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

- **Introduction to perspective projection by Thomas Sheppard**

- http://jwilson.coe.uga.edu/EMAT6680Fa08/Eckstein/Instructionalunit/day1/day1.html

Understanding 2D Homographies: Exact Estimation from Point Correspondences

- In previous lecture, we explore how to compute a **2D homography** exactly using **4 point correspondences**.

- Each correspondence contributes **two linear constraints**, leading to a system of **8 equations in 9 unknowns** (the elements of the homography matrix).

- This system yields an **infinite number of solutions**, but if the equations are **linearly independent**, the solution space—called the **null space**—will be **one-dimensional**.

- That means there are infinitely many solutions, all differing only by a **single scale factor**.

Unique solution up to scale of h

- Each **point correspondence** contributes **two linear equations** involving the **parameters of the homography matrix H**.

- So, with **4 point correspondences**, we obtain **8 linear equations in 9 unknowns**.

- This results in an 8 × 9 **design matrix** A, representing the system:

  $A_{8\times9}\boldsymbol{h}_{9\times1} = \mathbf{0}_{8\times1}$

- Here, **A** contains the **coefficients of the parameters in the unknown vector h**.

- If the 8 rows of A are linearly independent, the null space of A is one-dimensional, meaning the solution for h is **unique up to a scale factor.**

Rank of A and Degenerate Homography

- If there are **8 independent linear equations**, the **rank of matrix A is 8**. According to linear algebra, an 8×9 matrix can have a **maximum rank of 8**.

- This is valid only if all **8 rows of A** are **linearly independent**.

- However, if you **have 4 point correspondences** and **any two of the point pairs are identical**, it results in **a degenerate configuration of the homography**.

- In such a case, the **matrix A has rank less than 8**.

- Consequently, the system does **not yield a unique solution for h.**

Homographies are homogeneous objects

- When we define a **homography between two images**, it's important to remember that homographies are **homogeneous quantities**.

- This means that multiplying the homography matrix by any non-zero scalar does **not** change the transformation—it still represents the **same homography**.

- The reason is that we're working with **homogeneous coordinates,** where scaling by a non-zero factor **preserves the equivalence of points.**

Understanding "Up to Scale" in Homography

What do we mean by up to scale in the context of a homography?

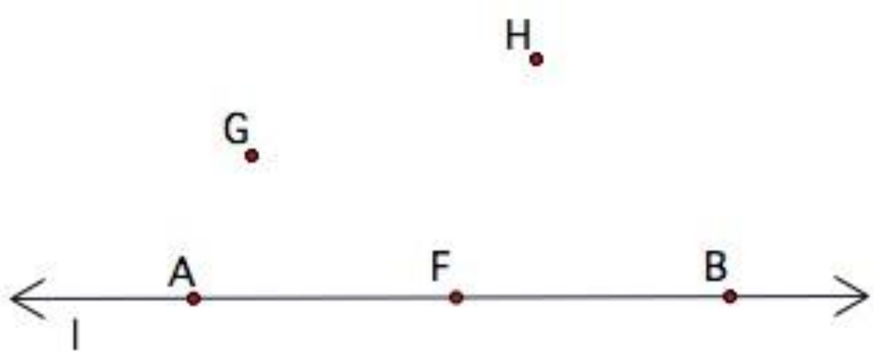
- If $H_{3\times3}$ is a solution to $A_{8\times9}\boldsymbol{h}_{9\times1} = \boldsymbol{0}_{8\times1}$ then **any scaled matrix kH** (where k ≠ 0) is also a **valid solution**.

- This holds because a **homography is a homogeneous transformation,** and we are working with points in **homogeneous coordinates.**

- **Note:** Homogeneous quantities are **invariant** under multiplication by a **non-zero scalar k**.

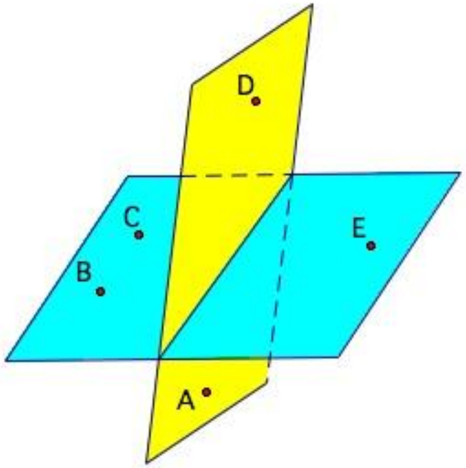
Review: Collinear points

Collinear points are points all in one line and **non collinear** points are points that are not on one line. Below points **A, F** and **B** are **collinear** and points **G** and **H** are **non collinear**.



Review: Coplanar points

- **Coplanar points** are points all in **one plane** and **non coplanar points** are points that are **not in the same plane**. Below points **B, C and E are coplanar, points D and A** are **coplanar** but points E and D would not be coplanar.



Recall: HOMOGENEOUS LINEAR SYSTEMS $\mathbf{Ax} = \mathbf{0}$

- A system of linear equations is said to be **homogeneous** if it can be written in the form $\mathbf{Ax} = \mathbf{0}$, where A is an $\mathbf{m \times n}$ **matrix** and $\mathbf{0}$ is the **zero vector** in $\mathbb{R}^{\boldsymbol{m}}$

  .

- Such a system $\mathrm{A}\mathbf{x} = \mathbf{0}$ *always* has at **least one solution**, namely, $\mathbf{x} = \mathbf{0}$ (the zero vector in $\mathbb{R}^{\boldsymbol{n}}$).

