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

Estimation – 2D Projective Transformations

- We spend a lot of time on statistics. We put **statistics** and **geometry** together, we get **computer vision**. We will start with a problem of how to **estimate homography automatically** even when there is **a noise in the image**.

- In computer science, we don't study statistical estimation as rigorously as in electrical engineering. Now we are starting chapter 4.

- In computer vision, we are frequently confronted with **estimation problems** in which **parameters of some functions** must be **estimated from measurements**.

Following are the fundamental problems in computer vision:

1. **2D homography:** Given a set of points x_i in $\mathbb{P}^2$ and corresponding points x_i' in $\mathbb{P}^2$ find a homography taking each x_i to x_i'.

2. **3D to 2D camera projection:** Given a set of points X_i in 3D space and corresponding points x_i in an image, find the 3D to 2D projective mapping taking each X_i to x_i .

3. **Fundamental matrix computation:** Given a set of points x_i in one image and a set of corresponding points x_i' in another image, find the **fundamental matrix F** relating the two images.

4. **Trifocal Tensor computation:** Given a set of point correspondences $x_i \leftrightarrow x_i' \leftrightarrow x_i''$ across three images, compute the trifocal tensor T_i^{jk} relating points or lines in three views.

“**Estimation** is all about figuring out some **unknown quantities** given **noisy measurements** related to that object.”

2D Homography Estimation

- Let us begin with homography estimation. Suppose we have a **set of points** x_i in **image #1** and their **corresponding points** x_i' in **image #2**. Our goal is to compute a homography matrix $H_{3\times3}$ such such that $\forall_i$, x_i' = Hx_i

- **Example:** Imagine we take **two photos** of the same **whiteboard**. In **image #1**, we mark several feature points, and in **image #2**, we attempt to mark **the corresponding points as precisely as possible**. We assume that each point x_i in **image #1** corresponds to a point x_i' in **image #2**.

- In practice, due to **noise and imperfections** in point selection, we cannot obtain **exact measurements**. Therefore, we aim to estimate the **best-fit homography** matrix $H_{3\times3}$ that accounts for these **measurement errors**

2D Homography Estimation

- To compute the homography matrix H, we need at **least 4 perfect point correspondences**.
- **Each point correspondence** provides **two linear constraints**, one for the x direction and one for the y direction.
- Since a homography matrix has **8 degrees of freedom**, a **minimum of 4 point correspondences** is required.
- Each correspondence yields **2 independent equations involving 9 unknowns** (the entries of H).
- When we combine 4 such correspondences, we **obtain 8 equations in 9 unknowns**.

- If **all point measurements are perfectly accurate**, the **null space** of the resulting **design matrix** will contain only one vector — the **homography matrix $H_{3\times3}$**.

Problems in 2D Homography Estimation

- In real-world scenarios, we rarely have perfect measurements. Errors in identifying corresponding points are common, even a 1 or 2 pixel difference can **occur due to discretization, lens distortion, or limitations in the image sensor.**
- These small errors can significantly impact the computed **homography matrix,** even subtle inaccuracies can lead to major deviations.
- Even if we carefully **select 4 points and zoom in to mark them with precision**, it's still likely that measurement errors will introduce substantial inaccuracies in the **estimated transformation.**

Problems in 2D Homography Estimation

- Ideally, we prefer to use **more than 4 point correspondences.**
- The intuition is that if we have many such points, the **random errors in measurements** will **cancel out through averaging**.

- However, having more than 4 points introduces **an over-constrained system.**
- This system may **no longer have an exact solution**, so we instead look for a **best-fit solution**, often using least squares.

How to solve over constraint system?

When we have **over constraint system** then what we will do

- **Least squares (maybe)**.
- We purpose of some **objective functions**.

- We have two **possible solutions**. Solution 1 is going to be better or worse than solution 2. How we decide whether one solution is better than the other.

- For example, if the **squared error between measurements and estimates is small**, then we will say that it is a good **solution or estimate**.

- In the **estimation problems**, whenever we have an **over constraint system**, then we must propose **some objective function** and determine **some algorithm that will minimize that objective function**.

How to solve over constraint system?

- **Hartley and Zisserman**, they have a name **for the best possible algorithm for estimation**. It is a general idea across all problems. They named it the **Gold Standard algorithm** for **any estimation problem**.

