Notes of Mathematics and 3D Geometry

Chao Wang

January 29, 2019

Contents

1 Vector calculus identities **4**
- 1.1 Gradient . . . 4
- 1.2 Divergence . . . 4
- 1.3 Curl . . . 4
- 1.4 Laplacian . . . 5
- 1.5 Gaussian/Divergence Theorem . . . 5

2 Fundamental matrix properties **6**
- 2.1 Conjugate/Hermitian transpose . . . 6
- 2.2 Orthogonal matrix and unitary matrix . . . 6
- 2.3 Matrix trace . . . 7
- 2.4 Symmetric matrix . . . 7
  - 2.4.1 Positive-definite symmetric matrix . . . 7
  - 2.4.2 Skew-symmetric matrix . . . 8
- 2.5 Matrix decomposition . . . 8
  - 2.5.1 QR and RQ decomposition . . . 8
  - 2.5.2 Singular value decomposition (SVD) . . . 8
  - 2.5.3 Eigenvalue decomposition . . . 9
  - 2.5.4 Cholesky decomposition . . . 9

3 Rotation matrix **10**
- 3.1 Conversion between different forms of a rotation . . . 10
  - 3.1.1 From rotation matrix to angle and axis . . . 10
  - 3.1.2 From rotation angle and axis to rotation matrix . . . 11
- 3.2 Quaternion and spatial rotation . . . 11
  - 3.2.1 Definition . . . 12
  - 3.2.2 Features and properties . . . 12
  - 3.2.3 Comparison between quaternion and common rotation matrix . . . 13
- 3.3 Skew-symmetric matrix and rotation . . . 13
  - 3.3.1 Eigenvalue decomposition . . . 13
  - 3.3.2 3-vector's skew-symmetric form . . . 13
  - 3.3.3 Relationship between skew-symmetric matrix and rotation . . . 14
  - 3.3.4 Geometric interpretation of 3-by-3 skew-symmetric rotation . . . 15
  - 3.3.5 Specific form of skew-symmetric rotation . . . 16

4 Linear Least-squares minimization **17**

5 Non-linear minimization: iterative estimation **18**
- 5.1 Newton iteration . . . 18
- 5.2 Weighted iteration . . . 19
- 5.3 Gauss-Newton method . . . 19

5.4 Gradient descent method . . . 20
5.5 Levenberg-Marquardt method . . . 20

6 2D projective geometry and transformation 21
6.1 Line and point . . . 21
6.1.1 Definition and properties . . . 21
6.1.2 Intersection . . . 21
6.1.3 Points at infinity and line at infinity . . . 22
6.2 Projective transformations . . . 22
6.2.1 Projective transformation between planes . . . 23
6.3 Conics . . . 24
6.3.1 Duality and dual conics . . . 26
6.3.2 Transformed conic . . . 26
6.3.3 The pole−polar relationship . . . 26
6.4 A hierarchy of projective transformations . . . 27
6.4.1 Isometries . . . 27
6.4.2 Similarity transformations . . . 28
6.4.3 Affine transformations . . . 28
6.4.4 Projective transformations . . . 29
6.5 Remove projective distortion from a perspective image of a global plane . . . 29
6.5.1 Remove entire projective distortion . . . 31
6.5.2 Affine rectification . . . 31
6.5.3 Metric rectification . . . 34

7 3D projective geometry and transformation 39
7.1 3D Points and planes . . . 39
7.2 Quadrics . . . 39
7.3 The hierarchy of 3D Projective transformation . . . 40
7.4 Remove 3D projective distortions . . . 40
7.4.1 Plane at infinity . . . 40
7.4.2 Absolute conic . . . 41
7.4.3 Absolute dual conic . . . 42

8 References 43

1 Vector calculus identities

Reference: `https://en.wikipedia.org/wiki/Vector_calculus_identities`

1.1 Gradient

The gradient (or gradient vector field) of a scalar function $f(x_1, x_2, \cdots, x_n)$ is denoted as ∇f, where ∇ (called the **nabla** symbol) denotes the vector differential operator, del. **a gradient's input is a scalar and its output is a vector**.

In the three-dimensional Cartesian coordinate system, the gradient of some function $f(x, y, z)$ is a vector given by:

$$grad(f) = \nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k} \tag{1.1}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the standard unit vectors.

1.2 Divergence

Divergence is a vector operator that produces a scalar field, giving the quantity of a vector field's source at each point.**A divergence's input is a vector and its output is a scalar**.

In three-dimensional Cartesian coordinates, the divergence of a continuously differentiable vector field $\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k}$ is defined as the scalar-valued function:

$$div\mathbf{F} = \nabla \cdot F = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}) \cdot (F_x, F_y, F_z) = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \tag{1.2}$$

So the divergence is actually the sum of partial derivative of all directions. Note that **the divergence operator is $\nabla\cdot$ (with a dot), while the gradient is ∇**.

1.3 Curl

In vector calculus, the curl is a vector operator that describes the **infinitesimal rotation** of a vector field in three-dimensional Euclidean space. The direction of the curl is the axis of rotation, as determined by the right-hand rule, and the magnitude of the curl is the magnitude of rotation.

In 3-dimensional Cartesian coordinates, the curl of a vector field $\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k}$ is denoted as

a vector field:

$$curl\mathbf{F} = \nabla \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_x & F_y & F_z \end{vmatrix}$$
$$= (\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z})\mathbf{i} + (\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x})\mathbf{j} + (\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y})\mathbf{k}$$

So **both the input and output of a curl is a vector**.

1.4 Laplacian

The Laplace operator is a **second order** differential operator in the n-dimensional Euclidean space, defined as the divergence $(\nabla\cdot)$ of the gradient (∇f). Thus if f is a twice-differentiable real-valued function, then the Laplacian of f is defined by

$$\triangle f = \nabla^2 f = \nabla \cdot \nabla f \tag{1.3}$$

NOTE that here ∇^2 is one operator symbol to denote laplacian instead of the power 2 of gradient. Laplacian operator can be denoted as $\triangle$ or $\nabla \cdot \nabla$ or ∇^2. So **both the input and output of laplacian is a scalar**.

1.5 Gaussian/Divergence Theorem

2 Fundamental matrix properties

2.1 Conjugate/Hermitian transpose

The **conjugate transpose** or **Hermitian transpose** of a $m \times n$ matrix $\mathbf{A}$ is denoted as $\mathbf{A}^*$ whose element is

$$(\mathbf{A}^*)_{ij} = \overline{\mathbf{A}_{ji}}, \tag{2.1}$$

where $\overline{\mathbf{A}_{ji}}$ is the complex conjugate (like $a+bi$'s conjugate is $a-bi$). Sometimes the conjugate transpose can also be represented as $\overline{\mathbf{A}^\top}$.

A conjugate/Hermitian transpose can be understood as the complex version of a real matrix transpose.

A complex square matrix is **Hermitian** if it is equal to its Hermitian transpose. That is, $\mathbf{A} = \mathbf{A}^*$. The Hermitian matrix is the complex version of a real symmetric matrix.

2.2 Orthogonal matrix and unitary matrix

A real square matrix $\mathbf{U}$ is **orthogonal** if $\mathbf{U}^\top\mathbf{U} = \mathbf{U}\mathbf{U}^\top = \mathbf{I}$. This means the row vectors (and column vectors) are all of unit norm, and its determinant is 1. The latter means that, for any orthogonal matrix $\mathbf{U}_n$, its rows/columns form an orthonormal basis of the Euclidean space $\mathbb{R}^n$ with the ordinary Euclidean dot product.

A complex square matrix $\mathbf{U}$ is **unitary** if its conjugate transpose $\mathbf{U}^*$ is its inverse. That is, $\mathbf{U}^*\mathbf{U} = \mathbf{U}\mathbf{U}^* = \mathbf{I}$. This can be understood as the complex version of an orthogonal matrix.

An orthogonal matrix $\mathbf{A}$, or even any $m \times n$ matrix with n orthogonal **columns**, has the following important features:

- Preserves inner products:
$$\mathbf{A}\mathbf{x} \cdot \mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x})^\top \mathbf{A}\mathbf{x} = \mathbf{x}^\top\mathbf{x}. \tag{2.2}$$

- Preserves norms:
$$||\mathbf{A}\mathbf{x}|| = ||\mathbf{x}||. \tag{2.3}$$
This is the direct result from the last one.

- Preserves distance:
$$||\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y}|| = ||\mathbf{x} - \mathbf{y}||. \tag{2.4}$$
You can prove it easily using squares on both sides.

- Preserves angles:
$$\angle(\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}) = \angle(\mathbf{x}, \mathbf{y}). \tag{2.5}$$
Since $\angle(\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}) = \arccos(\mathbf{A}\mathbf{x} \cdot \mathbf{A}\mathbf{y}/||\mathbf{A}\mathbf{x}||\ ||\mathbf{A}\mathbf{y}||) = \arccos(\mathbf{x} \cdot \mathbf{y}/||\mathbf{x}||\ ||\mathbf{y}||) = \angle(\mathbf{x}, \mathbf{y})$

2.3 Matrix trace

Matrix trace contains some properties:

$$\text{tr}(\mathbf{A}+\mathbf{B}) = \text{tr}(\mathbf{B}+\mathbf{A}) \tag{2.6}$$
$$\text{tr}(c\mathbf{A}) = c \cdot \text{tr}(\mathbf{A}) \tag{2.7}$$
$$\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA}) \tag{2.8}$$
$$\text{tr}(\mathbf{B^{-1}AB}) = \text{tr}(\mathbf{A}) \tag{2.9}$$

Eq. (2.9) is very important and it can be easily achieved by Eq. (2.8). And in Eq. (2.8-2.9), the matrices must satisfy the multiplication rule after commutation.

And there is an important relationship between Frobenius norm and matrix trace:

$$||\mathbf{A}||_F^2 = ||\mathbf{AR}||_F^2 = ||\mathbf{RA}||_F^2 = \text{tr}(\mathbf{A}^\top\mathbf{A}) \tag{2.10}$$

where $\mathbf{R}$ is any rotation matrix with valid dimensions. These are also very intuitive and easily to be checked using the property of orthogonal matrix in Eq. (2.3).

2.4 Symmetric matrix

A matrix is called **symmetric** if $\mathbf{A}^\top = \mathbf{A}$. Here we only deal with **real** symmetric matrix.

A symmetric matrix $\mathbf{A}$ has an important property called **Eigenvalue decomposition** that $\mathbf{A}$ can be decomposed into $\mathbf{A} = \mathbf{UDU}^\top$, where $\mathbf{D}$ is a diagonal matrix composed of its eigenvalues, and $\mathbf{U}$ an orthogonal matrix with eigenvectors. This is described in Section 2.5.3.

2.4.1 Positive-definite symmetric matrix

A symmetric matrix $\mathbf{A}$ is **positive-definite** if $\mathbf{x}^\top\mathbf{Ax} > 0$ for any non-zero vector $\mathbf{x}$. A positive-definite symmetric matrix has two important properties. The first one is:

Theorem 2.1. *Eigenvalues of a symmetric positive-definite matrix are all positive.*

Proof. We already have $\mathbf{A} = \mathbf{UDU}^\top$. It is known that for any orthogonal matrix, its columns form an orthonormal basis of the Euclidean space $\mathbb{R}^n$ with the ordinary Euclidean dot product. Therefore, any non-zero vector $\mathbf{x}$ can be presented linearly by $\mathbf{U}$'s column vectors as $\mathbf{x} = x_1 * \mathbf{u}_1 + x_2 * \mathbf{u}_2 + \cdots + x_n * \mathbf{u}_n$. Then, according to the definition of positive-definite matrix, we have

$$\begin{aligned}
\mathbf{x}^\top\mathbf{Ax} &= \mathbf{x}^\top\mathbf{UDU}^\top\mathbf{x} \\
&= (x_1 * \mathbf{u}_1^\top + \cdots + x_n * \mathbf{u}_n^\top)(\mathbf{u}_1 \cdots \mathbf{u}_n)diag(\lambda_1, \lambda_2, \cdots, \lambda_n)(\mathbf{u}_1 \cdots \mathbf{u}_n)^\top(x_1 * \mathbf{u}_1 + \cdots + x_n * \mathbf{u}_n) \\
&= \begin{bmatrix} x_1 * \mathbf{u}_1^2 & x_2 * \mathbf{u}_2^2 & \cdots & x_n * \mathbf{u}_n^2 \end{bmatrix} diag(\lambda_1, \lambda_2, \cdots, \lambda_n) \begin{bmatrix} x_1 * \mathbf{u}_1^2 & x_2 * \mathbf{u}_2^2 & \cdots & x_n * \mathbf{u}_n^2 \end{bmatrix}^\top \\
&= \lambda_1 x_1^2 + \lambda_2 x_2^2 + \cdots + \lambda_n x_n^2.
\end{aligned}$$

If it is positive for any $\mathbf{x}$, then all $\lambda_i > 0$. □

Another property of positive-definite symmetric matrix is the Cholesky decomposition/factorization, which is described in Section 2.5.4.

2.4.2 Skew-symmetric matrix

Section 3.3 shows details about skew-symmetric matrix and its relationship with rotation matrix.

2.5 Matrix decomposition

2.5.1 QR and RQ decomposition

Any real square matrix $\mathbf{A}$ can be decomposed as $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper-triangular. Similar to QR decomposition, there is RQ decomposition $\mathbf{A} = \mathbf{RQ}$. We can carry out RQ decomposition using the Givens rotations. See Page 579 in Ref. [1] for details.

2.5.2 Singular value decomposition (SVD)

Given any square matrix $\mathbf{A}$, the SVD is a factorization of $\mathbf{A} = \mathbf{UDV}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are unitary (or orthogonal for real matrix values) matrices, and $\mathbf{D}$ is a diagonal matrix with non-negative singular values. The decomposition may be carried out in such a way that the diagonal entries of D are in descending order.

SVD also exists for any non-square matrix. For a an $m \times n$ matrix $\mathbf{A}$, the standard SVD result is $\mathbf{A} = \mathbf{UDV}^\top$, where $\mathbf{U}$ is $m \times m$ unitary matrix (or orthogonal for real matrix values), $\mathbf{D}$ is $m \times n$ diagonal matrix with *non-negative* real diagonal values, and $\mathbf{V}$ is $n \times n$ unitary matrix. Suppose $\mathbf{A}$'s rank is $r <= n$, with only the first r diagonal values are non-negative in $\mathbf{D}$ and all the rest are 0.

Suppose $m >= n$, clearly $\mathbf{U}$ can also be represented as $m \times n$ matrix with orthogonal columns, $\mathbf{D}$ is $n \times n$ diagonal matrix, and $\mathbf{V}$ is $n \times n$ orthogonal matrix. The fact that $\mathbf{U}$'s columns are orthogonal means $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, and $||\mathbf{Ux}|| = ||\mathbf{x}||$ for any vector $\mathbf{x}$ based on Eq. (2.3).

