Lecture 4 → Parameter Estimation

① Eigenvector of square $(m \times m)$ matrix S: A vector $v \in R^m \neq 0$ is called eigenvector of S if there exists a scaler value $\lambda \in R$ (eigenvalue) such that:

$$\boxed{Sv = \lambda v}$$

right (eigenvector) — eigenvalue

★ To find the eigenvalues λ, we solve:

$Sv = \lambda v \Rightarrow (S - \lambda I)v = 0$, solving this we get:

(a) m^{th} order equation in λ
(b) Can have atmost "m" distinct solutions
(c) λ can be complex, even if S is real.

★ But it only has non-zero solution if $|S - \lambda I| = det(S - \lambda I) = 0$. why?

(a) If we have $det(S - \lambda I) = 0$, then $S - \lambda I$ would have full rank and could be inverted.
(c) The only solution would be a trivial one i.e. $V = (S - \lambda I)^{-1} 0 = 0$

② Singular Value decomposition.

(1) Is related to eigenvalue decomposition but also works for matrices that are not square.
(2) For every $(m \times n)$ matrix "A", there exists a decomposition (SVD) of the form

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n} \qquad m \geq n \qquad \sigma_1 \geq \sigma_2 \geq \sigma_3 \cdots \sigma_n \geq 0$$

$$U^T U = I$$
$$V^T V = I$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

orthogonal to each other.

U → orthogonal matrix of size $(m \times m)$ (u_i are called "left singular vectors")

V → orthogonal matrix of size $(n \times n)$ (v_i are called "right singular matrix")

Σ → diagonal matrix of size $(m \times n)$ with non-negative real numbers on the diagonal.

Eg. $A = U \Sigma V^T$ → ① Rotation, ② scaling, ③ Rotation

★ SVD is not unique. But if "A" is real, U, V can be chosen to be real.

★ The singular values σ_i are always real, non-negative and they are conventionally arranged in decreasing order on the main diagonal of Σ.

⑤ Application of SVD:

Conditioning → It describes the behaviour of a system $Ax=b$
→ It is important for numerical optimization techniques.
→ Has nothing to do with numerical errors

(A) Well-conditioned (stable) → small changes in A or b result in small changes to the solution x.

(B) Ill-conditioned (Instable) → small changes in A or b result in large changes in solution x.

* SVD gives a way to describe the conditioning of a matrix A.
* The condition number of a matrix A describes the degree of singularity

$$\boxed{\text{Cond}(A) = \frac{\sigma_1}{\sigma_n}}$$

i.e. ratio of largest and smallest singular value of a matrix.

* large conditioning no → Ill-conditioned
  small " " → well conditioned.

⑥ Direct-Linear Algorithm → General Algorithm.

objective → Given a sufficient no. of "m" of measurements ($y_i' \leftrightarrow y_i$), determine X such that $y_i' = X y_i$

Algorithm → ① set up a linear system of equations (n = no. of parameters)

$$\begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{m-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix}$$

* we have more equations than no. of parameters.

② If b is zero → solve with SVD
else solve with Pseudo inverse
③ Determine X from x.

⑦ Direct Linear Transform – Homogeneous solution

homogeneous case: $\boxed{b_i = 0}$ i.e. Overdetermined system (more measurements than needed)

$$\begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{m-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = 0$$

subject to $\|x_2\|^2 = 1$
↳ Additional constraint to avoid a trivial solution.

* Exact solution can be computed as the null space of A. ($m=n$)
  → usually doesn't exist.
* why? → due to measurement noise.
* How does this show? → "A" has full rank
* How do we solve this? → find the approximate solution of the over-determined problem.
* How do degenerate measurement show? → if rank(A) < n

⑧ To solve this, we introduce the "Lagrange multiplier".

→ It is a powerful strategy to minimize (or maximize) a function which is subjected to an additionally equality constraint.

→ As an example, we minimize a function "$f(x,y)$" subject to $\underbrace{g(x,y)=c}_{\text{additional constraint}}$

↳ The goal is to minimize an auxilliary function that punishes any deviation from the constraint

$$\underbrace{\Lambda(x,y,\lambda)}_{\text{Lagrange function}} = f(x,y) + \underbrace{\lambda}_{\text{Lagrange multiplier}} \cdot \underbrace{(g(x,y)-c)}_{\text{Constraint.}}$$

→ Compute the gradient and set to zero.

