

Applied Machine Vision

Projective Geometry

Matthew Dailey

Information and Communication Technologies
Asian Institute of Technology



Readings for these lecture notes:

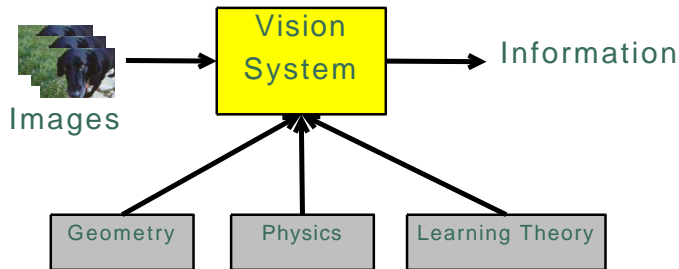
- Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004, Chapters 1–3.
- Szeliski, R. *Computer Vision: Algorithms and Applications*, Springer, 2010.

These notes contain material © Hartley and Zisserman (2004) and Szeliski (2010).

- 1 Introduction
- 2 2D projective geometry
- 3 3D projective geometry
- 4 Rigid (Euclidean) transformations

Introduction

Vision systems



The kind of information we want is application specific:

- 3D models
- Object poses
- Object categories
- Camera poses

Important applications include:

- Mobile robot navigation
- Industrial inspection and control
- Military intelligence
- Security
- Human-computer interaction
- Image retrieval from digital libraries
- Medical image analysis
- 3D model capture for visualization and animation

Introduction

Parts of the system

The “vision system” includes:

- Image **acquisition** hardware
 - Analog camera plus digital frame grabber, -or-
 - Digital camera with a fast serial interface (Firewire, USB, etc.)
- Image processing **support software**
- Your computer vision **algorithms**

Introduction

This summer

This semester we focus on algorithms for

- 3D reconstruction
- Learning (object detection and recognition)
- Sequential state estimation (e.g. tracking)

To understand modern 3D reconstruction techniques we need to understand how **cameras** transduce the **world** into **images**.

This involves understanding **projective geometry** and **camera models**.

Then we can begin to figure out how to invert what the camera does, or **reconstruct the 3D scene**.

- 1 Introduction
- 2 2D projective geometry**
- 3 3D projective geometry
- 4 Rigid (Euclidean) transformations

2D projective geometry

Introduction

We begin with 2D projective geometry because it's simple, then we'll generalize to 3D.

In particular, we consider **what happens when we perform projective transformations of the plane**.

Projective transformations model the distortions introduced by **projective cameras** (more on cameras later).

In projective cameras, funny things happen. Although straight lines stay straight, parallel lines are no longer parallel.

Projective geometry gives us the mathematics for these kinds of transformations.

2D projective geometry

The 2D projective plane: points in \mathbb{R}^2

A **point** in the plane can be represented as a pair (x, y) in \mathbb{R}^2 .

We consider \mathbb{R}^2 as a vector space and we write the point (x, y) as a vector.

This makes it possible to write transformations of points as matrices.

Generally, we write transformations on the left and points on the right, so we need to write points as column vectors, i.e. $(x, y)^T$.

We will typically write column vectors using bold upright symbols, e.g., $\mathbf{x} = (x, y)^T$.

2D projective geometry

The 2D projective plane: lines in \mathbb{R}^2

A **line** in the plane is normally represented by an equation like $ax + by + c = 0$. The parameters a , b , and c give us different lines.

This means we can write a line in \mathbb{R}^2 as the vector $(a, b, c)^T$.

2D projective geometry

The 2D projective plane: homogeneous coordinates and \mathbb{P}^2

$(a, b, c)^T$ represents the same line as $k(a, b, c)^T$ for any non-zero constant k .

A **homogeneous vector** is an equivalence class of vectors defined by scaling.

The set of homogeneous equivalence classes of vectors in $\mathbb{R}^3 - (0, 0, 0)^T$ is called the **projective space** \mathbb{P}^2 .

2D projective geometry

The 2D projective plane: homogeneous point representations

Since a point $\mathbf{x} = (x, y)^T$ lies on a line $\mathbf{l} = (a, b, c)^T$ iff $ax + by + c = 0$, we can equivalently write the inner product $(x, y, 1)(a, b, c)^T = (x, y, 1)\mathbf{l} = 0$.

If $(x, y, 1)\mathbf{l} = 0$, it is also true that $(kx, ky, k)\mathbf{l} = 0$.

This makes it convenient to represent a point $\mathbf{x} = (x, y)^T$ in \mathbb{R}^2 with the **homogeneous vector** $(x, y, 1)^T$.

This means the arbitrary point $\mathbf{x} = (x_1, x_2, x_3)^T$ in \mathbb{P}^2 can be used to represent the point $(x_1/x_3, x_2/x_3)$ in \mathbb{R}^2 .

This is nice! Why? Because now we can say that a homogeneous point $\mathbf{x} = (x_1, x_2, x_3)^T$ lies on line \mathbf{l} iff $\mathbf{x}^T \mathbf{l} = 0$.

2D projective geometry

The 2D projective plane: intersection of two lines

Another nice property: the **intersection of two lines** \mathbf{l} and \mathbf{l}' is the point $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$.

