Tracker Alignment

using Hand-Eye Calibration

Sebastian Grembowietz

Outline

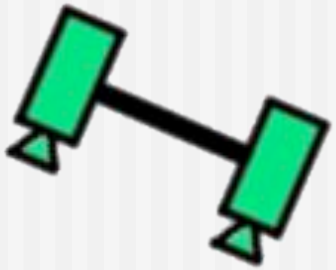
- Introduction
  - Tracker Alignment
    - Definition and Applications
  - Achieving Aligned Trackers
- Solving AX=XB
  - classical solution (Tsai)
    - Splitting and solving AX=XB
  - More robust solution (Daniilidis)
    - Complex Numbers, Dual Quaternions and Screws
    - Solving aq = qb
- Improving accuracy

Introduction of myself

- Sebastian Grembowietz
  - 23 years old
  - studying computer science at TUM
  - 5th semester
  - Current focus: Augmented Reality

Tracker Alignment

- Hybrid trackers



- Combining strengths
- Minimizing weaknesses

Applications

- More robust sensors:
  - ART + camera:
    - fewer dependence on illumination
    - obtaining 3d data from camera

  - Aurora + ART:
    - partly independent from line-of-view
    - high precision

Applications

- Robotics aided model generation
  - Robot arm and laser scanner combined

Applications

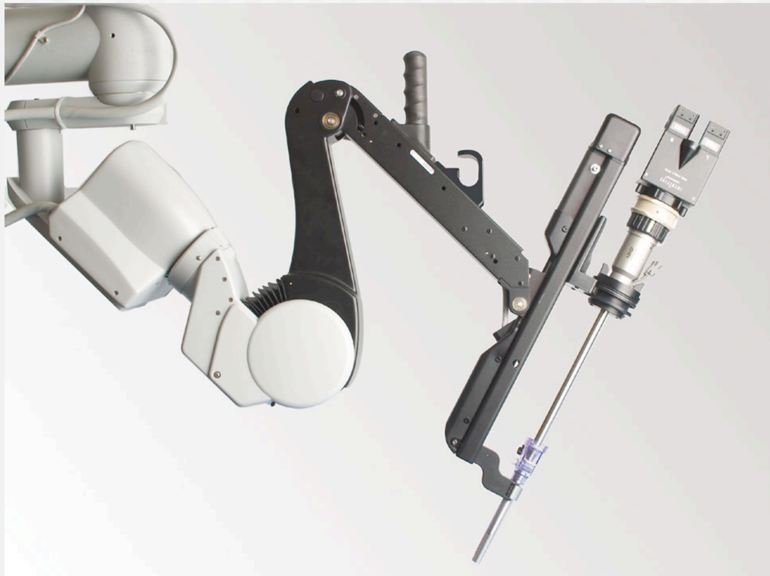
- Robotics aided model generation
  - Taking measurements for custom-made clothing

