Computer Vision

Dr. Syed Faisal Bukhari

Associate Professor

Department of Data Science

Faculty of Computing and Information Technology

University of the Punjab

Reference

These notes are based on

- Dr. Matthew N. Dailey's course: AT70.20: Machine Vision for Robotics and HCI

Derivation of a homography

$\vec{x}'_{3\times1} = H_{3\times3}\ \vec{x}_{i\ 3\times1}$ or $x'_i \propto Hx_i$

This is a homogenous equation. $x'_i \propto Hx_i$ means that

$$x'_i = k_i H x_i$$

This scalar k_i can be different for every point in order to eliminate the 3rd component

$$Hx_i = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$k = \frac{1}{w}$$

The scaling (or normalization) factor used to make the third component equal to 1

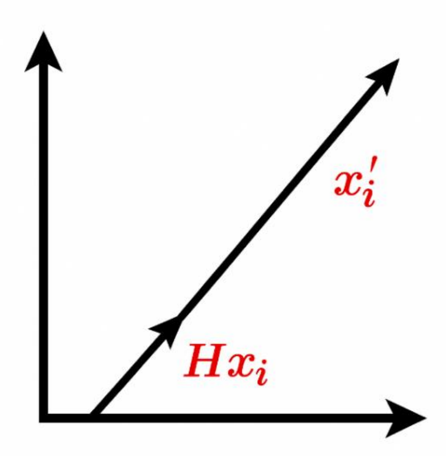
$$k = \frac{1}{w}$$

Obviously, k will be different for different points

$$x'_i = k_i H x_i$$

Geometric Meaning of Scaled Vectors in Homography

- If $x_i' = k_i H x_i$, then x_i' is a scaled version of Hx_i— they lie in the same direction.



- In vector algebra, if one vector is a **scalar multiple of another**, they are directionally aligned. This implies that the vectors x_i' and Hx_i are **collinear.**

$$\vec{x}'_{3\times1} \times H_{3\times3}\ \vec{x}_{i\ 3\times1} = 0$$

$$\vec{x}'_i \times H\ \vec{x}_i = 0$$

$$H\ \vec{x}_i = \begin{bmatrix} \vec{\mathrm{h}}^{1\mathrm{T}}\vec{\mathrm{x}}_{\mathrm{i}} \\ \vec{\mathrm{h}}^{2\mathrm{T}}\vec{\mathrm{x}}_{\mathrm{i}} \\ \vec{\mathrm{h}}^{3\mathrm{T}}\ \vec{\mathrm{x}}_{\mathrm{i}} \end{bmatrix}_{3\times1} \text{-----------------------------------(1)}$$

Let

$\vec{h}^{1T}$ = 1st row of H

$\vec{h}^{2T}$ = 2st row of H

$\vec{h}^{3T}$ = 3rd row of H

Equation (1) ⇒

A row of H as we represent it as a **column vector**. We transpose this vector multiplied by the vector x_i

Suppose

$$\vec{x}_i' = \begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} \text{-------------------------(2)}$$

Taking cross product of (1) and (2), we get

$$\begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} \times \begin{bmatrix} \vec{h}^{1T} \vec{x}_i \\ \vec{h}^{2T} \ \vec{x}_i \\ \vec{h}^{3T} \ \vec{x}_i \end{bmatrix}$$

$$= \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ x_i' & y_i' & w_i' \\ \vec{h}^{1T} \vec{x}_i & \vec{h}^{2T} \vec{x}_i & \vec{h}^{3T} \vec{x}_i \end{vmatrix}$$

