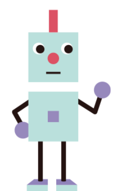
SIGGRAPH 2016 Course Notes

An Elementary Introduction to Matrix Exponential for CG



Hiroyuki Ochiai
Ken Anjyo
Ayumi Kimura

Version 1.0
May 5, 2016

Render the Possibilities
SIGGRAPH2016

About the authors:

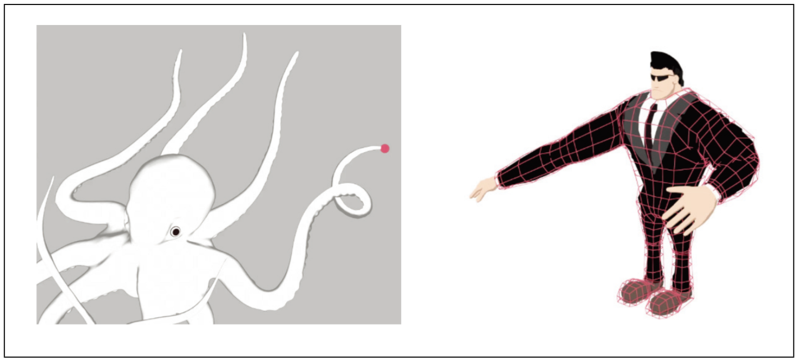
Hiroyuki Ochiai: Professor, Institute of Math-for-Industry, Kyushu University, Japan. He received his Ph.D. in mathematics in 1993 from the University of Tokyo. His research interests include representation theory of Lie groups and Lie algebras, algebraic analysis and group theory. He has been joining the CREST project *Mathematics for Computer Graphics* led by Ken Anjyo since 2010. He was a lecturer at SIGGRAPH Asia 2013 and SIGGRAPH2014 courses.: `http://mcg.imi.kyushu-u.ac.jp/`

Ken Anjyo: R&D supervisor at OLM Digital. He has published technical papers, several of which were presented at SIGGRAPH, SCA, NPAR and Eurographics. He is co-founder of the Digital Production Symposium (DigiPro, 2012-2016), served as SIGGRAPH Asia 2015 Courses Program co-chair, and was a coorganizer of the symposium "Mathematical Progress in Expressive Image Synthesis" (MEIS2013-2016). He organized several courses at SIGGRAPH and SIGGRAPH Asia in 2010, 2013, and 2014: `http://anjyo.org`

Ayumi Kimura: R&D manager at OLM Digital. She has been managing and supporting various research activities at the company since 2004. She is also a key contributor to establishing DigiPro as well as MEIS. She is a member of ACM SIGGRAPH.

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4289-6/16/07
DOI: http://dx.doi.org/10.1145/2897826.2927310

Table of Contents

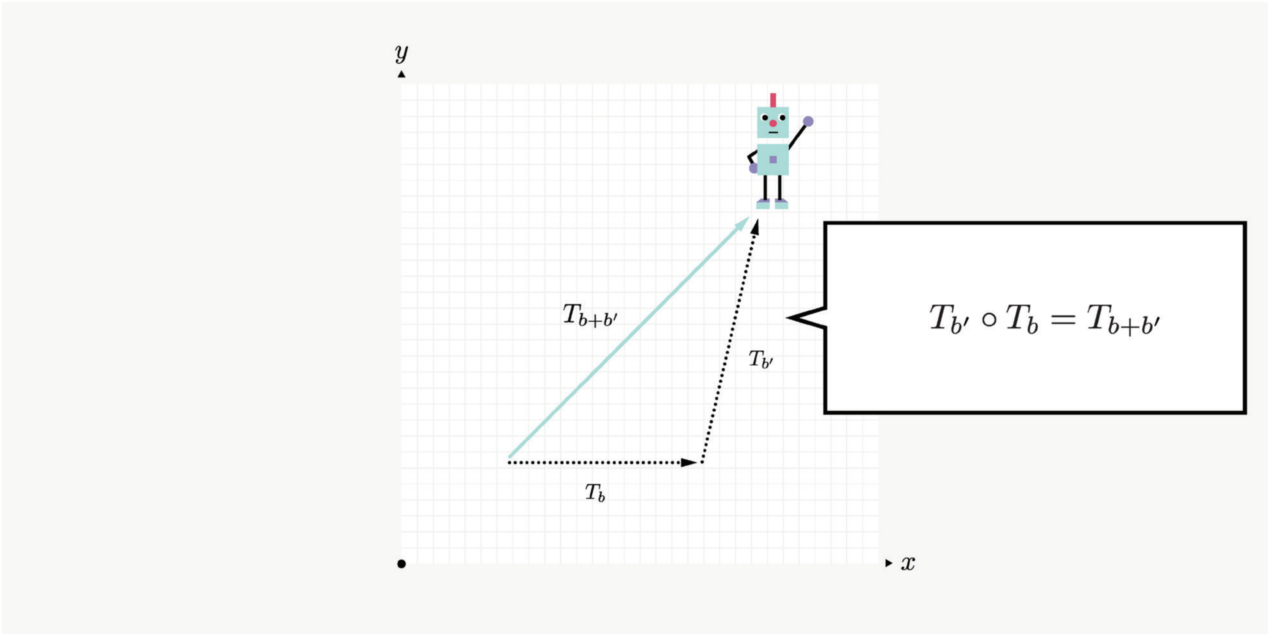


Introduction

This course presents an elementary, intuitive, and visual introduction to several mathematical basics for beginners in computer graphics. The mathematical concepts covered in this course include 2D/3D translation, rotation, affine transformation, quaternion, dual quaternion, exponential and logarithm. These are quite useful for various aspects of computer graphics, including curve/surface editing, deformation and animation of geometric objects. On one hand we don't assume that the audience has familiarity with such mathematical concepts. Elements of linear algebra and calculus at undergraduate level are sufficient for the attendees of this course. To clarify the relation between these concepts and computer graphics practice, several graphics techniques will also be demonstrated, such as morphing (ARAP), cage-based deformation and Poisson mesh editing.

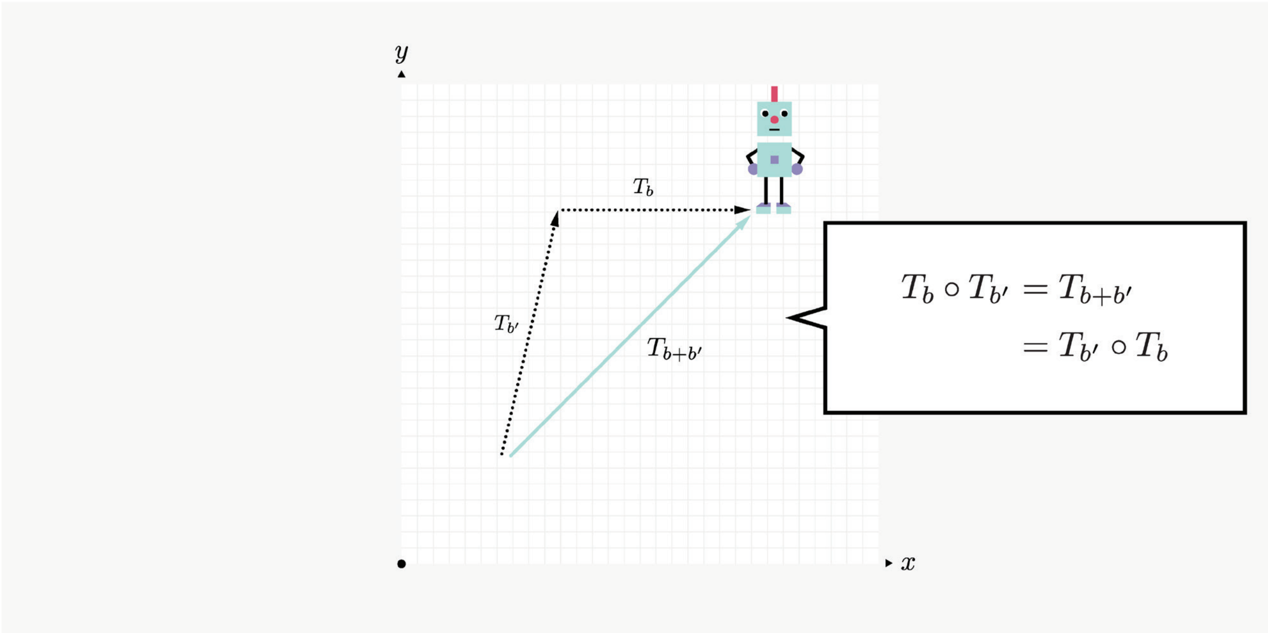
A unique and interesting feature of this course is that we demonstrate most of the concepts with animation videos, which help the audience get visual "experience" of mathematical concepts more concretely than learning from a textbook. The practical graphics applications will also illustrate the power of careful use of the mathematical concepts.

Starting with 2D translation, the present course notes provide 30 topics in total, each of which is given in a separate section. Each section includes representative slides taken from the original movies, which are followed by text explanation (typically one slide per topic, but sometimes more). During our presentation, we will explain the mathematical conepts along with their graphical meanings. We will then also provide more texts and equations for the slides regarding relatively advanced topics. The attendee may therefore add his/her own comments to make the course notes a nice guidebook depending on his/her degree.



1 | 2D TRANSLATION

- The first movie explains 2D translation and its properties.
- A composition of translation is again a translation.