We can easily check that the square of a singular value σ^2 is an eigenvalue of the matrix $\mathbf{A}^\top\mathbf{A}$ by

$$\mathbf{A}^\top\mathbf{A} = \mathbf{VDU}^\top\mathbf{UDV}^\top = \mathbf{VD}^2\mathbf{V}^\top = \mathbf{VD}^2\mathbf{V}^{-1}$$

2.5.3 Eigenvalue decomposition

Theorem 2.2. *A **symmetric** matrix* $\mathbf{A}$ *can be decomposed as* $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$*, where* $\mathbf{U}$ *is orthogonal and* $\mathbf{D}$ *is a diagonal matrix. Actually, diagonal values in* $\mathbf{D}$ *are the eigenvalues, and* $\mathbf{U}$*'s column vectors are corresponding eigenvectors.*

Proof. Ref. [1] doesn't give but cite a paper about the proof. □

2.5.4 Cholesky decomposition

Theorem 2.3. *A **positive-definite symmetric** (or more precisely, Hermitian) matrix* $\mathbf{A}$ *can be **uniquely** decomposed as* $\mathbf{A} = \mathbf{K}\mathbf{K}^\top$ *or* $\mathbf{A} = \mathbf{L}^\top\mathbf{L}$*, where* $\mathbf{K}$ *is an **upper-triangular** real matrix with positive diagonal entries, and* $\mathbf{L}$ *is a **lower-triangular** real matrix with positive diagonal entries.*

Proof.

1) Prove the decomposition of $\mathbf{A} = \mathbf{K}\mathbf{K}^\top$. Easily $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top = \mathbf{U}\mathbf{E}\mathbf{E}^\top\mathbf{U}^\top = \mathbf{V}\mathbf{V}^\top$, where $\mathbf{E}$ is the square root of each values in $\mathbf{D}$, and $\mathbf{V} = \mathbf{U}\mathbf{E}$. However, now $\mathbf{V}$ is still not upper-triangular. Use RQ decomposition described in Section 2.5.1, we get $\mathbf{V} = \mathbf{K}\mathbf{Q}$, and then $\mathbf{A} = \mathbf{K}\mathbf{Q}\mathbf{Q}^\top\mathbf{K}^\top = \mathbf{K}\mathbf{K}^\top$, where $\mathbf{K}$ is the upper-triangular matrix.

The proof of $\mathbf{A} = \mathbf{L}^\top\mathbf{L}$ is very similar to the above procedure. Just use $\mathbf{A} = \mathbf{U}^\top\mathbf{D}\mathbf{U}$ and also use QR decomposition on $\mathbf{V}$.

2) Prove the uniqueness of the decomposition. See Pages 583 in Ref. [1] for details. □

Cholesky factorization is usually used in solving equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b}$, since it is easy to check that $\mathbf{A}^\top\mathbf{A}$ is a symmetric positive-definite matrix through definition.

3 Rotation matrix

Here is an important theorem (or, can be treated as an axiom) that can be used to determine if a square real matrix is a rotation matrix or not:

Theorem 3.1. *A square matrix* $\mathbf{R}$ *is a rotation matrix if and only if* $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ *and* $\det(\mathbf{R}) = 1$.

The set of all rotation matrices of size n forms a group known as the special orthogonal group $SO(n)$, one example of which is the rotation group $SO(3)$.

3.1 Conversion between different forms of a rotation

3.1.1 From rotation matrix to angle and axis

According to Euler's rotation theorem, in 3-dimensional space, any rotation or sequence of rotations of a rigid body or coordinate system about a fixed point is equivalent to a single rotation by a given angle θ about a fixed axis (called the Euler axis) that runs through the fixed point. The Euler axis is typically represented by a unit vector $\mathbf{u}$.

Given any 3-by-3 rotation matrix $\mathbf{R}$, we can compute the axis this rotation rotates about, and the angle of this rotation by the method in [2].

Computation of rotation axis. A vector $\mathbf{u}$ parallel to the axis must satisfy

$$\mathbf{R}\mathbf{u} = \mathbf{u}, \tag{3.1}$$

since the rotation around the axis must result in itself. The above equation can be solved for $\mathbf{u}$ up to a scale factor unless $\mathbf{R} = \mathbf{I}$.

From Eq. (3.1) we can see that, $\mathbf{u}$ is the eigenvector of $\mathbf{R}$ corresponding to the eigenvalue $\lambda = 1$. Actually, every rotation matrix must have this eigenvalue, the other two eigenvalues being complex conjugates of each other. It follows that a general rotation matrix in three dimensions has, up to a multiplicative constant, only one real eigenvector.

We also have another way related to skew-symmetric matrix to determine $\mathbf{u}$. Firstly, Eq. (3.1) can be written as $(\mathbf{R}-\mathbf{I})\mathbf{u} = 0$. That is, $\mathbf{u}$ lies in the null space of $(\mathbf{R}-\mathbf{I})$. Therefore,

$$\begin{aligned} 0 &= \mathbf{R}^\top 0 + 0 \\ &= \mathbf{R}^\top(\mathbf{R}-\mathbf{I})\mathbf{u} + (\mathbf{R}-\mathbf{I})\mathbf{u} \\ &= (\mathbf{R}^\top\mathbf{R} - \mathbf{R}^\top + \mathbf{R} - \mathbf{I})\mathbf{u} \\ &= (\mathbf{R} - \mathbf{R}^\top)\mathbf{u} \end{aligned}$$

Note that $\mathbf{R}-\mathbf{R}^\top$ is skew-symmetric, and we know that any skew-symmetric matrix satisfies $[\mathbf{t}]_\times \mathbf{t} = 0$. So, we can choose $\mathbf{u}$ such that $[\mathbf{u}]_\times = \mathbf{R}-\mathbf{R}^\top$. That is, if

$$\mathbf{R} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix},$$

then

$$\mathbf{u} = \begin{bmatrix} h-f \\ c-g \\ d-b \end{bmatrix},$$

Computation of rotation angle. To find the angle of a rotation, once the axis of the rotation $\mathbf{u}$ is known, select a vector $\mathbf{v}$ perpendicular to the axis. Then the angle of the rotation θ is the angle between $\mathbf{v}$ and $\mathbf{Rv}$.

A more direct method, however, is to simply calculate the trace, i.e., the sum of the diagonal elements of the rotation matrix. Care should be taken to select the right sign for the angle to match the chosen axis:

$$\mathrm{tr}(\mathbf{R}) = 1 + 2\cos\theta \tag{3.2}$$

3.1.2 From rotation angle and axis to rotation matrix

Given an angle θ and an **unit** axis $\mathbf{u} = (u_x, u_y, u_z)^\top$, we can recover the rotation matrix that rotates around $\mathbf{u}$ with angle θ as

$$\mathbf{R} = \begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_x u_y(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_x u_z(1-\cos\theta) - u_y\sin\theta & u_y u_z(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix}. \tag{3.3}$$

Or, it can be written more concisely using the skew-symmetric form as shown in Eq. (3.16). See Section 3.3 for more details about skew-symmetric matrix.

3.2 Quaternion and spatial rotation

Unit quaternions, also known as versors, provide a convenient mathematical notation for representing orientations and rotations of objects in three dimensions. Compared to Euler angles, they are simpler to compose and avoid the problem of *gimbal lock*. Compared to rotation matrices, they are more compact, more numerically stable, and may be more efficient. When used to represent rotation, unit quaternions are also called **rotation quaternions**.

3.2.1 Definition

A Euclidean vector $\mathbf{a} = (a_x, a_y, a_z)$ can be rewritten as $\mathbf{a} = a_x i + a_y j + a_z k$, where i, j, k are unit vectors representing the three Cartesian axes. A rotation through an angle of θ around the axis defined by a **unit** vector $\mathbf{u} = (u_x, u_y, u_z) = u_x i + u_y j + u_z k$ can be represented by a quaternion. This can be done using an extension of Euler's formula:

$$\mathbf{q} = e^{\frac{\theta}{2}(u_x i + u_y j + u_z k)} = \cos\frac{\theta}{2} + (u_x i + u_y j + u_z k)\sin\frac{\theta}{2} = \cos\frac{\theta}{2} + \mathbf{u}\sin\frac{\theta}{2} \tag{3.4}$$

We can use four elements $\mathbf{q} = (s, x, y, z) = s + xi + yj + zk$ to denote quaternion. From Eq. (3.4), we can easily find that it must satisfy $s^2 + x^2 + y^2 + z^2 = 1$. Given $\mathbf{q}$, we can easily recover the angle θ and axis $\mathbf{u}$ this rotation is about.

Actually, a quaternion doesn't need to be unit, but only unit quaternion can be used as rotation quaternion, and it also has many convenient features and properties.

3.2.2 Features and properties

A rotation quaternion $\mathbf{q} = (s, x, y, z) = s + xi + yj + zk$ has many important properties:

1. **Inverse quaternion**. A quaternion $\mathbf{q}$'s inverse rotation matrix in quaternion is

   $$\mathbf{q}^{-1} = \overline{\mathbf{q}}/||\mathbf{q}||^2 = \overline{\mathbf{q}}, \tag{3.5}$$

   where $\overline{\mathbf{q}} = s - xi - yj - zk$ is the conjugate quaternion.

2. **Apply rotation on a point.** Given an arbitrary vector $\mathbf{p} = (p_x, p_y, p_z) = p_x i + p_y j + p_z k$, we can easily apply the rotation quaternion $\mathbf{q}$ on the vector and get the transformed one by

   $$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1} \tag{3.6}$$

3. **Quaternion multiplication**. Two rotation quaternions can be combined into one equivalent quaternion by the relation:

   $$\mathbf{q} = \mathbf{q}_1\mathbf{q}_2$$

   However, note that quaternion multiplication is not commutative. That is, $\mathbf{q}_1\mathbf{q}_2 \neq \mathbf{q}_2\mathbf{q}_1$.

4. **Relationship between quaternion and rotation matrix**. Note that $-\mathbf{q} = \mathbf{q}$ represents the same rotation. Other than this, the relationship between rotations and quaternions is unique. The conversion between quaternion and rotation matrix can be found in Ref. [3].

3.2.3 Comparison between quaternion and common rotation matrix

The representation of a rotation as a quaternion (4 numbers) is more compact than the representation as an orthogonal matrix (9 numbers). Furthermore, for a given axis and angle, one can easily construct the corresponding quaternion, and conversely, for a given quaternion one can easily read off the axis and the angle. Both of these are much harder with matrices or Euler angles.

Meanwhile, quaternion can also reduce error compared to Euler angles. When composing several rotations using Euler angles on a computer, rounding errors necessarily accumulate. A quaternion that is slightly off still represents a rotation after being normalised: a matrix that is slightly off may not be orthogonal any more and is harder to convert back to a proper orthogonal matrix.

Quaternions also avoid a phenomenon called *gimbal lock* which can result when, for example in pitch/yaw/roll rotational systems, the pitch is rotated 90° up or down, so that yaw and roll then correspond to the same motion, and a degree of freedom of rotation is lost. In a gimbal-based aerospace inertial navigation system, for instance, this could have disastrous results if the aircraft is in a steep dive or ascent. A very good video of gimbal lock problem is shown in Ref. [4].

3.3 Skew-symmetric matrix and rotation

A matrix is called *skew-symmetric* if $\mathbf{A}^\top = -\mathbf{A}$. Clearly, all diagonal values are 0.

3.3.1 Eigenvalue decomposition

Theorem 3.2. *A skew-symmetric matrix* $\mathbf{S}$ *can be decomposed as* $\mathbf{S} = \mathbf{U}\mathbf{B}\mathbf{U}^\top$ *where* $\mathbf{U}$ *is orthogonal, and* $\mathbf{B}$ *is a* ***block-diagonal*** *matrix of the form* $\mathbf{B} = diag(a_1\mathbf{Z}, a_2\mathbf{Z}, \ldots, a_m\mathbf{Z}, 0, \ldots, 0)$ *where*

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Proof. Ref. [1] doesn't give but cited a paper about the proof. □

Theorem 3.2 is similar to the Eigenvalue decomposition of a symmetric matrix shown in Theorem 2.2.

3.3.2 3-vector's skew-symmetric form

If a vector $\mathbf{a} = (a_1, a_2, a_3)^\top$ is a 3-vector, then we can define its skew-symmetric matrix as follows:

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.7}$$

(Note for the order of vector's 3 values appearing in the matrix.)

That is, **any** skew-symmetric matrix can be written in the form of $[\mathbf{a}]_\times$ for a suitable 3-vector $\mathbf{a}$.

Some features:

- Any vector $\mathbf{a}$ is $[\mathbf{a}]_\times$'s null-vector, since $[\mathbf{a}]_\times \mathbf{a} = \mathbf{a}^\top [\mathbf{a}]_\times = \mathbf{0}$. Therefore, a 3×3 skew-symmetric matrix is defined up to scale by its null-vector.
- The **cross product** of two 3-vectors $\mathbf{a} \times \mathbf{b}$ is $(a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1)^\top$, which can be related to skew-symmetric matrix as

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = (\mathbf{a}^\top [\mathbf{b}]_\times)^\top \tag{3.8}$$

3.3.3 Relationship between skew-symmetric matrix and rotation

Theorem 3.3. ***Any*** *3D rotation matrix can be written as a natural base exponential of a skew-symmetric matrix $e^{[\mathbf{t}]_\times}$, where $\mathbf{t}$ is an arbitrary n-vector.*

Proof. We only need to prove that every $e^{[\mathbf{t}]_\times}$ is a rotation matrix.

