CS 83: Computer Vision	Winter 2024

Quiz 3 — 01/31/2024

Prof. Pediredla — ***Student: Amittai Siavava***

Credit Statement

All typed work is my own, with reference to class notes especially on homographies and transformations. I also referred to some of my earlier notes on linear algebra (from **MATH 22**) on the interpretations of matrices and their null-spaces, column-spaces, and row-spaces.

Problem 1.

Given a set N of points $\mathbf{p}_i = (x_i, y_i), i \in \{1, \ldots, N\}$, in the image plane, we wish to find the best line passing through those points.

(i) One way to solve this problem is to find (a, b) that most closely satisfy the equations $y_i = ax_i + b$, in a least-squares sense. Write these equations in the form of a *heterogeneous* least-squares problem $A\mathbf{x} = \mathbf{b}$, where $\mathbf{x} = (a, b)^T$, and give an expression for the least-squares estimate of $\mathbf{x}$. Give a geometric interpretation of the error being minimized, and use a simple graph to visualize the error. Does this make sense when fitting a line to points in an image?

The set of N points satisfies the following system of equations:

$$\begin{aligned} y_1 &= ax_1 + b \\ y_2 &= ax_2 + b \\ &\vdots \\ y_N &= ax_N + b. \end{aligned}$$

We can rewrite the system as the following heterogeneous *least-squares* problem:

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{b}}.$$

To find the least-squares estimate of $\mathbf{x}$, we solve the normal equations $A^T A\mathbf{x} = A^T\mathbf{b}$:

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

$$\underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{b}}$$

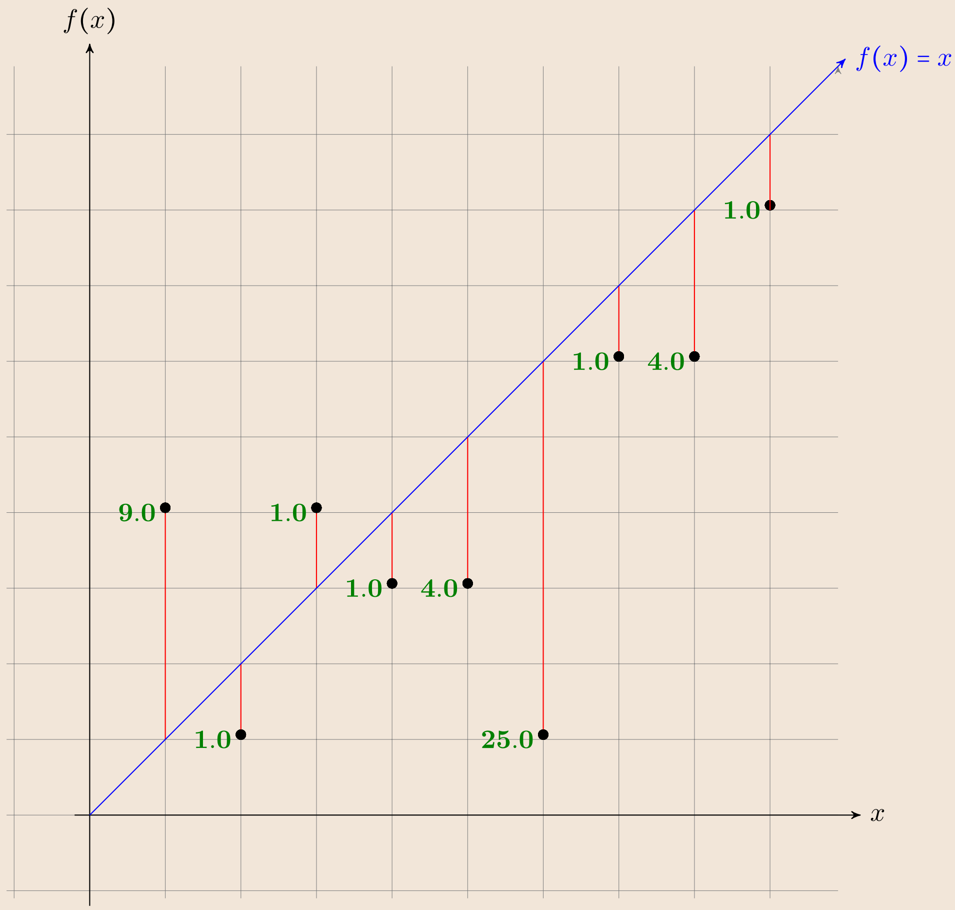
$$\underbrace{\begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & N \end{bmatrix}}_{A^T A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} y_i \end{bmatrix}}_{A^T\mathbf{b}}$$