Recall: HOMOGENEOUS LINEAR YSTEMS $Ax = 0$

- This **zero solution** is usually called the **trivial solution**.

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

Degenerate Configurations

- **Definition:** A situation where a configuration does not determine a **unique solution** for a particular class of transformation is termed ***degenerate***.

- Note that the **definition of degeneracy** involves both the **configuration** and the **type of transformation**. The degeneracy problem is not limited to a minimal solution, however. If additional (perfect, i.e. error-free) correspondences are supplied which are also collinear (lie on **l**), then the degeneracy is not resolved.

Degenerate Configurations

- Consider a **minimal solution** in which a homography is computed using **four-point correspondences**, and suppose that **three of the points x_1, x_2, x_3** are **collinear**.

- The question is **whether this is significant**, if the corresponding points x'_1, x'_2, x'_3 are also **collinear** then one might suspect that the homography is **not sufficiently constrained**, and there will **exist a family of homographies** mapping x_i to x'_i.

- On the other hand, if the corresponding points x'_1, x'_2, x'_3 are **not collinear** then clearly there can be **no transformation H taking x_i to x'_i** since a **projective transformation** must **preserve collinearity**.

Degenerate Configurations

- **Any shape** is valid as long as the **four points are not collinear**.

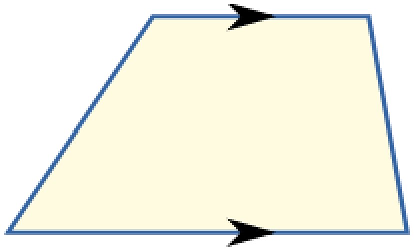
- **Two points** are **always collinear** —they define a straight line.

- If no **three points** lie on the **same line**, the configuration is **generally acceptable**.

- **Some degenerate cases** include having **3 points on a line** and a **4th point slightly off that line**. This still forms an unstable configuration.

- Ideally, **points forming a trapezoid or an irregular shape avoid degeneracy** and are suitable for computing homographies.

- A **trapezoid** (also called a **trapezium** in British English) is a **quadrilateral**—a four-sided polygon—with **at least one pair of parallel sides**.



- **Key Features:**
- It has **four sides**.
- **One pair** of opposite sides is **parallel** (called the **bases**).
- The other two sides are **non-parallel** (called the **legs**).
- If the non-parallel sides are equal in length, it's called an **isosceles trapezoid**.
- The **height** is the perpendicular distance between the parallel sides.

- **Reference:**
  https://www.mathsisfun.com/geometry/trapezoid.html

Degenerate Configurations

- In computer vision, we often encounter cases where solutions are **unique only up to scale**.

- A key challenge is constraining the scale during reconstruction tasks.

- **When the equations involved are not linearly independent, degenerate configurations arise**.

- For example, **having 4 points in one image** and **4 corresponding points** in another image allows a regular, consistent mapping without ambiguity.

Degenerate Configurations:

- Suppose we are given **4 points** that all lie on a **line in image 1**. These 4 points are correctly in correspondence to the other **4 points in image 2**. But these **4 points lie on a line in the two images**.

- Do you think we get a **unique solution for h**?

- No because **these 4 points are constraining** things **along that line**. We still have a **valid homography** mapping between these 4 points.

Degenerate Configurations:

- **For example**, we can **scale the image** in the direction **orthogonal** to the line **without effecting the mapping of 2 sets of points**. Actually, the geometric properties of these images will lead hopefully to a system of linearly independent **equations** in the coefficient of h, if so then we **have a unique solution**.

- **Note:**
  - If you ever implementing something like this then you should make sure that **no two** of your **four points is actually the same point**.
  - **No 3 points are collinear**.
  - These are some points about homographies.

Limitations of Exact Estimation of a Homography:

- Suppose we have **noisy points correspondences**. We don't get **exact measurements**. For example, we **don't get the exact corner of the windows** because our **window corner detector** is **approximate**.

- **The point** in the **left image** is **roughly equal** to **the point** in **the right image**. If we have **this kind of error**, then our **estimated homography** using 4 points correspondences would be **terrible**. It would be very **inaccurate**.

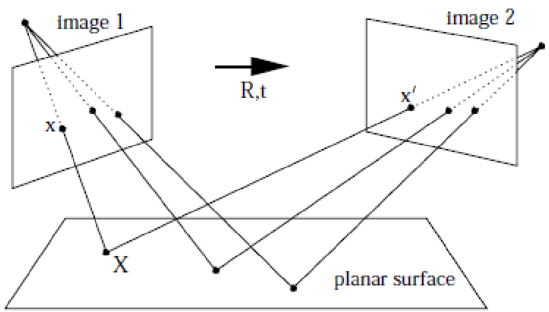
- **Consequently**, we look at **many points of correspondences** to get a better estimate of the geometric object of interest. In our case, a homography. That's one thing to look at. Also, there **exist outliers in the points**.