Given an arbitrary matrix $\mathbf{T}$, its natural base exponential expression is

$$e^{\mathbf{T}} = \mathbf{I} + \mathbf{T} + \mathbf{T}^2/2! + \cdots + \mathbf{T}^k/k! + \cdots \tag{3.9}$$

We know that a skew-symmetric matrix $\mathbf{S} = \mathbf{U}\mathbf{B}\mathbf{U}^\top$. So

$$\begin{aligned} e^{\mathbf{S}} &= \mathbf{I} + (\mathbf{U}\mathbf{B}\mathbf{U}^\top) + (\mathbf{U}\mathbf{B}\mathbf{U}^\top)^2/2! + \cdots + (\mathbf{U}\mathbf{B}\mathbf{U}^\top)^k/k! + \cdots \\ &= \mathbf{I} + (\mathbf{U}\mathbf{B}\mathbf{U}^\top) + (\mathbf{U}\mathbf{B}^2\mathbf{U}^\top)/2! + \cdots + (\mathbf{U}\mathbf{B}^k\mathbf{U}^\top)/k! + \cdots \\ &= \mathbf{U}e^{\mathbf{B}}\mathbf{U}^\top \end{aligned} \tag{3.10}$$

Here $\mathbf{U}$ is orthogonal, and $\mathbf{B}$ is a **block-diagonal** matrix of the form $\mathbf{B} = diag(a_1\mathbf{Z}, \ldots, a_m\mathbf{Z}, 0, \ldots, 0)$ where

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

We can easiy observe the powers $\mathbf{Z}$ are $\mathbf{Z}^2 = -\mathbf{I}$, $\mathbf{Z}^3 = -\mathbf{Z}$ and $\mathbf{Z}^4 = \mathbf{I}$. Therefore,

$$e^{\mathbf{Z}} = \mathbf{I} + \mathbf{Z} - \mathbf{I}^2/2! + \mathbf{Z}/3! + \cdots = \cos(1)\mathbf{I} + \sin(1)\mathbf{Z} = \mathbf{R}_{2\times 2}(1),$$

where $\mathbf{R}_{2\times 2}(1)$ means the 2×2 matrix representing a rotation through 1 radian. More generally,

$$e^{a\mathbf{Z}} = \cos(a)\mathbf{I} + \sin(a)\mathbf{Z} = \mathbf{R}_{2\times 2}(a).$$

Substituting it into Eq. (3.10) and we obtain

$$e^{\mathbf{S}} = \mathbf{U}e^{\mathbf{B}}\mathbf{U}^\top = \mathbf{U} \cdot diag(\mathbf{R}(a_1), \mathbf{R}(a_2), \cdots, \mathbf{R}(a_m), 1, \cdots, 1) \cdot \mathbf{U}^\top. \tag{3.11}$$

We can easily prove that $e^{\mathbf{S}}$ in Eq. (3.11) is a rotation matrix through this **important feature of a rotation matrix**: a square matrix $\mathbf{R}$ is a rotation matrix if and only if $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ and $\det(\mathbf{R}) = 1$.

Firstly,

$$e^{\mathbf{S}}(e^{\mathbf{S}})^\top = e^{\mathbf{B}}\mathbf{U}^\top = \mathbf{U} \cdot diag(\mathbf{R}(a_1)\mathbf{R}(a_1)^\top, \cdots, \mathbf{R}(a_m)\mathbf{R}(a_m)^\top, 1, \cdots, 1) \cdot \mathbf{U}^\top = \mathbf{I},$$

since $\mathbf{R}(a_i)\mathbf{R}(a_i)^\top = \mathbf{I}$ clearly.

Secondly,

$$\det(e^{\mathbf{S}}) = \det(\mathbf{U})\det(e^{\mathbf{S}})\det(\mathbf{U}^\top) = \det(e^{\mathbf{B}}) = \det(diag(\mathbf{R}(a_1), \cdots, \mathbf{R}(a_m), 1, \cdots, 1)) = 1,$$

since $\det(U) = 1$ or -1, and the determinant rule of a block matrix is

$$\det \begin{bmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} = \det(\mathbf{M})\det(\mathbf{N}) = \det \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{P} & \mathbf{N} \end{bmatrix}$$

Now we have proved that $e^{\mathbf{S}}$ is a rotation matrix for any skew-symmetric matrix $\mathbf{S} = \exp([\mathbf{t}]_\times)$. □

3.3.4 Geometric interpretation of 3-by-3 skew-symmetric rotation

Theorem 3.4. *The* 3×3 *matrix* $e^{[\mathbf{t}]_\times}$ *is a rotation matrix presenting a rotation through an angle* $||\mathbf{t}||$ *about/around the axis presented by the vector* $\mathbf{t}$.

This representation of a rotation is called the **angle-axis representation**.

Proof. From Theorem 3.2 we know that $[\mathbf{t}]_\times = \mathbf{U}diag(a\mathbf{Z}, 0)\mathbf{U}^\top$. Then, by matching the Frobenius norms of the matrices on both sides we can get:

$$2||\mathbf{t}||^2 = \text{tr}(diag(a\mathbf{Z}, 0)^\top \cdot diag(a\mathbf{Z}, 0)) = 2a^2.$$

That is, $a = ||\mathbf{t}||$. This is achieved based on Eq. (2.10).

Thus, from Eq. (3.11) we can get the rotation matrix

$$e^{[\mathbf{t}]_\times} = \mathbf{U}diag(\mathbf{R}(||\mathbf{t}||), 1)\mathbf{U}^\top \tag{3.12}$$

Now we can determine the angle and axis that rotation matrix in Eq. (3.12) presents based on the method described in Section 3.1.1.

First prove the axis is $\mathbf{t}$. It is very easy to check that the 3rd eigenvector $\mathbf{u}_3$ of orthogonal matrix $\mathbf{U}$ is an axis of the rotation matrix $e^{[\mathbf{t}]_\times}$ via Eq. (3.1). Moreover, we know that $[\mathbf{t}]_\times \mathbf{u}_3 = \mathbf{U}diag(a\mathbf{Z}, 0)\mathbf{U}^\top \mathbf{u}_3 = \mathbf{0}$. That is, $\mathbf{u}_3$ is the generator of the null space of $[\mathbf{t}]_\times$. We already know that $\mathbf{t}$ is in the null space of $[\mathbf{t}]_\times$. Thus, $\mathbf{u}_3$ must be a unit vector of $\mathbf{t}$. In other words, $\mathbf{t}$ is also an axis of the rotation.

Next, prove the angle is $||\mathbf{t}||$. This is very easy by using Eq. (3.2) and (3.12):

$$1+2\cos\theta = \text{tr}(e^{[\mathbf{t}]_\times}) = \text{tr}(diag(\mathbf{R}(||\mathbf{t}||)) = 1+2\cos||\mathbf{t}||. \tag{3.13}$$

Therefore, the rotation angle $\theta = ||\mathbf{t}||$. □

From Theorem 3.4, any 3×3 rotation matrix can be presented by a 3-vector $\mathbf{t}$. This is another form to parameter the rotation matrix compared to the Euler angle form.

3.3.5 Specific form of skew-symmetric rotation

We may write a specific formula for the rotation matrix corresponding to $e^{[\mathbf{t}]_\times}$. We observe that $[\mathbf{t}]_\times^3 = -||\mathbf{t}||^2[\mathbf{t}]_\times = -||\mathbf{t}||^3[\hat{\mathbf{t}}]_\times$, where $\hat{\mathbf{t}}$ represents a unit vector in the direction $\mathbf{t}$. Then, we have

$$\begin{aligned}
e^{[\mathbf{t}]_\times} &= \mathbf{I} + [\mathbf{t}]_\times + [\mathbf{t}]_\times^2/2! + \cdots + [\mathbf{t}]_\times^k/k! + \cdots \\
&= \mathbf{I} + ||\mathbf{t}||[\hat{\mathbf{t}}]_\times + ||\mathbf{t}||^2[\hat{\mathbf{t}}]_\times^2/2! + \cdots + ||\mathbf{t}||^k[\hat{\mathbf{t}}]_\times^k/k! + \cdots \\
&= \mathbf{I} + (||\mathbf{t}|| - ||\mathbf{t}||^3/3! + \cdots)[\hat{\mathbf{t}}]_\times + (||\mathbf{t}||^2/2! - ||\mathbf{t}||^4/4! + \cdots)[\hat{\mathbf{t}}]_\times^2 \\
&= \mathbf{I} + \sin||\mathbf{t}||[\hat{\mathbf{t}}]_\times + (1-\cos||\mathbf{t}||)[\hat{\mathbf{t}}]_\times^2 \\
&= \mathbf{I} + \frac{\sin||\mathbf{t}||}{||\mathbf{t}||}[\mathbf{t}]_\times + \frac{1-\cos||\mathbf{t}||)}{||\mathbf{t}||^2}[\mathbf{t}]_\times^2 \\
&= \cos||t||\mathbf{I} + \frac{\sin||\mathbf{t}||}{||\mathbf{t}||}[\mathbf{t}]_\times + \frac{1-\cos||\mathbf{t}||)}{||\mathbf{t}||^2}\mathbf{t}\mathbf{t}^\top
\end{aligned} \tag{3.14}$$

where the last line follows from the identity $[\mathbf{t}]_\times^2 = \mathbf{t}\mathbf{t}^\top - ||\mathbf{t}||^2\mathbf{I}$.

Some properties from this representation:

- If the rotation matrix presented by $\mathbf{t}$ is very small, then we have approximations $\sin||\mathbf{t}|| = ||\mathbf{t}||, \cos||\mathbf{t}|| = 1$. Thus,

  $$e^{[\mathbf{t}]_\times} = \mathbf{I} + [\mathbf{t}]_\times \tag{3.15}$$

  This form is widely usually used to linearize the rotation matrix if the rotation angles are small, known as **small angle assumption**.

- If $\mathbf{t}$ is written as $\mathbf{t} = \theta\hat{\mathbf{t}}$, where $\hat{\mathbf{t}}$ is a unit vector and $\theta = ||\mathbf{t}||$ is the angle of the rotation, then the rotation matrix in Eq. (3.14) can be written as

  $$\mathbf{R}(\theta, \hat{\mathbf{t}}) = \mathbf{I} + \sin\theta\cdot[\hat{\mathbf{t}}]_\times + (1-\cos\theta)\cdot[\hat{\mathbf{t}}]_\times^2 \tag{3.16}$$

  This is known as **Rodrigues' rotation formula**. This is the rotation matrix if both the (unit) axis and angle of the rotation are known as input. Its full matrix is shown in Eq. (3.3).

4 Linear Least-squares minimization

This section mainly refers to Section Appendix 5 in Ref. [1].

5 Non-linear minimization: iterative estimation

This section mainly refers to Section Appendix 6 in Ref. [1].

5.1 Newton iteration

Suppose we are given a hypothesized function relation $\mathbf{X} = \mathbf{f}(\mathbf{P})$, where $\mathbf{X}$ is a *measurement vector* and $\mathbf{P}$ is a *parameter vector* in Euclidean spaces $\mathbb{R}^N$ and $\mathbb{R}^M$, respectively. Meanwhile, a measured value of $\mathbf{X}$ approximating the true value $\overline{\mathbf{X}}$ is provided, and we wish to find the vector $\hat{\mathbf{P}}$ that most nearly satisfies the function relation. More precisely, we seek the vector $\hat{\mathbf{P}}$ satisfying $\mathbf{X} = \mathbf{f}(\hat{\mathbf{P}}) - \varepsilon$ which $||\varepsilon|| = ||\mathbf{f}(\hat{\mathbf{P}}) - \mathbf{X}||$ is minimized. Note that this problem will become a linear least-squares problem if $\mathbf{f}$ is a linear function $\mathbf{f}(\mathbf{P}) = \mathbf{AP}$.

To solve the case where $\mathbf{f}$ is not a linear function, we may start with an initial estimated value $\mathbf{P}_0$, and proceed to refine the estimate under the assumption that the function $\mathbf{f}$ is locally linear. We assume that the function $\mathbf{f}$ is approximated at point $\mathbf{P}_0$ by

$$\mathbf{f}(\mathbf{P}_0 + \Delta) = \mathbf{f}(\mathbf{P}_0) + \mathbf{J}\Delta$$

which is obtained by Taylor expansion of $\mathbf{f}(\mathbf{P}_0 + \Delta)$ at point $\mathbf{P}_0$. Here $\mathbf{J}$ is the jacobian matrix with element $J_{ij} = \partial f_i / \partial p_j$, where f_i is $\mathbf{f}$'s ith function ($i \in [1, N]$) and p_j is $\mathbf{P}$'s jth variable ($j \in [1, M]$).

Now we seek a point $\mathbf{f}(\mathbf{P}_1)$ with $\mathbf{P}_1 = \mathbf{P}_0 + \Delta$, which minimizes $\varepsilon_1 = \mathbf{f}(\mathbf{P}_1) - \mathbf{X} = \mathbf{f}(\mathbf{P}_0) + \mathbf{J}\Delta - \mathbf{X} = \varepsilon_0 + \mathbf{J}\Delta$, where we define $\varepsilon_i = \mathbf{f}(\mathbf{P}_i) - \mathbf{X}$. Thus, the it is required to minimize $||\varepsilon_1|| = ||\varepsilon_0 + \mathbf{J}\Delta||$ over Δ, which is a linear minimization problem and can be easily solved by

$$\mathbf{J}^\top \mathbf{J}\Delta = -\mathbf{J}^\top \varepsilon_0.$$

Thus, the solution vector $\hat{\mathbf{P}}$ is obtained by starting with an estimate $\mathbf{P}_0$ and computing successive approximations according to the formula

$$\mathbf{P}_{i+1} = \mathbf{P}_i + \Delta_i$$

where Δ_i is the solution to the linear least-square problem

$$\mathbf{J_i}^\top \mathbf{J_i} \Delta_i = -\mathbf{J_i}^\top \varepsilon_i \tag{5.1}$$

where $\mathbf{J}_i$ is the Jacobian $\partial \mathbf{f} / \partial \mathbf{P}$ at $\mathbf{P}_i$ and $\varepsilon_i = \mathbf{f}(\mathbf{P}_i) - \mathbf{X}$. However, it is possible that the iteration procedure converges to a local minimum value, or does not converge at all. The behavior of the iteration algorithm depends very strongly on the initial estimate $\mathbf{P}_0$.

5.2 Weighted iteration

As an alternative to all the dependent variables being equally weighted described in Section 5.1, it is possible to provide a weight matrix specifying the weights of the dependent variables **X**. To be more precise, one may assume that the measurement X satisfies a Gaussian distribution with **covariance matrix** Σ, and one wishes to minimize the **Mahalanobis distance**

$$||\varepsilon||_\Sigma = \sqrt[2]{\varepsilon^\top \Sigma^{-1} \varepsilon}. \tag{5.2}$$

In this case, the normal equations will become

$$\mathbf{J_i}^\top \Sigma^{-1} \mathbf{J_i} \Delta_i = -\mathbf{J_i}^\top \Sigma^{-1} \varepsilon_i \tag{5.3}$$

and rest of the algorithm remains unchanged. If the covariance matrix is identity matrix, then Mahalanobis distance reduces to Euclidean distance and the problem will reduce back to the common Newton iteration.

5.3 Gauss-Newton method

We consider an arbitrary scalar-valued function $g(\mathbf{P})$ where $\mathbf{P}$ is a vector. The optimization problem is simply to minimize $g(\mathbf{P})$ over all values of $\mathbf{P}$. We make two assumptions: that $g(\mathbf{P})$ has a well-defined minimum value, and that we know a point $\mathbf{P}_0$ reasonably close to this minimum.