Rule of Sarrus

$$\begin{array}{ccccc} \hat{\imath} & \hat{\jmath} & \hat{k} & \hat{\imath} & \hat{\jmath} \\ x_i' & y_i' & w_i' & x_i' & y_i' \\ \vec{h}^{1T}\vec{x}_i & \vec{h}^{2T}\vec{x}_i & \vec{h}^{3T}\vec{x}_i & \vec{h}^{1T}\vec{x}_i & \vec{h}^{2T}\vec{x}_i \end{array}$$

$$= \hat{\imath}\, \vec{h}^{3T}\vec{x}_i y_i' + \hat{\jmath}\vec{h}^{1T}\vec{x}_i w_i' + \hat{k}\vec{h}^{2T}\vec{x}_i - \hat{k}\vec{h}^{1T}\vec{x}_i y_i' - \hat{\imath}\, \vec{h}^{2T}\vec{x}_i w_i' - \hat{\jmath}\vec{h}^{3T}\vec{x}_i x_i'$$

$$= (\vec{h}^{3T}\vec{x}_i y_i' - \vec{h}^{2T}\vec{x}_i w_i')\, \hat{\imath} + (\vec{h}^{1T}\vec{x}_i w_i' - \vec{h}^{3T}\vec{x}_i x_i')\, \hat{\jmath} + (\vec{h}^{2T}\vec{x}_i x_i' - \vec{h}^{1T}\vec{x}_i y_i')\hat{k}$$

$$\begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} \times \begin{bmatrix} \vec{h}^{1T}\vec{x}_i \\ \vec{h}^{2T}\vec{x}_i \\ \vec{h}^{3T}\vec{x}_i \end{bmatrix} = \begin{bmatrix} y_i'\vec{h}^{3T}\vec{x}_i - w_i'\vec{h}^{2T}\vec{x}_i \\ w_i'\vec{h}^{1T}\vec{x}_i - x_i'\vec{h}^{3T}\vec{x}_i \\ x_i'\vec{h}^{2T}\vec{x}_i - y_i'\vec{h}^{1T}\vec{x}_i \end{bmatrix} \text{--------------------} (3)$$

- x_i', y_i' *and* w_i' are the **scalar elements** of $\vec{x}_i'$

$$\vec{x}_i' \times H\,\vec{x}_i = \begin{bmatrix} y_i'\vec{h}^{3T}\vec{x}_i - w_i'\vec{h}^{2T}\vec{x}_i \\ w_i'\vec{h}^{1T}\vec{x}_i - x_i'\vec{h}^{3T}\vec{x}_i \\ x_i'\vec{h}^{2T}\vec{x}_i - y_i'\,\vec{h}^{1T}\vec{x}_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3\times 1} \quad \text{------------(3)}$$

- These three equals to zero. Three linear equations in 9 unknowns ($\vec{h}^{1T}, \vec{h}^{2T}, \vec{h}^{3T}$).

- Each row gives one linear equation in 9 unknowns.

- It turns out **any two are linearly independent.**

- So, all **three equations** are not useful.

- We have **two independent equations** in **9** unknowns.

- How to turn out (3) into a design matrix.

$$\vec{x}_i = \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix}$$

$$y_i' \vec{h}^{\,3T} \vec{x}_i - w_i' \vec{h}^{2T} \vec{x}_i = 0 \quad \text{---------------------3a)}$$

Substitute values of $\vec{h}^{\,2T}$, $\vec{h}^{\,3T}$ and $\vec{x}_i$ in 3a) , we get

$\Rightarrow y_i' \,(h_{31}x_i + h_{32}y_i + h_{33}w_i)$ - $w_i' \,(h_{21}x_i + h_{22}y_i + h_{23}w_i)$ = 0

