A Novel Line Alignment Algorithm Using Geometric Algebra

(Based on Insights from 3D Point Registration)

Name: Haris Matsantonis **Supervisor: Prof. Joan Lasenby**

28th August 2024

UNIVERSITY OF CAMBRIDGE
Agri FoRwArdS
UKRI Engineering and Physical Sciences Research Council

Outline

- Point Cloud / 3D Line Registration
- Why Registration Matters?
- Point Cloud Registration Methods
- Point Cloud Registration Problems That Were Solved Using GA
  - Results using *Clifford Multivector Toolbox (Sangwine & Hitzer)*
- Extension to 3D Line Registration
  - Results using *Clifford Multivector Toolbox (Sangwine & Hitzer)*

- Sangwine, S. J., & Hitzer, E. (2017). Clifford Multivector Toolbox (for MATLAB). *Advances in Applied Clifford Algebras*, *27*(1), 539-558

Introduction

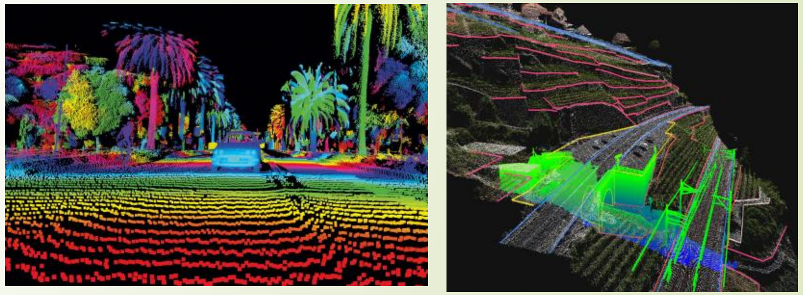
- **Research Focus:** Point Cloud Registration + Line-based 3D Registration
- **Sources:** RGB-D cameras + LiDAR Sensors, Photogrammetry

1. Depth map
2. Camera Calibration
3. Superimposition

Point Cloud

1. Emits laser pulses
2. Measures return time
3. Creates 3D maps

Point Cloud, Source: Geospatial World

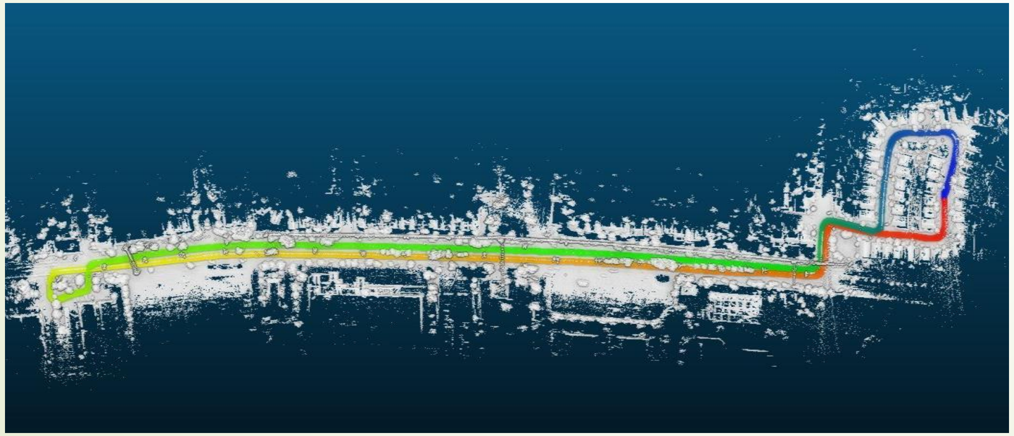


LiDAR Technology, Photogrammetry. Source: PIX4D

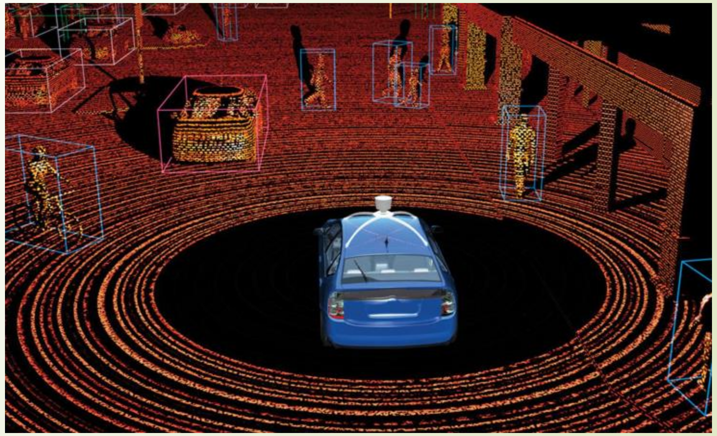
- **Goal:** Find a transformation which consists of a *rotation + translation (rigid registration)* to align two point clouds and two sets of 3D lines

Why do we care about ‘*Registration*’?

