Computer Vision

Dr. Syed Faisal Bukhari

Associate Professor

Department of Data Science

Faculty of Computing and Information Technology

University of the Punjab

Textbook

Multiple View Geometry in Computer Vision, Hartley, R., and Zisserman

Richard Szeliski, **Computer Vision: Algorithms and Applications**, 1st edition, 2010

Reference books

Readings for these lecture notes:

- Hartley, R., and Zisserman, A. **Multiple View Geometry in Computer Vision**, Cambridge University Press, 2004, Chapters 1-3.

- Forsyth, D., and Ponce, J. **Computer Vision: A Modern Approach**, Prentice-Hall, 2003, Chapter 2.

These notes contain material c Hartley and Zisserman (2004) and Forsyth and Ponce (2003).

References

These notes are based

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

- ❑Dr. Sohaib Ahmad Khan CS436 / CS5310 Computer Vision Fundamentals at LUMS

Recall: Translation, Rigid Body Transformation, Similarity, Affine

1. $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

2. $\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$

3. $\begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$

4. $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$

Recall: Translation Group

3 x 3 transform defined in Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} 1 & 0 & \mathrm{t_x} \\ 0 & 1 & \mathrm{t_y} \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{3\times1}$$

$$\Longrightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} \mathrm{x} + \mathrm{t_x} \\ \mathrm{y} + \mathrm{t_y} \\ 1 \end{bmatrix}_{3\times1}$$

Recall: Rigid Body Transformation(Euclidean Transformation)

3 x 3 transform defined in Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times 1} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}_{3\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{3\times 1}$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times 1} = \begin{bmatrix} x\cos\theta - y\sin\theta + t_x \\ x\sin\theta + y\cos\theta + t_y \\ 1 \end{bmatrix}_{3\times 1}$$

Recall: Similarity Group

3 x 3 transform defined in Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} \mathrm{s}\cos\theta & -\mathrm{s}\sin\theta & \mathrm{t_x} \\ \mathrm{s}\sin\theta & \mathrm{s}\cos\theta & \mathrm{t_y} \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{3\times1}$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} \mathrm{sx}\cos\theta - \mathrm{sy}\sin\theta + \mathrm{t_x} \\ \mathrm{sx}\sin\theta + \mathrm{sy}\cos\theta + \mathrm{t_y} \\ 1 \end{bmatrix}_{3\times1}$$

Recall: Affine Group

3 x 3 transform defined in Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{3\times1}$$

$$\Longrightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} a_{11}\mathrm{x} + a_{12}\mathrm{y} + a_{13} \\ a_{21}\mathrm{x} + a_{22}\mathrm{y} + a_{23} \\ 1 \end{bmatrix}_{3\times1}$$

- Contains **rotation, scaling, shear, translation** and any combination thereof
- **Preserves Parallel lines**

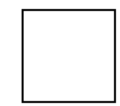
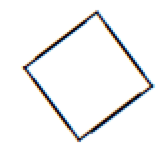
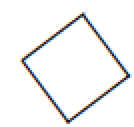
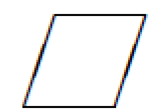
Recall: Projective Group (Homography)

3 x 3 transform defined in Homogeneous coordinates

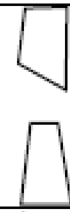
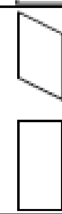
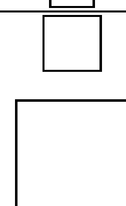
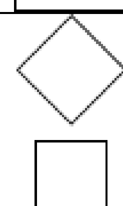
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times 1} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}_{3\times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{3\times 1}$$

- Simulates out of plane rotations
- **Preserves straight lines**
- Physical Interpretation: Plane + Camera

Hierarchy of 2D Transformations

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles Length Ratios	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism Length Ratios along a line	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines Length Cross-Ratios along a line	

2D projective geometry
A hierarchy of transforms

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, order of contact, tangent discontinuities and cusps, cross ratios
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines, linear combinations of vectors, the line at infinity $\mathbf{l}_\infty$
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Homography

- **Projective transformations** involve taking a point written in homogeneous coordinates, $\vec{x} = (x_1, x_2, x_3)^T$, and applying an **arbitrary linear transformation** that maps one 3D space into another.

