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**, 1^{st} 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).

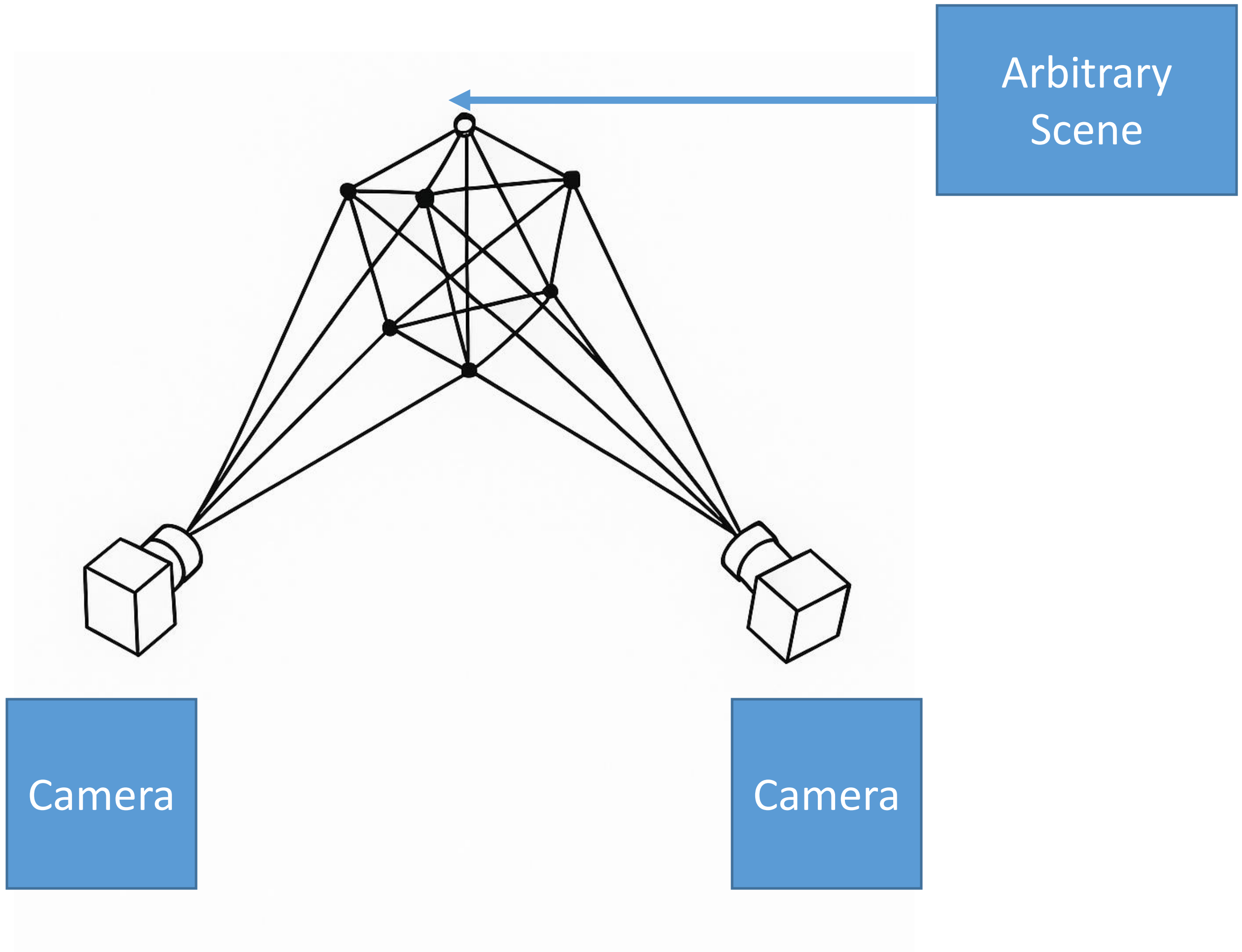
References

These notes are based

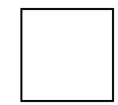
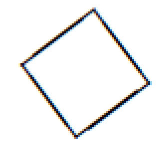
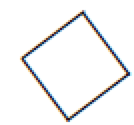
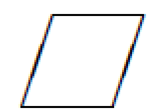
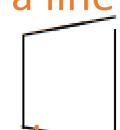
- Dr. Matthew N. Dailey's course: AT70.20: Machine Vision for Robotics and HCI