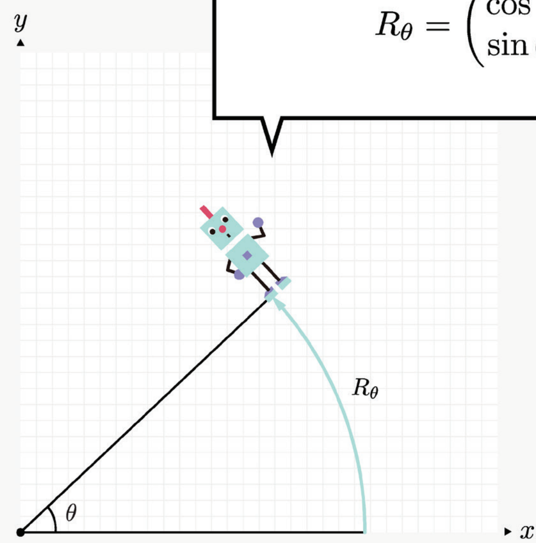


1 | 2D TRANSLATION

- A composition of translations does not matter the order: <u>commutative.</u>

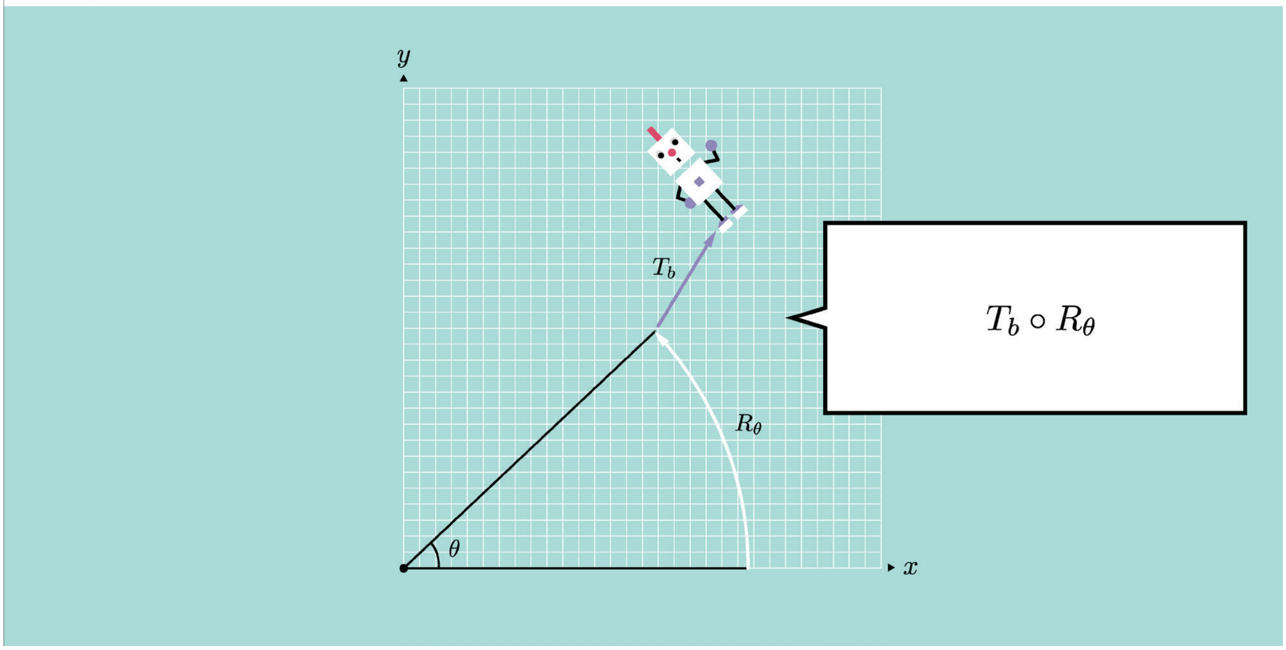
- The inverse translation is again a translation.

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

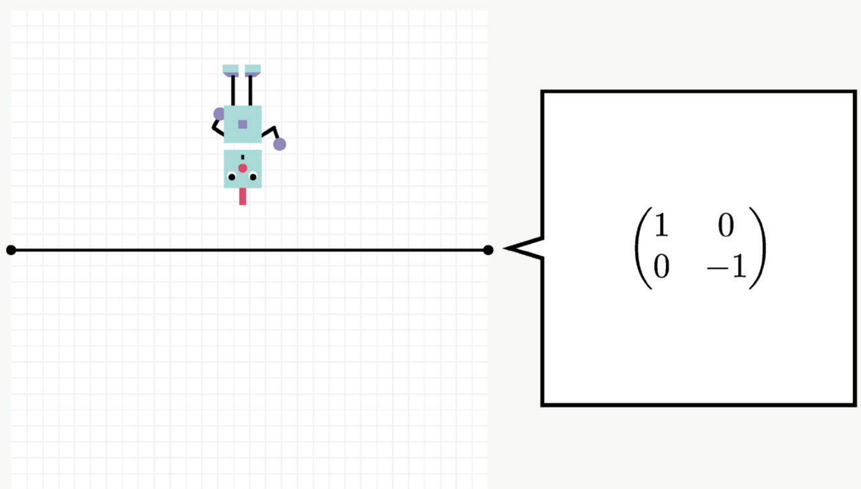


2 | 2D ROTATION

- 2D rotation is specified by the center of motion and angle of rotation.
- We here consider rotations centered at the origin
- A composition of rotations is a rotation.
- The angle of composited rotation is given by a sum.

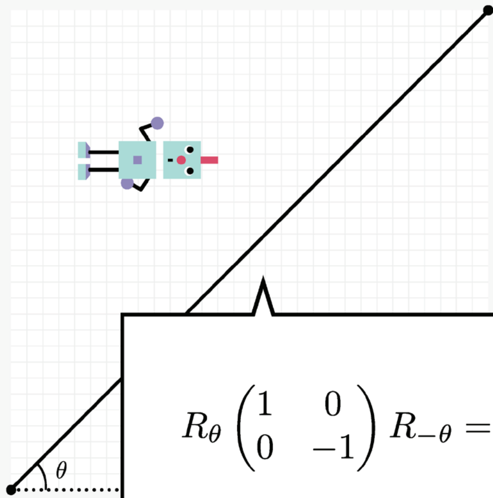


- A translation and a rotation preserve the length, the angle, and the area.
- Such a motion(=transformation) is called rigid.
- Conversely, any rigid transformation is written as a composition of translation and rotation.
- Such an expression is unique.
- Such an expression depends on the order: Non-commutative.



4 | 2D REFLECTION

- The flip with respect to x-axis is given by $(x, y) \rightarrow (x, -y)$.
- A matrix expression of this transformation is given by a simple diagonal matrix whose determinant is -1.
- This minus sign of the determinant means orientation-reversing.

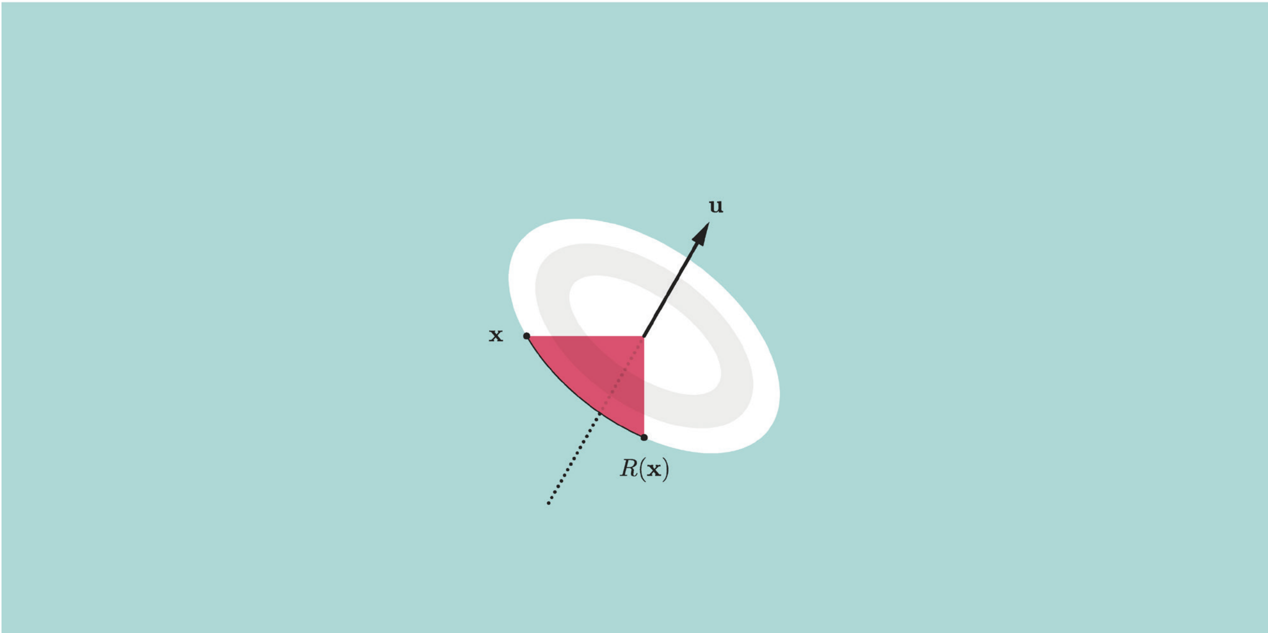


$$R_\theta \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} R_{-\theta} = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}$$

4 | 2D REFLECTION

- A 2D reflection along a line is a rigid transformation; preserve an angle, length, and area.
- 2D reflection does not preserve the orientation.
- A matrix expression of a general reflection along with a general line can be obtained by a product of simple reflection and rotations.