Limitations of Exact Estimation of a Homography

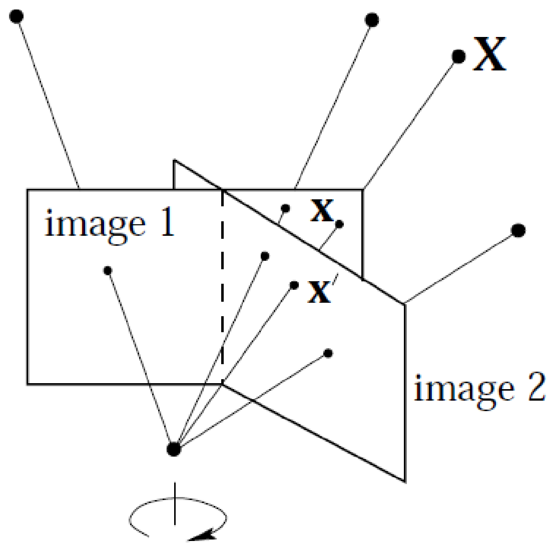
- Automatic methods are used **to find correspondences between two images, but they are inherently error-prone**.
- Despite best efforts, **errors in point matching** are common when aligning images.
- A key challenge is to design algorithms that can estimate **desired quantities accurately**, while still **tolerating some correspondence errors**.
- A major advantage of **homography is its visual nature**—**we can inspect the result and judge its accuracy directly.**

2D Homography Examples

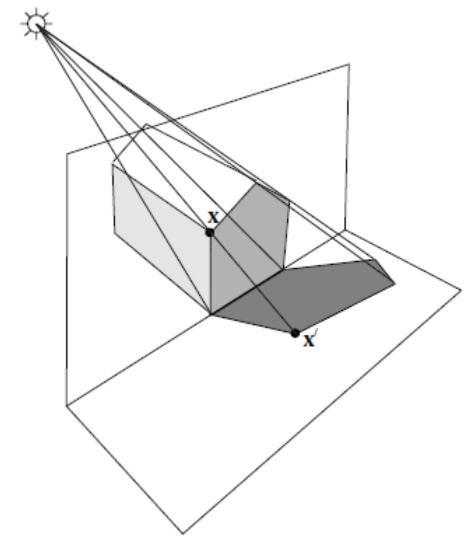
Here are a few of the most important examples of **homographies** (Hartley and Zisserman, 2004, Fig. 2.5):



Images of a plane from two cameras related by a **rotation** and a **translation**.



Images of **arbitrary objects** from **two cameras related by a rotation**.



Images of **shadows of planar** objects.

Note that images of **arbitrary** objects from two cameras related by a rotation and translation are **not** related by homographies.

2D Homography Examples

- If you **take two pictures** of this room and try to find homography between them, suppose you want to map points between two images. We should understand under what circumstances it **is geometrically valid to estimate a homography** and when it is not. Now we discuss the most critical cases, as in Fig 2.5 (in the previous slide), where a homography is **appropriate geometric entities** describing the mapping between two images.

- We will not talk about cameras and projective cameras. We have a basic understanding of a pinhole camera. It gives a point (focus) is called the center of the camera, and it provides us with an image plane. The image plane is behind the optical center of the camera.

2D Homography Examples

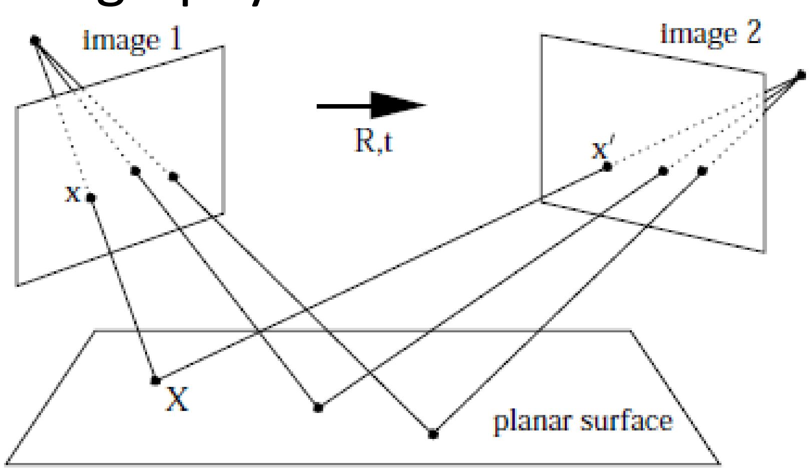
- In computer vision, we imagine that the **planar projection or the imaging plane is in front** of the camera. It has excellent geometric properties.

- For example, the **image does not get a flip or get inverted** when we assume that the **image plane is in front of the camera**. So, under what situations with real cameras or more or less real cameras (ideal pinhole camera) images are related by a homography.

Case 1: 2D Homography Examples

There are **three main situations**.

- **Case 1:** You are allowed to move the **camera**, but **the scene must be a plane**. If **the scene** is a **plane** and the camera is anything (e.g., rotation and translation are allowed), then you have a homography.



Images of a plane from two cameras related by a **rotation** and a **translation**

Case 1: 2D Homography Examples

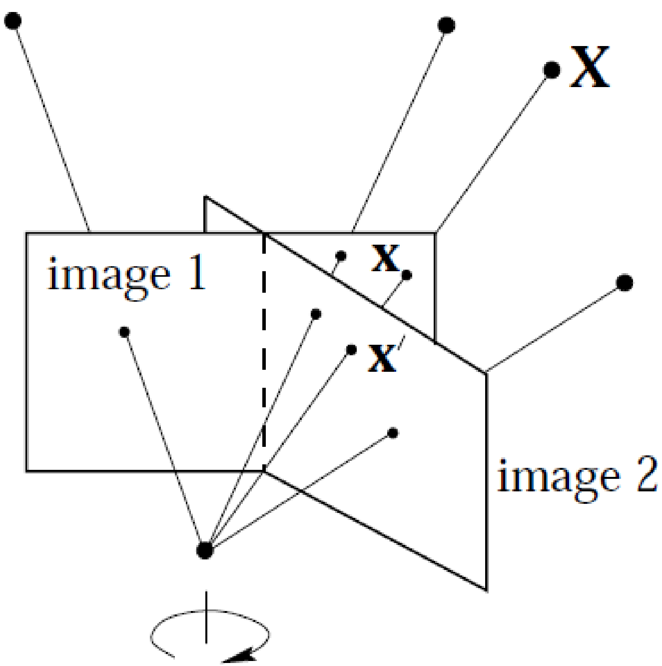
- If you take **two pictures of a planar surface** with a pinhole camera, then the **two images you get** will exactly be **related by a homography**. **Homography** is a projective mapping between two planes.