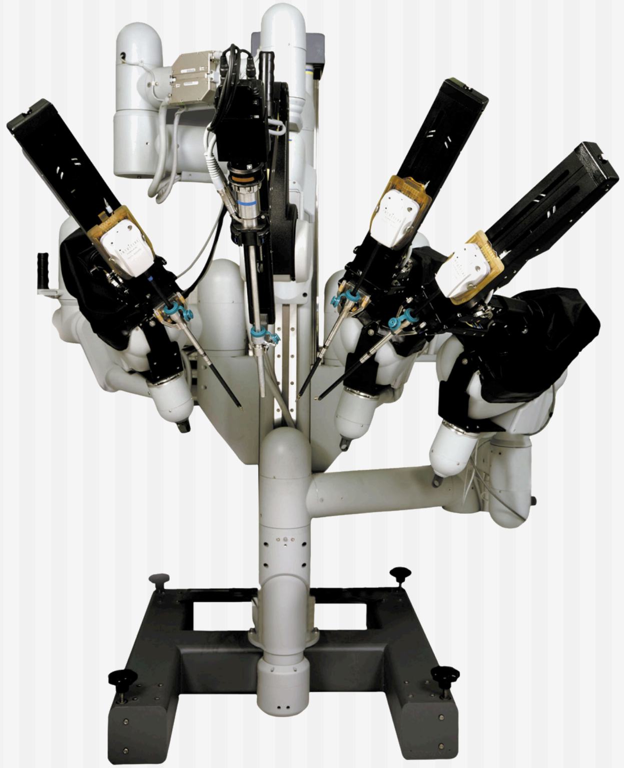
Applications

- Automated part assembly
  - Automatic welding guided by feedback from cameras

Applications

- Endoscopic Surgery
  - minimally invasive surgery



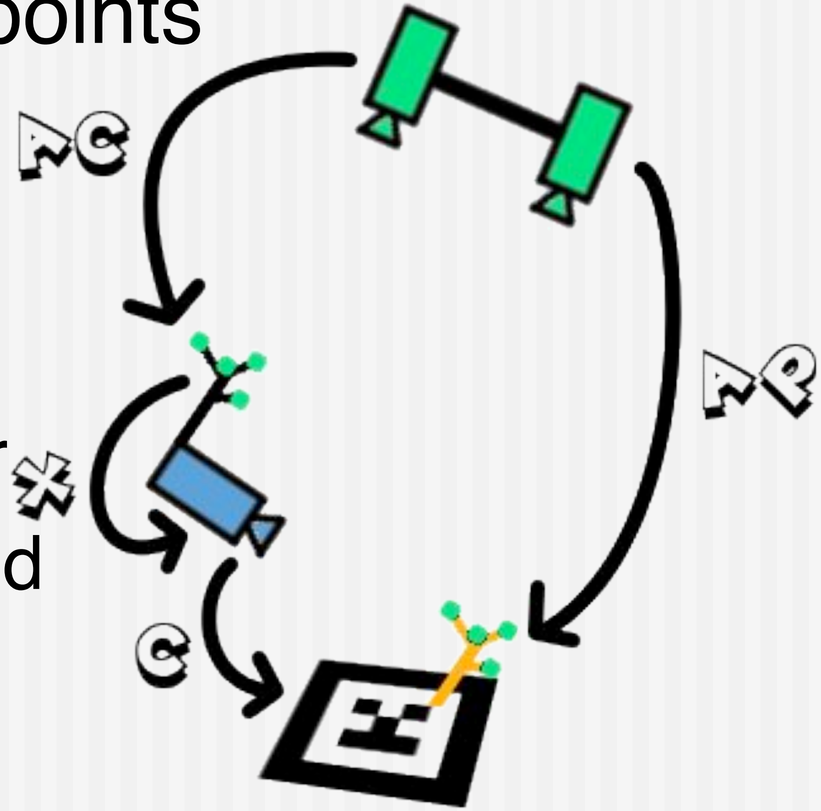


Aligning Trackers

- Forward Engineering
  - Hybrid tracker entirely designed
  - Only one way of mounting possible
  - Fixed alignment