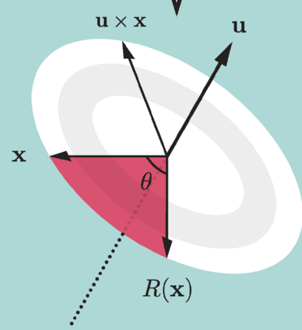
5 | 3D ROTATION: AXIS-ANGLES



5 | 3D ROTATION: AXIS-ANGLES

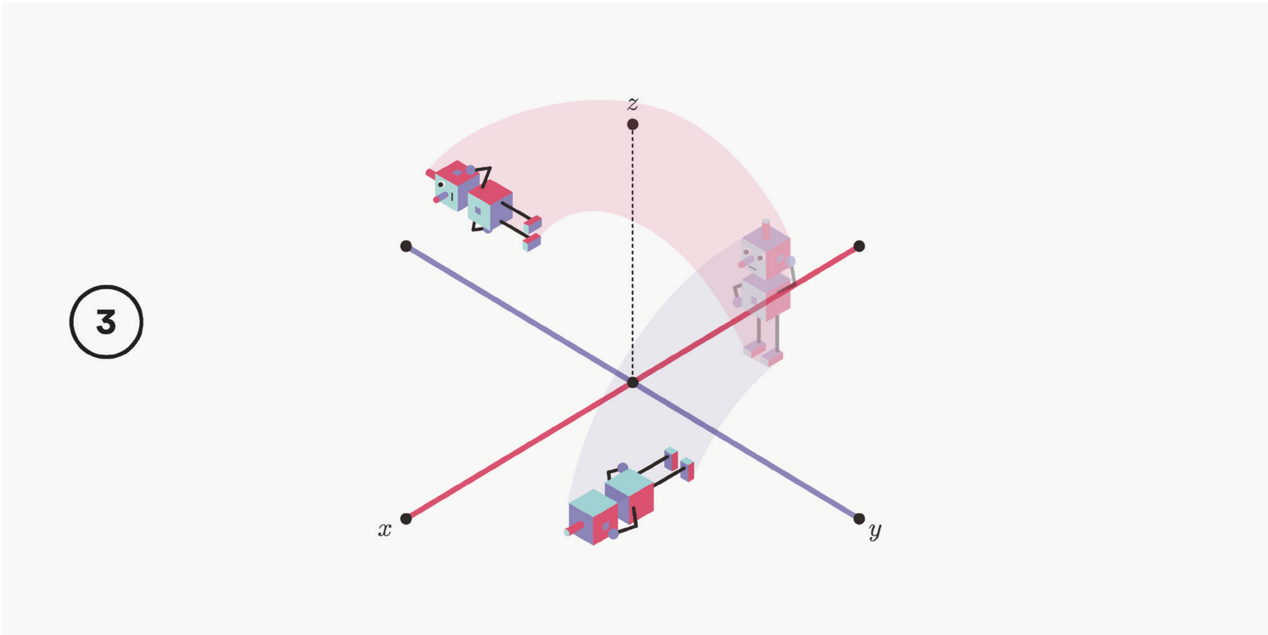
- There are several way to describe 3D rotation.
- 3D rotation can be specified by the rotation axis (= normal vector to rotation plane) and the rotation angle.
- The number of essential parameters is three.

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



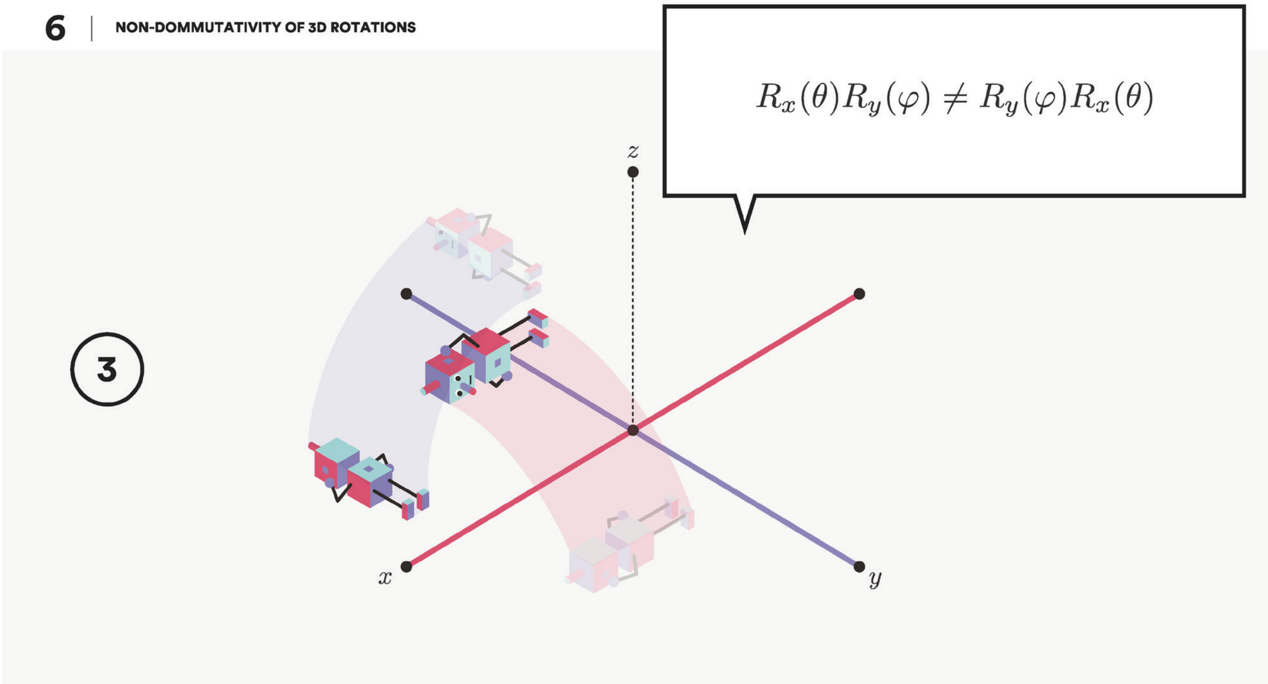
5 | 3D ROTATION: AXIS-ANGLES

- 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.



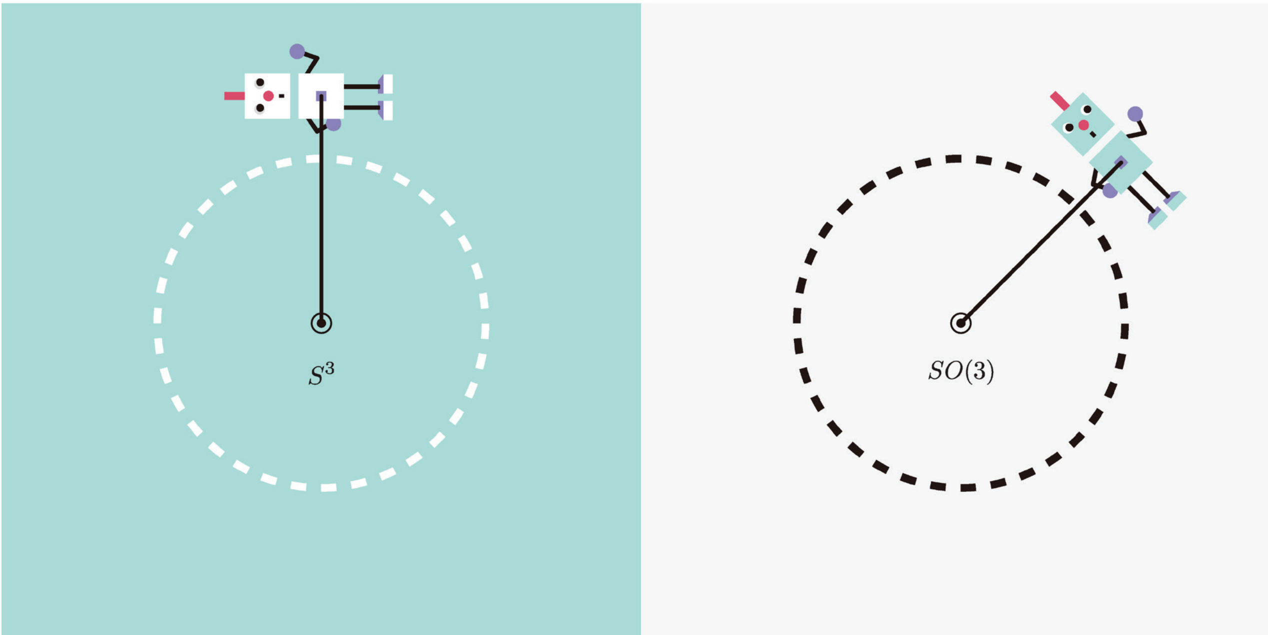
6 | NON-DOMMUTATIVITY OF 3D ROTATIONS

- 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.



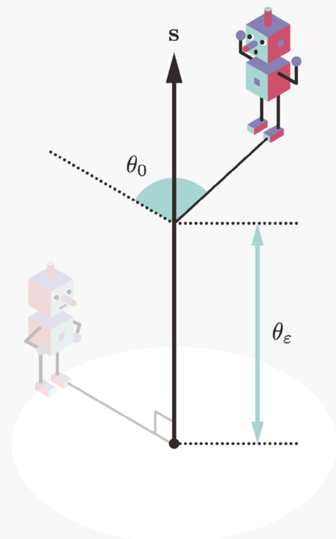
6 | NON-DOMMUTATIVITY OF 3D ROTATIONS

- 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. 😄



7 | QUATERNION VS. 3D ROTATION

- There is another way to express 3D rotations, using unit quaternion.
- A quaternion is a `number' with four real parameters, while a complex number is a number with two real parameters.
- A unit quaternion is a quaternion with length 1.
- A rotation corresponds to two unit quaternions: ± q



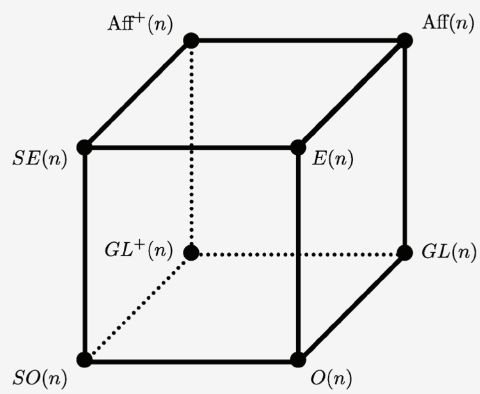
$$\hat{\mathbf{q}} = \cos\frac{\hat{\theta}}{2} + \hat{\mathbf{s}}\,\sin\frac{\hat{\theta}}{2}$$