- **“Any estimation algorithm minimizes the cost function that is the best possible cost function under a certain assumption.”**

Minimizing Errors in Homography Estimation

- Use more than the **minimum 4 correspondences** to improve robustness.
- Apply **RANSAC (Random Sample Consensus)** to identify and exclude **outliers** in **point correspondences**.
- Use **least squares optimization to** compute the **best-fit homography from an over-constrained system**.
- Perform **bundle adjustment** to **jointly refine camera parameters** and **point positions** for better accuracy.
- **Preprocess images** to **correct for lens distortion and normalize coordinates to reduce numerical instability.**

Why Not Always Use the Gold Standard Algorithm?

- In some problems, **minimizing the least squares error** is **appropriate**. But in others, a different cost function may be more suitable.
- For example, we might **minimize** the **negative log-likelihood** of the data under a **probabilistic model,** a valid, and widely-used approach.
- When proposing an **estimation algorithm**, we must ask: Why are we **using this method**? Is it **optimal for our objective**?

Why Not Always Use the Gold Standard Algorithm?

- If there's a **"Gold Standard"** algorithm known to perform best, why are we not using it? Is there a reason, such as **practicality or complexity**?

- Sometimes we choose **suboptimal methods** (e.g., in the Travelling Salesman Problem) because the ideal solution is **too expensive to compute**.

Why Avoid the Optimal Algorithm?

- Sometimes, even when we know the **optimal algorithm,** we choose not to use it. Why?

- One reason is that it may be **computationally expensive,** the optimal objective function might take too long to evaluate.
- In such cases, we prefer **faster, analytical solutions** that provide **reasonably good results**, even if they **are not optimal.**

- It's important to acknowledge that **we're using a sub-optimal method**, and assess how close it comes to the **Gold Standard solution.**

Numerical vs Analytical Solution

- The **analytical solution** gives the **exact solution**, whereas the **numerical solution** gives the **approximate solution**.

- **DLT (Direct Linear Transformation)** is an **analytical solution** with a lot of different estimation problems.

Statistical Estimation of a Homography

- **DLT (Direct Linear Transformation)** approximates the exact estimation of a homography by solving a system of linear equations that **constrain the homography parameters.**

- Instead of using **exactly 4 point correspondences**, we use **more than 4 to reduce the impact of measurement errors**. With more correspondences, the effect of **noise averages out**.

- **Measurement errors** are assumed to be **independent** and randomly distributed. Increasing the **number of measurements reduces the variance in the estimate.**

- This approach is grounded in the principles of statistical estimation: **more data leads to more reliable and stable results.**

Geometric vs Algebraic Functions

Geometric Function:

- Measures actual distance in the image space (e.g., Euclidean distance).
- **Example: Reprojection error: $\|x_i' - Hx_i\|$.**
  - **Non-linear optimization**; **more accurate** but **computationally intensive.**
  - Reflects real-world visual accuracy.

Algebraic Function:

- Measures how well equations are satisfied (e.g., Ah ≈ 0).
- Used in DLT: **minimizes ||Ah||.**
- Linear and easy to **compute using SVD**.
- Not always aligned with visual accuracy.

Practical Use:

- Start with **algebraic (DLT)** for **initial estimate**.
- Refine using **geometric methods** (e.g., bundle adjustment).

DLT (Direct Linear Transformation) and Error Functions

- The key idea in DLT is to use **more than 4 correspondences** to help reduce **estimation error** caused by **measurement inaccuracies.**

- Our aim is to **minimize** the **error in the estimated homography**. This error often arises because we cannot measure pixel locations with perfect precision.

- The **DLT algorithm** minimizes a specific kind of error, not necessarily the **geometric error** we might intuitively expect. Instead, **DLT minimizes an algebraic error**, which is simpler to compute but **lacks direct geometric meaning**. It's important to note that the function being minimized in DLT does not always have a clear geometric interpretation.

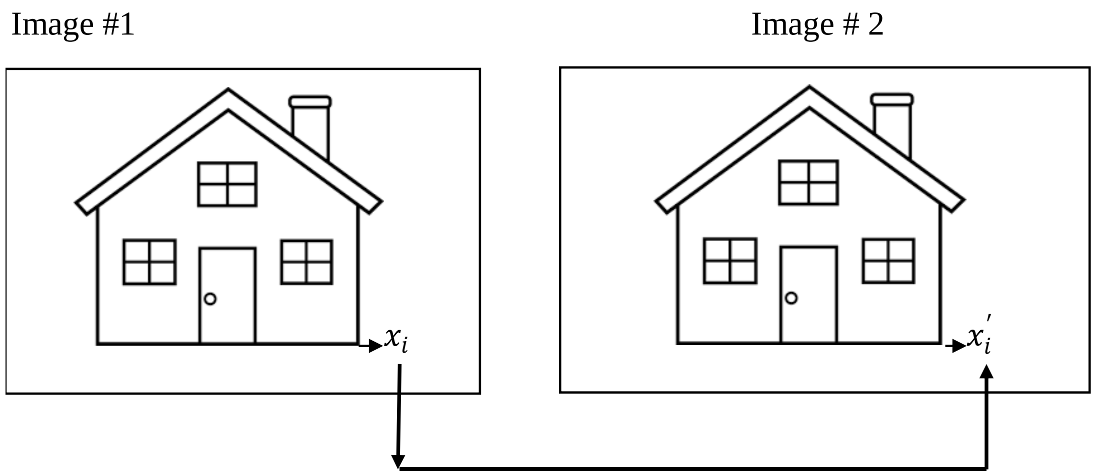
DLT (Direct Linear Transformation)