- High precision sensors (LiDAR, Kinect, etc) → Point Clouds **+** Lines = ***primary data format***
- Sensors can only capture scans within their limited view range
- Applications include (among others):
  1. 3D Reconstruction
  2. Autonomous Vehicle Localization
  3. 3D Terrain Mapping



3D Terrain Mapping. Source: Unmanned Systems Lab, TAMU



Visualization of a LiDAR point cloud. Source: Graham Murdock for Popular Science

Point Cloud Registration Methods

Aspect	Known Correspondences	Unknown Correspondences
	Predefined	Not predefined
Typical Method	Singular Value Decomposition (SVD)	Iterative Closest Point (ICP)
Key Approach	Direct (1 Iteration) computation using known pairs	Iteratively estimate and refine using nearest points
Applications	Medical imaging, Precise engineering	Robotics, Autonomous navigation

Points can miss critical linear features, leading to less accurate models.
What about Lines?

Broadening Horizons in Line Registration

GAPS

- **Over-Reliance on Traditional Methods:** Most (*point + line*) registration algorithms are built on Vector Algebra, leading to a narrow approach
- **Lack of Exploration:** Few comprehensive studies investigate alternative mathematical frameworks for (*point + line*) registration

AIM

- **Innovate with Geometric Algebra:** Develop a novel line registration framework using Geometric Algebra
- **Benchmark Against Standard Methods:** Assess effectiveness against Linear Algebra benchmarks (*SVD point solution*)

Inspiration for Extending from Point to Line Registration

Geometric Algebra Solutions for 3D Registration Problems

Problem 1

3D Point Registration with Known Correspondences

Problem 2

3D Point Registration with Unknown Correspondences

- *3D Euclidean Space Solution*
- *4D Spherical Space Solution*

Problem 3

?

→ *3D Line Registration?*

Recap: 3D Point Registration using Characteristic Multivectors

Let $\{a_k\}$, $k = 1, 2, \cdots m$, be a frame for the space, and $\{a^k\}$ be its reciprocal frame → reciprocal basis set of vectors guarantees orthonormality conditions

- Define $\{b_k\} = f(a_k)$. Then the ***r**^th^ **Characteristic Multivector*** is:

$$\theta_{(r)} f_{(r)} = \sum (a^{j_r} \wedge \cdots \wedge a^{j_1})(b_{j_1} \wedge \cdots \wedge b_{j_r})$$

the sum over the repeated indices is restricted by $0 < j_1 < \cdots < j_r \leq m$

- Then, we can recover the Rotor as (Lasenby et al. 2022):

$$\tilde{R} \propto 1 + [a^1 b_1 + a^2 b_2 + \cdots] + [(a^2 \wedge a^1)(b_1 \wedge b_2) + \cdots] +$$
$$[(a^3 \wedge a^2 \wedge a^1)(b_1 \wedge b_2 \wedge b_3) + \cdots] + \text{higher order terms}$$

- WORKS for *any dimension and signature*

- Lasenby A., Lasenby J., Matsantonis C. (2022). Reconstructing a Rotor from Initial and Final Frames using Characteristic Multivectors: with applications in Orthogonal Transformations. *Mathematical Methods in the Applied Sciences*, 1-18