- If we consider $\vec{x} = (x_1, x_2, x_3)^T$ as a **vector in $\mathbb{R}^3$**, then we're just applying a linear transformation from one vector space to another 3D space.

- These transformations are part of a special and useful class in **projective geometry ($\mathbb{P}^2$),** known as projectivities, homographies, or collineations.

- **Note:** Any **linear transformation** between vector spaces can be represented using **matrix multiplication**.

Homography: Degrees of Freedom

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\vec{\mathbf{x}}\,' = \mathrm{H}\,\vec{\mathbf{x}}$$

- H has how many degrees of freedom?
- Since H is a **homogeneous matrix**, it has **8 degrees of freedom** and one degree is lost due to **scale invariance**.
- In **projective geometry**, we represent this **3×3 transformation** as **matrix H**.
- A **homography** defines a **projective mapping** between **two planes.**
- **Note: Homographies** are **linear transformations** expressed in **homogeneous coordinates**.

What Types of Transformations Can We Achieve with Homography?

- A **homography** can model a variety of geometric transformations using matrix multiplication.

- We can **rotate** a point or object around an axis. This transformation is expressed through the homography matrix.

- We can enlarge or shrink an object by multiplying its coordinates with a **scale** factor, also handled by a homography.

- **Shearing** distorts the shape such that the x and y axes are no longer perpendicular in the transformed coordinate system. This too can be modeled using a homography.

- In essence, homography provides a unified framework for performing these **transformations through linear algebra.**

Central projection maps points on one plane to points on another plane

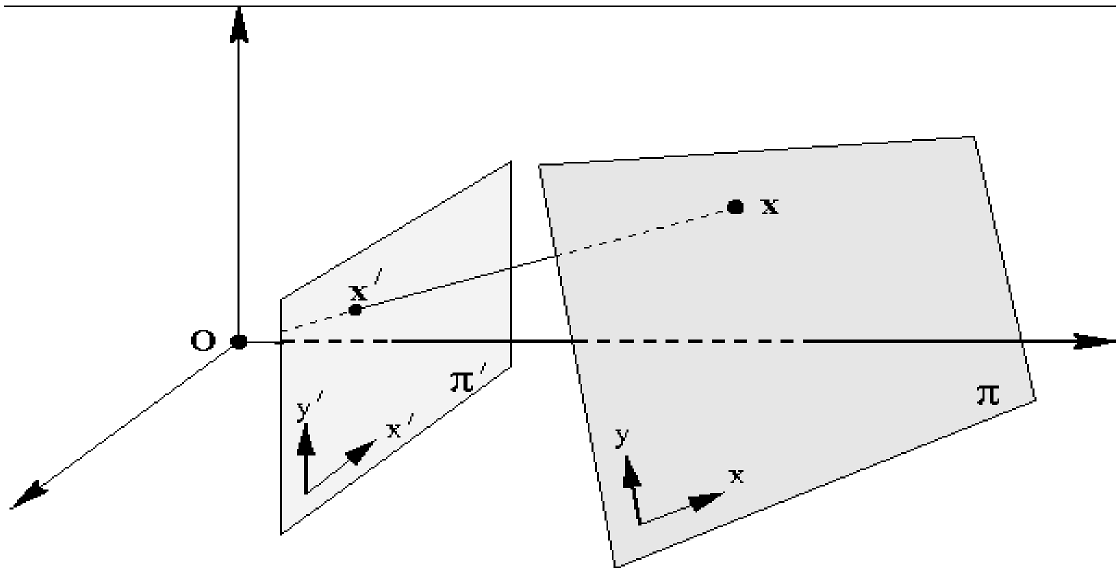


Fig 2.3

On RHS, of above figure we have a plane π. It has its own local coordinate system xy. This plane is embedded in $\mathbb{R}^{\mathbf{3}}$ in some arbitrary position, some arbitrary orientation and location in 3 space. On LHS, we have another plane $\boldsymbol{\pi}'$. We want to map our point, $\vec{x}$ in $\boldsymbol{\pi}$ to a point $\vec{x}'$ in π'. We take the aid of H in order to map $\vec{x}$ to some point $\vec{x}'$ in π'. This kind of transformations are accomplished by **homographies.**