- We have a left plane, and we have mapped it to the right plane.

- There is an **exact mapping** between the **left plane** and the **right plane.**

Case 2: 2D Homography Examples

Case 2: Two images are taken with the same camera but a rotation about the optical center. If **scene is anything (e.g., not a planar surface)** and you have rotations about the center of the camera then you have a homography.



Images of **arbitrary objects** from **two cameras related by a rotation**

Case 2: 2D Homography Examples

- **For example:** If you **have a camera** and you take a picture in some direction that give would give you image 1 and then you rotate a camera about its **own center**? **How you rotate a camera about its own center?**

- That is the difficult problem to deal. We will talk about it later when we use this assumption.

Case 2: 2D Homography Examples

- Suppose **we have some mechanism** that has the capability to very accurately **rotate a camera about its own center**.

- Maybe we have a **tripod** and it could be accurately rotated about the center of the camera.

- If we succeed in doing this—though it's not easy—then the **two images captured** from two **different orientations** of the same camera will be **perfectly related by a homography**.

Case 2: 2D Homography Examples

- We are **free to change the focal length** of the camera and **still the two images** are related by **homography**.

- The key is that the **center of the camera or camera center** must be the same for **the two images**. We are projecting 3D space to the same point.

- Note: It’s the distance from the **camera center** (optical center) to the **image plane,** where the 3D world is projected.

Homography as a Rubber Sheet Deformation

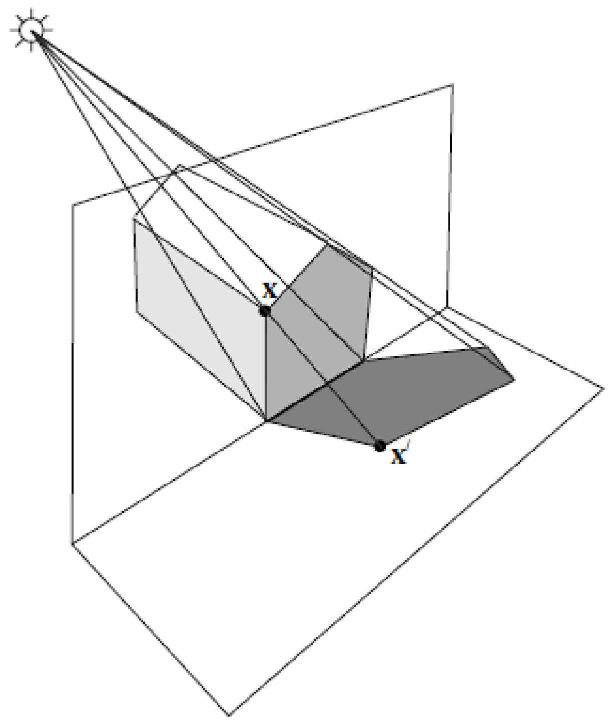
- Think of **homography** like **stretching a rubber** sheet pinned at the corners—you can **bend and flip it, but not tear it**.
- Homography can capture **non-linear transformations like these**, but **without tearing the surface**.
- This helps explain why **pure camera translation does not result in a homography**—it would require **"tearing"** the rubber sheet.
- **Try this:** Hold your finger in front of your face. Close one eye (e.g., left), then switch to the other.
- You'll notice the **finger jumps position—this** simulates **translation, not homography**.

Limits of Homography: Deformation vs. Translation

- Imagine the scene is static and both cameras are identical. Theoretically, you'd expect the same image.
- Like your eyes—same optics but small differences. What you get between left and right eye is not a smooth transformation.
- **Example:** Your finger appears on the right with one eye and left with the other. That's not a smooth mapping.
- **Homography cannot** represent this **kind of deformation caused by shifting viewpoints**.
- If the scene is **arbitrary** and you **move the camera**, **you can't relate the two images with a homography**.

Case 3: 2D Homography Examples

Case 3:



Images of **shadows of planar** objects.

Case 3: 2D Homography Examples

Case 3:

- This is the final case—perhaps a bit less exciting, but still important.

○ When you have a **planar surface** and want to understand the **shadow it casts** (e.g., due to sunlight), you're dealing with a **projective mapping between two planes**.

○ For example, **imagine a building**. The outer edge of the building is **one plane**, and **the ground**, where the shadow falls, is the **second plane**.

○ Essentially, you're seeing **projections** of points from one **plane** onto another **plane**.

○ That is, points from the surface of the building are mapped to corresponding points on the ground.

Case 3: 2D Homography Examples

Case 3: Example

- Suppose the sun is so far away from you. Maybe all points are on your body are approximately on the same plane w.r.t sun. So if you take the silhouette of my body as cast by the sun onto a planar surface, then **homography will approximately relate these points**.