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

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

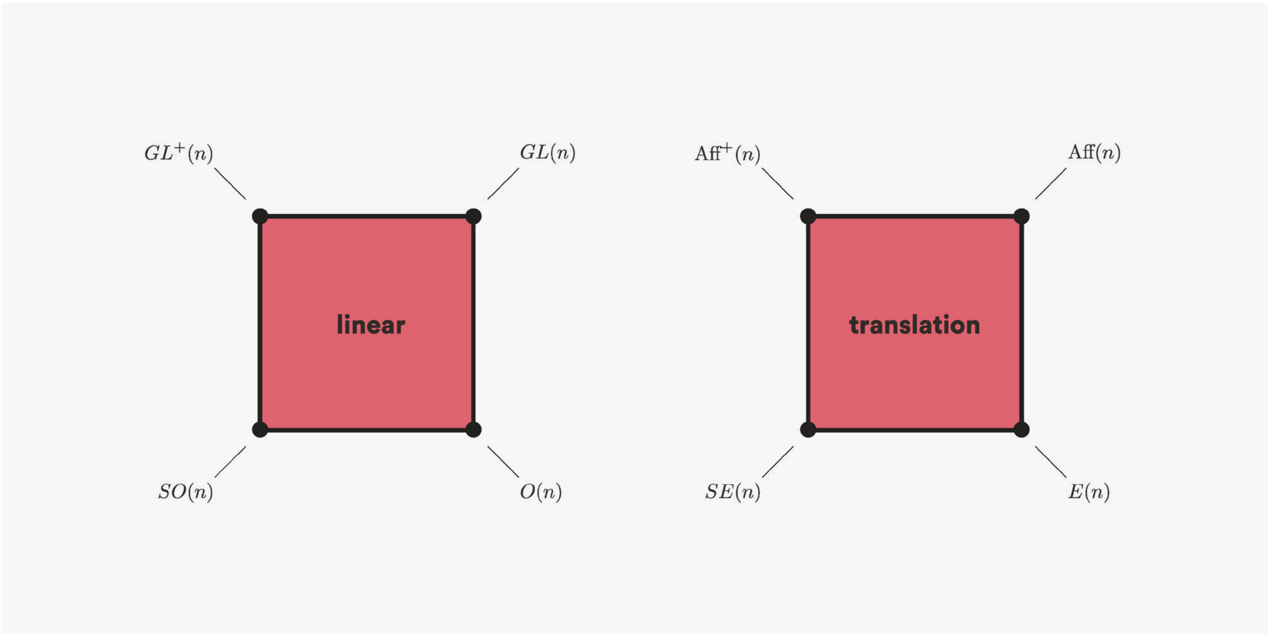
8 | DUAL QUATERNION

- An advanced topic: don't worry (and see [Anjyo14] 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.



9 | LIE GROUPS - SUBGROUPS

- There are several classes of transformations.
- Each class forms a group.
- We can classify them into several categories:
  - orientation-preserving vs. reversing,
  - fixing the origin vs. with-translation,
  - rigid vs. non-rigid.

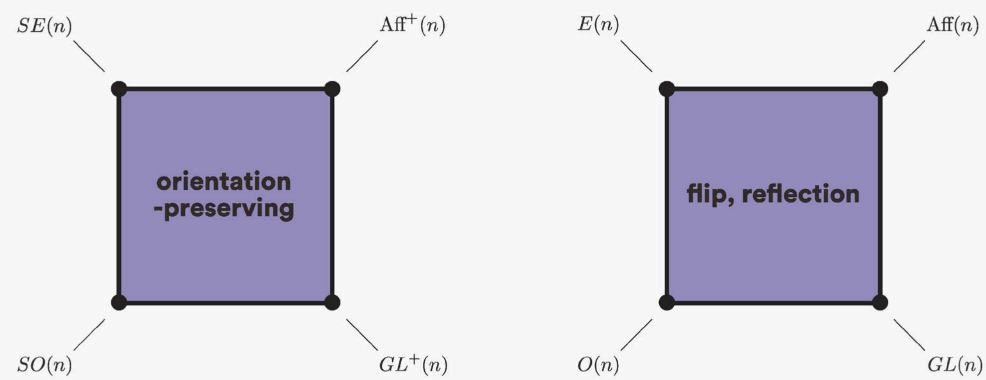


9 | LIE GROUPS - SUBGROUPS

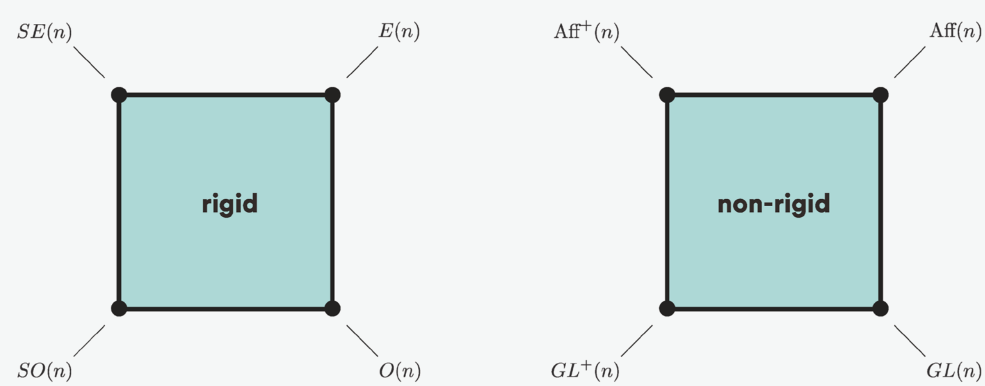
In this diagram, we use the following abbreviation:

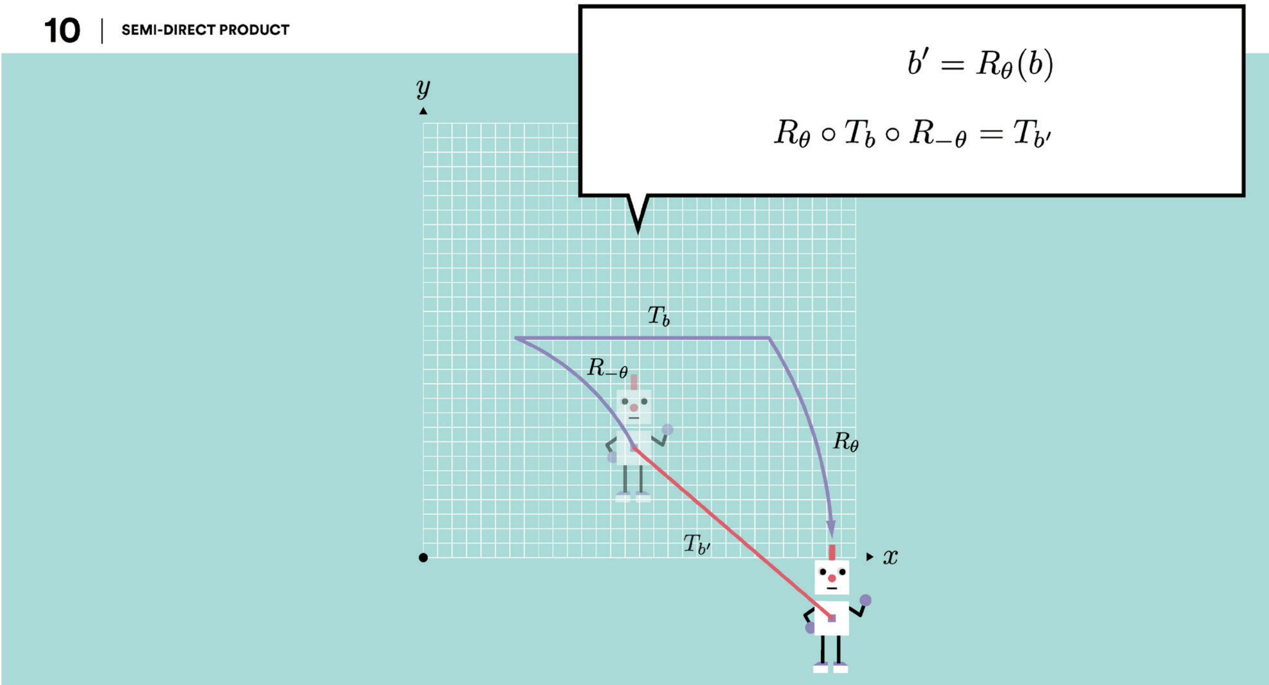
- E(n): n-dimensional Euclidean motion group
- SE(n): special Euclidean motion group
- Aff(n): affine transformation group
- GL(n): general linear group
- O(n): orthogonal group
- SO(n): special orthogonal group
- plus sign (+) for Aff(n), GL(n) means orientation-preserving

9 | **LIE GROUPS - SUBGROUPS**



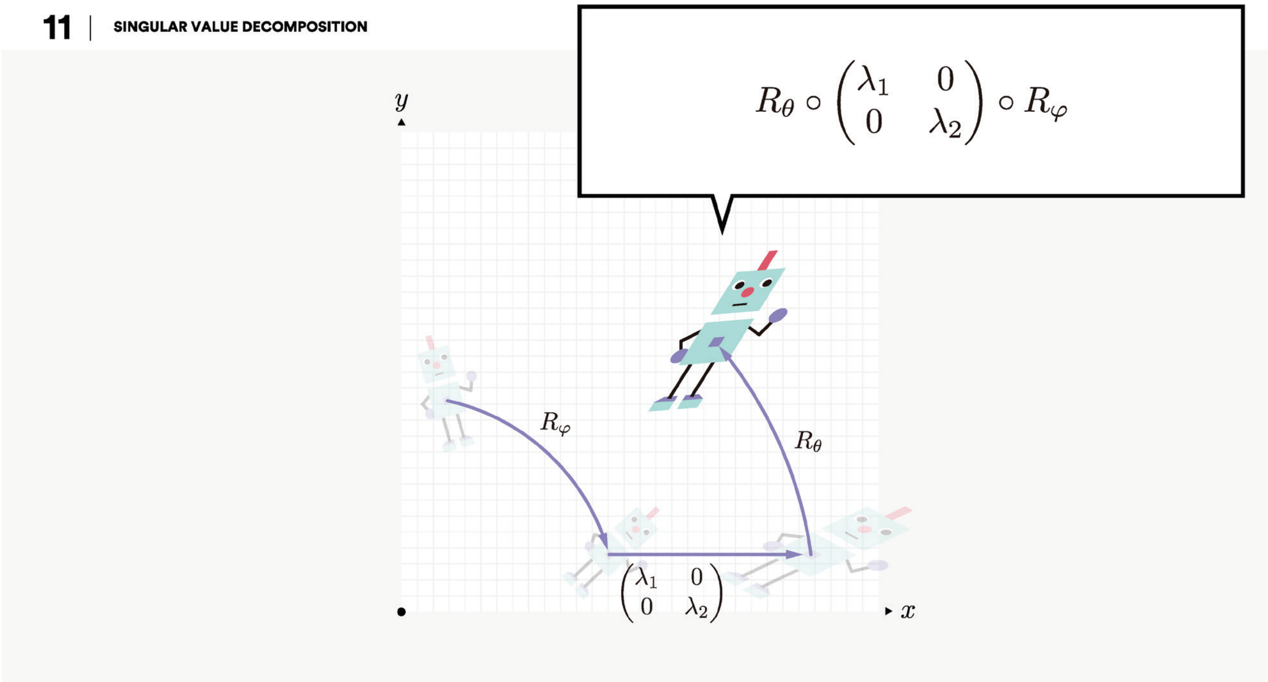
9 | **LIE GROUPS - SUBGROUPS**





10 | SEMI-DIRECT PRODUCT

- For 2D rigid transformation, a rotation and a translation are non-commutative.
- Any conjugate of translation is again a translation.
- Translations forms a `normal' subgroup.
- 2D rigid transformation group (= Euclidean motion group) is a semi-direct product of subgroups consisting of translations and rotation.
- These fancy terminology is not necessary, but helps to understand



11 | SINGULAR VALUE DECOMPOSITION

- Directional dilation gives a deformation
- A linear transformation is not rigid;
  does not preserve an angle, length, area.
- A linear transformation has SVD.
- 2D SVD: composition of two rotations and a xy-dilation

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

12 | EXPONENTIAL : TAYLOR EXPANSION

- 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.