We may expand $g(\mathbf{P})$ about $\mathbf{P}_0$ in a Taylor series to get

$$g(\mathbf{P}_0 + \Delta) = g + g_\mathbf{P} \Delta + \Delta^\top g_\mathbf{PP} \Delta / 2 + \cdots$$

where subscript $\mathbf{P}$ denotes differentiation. We wish to minimize this quality with respect to Δ. Therefore, by differentiating $g(\mathbf{P}_0 + \Delta)$ with respect to Δ and set it to zero, arriving at the equation $g_\mathbf{P} + g_\mathbf{PP} \Delta = 0$ or

$$g_\mathbf{PP} \Delta = -g_\mathbf{P}. \tag{5.4}$$

Here $g_\mathbf{PP}$ is the matrix of second derivatives, or called the **Hessian** of function g, of which the (i, j)-th entry is $\partial^2 g / \partial p_i \partial p_j$, and $g_\mathbf{P}$ is the gradient of g. The method of **Newton iteration** method consists in starting an initial value $\mathbf{P}_0$, and iteratively computing parameter increments Δ using Eq. (5.4) until convergence occurs.

Now we turn to the sort of cost function that arises in the least-squares minimization problem described in Section 5.1. Specifically, $g(\mathbf{P})$ is the squared norm of an error function

$$g(\mathbf{P}) = ||\varepsilon(\mathbf{P})||^2 / 2$$

where $\varepsilon(\mathbf{P}) = \mathbf{f}(\mathbf{P}) - \mathbf{X}$. The factor 1/2 is present for simplifying the succeeding computations.

Now, we can easily compute the gradient vector $g_{\mathbf{P}} = \varepsilon_{\mathbf{P}}^{\top}\varepsilon = \mathbf{J}^{\top}\varepsilon$, and also the Hessian

$$g_{\mathbf{PP}} = \varepsilon_{\mathbf{P}}^{\top}\varepsilon_{\mathbf{P}} + \varepsilon_{\mathbf{PP}}^{\top}\varepsilon \tag{5.5}$$

The iterative procedure in Eq. (5.4) using Hessian shown in Eq. (5.5) is called **Newton's Method**. If assuming that $\mathbf{f}(\mathbf{P})$ is linear, then $\varepsilon_{\mathbf{PP}} = 0$ and the Hessian will be $g_{\mathbf{PP}} = \varepsilon_{\mathbf{P}}^{\top}\varepsilon_{\mathbf{P}} = \mathbf{J}^{\top}\mathbf{J}$, and the problem in Eq. (5.4) will reduce to the same linear minimization problem shown in Eq. (5.1). This procedure in which $\mathbf{J}^{\top}\mathbf{J}$ is used as an approximation for the Hessian in Eq. (5.5) is known as the **Gauss-Newton Method**.

5.4 Gradient descent method

The negative (or down-hill) gradient vector $-g_{\mathbf{P}} = -\varepsilon_{\mathbf{P}}^{\top}\varepsilon$ defines the direction of most rapid decrease of the cost function. A strategy for minimization of g is to move iteratively in the gradient direction. This is known as **gradient descent**. The length of the step may be computed by carrying out a line search for the function minimum in the negative gradient direction. In this case, the parameter increment Δ is computed from

$$\lambda\Delta = -g_{\mathbf{P}} \tag{5.6}$$

where λ controls the length of the step.

We may consider this as an approximation of Hessian by $\lambda\mathbf{I}$. Even thought gradient descent is not a good minimization strategy, it can be useful in conjunction with Gauss-Newton iteration as a way of getting out of tight corners. The Levenberg-Marquardt method is essentially a Gauss-Newton method that transitions smoothly to gradient descent when Gauss-Newton updates fail.

5.5 Levenberg-Marquardt method

As described in the last section, Levenberg-Marquardt method is a slight variation on the Gauss-Newton iteration method. The normal equation in Eq. (5.1) is replaced by the augmented normal equation

$$(\mathbf{J_i}^{\top}\mathbf{J_i} + \lambda\mathbf{I})\Delta_i = -\mathbf{J_i}^{\top}\varepsilon_i \tag{5.7}$$

for some value λ that varies from iteration to iteration.

However, note that this is not the only form of Levenberg-Marquardt (LM) method. For instance, a sparse LM method will be used when dealing with large number of parameters to avoid frequently solving Eq. (5.7) whose complexity is $O(n^3)$ in the number of parameters. Refer to more materials such as [1] for more details about this.

6 2D projective geometry and transformation

This section summarizes from Section 2 in Ref. [1].

6.1 Line and point

6.1.1 Definition and properties

- A 2D line equation is $ax+by+c=0$, so a line can be represented as a vector $\mathbf{l}=(a,b,c)^\top$. Note that $\mathbf{l}$ and $k\mathbf{l}$ (where k is a constant) is actually a same line, so $\mathbf{l}=(a,b,c)^\top$ is actually in homogeneous format and is a representative of equivalence class.
- The set of such line equivalence classes is defined as **projective space** $\mathbb{P}^2$, or **projective plane** (each set of line equivalence class forms a plane). Note that $(0,0,0)^\top$ is not in $\mathbb{P}^2$ since it is not a line. So actually $\mathbb{P}^2=\mathbb{R}^3-(0,0,0)^\top$. Meanwhile, each homogeneous vector $\mathbf{x}=(x_1,x_2,x_3)^\top$ with $x_3\neq 0$ correspond to finite point in $\mathbb{R}^2$. So $\mathbb{P}^2$ is also equal to $\mathbb{R}^2$ augmented by ideal points (points with last coordinate $x_3=0$) with $0,0,0)^\top$ excluded too.
- A 2D point $(x,y)^\top$ on a line $\mathbf{l}=(a,b,c)^\top$ satisfies $ax+by+c=0$. If we use homogeneous format $\mathbf{x}=(x,y,1)^\top$ to represent a point, then it satisfies $\mathbf{x}^\top\mathbf{l}=0$.
- An arbitrary homogeneous point $\mathbf{x}=(x_1,x_2,x_3)^\top$ represents inhomogeneous point $(x_1/x_3,x_2/x_3)^\top$ in $\mathbb{R}^2$. Then, points as homogeneous vectors are elements in $\mathbb{P}^2$.
- Both a 2D line and a point has 2 degrees of freedom. For a line, we can use its slope and intercept on y-axis as the two parameters.

6.1.2 Intersection

Theorem 6.1. *The intersection of two lines* $\mathbf{l}$ *and* $\mathbf{l}'$ *is the point* $\mathbf{x}=\mathbf{l}\times\mathbf{l}'$.

Proof. The theorem is very clear since you can easily verify that $\mathbf{x}^\top\mathbf{l}=\mathbf{x}^\top\mathbf{l}'=0$. Then, $\mathbf{x}$ must be on both lines, that is, $\mathbf{x}$ is the intersection point. □

Theorem 6.2. *The line through two points* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *is* $\mathbf{l}=\mathbf{x}_1\times\mathbf{x}_2$.

Proof. Similar to the last theorem, you can easily verify that $\mathbf{l}^\top\mathbf{x}_1=\mathbf{l}^\top\mathbf{x}_2=0$. That is, $\mathbf{l}$ is passing through both $\mathbf{x}_1$ and $\mathbf{x}_2$. □

6.1.3 Points at infinity and line at infinity

Point at infinity. For two lines $\mathbf{l} = (a,b,c)^\top$ and $\mathbf{l}' = (a,b,c')^\top$, we can get their intersection point $\mathbf{x} = \mathbf{l} \times \mathbf{l}' = (b,-a,0)^\top$ in homogeneous format via Theorem 6.1. Obviously, its inhomogeneous point $(b/0,-a/0)^\top$ is some **infinite point** and makes no sense so far. Actually, the two lines $\mathbf{l}$ and $\mathbf{l}'$ are parallel to each other so they do not intersect, or in other words, they intersect in some infinite point. Here we define homogeneous points $\mathbf{x} = (x_1,x_2,0)^\top$ as **ideal points**, or **points at infinity**.

Line at infinity. Easily, all homogeneous points $\mathbf{x} = (x_1,x_2,x_3)^\top$ with $x_3 \neq 0$ correspond to finite points in $\mathbb{R}^2$. By adding infinite points to $\mathbb{R}^2$, the set becomes exactly the projective space $\mathbb{P}^2$. Note that the infinite point set $(x_1,x_2,0)^\top$ lie on a single line $\mathbf{l}_\infty = (0,0,1)^\top$, which is called **line at infinity**. It can be easily verified that a line $\mathbf{l} = (a,b,c)^\top$ intersects the line at infinity $\mathbf{l}_\infty = (0,0,1)^\top$ in point $(b,-a,0)^\top$, and the line $\mathbf{l}' = (a,b,c')^\top$ parallel to $\mathbf{l}$ intersects $\mathbf{l}_\infty$ at the same point $(b,-a,0)^\top$. That is, parallel lines intersect with line at infinity in a same point. Actually, the vector $(b,-a)$ is the normal direction of line $\mathbf{l}$. Therefore, line at infinity can be thought of as the set of normal directions of lines.

Based on above descriptions, in projective space $\mathbb{P}^2$, any two lines intersect in a single point (maybe point at infinity), and two distinct points lie on a single line. This is NOT true in the standard Euclidean geometry $\mathbb{R}^2$ in which parallel lines do not intersect. The study of geometry of $\mathbb{P}^2$ is known as **projective geometry**.

6.2 Projective transformations

A **projectivity** is an invertible mapping from points in $\mathbb{P}^2$ (that is homogeneous 3-vectors) to points in $\mathbb{P}^2$ that maps **lines to lines**.

Definition 6.1. *A **projectivity** is an invertible mapping h from $\mathbb{P}^2$ to itself such that three points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ lie on the same line if and only if $h(\mathbf{x}_1)$, $h(\mathbf{x}_2)$ and $h(\mathbf{x}_3)$ do.*

Theorem 6.3. *A mapping $h: \mathbb{P}^2 \Rightarrow \mathbb{P}^2$ is a **projectivity** if and only if there exists a non-singular 3×3 matrix H such that for any point in $\mathbb{P}^2$ represented by a vector $\mathbf{x}$ it is true that $h(\mathbf{x}) = \mathbf{H}\mathbf{x}$.*

Proof. Let $\mathbf{x}$ lie on a line $\mathbf{l}$, then $\mathbf{l}^\top\mathbf{x} = 0$. So $\mathbf{l}^\top\mathbf{H}^{-1}\mathbf{H}\mathbf{x} = 0$. That is, the transformed point $\mathbf{H}\mathbf{x}$ is on the line $\mathbf{H}^{-\top}\mathbf{l}$. Similarly, if three points lie on this line, then their corresponding transformed points are obviously on the same transformed line. □

As a result of Theorem 6.3, one may give an alternative definition of a **projective transformation** (or collineation) as follows:

Definition 6.2. *A planar **projective transformation** is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix $\mathbf{x}' = \mathbf{H}\mathbf{x}$.*

Note that $\mathbf{H}$ is a homogeneous matrix (i.e., any arbitrary scale doesn't change the transformation) so it has 8 degrees of freedom.

Transformed line. We have already proved in Theorem 6.3 that the transformed line is $\mathbf{l}' = \mathbf{H}^{-\top}\mathbf{l}$.

6.2.1 Projective transformation between planes

Figure 1 gives an example of central projection. A **central projection** is projection along rays through a common point (i.e., **projection center**). It defines a mapping from points on one plane π to corresponding points on another plane π' (the two corresponding points are on a same ray through projection center). Clearly, this kind of mapping also projects lines on one plane to lines on the other plane, considering that **a third plane passing through the projection center intersects the two planes π and π' with two lines**. Therefore, according to Definition 6.1, this mapping is a projectivity. And according to Definition 6.2, there exists a transformation $\mathbf{x}' = \mathbf{H}\mathbf{x}$ between points $\mathbf{x}$ and $\mathbf{x}'$ on their planes π and π', if a coordinate system is defined in each plane and points are represented in homogeneous coordinates (here each plane is just like a 2D image). Figure 2c gives another example of projective transformation between a global plane and its shadow on the ground plane with sun as the projection center.

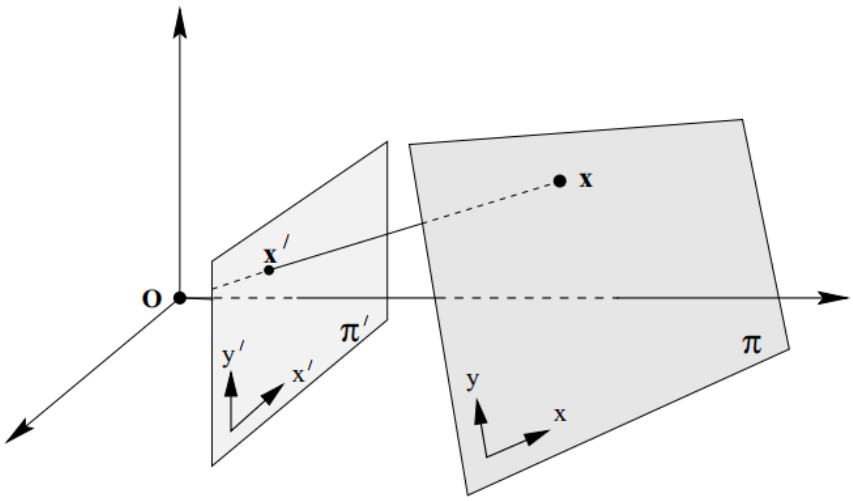


Figure 1: Central projection maps points from one plane to points on another plane, and also maps lines on one plane to lines on another plane. So there is a projective transformation between corresponding points on these two planes.

In camera projection model, the camera center is the projection center and 2D images are projected planes. Therefore, if the camera is only rotating (without translation), then there exist projective transformations between each pair of images (Figure 2b). Meanwhile, if camera moves but it takes pictures from coplanar points on a global plane surface, then projective transformations also exist between images (Figure 2a).

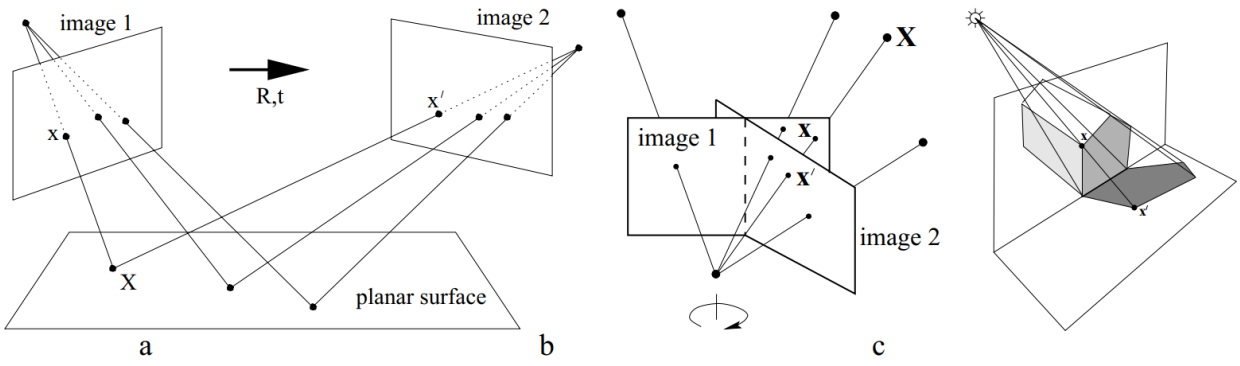


Figure 2: Examples of a projective transformation. (a) The projective transformation between two images induced by a world space plane (you can easily verify that the **concatenation** of two projective transformations is a projective transformation). (b) The projective transformation between two images with the same camera center (e.g. a camera rotating about its center or a camera varying its focal length); (c) The projective transformation between the image of a plane (the end of the building) and the image of its shadow onto another plane (the ground plane).

