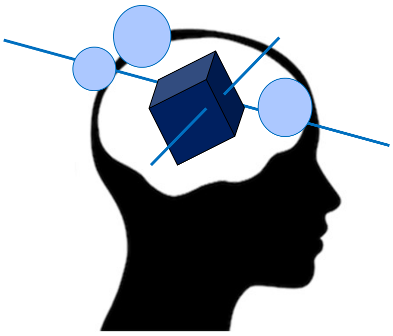
3D videogames

Mathematical representations of roto-translations

(a brief note)



Marco Tarini

130

Representations for roto-rotations (just a brief note)

- So far, we assumed that the **rotation** and **translation** components of a transformation are stored *separately*
  - We have seen reasons why this is convenient
- Mathematical representations exist, that store rotation + translation (aka *roto-translations*, aka *rigid* transformations) jointly:
  - 4x4 matrices (we have seen them, their pros and cons)
  - Dual quaternions

131

Representations for ~~rotations~~ **roto-translations**

aka “rigid” transforms

- 3×3 Normal Matrices
- Euler Angles
- Angle & Axis
- Quaternions

+ Translation (displacement vector)

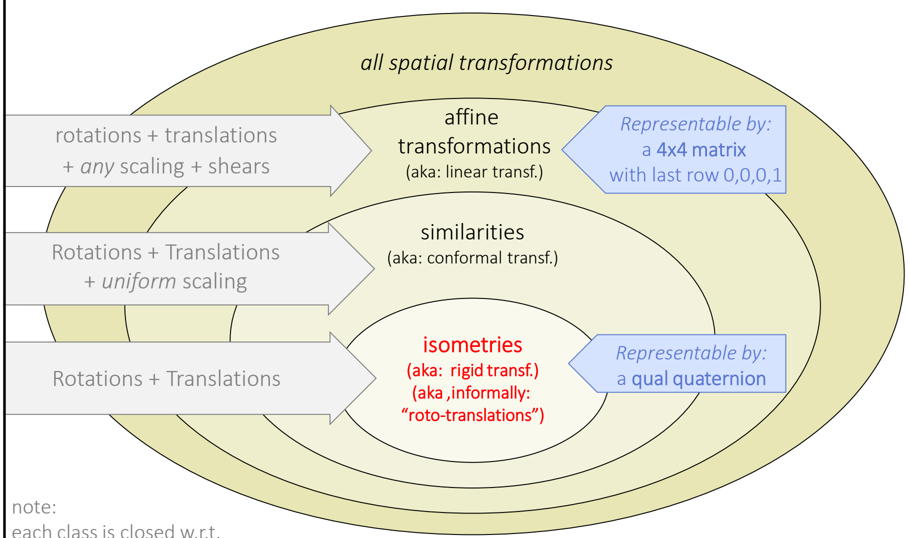
OR:

- 4×4 Matrices (or 3×4)
- Dual Quaternions

As there’s no need to store the last row, it’s (0,0,0,1)

132

A math classification of spatial transformations



note:
each class is closed w.r.t.
cumulation, and (when they are invertible) inversion

133

Marco Tarini
Unviersità degli studi di Milano

The math of Dual Quaternions in a nutshell 1/3

- Dual quaternions are a mathematical way to represent a roto-translation (aka, a "rigid motion")
- They result in very good interpolation between 2 (or more) roto-translations
- They are sometimes used in animation techniques
  - See lecture about skeletal animations, later

134

The math of Dual Quaternions in a nutshell 1/3

- New "fantasy" assumption: there is a ε such that $\varepsilon \neq 0$, $\varepsilon^2 = 0$
- A "dual number": $a + \varepsilon b$, with $a,b \in \mathbb{R}$
- A "dual quaternion": $p + \varepsilon q$, with $p,q \in \mathbb{H}$ ← quaternion set
- In can be stored as eight scalars (four for p , plus four for q)

135

The math of Dual Quaternions in a nutshell 3/3

- A dual quaternion (p + ε q) can represent (under distinct conditions):
  - a 3D point, OR
  - a 3D vector/versor, OR,
  - a roto-translation
    then, the "primal" quaternion **p** is the rotation, and
    the "dual" quaternion **q** encodes the translation (in some ways)
  - (OR, nothing)

We won't see the details,
but in case you'll ever need it,
you'll find them in the course web page

- We can apply a roto-translation to a 3D point or vector, by simply manipulating their representations
  - With multiplications & conjugations
  - Always keeping in mind all the "fantasy assumptions"
  - Same principle of representing 3D rotations with quaternions (or 2D rotations with complex numbers)

136

Q: why dual-quaternions?
A: better interpolation of rigid motions

- Problem with interpolating rotations and translations separately (intuitively):
  - must choose "which one goes first" (R then T, or, T then R)?
  - Different choices ⟶ very different interpolation results
  - Often, neither is what you had in mind
- Dual quaternions = a better* math abstraction to model roto-translations
  - * better interpolation of roto-translations

137