Square matrix on the left is invertible, [a] so we can solve for $\mathbf{x}$ by left-multiplying both sides by its inverse:

$$\underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & N \end{bmatrix}^{-1}}_{(A^T A)^{-1}} \underbrace{\begin{bmatrix} \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} y_i \end{bmatrix}}_{A^T\mathbf{b}}$$

The error being minimized is the sum of the square *vertical* distances of each point from the line:

$$\mathcal{E} = \sum_{i=1}^{N} \left\| y_i - \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} x_i \\ 1 \end{bmatrix} \right\|^2 .$$



$$\mathcal{E} = 0 + 9.0 + 1.0 + 1.0 + 1.0 + 4.0 + 25.0 + 1.0 + 4.0 + 1.0 = 47$$

Figure 1. Graph of points, fitted line, and error measures.

Notice that the error associated to each point varies quadratically. This is often desirable when fitting a line to points in an image since it penalizes points that are farther from the line by a greater scale than points that are closer to the line. *However, if there are outliers in the dataset, then using this method to fit a line to points in an image can result in one outlier point having a large effect on the fitted line and making it less accurate.*

[a] If the matrix is not invertible, then the system of equations has either no solution (i.e. is inconsistent) or infinitely many solutions.

(ii) Another way to solve this problem is to find $\ell = (a, b, c)$, defined up to scale, that most closely satisfies the equations $ax_i + by_i + c = 0$, in a least-squares sense. Write these equations in the form of a *homogeneous* least-squares problem $A\ell = 0$, where $\ell = (a, b, c)^T$ and $\ell \neq 0$. This problem has a trivial solution (zero vector) which is not of much use. Describe some ways of avoiding this trivial solution and corresponding algorithms for solving the resulting optimization problem. Is this approach more or less useful than the previous one? Why?

Hint: Think about how we can express the distance of a point from a line.

The set of N points satisfies the following system of equations:

$$ax_1 + by_1 + c = 0$$
$$ax_2 + by_2 + c = 0$$
$$\vdots$$
$$ax_N + by_N + c = 0.$$

We can rewrite the system as the following homogeneous *least-squares* problem:

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\ell} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}}.$$

To find the least-squares estimate of ℓ, we solve the normal equations $A^T A\ell = A^T\mathbf{0}$:

$$\underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\ell} = \underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}}$$

$$\underbrace{\begin{bmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i y_i & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i y_i & \sum_{i=1}^N y_i^2 & \sum_{i=1}^N y_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N y_i & N \end{bmatrix}}_{A^T A} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\ell} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{A^T\mathbf{0}}$$

The square matrix to the left is invertible, *but the vector on the right is the zero vector, so multiplying it by the inverse of the matrix will not give us useful information about ℓ.*

Instead, we want to find the *null space* of A,

$$\mathbf{Nul}\, A = \left\{ \vec{v} \in \mathbb{R}^3 \mid A\vec{v} = \mathbf{0} \right\}.$$

On finding the null space of $A^T A$:

We can find a basis for the null space of $A^T A$ (called the *null basis*) by reducing the matrix

$$\begin{bmatrix} A^T A & \mid & \mathbf{0} \end{bmatrix}$$

to reduced row echelon form. The null basis will be the set of vectors corresponding to the columns of the reduced matrix that do not have a pivot column in the original matrix.

On Comparative Utility of Approaches:

The second approach is, in a lot of ways, more useful than the first.

1. The right-hand side of the homogeneous least-squares problem is the zero vector, so we don't have to worry about complicated computation of $A^T \mathbf{b}$.
2. The homogeneous approach also gives us a basis for *all* vectors in the null space of $A^T A$. If there were multiple lines that are best fits for the points they they would be easier to find since they would be linear combinations of the null basis vectors.