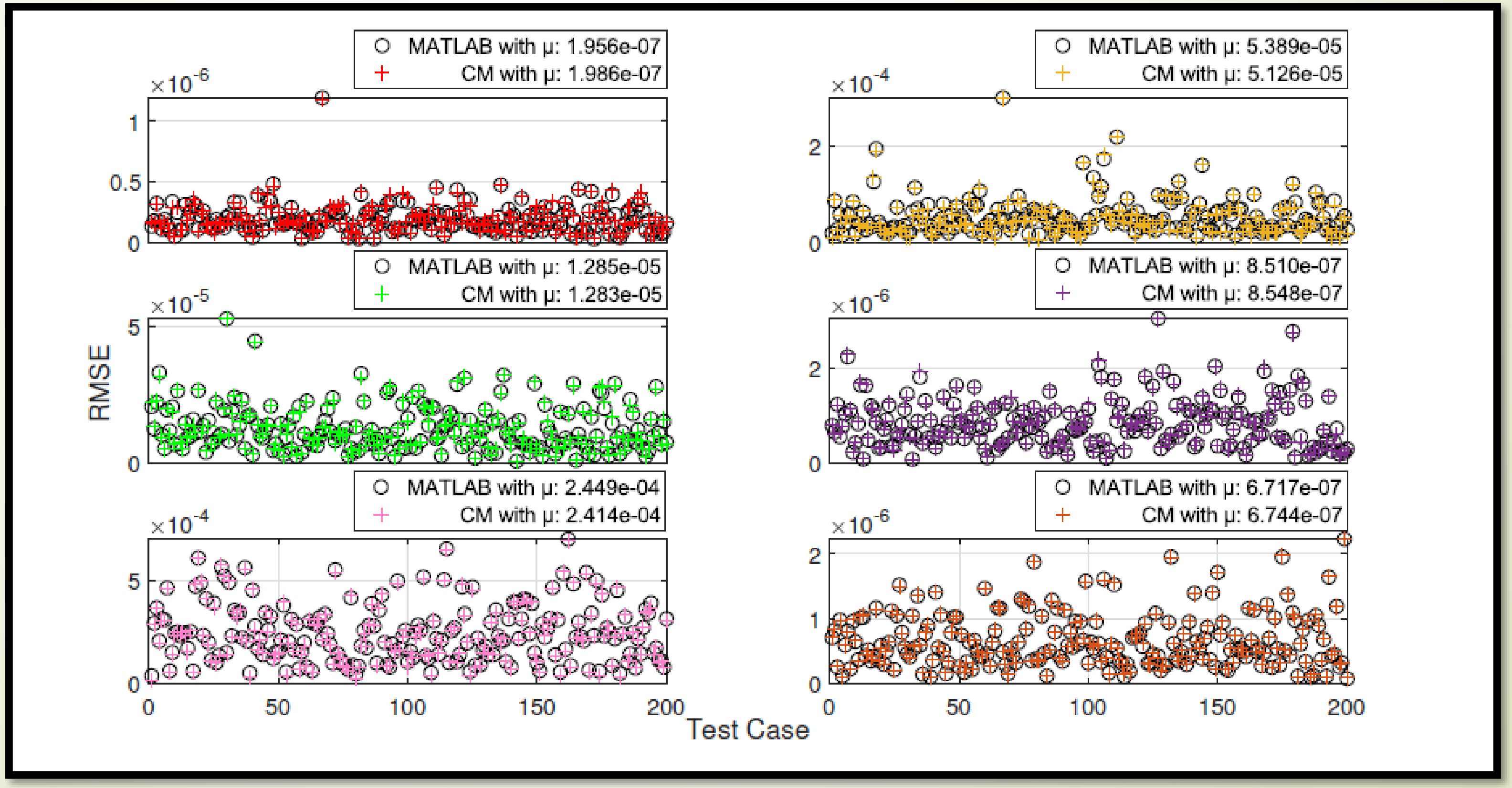
Problem 1: 3D Point Registration with Known Correspondences

Point Cloud	No. Points
Apple	867
Helix	3240
Tree	4806
Horse	8727
Armadillo	17297
Buddha Statue	31222

Point Clouds do no carry colour information – colours are added for better visualization purposes

