Lecture 10: Factorization and Dimensionality Reduction

1 Dynamic Scenes and Linear Basis Models

Previously we have considered image data generated by viewing a rigid scene in a moving camera. In this section we will consider images depicting more general deforming objects. Solving for both camera and object motions is an ill posed problem if points are allowed to move arbitrarily. However, in real scenes point motions are typically highly correlated. Figure 1 shows for images from a dataset consisting of points tracked through a sequence off hand. While all the points are allowed to move only a small subset of simple motions give rise to reasonable hand shapes. Restricting the set of shapes and motions regularizes the problem and makes it solvable.

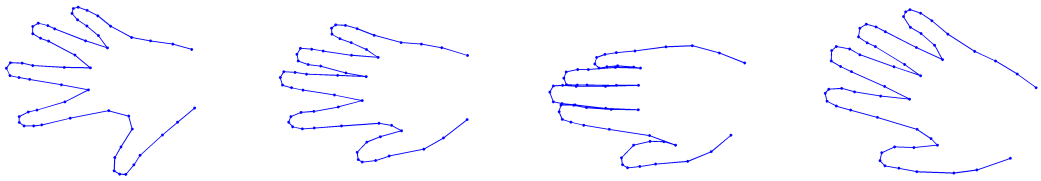

Figure 1: Four (out of 40) images of a deformable model with 56 tracked point. By concatenating x- and y-coordinates from all images each track can be seen as a data point in an 80 dimensional space.

From a mathematical point of view we can see the coordinates from each track as a data point in a high dimensional space. Let (x_{ij}, y_{ij}) be the coordinates of point j in image i. We form the measurement matrix M by

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

Here each column corresponds to the coordinates of a point track and every two consecutive rows correspond to an image. The columns of M can be seen as data points in a $2m$-dimensional space. To model dependence between points a common approach is to assume that the data points can be written as a linear combination of a few basis columns, or equivalently, to assume that the columns of M belong to a low dimensional subspace.

In linear algebra we have the following useful concepts:

- 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

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

and determine its rank.

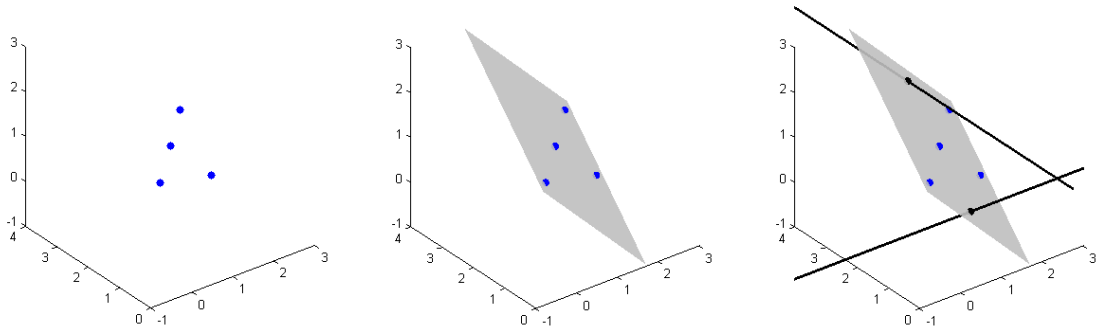

Figure 2: *Left* - The columns of M seen as points in $\mathbb{R}^3$. *Middle* - The factorization 2D basis B_1 and B_2 spanning a plane that all points belong to. *Right* - When a coordinate of a new data point is missing it can be recovered by enforcing that the points lie on the detected plane.

Since the columns of M have 3 entries they can be interpreted as points in $\mathbb{R}^3$, see Figure 2. The fact that we can write all the columns of M as a linear product of the basis vectors B_1 and B_2 shows that they all belong to a 2-dimensional subspace spanned by B_1 and B_2.

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.

As we have seen previously we can interpret the columns of M as data points in $\mathbb{R}^3$ and B as a basis for the 2-dimensional column space. Alternatively we can consider the rows of M as data points in $\mathbb{R}^4$ and C as a basis for the row space. In this case the matrix B contains the coefficients used to form the data points in M from the basis in C. The dimension of these two spaces is the same as the rank of the matrix. For the data shown in Figure 1 the row space will consist of a set of possible hand-shapes whereas the column space consists of a set of point trajectories.

The expression $M = BC^T$ is called a **factorization** of M and B and C are the **factors**. Without any additional constraints there are many possible factorizations. For example if we let $\tilde{B} = BH$ and $\tilde{C} = CH^{-1^T} = CH^{-T}$ we get

$$\tilde{B}\tilde{C}^T = BHH^{-1}C^T = BC^T = M, \tag{3}$$

for any invertible matrix H.