Figure 3 summarizes all common cases of projective transformation in camera projections. Note that only Figure 3(d) is not projective transformation between the two images while (e) is, since the camera center moves and there exists a projective transformation between two images only if the global points are coplanar.

6.3 Conics

In Euclidean geometry, conics are of three main types: hyperbola, ellipse and parabola. A typical conic's equation is $ax^2+bxy+cy^2+dx+ey+f=0$ or in matrix form

$$\mathbf{x}^\top \mathbf{C}\mathbf{x} = 0 \tag{6.1}$$

where $\mathbf{x} = (x, y, 1)^\top$ in homogeneous format and

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

The matrix $\mathbf{C}$ is symmetric matrix. We usually use the matrix $\mathbf{C}$ to represent a conic. Note that a conics has 5 degrees of freedom (up to scale) instead of 6. In other words, 5 points determine a conic.

Theorem 6.4. *The line* $\mathbf{l}$ *tangent to a conic* $\mathbf{C}$ *at a point* $\mathbf{x}$ *on* $\mathbf{C}$ *is given by* $\mathbf{l} = \mathbf{C}\mathbf{x}$.

Proof. Firstly, it is easy to prove that $\mathbf{l}$ passes through $\mathbf{x}$, since $\mathbf{x}^\top\mathbf{l} = \mathbf{x}^\top\mathbf{C}\mathbf{x} = 0$. Next we need to prove $\mathbf{l}$ is tangent to the conic. Suppose $\mathbf{l}$ meets the conic in another point $\mathbf{y}$. Then $\mathbf{y}^\top\mathbf{C}\mathbf{y} = 0$ and

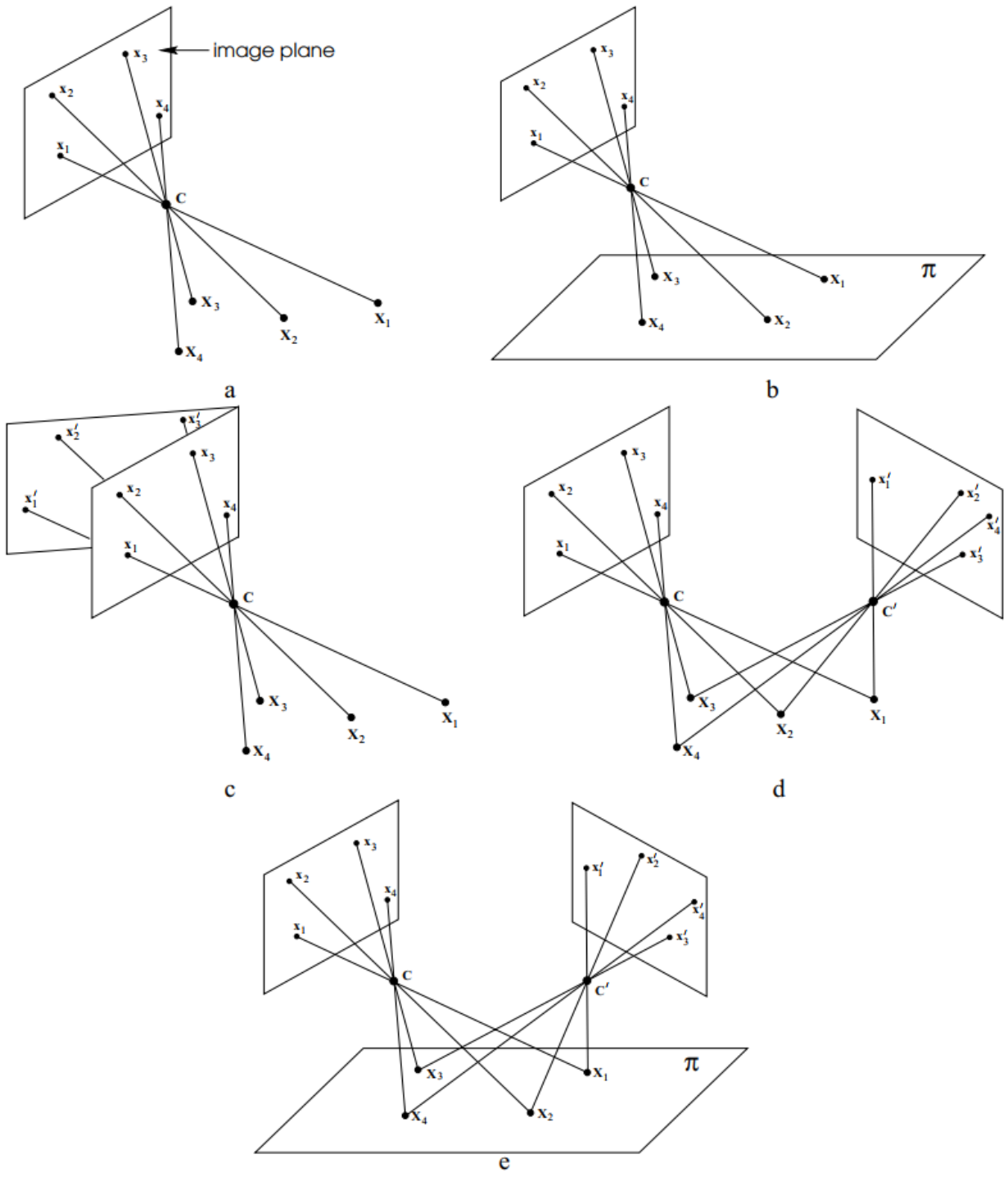


Figure 3: Different types of camera projections. (a) Image formation. (b) Image and global plane has projective transformation. (c) Two images has projective transformation. (d) Two images do NOT have projective transformation for arbitrary global points. (e) Two images has projective transformation for global coplanar points.

$\mathbf{y}^\top \mathbf{l} = \mathbf{y}^\top \mathbf{C}\mathbf{x} = 0 = \mathbf{x}^\top \mathbf{C}\mathbf{y}$. Then we have $(\mathbf{x}+\alpha\mathbf{y})^\top \mathbf{C}(\mathbf{x}+\alpha\mathbf{y}) = 0$ for any scalar value α (you can easily find it true by expanding the equation into terms). That is, the whole line defined by $\mathbf{x}, \mathbf{y}$ lies on the conic, which is clearly degenerate (matrix $\mathbf{C}$ is not full rank). Therefore, $\mathbf{l}$ has only 1 intersection point $\mathbf{x}$ on the conic which it is tangent to. □

Note that if $\mathbf{C}$ is not full-rank, then the conic is **degenerate**, such as two lines $\mathbf{C} = \mathbf{l}_1\mathbf{l}_2^\top + \mathbf{l}_2\mathbf{l}_1^\top$ or a repeated line $\mathbf{C} = \mathbf{l}\mathbf{l}^\top$.

6.3.1 Duality and dual conics

From the properties of point and lines, we can notice that there is a general duality principal between them. For instance, point on a line is $\mathbf{l}^\top \mathbf{x} = \mathbf{x}^\top \mathbf{l} = 0$ so their positions can be swapped, so are the situations in Theorem 6.1 and 6.2 which describes the intersection of points and lines.

A line $\mathbf{l}$ tangent to a point conic $\mathbf{C}$ satisfies $\mathbf{l}^\top \mathbf{C}^* \mathbf{l} = 0$, where $\mathbf{C}^*$ is the *adjoint matrix* of $\mathbf{C}$. A dual conic is a conic for lines, or a **line conic**.

We can get $\mathbf{C}^*$ like this: according to Theorem 6.4, a line tangent to a point conic is $\mathbf{l} = \mathbf{C}\mathbf{x}$, so we have $\mathbf{x} = \mathbf{C}^{-\mathbf{1}}\mathbf{l}$. Then, $0 = \mathbf{x}^\top \mathbf{C}\mathbf{x} = \mathbf{l}^\top \mathbf{C}^{-\top}\mathbf{l} = \mathbf{l}^\top \mathbf{C}^{-\mathbf{1}}\mathbf{l}$. That is, $\mathbf{C}^* = \mathbf{C}^{-1}$ for a non-singular symmetric matrix $\mathbf{C}$.

Note that dual conics are also known as **conic envelopes**. In geometry, an **envelope** of a family of curves in the plane is a curve that is tangent to each member of the family at some point, and these points of tangency together form the whole envelope. Therefore, lines satisfying $\mathbf{l}^\top \mathbf{C}^* \mathbf{l} = 0$ are tangent to point conic $\mathbf{C}$. That is, a point conic $\mathbf{C}$ is the envelope of line family $\mathbf{l}$ (see Figure 2.2 on Page 32 of Ref. [1]).

6.3.2 Transformed conic

Transformed point conic. Under point transformation $\mathbf{x}' = \mathbf{H}\mathbf{x}$, we have $0 = \mathbf{x}^\top \mathbf{C}\mathbf{x} = \mathbf{x}'^\top \mathbf{H}^{-\top}\mathbf{C}\mathbf{H}^{-1}\mathbf{x}'$. Therefore, the transformed conic will be $\mathbf{C}' = \mathbf{H}^{-\top}\mathbf{C}\mathbf{H}^{-\mathbf{1}}$.

Transformed dual conic. From Section 6.3.1 we know a dual conic (line conic) satisfies $\mathbf{l}^\top \mathbf{C}^* \mathbf{l} = 0$ with line $\mathbf{l} = \mathbf{C}\mathbf{x}$ tangent to original conic $\mathbf{C}$ with tangent point $\mathbf{x}$. And we also know that the transformed line is $\mathbf{l}' = \mathbf{H}^{-\top}\mathbf{l}$. Then, $\mathbf{l} = \mathbf{H}^\top \mathbf{l}'$, and further we have $0 = \mathbf{l}^\top \mathbf{C}^* \mathbf{l} = \mathbf{l}'^\top \mathbf{H}\mathbf{C}^*\mathbf{H}^\top \mathbf{l}'$. Therefore, the transformed dual conic is $\mathbf{C}^{*\prime} = \mathbf{H}\mathbf{C}^*\mathbf{H}^\top$.

6.3.3 The pole−polar relationship

An arbitrary point $\mathbf{x}$ and a point conic $\mathbf{C}$ define a line $\mathbf{l} = \mathbf{C}\mathbf{x}$. The line $\mathbf{l}$ is called the **polar** of $\mathbf{x}$ with respect to $\mathbf{C}$, and point $\mathbf{x}$ is the **pole** of $\mathbf{l}$ with respect to $\mathbf{C}$. Note that if point $\mathbf{x}$ is on the conic

$\mathbf{C}$, then $\mathbf{l}$ is the tangent line of $\mathbf{C}$ at $\mathbf{x}$, which is Theorem 6.4.

Theorem 6.5. *The polar line* $\mathbf{l} = \mathbf{C}\mathbf{x}$ *of the point* $\mathbf{x}$ *with respect to a conic* $\mathbf{C}$ *intersects the conic in two points. The two lines tangent to* $\mathbf{C}$ *at these points intersect at* $\mathbf{x}$.

Proof. Consider a point $\mathbf{y}$ on $\mathbf{C}$. The tangent line at $\mathbf{y}$ is $\mathbf{C}\mathbf{y}$, and this line contains $\mathbf{x}$ if $\mathbf{x}^\top \mathbf{C}\mathbf{y} = 0$. Using the symmetry of $\mathbf{C}$, the condition $\mathbf{x}^\top \mathbf{C}\mathbf{y} = (\mathbf{C}\mathbf{x})^\top \mathbf{y} = 0$ is that the point $\mathbf{y}$ lies on the line $\mathbf{C}\mathbf{x}$. Thus the polar line $\mathbf{C}\mathbf{x}$ intersects the conic in the point $\mathbf{y}$ at which the tangent line contains $\mathbf{x}$. □

Figure 4 illustrates the polar and pole relationship.

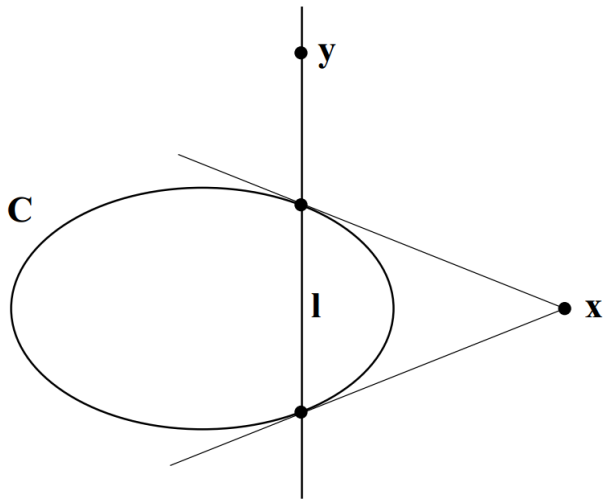


Figure 4: Polar and pole relationship.

6.4 A hierarchy of projective transformations

We use $PL(3)$ to denote the group of 3×3 projective transformations. Figure 5 shows common types/subgroups of projective transformations with a hierarchy structure. Each type of transformation group inherits the invariant properties from its upper types. For instance in Figure 5, the Euclidean transformation on the bottom row has all invariant properties from all types of transformations.

6.4.1 Isometries

A simple and basic type in $PL(3)$ is isometries which are transformations preserving Euclidean distance. One type of isometry is called **Euclidean transformation** (or **rigid transformation**)

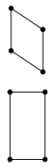
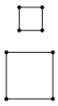
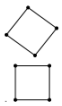
Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
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 (e.g. midpoints), linear combinations of vectors (e.g. centroids). 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. The circular points, $\mathbf{I}$, $\mathbf{J}$ (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Figure 5: **Geometric properties invariant to commonly occurring planar transformations**. The matrix $\mathbf{A}$ is 2×2 matrix, $\mathbf{R}$ is 2D rotation matrix and t_x, t_y is translation. The distortion column shows typical effects of the transformations on a square.

which is combination of rotation and translation with totally 3 dof:

$$\mathbf{H_E} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

6.4.2 Similarity transformations

The next subgroup is **similarity transformation**, which is an affine matrix with upper left 2×2 part a rotation matrix times a scale factor:

$$\mathbf{H_S} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

It is the combination of rotation, **isotropic** scaling and translation with totally 4 dof.

