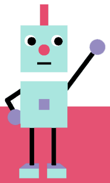


SIGGRAPH ASIA 2021 Course Notes

Opening the Black Box of Mathematics for CG

Hiroyuki Ochiai
Ken Anjyo
Ayumi Kimura (edit.)

SIGGRAPH ASIA 2021 TOKYO LIVE

Table of Contents

From the course proposal:

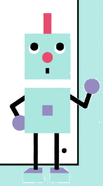
Mathematics is recognized as common basis for CG technology, but sometimes used as a black box in a CG software tool. We expect better understanding of maths will conduce to not only better CG tools, but also innovative ideas for a future production pipeline. The goal of this course is to pull the trigger for the graphics people to know more about usefulness and fun of the maths behind the scenes.

We therefore select a few typical CG topics ranging from elementary to standard levels, so that the course attendees can easily access the course content. We don't assume that the attendees have familiarity with highly advanced mathematics. Elements of linear algebra and calculus at undergraduate level would be enough.

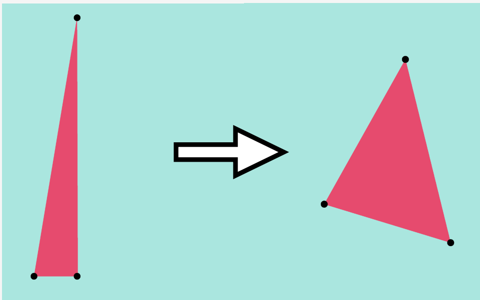
More specifically, we start with homogeneous coordinates, affine transformations and quaternions. These are common and basic mathematical concepts for CG. Matrix exponential and logarithm are then discussed for curve/surface editing, deformation and animation of geometric objects. Eigenvalues and eigenvectors are also well known mathematical concepts, yet appear with different faces in various graphics applications. In this course we intend to give a unified mathematical scope of these concepts. The corresponding graphics topics then include principal curvatures in geometry, PCA and Bayesian Inference in statistical approaches for animation, and

Spherical Harmonics for image-based rendering. As an advanced topic, we briefly describe the mathematical concept called duality. Dual quaternion and theory of distribution will then be explained as mathematical basis of interpolation techniques for computer animation.

A unique feature of this course is that we demonstrate most of the mathematical concepts without rigorous formulation, while first showing their graphics applications. We expect this makes it easy to understand the mathematics mentioned above and to open the door for more advanced mathematical approaches.

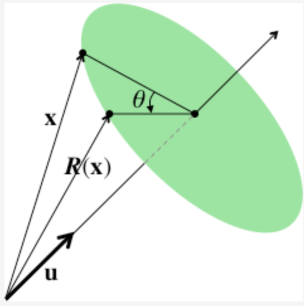


BACKGROUND



Homogeneous Coordinates

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} & A & & b_1 \\ & & & b_2 \\ & & & b_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$



3D axis rotation and quaternions

$$R\mathbf{x} = (\cos\theta)\mathbf{x} + (\sin\theta)(\mathbf{u}\times\mathbf{x}) + (1-\cos\theta)(\mathbf{u}\cdot\mathbf{x})\mathbf{u}$$

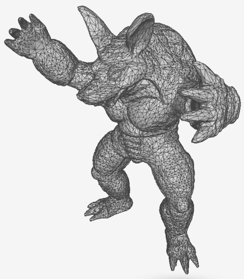
$$slerp(\boldsymbol{q}_0, \boldsymbol{q}_1, t) := \frac{\sin(1-t)\phi}{\sin\phi}\boldsymbol{q}_0 + \frac{\sin t\phi}{\sin\phi}\boldsymbol{q}_1$$

BACKGROUND

- Mathematics plays a key role in various CG fields
- However maths is sometimes in the black box of the CG tools
  - Used implicitly
  - Not sure what maths is actually utilized
  - Why so useful?

BACKGROUND

Geometry

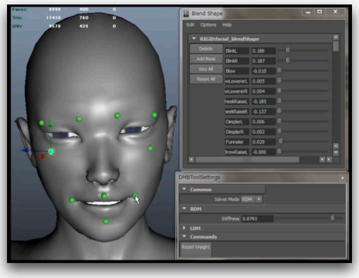


Curvature

$$H = \frac{\lambda_1 + \lambda_2}{2}$$

$$K = \lambda_1 \cdot \lambda_2$$

Animation (Example-based)



Covariance Matrix

$$C_{\mathbf{X}} = \langle (\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{X} - \mu_{\mathbf{X}})^T \rangle$$

$$Q^T C_{\mathbf{X}} Q = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix}$$

Rendering (Image-based)

Spherical Harmonics

$$L_{\mathrm{env}}(\omega') \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} L_\ell^m y_\ell^m(\omega')$$

$$\Lambda y_\ell^m = \lambda_\ell y_\ell^m,$$
$$\lambda_\ell = -\ell(\ell+1).$$

Eigenvalue "λ" everywhere in CG!

AIM AND SCOPE

Topics focused on

- Basic geometry and algebra - affine transform, rotation and quaternions
- CG application of eigenvalues, eigenvectors and eigenfunctions

Goal of this course

- Clarify which maths is used in practice
- Explain why so useful
- Guide to more advanced topics

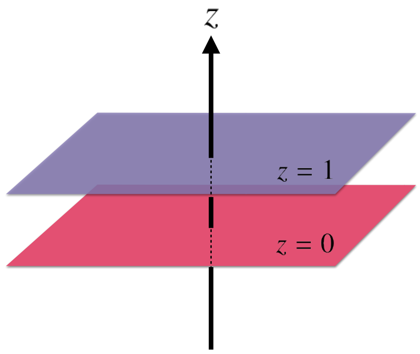
Affine transformation and Homogeneous coordinates

- Affine transformation is a composition of a linear transformation and a translation. In 3D, it is natural to use a vector with three components and a matrix of size three.

- However, in this picture we have to deal a linear transformation and a translation separately. If we introduce one-size-more, that is, a vector with four components and a matrix of size four, we can integrate both a linear transformation and a translation in a unified manner.

- This technique directly related with the concept of homogenous coordinates.

Inhomogeneous coordinate plane



$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + a \\ y + b \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x + a \\ y + b \\ 1 \end{pmatrix}$$

- A translation on a plane is expressed by matrix.
- Trick is to 'add one extra dimension'.
- 3D matrix multiplication restricted on a plane $z = 1$.

Relation between homogeneous and inhomogeneous coordinates

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = t \begin{pmatrix} \alpha \\ \beta \\ 1 \end{pmatrix}$$

$$\alpha = \frac{x}{z}, \ \beta = \frac{y}{z}$$

- $[x : y : z]$: homogeneous coordinates
- (α, β) : inhomogeneous coordinates

Rigid motion in 3D

Matrix expression of rigid motion in homogeneous coordinates

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Matrix expression of rigid motion in inhomogeneous coordinates

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \left(\begin{array}{ccc|c} & & & x \\ & R & & b \\ & & & c \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Example: Use of homogeneous coordinates in description of 3D rigid motion

- R : 3D rotation matrix
- $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$: translation vector(*)
- $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$: 3D vector, inhomongeneous coordinates (= usual expression)
- $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$: 3D vector, homogenous coordinates

* In the course notes, we sometimes denote a column vector $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ by $(a, b, c)^{\mathrm{T}}$

Affine transformation in 3D

Matrix expression of affine transformation
in homogeneous coordinates

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = A \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Matrix expression of rigid motion
in inhomogeneous coordinates

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \left(\begin{array}{ccc|c} & & & x \\ & A & & b \\ & & & c \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Example: Use of homogeneous coordinates for 3D affine transformation

- A : 3D regular matrix, i.e., $\det A \neq 0$
- $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$: translation vector
- $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$: 3D vector, inhomogeneous coordinates (= usual expression)
- $\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$: 3D vector, homogeneous coordinates

Example: A Homogeneous Coordinate in Pose Space Deformation

$$\begin{cases} v_1, \ldots, v_k \in \mathbb{R}^3 & \text{: points} \\ f_1, \ldots, f_k \in \mathbb{R} & \text{: positive} \end{cases}$$

Want to define the average of $v_1, \ldots, v_k$

Skeletal Subspace Deformation

Pose Space Deformation

Method 1 : inhomogenous coordinates

$$w_i = \frac{f_i}{\sum_{k=1}^{K} f_k} \text{ : weight, such that } \sum_{i=1}^{K} w_i = 1$$

$$\text{The average : } = \sum_{k=1}^{K} w_k v_k = \frac{\sum_{k=1}^{K} f_k v_k}{\sum_{k=1}^{K} f_k}$$

Method 2 : homogenous coordinates

$$\begin{pmatrix} v_1 \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} v_K \\ 1 \end{pmatrix} \in \mathbb{R}^4$$

linear combination

$$f_1 \begin{pmatrix} v_1 \\ 1 \end{pmatrix} + \cdots + f_K \begin{pmatrix} v_K \\ 1 \end{pmatrix} \in \mathbb{R}^{3+1}$$

: The average in homog coord

$$\parallel$$

$$\begin{pmatrix} f_1 v_1 + \cdots + f_K v_K \\ f_1 + \cdots + f_K \end{pmatrix} \in \mathbb{R}^{3+1}$$

$$\parallel$$

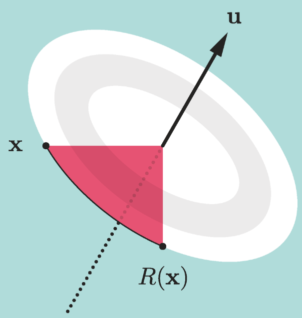
in homog coord $\begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} = \begin{pmatrix} x/t \\ y/t \\ z/t \\ 1 \end{pmatrix}$

$$\begin{pmatrix} \frac{\sum_{k=1}^{K} f_k v_k}{\sum_{k=1}^{K} f_k} \\ 1 \end{pmatrix} \in \mathbb{R}^{3+1}$$

Rotation, Quaternion,
Lie groups and Lie algebra

- 3D rotation often arises in CG.
- Quaternion is used for description of 3D rotations.
- The totality of rotations is a typical example of a Lie group.
- Lie algebra and the exponential map is a powerful device to deal with a Lie group.

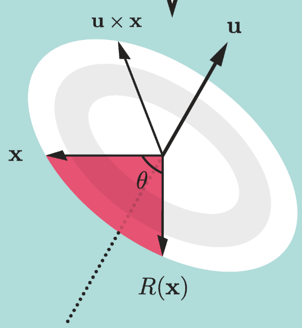
3D ROTATION: AXIS-ANGLE



- There are several ways to describe 3D rotation.
- 3D rotation can be specified by the rotation axis (= normal vector to rotation plane) and the rotation angle.
- The number of essential parameters is three.