Central projection maps points on one plane to points on another plane

- On the **right-hand side (RHS),** we see a **plane $\boldsymbol{\pi}$**. It has its own local coordinate system denoted by xy. This plane is situated in $\mathbb{R}^{\mathbf{3}}$, meaning it can have any orientation or position in 3D space.

- On the **left-hand side (LHS),** there is another **plane π′** with its own coordinate system x′y′

- Our goal is to map a point $\vec{x}$ on plane $\boldsymbol{\pi}$ to a corresponding point $\vec{x}'$ on plane π'.

- To achieve this mapping, we use a **homography matrix H**, which transforms the coordinates of $\vec{x}$ on $\boldsymbol{\pi}$ to the coordinates of $\vec{x}'$ on π'.

Limitations and Capabilities of Homography

- **Homography cannot model all types of transformations**, but it is still a powerful tool in projective geometry.

- For example, it **can perform shearing,** a transformation where the x and y axes become **non-orthogonal** in the output coordinate system.

- This demonstrates one of the key capabilities of homography: it allows for **non-rigid transformations** such as **rotation, scaling, and shearing**, but **not full 3D deformations** or depth-aware warping.

Application of a homography

- Why we need homographies?
- They come all the time in **computer vision**. They are very important objects. Let's look at a practical example of how we can **detect a homography** and apply it to serve a meaningful purpose, such as correcting **perspective distortions.**

(a)

(b)

Hartley Zisserman (2004). Fig 2.4

Application of a homography

- The LHS of Fig 2.4 represents some face of a building. If we take **any four points** on the face of this building. In the real world, these points must **lie on some plane**. Let's forget window is slightly indented towards the main plane.

- In Fig 2.4 (a) we have some effects of **perspective cameras** because camera is not pointing orthogonally to this plane. We also observe **parallel lines in the world** seems to be intersecting to **some finite points in Fig 2.4 (a)**. If we extend these points in Fig 2.4 (a) we find these parallel lines are actually intersecting.

- One of the important application in computer vision is the **eliminate** this **perspective distortions**.

- We take an image and transform in a way that the **parallel lines** in the **world** remains **parallel lines in an image**. This phenomenon is called **rectification**.

Application of a homography

- **Rectification:** It is an **elimination** of **projective distortions**. There are many cases, when it is useful to do rectification. In coming lectures, we will discuss it.

- To figure out, how to map points from plane based on Fig 2.4 (a) to points in plane based on Fig 2.4 (b).

- We need to **find homography** that models this transformation. It turns out for any mapping between **perspective projections** of two planes we find the **homography** that relates them.

Estimating Homography from Point Correspondences

- A **homography matrix** has **8 degrees of freedom**. To solve for these unknowns, we **need 8 independent equations**.

- Interestingly, just **4 point correspondences** between **two planes** are sufficient to estimate a homography.

- Each **point correspondence** (**a point on the plane** in one image **matched** with its counterpart on the plane in another image) provides **2 linear equations.**

- Therefore, **4 such correspondences yield 8 equations**, which are enough to compute the homography matrix.

Finding a Homography

- Understanding how to **estimate a homography** is crucial, and we'll dedicate significant time to this topic.

- Initially, we'll assume that we have **exact correspondences of 4 points between a plane in the left image and a plane in the right image**. Using this, we'll derive the **analytical solution** that maps these points.

- As we progress through the course, we'll explore additional real-world challenges, such as:
  - **Noise** in the image
  - **Measurement error**
  - **Outliers** or incorrect point correspondences between two images

Finding a Homography

- In the later stages, we'll learn how to **automatically estimate** a **homography** between **two images**, even in the presence of noise and mismatches.