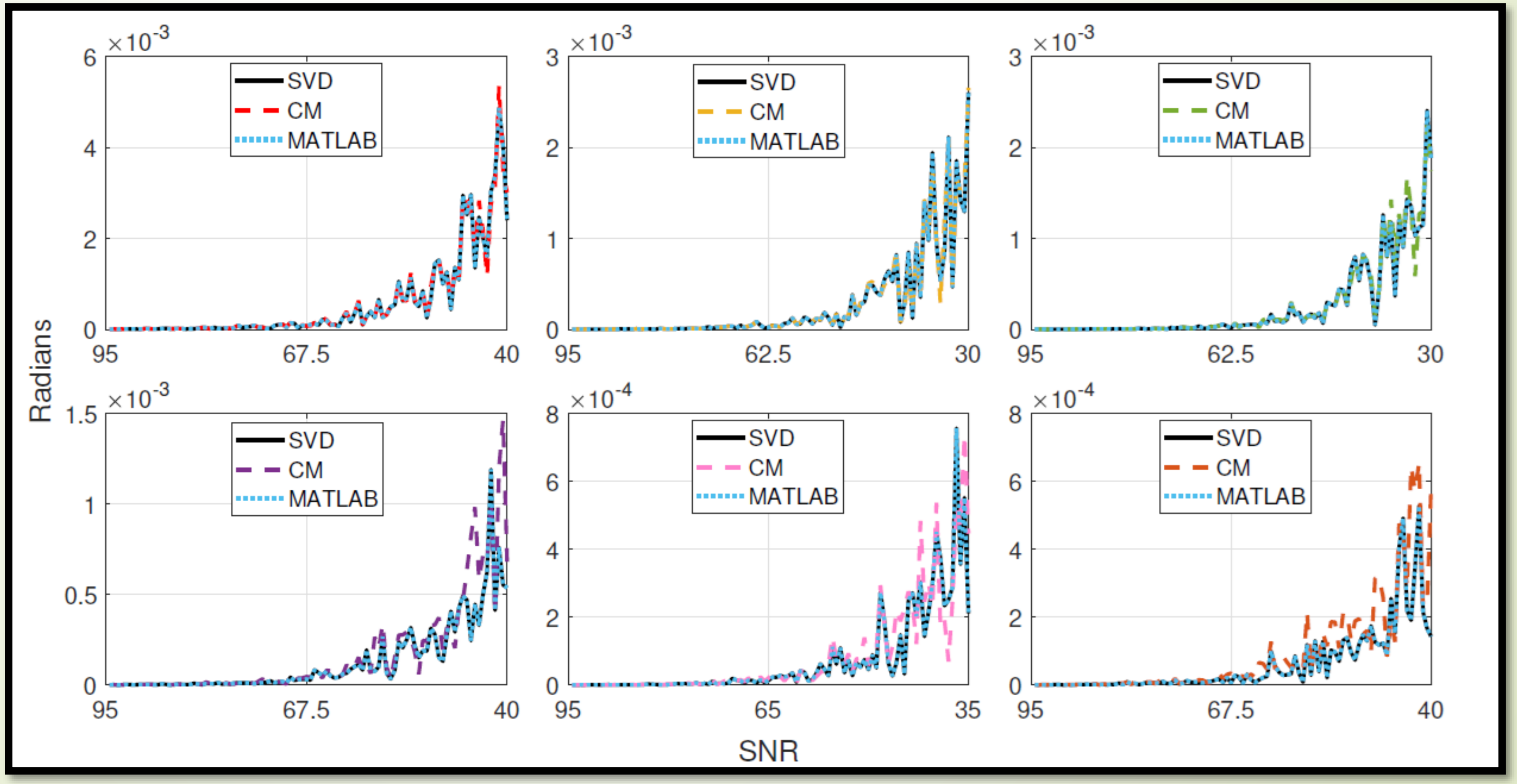
Problem 1 Results: No Noise

- RMSE and the Mean (μ) of MATLAB's built-in function and 3D Characteristic Multivector method for **200 random initial positions**: Apple (+), Helix (+), Tree (+), Horse (+), Armadillo (+), Buddha Statue (+)



Problem 1 Results: Noise (SNR)

- Rotational Convergence Error (plots share same x-axis) of MATLAB's built-in function, 3D Characteristic Multivector and self-coded SVD: Apple (--), Helix (--), Tree (--), Horse (--), Armadillo (--), Buddha Statue (--)