Aligning Trackers

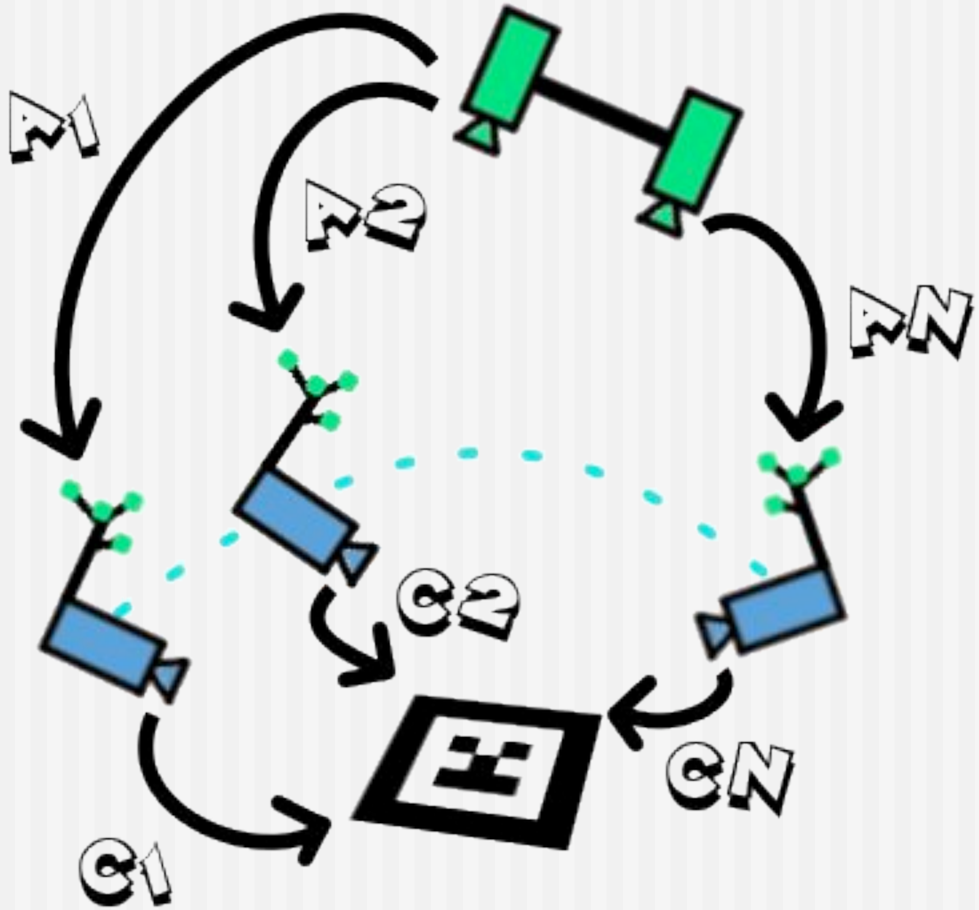
- Manually registering points



  - User points on marker
  - Correspondences used for tracker alignment

Aligning Trackers

- Automatic alignment from movements

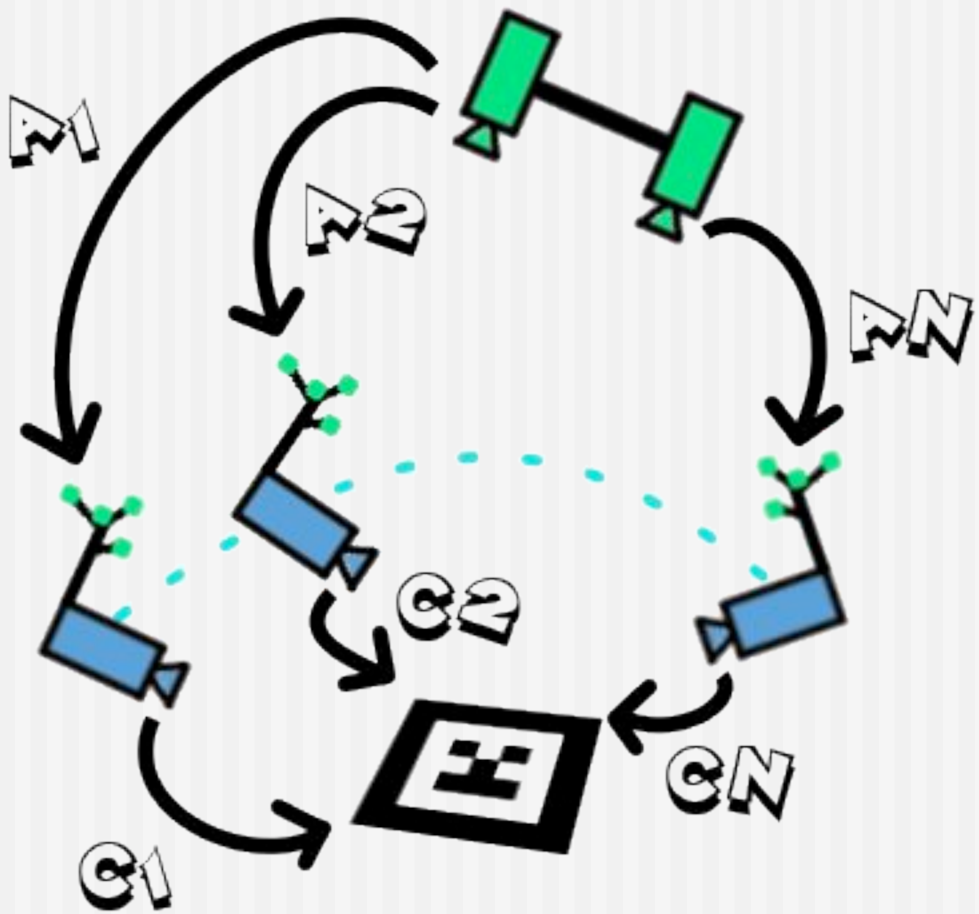


Setup

- Transforms obtained

$$A_i = \begin{bmatrix} R_{A_i} & \vec{t}_{A_i} \\ \vec{0}^T & 1 \end{bmatrix}$$

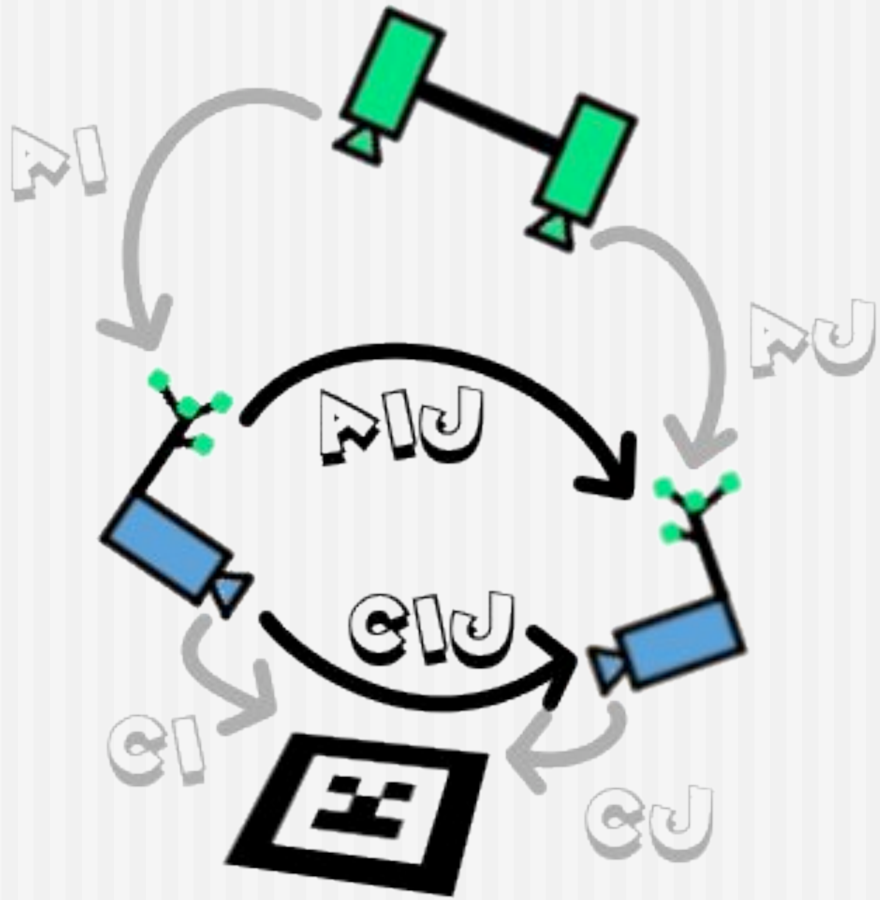
$$C_i = \begin{bmatrix} R_{C_i} & \vec{t}_{C_i} \\ \vec{0}^T & 1 \end{bmatrix}$$



Setup

- Transforms computed

$$A_{ij} = \begin{bmatrix} R_{A_{ij}} & \vec{t}_{A_{ij}} \\ \vec{0}^T & 1 \end{bmatrix}$$
$$= A_j A_i^{-1}$$
$$C_{ij} = \begin{bmatrix} R_{C_{ij}} & \vec{t}_{C_{ij}} \\ \vec{0}^T & 1 \end{bmatrix}$$
$$= C_j^{-1} C_i$$