MATRIX EXPONENTIAL

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

12 | EXPONENTIAL : TAYLOR EXPANSION

- Matrix exponential has several properties in common with scalar exponential function.
- Exponential law :

$$\exp(sA)\exp(tA) = \exp((s+t)A) \qquad s, t \in \mathbb{R}$$

$$A = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$$

$$\exp\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} + \frac{1}{2}\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}^2 + \frac{1}{6}\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}^3 + \cdots$$

13 | EXPONENTIAL : DIAGONAL

$$\exp\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} = \begin{pmatrix} 1 + a + \frac{1}{2}a^2 + \frac{1}{6}a^3 + \cdots & 0 \\ 0 & 1 + b + \frac{1}{2}b^2 + \frac{1}{6}b^3 + \cdots \end{pmatrix}$$

EXPONENTIAL : DIAGONAL

$$\exp\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} = \begin{pmatrix} e^a & 0 \\ 0 & e^b \end{pmatrix}$$

13 | EXPONENTIAL : DIAGONAL

- The matrix exponential for a diagonal matrix is reduced to the scalar exponential function.
- Thus, we do not necessarily treat infinite series.
- This formula gives fast and concise expression.
- We need diagonalization for general case.

$$A = \begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix}$$

$$\exp\begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix}^2 + \frac{1}{6}\begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix}^3 + \cdots$$

14 | 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

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

14 | EXPONENTIAL : ROTATION

- 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.

EXPONENTIAL :UNIPOTENT

$$\exp\begin{pmatrix}0 & 0 & b_1\\ 0 & 0 & b_2\\ 0 & 0 & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & b_1\\ 0 & 1 & b_2\\ 0 & 0 & 1\end{pmatrix}$$

15 | EXPONENTIAL : UNIPOTENT

- For a nilpotent matrix (some power of A is zero matrix), the matrix exponential is a finite sum.
- A typical example is a translation matrix (homogenous expression.)

$$A = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}$$

$$\exp(A) = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \frac{1}{4!}A^4 + \frac{1}{5!}A^5 + \cdots$$

$$A^3 = -|\mathbf{u}|^2 A$$

$$|\mathbf{u}| = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

16 | EXPONENTIAL : RODRIGUES

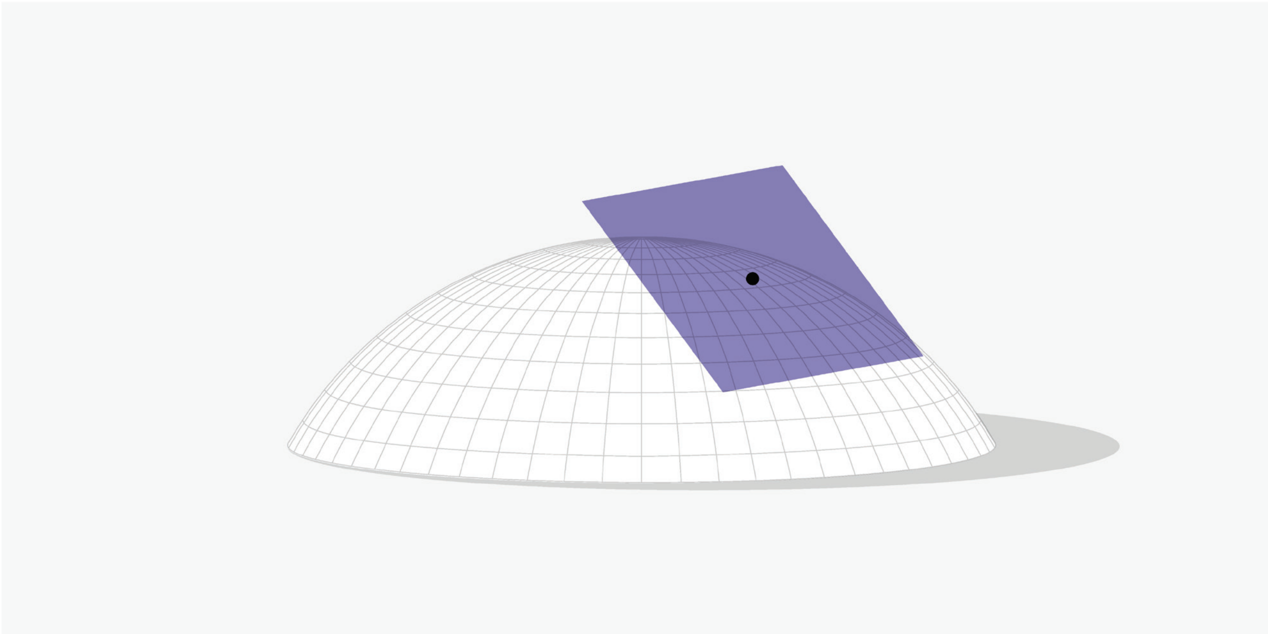
$$\exp(A) = I + \left(1 - \frac{1}{3!}|\mathbf{u}|^2 + \frac{1}{5!}|\mathbf{u}|^4 - \cdots\right) A + \left(\frac{1}{2} - \frac{1}{4!}|\mathbf{u}|^2 + \cdots\right) A^2$$

EXPONENTIAL : RODRIGUES

$$\exp(A) = I + \frac{\sin|\mathbf{u}|}{|\mathbf{u}|}A + \frac{1-\cos|\mathbf{u}|}{|\mathbf{u}|^2}A^2$$

16 | EXPONENTIAL : RODRIGUES

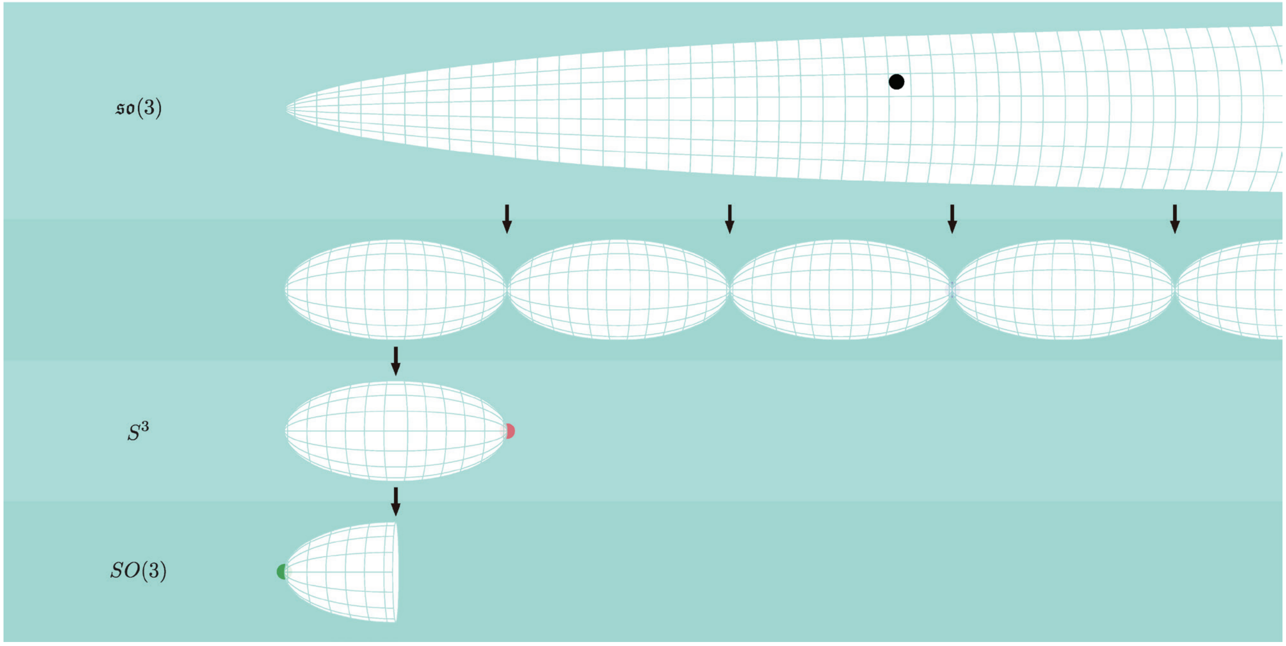
- For a skew symmetric matrix of size three,
  the matrix exponential has a closed expression using
  trigonometric functions.
- This is one form of Rodrigues formula.
- The rotation axis can be regarded as an eigenvector of A.
- The rotation angle can be regarded as a Frobenius norm of A.



17 | LIE ALGEBRA

- A <u>manifold</u> is a fancy name of curved space
- A matrix group is a curved space with a group structure:
  it is called a <u>Lie group.</u>
- A tangent space is a linear approximation of a curved space
- A tangent space of Lie group is called <u>Lie algebra.</u>
- Dual number is convenient for computation of Lie algebra.