Silhouette: Silhouette is the image of a person, animal, object or scene represented as a **solid shape** of a **single color**, usually black, with its edges matching the outline of the subject. The interior of a silhouette is featureless, and the whole is typically presented on a light background, usually white, or none at all.

The silhouette differs from an outline, which depicts the edge of an object in a linear form, while a **silhouette appears as a solid shape**. Silhouette images may be created in any visual artistic media but was first used to describe pieces of cut paper, which were then stuck to a backing in a contrasting color, and often framed

Silhouette of an aircraft

Reference: https://en.wikipedia.org/wiki/Silhouette

Approximate mapping

(a) (b)

Hartley and Zisserman (2004), Fig. 2.4

Approximate mapping

- Let us consider again the building example. **Let the point is on the corner of this white stuff inside the brick**. What is the point on the brick? We know we have some bricks inside. We have wooden frame or metal frame in the window. We know that the **depth is not same**.

- The point on the brick corner and the point on a **wooden corner do not lie on the same plane**. They are slightly offset from each other but when **the camera is so far away** from these **two points** that the difference in **depth is very small relative** to the distance of a **camera to that surface**. Then we will see in practice that this does not matter.

- The difference in **depth between the bricks and the wooden frame** of the window is **too small** to be measured by a camera. For practical purposes, many times we assume that the **scene is planar** even if it is not.

Approximate mapping

- Another example would be take **a tree**. The tree is clearly **not a planar object**. It is some kind of spherical ball. It has trunk and so on. If you are far away from the tree then the difference in the depth of the points on the tree is consistent with the plane. It depends on your application for all practical purposes these points are consistent with the plane w.r.t. to the camera. So in these cases it might be ok to do something **like in case 1**.

- **Note:** Suppose I have 4 points in the first image and I have only 3 points in the second image. Because one of the point is missed or cropped. Then we should pick any missed or cropped point then we can get a unique homography.

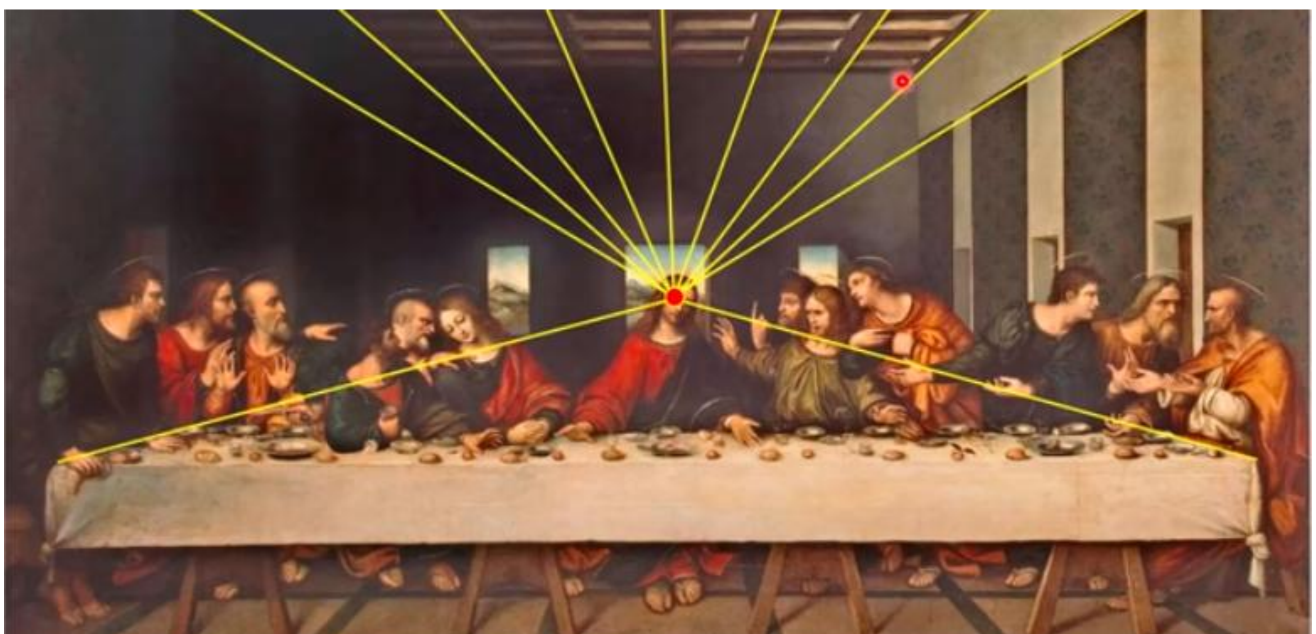
Perspective Projection

- ***Perspective projections*** are used to produce images which look natural. When we view scenes in everyday life far **away items** appear **small relative** to **nearer items**.