$\Rightarrow$ 0.h_{11} + 0.h_{12} + 0.h_{13} -$w_i'\, x_i h_{21}$ - $w_i'\, y_i\, h_{22}$ - $w_i' w_i h_{23}$ $+ y_i'\, x_i h_{31}$ + $y_i'\, y_i\, h_{32}$ + $y_i' w_i h_{33}$ = 0

$$\Rightarrow [0 \quad 0 \quad 0 \quad -w_i'\, x_i \quad -w_i'\, y_i \quad -w_i' w_i \quad y_i'\, x_i \quad y_i'\, y_i \quad y_i' w_i]_{1\times 9}$$

$$[h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad h_{33}]^T_{1\times 9} = [0]_{1\times 1}$$

$$[0 \quad 0 \quad 0 \quad -w_i'\, x_i \quad -w_i'\, y_i \quad -w_i' w_i \quad y_i'\, x_i \quad y_i'\, y_i \quad y_i' w_i]_{1\times 9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1}$$

$$= [0]_{1\times 1}$$

$$\Rightarrow \begin{bmatrix} \vec{0}^T & -w_i' \, \vec{x}_i^{\,T} & y_i' \, \vec{x}_i^{\,T} \end{bmatrix}_{1\times 9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = [0]_{1\times 1} \quad \text{-- --------(4)}$$

Similarly, we do the same with the second row of (3)

$$w_i' \vec{h}^{1T} \vec{x}_i - x_i' \vec{h}^{3T} \vec{x}_i = 0 \quad \text{----------------------------------4 a)}$$

Substitute values of $\vec{h}^{\,1T}$, $\vec{h}^{\,3T}$ and $\vec{x}_i$ in 4 a), we get

$$\Rightarrow w_i' \, (h_{11}x_i + h_{12}y_i + h_{13}w_i) - x_i' \, (h_{31}x_i + h_{32}y_i + h_{33}w_i) = 0$$

$$\Rightarrow w_i' \; x_i h_{11} + w_i' \, y_i \, h_{12} + w_i' \, w_i . h_{13} + 0h_{21} + 0h_{22} + 0h_{23} - x_i' \, x_i h_{31} - x_i' \, y_i \, h_{32} - x_i' w_i h_{33} = 0$$

$$\Rightarrow \begin{bmatrix} w_i' \, x_i & w_i' \, y_i & w_i' \, w_i & 0 & 0 & 0 & -x_i' \, x_i & -x_i' \, y_i & -x_i' w_i \end{bmatrix}_{1\times 9} \times \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}_{1\times 9}^T = [0]_{1\times 1}$$

$$\Rightarrow \; [w_i' \, x_i \quad w_i' \, y_i \quad w_i' \, w_i \quad 0 \quad 0 \quad 0 \quad -x_i' \, x_i \quad -x_i' \, y_i \quad -x_i' w_i]_{1\times 9}$$

$$\begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = [0]_{1\times 1}$$

$$\Rightarrow \left[w_i' \, \vec{x}_i{}^T \quad \vec{0}^T \quad -x_i' \, \vec{x}_i{}^T\right]_{1\times 9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = [0]_{1\times 1} \text{-----------------} (5)$$

Similarly, using the third row of (3)

$x_i' \vec{h}^{2T} \vec{x}_i - y_i' \, \vec{h}^{1T} \vec{x}_i$ = 0 ----------------------------5a)

Substitute values of $\vec{h}^{\,2T}$, $\vec{h}^{\,1T}$ and $\vec{x}_i$ in 5a), we get

$\Rightarrow x_i'(h_{21}x_i + h_{22}y_i + h_{23}w_i)$ - $y_i'\,(h_{11}x_i + h_{12}y_i + h_{13}w_i)$ = 0

$- y_i'\, x_i h_{11} \; - \; y_i'\, y_i\, h_{12} \; - \; x y_i'\, w_i h_{13} + \; x_i' x_i h_{21} + x_i' y_i\, h_{22} +$
$x_i' w_i . h_{23} \; + 0h_{31} + 0h_{32} + 0h_{33} = 0$

$$\Rightarrow [- y_i'\, x_i \quad - y_i'\, y_i \quad - y_i'\, w_i \quad x_i'\, x_i \quad x_i'\, y_i \quad x_i'\, w_i \quad 0 \quad 0 \quad 0]_{1\times 9}$$