3D ROTATION: AXIS-ANGLE

$$R(\mathbf{x}) = (\mathbf{u}\cdot\mathbf{x})\mathbf{u} + (\sin\theta)(\mathbf{x} - (\mathbf{u}\cdot\mathbf{x})\mathbf{u}) + (\cos\theta)(\mathbf{u}\times\mathbf{x})$$



- We can take an orthonormal basis on the rotation plane.
- We have an explicit 2D-like expression with these basis.
- This is one version of Rodrigues formula.

Quaternion and 4D/3D rotation

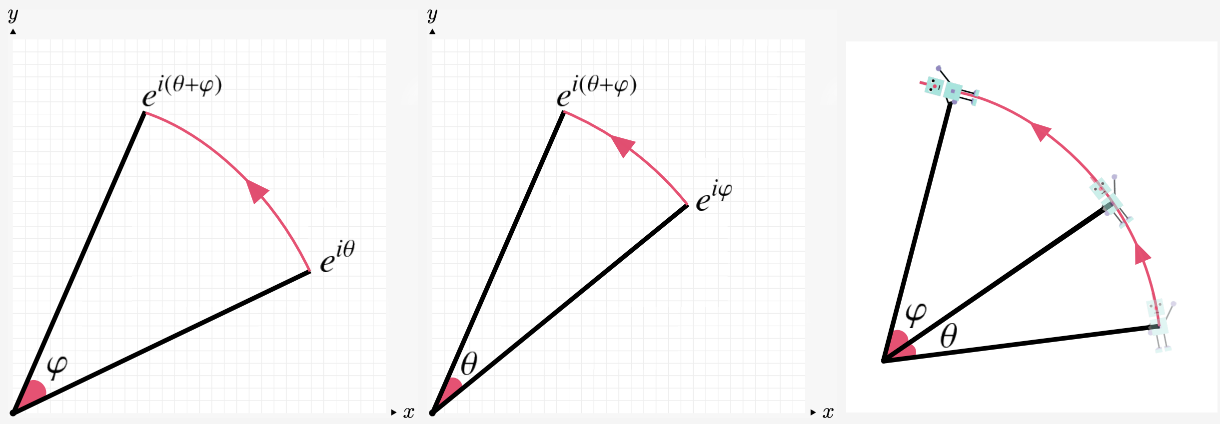
4D Rotation	$\mathbf{q} \mapsto \mathbf{pqr}$ $\mathbf{q} \in \mathbb{H}$ $\mathbf{p}, \mathbf{r} \in \mathbb{H}^{\times}$
3D Rotation	$\mathbf{q} \mapsto \mathbf{pqp}^{-1}$ $\mathbf{q} \in \mathrm{Im}\mathbb{H}$ $\mathbf{p} \in \mathbb{H}^{\times}$

$\mathbf{q}$: rotated vector

$\mathbf{p}, \mathbf{r}$: rotator

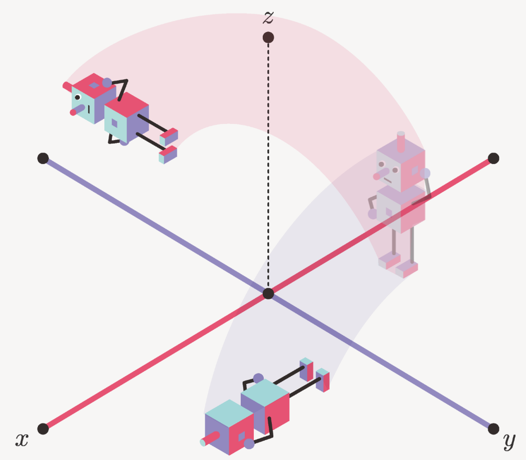
- Quaternion can express the 4D rotation as well as 3D rotation.
- 3D rotation is a special form of 4D rotation

Both points and rotations are complex numbers



- A formula $e^{i\theta}e^{i\varphi} = e^{i(\theta+\varphi)}$ of complex numbers has three geometric interpretation.
  - Rotate a point $e^{i\theta}$ by φ degree.
  - Rotate a point $e^{i\varphi}$ by θ degree.
  - The composition of the rotation by θ and that by φ.

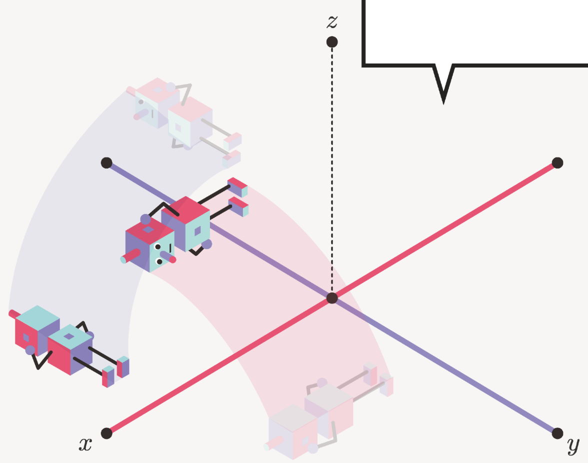
NON-COMMUTATIVITY OF 3D ROTATIONS



- While a composition of 2D rotations does not depend on the order, a composition of 3D rotations depends on the order.
- Non-commutative
- We here consider the composition of the rotation with respect to x-axis and that to y-axis.

NON-COMMUTATIVITY OF 3D ROTATIONS

$$R_x(\theta)R_y(\varphi) \neq R_y(\varphi)R_x(\theta)$$



- This is a main differences in 2D and 3D rotations.
- This non-commutativity causes complication theoretically and computationally.
- This non-commutativity makes our life interesting. 😄

EXPONENTIAL : TAYLOR EXPANSION

$$e^x = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \cdots$$

- Taylor series expansion of exponential function uses
  - power,
  - scalar multiple,
  - summation, and
  - convergence of infinite series
- These for operations are valid for matrix;
  Taylor expansion is the definition of matrix exponential.

EXPONENTIAL : TAYLOR EXPANSION

MATRIX EXPONENTIAL

$$\exp(A) = I + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 + \cdots$$

EXPONENTIAL : ROTATION

$$\exp\begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} = \begin{pmatrix} 1 - \frac{1}{2}\theta^2 + \frac{1}{4!}\theta^4 - \cdots & -\theta + \frac{1}{3!}\theta^3 - \frac{1}{5!}\theta^5 + \cdots \\ \theta - \frac{1}{3!}\theta^3 + \frac{1}{5!}\theta^5 - \cdots & 1 - \frac{1}{2}\theta^2 + \frac{1}{4!}\theta^4 - \cdots \end{pmatrix}$$

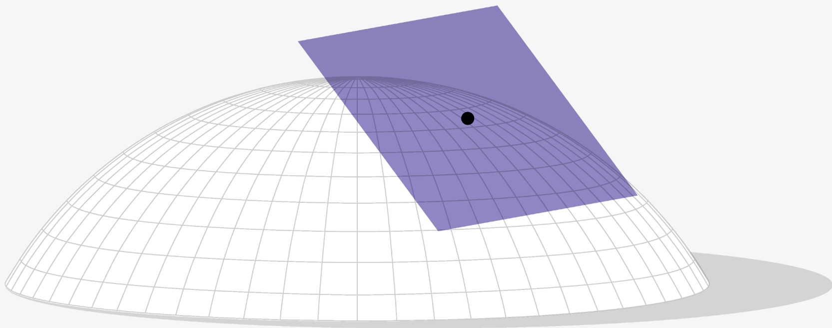
EXPONENTIAL : ROTATION

EXPONENTIAL : ROTATION

$$\exp\begin{pmatrix}0 & -\theta\\ \theta & 0\end{pmatrix} = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$$

- For a matrix with imaginary eigenvalues,
  matrix exponential is reduced to Taylor series of trigonometric function
- A rotation matrix can be regarded as an example of matrix exponential.
- Exponential law shows the additivity of angle variables.

LIE ALGEBRA



- A manifold is a fancy name of curved space
- A matrix group is a curved space with a group structure: it's called a Lie group.
- A tangent space is a linear approximation of a curved space
- A tangent space of Lie group is called Lie algebra.
- It is significant that the Lie algebra exactly has all the structure of Lie groups locally; there is no loss by a linear approximation.

For more details, refer to the [Anjyo17]

Summery: 3D rigid motion

- homogeneous coordinates ▶ treat both rotation and translation simultaneously.
- Lie group ▶ the totality of rotations has group structure.
- Lie algebra ▶ the linear approximation extra-ordinarily efficient.
- exponential map ▶ its many properties helps both theoretically and computationally.

MEMO

What and why linear?

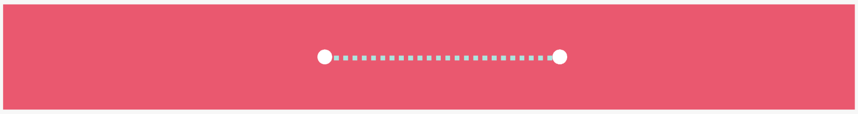
Linear structure is strong enough.

A linear interpolation can make a small number of data into a huge number of data.

An eigenvalue and eigenvector can analyze a large number of data N^2 by using a very small number of data, and find a feature of shape, phenomena, and control.

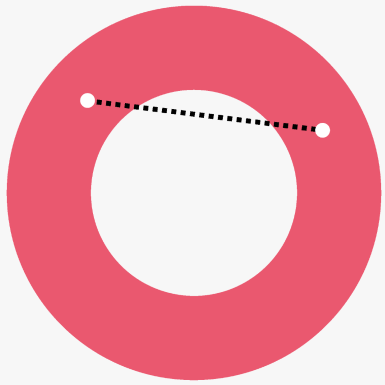
We will see these by example later.

FIELD OF BLENDING



- A linear interpolation is one of most used and simple method. For example, this is a background in keyframe animation.
- A linear interpolation is natural, fast, and efficient in most cases.
- A linear interpolation depends on the linear structure of the ambient space, the set of data.

FIELD OF BLENDING



- In most cases, naive values of position of points, velocity, density, RGB, etc. are used in interpolation.
  However, sometimes these parameters are not so appropriate for interpolation.

- Artifact: for a curved space, a linear interpolation of two objects is located outside of a space.
  For example, a linear interpolation of two rotations is not a rotation; it is not of determinant one.

FIELD OF BLENDING

- A linear interpolation is one of most used and simple method.
- Artifact: for a curved space, an linear interpolation of two objects is located outside of a space.
- If we can linearlize a curved space, we can use a linear interpolation for an interpolation in a curved space.
- Matrix exponential and its inverse (called matrix logarithm) give a linearization of a Lie group (curved space) into a Lie algebra (flat space).