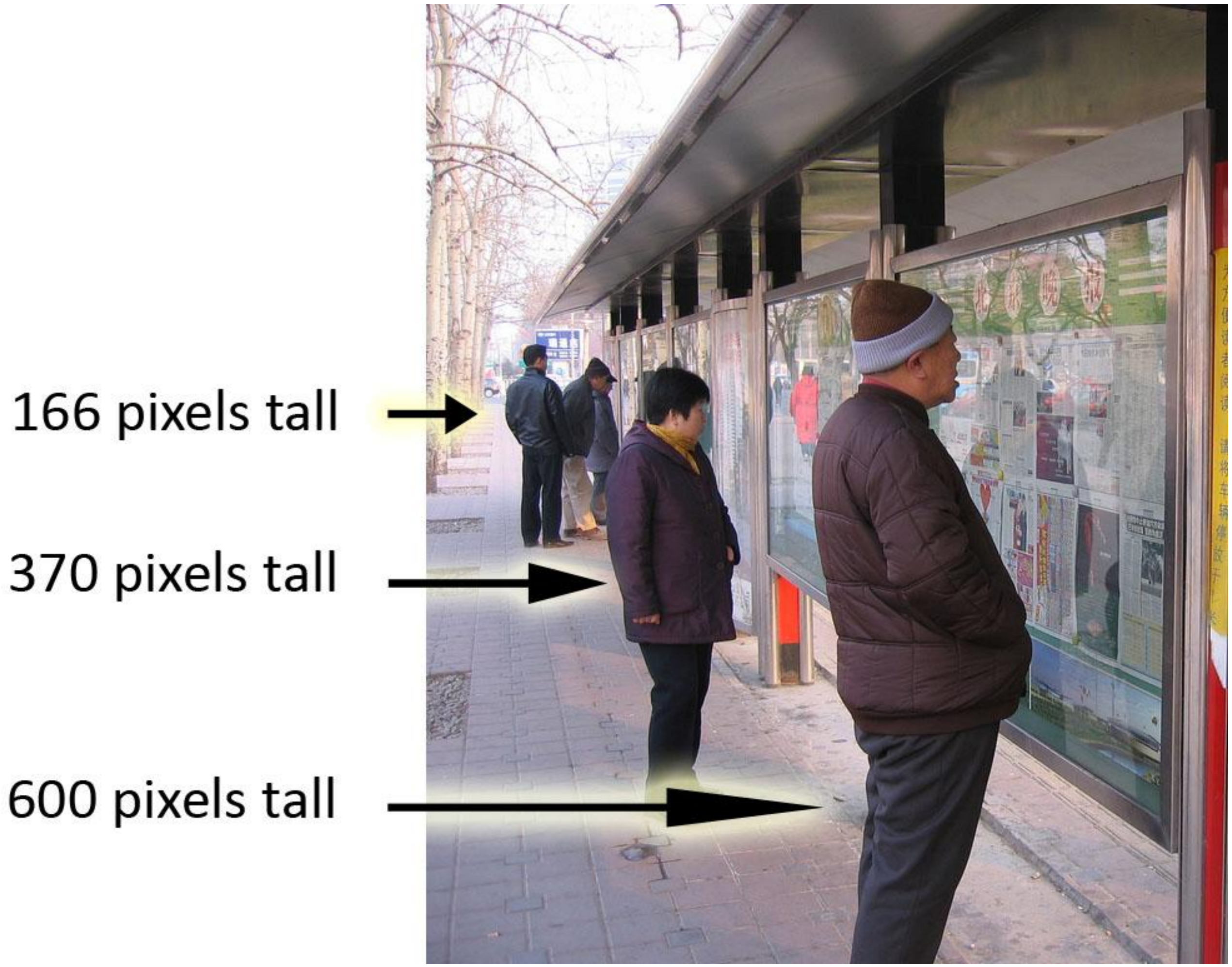
- A side effect of perspective projection is that **parallel lines** appear to **converge** on a ***vanishing point***.

- An important feature of **perspective projections** is that it **preserves straight lines**, this allows us to project only the end-points of **3D lines** and then draw a **2D line** between the projected endpoints.

Perspective Projection



OBJECTS IN THE REAL WORLD APPEAR SMALLER AS THEY MOVE FURTHER AWAY



Perspective Projection

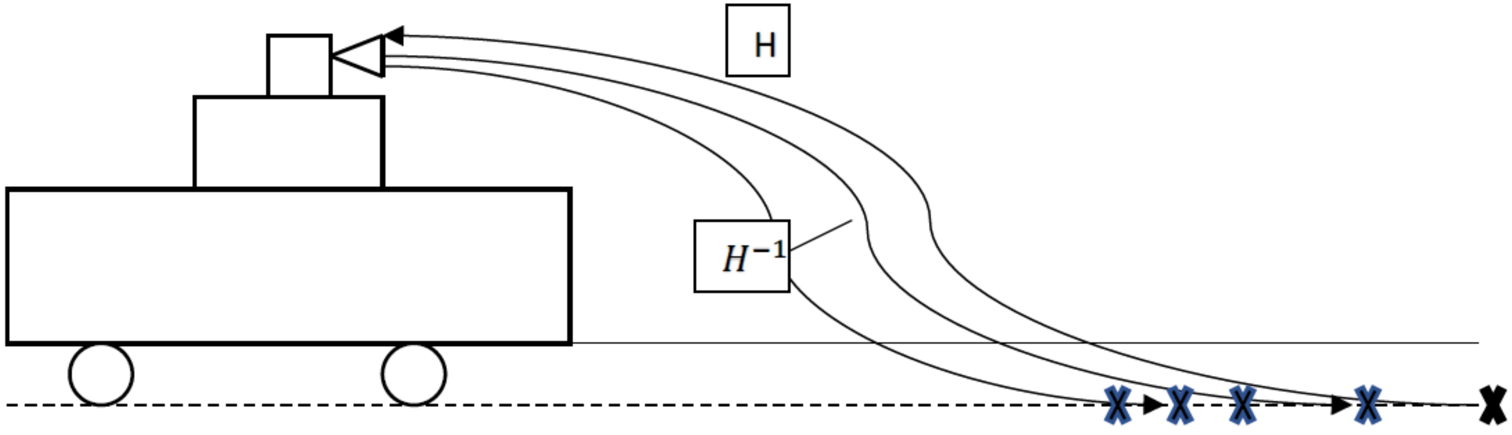
- Perspective projection depends on the relative position of **the eye** and the **view plane**.