6.4.3 Affine transformations

Another subgroup of $PL(3)$ is **affine transformation** which can be regarded as the combination of rotation, **anisotropic** scaling and translation. Each affine matrix is a projective transformation

matrix with last row $(0,0,1)$ like this:

$$\mathbf{H_A} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

Note that according to Singular Value Decomposition (SVD), any non-singular affine matrix can be decomposed into

$$\mathbf{A} = \mathbf{UDV} = \mathbf{UV^{-1}VDV} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{DR}(\phi)$$

That is, each affine matrix is the concatenation of a rotation ($R(\phi)$), an anisotropic scaling (λ_1, λ_2 of diagonal matrix $\mathbf{D}$), and a rotation back ($R(\phi)$) and finally another rotation ($R(\theta)$).

6.4.4 Projective transformations

A common projective transformation matrix has 8 degrees of freedom (dof):

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^\top & 1 \end{bmatrix}$$

with 2 more dof than affine matrix in $\mathbf{v}$, which affects the non-linear effects of the projectivity. Compare the mapping of an ideal point $(x_1, x_2, 0)^\top$ under the two transformations: an affine transformation converts/maps this ideal point to **another ideal** point $((x_1, x_2)\mathbf{A}^\top, 0)^\top$, while a projective transformation maps the ideal point to a **finite** point $((x_1, x_2)\mathbf{A}^\top, (x_1, x_2)\mathbf{v})^\top$. Therefore, a projective transformation can model vanishing points.

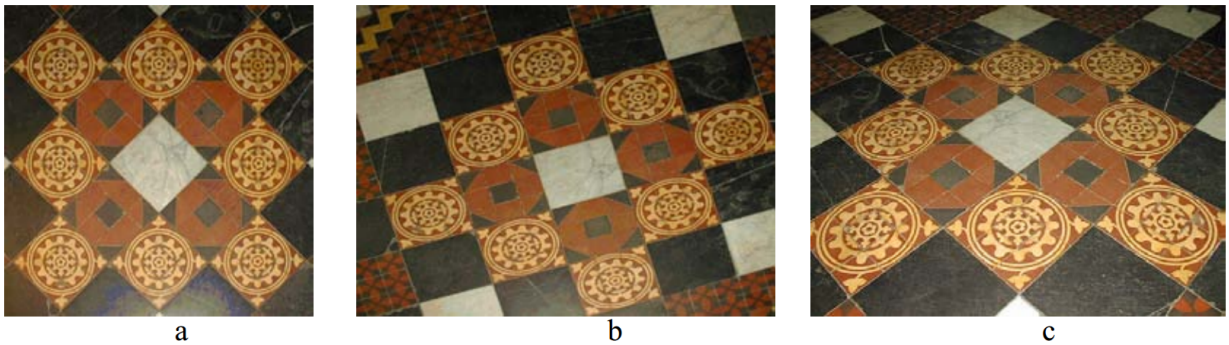
A projective transformation can be decomposed into a chain of transformations from its subgroups:

$$\mathbf{H} = \mathbf{H_S}\mathbf{H_A}\mathbf{H_P} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{v}^\top & 1 \end{bmatrix} \tag{6.2}$$

with $\mathbf{B}$ a non-singular matrix given by $\mathbf{B} = s\mathbf{RK} + \mathbf{tv}^\top$, and $\mathbf{K}$ an upper-triangular matrix normalized as $det(\mathbf{K}) = 1$. This decomposition is valid only if $v \neq 0$ and is unique if s is chosen positive. You can easily get this decomposition by running QR decomposition on $\mathbf{B} - \mathbf{tv}^\top$. We can see that each decomposed matrix has its own function and does not influence others. Consider the process of rectifying the perspective image of a global plane (such as example in Figure **??**): $\mathbf{H_P}$ (2 dof) moves line at infinity; $\mathbf{H_A}$ (2 dof) affects the affine properties but does not affect lines at infinity; $\mathbf{H_S}$ (4 dof) is general similarity transformation and does not influence affine or projective properties at all. Easily, transformation types in the decomposition in Eq. (6.2) can be **reversed**, i.e., $\mathbf{H}$ can also be decomposed into $\mathbf{H} = \mathbf{H_P}\mathbf{H_A}\mathbf{H_S}$ (of course values in matrices here are different).

6.5 Remove projective distortion from a perspective image of a global plane

The properties of projective transformation on planes can be used to remove projective distortion from a perspective image of a global plane. Figure 6 gives examples of different types of projective distortions in images taken on a global plane.



a b c

Figure 6: **Distortions arising under central projection**. (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.

There are two common ways to remove projective distortion from an image of a global plane:

1. The most intuitive way is to explicitly compute the projective transformation between the image plane and global plane, and apply the jtransformation on the image to get the transformed image without distortion (Section 6.5.1 and the second metric rectification method in Section 6.5.3).

2. Another way is to remove subgroups or 'components' of projective transformation one by another. As shown in Eq. (6.2), each projective transformation can be decomposed into a chain of transformations from its subgroup $\mathbf{H} = \mathbf{H_S}\mathbf{H_A}\mathbf{H_P}$. Therefore, if we want to measure the affine properties (such as parallelism, ratio of areas) on the original plane from the image, we only need to remove the 'projective component' $\mathbf{H_P}$ from $\mathbf{H}$ keep the rest. This process is called **affine rectification** (Section 6.5.2). If we want to measure the similarity/metric properties (angles, ratios of lengths) on the original plane from the image, then after removing $\mathbf{H_P}$ in affine rectification, we can further remove the affine component $\mathbf{H_A}$ and leave the similarity transformation only. This is called **metric rectification** (the first metric rectification method in Section 6.5.3).

Some important notes about the two above methods:

1. The first method is more straightforward but needs more information to recover the entire projective transformation, while the second way is decomposed into sub-steps and is more flexible in different situations.

2. The first method of computing entire projective transformation actually only removes 'projective' and 'affine' components and still leaves the similarity component (rotation, translation and isotropic scaling), but this doesn't influence further measure, since a similarity

transformation almost keeps all important invariant properties (Figure 5).

6.5.1 Remove entire projective distortion

The most intuitive way for projective rectification is to compute projective transformation directly and apply it onto image to remove entire projective distortion. Each projective transformation $\mathbf{H}$ has 8 dof. Therefore, in order to compute $\mathbf{H}$ between image and global plane, we only need coordinates of **4 pairs** of point correspondences between image plane and global plane (points must be non-colinear to ensure the coefficient matrix of the 8 equations is full-rank).

Note that the above method is not applicable to many situations, since we need to get points' coordinates in **global** plane as input, which are always hard or even impossible to be obtained. Another alternative method only utilizing points from the image will be introduced later in Section 6.5.3.

6.5.2 Affine rectification

Sometimes we only need to remove the 'projective component' from projective transformation in the image and measure affine properties (such as parallelism, ratio of areas) of original global plane from the image. To achieve this, we do not need to recover the entire projective transformation with 8 dof. From the decomposition of projective transformation $\mathbf{H} = \mathbf{H_S}\mathbf{H_A}\mathbf{H_P}$ in Eq. (6.2), a 'projective component' $\mathbf{H_P}$ only takes 2 dof (in $\mathbf{v}$ on the third row). Therefore, we only need to specify 2 dof in order to determine affine properties. Actually, these 2 dof are given by being associated with the line at infinity $\mathbf{l}_\infty$. In this section, we firstly introduce the 'invariant factor' under affine transformation, and then describe the affine rectification method based on the invariant factor.

Line at infinity and affinity. As we describe in Section 6.1.3, a line at infinity is denoted as $(0,0,1)^\top$. From Section 6.4.4 we know that an affine transformation maps a point at infinity (ideal point) to itself, while a projective transformation maps it to a finite point. This is also the similar to the dual case on line that **an affine transformation maps a line at infinity to itself**, while a projective transformation maps it to a finite line. To prove this, from Section **??** we know that a 2D line's transformed line is $\mathbf{l}' = \mathbf{H}^{-\top}\mathbf{l}$. Then we have:

$$\mathbf{l}'_\infty = \mathbf{H}_\mathbf{A}^{-\top}\mathbf{l}_\infty = \begin{bmatrix} \mathbf{A}^{-\top} & \mathbf{0} \\ -\mathbf{t}^\top\mathbf{A}^{-\top} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{l}_\infty$$

The converse is also true, i.e., an affine transformation is the most general linear transformation that fixes $\mathbf{l}_\infty$, that is, the vector $\mathbf{v}$ in the third row of transformation matrix must be $\mathbf{0}$. Then we have an important result:

Result 6.1. *The line at infinity* $\mathbf{l}_\infty$ *is a fixed line under the projective transformation* $\mathbf{H}$ ***if and only if*** $\mathbf{H}$ *is an affinity.*

Note that $\mathbf{l}_\infty$ is only fixed linewise but not pointwise. That is, a point at infinity will be mapped by an affinity to another point at infinity instead of itself (unless some special value cases in the affine matrix), even though the line at infinity is still the same.

Affine rectification method. Result 6.1 is the core that our affine rectification method refers to. Figure 7 illustrates the process for affine rectification. In this problem, initially we have the perspective image plane π_2 (such as Figure 8a) taken on the global plane π_1. The line at infinity $\mathbf{l}_\infty = (0,0,1)^\top$ from global plane π_1 is mapped to some finite line $\mathbf{l}$ in π_2. If we can get $\mathbf{l}$ from π_2, we can compute a transformation $\mathbf{H_P}'$ to transform $\mathbf{l}$ back to $\mathbf{l}_\infty$ and get a new transformed image π_3. Clearly $\mathbf{l}_\infty$ is fixed between the global plane and π_3. Therefore, according to Result 6.1, the transformation between global plane π_1 and π_3 must be an affinity. Then, π_3 is our result image with projective distortion removed (such as Figure 8b).

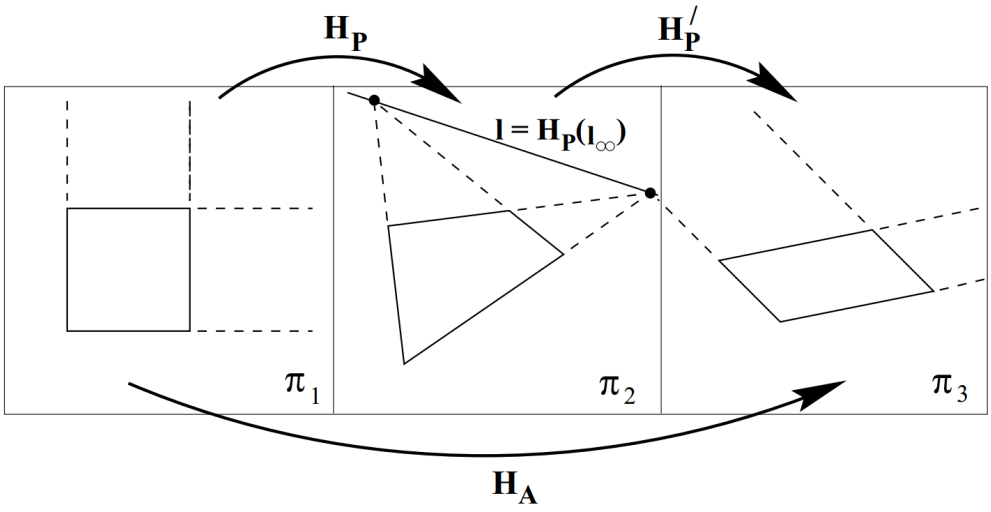


Figure 7: Affine rectification illustration.

Based on the above analysis, we have two sub-problems left:

1. How to find the imaged/distorted line at infinity $\mathbf{l}$ in the perspective image?
2. After we get $\mathbf{l}$, how to compute the new transformation $\mathbf{H_P}'$ to map $\mathbf{l}$ back to $\mathbf{l}_\infty$?

Here the second problem is easy to answer. Given $\mathbf{l} = (l_1, l_2, l_3)^\top$ (provided $l_3 \neq 0$), a suitable projective transformation matrix can be

$$\mathbf{H_P}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix}$$

You can easily verify that it transformed $\mathbf{l}$ back to $\mathbf{l}_\infty$ since $(\mathbf{H_P}')^{-\top}(l_1, l_2, l_3)^\top = (0,0,1)^\top = \mathbf{l}_\infty$.

To solve the first aforementioned problem, we have two common methods:

1. **Parallel line method**. In this method, we need to take as input **2 pairs of lines in the image that are parallel in the global plane**. In order to get the imaged line at infinity, we only need to find two **vanishing points** (i.e., imaged point at infinity), each of which is actually the imaged point of the intersection of two parallel lines in global plane. This is true based on the property that a projective transformation keeps line intersection. That is, two transformed parallel lines intersects at the transformed intersection point. Figure 8 illustrates this method. We use the 4 lines on the floor as input to find the two intersection points which are vanishing points, and then determine the imaged line at infinity.

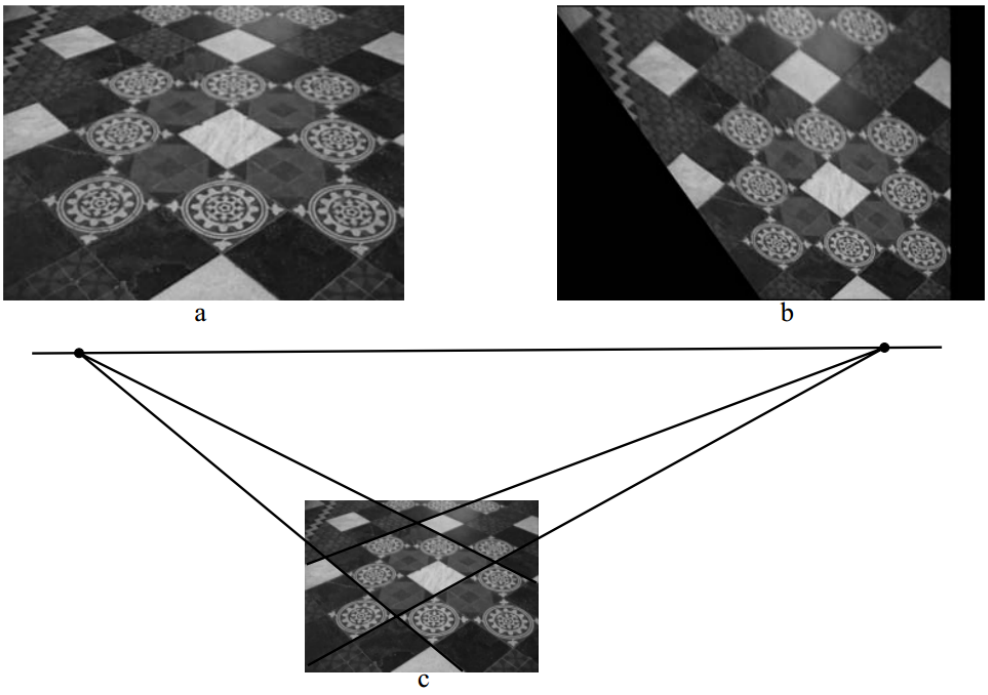


Figure 8: Affine rectification via parallel line method.