Reminder: the cross product of two vectors $\mathbf{x}_1 = (x_1, x_2, x_3)^T$ and $\mathbf{x}' = (x'_1, x'_2, x'_3)^T$ is defined as

$$\mathbf{x} \times \mathbf{x}' = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & x_2 & x_3 \\ x'_1 & x'_2 & x'_3 \end{vmatrix}$$

Another reminder: if $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$, \mathbf{x} is the vector normal to \mathbf{l} and \mathbf{l}' with magnitude equal to the area of the parallelogram formed by \mathbf{l} and \mathbf{l}' .

Proof: Since \mathbf{x} is orthogonal to both \mathbf{l} and \mathbf{l}' we know $\mathbf{l}^T \mathbf{x} = 0$ and $\mathbf{l}'^T \mathbf{x} = 0$, meaning \mathbf{x} lies on both \mathbf{l} and \mathbf{l}' !

Similarly, the line \mathbf{l} joining two points is just $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$.

2D projective geometry

Ideal points and the line at infinity

Where do the parallel lines $(a, b, c)^T$ and $(a, b, c')^T$ intersect?

The cross product turns out to be $(c' - c)(b, -a, 0)^T = (b, -a, 0)^T$ in \mathbb{P}^2 but this point has no inhomogeneous representation. (What is $(b/0, -a/0)^T$?)

We call such a point $(x_1, x_2, 0)^T$ in \mathbb{P}^2 an **ideal point** or a **point at infinity** along the direction $(x_1, x_2)^T$.

All points at infinity lie on the **line at infinity** $\mathbf{l}_\infty = (0, 0, 1)^T$ (remember that lines in \mathbb{P}^2 correspond to planes in \mathbb{R}^3 , points at infinity lie on the plane $x_3 = 0$, so we represent the plane $x_3 = 0$ by its normal vector $(0, 0, 1)^T$).

In \mathbb{P}^2 , then, we can say that any two lines intersect, even if they are parallel.

2D projective geometry

A model for the projective plane

Here's how to think of \mathbb{P}^2 : it is a **set of rays** through the origin in \mathbb{R}^3 .

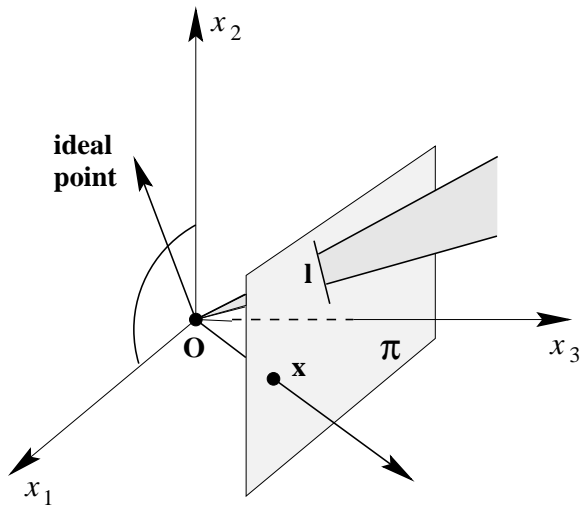
If we consider all vectors $k(x_1, x_2, x_3)^T$ as k varies, we obtain a ray through the origin. Each such ray is a **single point** in \mathbb{P}^2 .

A **line** between any two different points in \mathbb{P}^2 forms a **plane through the origin**.

Inhomogeneous representations of points and lines can be obtained by finding the intersection of their rays and planes with the plane $x_3 = 1$.

2D projective geometry

A model for the projective plane



Hartley and Zisserman (2004), Fig. 2.1

2D projective geometry

A model for the projective plane

In \mathbb{R}^3 , **lines** through the origin that lie in the x_1x_2 plane represent **ideal points** in \mathbb{P}^2 .

All other lines through the origin represent **points** in \mathbb{P}^2 .

Planes through the origin in \mathbb{R}^3 represent **lines** in \mathbb{P}^2 .

The vector $(a, b, c)^T$ representing a line in the Euclidean plane, when interpreted as a vector in \mathbb{R}^3 , is **orthogonal to the \mathbb{R}^3 plane representing the line in \mathbb{P}^2** .

You should try to prove that this must be so.

2D projective geometry

Conics

Conic sections or just **conics** are curves in the plane described by 2nd-degree equations.

In Euclidean geometry, conics can be parabolas, hyperbolas, and ellipses, or “degenerate” conics (two lines or a single line).

In inhomogeneous coordinates, we write conics as

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

We homogenize the equation by replacing x by x_1/x_3 and y by x_2/x_3 . This gives us

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$

2D projective geometry

Conics

With homogeneous points, similar to the point-line equation, conics can be written in matrix form:

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

Where the (symmetric) conic coefficient matrix \mathbf{C} is given by

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

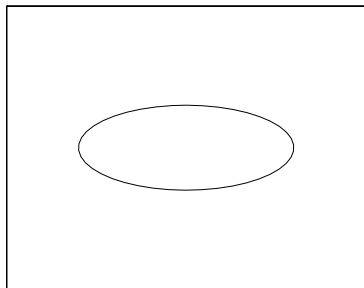
\mathbf{C} is **homogeneous** because scaling by a non-zero constant does not change the conic.

\mathbf{C} has five degrees of freedom, and each point on the conic gives us one equation linear in the conic's parameters, so we need five points to uniquely define a conic up to scale.

2D projective geometry

Conics

If we find the points \mathbf{x} satisfying $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$, we obtain the plane curve described by \mathbf{C} :



Solution of $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$
Hartley and Zisserman (2004), Fig. 2.2(a)

Exercise: write the circle about the origin with radius r in conic matrix form, and verify that the points on the circle satisfy $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$.