$$\times [h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad h_{33}]^T_{1\times 9} = [0]_{1\times 1}$$

$$\Rightarrow [-\,y_i'\,x_i \quad -\,y_i'\,y_i \quad -\,y_i'\,w_i \quad x_i'\,x_i \quad x_i'\,y_i \quad x_i'\,w_i \quad 0 \quad 0 \quad 0]_{1\times 9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = [0]_{1\times 1}$$

$$\Rightarrow \left[-y_i'\,\vec{x}_i{}^T \quad x_i'\,\vec{x}_i{}^T \quad \vec{0}^T\right]_{1\times 9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = [0]_{1\times 1} \text{----------------(6)}$$

Stacking up (4), (5), and (6), we get

$$\begin{bmatrix} \vec{0}^T & -w_i'\,\vec{x}_i{}^T & y_i'\,\vec{x}_i{}^T \\ w_i'\,\vec{x}_i{}^T & \vec{0}^T & -x_i'\,\vec{x}_i{}^T \\ -y_i'\,\vec{x}_i{}^T & x_i'\,\vec{x}_i{}^T & \vec{0}^T \end{bmatrix}_{\color{red}{\mathbf{3\times 9}}} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3\times 1} \text{----(7) or } \color{red}{\mathbf{(4.1)}}$$

$$\vec{h} = \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}, \mathrm{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$ ---------------7a) or **(4.2)**

Although there are three equations in (7), **only two of them are linearly independent** (since the third row is obtained, up to scale, from the sum of x_i' times the first row and y_i' times the second).

Thus, **each point correspondence** gives **two equations** in the entries of H. It is usual to **omit the third equation** in solving for H

$$\begin{bmatrix} \vec{0}^T & -w_i' \, \vec{x}_i^{\,T} & y_i' \, \vec{x}_i^{\,T} \\ w_i' \, \vec{x}_i^{\,T} & \vec{0}^T & -x_i' \, \vec{x}_i^{\,T} \end{bmatrix}_{\mathbf{2\times 9}} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{2\times 1} \text{---------(8)}$$

$\Rightarrow A\vec{h} = \vec{0}$

This is a generic point. We can repeat it for any correspondence.

Suppose we have at least 4 points correspondences

$\vec{x}_1 \leftrightarrow \vec{x}_1'$

$\vec{x}_2 \leftrightarrow \vec{x}_2'$

$\vec{x}_3 \leftrightarrow \vec{x}_3'$

$\vec{x}_4 \leftrightarrow \vec{x}_4'$

Recall:

$$\vec{x}_i' \times H\,\vec{x}_i = \begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} \times \begin{bmatrix} \vec{h}^{1T}\vec{x}_i \\ \vec{h}^{2T}\,\vec{x}_i \\ \vec{h}^{3T}\,\vec{x}_i \end{bmatrix}$$

Where

$\vec{x}_i' = \begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix}$ and $H\,\vec{x}_i = \begin{bmatrix} \vec{h}^{1T}\vec{x}_i \\ \vec{h}^{2T}\,\vec{x}_i \\ \vec{h}^{3T}\,\vec{x}_i \end{bmatrix}$

We stack up four points correspondences

$$\begin{bmatrix} \vec{0}^T & -w_1' \vec{x}_1{}^T & y_1' \vec{x}_1{}^T \\ w_1' \vec{x}_1{}^T & \vec{0}^T & -x_1' \vec{x}_1{}^T \\ \vec{0}^T & -w_2' \vec{x}_2{}^T & y_2' \vec{x}_2{}^T \\ w_2' \vec{x}_2{}^T & \vec{0}^T & -x_2' \vec{x}_2{}^T \\ \vec{0}^T & -w_3' \vec{x}_3{}^T & y_3' \vec{x}_3{}^T \\ w_3' \vec{x}_3{}^T & \vec{0}^T & -x_3' \vec{x}_3{}^T \\ \vec{0}^T & -w_4' \vec{x}_4{}^T & y_4' \vec{x}_4{}^T \\ w_4' \vec{x}_4{}^T & \vec{0}^T & -x_4' \vec{x}_4{}^T \end{bmatrix}_{8\times 9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times 1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{8\times 1}$$

- If the **8×9 matrix** has **rank 8** then we have a right null space, which is the solution to this linear system.