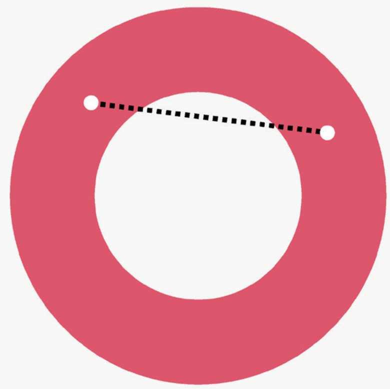
18 | SINGULARITIES OF EXPONENTIAL MAP



18 | SINGULARITIES OF EXPONENTIAL MAP

- An advanced topic (see [Ochiai14] or [Anjyo14] for details).
- This animation illustrates a precise behavior of matrix exponential for three-dimensional rotations as well as unit quaternion.
- The pinched points in the second figure shows singularity of the matrix exponential mapping.
- The third and fourth figures rephrase the property in section 7.

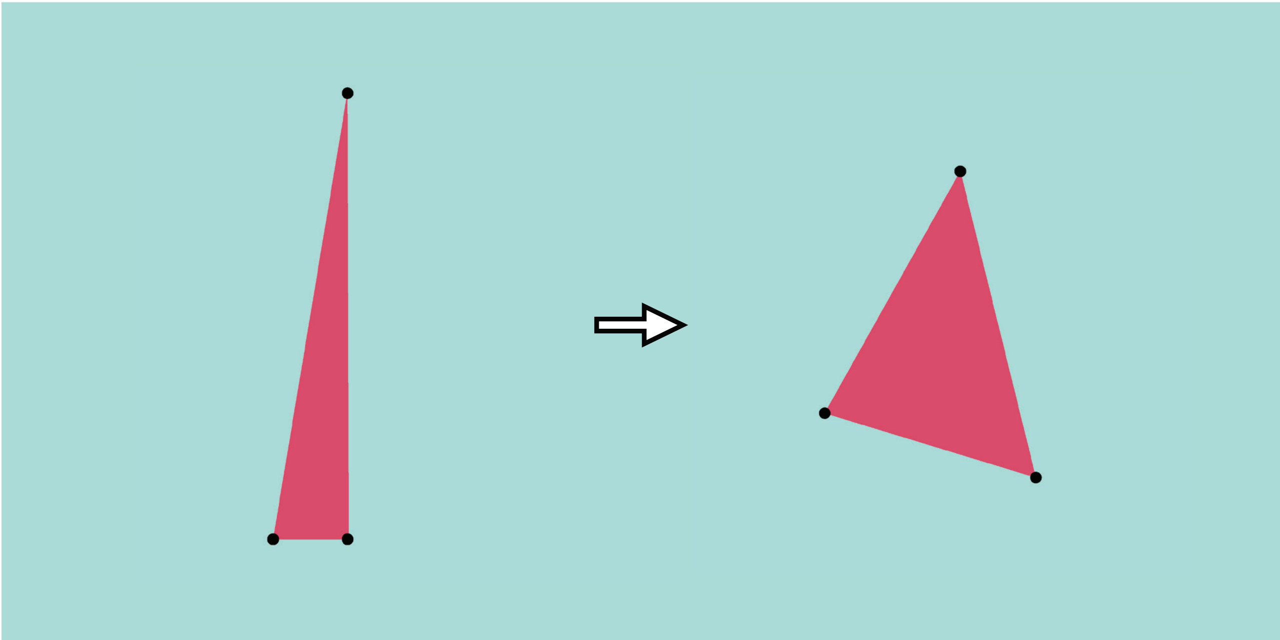
19 | FIELD OF BLENDING



19 | FIELD OF BLENDING

19 | 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).

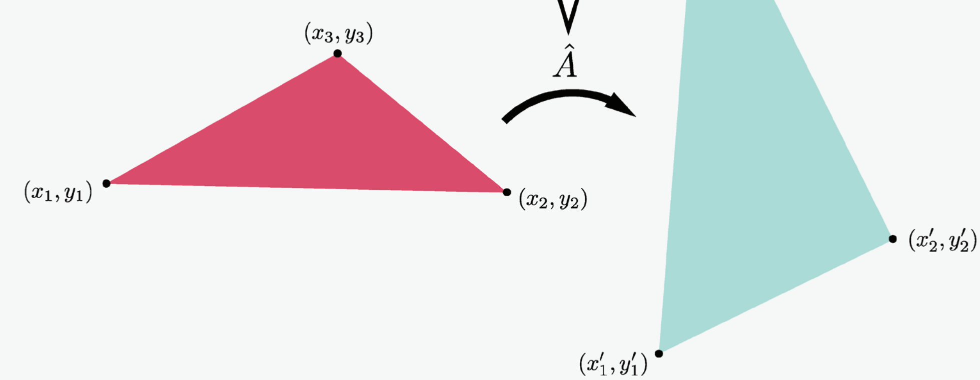


20 | INTERPOLATION OF TRIANGLES

- Let us consider how to interpolate between two given triangles.
- One answer to the above question is a linear interpolation of the corresponding vertices.
- In this section we try to use an interpolation of affine transformations.

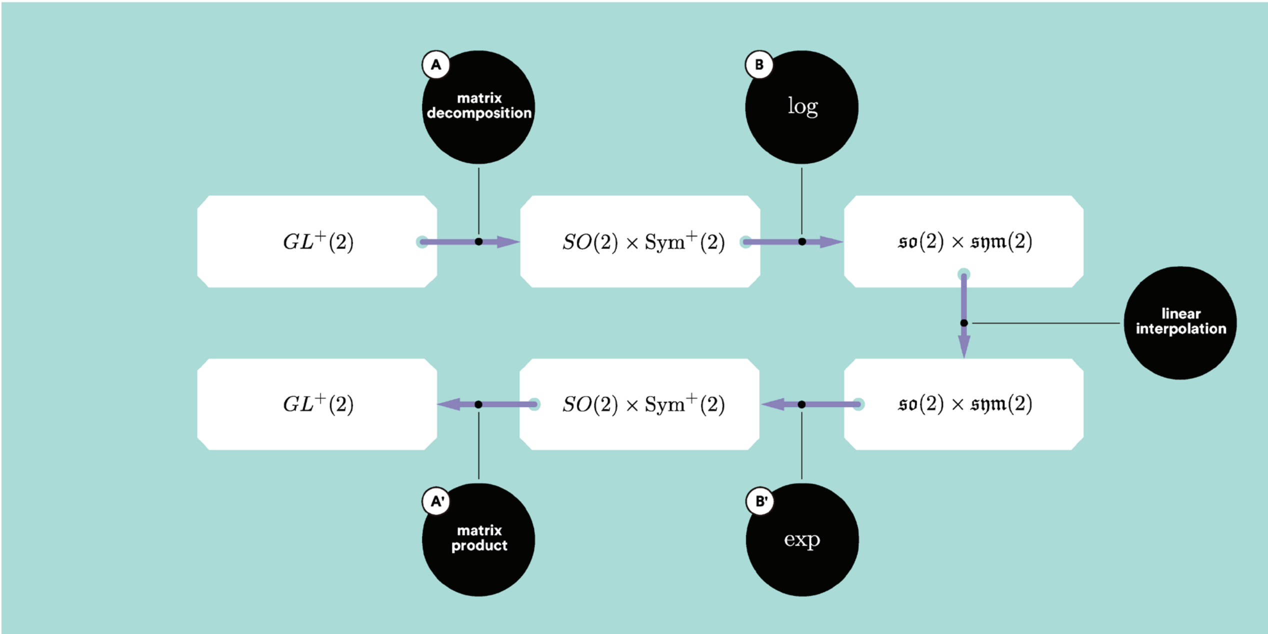
AFFINE TRANSFORMATION

$$\hat{A} = \begin{pmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}^{-1}$$



21 | SIMPLY TRANSITIVE ACTION OF AFF(2)

- $\hat{A}$ gives the unique affine transformation that maps (x_i, y_i) to (x'_i, y'_i), respectively, for i = 1,2, and 3.
- In our formulation, a linear interpolation between the two triangles is reduced to a linear interpolation between the 2D identity matrix I and A (the linear part of $\hat{A}$).



22 | LOG-EXP INTERPOLATION

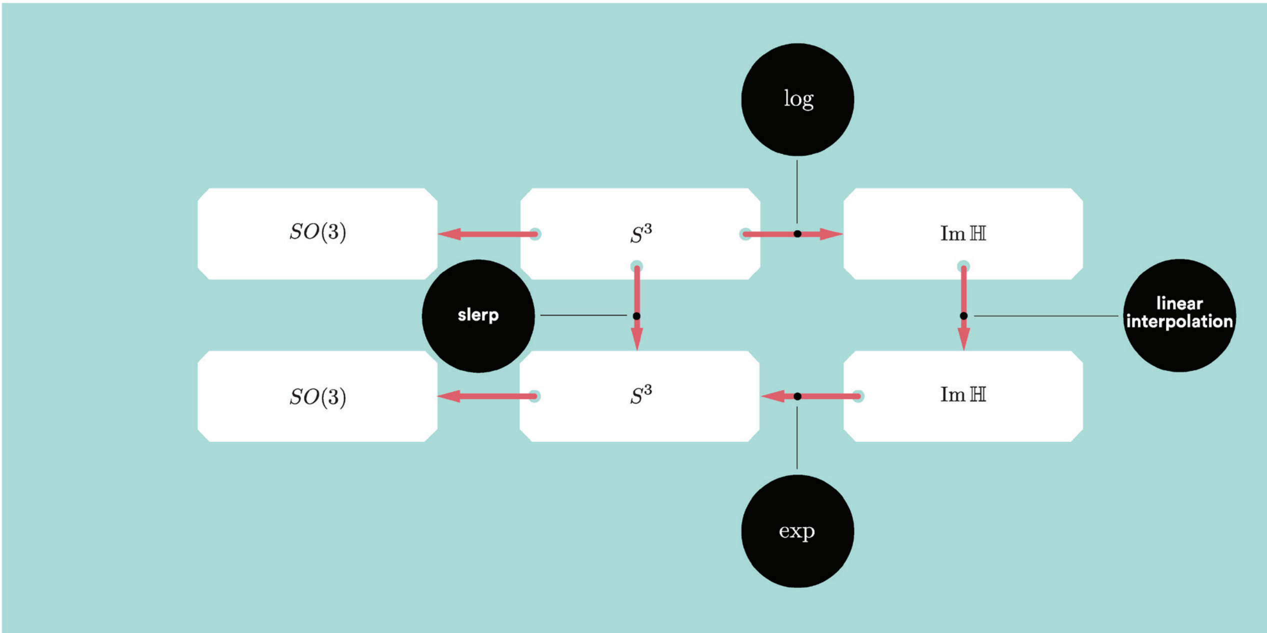
- The matrix decomposition in this diagram is based on the polar decomposition: $A = RS$, where R is a rotation matrix and S is a positive definite symmetric matrix.