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

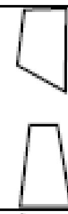
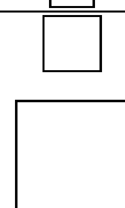
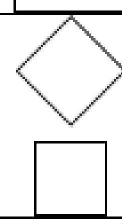
Homography: Two views of an arbitrary 3D scene



Hierarchy of 2D Transformations

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	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

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

Translation Group

3 x 3 transform defined in Homogeneous coordinates

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

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

Rigid Body Transformation (Euclidean Transformation)

3 x 3 transform defined in Homogeneous coordinates

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

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

Similarity Group

3 x 3 transform defined in Homogeneous coordinates

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

$$\Longrightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} \mathrm{sxcos}\theta - \mathrm{sysin}\theta + \mathrm{t_x} \\ \mathrm{sxsin}\theta + \mathrm{sycos}\theta + \mathrm{t_y} \\ 1 \end{bmatrix}_{3\times1}$$

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**

Projective Group (Homography)

3 x 3 transform defined in Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{3\times1} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}_{3\times3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{3\times1}$$

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

GL(3) - General Linear Group of Degree 3

- **Set of all invertible 3 × 3 matrices** (det ≠ 0)
- Notation: $GL(3) = \{ A \in \mathbb{R}^{3x3} \mid \det(A) \neq 0 \}$
- Includes **all linear transformations** of 3D space
- Preserves vector space structure, not scale. GL(3) keeps the **“vector math”** consistent but allows geometric distortions (like stretching, skewing, and flipping) — it does **not** keep distances or proportions constant.
- Used in classical linear transformations

PL(3) - Projective Linear Group of Degree 3

- PL(3) = GL(3) modulo scalar matrices
- Two matrices are equivalent if **they differ by a non-zero scalar**
- Works with **homogeneous coordinates** (projective geometry)
- PL(3) transformations preserve projective structure
- Common in computer vision and image **homographies**

Summary: GL(3) vs PL(3)

GL(3):

- Invertible 3×3 matrices
- Includes scale
- Linear transformations

PL(3):

- Equivalence classes of GL(3)
- Ignores scale
- Projective transformations

Example: Matrix Equivalence in PL(3)

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B = 2A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

- In **GL(3): A ≠ B**

- In PL(3): A ∼ B (same **transformation up to scale**)

2D projective geometry
A hierarchy of transformations

We can create a hierarchy of transformations based on the **restrictions** that we put on a **linear transformation H**.

- The **real linear group GL(3)** consists of all invertible real **3×3 matrices**.[1]

- When we place all members of **GL(3)** related by **scale in an equivalence class**, we obtain the **projective linear group PL(3)**.

[1] **Recall** that a group is a set paired with an operation that has an inverse, associativity, an identity, and closure.

2D projective geometry
A hierarchy of transformations

There are three important **subgroups of PL(3):**

- The **affine group** in which the bottom row is **constrained to (0, 0, 1);**

- The **similarity group** in which the rows and columns of the upper-left **2×2 submatrix are orthogonal**;

- The **Euclidean or isometry** group in which the **upper-left 2×2 submatrix is orthonormal**.

Projective Linear Group PL (3)

- **PL(3):The elements of this set are** 3×3 homogenous invertible matrices, where we consider kH to be equal to H. We have a variety of **different subgroups**:

- The **affine group** within **PL(3**) is just the set of 3×3 homogeneous invertible matrices such that the **bottom row** is **0 0 1**. Homographies having this property has some special characteristics.