Problem 2.

The equation for a conic in the plane using inhomogeneous coordinates (x, y) is

$$ax^2 + bxy + cy^2 + dx + ey + f = 0. \tag{2.1}$$

(a) Suppose you have a given a set of inhomogeneous points $\mathbf{x}_i = (x_i, y_i)$, $i \in \{1, \ldots, N\}$. Derive an expression for the least squares estimate of a conic $\mathbf{c} = (a, b, c, d, e, f)$ passing through those points.

Note: your expression may take the form of a null vector or eigenvector of a matrix. If so, you must provide expressions for the matrix elements.

To determine the matching conic, we need to solve the equation $A\mathbf{x} = \mathbf{0}$:

$$\underbrace{\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & x_Ny_N & y_N^2 & x_N & y_N & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}}$$

We solve for $\mathbf{c}$ by first multiplying both sides by A^T in order to convert the matrix on the left into a square matrix, then either [1] multiplying both sides by the inverse of the square matrix if the right-hand side is nonzero or [2] finding the null space if the right-hand side is zero. In this case, approach [2] is the desired one:

$$\underbrace{\begin{bmatrix} x_1^2 & x_2^2 & \cdots & x_N^2 \\ x_1y_1 & x_2y_2 & \cdots & x_Ny_N \\ y_1^2 & y_2^2 & \cdots & y_N^2 \\ x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & x_Ny_N & y_N^2 & x_N & y_N & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{A\mathbf{0}}$$

$$
\underbrace{\begin{bmatrix}
\sum_{i=1}^N x_i^4 & \sum_{i=1}^N x_i^3 y_i & \sum_{i=1}^N x_i^2 y_i^2 & \sum_{i=1}^N x_i^3 & \sum_{i=1}^N x_i^2 y_i & \sum_{i=1}^N x_i^2 \\
\sum_{i=1}^N x_i^3 y_i & \sum_{i=1}^N x_i^2 y_i^2 & \sum_{i=1}^N x_i y_i^3 & \sum_{i=1}^N x_i^2 y_i & \sum_{i=1}^N x_i y_i^2 & \sum_{i=1}^N x_i y_i \\
\sum_{i=1}^N x_i^2 y_i^2 & \sum_{i=1}^N x_i y_i^3 & \sum_{i=1}^N y_i^4 & \sum_{i=1}^N x_i y_i^2 & \sum_{i=1}^N y_i^3 & \sum_{i=1}^N y_i^2 \\
\sum_{i=1}^N x_i^3 & \sum_{i=1}^N x_i^2 y_i & \sum_{i=1}^N x_i y_i^2 & \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i y_i & \sum_{i=1}^N x_i \\
\sum_{i=1}^N x_i^2 y_i & \sum_{i=1}^N x_i y_i^2 & \sum_{i=1}^N y_i^3 & \sum_{i=1}^N x_i y_i & \sum_{i=1}^N y_i^2 & \sum_{i=1}^N y_i \\
\sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i y_i & \sum_{i=1}^N y_i^2 & \sum_{i=1}^N x_i & \sum_{i=1}^N y_i & \sum_{i=1}^N 1
\end{bmatrix}}_{A^T A}
\underbrace{\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}}_{\mathbf{c}}
=
\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{A\mathbf{0}}
$$

To solve for **c**, we need to find the null space of $A^T A$ in the same manner as in Problem 1. part (ii).

(b) In general, what is the minimum value of N that allows a unique solution for **c**?

Since there are five unknowns in **c** — a, b, c, d, e, f — we need at least five equations to solve for **c**. Therefore, the minimum value of N is 5.

(c) Homogenize equation 2.1 by making the substitutions $x \leftarrow x_1/x_3$ and $y \leftarrow y_1/y_3$, and show that in terms of homogeneous coordinates ($\mathbf{x} = (x_1, x_2, x_3)$) the conic can be expressed in matrix form

$$\mathbf{x}^T \mathbf{C}\mathbf{x} = 0,$$

where $\mathbf{C}$ is a symmetric matrix.