2 Low Rank Approximation

When using real data our measurements are usually corrupted by noise. Therefore the measurement matrix is normally of full rank and it is not possible to find any subspace containing the data points unless the whole space is used. In order to reduce the dimensionality of the data we therefore have to remove noise. Assuming the we add Gaussian noise to a matrix with rank r to get the measurement matrix M the maximum likelihood estimator is

$$\min_{\text{rank}(X)=r} \|X - M\|_F^2, \tag{4}$$

that is, we want to find the matrix X that is closest to M in a least squares sense and has rank$(X) = r$.

The solution to this problem can be obtained from the SVD of M.

Theorem 1 (Ekhart-Young 1936)**.** *If rank$(M) = k > r$ and M has the singular value decomposition $M = USV^T$ with*

$$S = \begin{bmatrix} diag(\sigma_1, \sigma_2, ..., \sigma_k) & 0 \\ 0 & 0 \end{bmatrix}, \tag{5}$$

then the solution to (4) *is given by $X = US_rV^T$ where*

$$S_r = \begin{bmatrix} diag(\sigma_1, \sigma_2, ..., \sigma_r, 0, 0, ...) & 0 \\ 0 & 0 \end{bmatrix}. \tag{6}$$

To find the best rank r approximation of M we thus take its SVD and set all but the first r singular values to zero. We can directly obtain a factorization $X = BC^T$ from the SVD $X = US_rV^T$. Let U' and V' be the first r columns of U and V and S'_r be the top $r \times r$ block of S_r. Since all but the first r columns of U vanish in the multiplication US_r, and similarly all but the first r rows of V^T vanish in S_rV^T, it is clear that

$$X = US_rV^T = U'S'_rV'^T. \tag{7}$$

Thus we can for example let $B = U'S'_r$ and $C = V'$ to obtain $X = BC^T$.

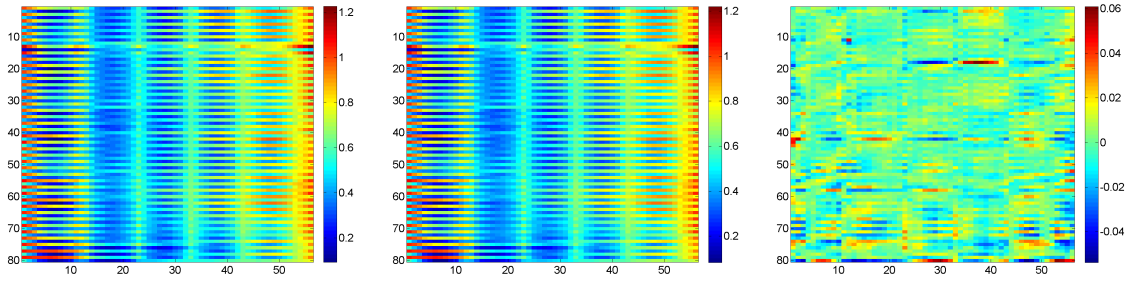

Figure 3: *Left* - The measurement matrix for the hand data set in Figure 1. *Middle* - A rank 5 approximation of the measurement matrix. *Right* - The difference between the measurement matrix and the rank 5 approximation.

Figure 1 shows the measurement matrix for the hand data set from Figure 1. The left image shows the original noisy measurement matrix and the middle image shows the low rank approximation obtained using SVD as described above. The difference is shown in the right image, note the different scales. While the two matrices are very similar the first one has $80 \cdot 56 = 4480$ elements while the second one can be represented with $(80 + 56) \cdot 5 - 5^2 = 665$ numbers.

If the assumption that all hand shapes can be written as a linear combination of five basis shapes is accurate, we can use the obtained factorization to compute new hand shapes. To generate a new image we need to know the x- and y- coordinates of all the points. According to our assumption these should be linear combinations of the rows of C^T. We therefore should have

$$\begin{pmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{pmatrix} = B_{\text{new}}C^T, \tag{8}$$

where $(x_1, y_1), (x_2, y_2), ...$ are the unknown point coordinates and B_{new} is a 2×5 matrix of parameters that specify the point coordinates. Thus we have a linear function $f(B_{\text{new}}) := B_{\text{new}}C^T$ that can be used to generate new images. It can be difficult to find reasonable values for the parameters in B_{new} since there are many possible choices of C^T that can be obtained from the factorization. One way to generate reasonable values is to instead specify the coordinates of a few (in this case 5) points and determine the values of B_{new} from these. If we for example specify the first 5 point positions we get

$$\begin{pmatrix} x_1 & x_2 & \dots & x_5 \\ y_1 & y_2 & \dots & y_5 \end{pmatrix} = B_{\text{new}}C^T_{5\times 5}, \tag{9}$$

where $C_{5\times5}$ are the first five rows of C. Assuming that $C_{5\times5}$ is invertible we get

$$B_{\text{new}} = \begin{pmatrix} x_1 & x_2 & \dots & x_5 \\ y_1 & y_2 & \dots & y_5 \end{pmatrix} C_{5\times5}^{-T}. \tag{10}$$

Inserting (10) in (8) gives a new linear function that takes 5 point positions and returns the coordinates for all the points 56.