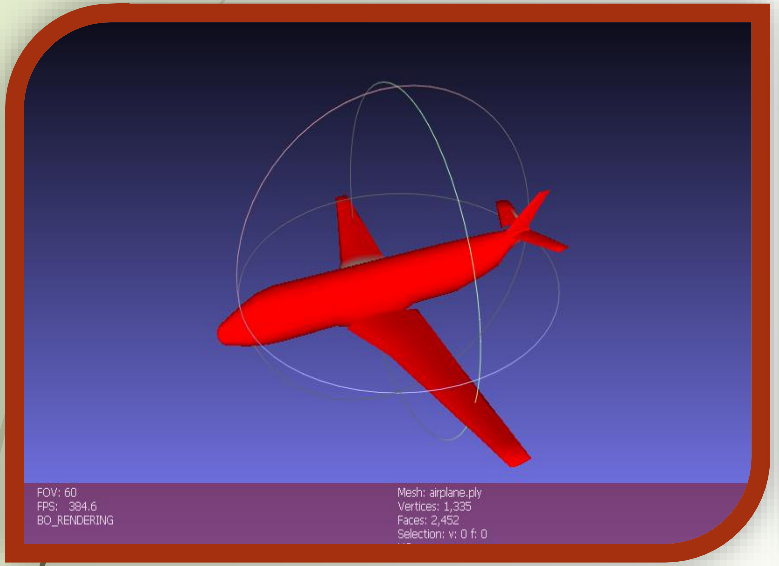
Problem 2a: 3D Registration Problem with Unknown Correspondences (3D Solution)

Standard ICP Algorithm [Besl & McKay 92]

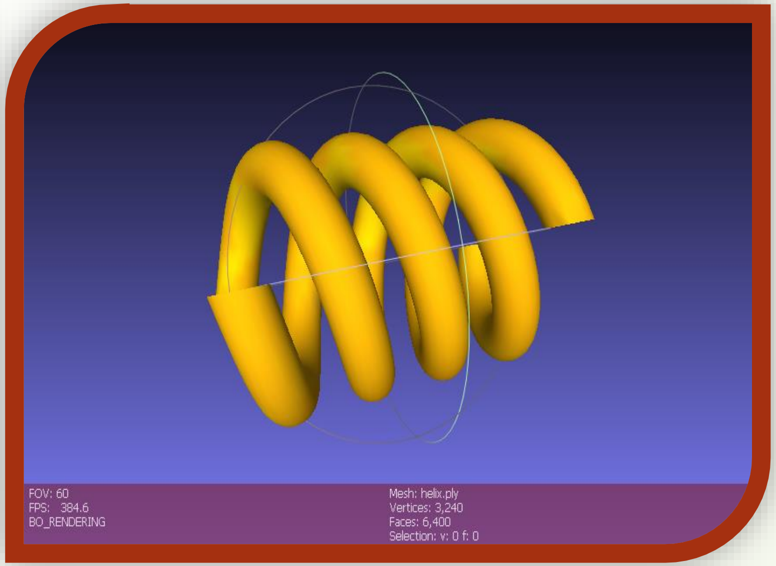
1. Determine corresponding points
2. Compute the Rotation **R** via SVD
3. Compute the translation **t** based on **R**
4. Apply **R** and **t** to the points of the point cloud to be registered
5. Compute the error E(**R**,**t**)
6. If error decreased and error > threshold
   - Repeat these steps
   - Stop and output final alignment, otherwise

Key Note: The GA-ICP algorithm effectively follows the above calculation steps but **replaces step 2** with a **rotor-calculation** step, i.e. ***3D Characteristic Multivector***

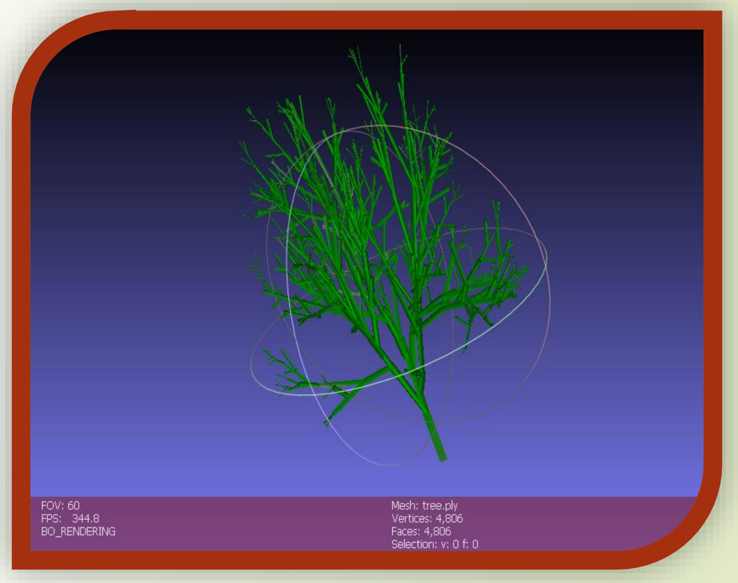
Problem 2a : Point Cloud Data



AIRPLANE (1335 Points)