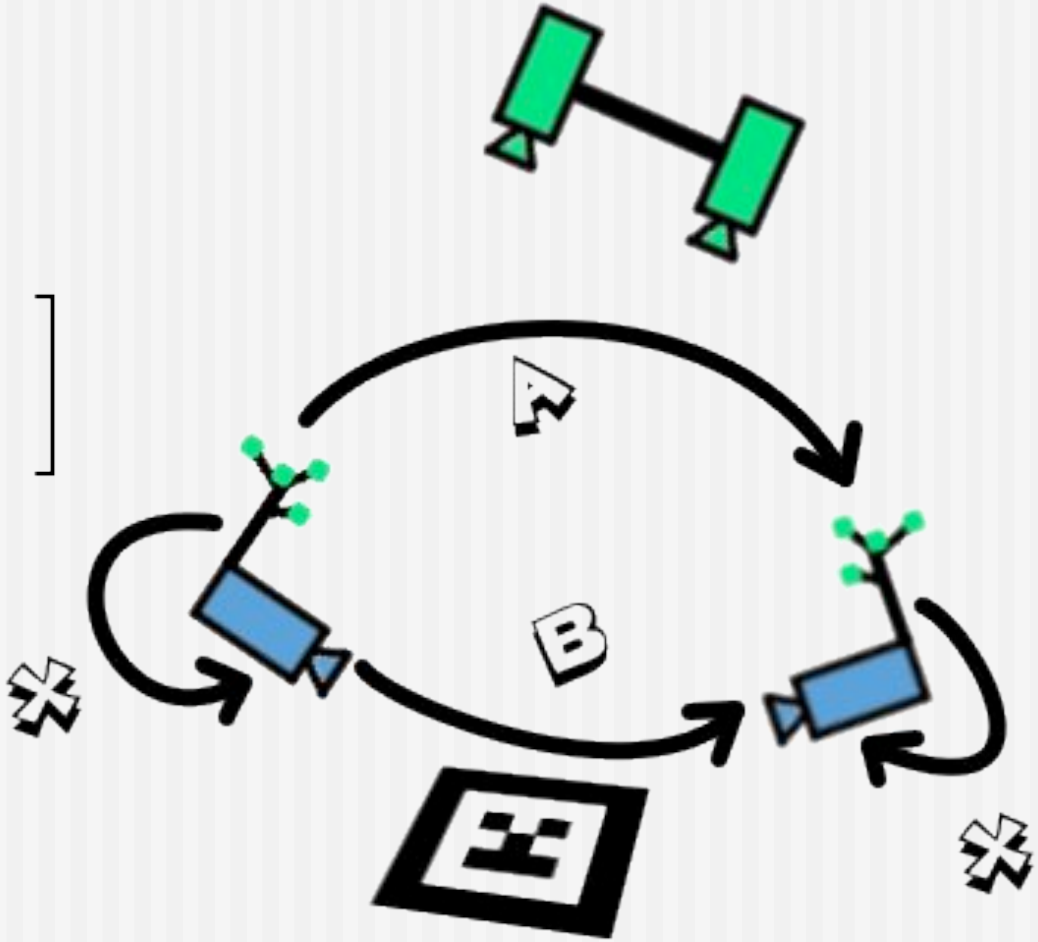
Setup

- Transform wanted

$$X = \begin{bmatrix} R_X & \vec{t}_X \\ \vec{0}^T & 1 \end{bmatrix}$$

from

$$AX = XB$$



Classical Solution

- 1989: Roger Tsai and Reimar Lenz




Splitting AX=XB

- AX=XB in matrix form

$$\begin{bmatrix} R_A & \vec{t}_A \\ \vec{O}^T & 1 \end{bmatrix} \begin{bmatrix} R_X & \vec{t}_X \\ \vec{O}^T & 1 \end{bmatrix} = \begin{bmatrix} R_X & \vec{t}_X \\ \vec{O}^T & 1 \end{bmatrix} \begin{bmatrix} R_B & \vec{t}_B \\ \vec{O}^T & 1 \end{bmatrix}$$

- multiplied out

$$R_A R_X = R_X R_B$$

$$R_A \vec{t}_X + \vec{t}_A = R_X \vec{t}_B + \vec{t}_X$$

2 Phases

- Solving AX=XB in 2 phases
- Phase 1 - computing R_X
  - modified Rodrigues formula
- Phase 2 - computing $\vec{t}_X$

Phase 1 - Rotation

- Rodrigues formula corresponding to R_X

$$\vec{r} = 2\sin\frac{\theta}{2}\vec{n}$$

- Rodrigues formula computed soon

$$\vec{r'}_x = \lambda\vec{r}_x = \frac{1}{\sqrt{4 - |\vec{r}_x|^2}}\vec{r}_x$$

Phase 1 - Rotation

- Step 1 - For all movement pairs

$$(\vec{r}_{A_{ij}} + \vec{r}_{C_{ij}}) \times \vec{r'}_x = \vec{r}_{C_{ij}} - \vec{r}_{A_{ij}}$$

rewritten to

$$[\vec{r}_{A_{ij}} + \vec{r}_{C_{ij}}]_x \vec{r'}_x = \vec{r}_{C_{ij}} - \vec{r}_{A_{ij}}$$

and solved using linear least squares

$$[\vec{a}]_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Phase 1 - Computing Rotation