The choice of using the first five points to construct the function above is arbitrary and we may use any collection of 5 points. For numerical reasons it is often beneficial to use points that are evenly distributed in the scene. In Figure 4 we selected the five marked with a red dot and varied the coordinates of one of them. The blue curves show the positions obtained for all points by using (10) and (8). As long as we stay in the vicinity of the previously observed shapes the silhouettes look reasonable, however if we move too far away from previously observed shapes the result may start to look unreasonable, as in the right example of Figure 4).

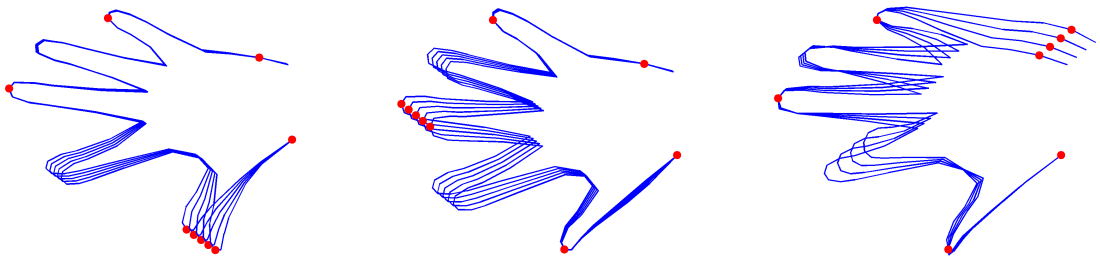

Figure 4: Modifying the point coordinates of 5 of the observed points to generate new shapes using the computed row space of C^T.

3 Structure from Motion with Affine Cameras

If we use the so called affine camera model the structure from motion problem is closely related to factorization. In the affine camera model viewing rays are assumed to be perpendicular to the image plane, see Figure 5. If the observed scene points are roughly at the same distance from the camera this model often works well as an approximation of the pinhole camera model.

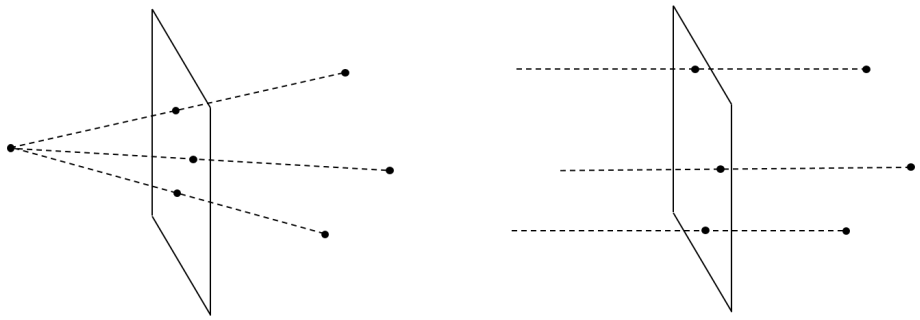

Figure 5: *Left* - A regular pinhole camera where all viewing rays go through the camera center. *Right* - The affine camera model where the viewing rays are parallel to the normal of the image ray.

If P is the camera matrix of an affine camera then its third row is of the form $[0\ 0\ 0\ t_3]$. Since the scale of the camera matrix is arbitrary we may assume that $t_3 = 1$, and therefore the camera matrix has the form

$$P = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}, \tag{11}$$

where A is a 2×3 matrix and t is a 2×1 vector. By using regular Cartesian coordinates for both image points and scene points the camera equations can be simplified. If x_{ij} is the projection of the scene point X_j in the affine cameras P_i then the projection can be written

$$x_{ij} = A_i X_j + t_i. \tag{12}$$

To find he maximum likelihood estimate we therefore need to solve

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \|x_{ij} - A_i X_j + t_i\|^2. \tag{13}$$

Dy differentiating with respect to t_i it can be seen that 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 $\bar{x}_i = \frac{1}{m} \sum_j x_{ij}$. To simplify the problem we therefore change coordinates so that all these mean values are zero by translating all image points and scene points. Using $\tilde{x}_{ij} = x_{ij} - \bar{x}_i$ and $\tilde{X}_i = X_i - \bar{X}$, gives the simplified problem

$$\min \sum_{ij} ||\tilde{x}_{ij} - A_i \tilde{X}_j||^2. \tag{14}$$

In matrix form we can write this as

$$\min \left\| \underbrace{\begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1m} \\ \tilde{x}_{21} & \tilde{x}_{22} & \dots & \tilde{x}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_{n1} & \tilde{x}_{n2} & \dots & \tilde{x}_{nm} \end{bmatrix}}_{M} - \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 & \dots & \tilde{X}_m \end{bmatrix}}_{\text{rank 3 matrix}} \right\|^2. \tag{15}$$