SLERP

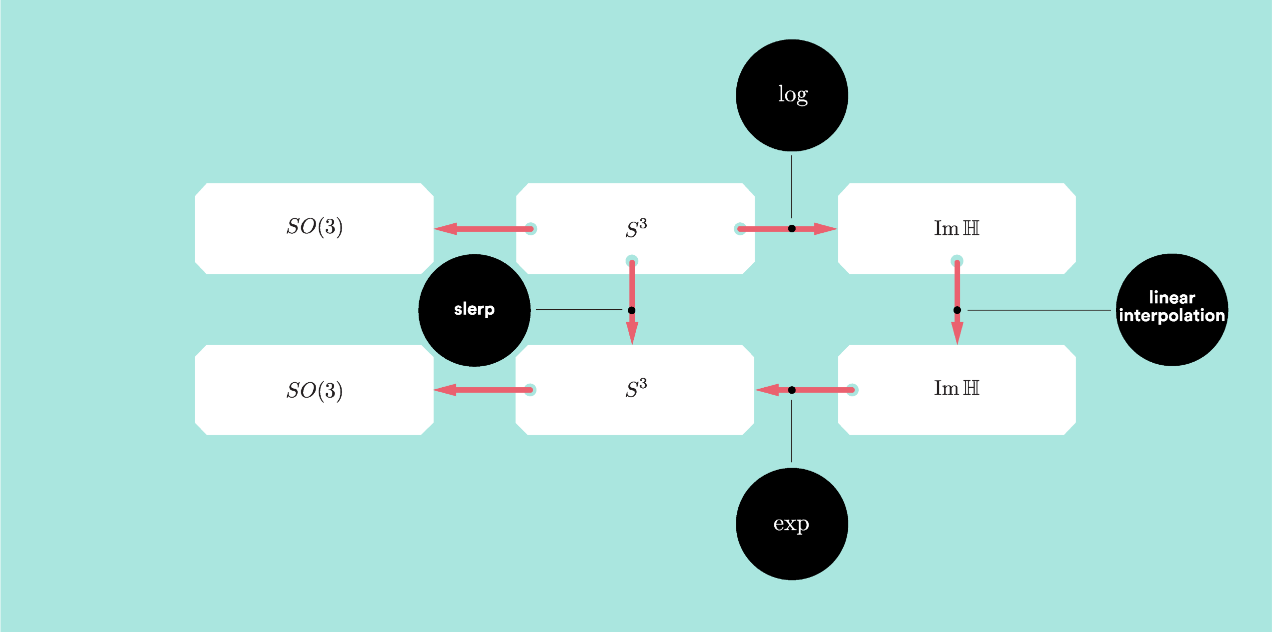


Diagram explanation of slerp

- **slerp is the middle vertical arrow.**
  This map is the composition of three arrows; log, linear-interpolation, and exp.
  In other words, if we sandwich slerp by log and exp, then it is reduced to linear interpolation, which is as simple as possible.

- **slerp lives in the quaternion world.**
  We can move to the rotation by the arrow from the middle to the left.

SLERP

- This diagram simply shows that

  Slerp = exp ○ (linear interpolation) ○ ***log***

- It also explains the following formula:

for unit quaternions

$$\mathrm{slerp}(q_0, q_1, t) = \frac{\sin((1-t)\theta)}{\sin\theta} q_0 + \frac{\sin(t\theta)}{\sin\theta} q_1$$

$$q_0, q_1 \in \mathbb{S}^3$$

because we have:

$$\mathrm{slerp}(q_0, q_1, t) = \mathrm{slerp}(1, q_1 q_0^{-1}, t) q_0$$

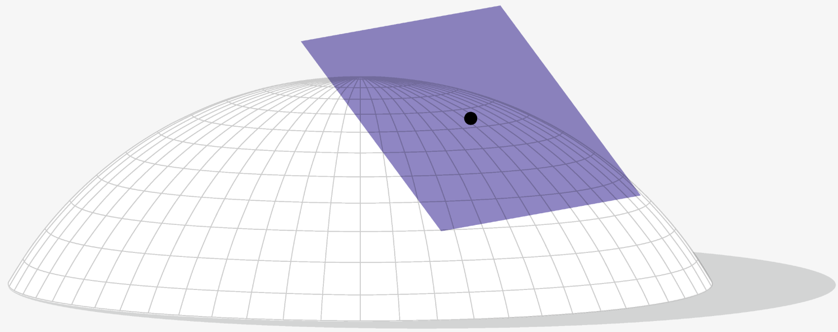
$$\mathrm{slerp}\ (1, \exp(\theta\mathbf{u}), t) = \exp(t\theta\mathbf{u})$$

- The formula in the last line shows that slerp is considered to be a linear interpolation $t\theta\mathbf{u}$ inside the exponential function.

- The second last formula shows another property of slerp: equvariance.

- If we move two given points simultaneously by some rotation,
  then the interpolated point also moves by exactly the same rotation.
  This reduces the slerp map from two input into one input.

Lie group and Lie algebra



- It is not realistic to draw a picture of a group, however, geometrically, a group is considered to be a smooth curved space, like a sphere. (The correct terminology in math is manifold.)

  In general, a curved space is approximated by a linear space, called a tangent space.

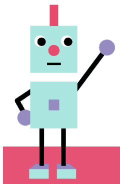
- Dual number is convenient for computation of Lie algebra.

Interpolation on a curved space

- Matrix exponential and its inverse (called a matrix logarithm) give a linearization of a Lie group (curved space) into a Lie algebra (flat space, vector space).
- In other words, the exponential map gives a parametrization of a curved space by a linear space, and this parametrization enables us to interpolate linearly.
  This simple idea widely spreads to an extension of interpolation.

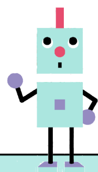
It is efficient that:

- Use the linear structure of data.
  Find a good linear structure for a curved space.
- Note that a natural linear structure may not reflect a good linear structure.
  For example, a linear structure of a matrix space is not appropriate for a linear structure on rotation matrices.



Eigenvalues, Eigenvectors and Eigenfunctions

- Where do eigenvalues appear in CG?
- Eigenvalues and eigenvectors in statistical data analysis and differential geometry.
- Eigenfunctions in Image-based rendering.



What are "eigenvalue" and "eigenvector"??

Let's start with the definition for a matrix case:

Assume that, for a given square matrix A, there exist non-zero vector $\mathbf{u}_i$ and a (real or complex) number λ_i which satisfy the following equation:

$$\boxed{A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \ \text{for } 1 \leq i \leq m}$$

Then we refer to λ_i as an *eigenvalue* of A and $\mathbf{u}_i$ as *eigenvector*.

- *The definition is very clear! But hard to imagine what it actually means…*
- *Let's consider a bit deeper mathematical aspect of eigenvalue and eigenvector.*
  - *Of course let's see what they act in our graphics applications!!*

EIGENVALUES AND EIGENVECTORS IN STATISTICAL APPROACHES

Covariance matrix:

$$C_{\mathbf{X}} = \langle (\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{X} - \mu_{\mathbf{X}})^T \rangle$$

Gaussian distribution:

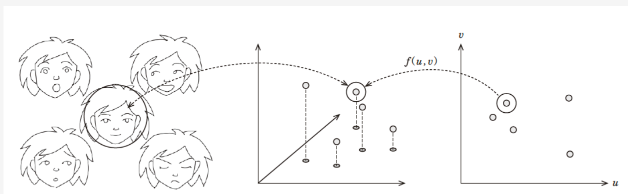
$$p(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi}}\right)^m \frac{1}{\sqrt{\det C_{\mathbb{X}}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_{\mathbb{X}})^T C_{\mathbb{X}}^{-1} (\mathbf{x} - \mu_{\mathbb{X}})\right\}$$

$$U^T C_{\mathbb{X}} U = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix}$$

Bayesian Inference

Direct manipulation blendshapes with prior data

Making variations with PCA



From left: 2D drawings (input); high dim space of the drawings; Reducing the dim to 2D.
See Baxter and Anjyo "Latent Doodle Space" , EUROGRAPHICS 2006.

BASICS IN PROBABILISTIC/STATISTICAL APPROACHES

We assume that the probability $\mathbf{p}(A)$ for an event A in the whole space Ω to occur is assigned, where $\mathbf{p}$ satisfies the following properties, for any A and B in Ω ($A \cap B = \phi$),

$$0 \leq \mathbf{p}(A) \leq 1 \text{ and } \mathbf{p}(\Omega) = 1,$$
$$\mathbf{p}(A \cup B) = \mathbf{p}(A) + \mathbf{p}(B).$$

We then note that $\mathbf{p}(A^C) = 1 - \mathbf{p}(A)$, where $A^C = \Omega \setminus A$ denotes the complement of A in Ω.

Let $\mathbf{X}$ be a stochastic variable. We define the probability distribution function $P_{\mathbf{X}}(x): \mathbb{R} \to [0, 1]$ by

$$P_{\mathbf{X}}(x) = \mathbf{p}(\mathbf{X} \leq x).$$

If $P_{\mathbf{X}}(x)$ is differentiable, we then have:

$$p_{\mathbf{X}}(x) := \frac{dP_{\mathbf{X}}}{dx} = \lim_{\Delta x \to 0} \frac{P_{\mathbf{X}}(x + \Delta x) - P_{\mathbf{X}}(x)}{\Delta x} \equiv \lim_{\Delta x \to 0} \frac{\mathbf{P}(x < X \leq x + \Delta x)}{\Delta x}.$$

$p_{\mathbf{X}}(x)$ is referred to as the *probability density function* of $\mathbf{X}$: $p_{\mathbf{X}}(x) \geq 0$ and $\int_{-\infty}^{+\infty} p_{\mathbf{X}}(x)dx = 1$.

$$\mathbf{p}(x_1 < \mathbf{X} \leq x_2) = \int_{x_1}^{x_2} \frac{dP_{\mathbf{X}}}{dx}(x)dx.$$

Muti-dimensional stochastic variable

Similarly to the stochastic variable X, we can consider a multi-dimensional stochastic variable $\mathbb{X} = (\mathrm{X}_1, \mathrm{X}_2, \ldots \mathrm{X}_\mathrm{m})^\mathrm{T}$ and its probability density function:

Jointly probability distribution function of $\mathbb{X}$ $P_{\mathbb{X}}(x_1, x_2, \cdots, x_m) := \mathbf{p}(X_1 \leq x_1, X_2 \leq x_2, \cdots, X_m \leq x_m)$

Probability density function of $\mathbb{X}$ $p_{\mathbb{X}}(x_1, x_2, \cdots, x_m) := \frac{\partial^m P_{\mathbb{X}}}{\partial x_1 \partial x_2 \cdots \partial x_m}(x_1, x_2, \cdots, x_m).$

Let $g(\mathbb{X})$ be a (scalar/vector-valued) function of $\mathbb{X}$. The mean value of $g(\mathbb{X})$, denoted by $\langle g(\mathbb{X}) \rangle$ is defined as:

$$\langle g(\mathbb{X}) \rangle := \int_{\mathbb{R}^m} g(x_1, x_2, \cdots, x_m) p_{\mathbb{X}}(x_1, x_2, \cdots, x_m) dx_1 dx_2 \cdots dx_m$$

If we put $g(\mathbb{X}) = \mathbb{X}$ for example, we get the mean value of $\mathbb{X}$, denoted by $\mu_{\mathbb{X}}$ as:

$$\mu_{\mathbb{X}} := \langle \mathbb{X} \rangle = (\int_{\mathbb{R}^m} x_1 p(\mathbf{x}) d\mathbf{x}, \int_{\mathbb{R}^m} x_2 p(\mathbf{x}) d\mathbf{x}, \cdots, \int_{\mathbb{R}^m} x_m p(\mathbf{x}) d\mathbf{x})^T$$

where we put $\mathbf{x} = (x_1, x_2, \cdots, x_m)$ and $d\mathbf{x} = dx_1 dx_2 \cdots dx_m$ for convenience.