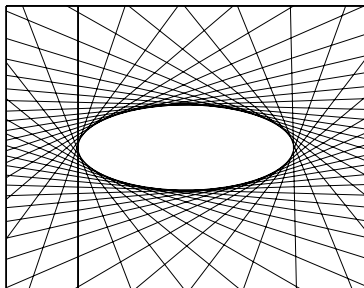
2D projective geometry

Conics

What happens if we take an arbitrary point \mathbf{x} on the conic and compute $C\mathbf{x}$, treating C as a **linear operator**?

We obtain a 3-vector whose inner product with \mathbf{x} is 0. This means that $\mathbf{l} = C\mathbf{x}$ necessarily represents a **line passing through \mathbf{x}** .

In fact, the resulting line is special: it is the **tangent** to the conic at \mathbf{x} .



The line $\mathbf{l} = C\mathbf{x}$ is the tangent to C through \mathbf{x} .

Hartley and Zisserman (2004), Fig. 2.2(b)

2D projective geometry

Projective transformations

There is an important class of transformations on \mathbb{P}^2 called **projectivities** or **homographies** or **collineations**.

A projectivity (homography) is a transform that can be represented by a 3×3 non-singular matrix H .

Homographies are **linear mappings** of homogeneous coordinates:

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

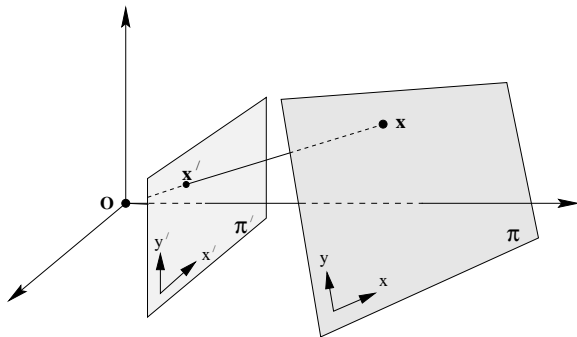
or simply $\mathbf{x}' = H\mathbf{x}$.

Since scaling H doesn't change the result, we say H is a **homogeneous matrix** with 8 degrees of freedom.

2D projective geometry

Projective transformations as central projections

Homographies can also be thought of as **central projections** mapping one plane to another:



Hartley and Zisserman (2004), Fig. 2.3

[Actually, central projections between rectilinear coordinate systems as shown here are called **perspectivities** and only have 6 degrees of freedom.]

2D projective geometry

Projective transformations: rectification

One application of homographies is **rectification**. As an example, suppose we want to remove projective distortion from a perspective image of a plane:



(a)



(b)

Hartley and Zisserman (2004), Fig. 2.4

2D projective geometry

Projective transformations: rectification

By picking any four points in an original image and the desired corresponding points in the new image, we obtain 8 linear equations in the 9 unknowns of H , allowing us to compute the parameters of H :

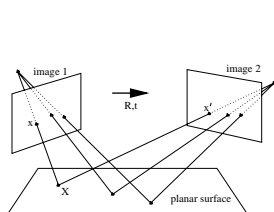
$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}.$$

See the example `compute_homography.m` that implements this idea.

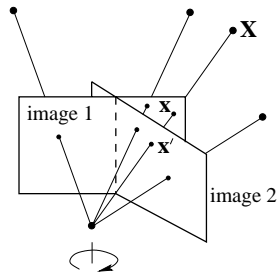
2D projective geometry

Projective transformations: examples

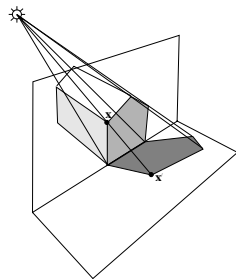
Here are a few of the most important examples of homographies (Hartley and Zisserman, 2004, Fig. 2.5):



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 projective geometry

Projective transformations: examples

Given a point homography $\mathbf{x}' = \mathbf{H}\mathbf{x}$:

- The corresponding **line homography** is $\mathbf{l}' = \mathbf{H}^{-T}\mathbf{l}$
- The corresponding **conic homography** is $\mathbf{C}' = \mathbf{H}^{-T}\mathbf{C}\mathbf{H}^{-1}$

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** $PL(3)$.


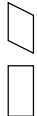
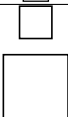
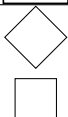
There are three important subgroups of $PL(3)$:

- 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.

¹Recall 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 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 l_∞
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

Hartley and Zisserman (2004), Table 2.1

2D projective geometry

Action of projectivities and affinities on ideal points

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

Affinities map ideal points to ideal points:

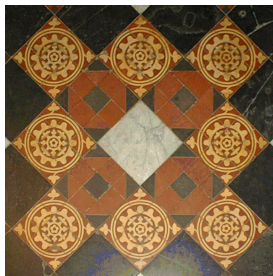
$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}$$

but general projectivities can map ideal points to finite points:

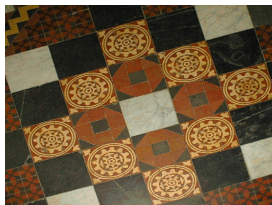
$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{pmatrix}$$

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 I_∞)

Hartley and Zisserman (2004), Fig. 2.6

- 1 Introduction
- 2 2D projective geometry
- 3 3D projective geometry**
- 4 Rigid (Euclidean) transformations

3D projective geometry

Introduction

Now we move to **projective 3-space**, or \mathbb{P}^3 .