Since the A_i has only 3 columns the second term will be rank 3 matrix. Thus our problem is to find the matrix of rank 3 that best approximates M and factor it into AX^T where A contains the cameras and X^T the 3D points. As described in the previous section the best approximating matrix can be found by computing the SVD of M and setting all but the first 3 singular values to zero.

We summarize the algorithm for affine cameras here:

1. Re-center all images so that the center of mass of the image points is zero in each image.
2. Form the measurement matrix M.
3. Compute the SVD:
$$M = USV^T. \tag{16}$$
4. A solution can be found by extracting the cameras from $U(:, 1:3)$ and the structure from $S(1:3, 1:3) * V(:, 1:3)'$.
5. Transform back the solution to the original image coordinates.

Note that the approach only works when all points are visible in all images. Furthermore, the camera model is affine, which is a simplification. This is often a good approximation when the scene point have roughly the same depth.

4 The Missing Data Problem

Now lets consider the degrees of freedom (DOF) of the set of matrices that can be written BC^T if B is of size $m \times r$ and C is $n \times r$. The two factors have $(m+n)r$ elements in total, however since the product (3) is independent of the $r \times r$ matrix H we get $(m+n)r - r^2$. In contrast a general $m \times n$ matrix has mn elements. Since $mn = (m+n)n - n^2 = (m+n)m - m^2$ it is clear that BC^T has fewer degrees of freedom if $r < m$ or $r < n$. In the example in Exercise 1 we have $mn = 12$ and $(m+n)r - r^2 = 10$. Thus BC^T can be seen as a compressed representation where we can specify the content of M using fewer values than its total number of elements.

The fact that BC^T has fewer DOF than a general matrix of size $m \times n$ makes it possible to recover M even if only a subset of the elements of M is known.

Exercise 3. Find the elements m_{15} and m_{26} of

$$M = \begin{pmatrix} 1 & 2 & 2 & 0 & m_{15} & 1 \\ 2 & 3 & 2 & 1 & 1 & m_{26} \\ 1 & 1 & 0 & 1 & 0 & 2 \end{pmatrix}, \tag{17}$$

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

To the right Figure 2 we give a geometric interpretation of the above exercise. Then varying m_{15} and m_{26} we obtain the points of two lines. If the unknown points are assumed to belong to the same subspace as the rest of the columns of M they have to lie in the intersections between the lines and the plane.

When some elements of M are unknown we cannot use the SVD to recover a low rank approximation of M. If W is a matrix that has $w_{ij} = 1$ if m_{ij} is known and $w_{ij} = 0$ otherwise we seek a solution to

$$\min_{B,C} \|W \odot (BC^T - M)\|_F^2, \tag{18}$$

where $\odot$ denotes element-wise multiplication. This problem has to be solved for relatively high levels of missing data using iterative local methods. Figure 6 shows a solution recovered from a measurement matrix containing roughly 50% of the data from the hand data set.

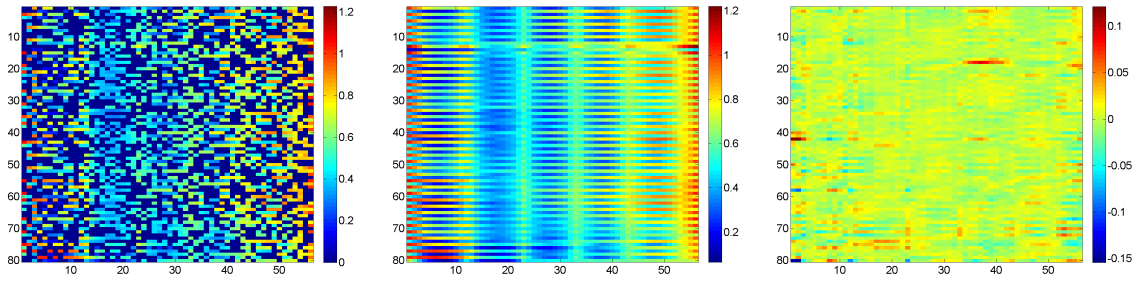

Figure 6: *Left* - The measurement matrix with roughly 50% missing entries for the hand data set in Figure 1. *Middle* - A rank 5 approximation obtained using local optimization. *Right* - The difference between the true measurement matrix (without missing data) and the obtained rank 5 approximation.