$$\nabla_{x,y,\lambda} \Lambda(x,y,\lambda) = 0$$

⑨ Excursus: matrix-Vector differentiation

→ Let x, b → Vectors
A → matrix
$f(x)$ → Vector function

→ differentiating $f(x)$ with respect to vector x:

$$\frac{d}{dx} f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

→ for Vector products: $\frac{d}{dx}(b^T x) = \frac{d}{dx}(x^T b) = b$

$$\frac{d}{dx}(x^T x) = 2x$$

→ for matrix-vector products $\frac{d}{dx}(x^T A x) = (A^T + A)x$

→ if A is symmetric, this further simplifies to: $\frac{d}{dx}(x^T A x) = 2Ax$

⑩ Our goal is to (1) minimize $\arg\min \|Ax\|_2^2$ under the condition $\|x\|_2^2 = 1$

→ using Lagrange multiplier, we minimize instead:

$$\Lambda(x,\lambda) = \|Ax\|_2^2 + \lambda(1 - \|x\|_2^2) = (Ax)^T Ax + \lambda(1 - x^T x)$$

→ computing the gradient with respect to x and setting to 0:

$$\nabla_x \Lambda(x,\lambda) = 2A^T Ax - 2\lambda x = 0 \Rightarrow A^T A x = \lambda x$$

x is the eigenvector of $A^T A$

→ Looking now at the minimum

$$\min(\Lambda(x, \lambda)) = \min((Ax)^T Ax + \lambda(1 - x^T x))$$

$$= \min(\lambda x^T x + \lambda(1 - x^T x))$$

$$= \min(\lambda)$$

→ x is the eigenvector of $A^T A$ corresponding to the smallest eigen value

(11) How do we get the eigenvector of $A^T A$ corresponding to the smallest eigenvalue?

→ Recalling the SVD: $A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n}$ $\{m \geq n\}$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$

$U^T U = I$
$V^T V = I$

$$A^T A = V \Sigma^T U^T U \Sigma V^T = V \Sigma^2 V^T$$

* Reformulating the last equation to: $A^T A V = \Sigma^2 V$
* Eigenvalues $\lambda_1, \ldots, \lambda_n$ of $A^T A$ are the eigenvalues of $A^T A$ since it is a symmetric matrix.

* and $A^T A v_i = \lambda_i v_i$ → v_i = eigenvector of $A^T A$

$$\boxed{\sigma_i = \sqrt{\lambda_i}}$$

→ Connection b/w eigenvalues and singular values.

* last column "V" is the minimiser of the problem. (solution of our problem)

(12) In inhomogeneous case: $b_i \neq 0$
Eg → inhomogeneous solution to homography estimation

→ Solve $Ax = b$ } Exact solution usually does not exist

→ instead solving

$$\arg\min \|Ax - b\|_2^2 = \arg\min (Ax - b)^T (Ax - b)$$

→ computing the gradient and setting it to 0.

$$2A^T A x - 2A^T b = 0$$

$$= x = \underbrace{(A^T A)^{-1} A^T}_{} b$$

Called Pseudo-inverse matlab operator $x = A \backslash b$

(13) Direct Linear Transform: Advantages. Disadvantages

Advantages:
(1) No initialization required
(2) 1 iteration

Disadvantages:
(1) Sensitive to imprecise measurements / outliers
(2) usually no minimal parametrization
(3) constraints enforced afterwards

(13) To cope up with the disadvantages of direct linear Transform, we do a "Iterative non-linear Estimation".

$$Ax = b \Rightarrow f(x, a) = b$$

Properties: (1) often slower than DLT
(2) Requires initialization → typically use linear solution or sample parameter space.
(3) No guarenteed convergence, local minima
(4) stopping criteria required.

optimization algorithms such as: (1) Newton's method
(2) Levenberg - Marquardt

(14) INE - statistical cost functions and maximum likelihood Estimation.

(1) minimises an optimal cost function related to noise level.
(2) Assumes 0 mean - isotropic noise (Gaussian) (outliers are assumed to be removed)

$$Pr(x) = \frac{1}{2\pi\sigma^2} e^{-d(x, \bar{x})^2 / 2\sigma^2}$$

x → measured coordinates
$\hat{x}$ → estimated coordinates
$\bar{x}$ → true coordinates

Case 1: Calibration pattern → Given a Homography "H", the probability distribution for data x_i' perturbed by noise is given by:

$$\boxed{Pr(\{x_i'\} \mid H) = \prod_i \frac{1}{2\pi\sigma^2} e^{-d(x_i', H\bar{x}_i)^2/(2\sigma^2)}}$$

Probability of measuring x_i' given the true homography "H".

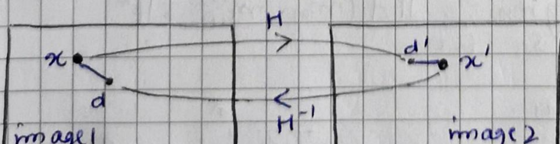
→ The maximum likelihood estimate (MLE) of the homography "$\hat{H}$" maximizes the probability to get the measured data:

Taking the log.

$$\log Pr(\{x_i'\} \mid H) = -\frac{1}{2\sigma^2} \sum d(x_i', H\bar{x}_i)^2 + \text{constant}$$

→ therefore, we solve: $\hat{H} = \underset{H}{\arg\min} \sum_i d(x_i', H\bar{x}_i)^2$

Case 2:



x → measured coordinates
$\hat{x}$ → Estimated coordinates
$\bar{x}$ → true coordinates.