Things will mostly generalize from \mathbb{P}^2 .

We'll use homogeneous coordinates in \mathbb{R}^4 to represent points **and planes** in \mathbb{P}^3 .

We'll see that parallel lines and parallel planes intersect on the **plane at infinity** π_∞ .

3D projective geometry

Points in \mathbb{P}^3

We represent a 3D point $(X, Y, Z)^T$ in homogeneous coordinates $\mathbf{X} = (X_1, X_2, X_3, X_4)^T$ with $X_4 \neq 0$ and

$$X = X_1/X_4, Y = X_2/X_4, Z = X_3/X_4.$$

Homogeneous coordinates with $X_4 = 0$ represent **points at infinity**.

A **projective transformation on \mathbb{P}^3** is a linear transformation on homogeneous 4-vectors, represented by a non-singular 4×4 matrix: $\mathbf{X}' = \mathbf{H}\mathbf{X}$.

\mathbf{H} is homogeneous and in general has 15 degrees of freedom.

3D projective geometry

Planes in \mathbb{P}^3

Whereas points and lines are dual in \mathbb{P}^2 , **points and planes** are dual in \mathbb{P}^3 .

A plane in 3-space is written

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0.$$

The equation is unaffected by scalar multiplication, so we can represent a plane as the **homogeneous vector** $(\pi_1, \pi_2, \pi_3, \pi_4)^T$.

Homogenizing the point \mathbf{X} , we get

$$\pi_1 X_1 + \pi_2 X_2 + \pi_3 X_3 + \pi_4 X_4 = 0$$

or simply $\pi^T \mathbf{X} = 0$ to express that the point \mathbf{X} lies on the plane π .

3D projective geometry

Points and planes in \mathbb{P}^3

Suppose we have 3 points in general position. We can find the plane they lie in by constructing the equation


$$X\pi = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} \pi = \mathbf{0}$$

As long as the 3×4 matrix X has rank² 3, the solution π is the **right null space**³ of X .

If X is rank 2, we have collinear points and the equation defines a **pencil of planes**⁴ with the line of collinear points as its axis.

²The rank of matrix A is the number of linearly independent rows or columns in A .

³The right null space of matrix A , written $\mathcal{N}(A)$, is the set of all solutions of $Ax = \mathbf{0}$.

⁴Just a fancy term for the set of planes intersecting at a given line. 

3D projective geometry

Points and planes in \mathbb{P}^3

Analogous to the definition of the line through two points in \mathbb{P}^2 , the plane through three points in \mathbb{P}^3 can be written in terms of determinants and minors.

We finally obtain (see text for proof)

$$\pi = \begin{pmatrix} (\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_3) \times (\tilde{\mathbf{X}}_2 - \tilde{\mathbf{X}}_3) \\ -\tilde{\mathbf{X}}_3^T (\tilde{\mathbf{X}}_1 \times \tilde{\mathbf{X}}_2) \end{pmatrix}$$

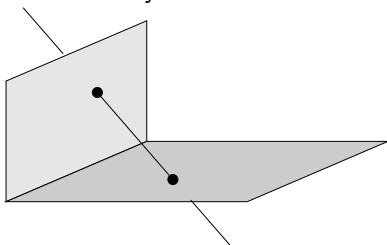
where $\tilde{\mathbf{X}}$ is the inhomogeneous representation of point \mathbf{X} .

So we can find the plane through three points by finding $\mathcal{N}(\mathbf{X})$ or by using some vector algebra.

3D projective geometry

Lines in \mathbb{P}^3

Lines in 3-space can be defined by their **intersection with two given planes**:



Hartley and Zisserman (2004) Fig. 3.1

This means a line in 3-space has **4 degrees of freedom**.

Wouldn't it be natural to represent lines as 5-element homogeneous vectors?

Unfortunately this doesn't work well with 4-vectors for lines and planes.

Several alternative representations for lines have been proposed.

3D projective geometry

Lines in \mathbb{P}^3 : homogeneous point representation

In the **homogeneous point representation** of a line, given two homogeneous points \mathbf{A} and \mathbf{B} on the line, we write

$$W = \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix},$$

in which case:

- The **span**⁵ of W^T is the **pencil of points**⁶ $\lambda\mathbf{A} + \mu\mathbf{B}$ on the line.⁷
- The span of the 2-dimensional right **null-space** of W is the **pencil of planes** with the line as the axis.

⁵The set of all linear combinations of the columns of a matrix.

⁶Just a fancy term for the set of points on a line.

⁷To convince yourself of this, consider the inhomogeneous parametric representation of the line $\tilde{\mathbf{A}} + t(\tilde{\mathbf{B}} - \tilde{\mathbf{A}})$ with $t = \frac{\mu}{\lambda + \mu}$.

3D projective geometry

Lines in \mathbb{P}^3 : dual homogeneous plane representation

There is a dual representation of a line as the **intersection of two planes** \mathbf{P} and \mathbf{Q} :

$$\mathbf{W}^* = \begin{bmatrix} \mathbf{P}^T \\ \mathbf{Q}^T \end{bmatrix}$$

in this representation,

- The **span** of \mathbf{W}^{*T} is the **pencil of planes** $\lambda'\mathbf{P} + \mu'\mathbf{Q}$ with the line as an axis.
- The span of the 2-dimensional **null-space** of \mathbf{W}^* is the **pencil of points** on the line.