- In the usual arrangement the **eye lies on the z-axis** and the **view plane** is the **xy plane**.

- To determine the **projection** of a **3D point** connect the **point** and **the eye by a straight line**, where the **line intersects the view plane**. This **intersection** point is the **projected point**.

Application of a homography

- We will discuss a practical application of homography. In literature, you will see many applications of a homography.



Application of a homography: Projective Distortion in Road Scenes

- Imagine a **road** viewed by a **camera mounted** on a **car**. The camera has an optical center and a projection plane in front of it. When **ground points** are projected onto the image plane, a **projective distortion occurs**.

- Points that are closer to the camera appear more spread out in the image compared to equally spaced points farther away.

- This results in a visual effect where the sides of the road appear to converge in the distance. **This distortion reflects the transformation from the ground plane to the image plane.**

Application of a homography

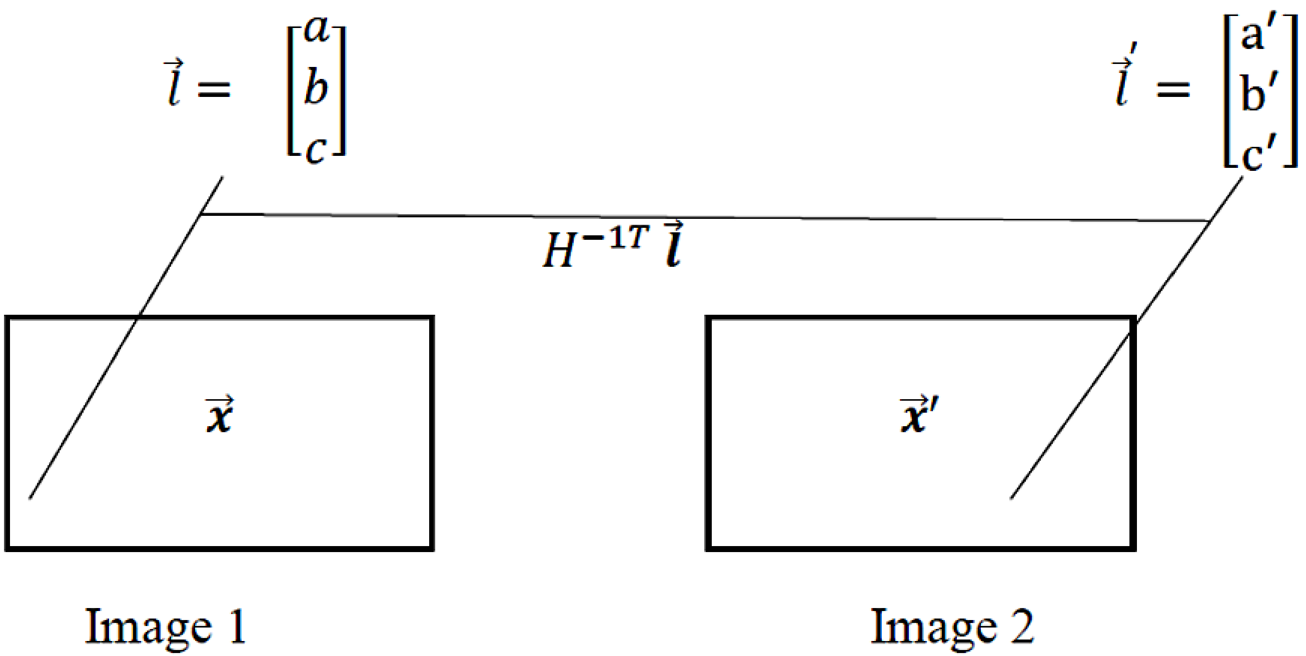
- We will talk about **camera calibration**. The camera calibration is quite straight forward to come up with the **homography** that will map once you fixed the camera **w.r.t car**. We can estimate this homography.

- If you understand 2D homography then you will be able to understand 3D homography and camera projection.

Homography of a line

- If you have a homography between two images then that can give you some interesting things as well. You could also come up with a **homography for a line**.

- If $\vec{x}' = H\,\vec{x}$ then how to map between lines?



Homography of a Line

- **A homography** between **two images** can relate both **points and lines**.
- This is useful for understanding how lines transform between views.
- If a point transforms as $x' = Hx$, **we ask: how does a line transform?**
- A **line** $\vec{l}$ in **Image 1** is represented as: $\vec{l} = [a \quad b \quad c]^T$
- Its corresponding line $\vec{l}'$ in **Image 2** is: $\vec{l}' = [a' \quad b' \quad c']^T$

- **The transformation** is given by: $\vec{l}' = H^{-1T}\vec{l}$

Homography of a line

If we have a **line** $(a, b, c)^T$ in **image 1** and we want to figure out what is the corresponding **line in image 2**. How can we understand this problem?

What makes a line. It is a cross product of **two points on that line**. If we have **a point** on **a line** then we know that the **dot product** of the **point** and the **line** is **zero**.

Suppose $\vec{x}_1$ and $\vec{x}_2$ lie on $\vec{l}$ then

$$\vec{x}_1^{\,T}\vec{l} = 0$$

$$\vec{x}_2^{\,T}\vec{l} = 0$$

Can we use this idea to come up with something? It turns out we have this relationship.

$\vec{l}' = H^{-1T}\,\vec{l}$ ------------------------------*(1)*

Homography of a line

What does it mean?