- Suppose we have **four corresponding points** between these **two planes**. Can we find a homography that maps between these points?
- It turns out that we can

$\vec{x}' = H\ \vec{x}$

$$\Rightarrow \begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Where H = $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$

$x_1' = h_{11}x_1 + h_{12}x_2 + h_{13}x_3$ **-------------------------------(1)**

$x_2' = h_{21}x_1 + h_{22}x_2 + h_{23}x_3$ **-------------------------------(2)**

$x_3' = h_{31}x_1 + h_{32}x_2 + h_{33}x_3$ **-------------------------------(3)**

The vector $\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix}$ is only defined **upto scale**. We can multiply it by any factor. We want **inhomogeneous coordinates** of this point then we divide $\frac{x_1'}{x_3'}$ and $\frac{x_2'}{x_3'}$

(1)÷(3):

$$\frac{x_1'}{x_3'} = \frac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \text{ ---------------(4)}$$

Which represents the actual x position of the point in the output image.

(2)÷(3):

$$\frac{x_2'}{x_3'} = \frac{h_{21}x_1 + h_{22}x_2 + h_{23}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \text{ ---------------(5)}$$

So, we have **some equations** in the **parameters of h**. Both of **these equations are constraint the values of the parameters** $\boldsymbol{h_{11}, h_{12}, h_{13}, \ldots, h_{33}}$. We can convert these equations into a linear system. Currently these 2 equations don' t look linear.

(4) ⇒

$$h_{31}x_1x_1' + h_{32}x_2 x_1' + h_{33}x_3x_1' = h_{11}x_1x_3' + h_{12}x_2x_3' + h_{13}x_3 x_3'$$

$$\Rightarrow h_{11}x_1x_3' + h_{12}x_2x_3' + h_{13}x_3 x_3' - h_{31}x_1x_1' - h_{32}x_2 x_1' - h_{33}x_3x_1' = 0 \text{---}\mathbf{(6)}$$

(5) ⇒

$$h_{31}x_1 x_2' + h_{32}x_2 x_2' + h_{33}x_3x_2' = h_{21}x_1x_3' + h_{22}x_2x_3' + h_{23}x_3 x_3'$$

$$\Rightarrow h_{21}x_1x_3' + h_{22}x_2x_3' + h_{23}x_3 x_3' - h_{31}x_1 x_2' - h_{32}x_2 x_2' - h_{33}x_3x_2' \;\; 0 \text{--}\mathbf{(7)}$$

- We have two equations (i.e., Equations (6) and (7)) in **9 unknowns**. We should apply some technique to reduce them to **two equations** in **8 unknowns** or there are some linear estimation technique.

- There are variety of ways to solve the Equation (6). We have one linear equation in several unknowns.

These measurements, we get from the image or specified by the user.

Example term $\boldsymbol{h_{31}} x_1 x_1'$

In this case, only one unknown

$h_{11}x_1x_3' + h_{12}x_2x_3' + h_{13}x_3\,x_3'$ - $h_{31}x_1x_1' - h_{32}x_2\,x_1' - h_{33}x_3x_1'$= 0------(6)

We can write (6) as matrix multiplication

$$[x_1x_3' \quad x_2x_3' \quad x_3\,x_3' \quad 0 \quad 0 \quad 0 \quad -x_1x_1' \quad -x_2\,x_1' \quad -x_3x_1']_{1\times 9} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}_{9\times 1} = 0$$

------------(8)

$$h_{21}x_1x_3' + h_{22}x_2x_3' + h_{23}x_3\, x_3' - h_{31}x_1\, x_2' - h_{32}x_2\, x_2' - h_{33}x_3x_2'$$ = 0-------(7)

We can write (7) as matrix multiplication

$$\begin{bmatrix} 0 & 0 & 0 & x_1x_3' & x_2x_3' & x_3\, x_3' & -x_1x_2' & -x_2\, x_2' & -x_3x_2' \end{bmatrix}_{1\times 9} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}_{9\times 1} = 0$$

-------------------(9)