Example: find \mathbf{W} and \mathbf{W}^* for the points $(1, 0, 0, 1)^T$ and $(2, 0, 0, 1)^T$ and verify that $\mathbf{W}^*\mathbf{W}^T = \mathbf{W}\mathbf{W}^{*T} = 0$.

3D projective geometry

Lines in \mathbb{P}^3 : Plücker matrix representation

We can also represent a line through \mathbf{A} and \mathbf{B} by a 4×4 skew-symmetric⁸ homogeneous matrix \mathbf{L} defined by

$$l_{ij} = A_i B_j - B_i A_j$$

or equivalently,

$$\mathbf{L} = \mathbf{A}\mathbf{B}^T - \mathbf{B}\mathbf{A}^T.$$

\mathbf{L} is the **Plücker** matrix representation of a line.

⁸Recall that a skew-symmetric matrix is a matrix \mathbf{A} such that $\mathbf{A}^T = -\mathbf{A}$.

3D projective geometry

Lines in \mathbb{P}^3 : Plücker matrix representation

Some nice properties of L :

- The rank of L is 2 and its 2-dimensional null-space is spanned by the pencil of planes with the line as an axis ($LW^{*T} = 0$).
- L has 4 degrees of freedom, the same as a line, since it is skew-symmetric (having only 6 independent non-zero elements), homogeneous, and constrained so that $\det L = 0$ (its rank is 2).
- The relation $L = \mathbf{AB}^T - \mathbf{BA}^T$ is the generalization to \mathbb{P}^3 of the relation $\mathbf{l} = \mathbf{x} \times \mathbf{y}$ of \mathbb{P}^2 .
- L is independent of the points \mathbf{A} and \mathbf{B} .
- Under the point transformation $\mathbf{X}' = \mathbf{HX}$, the matrix L transforms as $L' = \mathbf{HLH}^T$.

Example: write the line through $(1, 0, 0, 1)$ and $(2, 0, 0, 1)$ as a Plücker matrix.

3D projective geometry

Lines in \mathbb{P}^3 : dual Plücker matrix representation

A **dual** Plücker representation can be obtained using the **intersection of two planes** \mathbf{P} and \mathbf{Q} :

$$\mathbf{L}^* = \mathbf{P}\mathbf{Q}^T - \mathbf{Q}\mathbf{P}^T$$

\mathbf{L}^* can be derived directly from \mathbf{L} by a simple rewrite rule (see text).

3D projective geometry

Lines in \mathbb{P}^3 : dual Plücker matrix representation

The advantage of the Plücker matrix and dual Plücker matrix representations is that **joins** and **indicence** properties are easily represented:

- The **plane** defined by the **join of a point \mathbf{X} and a line L** is $\pi = L^*\mathbf{X}$, and $L^*\mathbf{X} = \mathbf{0}$ iff \mathbf{X} is on L .
- The **point** defined by the **intersection of a line L and a plane π** is $\mathbf{X} = L\pi$, and $L\pi = \mathbf{0}$ iff L is on π .

Example: find the intersection of the line through $(1, 0, 0, 1)$ and $(2, 0, 0, 1)$ with the plane $X = 1$ (note that the plane $X = 1$ is represented as $(1, 0, 0, -1)^T$).

3D projective geometry

Lines in \mathbb{P}^3 : Plücker line coordinates

A line can also be represented by the 6 independent non-zero elements of the skew-symmetric Plücker matrix L .

This representation is called the Plücker line coordinate representation of a line.

See the text for the properties of this representation.

3D projective geometry

Quadrics and dual quadrics

A **quadric** is the 3D analog of a conic, defined by

$$\mathbf{X}^T \mathbf{Q} \mathbf{X} = 0$$

where \mathbf{Q} is a symmetric 4×4 matrix.

Here are some properties of quadrics:

- 9 degrees of freedom (10 independent elements, one lost to scale)
- Defined by 9 points in general position
- If \mathbf{Q} is singular, the quadric is **degenerate** and can be described by fewer points
- $\pi = \mathbf{Q}\mathbf{X}$ is the **polar plane** of \mathbf{X} with respect to \mathbf{Q} . If \mathbf{X} is outside \mathbf{Q} then π is defined by the points of contact of the rays through \mathbf{X} tangent to \mathbf{Q} ; if \mathbf{X} is on \mathbf{Q} then π is the tangent plane to \mathbf{Q} at \mathbf{X} .
- The intersection of a plane π and a quadric \mathbf{Q} is always a conic \mathbf{C} .
- Given a 3D homography $\mathbf{X}' = \mathbf{H}\mathbf{X}$, the quadric \mathbf{Q} transforms as

$$\mathbf{Q}' = \mathbf{H}^{-T} \mathbf{Q} \mathbf{H}^{-1}$$

3D projective geometry

Quadrics and dual quadrics

The dual of a point quadric is an equation on planes: the tangent planes π to point quadric Q satisfying $\pi^T Q^* \pi = 0$.