- If we take $\vec{l}$ and multiply by H^{-1T} then we will get the corresponding representation of a **line** in **another image**.

- Suppose $\vec{x}_1{}'$ and $\vec{x}_2{}'$ are **corresponding points** on line $\vec{l}'$. We want to figure out if this is true?

- We can take the dot product of each of these equations with the particular vector. We do matrix multiplication. We should be able to determine if the dot product is equal to zero then we have some correspondence.

$\vec{l}' = H^{-1T}\ \vec{l}$ ------------------------------(1)

$Take\ dot\ proudct\ of\ (1)\ with\ \ \vec{x}_1'^T$

$\Rightarrow \vec{x}_1'^T \vec{l}' = \vec{x}_1'^T\ H^{-1T}\ \vec{l}$ $\qquad \because\ \vec{x}_1'^T \vec{l}' = 0$

$\Rightarrow 0 = \vec{x}_1'^T\ H^{-1T}\ \vec{l}$

$\Rightarrow\ \vec{x}_1'^T\ H^{-1T}\ \vec{l} = 0$ --------------------(2)

Homography of a line

We are trying to get back to relationship between points

$\because \vec{x}' = H\,\vec{x}$.

Substitute the value of $\vec{x}'$ in (2), we get

$\Rightarrow (H\,\vec{x})^T H^{-1T}\,\vec{l} = 0$

$\Rightarrow \vec{x}^T\, H^T H^{-1T}\,\vec{l} = 0$ $\qquad \because (AB)^T = B^T\,A^T$

$\Rightarrow \vec{x}^T\,(H^{-1}H)^T\vec{l} = 0$

$\Rightarrow \vec{x}^T\,\vec{l} = 0$

It means $\vec{x}_1$ lies on $\vec{l}$.

Given a point homography, the corresponding line homography is

$\vec{l}' = H^{-1T}\vec{l}$

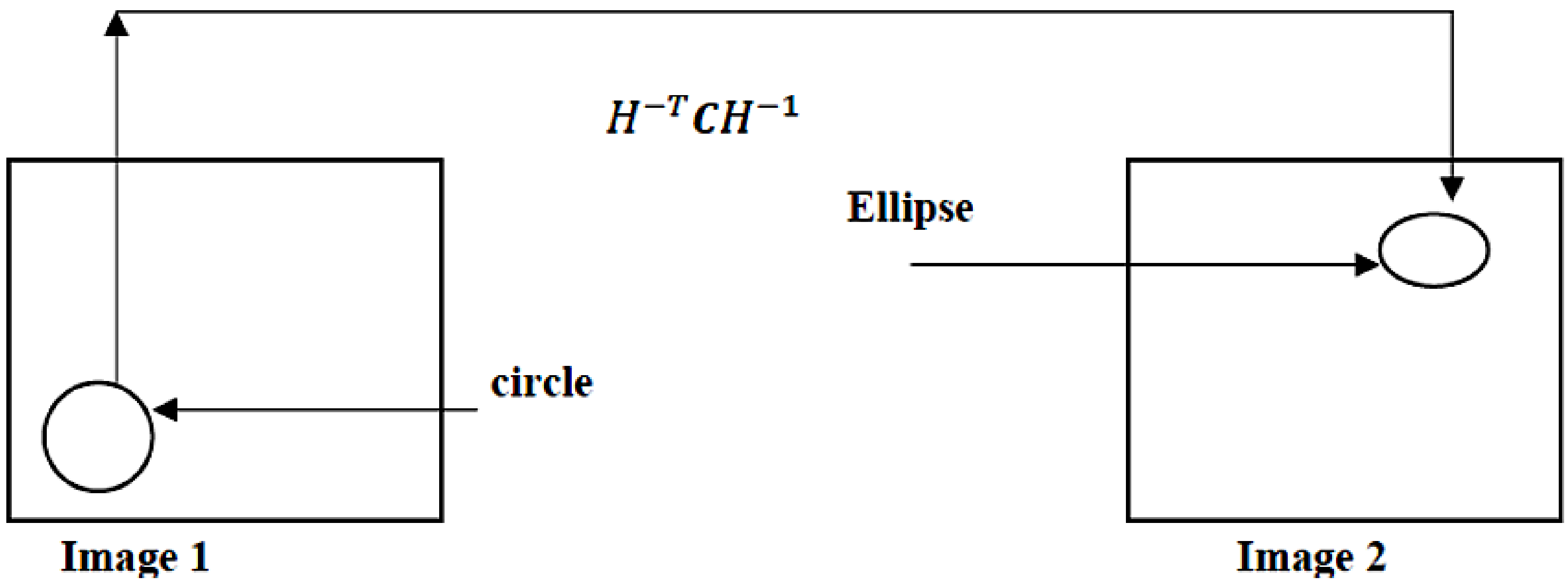
Or

$\vec{l}' = H^{-T}\vec{l}$

Conic Homography

The corresponding conic homography is

$C' = H^{-T} C\, H^{-1}$



Suppose we have circle in image 1 and we want to map it to ellipse in image 2. We can this transformation using the relationship $H^{-T} C\, H^{-1}$

Conic Homography

- A conic in one image can be transformed to another using the conic homography:

  $C' = H^{-T} C\, H^{-1}$

- For instance, if we have a **circle in Image 1** and wish to map it to an **ellipse in Image 2**, we apply the above transformation.

- This transformation uses the homography matrix H to compute the corresponding conic in the second image.