- Step 2 - obtain $\vec{r}_x$

$$\vec{r}_x = \frac{2\vec{r'}_x}{\sqrt{1 + |\vec{r'}_x|}}$$

Phase 2 - Translation

- R_X known
- Computation of $\vec{t}_X$ from

$$(R_{A_{ij}} - I)\vec{t}_x = R_X \vec{t}_{Cij} - \vec{t}_{A_{ij}}$$

using linear least squares

Modern Solution

- 1998: Kostas Daniilidis

Complex Numbers

- Short introduction:

$$z = a + bi = (a, b)^T$$
$$i^2 = -1$$

- Addition

$$\begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} a + c \\ b + d \end{pmatrix}$$

Complex Numbers

- Multiplication

$$\begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac - bd \\ ad + bc \end{pmatrix}$$

- Conjugate

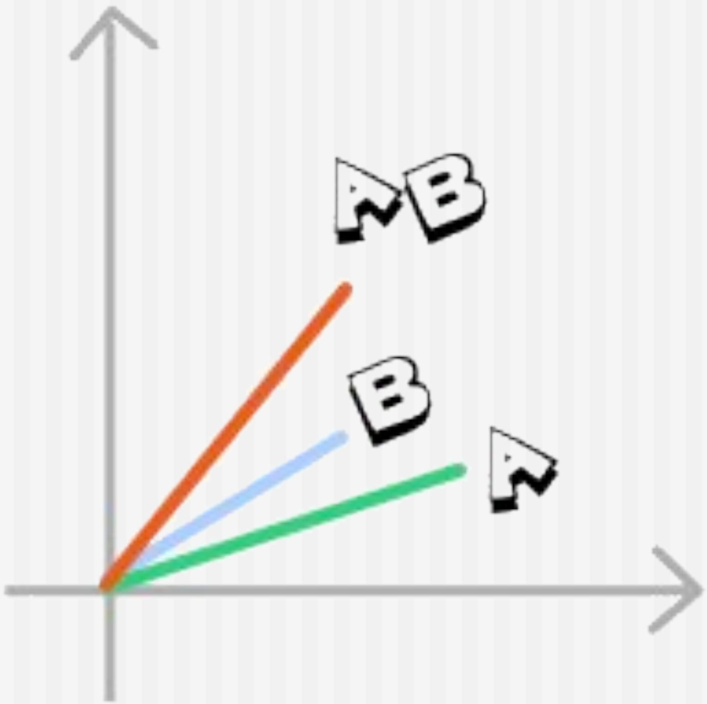
$$\bar{a} = a - bi = (a, -b)$$

- Norm

$$|x| = \sqrt{a^2 + b^2} = |(a, b)^T|$$

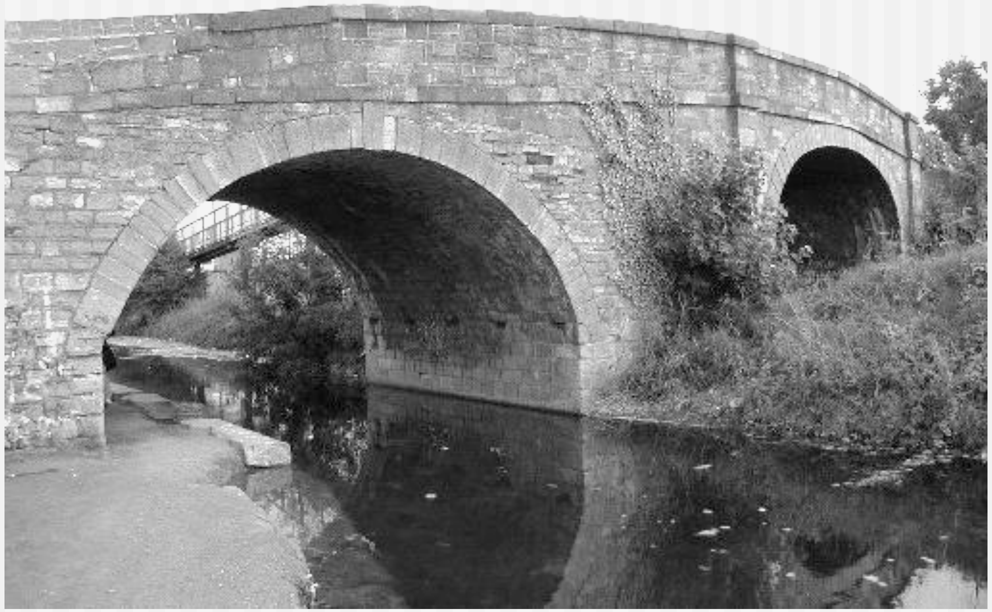
Complex Numbers

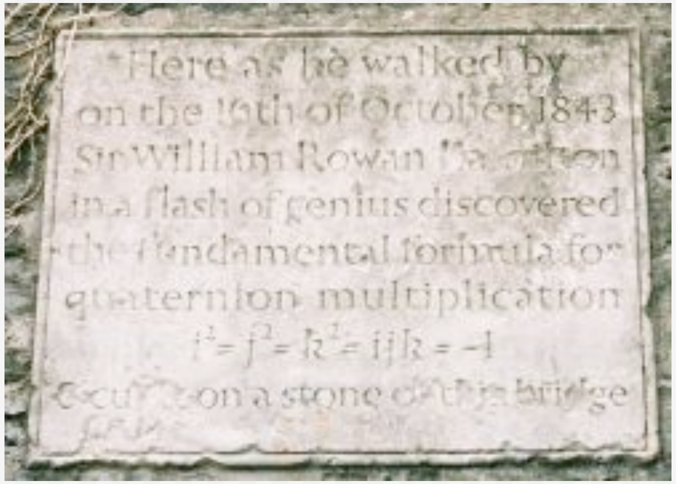
- Multiplication = Rotation in 2D
- angles added
- norms multiplied