2. **Length ratio method**. In this method, we need to take as input **a pair of 2 intervals on two intersected lines in the image and these intervals' corresponding values in global plane**. The idea of this method is to compute 1D projective transformation $\mathbf{H}_1$ between pairs of points on a imaged line and global line, and then compute the imaged vanishing point using $\mathbf{H}_1$. The input is the ratio of two intervals between three colinear points $\mathbf{a},\mathbf{b},\mathbf{c}$ on the world line and $\mathbf{a}',\mathbf{b}',\mathbf{c}'$ on the imaged line. Suppose their ratio is $a:b$ on the world line, and $a':b'$ on the imaged line. So, we can present $\mathbf{a},\mathbf{b},\mathbf{c}$'s 1D coordinates as $(0,1)^\top,(a,1)^\top,(a+b,1)^\top$, and $\mathbf{a}',\mathbf{b}',\mathbf{c}'$'s coordinates as $(0,1)^\top,(a',1)^\top,(a'+b',1)^\top$. Then, we can use the 3 pairs of corresponding points to compute the 1D projective transformation $\mathbf{H}_1$ between the imaged line and global line, since $\mathbf{H}_1$ is 2×2 matrix with only 3 dof. After this, the vanishing point on the imaged line is $\mathbf{H}_1(1,0)^\top$. We can use the same way to get another vanishing point on

the other input line, and then the final imaged line at infinity. Figure 9 illustrates this method.

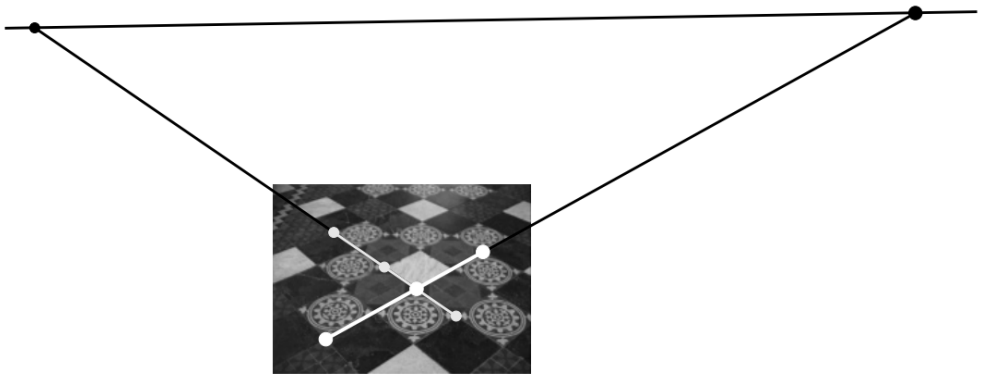


Figure 9: Affine rectification via length ratio method.

6.5.3 Metric rectification

Similar to affine rectification, each projective transformation only has 4 dof more than a similarity transformation, so we only need to specify 4 dof in order to determine metric/similarity properties, or specify 2 more dof based on the result of affine rectification. Similar the the last section, in this section, we firstly introduce the 'invariant factor' under similarity transformation, and then describe the metric rectification method based on the invariant factor.

1. Circular points and their dual

As described in the last section, $\mathbf{l}_\infty$ is only fixed linewise but not pointwise by an affinity. However, under any similarity transformation there are two points on $\mathbf{l}_\infty$ which are fixed. These are the **circular points** (also called the **absolute points**) denoted by $\mathbf{I}, \mathbf{J}$, with canonical coordinates

$$\mathbf{I} = (1, i, 0)^\top, \mathbf{J} = (1, -i, 0)^\top \tag{6.3}$$

The circular points are a pair of complex conjugate ideal points. The name **circular points** arises because they are the intersection points between a circle and $\mathbf{l}_\infty$. By combining their equations

$$\begin{cases} x_1^2 + x_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0 \\ \mathbf{l}_\infty^\top (x_1, x_2, x_3)^\top = 0 \end{cases}$$

we can get

$$\begin{cases} x_1^2 + x_2^2 = 0 \\ x_3 = 0 \end{cases} \tag{6.4}$$

whose solution is exactly $\mathbf{I}, \mathbf{J}$ in Eq. (6.3), i.e. any circle intersects $\mathbf{l}_\infty$ in the circular points.

In Euclidean space, a 2D circle is determined by 3 points, while In general formula Eq. (6.1), a conic is determined by 5 points, 2 of which is circular points for a circle.

2. Circular points are fixed by similarity

To see that they are fixed under an orientation-preserving similarity:

$$\mathbf{I}' = \mathbf{H_S}\mathbf{I} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} = se^{-i\theta}\mathbf{I} = \mathbf{I} \tag{6.5}$$

since according to Eular's equation, $e^{i\theta} = \cos\theta + i\sin\theta$. Analogous proof is easy for $\mathbf{J}$. The converse is also true and easy to prove. If $\mathbf{H_S}\mathbf{I} = s\mathbf{I}$, then h_{31}, h_{32} in the last row must be both 0 and the upper left 2×2 matrix must be a rotation matrix. So we have an important result:

Result 6.2. *The circular points,* $\mathbf{I}, \mathbf{J}$ *are fixed points under the projective transformation* $\mathbf{H}$ ***if and only if*** $\mathbf{H}$ *is a similarity.*

3. The conic dual to the circular points

The dual conic to circular points is

$$\mathbf{C}_\infty^* = \mathbf{I}\mathbf{J}^\top + \mathbf{J}\mathbf{I}^\top, \tag{6.6}$$

which is a degenerate line conic consisting two circular points. In Euclidean coordinate system we can use $\mathbf{I}, \mathbf{J}$'s canonical coordinate in Eq. (6.3) to compute

$$\mathbf{C}_\infty^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{6.7}$$

We can verify that $\mathbf{C}_\infty^*$ is fixed under transformations in analogous to the fixed properties of circular points $\mathbf{I}, \mathbf{J}$, since according to Section 6.3.2, the transformed conic is $\mathbf{C}_\infty^{*\prime} = \mathbf{H_S}\mathbf{C}_\infty\mathbf{H_S}^\top = \mathbf{H_S}(\mathbf{I}\mathbf{J}^\top + \mathbf{J}\mathbf{I}^\top)\mathbf{H_S}^\top = \mathbf{C}_\infty$. The converse is also true to be verified. So we have a result similar to the previous one:

Result 6.3. *The dual conic* $\mathbf{C}_\infty^*$ *is fixed under the projective transformation* $\mathbf{H}$ ***if and only if*** $\mathbf{H}$ *is a similarity.*

Note that only under similarity transformation, $\mathbf{I}, \mathbf{J}$ are fixed and have canonical coordinates in Eq. (6.3). Under other types of transformation, $\mathbf{I}, \mathbf{J}$ are mapped to other points such as finite points. For the dual conic, its general form is Eq. (6.6) in **any** coordinate systems, and its special form in Eq. (6.7) is only in Euclidean system.

Some properties of $\mathbf{C}_\infty^*$:

1. $\mathbf{C}_\infty^*$ has 4 dof. A 3×3 symmetric matrix has 5 dof, but $\mathbf{C}_\infty^*$'s rank is 2, which reduces 1 dof.
2. $\mathbf{l}_\infty$ is null vector of $\mathbf{C}_\infty^*$. Since $\mathbf{I}, \mathbf{J}$ are circular points on $\mathbf{l}_\infty$, then $\mathbf{C}_\infty^*\mathbf{l}_\infty = (\mathbf{I}\mathbf{J}^\top + \mathbf{J}\mathbf{I}^\top)\mathbf{l}_\infty = 0$.

4. Angle between two lines on the projective plane

The general format of the angle between two lines $\mathbf{l}$ and $\mathbf{m}$ is

$$\cos\theta = \frac{\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m}}{\sqrt{(\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{l})(\mathbf{m}^\top \mathbf{C}_\infty^* \mathbf{m})}}, \tag{6.8}$$

which is the form in any coordinate systems. In the Euclidean system, $\mathbf{C}_\infty^*$ is the matrix shown in Eq. (6.7), and then the angle between two lines is the common form we are familiar with:

$$\cos\theta = \frac{\mathbf{l}^\top \mathbf{m}}{(\mathbf{l}^\top \mathbf{l})(\mathbf{m}^\top \mathbf{m})},$$

We can verify that the angle in Eq. (6.8) is invariant to projective transformations, since the transformed numerator in the equation is

$$\mathbf{l}^{\top\prime} \mathbf{C}_\infty^{*\prime} \mathbf{m}' = (\mathbf{l}^\top \mathbf{H}^{-1})(\mathbf{H}\mathbf{C}_\infty^* \mathbf{H}^\top)(\mathbf{H}^{-\top}\mathbf{m}) = \mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m}.$$

Similarly the denominator is also unchanged. Based on Eq. (6.8), we can redefine the orthogonality of two lines with:

Result 6.4. *Two lines* $\mathbf{l}, \mathbf{m}$ *are orthogonal if* $\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m} = 0$ *in any coordinate system.*

5. Metric rectification method.

Review that according to Result 6.1, the affine rectification method transforms the imaged $\mathbf{l}_\infty$ to its fixed position to obtain the affine transformation between transformed image and original global plane. Similarly, the metric rectification method here transforms the circular points $\mathbf{I}, \mathbf{J}$ to their fixed canonical coordinates in Eq. (6.3). Then, according to Result 6.2, the transformation between the transformed plane and original global plane must be a similarity. However, in practice instead of transforming $\mathbf{I}, \mathbf{J}$, it is easier and more straightforward to follow Result 6.3 to transform $\mathbf{C}_\infty^{*\prime}$ from the perspective image plane to its original form $\mathbf{C}_\infty^*$ in Eq. (6.7). Then the transformation between transformed plane and original plane is a similarity.

Using the decomposition chain of each projective transformation, we have

$$\begin{aligned}
\mathbf{C}_\infty^{*\prime} &= (\mathbf{H_P H_A H_S}) \mathbf{C}_\infty (\mathbf{H_P H_A H_S})^\top \\
&= (\mathbf{H_P H_A})(\mathbf{H_S C}_\infty \mathbf{H_S}^\top)(\mathbf{H_P H_A})^\top \\
&= (\mathbf{H_P H_A}) \mathbf{C}_\infty (\mathbf{H_P H_A})^\top \\
&= \begin{bmatrix} \mathbf{K}\mathbf{K}^\top & \mathbf{K}\mathbf{K}^\top \mathbf{v} \\ \mathbf{v}^\top \mathbf{K}\mathbf{K}^\top & \mathbf{v}^\top \mathbf{K}\mathbf{K}^\top \mathbf{v} \end{bmatrix},
\end{aligned} \tag{6.9}$$

where $\mathbf{K}, \mathbf{v}$ are components of $\mathbf{H_A}, \mathbf{H_P}$ shown in Eq. (6.2) , respectively. Eq. (6.9) shows that the transformed dual conic contains only affine and projective transformation. Therefore, if we can identify $\mathbf{C}_\infty^{*\prime}$ in the perspective image, we can compute a transformation $\mathbf{H}$ to transform $\mathbf{C}_\infty^{*\prime}$ back

to original $\mathbf{C}_\infty^*$. Clearly, $\mathbf{C}_\infty^{*\prime}$ is a symmetric matrix with rank 2 (its third row is linear representation of the first two rows), so using SVD we can get $\mathbf{C}_\infty^{*\prime} = \mathbf{U}\mathbf{C}_\infty^*\mathbf{U}^\top$, where $\mathbf{C}_\infty^*$ is the original form in Eq. (6.7). Then, $\mathbf{C}_\infty^* = \mathbf{U}^{-1}\mathbf{C}_\infty^{*\prime}\mathbf{U}^{-\top}$, so the final transformation we want is $\mathbf{H} = \mathbf{U}^{-1}$.

Now the only problem left is how to identify $\mathbf{C}_\infty^{*\prime}$ in the perspective image. Similar to affine rectification strategy, here we also have two methods:

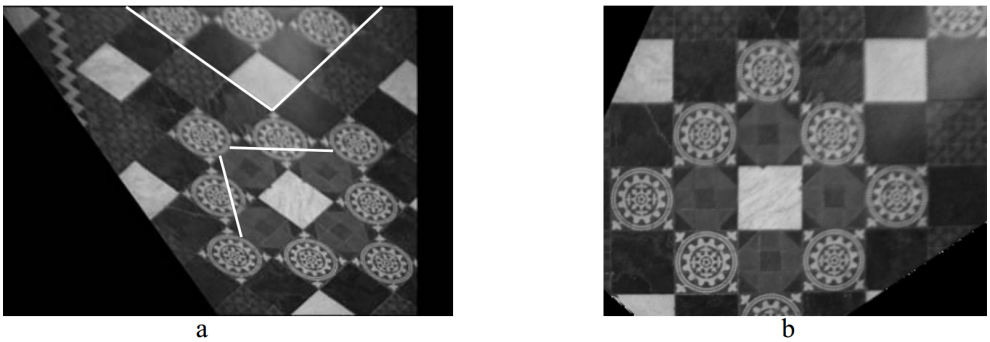
(1) **Start from the affinely rectified image plane**. Suppose an image **has been already affinely rectified**, then we only require two constraints to specify 2 dof in order to determine a metric rectification. The input of this rectification method is two pairs of imaged lines $\mathbf{l}', \mathbf{m}'$ which are vertical on the **global** plane. From Result 6.4, $\mathbf{l}'^\top \mathbf{C}_\infty^{*\prime} \mathbf{m}' = 0$. So using Eq. (6.9) with $\mathbf{v} = \mathbf{0}$ (since the image is already affinely rectified so the projective component is gone now), we have

$$\begin{bmatrix} l'_1 & l'_2 & l'_3 \end{bmatrix} \begin{bmatrix} \mathbf{K}\mathbf{K}^\top & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} m'_1 \\ m'_2 \\ m'_3 \end{bmatrix} = 0,$$

which is a linear constraint on the 2×2 matrix $\mathbf{S} = \mathbf{K}\mathbf{K}^\top$ that is symmetric with 2 dof (overall scaling is unimportant). Therefore, two such pairs of vertical lines on the global plane are enough to compute $\mathbf{S}$ and further $\mathbf{K}$ up to scale by Cholesky decomposition. Figure 10 illustrates this method.