- The map Ⓐ in the diagram is bijective (i.e., one-to-one and onto).

- The log map Ⓑ in the diagram is defined with

$$\log \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} \text{ and } \log(S),$$

where $R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$.



23 | SLERP

- This diagram simply shows that

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

- It also explains the following formula:

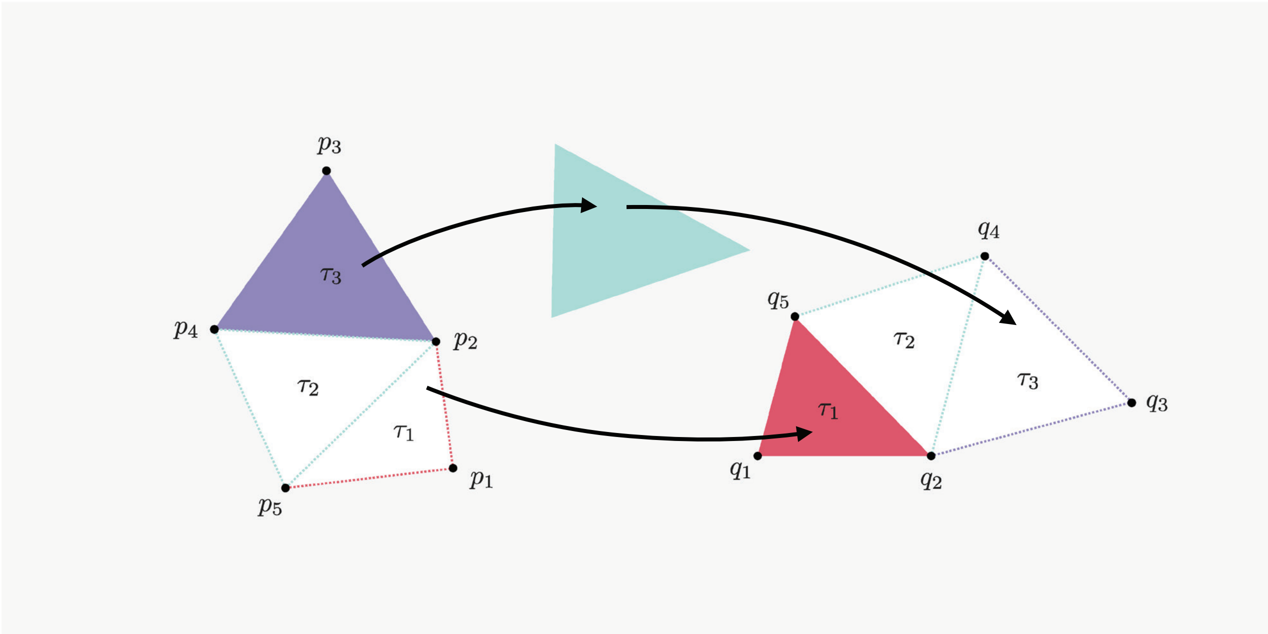
$$\mathrm{slerp}(q_0, q_1, t) = \frac{\sin((1-t)\theta)}{\sin\theta} q_0 + \frac{\sin(t\theta)}{\sin\theta} q_1,$$

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

because we have:

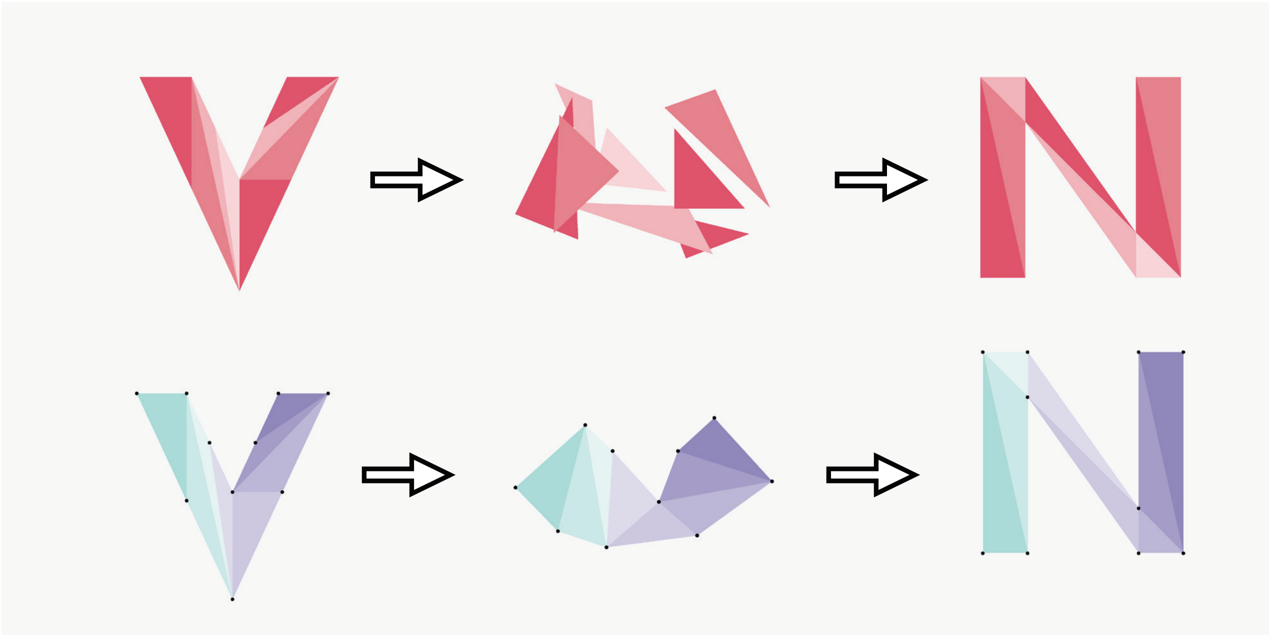
$$\mathrm{slerp}(q_0, q_1, t) = \mathrm{slerp}(1, q_1 q_0^{-1}, t) q_0,$$
$$\mathrm{slerp}(1, \exp(\theta\mathbf{u}), t) = \exp(t\theta\mathbf{u}).$$

24 | LOCAL AFFINE MAPS



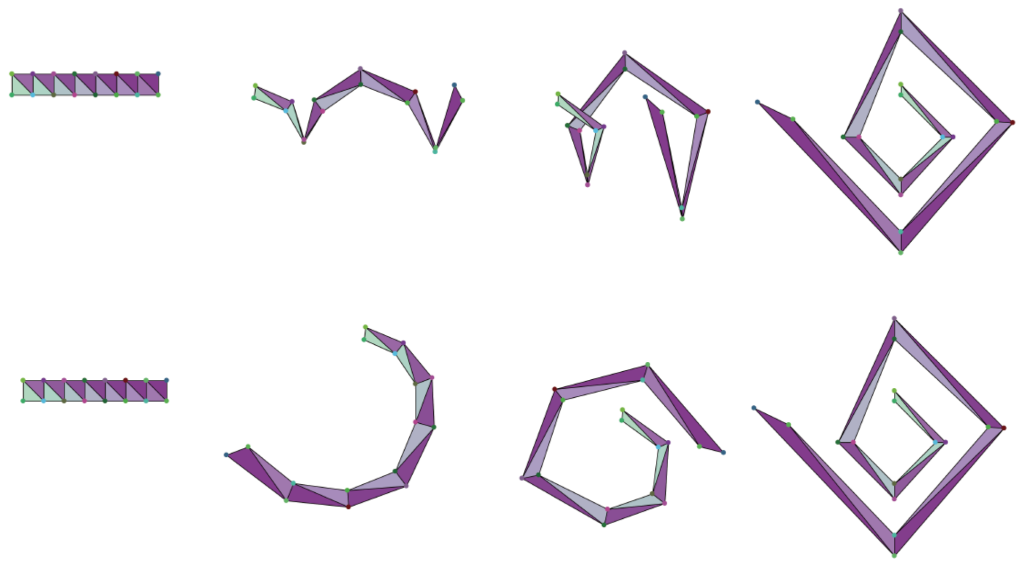
24 | LOCAL AFFINE MAPS

- A local affine map simply means the affine transform that gives one-to-one mapping for a given triangle pair.