Quaternions

- Discovered by Hamilton in 1834





Quaternions

- extension of complex numbers to 4D

$$q = \begin{pmatrix} s \\ \vec{q} \end{pmatrix} = s + q_1 i + q_2 j + q_3 k$$

- imaginary units *i, j, k*

	i	j	k
i	-1	k	$-j$
j	$-k$	-1	i
k	j	$-i$	-1

Quaternion

- Conjugate

$$\bar{q} = \begin{pmatrix} s \\ -\vec{q} \end{pmatrix}$$

- Addition

$$p + q = \begin{pmatrix} r \\ \vec{p} \end{pmatrix} + \begin{pmatrix} s \\ \vec{q} \end{pmatrix} = \begin{pmatrix} r + s \\ \vec{p} + \vec{q} \end{pmatrix}$$

Quaternions

- Multiplication

$$
\begin{aligned}
pq =& rs + \vec{q}\vec{p} + s\vec{p} + r\vec{q} + \vec{p} \times \vec{q} \\
=& \begin{pmatrix} rs - p_1q_1 - p_2q_2 - p_3q_3 \\ q_1r + sp_1 + q_2p_3 - q_3p_2 \\ q_2r + sp_2 + q_3p_1 - q_1p_3 \\ q_3r + sp_3 + q_1p_2 - q_2p_1 \end{pmatrix}
\end{aligned}
$$

Quaternions

- Norm

$$|q| = \sqrt{q\bar{q}} = \sqrt{s^2 + q_1^2 + q_2^2 + q_3^2}$$

- Unit quaternions: norm of 1

Using Quaternions

- Unit quaternion used for rotation in 3D

$$q = \begin{pmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2}\vec{n} \end{pmatrix}$$

- Rotating x to y

$$\begin{pmatrix} 0 \\ \vec{y} \end{pmatrix} = q \begin{pmatrix} 0 \\ \vec{x} \end{pmatrix} \bar{q}$$

Superiority of quaternions...

...concerning rotations:

- size
  - 4 values instead of 9
- interpolation
  - SLERP: Spherical Linear Interpolation
- no gimbal lock
- robustness thanks to easy normalization

Dual Numbers

- Definition

$$\breve{x} = a + b\epsilon = \begin{pmatrix} a \\ b \end{pmatrix} \qquad \epsilon^2 = 0$$

- Addition

$$\breve{a} + \breve{b} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix}$$

Dual Numbers

- Multiplication

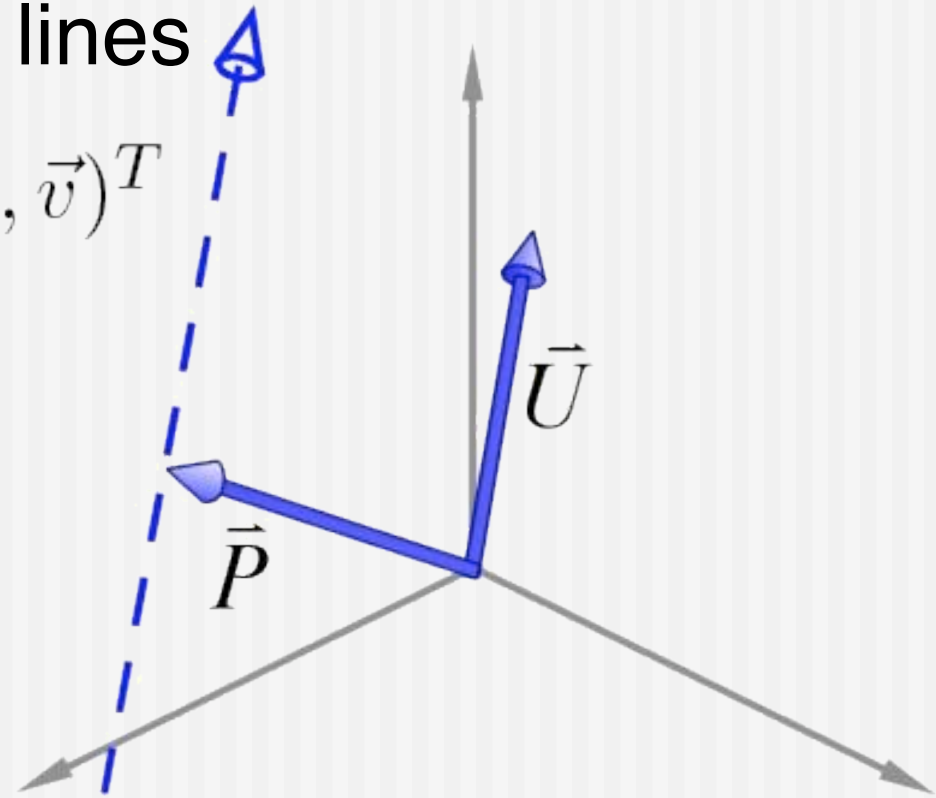
$$\check{a}\check{b} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$
$$= (a_1 + a_2\epsilon)(b_1 + b_2\epsilon)$$
$$= a_1 b_1 + a_1 b_2 \epsilon + a_2 b_1 \epsilon + a_2 b_2 \epsilon^2$$
$$= a_1 b_1 + (a_1 b_2 + a_2 b_1)\epsilon$$
$$= \begin{pmatrix} a_1 b_1 \\ a_1 b_2 + a_2 b_1 \end{pmatrix}$$