- We'll **formulate a linear system** to solve for the parameters of the **homography matrix H** in 2D.

- This time, we'll take a different approach. Previously, we used the **cross product of vectors** $\vec{x}_i$ and $\vec{x}_i'$ to derive **constraints** on the homography.

- So, we will look it differently this time. We have corresponding points we called them $x_i \leftrightarrow x_i'$

- Now, instead, we'll focus directly on the **corresponding points**, denoted as $x_i \leftrightarrow x_i'$, to construct our equations.



- We identify specific points in **Image #1** and find their **corresponding points in Image #2**. For example, let's denote the point x_i in the **left image** and its match x_i' in the **right image**. These pairs are assumed to be in **correspondence**.
- It might be the **corner of the building**. Somehow through a **computational process** or **human process**, we have decided that these **two points are in correspondence**, and we have some number of correspondence at least **4 but maybe 20 or 30 or even 100 points correspondences** between these two images.

Homography and Error Considerations

- **Exact positions** of the **corresponding points cannot** be **perfectly measured**. This introduces measurement error, which in turn **leads to estimation error.**

- From the equations, we know that a homography exists between the two images, where **H represents** the **transformation** from Image #1 to Image #2.

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

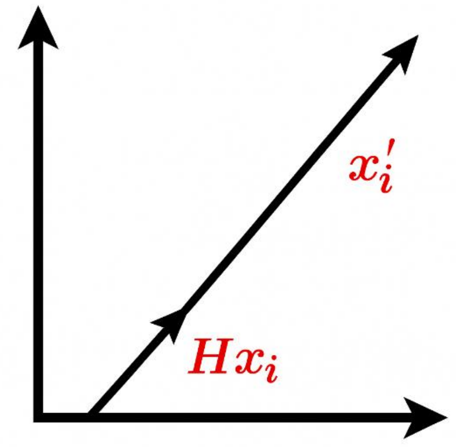
$$k = \frac{1}{w}$$

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.**

Geometric Meaning of Scaled Vectors in Homography

- The geometric consequence is that their **cross-product** is zero: $\boldsymbol{x_i' \times H x_i = 0}$.

- **This cross-product being zero means the vectors lie on the same line or plane**, they span a degenerate parallelogram.

- Geometrically, the **magnitude of the cross-product** gives the area of the **parallelogram formed by the vectors**.

- Hence, when the area is **zero**, the vectors are aligned, confirming the relationship

$$\vec{x}'_{3\times 1} \times H_{3\times 3} \ \vec{x}_{i\ 3\times 1} = 0$$

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

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

Let

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

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

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

Equation (1) $\Rightarrow$

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

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

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

$$\Rightarrow [0 \quad 0 \quad 0 \quad -w_i' \, x_i \quad -w_i' \, y_i \quad -w_i' w_i \quad y_i' \, x_i \quad y_i' \, y_i \quad y_i' w_i]_{1\times9}$$

$$[h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad h_{33}]^T_{1\times9} = [0]_{1\times1}$$

$$[0 \quad 0 \quad 0 \quad -w_i' \, x_i \quad -w_i' \, y_i \quad -w_i' w_i \quad y_i' \, x_i \quad y_i' \, y_i \quad y_i' w_i]_{1\times9} \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}_{9\times1}$$

$$= [0]_{1\times1}$$

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

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

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

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

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

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

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

$$\Rightarrow \begin{bmatrix} w'_i\, x_i & w'_i\, y_i & w'_i\, w_i & 0 & 0 & 0 & -x'_i\, x_i & -x'_i\, y_i & -x'_i w_i \end{bmatrix}_{1\times 9}$$

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

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

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

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

$$\vec{h} = \begin{bmatrix} \vec{h}^1 \\ \vec{h}^2 \\ \vec{h}^3 \end{bmatrix}, \text{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$ ---------------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

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

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

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).