* Assume zero-mean isotropic gaussian noise $P_r(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{d(x, \bar{x})^2}{2\sigma^2}}$

* Error in both the images (from one image to another image)

$$Pr(\{x_i'\} \mid H) = \prod_i \frac{1}{2\pi\sigma^2} e^{-(d(x_i, \bar{x}_i)^2 + d(x_i', H\bar{x}_i)^2)/2\sigma^2}$$

* Then we solve the symmetric minimization problem

$$\hat{H} = \operatorname{argmin} \sum_i d(x_i, H^{-1}x_i')^2 + d(x_i', Hx_i)^2$$

we seek the corrected image measurements that play the role of the true measurements.

Case 3: Geometric Distance

→ optimization of H and measurements
→ Reprojection error

x → measured coordinates
$\hat{x}$ → estimated "
$\bar{x}$ → true coordinates
$d(\cdot,\cdot)$ → Euclidian distance in image.

$$(\hat{H}, \hat{x}_i, \hat{x}_i') = \operatorname*{argmin}_{H, \hat{x}_i, \hat{x}_i'} \sum_i d(x_i, \hat{x}_i)^2 + d(x_i', \hat{x}_i')^2$$

subject to $\hat{x}_i' = \hat{H}\hat{x}_i$

(15) Iterative minimization

→ General non-linear problem formulation for "m" measurements:

$$\hat{x} = \operatorname*{arg\,min}_{x} \sum_{i=1}^{n} r^{(i)}$$

$\hat{x}$ → estimate; x → Parameter Vector; $r^{(i)}$ → Residual / Reprojection error

→ Which cost function for computing the residuals?
→ How to represent the parameter vector?

* Back to the concrete problem of camera pose estimation from 2D/3D correspondences:

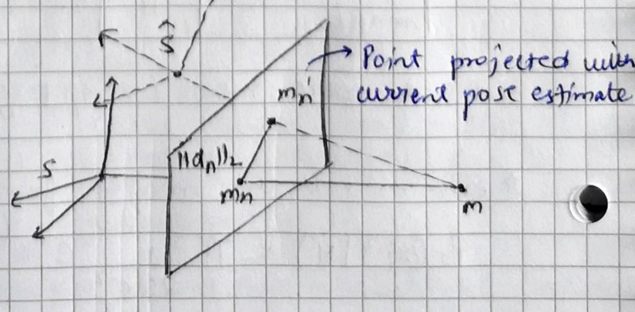
→ Parameter's to estimate: camera pose (rotation and translation)?

(16) Least square Estimation.

→ Typical Problem formulation (error in normalised image space):

$$\hat{s} = \operatorname*{argmin}_{s} \sum_{i=1}^{n} r_n^{(i)LS} = \operatorname*{arg\,min}_{s} \sum_{i=1}^{m} \| d(m_n^{(i)}, m_w^{(i)}, s) \|^2$$

s → extrinsic camera parameters with 6 DOF

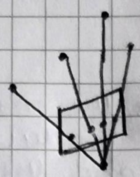


* All measurements are computed with equal weight.

how good is my point.

* If information about the quality of measurements is given, how to incorporate this?

→ Weighted least square Estimation (WLS)
→ Incorporation of simple stochastic model: parameters and measurements modelled as gaussian random variables.



$$\hat{s} = \arg\min_{s} \sum_{i=1}^{n} r_n^{(i)\,WLS} = \arg\min_{s} \sum_{i=1}^{n} d_n(m_n^{(i)}, m_w^{(i)}, s)^T \underbrace{P_n^{-1}(m_n^{(i)}; m_w^{(i)})}_{\text{joint covariance of 2D/3D correspondences.}} d_n(m_n^{(i)}, m_w^{(i)}, s)$$

allows • (i) incorporating measurement uncertainties
(ii) Representation / Estimation of errors.

But (i) still not robust in case of wrong correspondences. (outliers)

→ Robust Estimation (m-estimators)

(i) minimize the sum of a function of residuals:

$$\hat{s} = \arg\min_{s} \sum_{i=1}^{n} \rho(r^{(i)})$$ → function appears as weight.

(ii) Assumptions (i) only few outliers
(ii) Initialization close to solution

(iii) Example (H-matrix estimation): still not robust in case of wrong correspondences

⇒ Summary:

- Robust Estimation: RANSAC