Plücker Coordinates

- Dual numbers used in vectors as representation of lines

$$L = (\vec{p}, \vec{u} \times \vec{p})^T = (\vec{u}, \vec{v})^T$$

- direction u
- moment v



Dual Quaternions

- Dual Quaternion = Quaternion using dual numbers

$$\check{q} = (\check{s}, \vec{\check{q}})^T = q + q\prime\epsilon = (q, q\prime)^T$$

with q, q‘ quaternions

Line transformations

- Line as Plücker coordinate $l = (\vec{l}, \vec{m})^T$
  with additional constraint $|\vec{l}| = 1$

- line transformations by $(R, \vec{t})$

- line transformations using dual quaternions $\check{l}_b = \check{q}\check{l}_a\bar{\check{q}}$

Line transformations

- Proof: From vector ...

$$\vec{l}_a = R\vec{l}_b$$

$$\vec{m}_a = R\vec{m}_b + \vec{t} \times R\vec{l}_b$$

$$x = (0, \vec{x})^T$$

... to quaternion ...

$$(0, \vec{t} \times \vec{q}) = \frac{1}{2}(q\bar{t} + tq)$$

$$l_a = q l_b \bar{q}$$

$$m_a = q m_b \bar{q} + \frac{1}{2}(q l_b \bar{q} \bar{t} + t q l_b \bar{q})$$

Line transformations

$$l_a = q l_b \bar{q}$$

$$m_a = q m_b \bar{q} + \frac{1}{2}(q l_b \bar{q} \bar{t} + t q l_b \bar{q})$$

$$q' = \tfrac{1}{2} t q$$

$$\check{q} = q + q' \epsilon$$

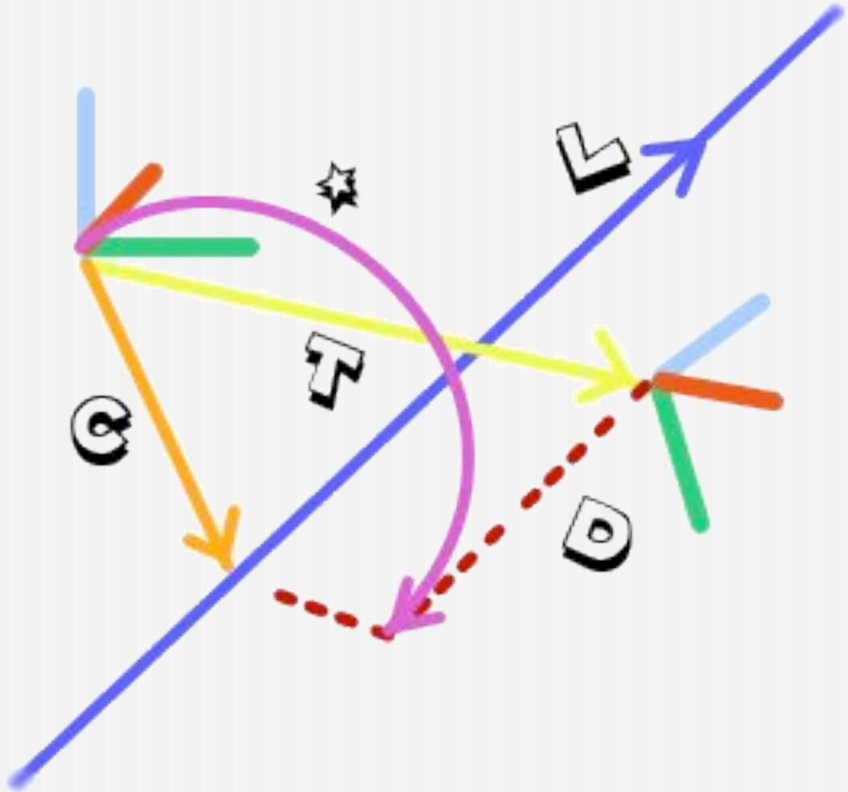
... to dual quaternions

$$l_a + m_a \epsilon = (q + q'\epsilon)(l_b + m_b \epsilon)(\bar{q} + \bar{q}'\epsilon)$$

or equivalently $\check{l}_a = \check{q} \check{l}_b \bar{\check{q}}$

Screws

- Rigid transformation (R, t) equivalent to screw
- rotation axis l not through origin
- pitch d
- same angle



Screw from Transform

- Direction l of screw axis
  - parallel to rotation axis of transform
- Pitch d $d = \vec{t}^T \vec{l}$
  - projecton of translation on rotation axis
- Moment m
  - using auxilliary point c

$$\vec{m} = \vec{c} \times \vec{l} = \frac{1}{2}(\vec{t} \times \vec{l} + \vec{l} \times (\vec{t} \times \vec{l}) \cot \frac{\theta}{2})$$