DLT Algorithm for 2D Homography Estimation

Objective: Given n ≥ 4 point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that x_i'= Hx_i

Algorithm:

I. For each correspondence $x_i \leftrightarrow x_i'$, compute a 2×9 matrix A_i from 4.1 Only the first two rows need be used in general.

II. Assemble the ***n*** **2 × 9** matrices A_i into a single 2*n* × 9 matrix A.

III. Perform **SVD** on A (section A4.4(*p*585)). The **unit singular vector** corresponding to the **smallest singular value** is the **solution** **h**. Specifically, if A = UDV^T with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then **h** is the **last column of V.**

IV. The matrix H is determined from **h** as in (4.2).

Homography Estimation with Point Correspondences

- If an **8 × 9 matrix** has rank 8, then there exists a **right null space**, which provides the **solution to this linear system**.

- A practical approach to solving for homography uses 4 point correspondences, as we're not aiming for an exact solution.

- To obtain a more accurate estimate of the homography, we often **use more point correspondences, intentionally over-constraining the system**.

- When we have **more than 4 point correspondences**, **say n**, we apply the same method to these **n correspondences**, resulting in a **2n × 9 linear system,** expressed as:

$$A_{2n\times 9}\vec{h}_{9\times 1} = \vec{O}_{2n\times 1}$$

Approximating Homography Solution

$$A_{2n\times 9}\vec{h}_{\,9\times 1} = \vec{O}_{\,2n\times 1}$$

- The matrix A has 2n rows and 9 columns, so an exact solution does not exist.

- Rather than searching for an exact solution, we aim to find the **best possible approximation** based on certain measurements.

- Our goal is to find a **vector h** such that $A\vec{h}$, which is a **2n × 1** column vector, is as close to zero as possible, thereby **approximating a solution for h**.

Minimizing $\|\vec{h}\|$ or $\|A\vec{h}\|$

Instead of demanding an **exact solution**, we look for an approximate one, specifically, a vector $\vec{h}$ that **minimizes an appropriate cost function**.

This brings up a key question: **what exactly should be minimized?**

- To avoid the **trivial solution h = 0**, we need to impose an **additional constraint**.

- A common approach is to constrain the **norm of h**, typically setting $\|\vec{h}\| = 1$

- The specific value of the norm is not critical, since the homography **H is defined only up to scale**.

- Since there is no exact solution to **Ah = 0**, it makes sense to **minimize** $\|A\vec{h}\|$ instead, subject to the constraint that $\|\vec{h}\| = 1$

Minimizing $\left\|A\vec{h}\right\|$ in Homography Estimation

- **Idea:** We aim to minimize $\left\|A\vec{h}\right\|$ which represents the **Euclidean norm** of the vector $A\vec{h}$.
- This means minimizing the Euclidean length of Ah, resulting in a vector of **size 2n $\times$ 1**.
- For instance, with **10 point correspondences**, $\left\|A\vec{h}\right\|$ becomes a **20 $\times$ 1 dimensional vector**.
- We want **this vector** to be as **short as possible**, as it represents **the error vector**.
- Minimizing $\left\|A\vec{h}\right\|$ is the most straightforward cost function for estimating 2D homography.
- **However, there's a challenge:** this formulation **includes a trivial solution.**

Recall: Understanding Homogeneous Systems

- A **system of linear equations** is called **homogeneous** if it can be expressed as **Ax = 0**, where A is an m × n matrix and 0 is the zero vector in $\mathbb{R}^m$.