Here $Q^* = Q^{-1}$ if Q is invertible, or $Q^* = \text{adjoint } Q$ otherwise.⁹

Dual quadrics transform as

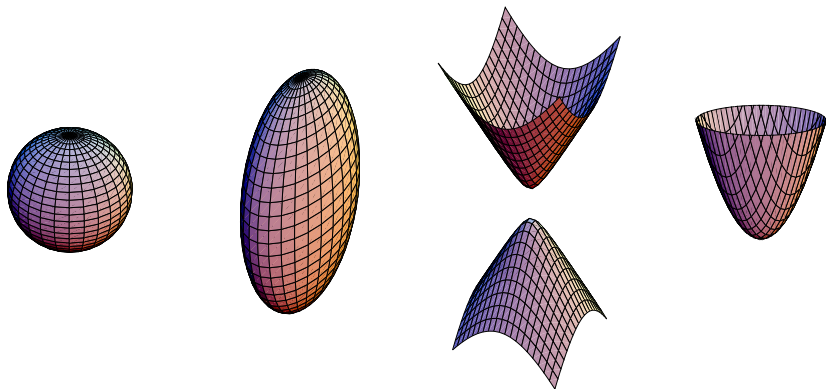
$$Q^{*'} = H Q^* H^T.$$

⁹For the curious, the adjoint of matrix M is defined as the transpose of the matrix where element ij is $(-1)^{i+j} \det \hat{M}_{ij}$ where \hat{M}_{ij} is obtained from M by striking out the i -th row and j -th column.

3D projective geometry

Types of quadrics

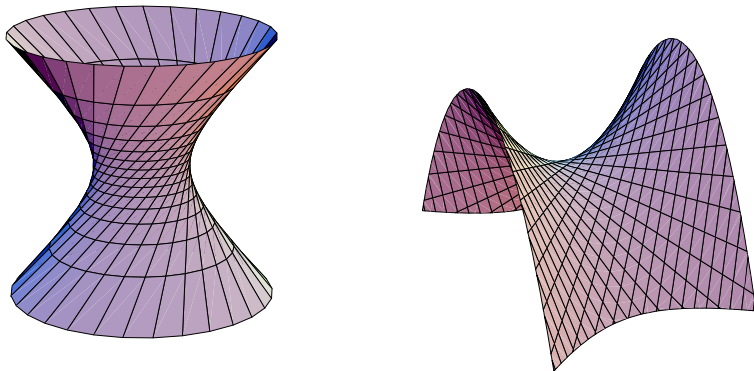
Depending on the rank and signs of the singular values of Q , we obtain a variety of different topologies:



Quadrics projectively equivalent to a sphere (ellipsoid, hyperboloid of two sheets, paraboloid), also called **non-ruled quadrics**, Hartley and Zisserman (2004), Fig. 3.2

3D projective geometry

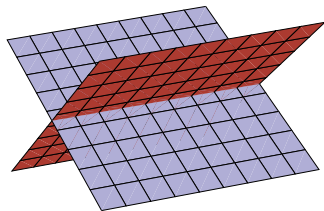
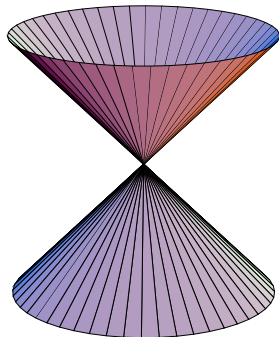
Types of quadrics



Hyperboloids of one sheet, also called **ruled quadrics**, Hartley and Zisserman (2004), Fig. 3.3

3D projective geometry

Types of quadrics



Degenerate quadrics, Hartley and Zisserman (2004), Fig. 3.4

3D projective geometry

Twisted cubics

Conics in 2D can be expressed parametrically in \mathbb{P}^2 as

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = A \begin{pmatrix} 1 \\ \theta \\ \theta^2 \end{pmatrix} \quad (1)$$

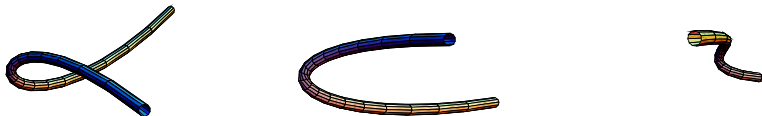
By analogy, various kinds of space curves called **twisted cubics** can be expressed parametrically in \mathbb{P}^3 as

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = A \begin{pmatrix} 1 \\ \theta \\ \theta^2 \\ \theta^3 \end{pmatrix} \quad (2)$$

3D projective geometry

Twisted cubics

Some views of a twisted cubic:

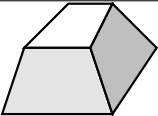
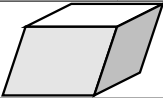

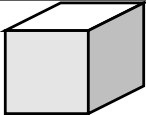


Hartley and Zisserman (2004), Fig. 3.5

3D projective geometry

Hierarchy of transforms

As we saw in 2D, 3D homographies form a hierarchy from most general to most constrained:

Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity π_∞ .
Similarity 7 dof	$\begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$		The absolute conic Ω_∞ .
Euclidean 6 dof	$\begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$		Volume

Hartley and Zisserman (2004), Table 3.1

3D projective geometry

The plane at infinity

Recall that in planar projective geometry the **line at infinity** l_∞ contained the intersections of parallel lines.

In the projective geometry of 3-space, the corresponding object is the **plane at infinity** π_∞ .

The **canonical position** of the plane at infinity is $\pi_\infty = (0, 0, 0, 1)^T$ in affine 3-space.

Properties of the plane at infinity:

- Two planes are parallel iff their line of intersection is on π_∞ .
- Two lines are parallel iff their point of intersection is on π_∞ .

3D projective geometry

The plane at infinity: motivation

Why do we care about π_∞ ?