Covariance matrix

For a given $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_m)^T$, we define the *covariance* matrix:

$$C_{\mathbb{X}} := \langle (\mathbb{X} - \mu_{\mathbb{X}})(\mathbb{X} - \mu_{\mathbb{X}})^T \rangle$$

The covariance matrix $C_{\mathbb{X}}$ is an $m \times m$ real-valued square matrix, which is symmetric and positive semi-definite, i.e., all the *eigenvalues* of $C_{\mathbb{X}}$ are non-negative real numbers.

Putting $A = C_{\mathbb{X}}$ for a moment, we have:

$$\boxed{A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \ \text{for } 1 \leq i \leq m}$$

where $\lambda_i \geq 0$ is an eigenvalue with its eigenvector $\mathbf{u}_i$ such that $\langle \mathbf{u}_j, \mathbf{u}_k \rangle = \delta_{jk}$ for $1 \leq j, k \leq m$.

It then follows that $AU = U \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix}$, where $U = (\mathbf{u}_1 \mathbf{u}_2 \cdots \mathbf{u}_m)$.

$$\Rightarrow U^{-1}AU = U^T AU = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix} \equiv \Lambda$$

Gaussian distribution

Definition $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_m)^T$ is called *Gaussian*, if its probability density function is given by:

$$p(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi}}\right)^m \frac{1}{\sqrt{\det C_{\mathbb{X}}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_{\mathbb{X}})^T C_{\mathbb{X}}^{-1} (\mathbf{x} - \mu_{\mathbb{X}})\right\}$$

where the covariance matrix $C_{\mathbb{X}}$ positive definite, i.e., all the *eigenvalues* λ_i are positive.

Proposition Assume that $\mathbb{X}$ is *Gaussian*. If we change the stochastic variable from $\mathbb{X}$ to

by $\mathbb{Z} = U^T(\mathbb{X} - \mu_{\mathbb{X}})$, the probability density function of $\mathbb{Z}$ is:

$$p(\mathbf{z}) = \left(\frac{1}{\sqrt{2\pi}}\right)^m \frac{1}{\sqrt{\lambda_1 \cdots \lambda_m}} \exp\left\{-\frac{1}{2}\mathbf{z}^T \Lambda^{-1} \mathbf{z}\right\}$$

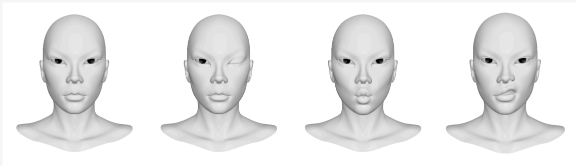
$$\Lambda = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix}$$

APPLICATION: DIRECT MANIPULATION BLENDSHAPES (DMB)

Blendshape facial animation is known as a standard practice in digital production workplace.

Our blendshape model will be described with: $\mathbf{f} = \mathbf{B}\mathbf{w} + \mathbf{f}_0$

- $\mathbf{f}$ is a $3n$ dimensional vector containing the components of each of the n vertices or control points on the face vectorized in some arbitrary order such as xyzxyzxyz... and $_0$ is the neutral shape in similarly vectorized form.
- $\mathbf{B} \in \mathbb{R}^{3n\times p}$ contains the p blendshape targets $\mathbf{f}_i$ so that the i-th column of $\mathbf{B}$, $\mathbf{b}_i$, is given by $\mathbf{b}_i = \mathbf{f}_i - \mathbf{f}_0$ for $1 \le i \le p$, i.e., $\mathbf{B} = (\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_p)$.
- $\mathbf{w} \in \mathbb{R}^p$ are the blendshape weights.



Neutral face f_0 (leftmost) and target face examples

The position-constraint problem in DMB

To reduce the task of tuning the weight (vector) parameter $\mathbf{w}$, we introduce the drag-and-drop operations for artists to edit the face geometry and then automatically estimate the weight that prescribes the edited face geometry.

More specifically, our edit process is described as follows:

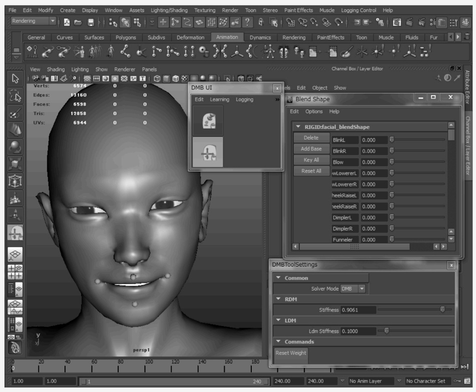
1. Let $\mathbf{w}_b$ be the weight vector of the current face $\mathbf{f}_b$ (before edit): $\mathbf{f}_b = \mathbf{B}\mathbf{w}_b + \mathbf{f}_0$.
2. Let $\mathbf{f}_{ed}$ be the edited face: $\mathbf{f}_{ed} := \mathbf{d} + \mathbf{f}_0$, which is obtained from a "drag & drop" operation.
3. **[position-constraint problem]**

Find $\mathbf{w}_a$ the weight for the edited face geometry:

$\mathbf{f}_a = \mathbf{B}\mathbf{w}_a + \mathbf{f}_0$. This is done by minimizing:

$$\|\overline{\mathbf{f}}_a - \overline{\mathbf{f}}_{ed}\| = \|\overline{\mathbf{B}}\mathbf{w}_a - \overline{\mathbf{d}}\|$$

where $\overline{\mathbf{f}}$ denotes a vector consisting of the position-constrained *vertices* of $\mathbf{f}$, which are specified in step 2.



Solving the position-constraint problem - (1) Statistical analysis of the prior data

- We have lots of prior data of motion capture or animation archives.
- We want to accelerate the edit process by "learning" the prior data.

Let $\mathbb{X} \equiv \{\mathbf{f}^i\}_{i=1}^{\mathfrak{n}_f}$ be the prior data (expressed by blendshapes), where $\mathfrak{n}_f$ means the number of frames. Consider the covariance matrix $C_{\mathbb{X}}$, its eigenvalues $\{\lambda_j\}$ and eigenvectors $\{\mathbf{u}_j\}$:

$$U^T C_{\mathbb{X}} U = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_{3n} \end{pmatrix} \quad \text{and} \quad \mathbf{U} = (\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_{3n}),$$

where we have $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{3n} \geq 0$. Then we may assume:

$$\lambda_j > 0 \ (1 \leq j \leq k_r) \text{ and } \lambda_k = 0 \text{ for any } k > k_r, \text{ where } k_r := rank(C_{\mathbb{X}}).$$

Solving the position-constraint problem - (2) PCA-like solver

Solve the problem by regularization (adding user-specified regularization parameter α):

$$\min_{\mathbf{w}} \|\overline{\mathbf{B}}\mathbf{w} - \overline{\mathbf{d}}\|^2 + \alpha\|\mathbf{w}\|^2$$

We find the solution in the form: $\mathbf{B}\mathbf{w} + \mathbf{f}_0 = \mathbf{U}_q\mathbf{c} + \mathbf{e}_0$, where $\mathbf{c} = (c_1, c_2, \cdots, c_q)^T \in \mathbb{R}^q$ and $\mathbf{U}_q = (\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_q)$ for $1 \leq q \leq k_r$.

Note that the coefficient vector $\mathbf{c} = (c_1, c_2, \cdots, c_q)^T$ behaves as q-dim Gaussian:

$$p(\mathbf{c}) = \left(\frac{1}{\sqrt{2\pi}}\right)^q \frac{1}{\sqrt{\lambda_1 \cdots \lambda_q}} \exp\left\{-\frac{1}{2}\mathbf{c}^T \Lambda^{-1} \mathbf{c}\right\}$$

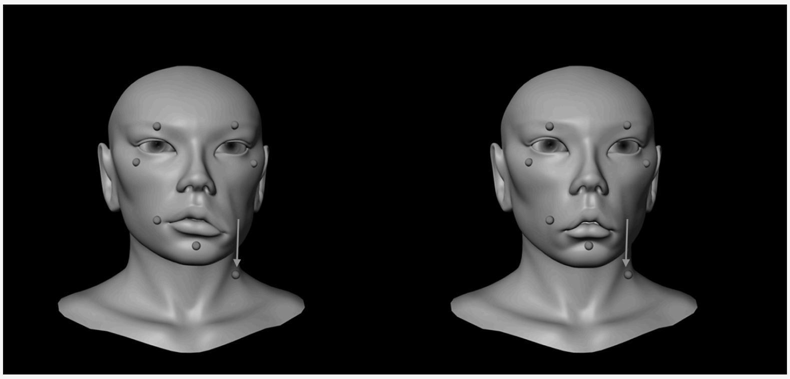
The position-constraint problem is reduced to:

$$\min_{\mathbf{c}} \|\overline{\mathbf{U}}_q\mathbf{c} - \overline{\mathbf{d}}\|^2 + \beta\|\mathbf{c}\|^2_{\Lambda_q}, \text{ where } \|\mathbf{c}\|_{\Lambda_q} = \sqrt{\sum_{k=1}^{q} \frac{c_k^2}{\lambda_k}}.$$

We then have $\mathbf{w}$ from $\mathbf{c}$ (or vice versa) via:

$$\mathbf{w} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{U}_q\mathbf{c} + \mathbf{e}_0 - \mathbf{f}_0),$$
$$\mathbf{c} = \mathbf{U}_q^T(\mathbf{B}\mathbf{w} + \mathbf{f}_0 - \mathbf{e}_0).$$

Solving the position-constraint problem - (3) Result



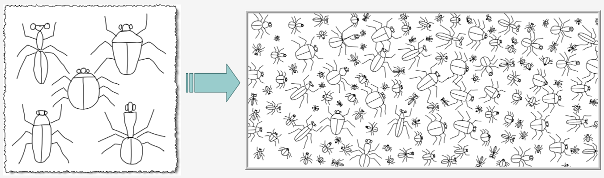
Direct manipulation edit around the left side of mouth:
Estimating correlation between facial muscles (*Left*: asymmetric behavior around lips.
Right: symmetric behavior by considering the covariance matrix of the training data.

