CSE 152A: Week 5 Discussion

Agenda

- Questions on HW2?

- More Linear Algebra Concepts
  - Linear Systems of Equations
  - Matrix Rank
  - Null Space
  - SVD

- Go over the solution for HW1 Q4

Linear System of Equations

$$x + 3y = 6$$
$$2x + 12y = 12$$

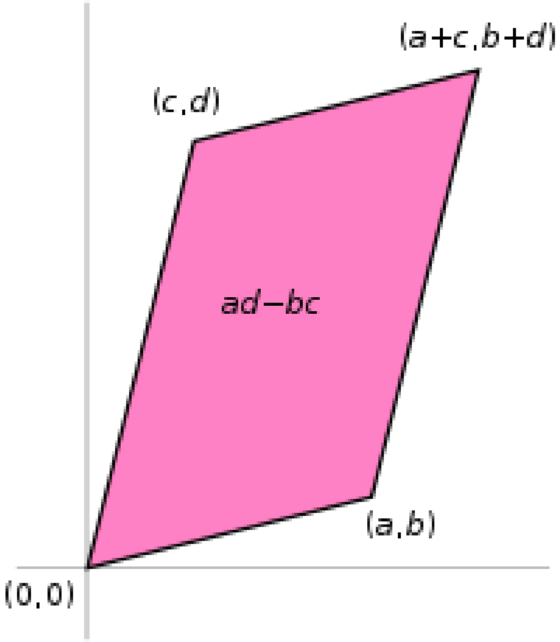
$$\begin{pmatrix} 1 & 3 \\ 2 & 12 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 6 \\ 12 \end{pmatrix}$$

$$Ax = b$$

- In other words: find the vector *x* such that after being transformed by A, lands on the vector *b*.
- Can solve using Gaussian elimination, finding the inverse of A, etc

Determinant

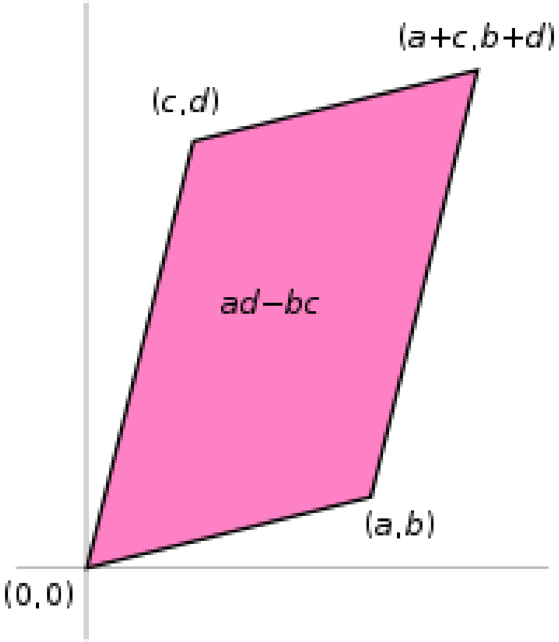
- Scalar value that is a function of the entries of a square matrix
  - Can think of it as the scaling factor and orientation induced by the mapping represented by the matrix



https://en.wikipedia.org/wiki/Determinant

Determinant

- If the determinant is non-zero (square matrix), then there would be a unique solution to *Ax=b*
  - i.e., A is invertible



https://en.wikipedia.org/wiki/Determinant

Determinant

- If the determinant is zero (square matrix), then there would not be a unique solution (if a solution at all) to $Ax=b$
  - Area/volume is collapsing after applying matrix A
  - In 2D, this means that our plane is collapsing onto a line or even a point.

- When the determinant is zero, there would be infinitely many vectors x that land on the vector b after being transformed by A (if a solution exists).

Rank

- The rank of a matrix is the dimension of the vector space spanned by its columns
  - This is identical to the dimension of the vector space spanned by its rows

- For an $m \times m$ matrix, we say that it is full rank if it has rank m.
  - An inverse can be found

- If it has rank $< m$, we say that the matrix is singular
  - Inverse does not exist

- Inverse does not exist for non-square matrices

Null Space

$$Ax = 0$$

- The null space of the matrix A consists of the vectors x such that $Ax = 0$
  - The set of vectors that after being transformed by A, land on the zero vector

- For square matrices, if the matrix is full rank, then the only solution is the trivial solution

Singular Value Decomposition (SVD)

$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times p}\mathbf{\Sigma}_{p\times p}\mathbf{V}^T_{p\times n}$$
$$= \begin{bmatrix} \mathbf{u}_0 & \cdots & \mathbf{u}_{p-1} \end{bmatrix} \begin{bmatrix} \sigma_0 & & \\ & \ddots & \\ & & \sigma_{p-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_0^T \\ \cdots \\ \mathbf{v}_{p-1}^T \end{bmatrix},$$