HELIX (3240 Points)



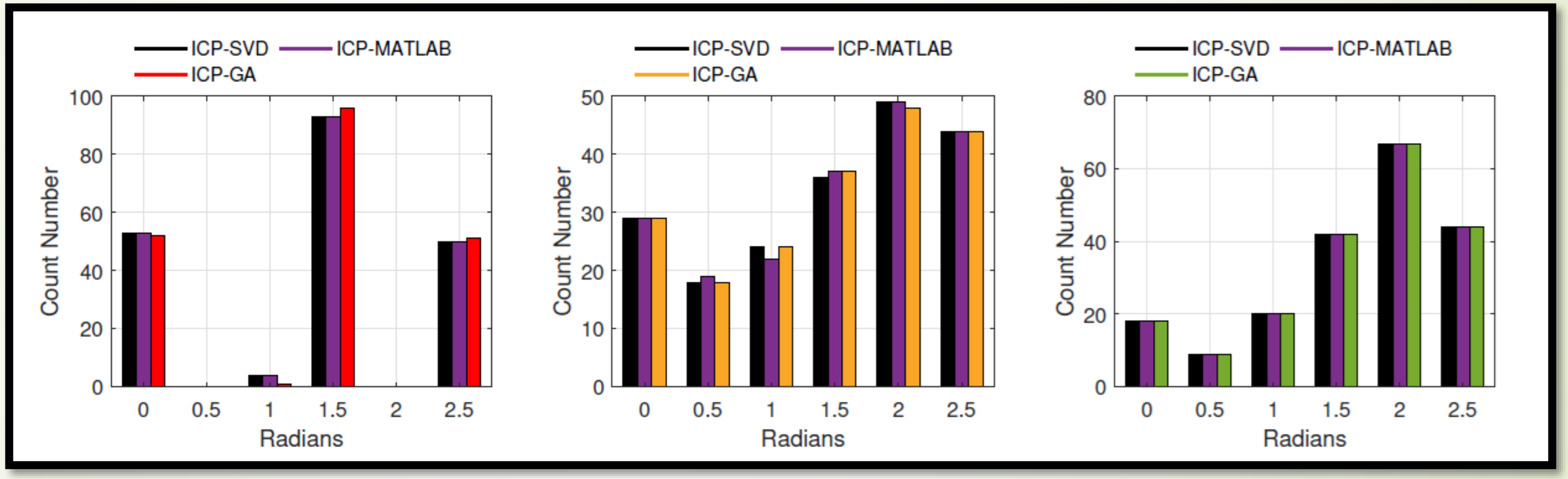
TREE (4806 Points)

Point Clouds do no carry colour information – colours are added for better visualization purposes

Problem 2a Results: No Noise / Noise

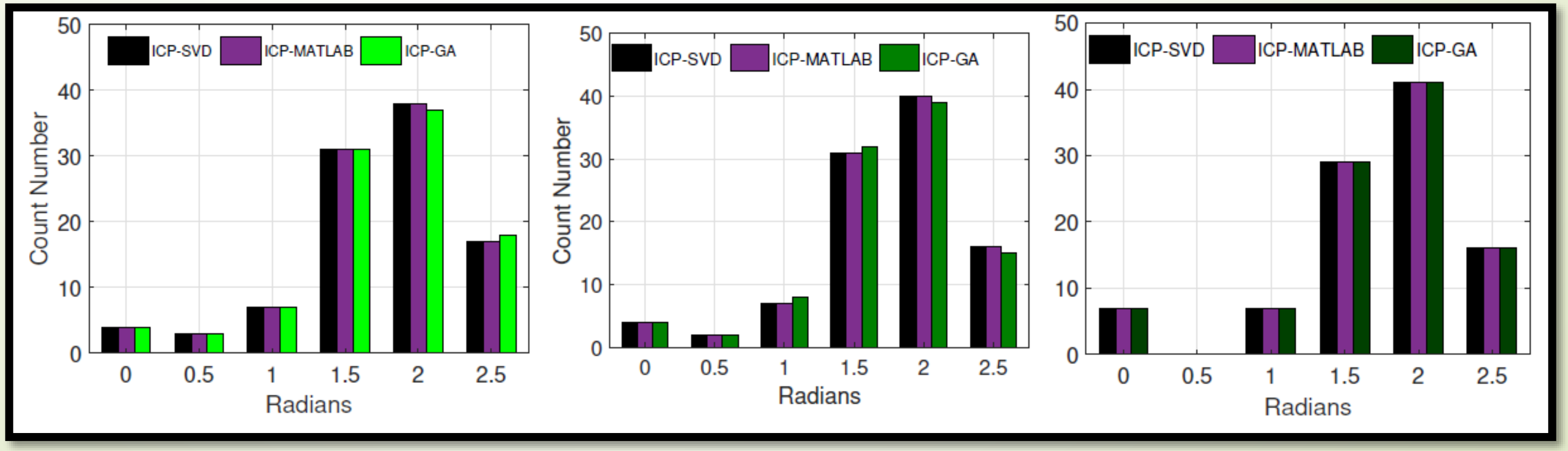
- Comparison with MATLAB's ***pointToPoint (built-in ICP function)*** and ***self-coded ICP-SVD***