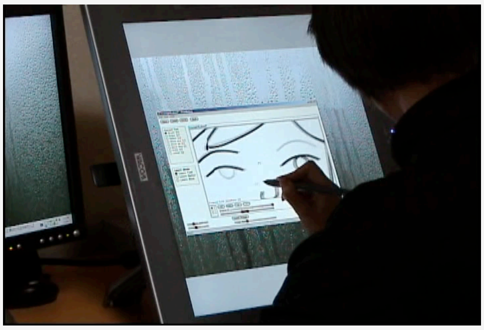
* See more details from Anjyo, Todo and Lewis: "A Practical Approach to Direct Manipulation Blendshapes", Journal of Graphics Tools 2012.

MEMO

APPLICATION: MAKING DOODLE VARIATIONS FROM EXAMPLE

- Problem definition:
  - Only treat black-and-white line drawings
  - Given *N* similar exemplar drawings (doodles)
  - Construct more doodles that resemble the exemplars
- Basic Technique
  - PCA + RBF interpolation
- Example result:





See more from Baxter and Anjyo "Latent Doodle Space", EUROGRAPHICS 2006. https://vimeo.com/3235882

MEMO

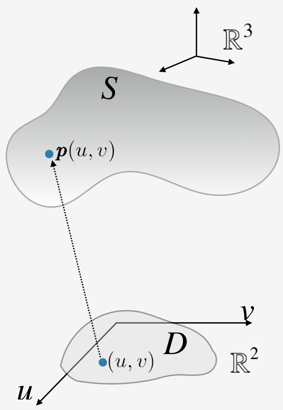
CURVATURES AND EIGENVALUES

In Differential Geometry we consider a surface in the *continuous* world:

$$S = \{\boldsymbol{p}(u,v) = (x(u,v), y(u,v), z(u,v))^T \in \mathbb{R}^3 | (u,v) \in D\}$$

Here we assume that, for any $(u,v) \in D$, $\frac{\partial(x,y,z)}{\partial(u,v)}$ is always of rank 2, where

$$\frac{\partial(x,y,z)}{\partial(u,v)} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix} \text{(Jacobian matrix).}$$



Principal curvatures λ_1 and λ_2 are the eigenvalues of **Weingarten matrix** W:

$$W = \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \begin{pmatrix} L & M \\ M & N \end{pmatrix},$$

where $E = \langle \boldsymbol{p}_u, \boldsymbol{p}_u \rangle;\ F = \langle \boldsymbol{p}_u, \boldsymbol{p}_v \rangle;\ G = \langle \boldsymbol{p}_v, \boldsymbol{p}_v \rangle$

$L = \langle \boldsymbol{p}_{uu}, \boldsymbol{n} \rangle;\ M = \langle \boldsymbol{p}_{uv}, \boldsymbol{n} \rangle;\ N = \langle \boldsymbol{p}_{vv}, \boldsymbol{n} \rangle$

MEMO

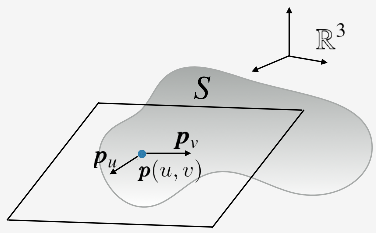
The 1st fundamental form - Definition

Consider the following “symbolic” formula of the *first fundamental form* of the surface S:

$$\mathrm{I} := E\, dudu + 2F\, dudv + G\, dvdv$$

Or we have the matrix form:

$$\mathbf{I} := \begin{pmatrix} E & F \\ F & G \end{pmatrix},$$



where we recall:

$$E = \langle \boldsymbol{p}_u, \boldsymbol{p}_u \rangle;\ F = \langle \boldsymbol{p}_u, \boldsymbol{p}_v \rangle;\ G = \langle \boldsymbol{p}_v, \boldsymbol{p}_v \rangle$$

Note that E, F and G are determined with the 1st derivatives of $\boldsymbol{p}(u, v)$.

The 1st fundamental form - For what?

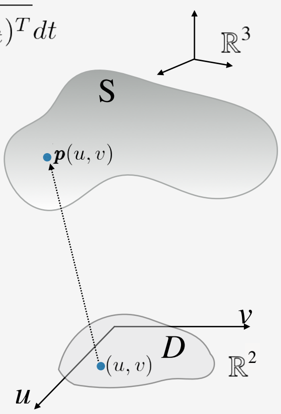
The first fundamental form I is used for:

- Measuring the curve length l of a curve on S that comes from the curve $c(t) = (u(t), v(t)) \in D$ $(a \leq t \leq b)$

$$l = \int_a^b \sqrt{Eu_t^2 + 2Fu_tv_t + Gv_t^2}dt = \int_a^b \sqrt{(u_t, v_t)\mathbf{I}(u_t, v_t)^T}dt$$

- Measuring the surface area A of S for any $U \subset D$:

$$A = \iint_U \sqrt{EG - F^2}dudv = \iint_U \sqrt{\det(\mathbf{I})}dudv$$



where we recall *the role of* D, which defines S.

$$S = \{\boldsymbol{p}(u, v) = (x(u, v), y(u, v), z(u, v))^T \in \mathbb{R}^3 | (u, v) \in D\}$$

Note that $\det(\mathbf{I}) \neq 0$, since S is regular (non-degenerate).

The 2nd fundamental form - Definition

The "symbolic" formula of the *second fundamental form* of surface S:

$$\mathrm{II} := L\,dudu + 2M\,dudv + N\,dvdv$$

Or we have the matrix form:

$$\mathbf{II} := \begin{pmatrix} L & M \\ M & N \end{pmatrix},$$

where we recall: $L = \langle \boldsymbol{p}_{uu}, \boldsymbol{n} \rangle;\ M = \langle \boldsymbol{p}_{uv}, \boldsymbol{n} \rangle;\ N = \langle \boldsymbol{p}_{vv}, \boldsymbol{n} \rangle$

Note that L, M and N are determined by the 2nd derivatives of $\boldsymbol{p}(u, v)$ and the normal vector at $\boldsymbol{p}$.

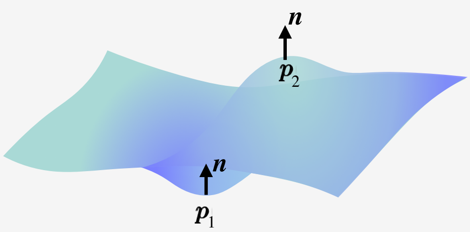
Since $\langle \boldsymbol{p}_u, \mathbf{n} \rangle = \langle \boldsymbol{p}_v, \mathbf{n} \rangle = 0$, we also have:

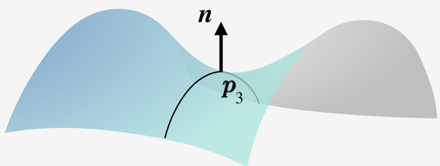
$$L = -\langle \boldsymbol{p}_u, \boldsymbol{n}_u \rangle \ ;\ M = -\langle \boldsymbol{p}_u, \boldsymbol{n}_v \rangle = -\langle \boldsymbol{p}_v, \boldsymbol{n}_u \rangle \ ;\ N = -\langle \boldsymbol{p}_v, \boldsymbol{n}_v \rangle$$

The 2nd fundamental form - For what?

The second fundamental form II describes local behavior of surface S. For example* we know:

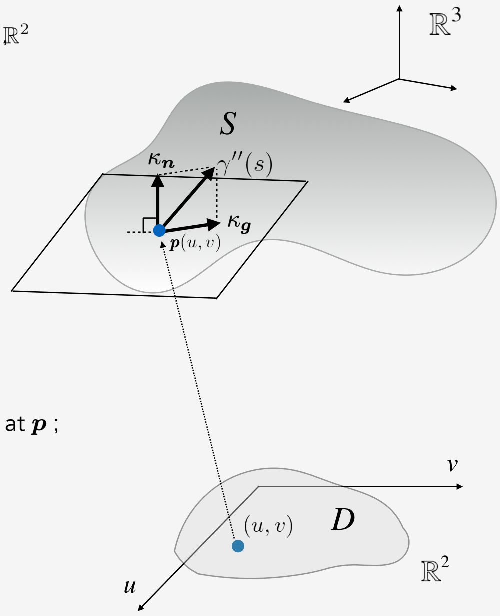
- If II is positive definite at point $\boldsymbol{p}$, then S is concave around ($\boldsymbol{p}_1$ in the figure).
- If II is negative definite at point $\boldsymbol{p}$, then S is convex around ($\boldsymbol{p}_2$ in the figure).
  (The above two cases are $LN - M^2 > 0$).
- If II is indefinite , i.e, $LN - M^2 < 0$, at point $\boldsymbol{p}$, then S is saddle-shaped ($\boldsymbol{p}_3$ in the figure).





* Learn more with S. Kobayashi "Differential Geometry for Curves and Surfaces", Springer 2019.

Curvature



1. The curvature of a planar curve: $\gamma(s) = (x(s), y(s))^T \in \mathbb{R}^2$
   where s denotes arc-length parameter.

   As usual, denoting $\frac{d}{ds}$ by $\prime$ (prime), we have $1 = |\frac{d\gamma}{ds}|^2 = \langle \gamma', \gamma' \rangle$, which leads to $\langle \gamma'', \gamma' \rangle = 0$.

   We set $\boldsymbol{n} := (-y'(s), x'(s))^T$, which is the vector perpendicular to the tangent γ' and $\|\boldsymbol{n}\| = 1$. So we have:

   $$\boxed{\gamma''(s) = \kappa(s)\boldsymbol{n}}$$

   We refer to the coefficient $\kappa(s) = \langle \gamma''(s), \boldsymbol{n} \rangle$ as the normal curvature at s .