- Just like l_∞ in \mathbb{P}^2 , π_∞ in \mathbb{P}^3 is **fixed** under **affine** transformations but **moved** under **projective** transformations.
- When we work on 3D **reconstruction** from multiple views, we'll get a **projective reconstruction** then we'll need to find a transformation H giving us a **Euclidean reconstruction**.
- One way to do this is to first transform from a projective frame to an affine frame, then from affine to Euclidean.
- In **planar** geometry, if we could find l_∞ in the image then apply H_p mapping l_∞ to $(0, 0, 1)$, we could also use H_p to transform our projective reconstruction into a **2D affine frame**.
- Similarly, in 3-space, if we can find π_∞ (e.g. using parallel lines and planes in an image), we could find a transform H_p mapping the observed π_∞ to $(0, 0, 0, 1)$. H_p would map our reconstruction into a **3D affine frame**.

3D projective geometry

The absolute conic

The **absolute conic** Ω_∞ is a conic on π_∞ .

As π_∞ is fixed under affine transformations, the absolute conic is **fixed under similarity transforms**.

In a metric frame, $\Omega_\infty = \mathbf{I}$.

Informally we can say that Ω_∞ records **non-metric distortions** we've applied in 3-space.

In principle, if we could find Ω_∞ in an affine frame (by e.g. comparing observed to known angles), we could apply an affine transform H_a mapping Ω_∞ to \mathbf{I} . This would also undo the affine distortions to the scene.

3D projective geometry

The absolute conic

The dual of the absolute conic is a degenerate quadric called the **dual absolute quadric** Q_{∞}^* .

The dual absolute conic is **fixed under similarity transforms** and can be identified directly in a projective frame by observing angles between planes.

Outline

- 1 Introduction
- 2 2D projective geometry
- 3 3D projective geometry
- 4 Rigid (Euclidean) transformations**

Rigid (Euclidean) transformations

Introduction

Rigid or Euclidean transformations involve only rotations and translations.

We usually think of rigid transformations as **transforms between different coordinate systems** in \mathbb{R}^3 .

The vector representation of a point $\mathbf{X} = (X, Y, Z)^T$ should be understood as $\mathbf{X} = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$ where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are an orthonormal right-handed basis for \mathbb{R}^3 with respect to some origin \mathbf{O} .

Suppose we have two coordinate systems or frames $A = (\mathbf{O}_A, \mathbf{i}_A, \mathbf{j}_A, \mathbf{k}_A)$ and $B = (\mathbf{O}_B, \mathbf{i}_B, \mathbf{j}_B, \mathbf{k}_B)$.

Problem: given a point \mathbf{X} in frame A , what is that same point represented in frame B ?

Rigid (Euclidean) transformations

Rotation and translation

Any rigid transform can be decomposed into a rotation and translation:

$$\mathbf{X}' = \mathbf{R}_{A/B}\mathbf{X} + \mathbf{O}_{A/B}$$

where $\mathbf{R}_{A/B}$ is a **rotation matrix** rotating points from frame A to frame B and $\mathbf{O}_{A/B}$ is the representation of the origin of frame A in frame B .

If \mathbf{X} and \mathbf{X}' are represented in homogeneous coordinates, we write

$$\mathbf{X}' = \mathbf{H}\mathbf{X} = \begin{bmatrix} \mathbf{R}_{A/B} & \mathbf{O}_{A/B} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}.$$

Rigid (Euclidean) transformations

Rotation matrices

The rotation $R_{A/B}$ of a point from frame $A = (\mathbf{O}_A, \mathbf{i}_A, \mathbf{j}_A, \mathbf{k}_A)$ to frame $B = (\mathbf{O}_B, \mathbf{i}_B, \mathbf{j}_B, \mathbf{k}_B)$ can be written

$$R_{A/B} = \begin{bmatrix} \mathbf{i}_A \cdot \mathbf{i}_B & \mathbf{j}_A \cdot \mathbf{i}_B & \mathbf{k}_A \cdot \mathbf{i}_B \\ \mathbf{i}_A \cdot \mathbf{j}_B & \mathbf{j}_A \cdot \mathbf{j}_B & \mathbf{k}_A \cdot \mathbf{j}_B \\ \mathbf{i}_A \cdot \mathbf{k}_B & \mathbf{j}_A \cdot \mathbf{k}_B & \mathbf{k}_A \cdot \mathbf{k}_B \end{bmatrix} \quad (3)$$

The **columns** of R are the projections of the basis vectors for A onto the basis vectors for B .

Rigid (Euclidean) transformations

Rotation matrices

Some important properties of rotation matrices:

- A rotation matrix is **orthogonal**.
- Any rotation matrix can be decomposed into the **product of 3 simple rotations**, i.e., around **\mathbf{i}** , **\mathbf{j}** , and **\mathbf{k}** .
- The **inverse** of a rotation matrix is its **transpose**.
- The **determinant** of a rotation matrix is 1.
- The **columns** of a rotation matrix form a right-handed coordinate system.
- The **rows** of a rotation matrix form a right-handed coordinate system.

Rigid (Euclidean) transformations

Simple rotation matrices

Here are the simple rotation matrices:

Rotation of α around \mathbf{i} : Rotation of β around \mathbf{j} : Rotation of γ around \mathbf{k} :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

$$\begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$\begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

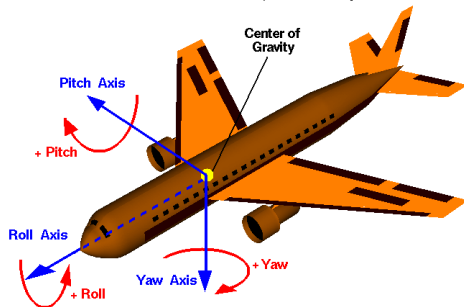
Rigid (Euclidean) transformations

Simple rotation matrices

Note that in **aerial robotics**, particular conventions are used for the local coordinate system.