By substituting $x \leftarrow x_1/x_3$ and $y \leftarrow y_1/y_3$, and simplifying the resulting equation, we get:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

$$a\left(\frac{x_1}{x_3}\right)^2 + b\frac{x_1 y_1}{x_3 y_3} + c\left(\frac{y_1}{y_3}\right)^2 + d\frac{x_1}{x_3} + e\frac{y_1}{y_3} + f = 0$$

$$\frac{x_1^2}{x_3^2}a + \frac{x_1 y_1}{x_3 y_3}b + \frac{y_1^2}{y_3^2}c + \frac{x_1}{x_3}d + \frac{y_1}{y_3}e + f = 0$$

$$\frac{x_1^2}{x_3^2}(a) + \frac{x_1 y_1}{x_3 y_3}(b/2 + b/2) + \frac{y_1^2}{y_3^2}(c) + \frac{x_1}{x_3}(d/2 + d/2) + \frac{y_1}{y_3}(e/2 + e/2) + f = 0$$

$$\begin{bmatrix} (a)\frac{x_1}{x_3} & (b/2)\frac{x_1}{x_3} & (d/2)\frac{x_1}{x_3} \\ (b/2)\frac{y_1}{y_3} & (c)\frac{y_1}{y_3} & (e/2)\frac{y_1}{y_3} \\ d/2 & e/2 & f \end{bmatrix} \begin{bmatrix} \frac{x_1}{x_3} \\ \frac{y_1}{y_3} \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} \frac{x_1}{x_3} & \frac{y_1}{y_3} & 1 \end{bmatrix} \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{bmatrix} \frac{x_1}{x_3} \\ \frac{y_1}{y_3} \\ 1 \end{bmatrix} = 0$$

$$\underbrace{\begin{bmatrix} x & y & 1 \end{bmatrix}}_{\mathbf{x}^T} \underbrace{\begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{x}} = 0$$

(d) Suppose we apply a projective transformation to our points $\mathbf{x}_i^{'} = \mathbf{H}\mathbf{x}_i$. The transformed points x_i' will lie on a transformed conic represented by a new symmetric matrix $\mathbf{C}'$. Write an equation that specifies the relationship between $\mathbf{C}'$ and $\mathbf{C}$ in terms of the homography $\mathbf{H}$.

$$\begin{aligned} \mathbf{x}^T\mathbf{C}\mathbf{x} &= 0 \\ (\mathbf{H}\mathbf{x})^T\mathbf{C}(\mathbf{H}\mathbf{x}) &= 0 \\ \mathbf{x}^T\underbrace{\mathbf{H}^T\mathbf{C}\mathbf{H}}_{\mathbf{C}'}\mathbf{x} &= 0 \qquad \text{(since } (Ab)^T = b^TA^T\text{)} \end{aligned}$$

The transformed conic satisfies the equation $\mathbf{x}^T\mathbf{C}'\mathbf{x} = 0$ where

$$\mathbf{C}' = \mathbf{H}^T\mathbf{C}\mathbf{H}$$

Problem 3.

The affine transform in heterogeneous coordinates is given by

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.1}$$

Affine transformations are combinations of:

(i) Arbitrary linear transformations with 4 degrees of freedom (a, b, d, e).

(ii) Translations with 2 degrees of freedom (c, f).

Does affine transformation apply translation first followed by arbitrary linear transformation, or the other way around? Prove your answer mathematically.

Hint: check if

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a & b & 0 \\ d & e & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ d & e & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}$$

By computing the matrix products, we see that:

$$\begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a & b & 0 \\ d & e & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} a & b & 0 \\ d & e & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b & ac+bf \\ d & e & dc+ef \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Our desired affine transformation matrix (3.1) matches the first matrix product (3.2). Since matrices multiply from right to left (i.e. $\mathbf{ABc} = \mathbf{A}(\mathbf{Bc})$), the linear transformations (rotation, shear, scaling, etc.) are applied first before the translation. Intuitively, this makes sense because translating first would result in the image points being at the wrong coordinates during transformation.