Stacking up equations (8) and (9):

$$\begin{bmatrix} x_1x_3' & x_2x_3' & x_3\,x_3' & 0 & 0 & 0 & -x_1x_1' & -x_2\,x_1' & -x_3x_1' \\ 0 & 0 & 0 & x_1x_3' & x_2x_3' & x_3\,x_3' & -x_1x_2' & -x_2\,x_2' & -x_3x_2' \end{bmatrix}_{2\times 9} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}_{9\times 1}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{2\times 1}$$

If we take the 4 point correspondences in the two images, then we have **8 equations in 9 unknowns** and the size of design matrix is 8×9

$$A_{8\times 9}\boldsymbol{h}_{9\times 1} = \mathbf{0}_{8\times 1}$$

- We have 8 equations in 9 unknowns. This system is under constraint. We don’t have enough constraint.

$$
\begin{bmatrix}
-x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1 x_1' & y_1\, x_1' & x_1' \\
0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 y_1' & y_1\, y_1' & y_1' \\
-x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 x_2' & y_2\, x_2' & x_2' \\
0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 y_2' & y_2\, y_2' & y_2' \\
-x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3 x_3' & y_3\, x_3' & x_3' \\
0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3 y_3' & y_3\, y_3' & y_3' \\
-x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4 x_4' & y_4\, x_4' & x_4' \\
0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4 y_4' & y_4\, y_4' & y_4'
\end{bmatrix}_{8\times 9}
\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}_{9\times 1}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{8\times 1}
$$

- The solution of above equation is happened to be the null space of A

  h = null (A)

- It turns out that any scalar times h solves this system. Since we have **8 equations** in **9 unknowns**, so there **exist infinite number of solutions** but all of them are just scalar times the null space of A

  i.e., the null vector of A.

Review

Definition: A **subspace** of $\mathbb{R}^n$ is any set *H* in $\mathbb{R}^n$ that has **three** properties:

The **zero vector** is in *H*.

For each ***u*** and ***v*** in *H*, the sum **u + v** is in *H*.

For each **u** in *H* and each **scalar *c***, the **vector *cu*** is in *H*.

In words, a **subspace** is ***closed*** under **addition** and **scalar multiplication**

Review

Definition: The **null space** of matrix A is the set Nul A of all solutions to the **homogeneous equation A$\vec{x}$ = 0**.

When A has n columns, the solutions of A$\vec{x}$ = 0 belong to $\mathbb{R}^n$, and the null space of A is a subset of $\mathbb{R}^{\boldsymbol{n}}$. In fact, Nul A has the properties of a subspace of $\mathbb{R}^{\boldsymbol{n}}$.

Dimension of Null Space: The dimension of null space is the number of free variables. Free variables not corresponding to leading one.

The dimension of the column space is the number of the leading 1’s.

Trapezoid: A trapezoid is quadrilateral with two sides parallel

What Is the Dimension of the Null Space?

- The **dimension of the null space** (also called the **nullity** of a matrix A) is the number of **linearly independent vectors** that form a basis for the null space of A.

- In simple terms:
- It tells you **how many independent directions** exist in the space of solutions to $Ax = 0$.

Dimension of the Null Space: Why Does It Matter?

- If the null space is **zero-dimensional**, the only solution is the **trivial solution:** $x = 0$.

- If the null space is **one-dimensional**, all solutions **are scalar multiples** of a **single nonzero vector**.

- If it's **two-dimensional** (or more), there **are infinitely many directions** in which solutions exist.

How Is It Calculated?

- Let's say **matrix A** has dimensions $\boldsymbol{m \times n}$ and **rank** $\boldsymbol{r}$.

  Then:

  nullity(A) = $n - r$

Where:

- **n is the number of columns** (number of unknowns).
- **r is the rank of A** (number of independent equations).

Example in Homography

- Suppose you have an 8×9 matrix A, built from 4 point correspondences.
- n = 9 (9 elements in homography matrix H)
- r = 8 (assuming full row rank from 8 equations)
- So, nullity(A)=9−8 =1