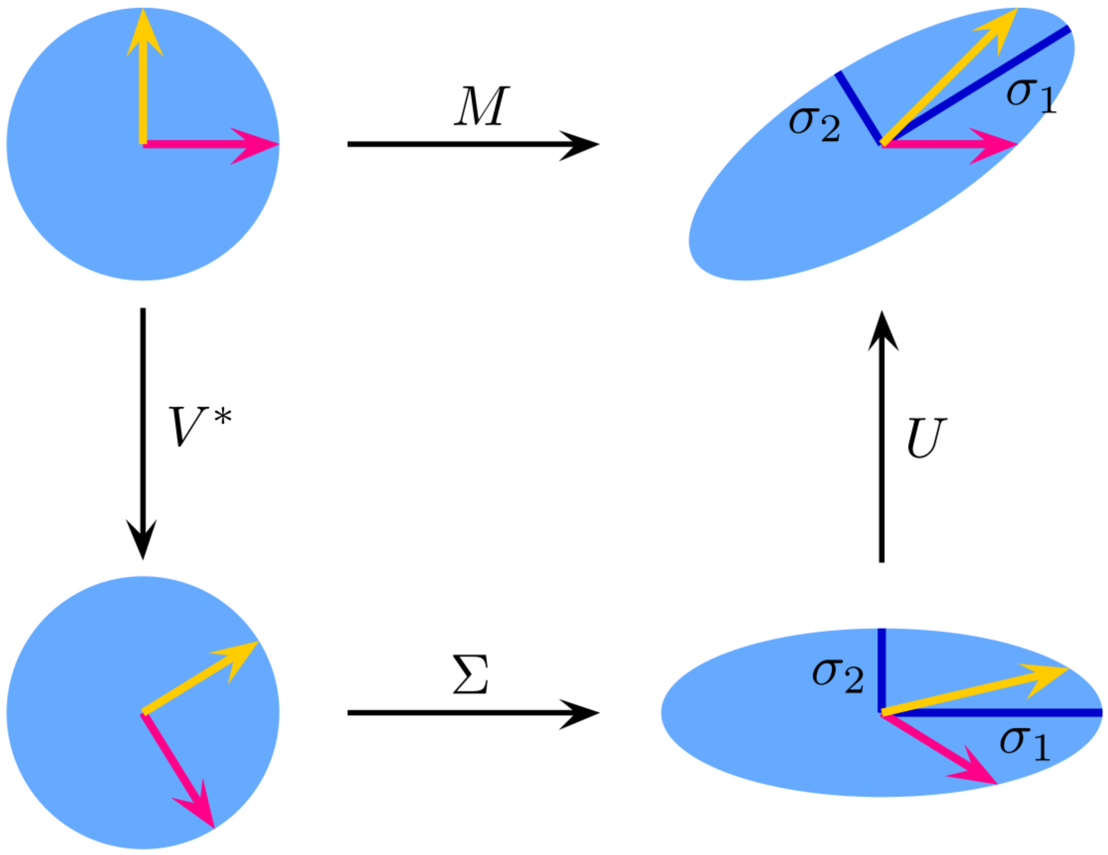
where $p = \min(m, n)$. The matrices $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and so are their column vectors,

$$\mathbf{u}_i \cdot \mathbf{u}_j = \mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}. \tag{A.2}$$

The singular values are all non-negative and can be ordered in decreasing order

$$\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_{p-1} \geq 0. \tag{A.3}$$

Singular Value Decomposition (SVD)



$$M = U \cdot \Sigma \cdot V^*$$

Singular Value Decomposition (SVD)

- HW2 Q2.3 (Essential Matrix)
  - We can use SVD to check the rank of the matrix

  - The rank of the matrix is equal to the number of non-zero singular values

8-point Algorithm

Estimating F: 8-point algorithm

- The fundamental matrix F is defined by

$$\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0$$

  for any pair of matches x and x' in two images.

- Let $\mathbf{x} = (u,v,1)^{\top}$ and $\mathbf{x}' = (u',v',1)^{\top}$, $\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

- Each match gives a linear equation:

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

8-point Algorithm

8-point algorithm

Given n point correspondences, set up a system of equations:

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

- In reality, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$.

8-point Algorithm

Solving homogeneous systems

- In reality, instead of solving $\mathbf{Af} = 0$, we seek **f** to minimize $\|\mathbf{Af}\|$.
- Singular value decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

$$\mathbf{U}, \mathbf{V} \text{ are rotation matrices}$$

$$\mathbf{\Sigma} = \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{bmatrix}$$

- Solution **f** given by the last column of **V**.

8-point Algorithm

- Make sure you read your documentation when using any SVD algorithm

- We want the last column of V. Numpy returns V^T, so you should obtain the last row of V^T.

- Then, you would need to reshape it to a 3×3 matrix.

HW2

- Read the hints!
- As always, check the shapes of the matrices and submatrices as you go

1.1: Edge Detection

- Use the function we’ve provided for you to create the kernel

- Can use convolve2d for convolution

1.2: Corner Detection

- Use numpy matrix indexing to your advantage

- We are finding “corner-like” features
  - Your features may not always be on the corner as you would define it in real life

- Possibly consider ignoring the edge values (in gradient calculation, second moment matrix, etc)
  - These regions are quite noisy

2.1: Epipolar Geometry

- Where is the point in 3D?
    - The intersection of the rays

- There would be a couple ways to solve this

- You would likely want to convert everything to world coordinates

2.2: The Epipolar Constraint

- You should convert the points into homogeneous coordinates and use the fundamental matrix

- Algebraically prove F_33 = 0.

2.3: Essential Matrix

- You can use Numpy SVD to check the singular values in the Σ matrix
  - The rank is equal to the number of non-zero singular values

3: SSD, NCC, Naïve Matching

- The locations would be in x-y coordinates
  - x is indexed by the columns of the matrix
  - y is indexed by the rows of the matrix

- Use the equation given in the assignment for the matching scores

- Matches may not be 100% perfect

4: Epipolar Geometry

- Again, you can use Numpy's SVD to find F
- Reshape F into a 3x3 matrix