- Every **homogeneous system Ax = 0** has **at least one solution:** x = 0 (the zero vector in $\mathbb{R}^n$).

- This **zero vector** is known as the **trivial solution**.

- The **homogeneous system Ax = 0** has a **nontrivial solution** if and only if the system has **at least one free variable**.

Problem: $\|A\vec{h}\|$ can be trivially minimized by

$$\vec{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{9\times 1}$$

This vector represents the **zero homography**

$$\vec{x}_i' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

???

Is it true?

$$\vec{x}_i' = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} ??$$

Equal is not truly equal to. We mean proportional to.

$$\vec{x}_i' \propto \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

It is true. We have to **avoid this point**.

Trivial Minimization of $\|\mathrm{A}\vec{h}\|$ and Its Implications

- Minimizing $\|\mathrm{A}\vec{h}\|$ can be trivially achieved by choosing h = [0, 0, ..., 0]^T (a 9 × 1 zero vector).
- This corresponds to a **zero homography**, which is **not useful in practice**.
- Now consider x_i'= [0, 0, 0]^T. Is this valid?
- This expression is not about strict equality but proportionality: $x_i' \propto [0, 0, 0]^T$.
- Yes, it is mathematically true, but we must avoid this **degenerate case.**
- Applying a zero homography matrix to any point [x, y, w] results in an undefined transformation: [?, ?, ?]^T.
- Thus, **the trivial solution** must be excluded from consideration.

Avoiding the Trivial Solution: $\vec{\mathrm{h}} = \vec{0}$

- To ensure a meaningful homography, we must avoid the trivial solution $\vec{\mathrm{h}} = \vec{0}$ in projective geometry.

- Instead of performing **unconstrained minimization**, we apply **constrained minimization** by enforcing the condition $\|\vec{h}\| = 1$ (let).

- The **norm** $\|\vec{h}\|$ just needs to be. It doesn't have to be **exactly 1**, it can be **any nonzero** value**.**

- To **avoid the trivial case**, we normalize the point x_i' and solve for a homography that minimizes the **error vector** under the constraint $\|\vec{h}\| = 1$

- Thus, we **minimize** $\|A\vec{h}\|$ subject to the constraint that $\|\vec{h}\| = 1$**.**

Minimizing $\|A\mathbf{h}\|$ with Constraint $\|\mathbf{h}\| = 1$

- How can we find a vector $\mathbf{h}$ that minimizes $\|A\vec{h}\|$,subject to the constraint $\|\vec{h}\|$ = 1?

- Is there a simplified way to approach this?

  Yes: Use Singular Value Decomposition (SVD).

- **We aim to solve:**

$$\hat{\vec{h}} = \underset{\vec{h}}{\operatorname{argmin}} \|A\vec{h}\|, \text{ subject to the constraint } \|\vec{h}\| = 1$$

Or equivalently:

- "The optimal $\hat{\vec{h}}$ is the vector that minimizes $\|A\vec{h}\|$over all **unit vectors $\mathbf{h}$**."

Minimizing $\|A\mathbf{h}\|$ with Constraint $\|\mathbf{h}\| = 1$

Or

- "$\hat{\vec{h}}$ is equal to argmin over all possible $\|A\vec{h}\|$ subject the constraint $\|\vec{h}\| = 1$

Final Result:

- "The solution $\hat{\vec{h}}$ is the **unit eigenvector** of **A^TA** corresponding to the **smallest eigenvalue**."

Understanding the Optimization Problem

- **We want to solve:**

$\hat{\vec{h}}$ = $\underset{\vec{h}}{\operatorname{argmin}} \left\|A\vec{h}\right\|$, subject to the constraint $\left\|\vec{h}\right\| = 1$

Means:

$\hat{\vec{h}}$ = argmin $\left\|A\vec{h}\right\|$: **Find the vector $\hat{\vec{h}}$ that gives the smallest possible value of** $\left\|A\vec{h}\right\|$.