The X axis is the forward direction of the aircraft, Y is right, and Z is down.

Rotations are specified in terms of roll ϕ then pitch θ then yaw ψ .



<http://en.wikipedia.org/wiki/File:Rollpitchyawplain.png>

Rigid (Euclidean) transformations

Simple rotation matrices

Roll ϕ is a rotation around X , pitch θ is a rotation around Y , and yaw ψ is a rotation around Z :

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_Z(\psi)\mathbf{R}_Y(\theta)\mathbf{R}_X(\phi) \\ &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \theta & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \theta & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned}$$

Source: Sturm et al., TUMx: AUTONAVx Autonomous Navigation for Flying Robots, <http://courses.edx.org>. Note error in the video version.

Rigid (Euclidean) transformations

Alternative representations of rotation

Another representation of rotations is the **axis/angle** a.k.a. **twist coordinate** representation.

We specify a **rotation axis** \mathbf{n} and a **rotation angle** θ .

The 4 parameter version is not minimal.

For a minimal 3 parameter version: use the length of the vector as the rotation angle.

Rodriguez's formulae:

$$\mathbf{R}(\mathbf{n}, \theta) = \mathbf{I} + \sin \theta [\mathbf{n}]_{\times} + (1 - \cos \theta) [\mathbf{n}]_{\times}^2$$

$$\theta = \cos^{-1} \left(\frac{\text{trace } \mathbf{R} - 1}{2} \right)$$

$$\mathbf{n} = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Rigid (Euclidean) transformations

Example

Let $A = ((0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 0)^T, (0, 0, 1)^T)$ (i.e., the world coordinate frame), and let

$$B = ((2, -1, 1)^T, (0, 1, 0)^T, (-1, 0, 0)^T, (0, 0, 1)^T).$$

Try to write the rotation from frame A to frame B and the origin of frame A , represented in frame B .

Then use that to transform the point $(1, \frac{1}{2}, \frac{1}{2})^T$ from frame A to frame B .

Rigid (Euclidean) transformations

Example

Step 1: compute the rotation matrix from frame A to frame B :

$$R = \begin{bmatrix} \mathbf{i}_A \cdot \mathbf{i}_B & \mathbf{j}_A \cdot \mathbf{i}_B & \mathbf{k}_A \cdot \mathbf{i}_B \\ \mathbf{i}_A \cdot \mathbf{j}_B & \mathbf{j}_A \cdot \mathbf{j}_B & \mathbf{k}_A \cdot \mathbf{j}_B \\ \mathbf{i}_A \cdot \mathbf{k}_B & \mathbf{j}_A \cdot \mathbf{k}_B & \mathbf{k}_A \cdot \mathbf{k}_B \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rigid (Euclidean) transformations

Example

Step 2: compute the origin of frame A in frame B . We compute the difference between the two frames' origins in world coordinates then project the result onto frame B 's basis:

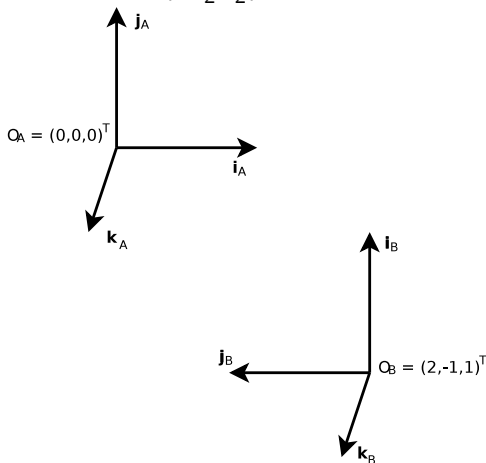
$$[\mathbf{i}_B \quad \mathbf{j}_B \quad \mathbf{k}_B]^T (O_A - O_B) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} -2 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$$

Now we can easily transform from frame A to frame B .

Rigid (Euclidean) transformations

Example

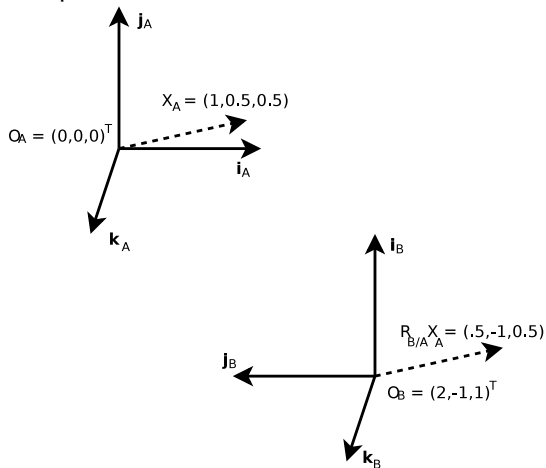
Now let's transform the point $(1, \frac{1}{2}, \frac{1}{2})^T$ in frame A:



Rigid (Euclidean) transformations

Example

First we rotate the point into frame B :



Rigid (Euclidean) transformations

Example

Then we translate by $O_{A/B}$ (the origin of frame A represented in frame B):