2. The curvature of a curve $\gamma(s) = \boldsymbol{p}(u(s), v(s))$ on S:

   $$\gamma''(s) = \kappa_{\boldsymbol{g}} + \kappa_{\boldsymbol{n}},$$

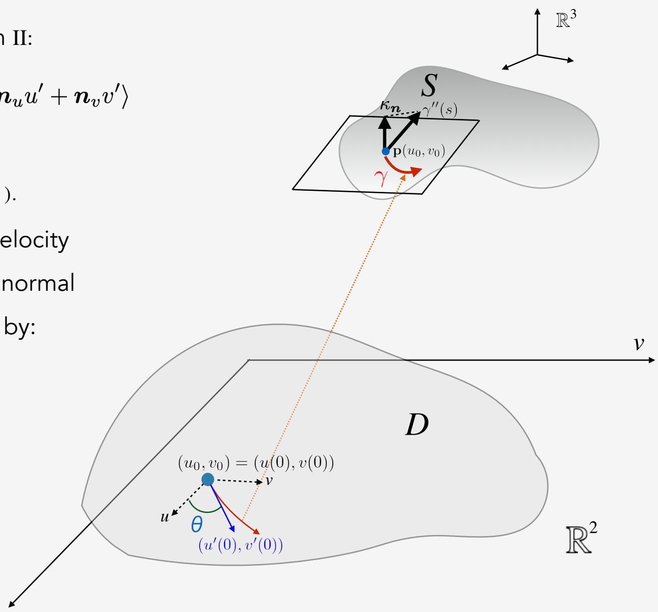
   where *geodesic curvature vector* $\kappa_{\boldsymbol{g}}$ lies on the tangent plane at $\boldsymbol{p}$;

   normal curvature vector $\boxed{\kappa_{\boldsymbol{n}} = \kappa_n(s)\,\boldsymbol{n}}$;

   $\boldsymbol{n}$: unit surface normal at $\boldsymbol{p}$.

   The coefficient $\kappa_n(s) = \langle \gamma''(s), \boldsymbol{n} \rangle$ is referred to as the *normal curvature* of γ .

Normal curvature of a surface



- κ_n is calculated with the 2nd fundamental form $\mathbf{II}$:

$$\kappa_n = \langle \gamma''(s), \boldsymbol{n} \rangle = -\langle \gamma'(s), \boldsymbol{n}' \rangle = -\langle \boldsymbol{p}_u u' + \boldsymbol{p}_v v', \boldsymbol{n}_u u' + \boldsymbol{n}_v v' \rangle$$
$$= L(u')^2 + 2Mu'v' + N(v')^2$$

(Recall $L = -\langle \boldsymbol{p}_u, \boldsymbol{n}_u \rangle$; $M = -\langle \boldsymbol{p}_u, \boldsymbol{n}_v \rangle$; $N = -\langle \boldsymbol{p}_v, \boldsymbol{n}_v \rangle$).

- Consider a curve $(u(s), v(s))$ in D whose initial velocity is given: $(u'(0), v'(0)) = (r\cos\theta, r\sin\theta)$. Then the normal curvature at $\boldsymbol{p} = \boldsymbol{p}(u_0, v_0) = \boldsymbol{p}(u(0), v(0))$ is given by:

$$\kappa_n(\boldsymbol{p}) = \frac{L\cos\theta^2 + 2M\cos\theta\sin\theta + N\sin\theta^2}{E\cos\theta^2 + 2F\cos\theta\sin\theta + G\sin\theta^2}$$
$$= \frac{L\alpha^2 + 2M\alpha\beta + N\beta^2}{E\alpha^2 + 2F\alpha\beta + G\beta^2},$$

where $(\alpha, \beta) \neq (0,0) \in \mathbb{R}^2$ and $\beta = \alpha\tan\theta$.

(Recalling the following eq., the proof is easy:

$$1 = Eu'(0)^2 + 2Fu'(0)v'(0) + Gv'(0)^2$$
$$= r^2(E\cos^2\theta + 2F\cos\theta\sin\theta + G\sin^2\theta) \text{).}$$

Principal curvature as eigenvalues

At a point $\boldsymbol{p}$ on S, the normal curvature $\kappa_n(\boldsymbol{p})$ can then be considered as a function of α and β. So we refer to it as $\lambda(\alpha,\beta)$ for convenience. Now let's find minimun and maximun of $\kappa_n(\boldsymbol{p}) \equiv \lambda(\alpha,\beta)$ over $(\alpha,\beta) \neq 0 \in \mathbb{R}^2$.

Proposition The minimum and maximun values of $\kappa_n(\boldsymbol{p})$ over $(\alpha,\beta) \neq 0 \in \mathbb{R}^2$ are given by the eigenvalues of the Weingarten matrix:

$$W = \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \begin{pmatrix} L & M \\ M & N \end{pmatrix},$$

where we note that the eigenvalues are real numbers, since W is a real-valued symmetric matrix.

Sketch of the proof:

Since $\lambda \equiv \lambda(\alpha,\beta) = \dfrac{L\alpha^2 + 2M\alpha\beta + N\beta^2}{E\alpha^2 + 2F\alpha\beta + G\beta^2}$, we have $L\alpha^2 + 2M\alpha\beta + N\beta^2 - (E\alpha^2 + 2F\alpha\beta + G\beta^2)\lambda = 0$.

If λ attains the minimum or maximun value at $(\alpha,\beta) \neq 0$, then we have $\frac{\partial\lambda}{\partial\alpha} = \frac{\partial\lambda}{\partial\beta} = 0$, which means: $\begin{cases} (L-\lambda E)\alpha + (M-\lambda F)\beta = 0, \\ (M-\lambda F)\alpha + (N-\lambda G)\beta = 0. \end{cases}$

This is equivalent to: $\begin{pmatrix} L & M \\ M & N \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda \begin{pmatrix} E & F \\ F & G \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. Since $\det(\mathbf{I}) = \det\begin{pmatrix} E & F \\ F & G \end{pmatrix} \neq 0$, we get $\boxed{W\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda\begin{pmatrix} \alpha \\ \beta \end{pmatrix}.}$

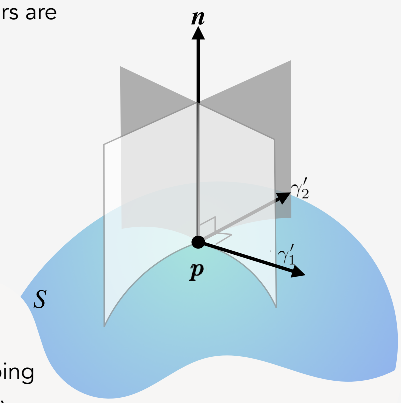
$$\boxed{A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \text{ for } 1 \leq i \leq 2.}$$

The eigenvalues λ_1 and λ_2 are called *principal curvature*.

Mean curvature, Gauss curvature and DDG

- Let λ_1 and λ_2 be the principal curvatures at $\boldsymbol{p}$ on S. Their (unit) eigenvectors are called *principal directions*.
  - If $\lambda_1 \neq \lambda_2$, their principal directions, denoted by γ_1' and γ_2' respectively, are orthogonal to each other.
  - Using the principal curvatures λ_1 and λ_2, we define mean curvature H and Gauss curvature K as follows:

$$H := \frac{\lambda_1 + \lambda_2}{2}$$
$$K := \lambda_1\lambda_2.$$



- Discrete Differential Geometry (DDG) explores a new framework of describing *the "discrete" world* (Differential Geometry deals with the continuous world).
  - How to define a normal vector in DDG?
  - Mean curvature, Gauss curvature, etc. are reformulated in DDG.

As for more about DDG, please refer to the books or previous SIGGRAPH or SIGGRAPH Asia course notes by Mathieu Desbrun and his colleagues, such as De Goes, Desbrun, Tong "Vector Field Processing on Triangle Meshes" SIGGRAPH Asia 2015 Course Notes.

EIGENFUNCTIONS IN IMAGE-BASED RENDERING

Spherical Harmonics

The rendering equation: $L(\mathbf{x}, \omega) = \int_{S^2} L_{\mathrm{env}}(\omega')T(\mathbf{x}, \omega')d\omega' \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} L_\ell^m T_\ell^m(\mathbf{x})$

$$L_{\mathrm{env}}(\omega') \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} L_\ell^m y_\ell^m(\omega')$$

and

$$T(\mathbf{x}, \omega') \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} T_\ell^m(\mathbf{x}) y_\ell^m(\omega') \ ,$$

where

$$\Lambda\, y_\ell^m = -\ell(\ell+1) y_\ell^m$$

Laplace-Beltrami operator on sphere S^2

Eigenfunction (Spherical harmonics) of Λ

Eigenvalue of Λ

Images courtesy of Kei Iwasaki

MEMO

FOURIER EXPANSION

We start with the classical result:

Fourier expansion theorem Suppose that $f : [-\pi, \pi] \to \mathbb{R}$ is a real-valued smooth function satisfying $f(-\pi) = f(\pi)$. Then f is expressed in the following form:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos nx + b_n \sin nx),$$

where we have the coefficients by

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos nt\ dt\ (n = 0, 1, 2, \cdots), \quad b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin nt\ dt\ (n = 1, 2, \cdots).$$

As is well known, the above expression of f, called *Fourier expansion*, is derived from the following "orthogonal" property of the trigonometric functions:

$$\int_{-\pi}^{\pi} \cos^2 nt\ dt = \int_{-\pi}^{\pi} \sin^2 nt\ dt = \pi,$$

$$\int_{-\pi}^{\pi} \cos nt \cos mt\ dt = \int_{-\pi}^{\pi} \sin nt \sin mt\ dt = 0, (m \neq n)$$

$$\int_{-\pi}^{\pi} \cos nt \sin mt\ dt = 0.$$

Fourier expansion is an eigenfunction expansion

We then note that: $\frac{d^2}{dx^2}\cos nx = -n^2\cos nx;\quad \frac{d^2}{dx^2}\sin nx = -n^2\sin nx.$

This means that $\cos nx$ or $\sin nx$ is a solution of the following *linear differential equation*:

$$\boxed{\frac{d^2}{dx^2}\mathbf{u}_n = -n^2\mathbf{u}_n}$$

If we put $A = \frac{d^2}{dx^2}$ and $\lambda_n = -n^2$, the above equation leads to a general form:

$$\boxed{A\mathbf{u}_n = \lambda_n \mathbf{u}_n}$$

where A is a linear operator from vector (or typically function) space V to itself.

If $\mathbf{u}_n \neq 0$ in V, then $\mathbf{u}_n$ is called an eigenfunction for the eigenvalue λ_n.

In the Fourier expansion case, V is the function space consisting of smooth and periodic functions. It's crucial that any element of V can then be expressed by the Fourier expansion, which gives an eigenfunction expansion prescribed by the linear operator (1D Laplacian) A.

An element of V can be considered the function from S^1 (circle) to $\mathbb{R}$. What if S^2, instead of S^1 ??