Screw to Dual Quaternion

$$\check{q} = \begin{pmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2}\vec{l} \end{pmatrix} + \begin{pmatrix} -\frac{d}{2}\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2}\vec{m} + \frac{d}{2}\cos\frac{\theta}{2}\vec{l} \end{pmatrix}\epsilon$$

$$= \begin{pmatrix} \cos\left(\frac{\theta+d\epsilon}{2}\right) \\ \sin\left(\frac{\theta+d\epsilon}{2}\right)(\vec{l}+\vec{m}\epsilon) \end{pmatrix}$$

$$= \begin{pmatrix} \cos\frac{\check{\theta}}{2} \\ \check{\vec{l}}\sin\frac{\check{\theta}}{2} \end{pmatrix}$$

Back to Tracker Alignment

- AX=XB with dual quaterions

$$\check{a}\check{q} = \check{q}\check{b} \qquad \check{a} = \check{q}\check{b}\bar{\check{q}}$$

- Screw Congruence Theorem
  - Pitch and angle of a screw invariant under rigid transformations

$$\sin\frac{\check{\theta}_a}{2}\begin{pmatrix} 0 \\ \vec{\check{a}} \end{pmatrix} = \check{q}\begin{pmatrix} 0 \\ \sin\frac{\check{\theta}_b}{2}\vec{\check{b}} \end{pmatrix}\bar{\check{q}} = \sin\frac{\check{\theta}_b}{2}\check{q}\begin{pmatrix} 0 \\ \vec{\check{b}} \end{pmatrix}\bar{\check{q}}$$

Tracker Alignment

- Simplified

$$\begin{pmatrix} 0 \\ \check{\vec{a}} \end{pmatrix} = \check{q} \begin{pmatrix} 0 \\ \check{\vec{b}} \end{pmatrix} \bar{\check{q}}$$

- Note: only motion of the screw axis left

Tracker Alignment

- Splitting $\check{a} = \check{q}\check{b}\bar{\check{q}}$ in dual and non-dual part

$$a = qb\bar{q}$$

$$a' = qb\bar{q}' + qb'\bar{q} + q'b\bar{q}$$

- rearranging and using $\bar{q}q' + \bar{q}'q = 0$

$$aq - qb = 0$$

$$(a'q - qb') + (aq' - q'b) = 0$$

Tracker Alignment

- Disregarding the scalar parts to set up the linear system

$$S \begin{pmatrix} q \\ q' \end{pmatrix} = 0$$

with S being 6x8 matrix

$$\begin{pmatrix} \vec{a} - \vec{b} & [\vec{a} + \vec{b}]_x & 0_{3\times 1} & 0_{3\times 3} \\ \vec{a'} - \vec{b'} & [\vec{a'} + \vec{b'}]_x & \vec{a} - \vec{b} & [\vec{a} + \vec{b}]_x \end{pmatrix}$$

Tracker Alignment

- Construct 6*n*x8 matrix T for *n* motion pairs

$$T = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{pmatrix}$$

- Compute SVD

$$T = USV^T$$

and obtain

$$\begin{pmatrix} q \\ q' \end{pmatrix} = \lambda_1 \vec{v_7} + \lambda_2 \vec{v_8} = \lambda_1 \begin{pmatrix} \vec{x_1} \\ \vec{y_1} \end{pmatrix} + \lambda_2 \begin{pmatrix} \vec{x_2} \\ \vec{y_2} \end{pmatrix}$$

Tracker Alignment

- Compute the lambdas using unit quaternion criterions

$$q^T q = 1$$
$$q^T q' = 0$$

- Constraits result in

$$\lambda_1^2 \vec{x_1}^T \vec{x_1} + 2\lambda_1\lambda_2 \vec{x_1}^T \vec{x_2} + \lambda_2^2 \vec{x_2}^T \vec{x_2} = 1$$

and

$$\lambda_1^2 \vec{x_1}^T \vec{y_1} + \lambda_1\lambda_2 (\vec{x_1}^T \vec{y_2} + \vec{x_2}^T \vec{y_1}) + \lambda_2^2 \vec{x_2}^T \vec{y_2} = 0$$

Tracker Alignment

- Assuming $\vec{x_1}^T \vec{y_1} \neq 0$ so that $\lambda_2 \neq 0$ and substituting $s = \lambda_1/\lambda_2$ results in

$$\lambda_2^2(s^2 \vec{x_1}^T \vec{x_1} + 2s\vec{x_1}^T \vec{x_2} + \vec{x_2}^T \vec{x_2}) = 1$$

  and thus two solutions for s
- Choose the s maximizing

$$s^2 \vec{x_1}^T \vec{x_1} + 2s\vec{x_1}^T \vec{x_2} + \vec{x_2}^T \vec{x_2}$$

  and easily obtain the lambas

Tracker Alignment

- Convert dual quaternions to $(R, \vec{t})$
  - non-dual part corresponds to rotation
  - translation by using $\vec{t} = 2q'\bar{q}$

Improving accuracy

- Preselection by applying threshold angle for rotations
$$\theta \leq \alpha \leq (\pi - \theta)$$
- Optimality criterion to be minimized for all motion pairs
$$s_{ij,kl} = |a_{ij}^T a_{kl}|$$
- Using as many stations as possible
- RANSAC approach

Thank you...

... for your attention