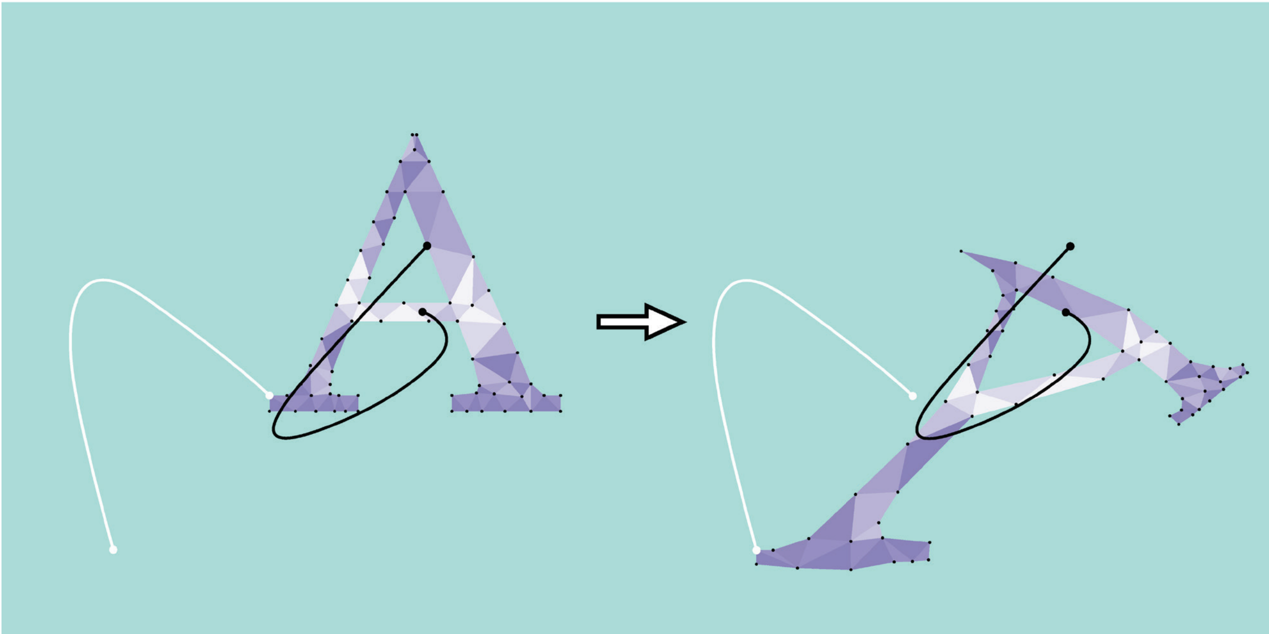
25 | LOCAL TO GLOBAL

- Next we consider how to make a morphing animation between 2D shapes consisting of triangles.
- Suppose that two 2D shapes are compatibly triangulated.
  - Each of the 2D shapes (say, V and N in the above figure) are triangulated,
  - There's a one-to-one mapping between the sets of triangles of these 2D shapes.
- To make a morphing animation from V to N, we consider how to interpolate between the triangle pair.
  - A simple interpolation (in section 22) between the local affine transform of the triangle pair cannot work well.
  - We thus need to introduce an energy-constraint for interpolating all the local affine transforms.



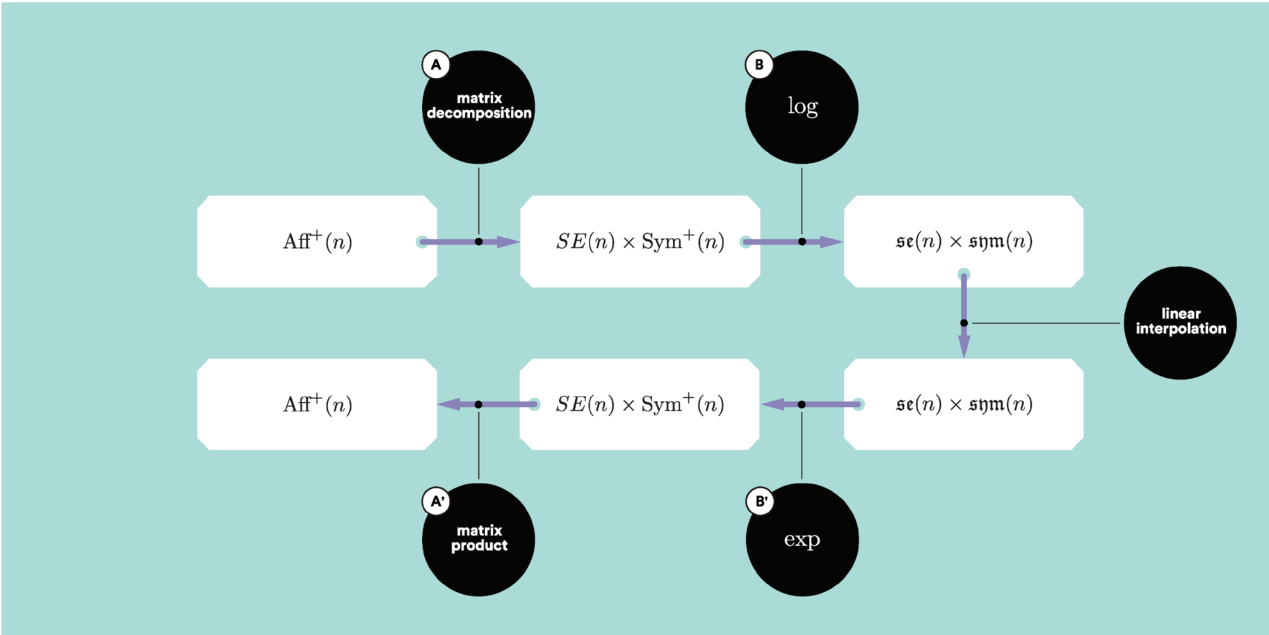
26 | GLOBAL INTERPOLATION: ROTATION CONSISTENCY

- Our energy model is useful for morphing between two 2D shapes that are compatibly triangulated (see [Ochiai14] for details).
- An interesting features of this model is rotation consistency.
  - Our energy model can deal with large rotations (over 180 degrees).
  - The top animation is obtained with the original ARAP formulation, while the bottom is by our model.



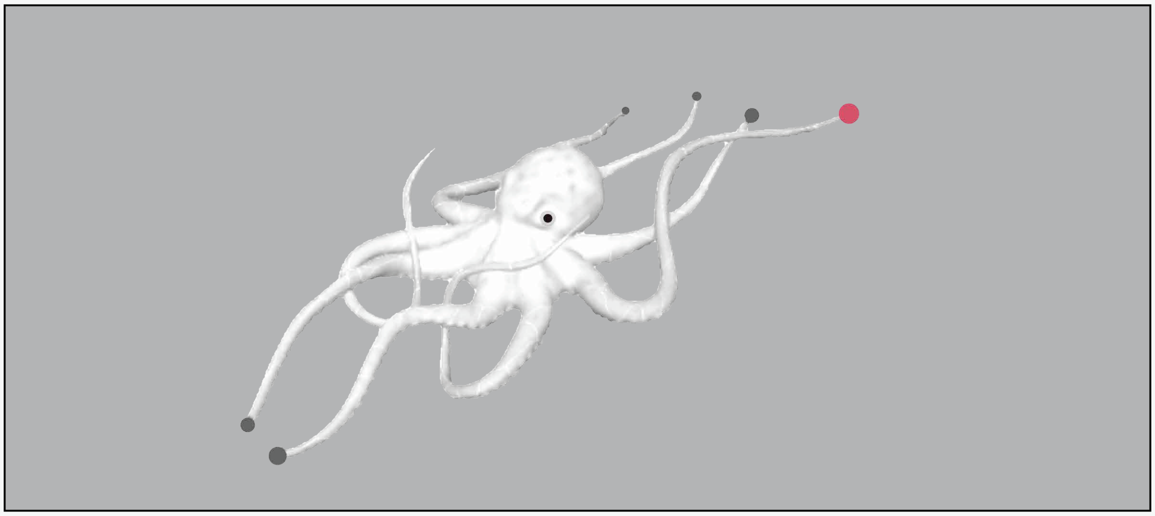
27 | GLOBAL INTERPOLATION: PATH CONSTRAINT

- A few more feature of this model are regarding path constraints (see [Ochiai14] for details):
  - Loci of several vertices can follow a specified curve (see the above animation).
  - The barycenter can trace a specified locus that changes over time.



28 | PARAMETRIZATION MAP AND ITS INVERSE

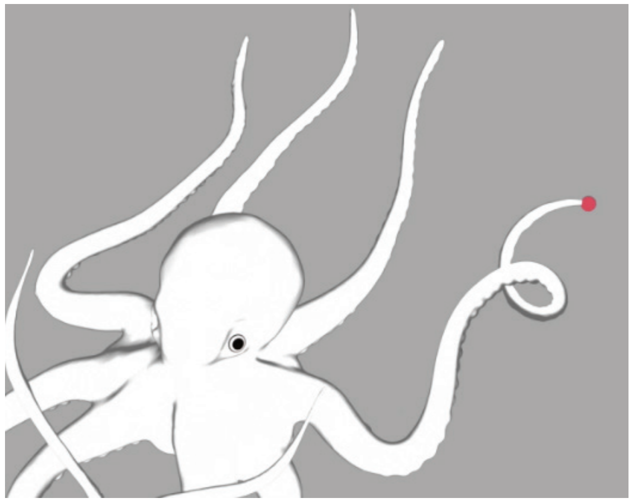
- The above diagram holds for n = 2 and 3.
  - This is derived from an argument similar to the Log-Exp interpolation (section 22).



GRAY HANDLES FIXED RED HANDLE UNDER OPERATION

29 | POISSON MESH EDITING

- Poisson mesh editing is commonly used for mesh edit and deformation/animation.
  - [key idea] Solving Poisson equation under Dirichlet boundary condition means stitching together the previously disconnected triangular meshes.
  - Edit means changing the boundary condition.



29 | POISSON MESH EDITING

- Rotation and scaling can be performed efficiently with our framework.
  - Log-Exp interpolation solves the gimbal lock problem.

30 | CAGE-BASED DEFORMER

- There are several cage-based deformers available, which also exhibit fundamental deformation commands including rotation.
- The video demonstrates our cage-based technique (see Liu and Anjyo's article in SIGGRAPH Asia 2015 Technical Briefs for details).
    - This deformer allows real-time edit, including local control of scale and stretch.

References

- The following references detail the lecture of this course. The original references of CG techniques and mathematics discussed in this course can also be found in them:

[Ochiai14] Ochiai and Anjyo: Mathematical Basics of Motion and Deformation in Computer Graphics, SIGGRAPH2014 Course Notes 19.

[Anjyo14] Anjyo and Ochiai: Mathematical Basics of Motion and Deformation in Computer Graphics, Synthesis Lectures on Computer Graphics and Animation, Morgan & Claypool Publishers 2014

- An updated version of the course notes will be found at:

http://mcg.imi.kyushu-u.ac.jp/

ABOUT THE VIDEO:

CONCEPTUAL DESIGN AND DIRECTION	HIROYUKI OCHIAI, KEN ANJYO
ART DIRECTION AND DESIGN	SHINGO KITADA ERI SATO
ANIMATION	KAZUYUKI ISHIHARA HITOMI OHUCHI SAYOKO HABA LIU GENGDAI RICHARD ROBERTS SHIZUO KAJI
MUSIC	TAKERU OKUMURA
PRODUCTION DESIGN	AYUMI KIMURA
ACKNOWLEDGMENTS	OLM DIGITAL, INC. JAPAN SCIENCE AND TECHNOLOGY AGENCY (JST)