Projective Linear Group PL (3)

- The **similarity group** in which the rows and columns of the **upper left 2×2 submatrix are orthogonal**. In this case, we place an additional restriction on homography, which creates the upper left **2×2 submatrix should be orthogonal**. The rows and columns are orthogonal to each other which creates a group called the similarity group.

- **Finally**, we go even further and restrict this homography matrix even more then we obtain the **Euclidean or isometry group**. In this case, we have upper left **2×2 submatrix as orthonormal**.

Projective Linear Group PL (3)

- In the **similarity group**, we say the **2 × 2 submatrix** is **orthogonal**. It means rows and columns have to be **orthogonal** to each other. The dot product between each **row or each column** has to be **orthogonal, i.e., zero**.

- In the **Euclidean group**, we say the **upper left 2 × 2** is **orthonormal**, which means the **norm of each row or each column is 1**.

Projective Linear Group PL (3)

- First of all, the most **general type of homography** is called the **projective group**. This is called PL(3). In it, we are allowed to **write whatever co-efficient we want** for the elements of this matrix as long as this **matrix is invertible**, then we have elements of PL(3).

- What kind of transformation this group model?
- Projective homography can do a lot of different types of distortions.
- We can take a **trapezoid** and map it to a **different type of trapezoid**.

Projective Linear Group PL (3)

1. Affine transformation:

Here we are saying that the bottom row of this matrix is **0 0 1**. In this case, we are effectively preventing a certain type of distortion. We have **6 degrees of freedom**. Now the transformations, we can model are called using the affine transformations: **Rotations, scale, translations, and shear**.

2. Similarity transformation:

We have imposed further restrictions. It has only **4 degrees of freedom**. We will write the upper left 2×2 submatrix using the coefficient $r_{11}\ r_{12}\ r_{13}\ r_{14}$. The rows and columns of this matrix are **orthogonal to each other**. It allows us to model **translations, scales,** and **rotations**. But, **shear** is not part of similarity transformations.

Projective Linear Group PL (3)

3. Euclidean transformation:

It allows us to model **rotations and translations but not scales**. These transformations are very interesting, as we see later when we get to **3D because they model the possible motion of the rigid body**.

Invariant properties of PL (3)

- **Invariant property under Euclidean transformations:** In Euclidean transformations, if an object has length or area before applying transformations, then it will have the same **length** and **area** after applying the transformations.
-
- **Invariant property under affine:** The most important invariant property under affine is **parallelism**. If two lines are parallel in the first image and then you transform that image by affinities, then parallel lines still are parallel in the resulting image.

Invariant properties of PL (3)

- **Invariant property under Projective transformation:**
- **Projective transformation** are real-life transformations between points that are taken when the point lies in the same plane, and you take pictures of them with different cameras.

- It turns out there are some properties that are preserved. For example, **collinearity**. If you have a line in image #1 and you transform it by any homography. That **line** will still be a **line in a resulting image**.

2D projective geometry
Action of projectivities and affinities on ideal points

The key difference between a **projective** and **affine transformation** is that the vector **v** is not null for a projectivity.

What happens when we apply a homography H to a point at infinity?

Affinities map **ideal points** to **ideal points**:

$$\begin{bmatrix} A & \boldsymbol{t} \\ \boldsymbol{O}^{\boldsymbol{T}} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} = \begin{bmatrix} A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ 0 \end{bmatrix}$$

but **general projectivities** can map ideal points to **finite points**:

$$\begin{bmatrix} A & \boldsymbol{t} \\ \boldsymbol{V}^{\boldsymbol{T}} & v \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} = \begin{bmatrix} A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ v_1 x_1 + v_2 x_2 \end{bmatrix}$$

2D projective geometry
Action of projectivities and affinities on ideal points

- What happens when we apply a homography to a point at infinity?

  Affinities map ideal points to ideal points.

  "Parallel lines remain parallel under affine transformations."

- How can we interpret it?