(2) **Compute it from the original perspective image**. If we start from the original image, we need to compute $\mathbf{C}_\infty^{*\prime}$ directly. According to previous description, $\mathbf{C}_\infty^{*\prime}$ has 4 dof but 1 dof is reduced because it is not full rank, which condition is hard to use during computation. Therefore, we still need to provide 5 constraints. That is, by providing 5 pairs of lines which are vertical on the global plane as input, we can use Result 6.4 to compute $\mathbf{C}_\infty^{*\prime}$ linearly. Figure 11 illustrates this method.

Comparison with previous methods. We have already introduced an intuitive method of computing entire projective transformation in Section 6.5.1. However, this method takes as input 5 pairs of points from global plane which are very hard to achieve (since we usually only have an image as input and know nothing about the global space). Analogous situation also exists in affine rectification method which needs length ratio from global plane. Compared to them, the metric rectification methods described here only takes as input pairs of lines that are vertical on global plane, which information is much easier to be obtained directly from the image plane (Figure 10a and 11a).



a b

Figure 10: Metric rectification with two pairs of vertical lines from affinely rectified image.

a b

Figure 11: Metric rectification with five pairs of vertical lines from original image.

7 3D projective geometry and transformation

3D projective geometry and transformation in this section is very analogous to the 2D case in previous Section 6. Most of its contents can be regarded as a simple extension from 2D case to 3D.

Similar to $\mathbb{P}^2$ described in Section 6, $\mathbb{P}^3$ is equal to $\mathbb{R}^3$ augmented with a set of ideal points which are on the plane $\boldsymbol{\pi}_\infty = (0,0,0,1)^\top$ (that is, an ideal point is like $(x_1,x_2,x_3,0)^\top$).

7.1 3D Points and planes

A 3D homogeneous point can be written as $\mathbf{X} = (X,Y,Z,1)^\top$, and a 3D plane can be written as $\boldsymbol{\pi} = (\pi_1,\pi_2,\pi_3,\pi_4)$. A point on a plane is written as

$$\boldsymbol{\pi}^\top \mathbf{X} = 0. \tag{7.1}$$

We can also use $\mathbf{n}^\top \widetilde{\mathbf{X}} + d = 0$, where $\mathbf{n} = (\pi_1,\pi_2,\pi_3)^\top$ is the normal of the plane, $\widetilde{\mathbf{X}} = (X,Y,Z)^\top$ is the inhomogeneous point, and $d = \pi_4$.

One important property is that, **3 planes define a point**, and **3 points define a plane**. That is, plane and point are dual to each other, just like a line and point are dual to each other in the 2D case.

Parametrization of point on a plane. A point $\mathbf{X}$ on a plane $\boldsymbol{\pi}$ can be written as $\mathbf{X} = \mathbf{M}\mathbf{x}$ where $\mathbf{x}$ is arbitrary 3-vector.

Transformation of a plane. Under a point transformation $\mathbf{x}' = \mathbf{H}\mathbf{X}$, a plane transforms as

$$\boldsymbol{\pi}' = \mathbf{H}^{-\top}\boldsymbol{\pi}, \tag{7.2}$$

since $0 = \boldsymbol{\pi}^\top \mathbf{H}^{-1}\mathbf{H}\mathbf{X} = \boldsymbol{\pi}'\mathbf{x}'$.

7.2 Quadrics

A quadric is a surface defined by

$$\mathbf{X}^\top \mathbf{Q}\mathbf{X} = 0 \tag{7.3}$$

where $\mathbf{Q}$ is a 4×4 symmetric matrix. A quadric $\mathbf{Q}$ has 9 dof (10 elements less one for scale).

Properties:

1. A quadric defines a polarity between a point and a plane, in a similar manner to the polarity defined by a conic between a point and a line. The plane $\boldsymbol{\pi} = \mathbf{Q}\mathbf{X}$ is the polar plane of $\mathbf{X}$ with respect to $\mathbf{Q}$. Similarly, if $\mathbf{X}$ is on quadric $\mathbf{Q}$, then plane $\boldsymbol{\pi} = \mathbf{Q}\mathbf{X}$ is the tangent plane to $\mathbf{Q}$ at $\mathbf{X}$.
2. The intersection of a plane and a quadric is a conic.

3. Under a point transformation $\mathbf{X}' = \mathbf{H}\mathbf{X}$, a quadric transforms as

$$\mathbf{Q}' = \mathbf{H}^{-\top}\mathbf{Q}\mathbf{H}^{-1}, \tag{7.4}$$

since $0 = \mathbf{X}^\top\mathbf{Q}\mathbf{X} = \mathbf{X}'\mathbf{H}^{-\top}\mathbf{Q}\mathbf{H}^{-1}\mathbf{X}'$.

4. Analogous to 2D case that the dual of a point conic is line conic, the dual of a 3D point quadric is a **plane quadric** satisfying $\boldsymbol{\pi}^\top\mathbf{Q}^*\boldsymbol{\pi} = 0$, where $\mathbf{Q}^* = \mathbf{Q}^{-1}$ if $\mathbf{Q}$ is invertible.

   Under a point transformation $\mathbf{X}' = \mathbf{H}\mathbf{X}$, a dual quadric transforms as

$$\mathbf{Q}^{*\prime} = \mathbf{H}\mathbf{Q}^*\mathbf{H}^\top, \tag{7.5}$$

   since $0 = \boldsymbol{\pi}^\top\mathbf{Q}^*\boldsymbol{\pi} = \boldsymbol{\pi}'^\top\mathbf{H}\mathbf{Q}^*\mathbf{H}^\top\boldsymbol{\pi}'$.

7.3 The hierarchy of 3D Projective transformation

The properties of 3D transformation can be simply regarded as an extension from 2D transformations (Section 6.4) to 3D. Please refer to Section 3.4 in Ref. [1] for details.

7.4 Remove 3D projective distortions

In Section 6.5, we described 2D planar projective geometry and distortion removal which identifies three key components: line at infinity $\mathbf{l}_\infty$, circular points $\mathbf{I}, \mathbf{J}$ on $\mathbf{l}_\infty$, and circular points' dual conic $\mathbf{C}^*_\infty$. In 3D projective geometry, their corresponding geometric entities are: plane at infinity $\boldsymbol{\pi}_\infty$, absolute conic $\boldsymbol{\Omega}_\infty$ on $\boldsymbol{\pi}_\infty$, and absolute conic's dual quadric $\mathbf{Q}^*_\infty$.

7.4.1 Plane at infinity

Corresponding to line at infinity $\mathbf{l}_\infty$ with canonical position $(0,0,1)^\top$ in 2D case, the plane at infinity $\boldsymbol{\pi}_\infty$ has the canonical position $(0,0,0,1)^\top$ in affine 3-space. It contains the directions $\mathbf{D} = (x_1, x_2, x_3, 0)$, and enables the identification of affine properties such as parallelism: two planes are parallel if and only if their intersection line is on $\boldsymbol{\pi}_\infty$.

Analogous to $\mathbf{l}_\infty$'s property shown Result 6.1, we have similar result for $\boldsymbol{\pi}_\infty$ that is fixed under 3D affinity:

Result 7.1. *The plane at infinity, $\boldsymbol{\pi}_\infty$, is a fixed plane under the projective transformation* $\mathbf{H}$ *if and only if* $\mathbf{H}$ *is an affinity.*

The proof is also similar to that of Result 6.1. Also similar to $\mathbf{l}_\infty$, $\boldsymbol{\pi}_\infty$ is only fixed planewise but not pointwise (that is, a point on $\boldsymbol{\pi}_\infty$ will be transformed to another point on $\boldsymbol{\pi}_\infty$).

Section 6.5.2 describes 2D affine rectification which transforms distorted $\mathbf{l}_\infty$ to its canonical position and finds the affinely rectified result image w.r.t. Result 6.1. Analogous to this, we can run 3D affine rectification by transforming distorted $\boldsymbol{\pi}_\infty$ to its canonical position, and then get affinely rectified result w.r.t. Result 7.1.

7.4.2 Absolute conic

Corresponding to circular points $\mathbf{I}, \mathbf{J}$ on $\mathbf{l}_\infty$, the absolute conic $\boldsymbol{\Omega}_\infty$ is a (point) conic on $\boldsymbol{\pi}_\infty$. Analogous to circular points' equation in Eq. (6.4), points on $\boldsymbol{\Omega}_\infty$ satisfy:

$$\begin{cases} x_1^2 + x_2^2 + x_3^2 = 0 \\ x_4 = 0 \end{cases} \tag{7.6}$$

which denotes a point conic because the first equation can be presented as

$$(x_1, x_2, x_3)\mathbf{I}(x_1, x_2, x_3)^\top = 0$$

so that $\boldsymbol{\Omega}_\infty$ corresponds to a conic $\mathbf{C}$ with matrix $\mathbf{C} = \mathbf{I}$. It is thus a conic of *purely imaginary points* on $\boldsymbol{\pi}_\infty$.

Analogous to circular points $\mathbf{I}, \mathbf{J}$ which are fixed under 2D similarity transformation in Result 6.2, a key property of $\boldsymbol{\Omega}_\infty$ is that it is a fixed conic under any 3D similarity transformation:

Result 7.2. *The absolute conic, $\boldsymbol{\Omega}_\infty$, is a fixed conic under the projective transformation* $\mathbf{H}$ ***if and only if*** $\mathbf{H}$ *is a similarity.*

Proof. The proof is also similar to that in Result 6.2. Firstly, since the absolute conic lies in the plane at infinity, a transformation fixing it must fix the plane at infinity, and hence must be affine. Such a transformation is of the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

Restricting to the plane at infinity, the absolute conic is represented by the matrix $\mathbf{I}_{3\times 3}$. According to Section 6.3.2, the transformed conic of is $\mathbf{A}^{-\top}\mathbf{I}\mathbf{A}^{-\mathbf{1}} = \mathbf{A}^{-\top}\mathbf{A}^{-\mathbf{1}}$, where $\mathbf{A}$ is the top-left 3×3 sub-matrix of $\mathbf{H}_{4\times 4}$. Therefore, if $\boldsymbol{\Omega}_\infty$ is fixed, then $\mathbf{A}^{-\top}\mathbf{A}^{-\mathbf{1}} = \mathbf{I}$, which means $\mathbf{A}^{-1}$ is orthogonal, then $\mathbf{A}$ is a rotation matrix, i.e., $\mathbf{H}$ is similarity. Obviously, the contrary is also correct. □

Similar to 2D line angle defined in Eq. (6.8), we have 3D line angle defined as

$$\cos\theta = \frac{\mathbf{d}_1^\top \boldsymbol{\Omega}_\infty \mathbf{d}_2}{\sqrt{(\mathbf{d}_1^\top \boldsymbol{\Omega}_\infty \mathbf{d}_1)(\mathbf{d}_2^\top \boldsymbol{\Omega}_\infty \mathbf{d}_2)}}, \tag{7.7}$$

where $\mathbf{d}_1$ and $\mathbf{d}_2$ are 3D line directions.

7.4.3 Absolute dual conic

The dual of the absolute conic $\boldsymbol{\Omega}_\infty$ is a degenerate dual quadric in 3-space called the absolute dual conic $\mathbf{Q}_\infty^*$. Geometrically, $\mathbf{Q}_\infty^*$ consists of the **planes tangent** to $\boldsymbol{\Omega}_\infty$, so that $\boldsymbol{\Omega}_\infty$ is the "rim" of $\mathbf{Q}_\infty^*$. This is called a *rim quadric*. Algebraically, $\mathbf{Q}_\infty^*$ is represented by a 4×4 homogeneous matrix of rank 3, which in metric 3-space has the canonical form

$$\mathbf{Q}_\infty^* = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}. \tag{7.8}$$

This is analogous to $\mathbf{C}_\infty^*$, the dual conic to 2D circular points in Eq. (6.6).

Section 3.7 (Page 84) in Ref. [1] shows why $\mathbf{Q}_\infty^*$ is the set of planes tangent to $\boldsymbol{\Omega}_\infty$, and also gives another view of the meaning of absolute dual quadric.

Analogous to Result 6.3, the absolute dual quadric is fixed under similarity:

Result 7.3. *The absolute dual quadric, $\mathbf{Q}_\infty^*$ is fixed under the projective transformation $\mathbf{H}$ if, and only if, $\mathbf{H}$ is a similarity.*

Proof. The proof is similar to Result 6.3. The absolute dual quadric can be fixed if and only if $\mathbf{Q}_\infty^* = \mathbf{H}\mathbf{Q}_\infty^*\mathbf{H}^\top$ according to Eq. (7.5). So, by applying an arbitrary transformation

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^\top & k \end{bmatrix}$$

and expand the equation, we can ensure that $\mathbf{A}$ is rotation matrix and $\mathbf{v} = \mathbf{0}$. That is, $\mathbf{H}$ is a similarity. □

Result 7.4. *The plane at infinity $\boldsymbol{\pi}_\infty$ is the null-vector of $\mathbf{Q}_\infty^*$.*

This result holds in any coordinate frame. Firstly when $\mathbf{Q}_\infty^*$ has canonical form in Eq. (7.8 and $\boldsymbol{\pi}_\infty = (0,0,0,1)^\top$, clearly $\mathbf{Q}_\infty^*\boldsymbol{\pi}_\infty = \mathbf{0}$. Then, in any frame under point transformation $\mathbf{H}$, we have $\mathbf{Q}_\infty^{*\prime} = \mathbf{H}\mathbf{Q}_\infty^*\mathbf{H}^\top$ and $\boldsymbol{\pi}' = \mathbf{H}^{-\top}\boldsymbol{\pi}$, so $\mathbf{Q}_\infty^{*\prime}\boldsymbol{\pi}' = \mathbf{H}\mathbf{Q}_\infty^*\boldsymbol{\pi} = \mathbf{0}$.

Similar to line conic, we have some result related to angle between planes:

Result 7.5. *The angle between two planes $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ in any frame is given by*

$$\cos\theta = \frac{\boldsymbol{\pi}_1^\top \mathbf{Q}_\infty^* \boldsymbol{\pi}_2}{\sqrt{(\boldsymbol{\pi}_1^\top \mathbf{Q}_\infty^* \boldsymbol{\pi}_1)(\boldsymbol{\pi}_2^\top \mathbf{Q}_\infty^* \boldsymbol{\pi}_2)}} \tag{7.9}$$

The proof is easy by using similar transformation rule of plane and absolute dual quadric. In a Euclidean frame, Eq. (7.9) is reduced to the common form we know using plane normals.

8 References

[1] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] Link: conversion from rotation matrix to axis-angle format. `https://en.wikipedia.org/wiki/Rotation_matrix#Conversion_from_and_to_axis.E2.80.93angle`.

[3] Link: quaternions and spatial rotation. `https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation`.

[4] Link: video tutorial of gimbal lock. `https://www.youtube.com/watch?v=zc8b2Jo7mno`.