No Noise
Rotational Convergence Error



- Airplane
- Helix
- Tree

Noise
Rotational Convergence Error



Tree

- $\sigma = 0.0001$
- $\sigma = 0.0008$
- $\sigma = 0.0015$

Problem 2b: 3D Point Registration Problem with Unknown Correspondences (4D Solution)

Standard ICP Algorithm [Besl & McKay 92]

1. Determine corresponding points
2. Compute the Rotation **R** via SVD
3. Compute the translation **t** based on **R**
4. Apply **R** and **t** to the points of the point cloud to be registered
5. Compute the error E(**R**,**t**)
6. If error decreased and error > threshold
   - Repeat these steps or otherwise stop and output final alignment

? **Key Note:** The 4D algorithm effectively follows the above calculation steps but **replaces steps 2 and 3** with a **4D rotor (motor)-calculation** step. Calculations are done in **4D Spherical Space** instead of **3D Euclidean Space** ?

Problem 2b: The 1D-Up Approach in Conformal Geometric Algebra (CGA)

- Euclidean, Hyperbolic, Spherical and Inversive geometries are put on the same footing
- **Null Vectors:** Construct new null vectors representing key points in space

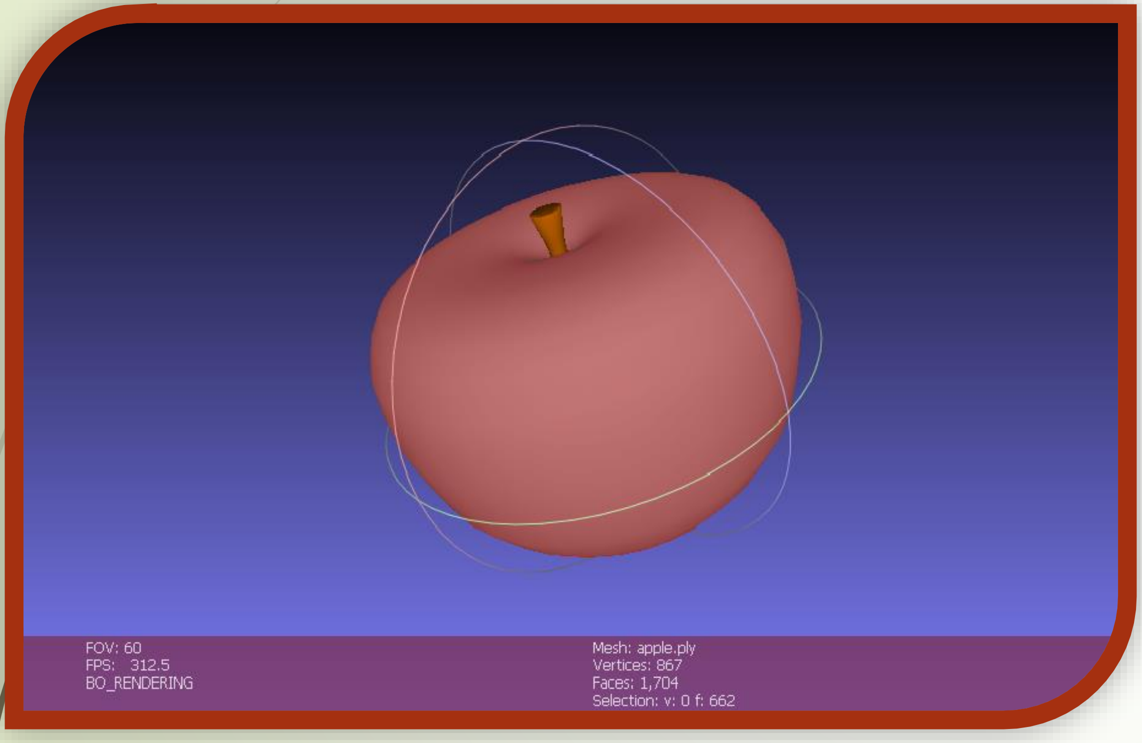
$$n = e + \bar{e}, \quad \bar{n} = e - \bar{e}$$