  If you have transformations between two images. You would like to prove that **parallel lines** are still going to be **parallel** under **affinities**. It is very simple to prove because **affinities preserve parallelism**. When you have 2 parallel lines then their **point of intersection on a line at infinity**.

2D projective geometry
Action of projectivities and affinities on ideal points

- **Point at infinity in homogenous representation of a 2D point:**

  The third component of your vector representing the point is zero. If the **bottom row is 0 0 1** then **points remain ideal**.

- **Point at finite in homogeneous representation of a 2D point:**

  If you have **arbitrary bottom row** of your homography the that means you have arbitrary homography. In this case **ideal points are not preserved.**

2D projective geometry
Action of projectivities and affinities on ideal points

Similarity: circularity is invariant

Affinity: parallelism is invariant

Projectivity: the line at infinity becomes finite (parallel lines on the plane intersect at finite points on l_∞)

Hartley and Zisserman (2004), Fig. 2.6

2D projective geometry
Action of projectivities and affinities on ideal points

(a)

(b)

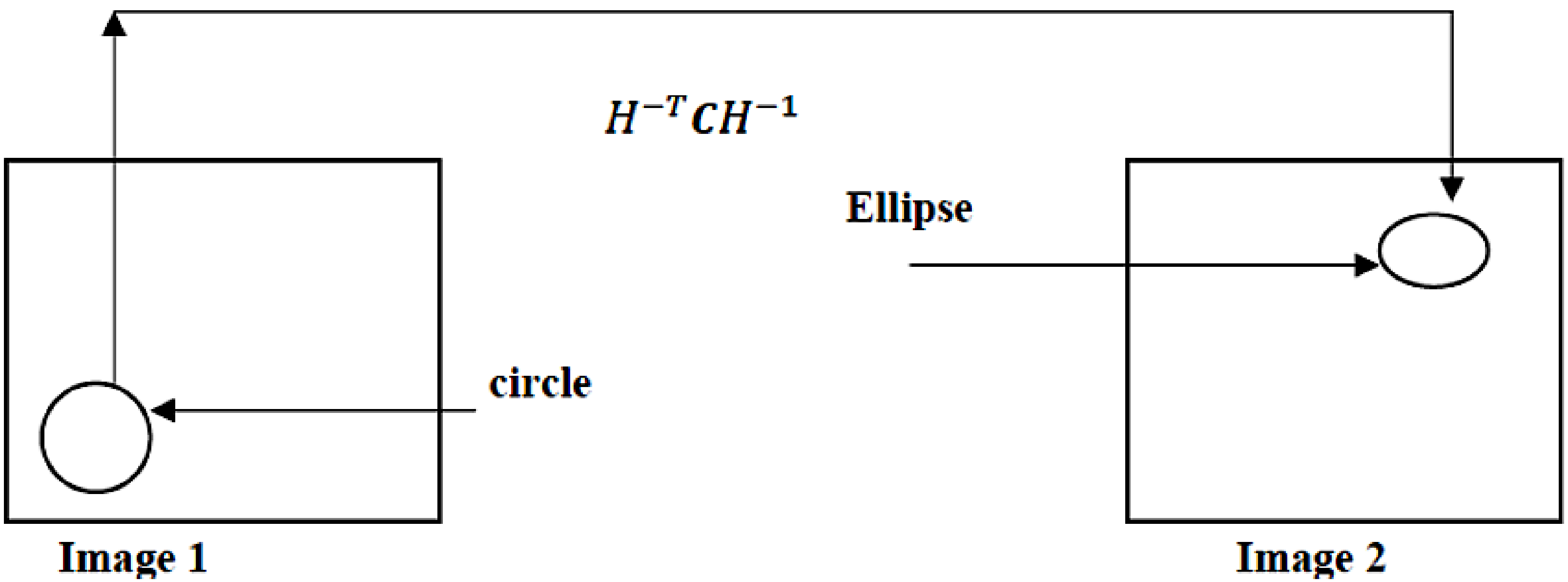
(c)

Distortions arising under central projection. Images of a tiled floor. **(a) Similarity**: the circular pattern is imaged as a circle. A square tile is imaged as a square. Lines which are parallel or perpendicular have the same relative orientation in the image. **(b) Affine**: The **circle** is imaged as an **ellipse**. Orthogonal world lines are not imaged as orthogonal lines. However, the sides of the square tiles, which are parallel in the world are parallel in the image. **(c) Projective**: Parallel world lines are imaged as converging lines. Tiles **closer to the camera** have a **larger image** than those further away.