The rendering equation needs integral computation on S^2

Consider the PRT method (Precomputed Radiance Transfer) for diffuse reflection. Then we calculate the following integral:

$$L(\mathbf{x}, \omega) = \int_{S^2} L_{\text{env}}(\omega')T(\mathbf{x}, \omega')d\omega',$$

where $L_{\text{env}}(\omega')$ denotes the light intensity at $\mathbf{x}$, that comes from the direction ω' and $T(\mathbf{x}, \omega')$ is the transfer function that describes how $\mathbf{x}$ responds to the incoming light.

For efficient computation (in real-time!) to get high quality images, we should find a good system of eigenfunctions that express the above integrants with fewer terms.

- *Recall that we get the Fourier expansion case, where we consider 1D Laplacian to get the eigenfunction system:* $\{1, \cos x, \sin x, \cos 2x, \sin 2x, \cdots\}$.
- *For the S^2 case, we first consider 3D Laplacian and then restrict it on S^2. This means that we want to find eigenfunctions for the Laplace-Beltrami operator.*

Laplace-Beltrami operator and Spherical Harmonics (SH) on S^2

Consider 3D Laplacian: $\quad \Delta := \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2} + \dfrac{\partial^2}{\partial z^2}$.

By changing 3D coordinate with: $\mathbf{x} = (r\sin\theta\cos\varphi, r\sin\theta\sin\varphi, r\cos\varphi)$, we have:

$$\Delta f = \frac{\partial^2 f}{\partial r^2} + \frac{2}{r}\frac{\partial f}{\partial r} + \frac{\cot\theta}{r^2}\frac{\partial f}{\partial \theta} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 f}{\partial \varphi^2} = \frac{\partial^2 f}{\partial r^2} + \frac{2}{r}\frac{\partial f}{\partial r} + \frac{1}{r^2}\Lambda f ,$$

where we define *Laplace-Beltrami operator*:

$$\Lambda f := \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}(\sin\theta\frac{\partial f}{\partial\theta}) + \frac{1}{\sin^2\theta}\frac{\partial^2 f}{\partial\varphi^2} = \cot\theta\frac{\partial f}{\partial\theta} + \frac{\partial^2 f}{\partial\theta^2} + \frac{1}{\sin^2\theta}\frac{\partial^2 f}{\partial\varphi^2} .$$

The Laplace-Beltrami operator is applied to the functions defined on S^2.

For a given $\ell = 0, 1, 2, \cdots,$ we consider the following vector space:

$$\mathcal{H}_\ell := \{f \text{ polynomial on } \mathbb{R}^3 |\ \Delta f = 0 \text{ and } f(rx, ry, rz) = r^\ell f(x, y, z) \text{ for any } x, y, z \text{ and } r > 0\}$$

When we consider $f \in \mathcal{H}_\ell$ as a function on S^2, f is called an ℓ-th order <u>spherical harmonics.</u>

SH functions are the eigenfunctions of Laplace-Beltrami operator on S^2

Consider $\widetilde{\mathcal{H}_\ell} := \{f : S^2 \to \mathbb{R} |\ f$ is ℓ-th order spherical harmonics$\}$.

(a) Then $\widetilde{\mathcal{H}_\ell}$ is a vector space with $\dim \widetilde{\mathcal{H}_\ell} = 2\ell + 1$

(b) Moreover we have $L^2(S^2) = \bigoplus_{\ell=0}^{\infty} \widetilde{\mathcal{H}_\ell}$(direct sum),where $L^2(S^2) := \{f : S^2 \to \mathbb{R} \mid \int_{S^2} |f(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} < \infty\}$.

*The vector space $L^2(S^2)$ is endowed with the inner product $\langle f, g\rangle := \int_{S^2} f(\boldsymbol{\omega}) g(\boldsymbol{\omega}) d\boldsymbol{\omega}$.

It follows from (b) that we can calculate the approximation of the integral by SH expansions:

$$L(\mathbf{x}, \omega) = \int_{S^2} L_{\text{env}}(\omega') T(\mathbf{x}, \omega') d\omega' \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} L_\ell^m T_\ell^m(\mathbf{x})$$

$$L_{\text{env}}(\omega') \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} L_\ell^m y_\ell^m(\omega') \ \text{ and } \ T(\mathbf{x}, \omega') \approx \sum_{\ell=0}^{n-1} \sum_{m=-\ell}^{\ell} T_\ell^m(\mathbf{x}) y_\ell^m(\omega'),$$

$$\boxed{A\mathbf{u}_n = \lambda_n \mathbf{u}_n}$$

where $d\omega' = \sin\theta d\theta d\phi$ and $\{\, y_l^m(\omega)\}_{-l \le m \le l}$ is the orthogonal basis of $\widetilde{\mathcal{H}_\ell}$.

We also note that: $\int_{S^2} y_\ell^m(\boldsymbol{\omega}) y_{\ell'}^{m'}(\boldsymbol{\omega}) d\boldsymbol{\omega} = \delta_{\ell,\ell'} \delta_{m,m'}$; $\boxed{\Lambda y_\ell^m(\theta, \varphi) = -\ell(\ell+1) y_\ell^m(\theta, \varphi)}$.

As for explicit description of the basis function $y_l^m(\omega) \equiv y_l^m(\theta, \varphi)$, please refer to the following paper, for instance: Sloan, Krautz, Synyder "Precomputed radiance transfer for real-time dynamic low-frequency environments", ACM TOG 2002.

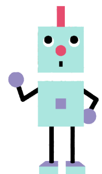
Why are "eigenvalue" and "eigenvector"?

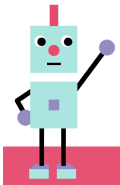
- We have understood that eigenvalue, eigenvector and eigenfunction are very useful mathematical concepts!

  - The concepts are not only for matrix, but also for (linear) differential operators.
  - Many applications in statistics, geometry and rendering.

$$\boxed{A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \ \text{for } 1 \le i \le m}$$

- Let's explore more graphics applications!





Duality in CG

MEMO

The terminology 'dual' appears in several situations:

(I) A linear map from a vector space V to $\mathbb{R}$ is called a dual (vector) of V .
All the dual vectors of V is defined to be the dual vector space.
An example of a dual vector is the coordinate function.
A fancy example is a probability density function, which is regarded as the dual of set of measurable sets.

As a natural generalization, Schwartz distribution, a class of generalized function, is also regarded as a dual of some vector space, the space of test functions. Schwartz distribution unifies the usual function and Dirac distribution.

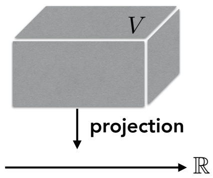
(II) Another (very different) usage of 'dual' is a dual number, typically as dual quaternions.
Dual number is a generalization of usual numbers with introducing a fancy virtual number ε with the strange relation $\varepsilon^2 = 0$.
Note that ε is not zero.

This number system is a purely algebraic object, so that any computation can be extend to this new numbers.
Surprisingly, the dual number also has a geometric interpretation, a linear approximation of curved object, like a surface, a curve.
This technique is used in automatic differentiation as well.

DUAL 1: Dual Vector Space

$\begin{cases} V & \text{: vector space} \\ V^* & \text{: dual vector space} \end{cases}$ dual

$\begin{cases} p \in V & \text{: point} \\ \xi \in V^* \text{ is a linear map} & \xi : V \to \mathbb{R} \end{cases}$ → $\xi(p)$: observation



ξ is regarded as a "coordinate" on V.

DUAL 1: Application of Dual Vector Space

M : manifold (e.g. surface in $\mathbb{R}^3$)

vector field :

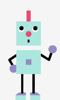
assign a <u>tangent vector</u> on each point $\in M$

↕ dual

differential form (1-form) :

assign a <u>cotangent vector</u> on each point $\in M$

$TM = \bigcup_{x \in M} T_x M$	$\xleftrightarrow{\text{dual}}$	$T^*M = \bigcup_{x \in M} T_x^* M$
tangent bundle		cotangent bundle



$\Omega^1 M = T^* M$ (1-form)

↓ exterior product

$\wedge^2 \Omega^1 M = \Omega^2 M$ (2-form)

DUAL 1 : Schwartz Distribution

Example $\begin{cases} \text{Function } \varphi(x) \\ \text{Dirac delta} \end{cases}$

Example 1: A function $\varphi(x)$ is considered to be density. (Idea of probability)

$$\underset{\text{test function}}{\mathcal{S} \ni \varphi(x)} \longmapsto T_\rho(\varphi) = \int_{\mathbb{R}^3} \varphi(x)\rho(x)\,dx \in \mathbb{R}$$

$T_\rho(\varphi)$ lives in the dual of $\mathcal{S}$

called Schwartz space

Example 2: Dirac Delta

$\delta : \mathcal{S} \ni \varphi(x) \mapsto \varphi(0) \in \mathbb{R}$

δ is also regarded as an element of dual of $\mathcal{S}$.

Definition Schwartz distribution is an element of dual of $\mathcal{S}$.

Remark
Since the dimension of $\mathcal{S}$ is infinite, the rigorous definition requires some continuity, regularity, related with the topology in the functions space.

It is a bit complicated, but don't be afraid; In an actual application, this continuity often holds.

DUAL 2: Dual Numbers

Dual number ε : $\varepsilon^2 = 0$

- Extend usual number by adding ε .
- Keep the computational law

$$\begin{cases} (ab)c = a(bc) & \text{: associative} \\ a(b+c) = ab + ac & \text{: distributive} \end{cases}$$

The role of ε several different applications

$$\begin{cases} \bullet\ \textbf{Quaternion} \longrightarrow \textbf{Dual quaternion} \\ \bullet\ \textbf{Manifold} \longrightarrow \textbf{Tangent space} \\ \bullet\ \textbf{Function} \longrightarrow \textbf{Automatic differentiation} \end{cases}$$

e.g. $\sin(x+\varepsilon) = \sin(x) + \varepsilon\cos(x)$

DUAL QUATERNION

A quaternion number can express every 3D rotation.
A dual quaternion can express every 3D rigid motion.

- An advanced topic: don't worry (and see [Anjyo17] for details).
- A unit dual quaternion can express Screw motion in 3D.
- It is a eight-dimensional real vector, with six free parameters.
- Dual number: a fancy but simple law of computation
- We have a reasonable extension of functions such as polynomials, exponential, trigonometric, for dual variables.