Subject to: A constraint that $\vec{h}$ must be a **unit vector**, i.e., $\left\|\vec{h}\right\| = 1$

$\hat{\vec{h}}$ = The resulting **optimal vector** that solves this minimization problem.

argmin = Argument of the Minimum

argmin vs argmax

- **argmin:** Find the input where the function is **smallest**
- **argmax:** Find the input where the function is **largest**
- We want to find vector $\hat{\vec{h}}$ that **minimizes** the norm $\|A\vec{h}\|$ subject to the constraint i.e., norm or length of $\vec{h}$ is 1 i.e., $\|\vec{h}\| = 1$.

Understanding $A^T A$ in optimization problem?

- It is a **covariance matrix** for A. Rows of A are being treated as points. If A is 2n×9, then A^T is 9×2n

$$\Rightarrow A^T{}_{9\times 2n} A_{2n\times 9} = (A^T A)_{9\times 9}$$

- The resulting matrix A^TA is 9 × 9. We need to find the **eigen vector** corresponding to the **smallest eigenvalue** of this resulting matrix. We have a pair:
  Pair 1: Eigenvector 1 and Eigenvalue 1
  Pair 2: Eigenvector 2 and Eigenvalue 2

- The rank of this matrix would be **9**. Because we assume we have more **than four correspondences** and we have some noise in our system. That's the measurement noise in our correspondences.

- So the rank of this matrix would be **full rank**.

What is $A^T A$?

- One of these **eigenvectors** that correspond to **smallest eigenvalue**. How to explain it intuitively. The idea is just that the **eigenvector** that is corresponding to the **smallest eigenvalue** is just in the direction along this **multi-dimensional space** that loses the **least amount of the information** about this covariance matrix $\mathbf{A}^T\mathbf{A}$.

- **Note:** A matrix is said to have **full rank** if its rank equals the largest possible for a matrix of the same dimensions, which is the lesser of the number of rows and columns. A matrix is said to be **rank deficient** if it does not have full rank.

Recall: Full rank matrix

- If our matrix is an $m \times n$ matrix with **$m < n$**, then it **has full rank** when its **m rows are linearly independent**.

- If **$m > n$**, the matrix has **full rank** when its **n columns are linearly independent**.

- If **$m = n$**, the matrix has **full rank** either when its **rows or its columns** are **linearly independent** (when the rows are linearly independent, so are its columns in this case).

- Try to think about it intuitively geometrically, when we take a **9×9 covariance matrix** and we are getting closer and closer to rank 8 that corresponds to this ellipsoid and crashing it **until it becomes rank 8**. It means you have **flattened** it. Instead of having a **3D ellipsoid** you just have a **ellipse**.

- What is the easiest way to calculate this eigenvector? SVD.

- Let USV^T be the **SVD** of A. $\hat{\vec{h}}$ is the column of V corresponding to the **smallest singular value** (diagonal element of S).

- SVD gives us three matrices when we take the product of these three matrices, we get back our original matrix

- **Note:** U and V are orthogonal matrices and V is a diagonal matrix.

Problem statement

Find a good homography H satisfying (approximately)

$\vec{x}'_{3\times1} = H_{3\times3}\ \vec{x}_{i\ 3\times1}$ or $x_i' \propto Hx_i$

The basic DLT for H

Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that x_i = H x_i'

Algorithm

(i) For each correspondence $x_i \leftrightarrow x_i'$ compute the matrix A_i from (7). Only the first two rows need be used in general. When w_i' = 0, use different rows. If you have ideal points in some cases then use different rows.

(ii) Assemble the ***n*** **2 × 9 matrices** A_i into a single 2n × 9 matrix A.

(iii) Obtain the **SVD of A** = UDV^T. The **unit singular vector** corresponding to the **smallest singular value** is the solution **h**. Specifically, if A = UDV^T with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then **h** is the last column of V.

(iv) Rearrange **h** to obtain H as in (7a).