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

The equation of a conic in **inhomogeneous coordinates** is

$ax^2 + bxy + cy^2 + dx + ey + f = 0$

i.e. a polynomial of degree 2. "**Homogenizing**" this by the replacements:

$x \longmapsto \frac{x_1}{x_3}, y \longmapsto \frac{x_2}{x_3}$ gives

$\Rightarrow ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$ ----------(1)

or in matrix form

$\vec{x}^T C \vec{x} = 0$

where the conic coefficient matrix C is given by

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

Conic

- C is homogeneous because scaling by non-zero constant does not change the conic

- It is a symmetric matrix. It has 9 elements but only 6 are unique. Also, it is a homogeneous matrix. We can scale C and we can still satisfy the equation $\vec{x}^T C \vec{x} = 0$

- It means we lose one degree of freedom.

- **Note:** It we divide every element by f then we fix the scale factor by f. As a result, we have **5 degrees of freedom** of this matrix

Five points define a conic

- Suppose we wish to compute the conic which passes through a set of points, $\boldsymbol{x_i}$.
- How many points are we free to specify before the conic is determined uniquely?
- The question can be answered constructively by providing an algorithm to determine the conic. From (1) each point $\boldsymbol{x_i}$ places one constraint on the conic coefficients, since if the conic passes through $(\boldsymbol{x_i}, \boldsymbol{y_i})$ then

$$a x_i^2 + b x_i y_i + c y_i^2 + d\, x_i + e\, y_i + f = 0$$

This constraint can be written as

$$[x_i^2 \quad x_i y_i \quad y_i^2 \quad x_i \quad y_i \quad 1]_{1\times 6}\, \vec{c}_{6\times 1} = 0$$

Where $\vec{c}_{6\times 1} = [a \quad b \quad c \quad d \quad e \quad f]^T$ is the conic C represented as a 6-vector

Five points define a conic

- Stacking the constraints from five points we obtain

$$\begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3 y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4 y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5 y_5 & y_5^2 & x_5 & y_5 & 1 \end{bmatrix}_{5\times 6} \vec{c}_{6\times 1} = \vec{0}_{5\times 1}$$

- The **conic** is the **null vector** of this 5×6 matrix. This shows that a conic is determined uniquely (up to scale) by **five points** in general position.

Transformation of conics

$\boldsymbol{x}^{\boldsymbol{T}}$C $\boldsymbol{x}$ = 0 ----------------------(2)

$\because$ $\boldsymbol{x}'$= H$\boldsymbol{x}$

Pre multiply by H^{-1}

H^{-1} $\boldsymbol{x}'$= H^{-1} H$\boldsymbol{x}$

$\Rightarrow H^{-1}$ $\boldsymbol{x}'$ = $\boldsymbol{x}$

or

$\boldsymbol{x}$ = H^{-1} $\boldsymbol{x}'$

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

$(H^{-1}\ \boldsymbol{x}')^{\boldsymbol{T}}\mathrm{C}H^{-1}\ \boldsymbol{x}' = 0$

$\boldsymbol{x}'^{\boldsymbol{T}}\ H^{-1T}\ CH^{-1}\ \boldsymbol{x}' = 0$

which is a quadratic form $\boldsymbol{x}'^{\boldsymbol{T}}\ C'\ \boldsymbol{x}'$ *with* $C' = H^{-1T}\ CH^{-1}$. This gives the transformation rule for a conic:

Under a point transformation $\mathbf{x}'$= H$\mathbf{x}$, a conic C transforms to

