & \tilde{x}_{22} & \dots & \tilde{x}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_{n1} & \tilde{x}_{n2} & \dots & \tilde{x}_{nm} \end{bmatrix}}_{M} - \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 & \dots & \tilde{X}_m \end{bmatrix}}_{\text{rank 3 matrix}} ||^2$$

Affine Cameras

Algorithm

- Re center all images such that the center of mass of the points is zero.
- Form the measurement matrix M.
- Compute the svd:
$$[U, S, V] = svd(M);$$
- A solution is given by the cameras in $U(:, 1:3)$ and the structure in $S(1:3, 1:3) * V(:, 1:3)'$.
- Transform back to the original image coordinates.

Factorization

- Requires all points to be visible in all images.
- Could work for perspective cameras if all points have roughly the same distance to the cameras.

Affine Cameras

Demonstration...

Non-Rigid Structure from Motion

Problems where the 3D points move have higher rank than 3.

Shape Basis Assumption:

The 3D point positions can be written as a linear combination of basis shapes

$$\begin{bmatrix} X_1^t & X_2^t & \ldots & X_n^t \end{bmatrix} = \sum_{k=1}^{K} \alpha_k^t B_k$$

X_i^t - position of scene point i at time t.
B_k - basis shape k (matrix of size $3 \times n$, independent of t)
α_k^t - coefficients at time t.

Non-Rigid Structure from Motion

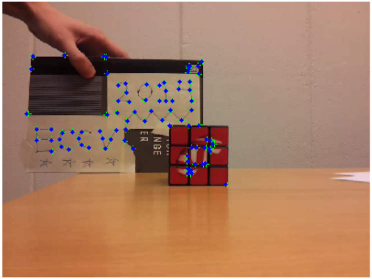
In matrix form: The 3D point positions can be written as a linear combination of basis shapes

$$\begin{bmatrix} X_1^1 & X_2^1 & \dots & X_n^1 \\ X_1^2 & X_2^2 & \dots & X_n^2 \\ \vdots & \vdots & & \vdots \\ X_1^T & X_2^T & \dots & X_n^T \end{bmatrix} = \begin{bmatrix} \alpha_1^1 & \alpha_2^1 & \dots & \alpha_K^1 \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_K^2 \\ \vdots & \vdots & & \vdots \\ \alpha_1^T & \alpha_2^T & \dots & \alpha_K^T \end{bmatrix} \underbrace{\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_K \end{bmatrix}}_{3K \times n}$$

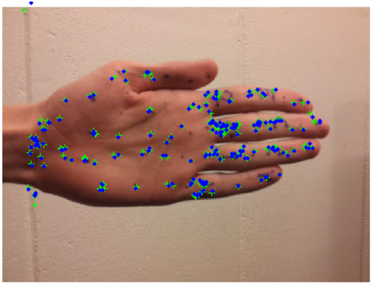
The rank of the matrix (3 times the number of basis elements) describes the complexity of the point motions.

Non-Rigid Structure from Motion

Examples:



rank 3



rank 5

With affine cameras (assuming translations have been eliminated):

$$\begin{bmatrix} x_1^t & x_2^t & \dots & x_n^t \end{bmatrix} = A_t \sum_{k=1}^{K} \alpha_k^t B_k$$

x_i^t projection of point i at time t.
In matrix form:

$$\begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^T & x_2^T & \dots & x_n^T \end{bmatrix} = \begin{bmatrix} \alpha_1^1 A_1 & \alpha_2^1 A_1 & \dots & \alpha_K^1 A_1 \\ \alpha_1^2 A_2 & \alpha_2^2 A_2 & \dots & \alpha_K^2 A_2 \\ \vdots & \vdots & & \vdots \\ \alpha_1^T A_T & \alpha_2^T A_T & \dots & \alpha_K^T A_T \end{bmatrix} \underbrace{\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_K \end{bmatrix}}_{3K \times n}$$

Projective Factorization

The factorization approach can also be used for solving

$$\lambda_{ij}\mathbf{x}_{ij} = P_i X_j,$$

if the depth λ_{ij} is also known!
Minimze

$$\sum_{ij} |\lambda_{ij}\mathbf{x}_{ij} - P_i X_j|^2.$$

Note: This is not a minimization of the reprojection errors (not maximal likelihood estimator).

Projective Factorization

In matrix form

$$|| \underbrace{\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \lambda_{12}\mathbf{x}_{12} & \dots & \lambda_{1n}\mathbf{x}_{1n} \\ \lambda_{21}\mathbf{x}_{21} & \lambda_{22}\mathbf{x}_{22} & \dots & \lambda_{2n}\mathbf{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \lambda_{m2}\mathbf{x}_{m2} & \dots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix}}_{M} - \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}}_{\text{4 columns}} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n \end{bmatrix}}_{\text{4 rows}} ||^2$$

Find the best rank 4 approximation of M.
Use svd:

- $[U, S, V] = svd(M);$
- The cameras are in $U(:, 1 : 4)$.
- The points are in $S(1 : 4, 1 : 4) * V(:, 1 : 4)'$.