Dual quaternion

We introduce the dual quaternion space: $\mathbb{H}(\varepsilon)$

- Element: $\mathbf{p} + \mathbf{q}\varepsilon \in \mathbb{H}(\varepsilon)$ $\begin{cases} \mathbf{p}, \mathbf{q} \in \mathbb{H} \\ \varepsilon^2 = 0 \end{cases}$ ε : a simbol

- $\mathbb{H}(\varepsilon)$ **: algebra** $\begin{cases} \text{addition, multiplication} \\ \text{associative, distributive} \\ \text{Non-computative multiplication} \end{cases}$

addition, multiplication	**2 operations**
associative, distributive	Compute as usual
Non-computative multiplication	Compute as quaternion

- Conjugate: $\overline{\mathbf{p} + \mathbf{q}\varepsilon} = \overline{\mathbf{p}} - \overline{\mathbf{q}}\varepsilon$
- Norm: $|\mathbf{p} + \mathbf{q}\varepsilon|^2 = (\mathbf{p} + \mathbf{q}\varepsilon)(\overline{\mathbf{p} + \mathbf{q}\varepsilon})$

- $\mathbb{H}(\varepsilon)$ is the algebra, with two operations, addition and multiplication.
- This algebra has the standard properties: associative and distributive, so that we can do the computations as usual numbers.
- However, the multiplication is non-commutative; this is like a quaternion number.
- Note that ε is commutative with arbitrary dual quaternions.

Example of computation of dual quaternion

$$
\begin{aligned}
|\mathbf{p}+\mathbf{q}\varepsilon|^2 &= (\mathbf{p}+\mathbf{q}\varepsilon)(\overline{\mathbf{p}+\mathbf{q}\varepsilon}) \\
&= (\mathbf{p}+\mathbf{q}\varepsilon)(\overline{\mathbf{p}}-\overline{\mathbf{q}}\varepsilon) \\
&= \mathbf{p}\overline{\mathbf{p}}+\mathbf{q}\overline{\mathbf{p}}\varepsilon-\mathbf{p}\overline{\mathbf{q}}\varepsilon-\mathbf{q}\overline{\mathbf{q}}\varepsilon^2 \\
&= \mathbf{p}\overline{\mathbf{p}}+(\mathbf{q}\overline{\mathbf{p}}-\mathbf{p}\overline{\mathbf{q}})\varepsilon \\
&= |\mathbf{p}|^2-2\,\mathrm{Im}(\mathbf{p}\overline{\mathbf{q}})\varepsilon
\end{aligned}
$$

distributive

$\varepsilon^2=0$

Computation Rule: ε commutes with everything; this is a rule of $\mathbb{H}(\varepsilon)$

- A dual quaternion with norm 1 is called unit dual quaternion.

$$|\mathbf{p}+\mathbf{q}\varepsilon|=1$$
$$\iff |\mathbf{p}|=1 \text{ and } \mathrm{Im}(\mathbf{p}\,\overline{\mathbf{q}})=0$$

- Unit dual quaternion has 8-2=6 parameters

Unit dual quaternion action as 3D rigid motion

$$
\begin{aligned}
&(\mathbf{p}+\mathbf{q}\varepsilon)(1+\mathbf{r}\varepsilon)(\overline{\mathbf{p}+\mathbf{q}\varepsilon})\\
&=\big(\mathbf{p}+(\mathbf{q}+\mathbf{pr})\varepsilon\big)(\overline{\mathbf{p}}-\overline{\mathbf{q}}\varepsilon)\\
&=\mathbf{p}\overline{\mathbf{p}}+\big((\mathbf{q}+\mathbf{pr})\overline{\mathbf{p}}-\mathbf{p}\overline{\mathbf{q}}\big)\varepsilon\\
&=\underbrace{\mathbf{p}\overline{\mathbf{p}}}_{=\,\mathbf{1}}+\underbrace{\big(\underbrace{\mathbf{pr}\overline{\mathbf{p}}}_{\text{3D rotation}}+\underbrace{(\mathbf{q}\overline{\mathbf{p}}-\mathbf{p}\overline{\mathbf{q}})}_{\text{3D translation}}\big)\varepsilon}_{\text{3D rigid motion}}
\end{aligned}
$$

Unit Dual Quaternion

- $\mathbf{r} \in \operatorname{Im}(\mathbb{H}) \cong \mathbb{R}^3$: a 3D point
- $\mathbf{p}+\mathbf{q}\varepsilon$: unit dual quaternion

- $\mathbf{p}\overline{\mathbf{p}} = 1$: by the assumption
- $\mathbf{pr}\overline{\mathbf{p}}$: 3D rotation
- $(\mathbf{q}\overline{\mathbf{p}} - \mathbf{p}\overline{\mathbf{q}})\varepsilon = 2\operatorname{Im}(\overline{\mathbf{p}}\mathbf{q})\varepsilon$: 3D translation

Exponential function: for several classes of numbers

$$\exp(x) = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots$$

Key

often has a closed expression using exp, sin, cos,...(not infinite series)

real number $\xrightarrow{\text{exp}}$ **positive real**

$a + b\varepsilon \quad (a, b \in \mathbb{R}) \xrightarrow{\text{exp}} e^a(1 + b\varepsilon) = x + y\varepsilon \quad (x > 0, y \in \mathbb{R})$

pure imaginary complex number $\xrightarrow{\text{exp}}$ **unit complex number** $e^{i\theta} = \cos\theta + i\sin\theta$

$|z| = 1$

$\mathrm{Im}\mathbb{H} \xrightarrow{\text{exp}}$ **unit quaternion**

$\varepsilon\mathrm{Im}\mathbb{H} \xrightarrow{\text{exp}} 1 + \varepsilon\mathrm{Im}\mathbb{H}$

$\mathbb{H}(\varepsilon) \xrightarrow{\text{exp}}$ **invertible dual quaternion**

All these exponential map is surjective, so that every numbers in the right-hand side can be expressed as an exponential of the left-hand side. It may not be unique, however.

Exponential law in dual quaternion

$$\exp(s(\mathbf{p}+\mathbf{q}\varepsilon))\exp(t(\mathbf{p}+\mathbf{q}\varepsilon)) = \exp((s+t)(\mathbf{p}+\mathbf{q}\varepsilon))$$

$$\mathbf{p}+\mathbf{q}\varepsilon \in \mathbb{H}(\varepsilon)$$

$$s, t \in \mathbb{R}$$

The key of the interpolation, behind

- Dual quaternion is non-commutative,
  so the exponential law does not hold in general;
  $$\exp(\mathbf{p}+\mathbf{q}\varepsilon)\exp(\mathbf{r}+\mathbf{s}\varepsilon) \stackrel{?}{=} \exp(\mathbf{p}+\mathbf{q}\varepsilon+\mathbf{r}+\mathbf{s}\varepsilon)$$
- However, the exponential law in the above form holds, because dual quaternions in this form are commutative !
- This enables us to interpolation in dual quaternion

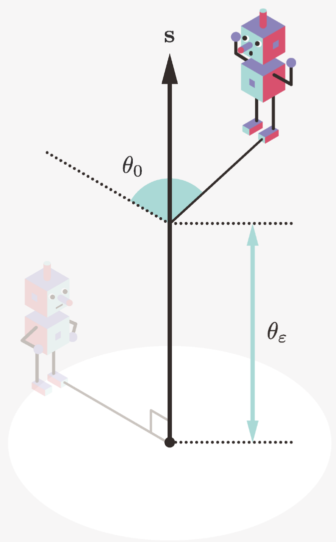
Analogy between DUAL QUATERNION and COMPLEX NUMBERS

Analogy

$\mathbb{H}(\varepsilon)$	$\mathbb{C}$
$\hat{\mathbf{q}} = \cos\frac{\hat{\theta}}{2} + \hat{\mathbf{s}}\,\sin\frac{\hat{\theta}}{2} = \exp\left(\frac{\hat{\theta}}{2}\,\hat{\mathbf{s}}\right)$	$x + yi = \cos\frac{\theta}{2} + i\sin\frac{\theta}{2} = \exp\left(\frac{\theta}{2}\,i\right)$
$\hat{\theta} = \theta_0 + \varepsilon\,\theta_\varepsilon \in \mathbb{R}(\varepsilon) = \mathbb{R} + \mathbb{R}\,\varepsilon$ **: real dual number**	$\theta \in \mathbb{R}$ **: real number**
$\hat{\mathbf{s}} = \mathbf{s}_0 + \varepsilon\,\mathbf{s}_\varepsilon \in \mathrm{Im}(\mathbb{H}(\varepsilon)) = \mathrm{Im}\mathbb{H} + (\mathrm{Im}\mathbb{H})\varepsilon$ **: unit imaginary unit quaternion**	$i \in \mathrm{Im}\,\mathbb{C} = i\mathbb{R}$ **: unit imaginary complex number**

- The right column, complex numbers, can be regarded as a 'toy model' of the left column, dual quaternion.
- Unit dual quaternion expresses all 3D rigid motion, 6 parameters, while unit complex number expresses all 2D rotations, 2 parameters.
- In both pictures, unit imaginary number is the source of magic, which makes the exponential function into the trigonometric function, sine and cosine.

DUAL QUATERNION

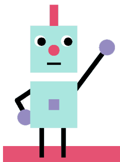


$$\hat{\mathbf{q}} = \cos\frac{\hat{\theta}}{2} + \hat{\mathbf{s}}\,\sin\frac{\hat{\theta}}{2}$$

$$\hat{\theta} = \theta_0 + \varepsilon\,\theta_\varepsilon$$

$$\hat{\mathbf{s}} = \mathbf{s}_0 + \varepsilon\,\mathbf{s}_\varepsilon$$

- Dual quaternion expression of rigid motion well fits into screw theory:
  - Pitch is θ_ε, which is the epsilon part of dual real number $\hat{\theta}$.
  - Rotation angle along the rotation axis is θ_0, which is the neutral part of $\hat{\theta}$.
  - It is significant that the combination $\hat{\theta} = \theta_0 + \varepsilon\,\theta_\varepsilon$ does arise in dual quaternion expression as above 😄.

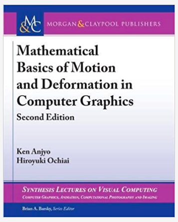


References

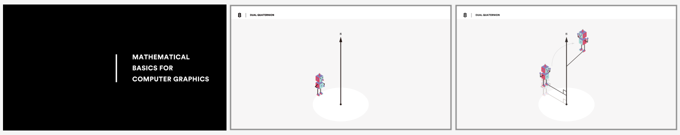
Further References

The original references of CG techniques and mathematics discussed in this course can also be found in:

[Anjyo17] Anjyo and Ochiai:
Mathematical Basics of Motion and Deformation in Computer Graphics: Second Edition,
Synthesis Lectures on Computer Graphics and Animation,
Morgan & Claypool Publishers 2017



MATHEMATICAL BASICS FOR COMPUTER GRAPHICS
https://youtu.be/I2Y-pJYmu9A



QUESTIONS？ — Feel free to ask us :)

Hiroyuki Ochiai

email: ochiai@imi.kyushu-u.ac.jp

https://researchmap.jp/read0162829?lang=en

Ken Anjyo

email: anjyo@acm.org

http://anjyo.org

MEMO