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

Exact homography using 4 points correspondences

$P_1(x_1, y_1, 1) \longmapsto P_1'(x_1', y_1', 1)$
$P_2(x_2, y_2, 1) \longmapsto P_2'(x_2', y_2', 1)$
$P_3(x_3, y_3, 1) \longmapsto P_3'(x_3', y_3', 1)$
$P_4(x_4, y_4, 1) \longmapsto P_4'(x_4', y_4', 1)$

$$\begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1x_1' & y_1\,x_1' & x_1' \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1y_1' & y_1\,y_1' & y_1' \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2x_2' & y_2\,x_2' & x_2' \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2y_2' & y_2\,y_2' & y_2' \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3x_3' & y_3\,x_3' & x_3' \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3y_3' & y_3\,y_3' & y_3' \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4x_4' & y_4\,x_4' & x_4' \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4y_4' & y_4\,y_4' & y_4' \end{bmatrix}_{8\times 9} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}_{9\times 1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{8\times 1}$$

$\mathrm{A}_{8\times 9}\mathbf{h}_{9\times 1} = \mathbf{0}_{8\times 1}$

h = null (A)

```
% Simple example of computing a homography from 4 exact point
%correspondences
% The 4-points in the left image

X = [ 200  250  210  255 ;     150  145  170  162 ;       1    1    1    1 ];

% The 4-points in the right image
Xprime = [ 200  250  200  250 ;         150  150  170  170 ;       1   1   1   1 ];

% We have Xprime = H * X
% Let's loop over the points and extract the linear equations in H

A = [];
```

```
% size(X,2) will give us column of X, which is 4
for iPoint = 1:size(X,2)
        x = X(:, iPoint);                    % for each point in left image
        xprime = Xprime(:,iPoint);           % for each point in right image
       % The equation is
        % xprime(1) = ( h11 x(1) + h12 x(2) + h13 ) / ( h31 x(1) + h32 x(2) + h33 )
        % Put this in the form ( coeffs * (h11 h12 h13 h21 h22 h23 h31 h32 h33)' )

        coeffs =  [ -x(1), -x(2), -1,  0, 0, 0,  xprime(1) * x(1),  xprime(1) * x(2),  xprime(1)];
        A = [ A ; coeffs ];
        % Do the same for the equation
        % xprime(2) = ( h21 x(1) + h22 x(2) + h23 ) / ( h31 x(1) + h32 x(2) + h33 )

        coeffs =  [0, 0, 0, -x(1), -x(2), -1, xprime(2) * x(1),  xprime(2) * x(2),  xprime(2)];
         A = [ A ; coeffs ];
end
```

```
% The solution is the right null space of A
h = null(A);
% Form the 9 parameters of H into a matrix
H = reshape( h, 3, 3)';
% Generate 100 random.
% diag([640,480]) creates a diagonal matrix 640   0; 0   480
 Xtest = [ ( rand(2,100)' * diag([640,480]) )' ;  ones(1,100)];

% Plot themfigure(1);
figure(1);
plot(Xtest(1,:), Xtest(2,:), 'rx' );
hold on;
plot(X(1,:), X(2,:),'bo');
axis([0,640,0,480],'equal','ij');
title('Left image points X');
hold off;
```

```
% Transfer the test points
Xtestprime = H * Xtest;
Xtestprime = Xtestprime ./ repmat(Xtestprime(3,:),3,1);

Xprimeest = H * X;Xprimeest = Xprimeest ./ repmat(Xprimeest(3,:),3,1);
% Plot themfigure(2);plot( Xtestprime(1,:), Xtestprime(2,:), 'rx’ );
figure(2);
plot( Xtestprime(1,:), Xtestprime(2,:), 'rx' );
hold on;
plot(Xprimeest(1,:),Xprimeest(2,:),'bo');
axis([0,640,0,480],'equal','ij');
title('Right image points X');
hold off;
```

Resulting right image after applying homography on the left image