- The rotors that keep $\bar{e}$ invariant define a spherical space (4D)

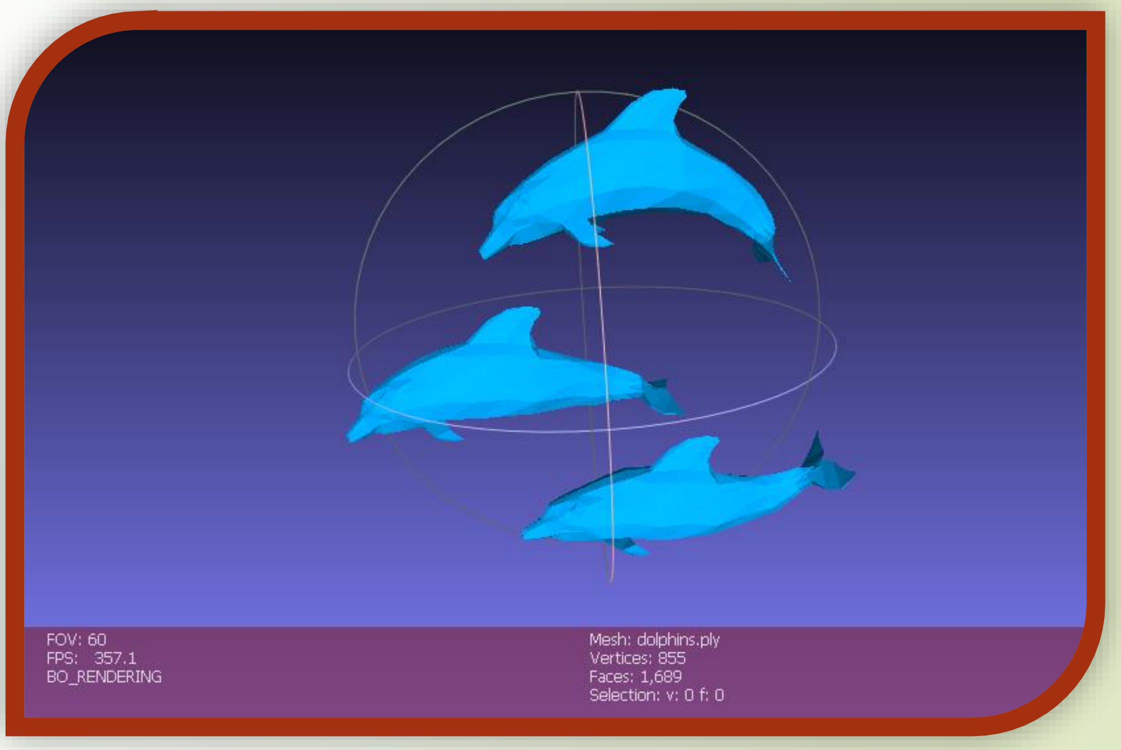
- **Benefits:**
  - *Reduced Complexity:* Working on a space with Euclidean signature, simplifies operations and improves computational efficiency.
  - *Applications:* Useful in geometric object matching, line fitting, and quantum mechanics.

- Lasenby, A. (2011). Rigid body dynamics in a constant curvature space and the '1D-up' approach to conformal geometric algebra. *Guide to geometric algebra in practice*, 371-389.

Problem 2b: Point Clouds



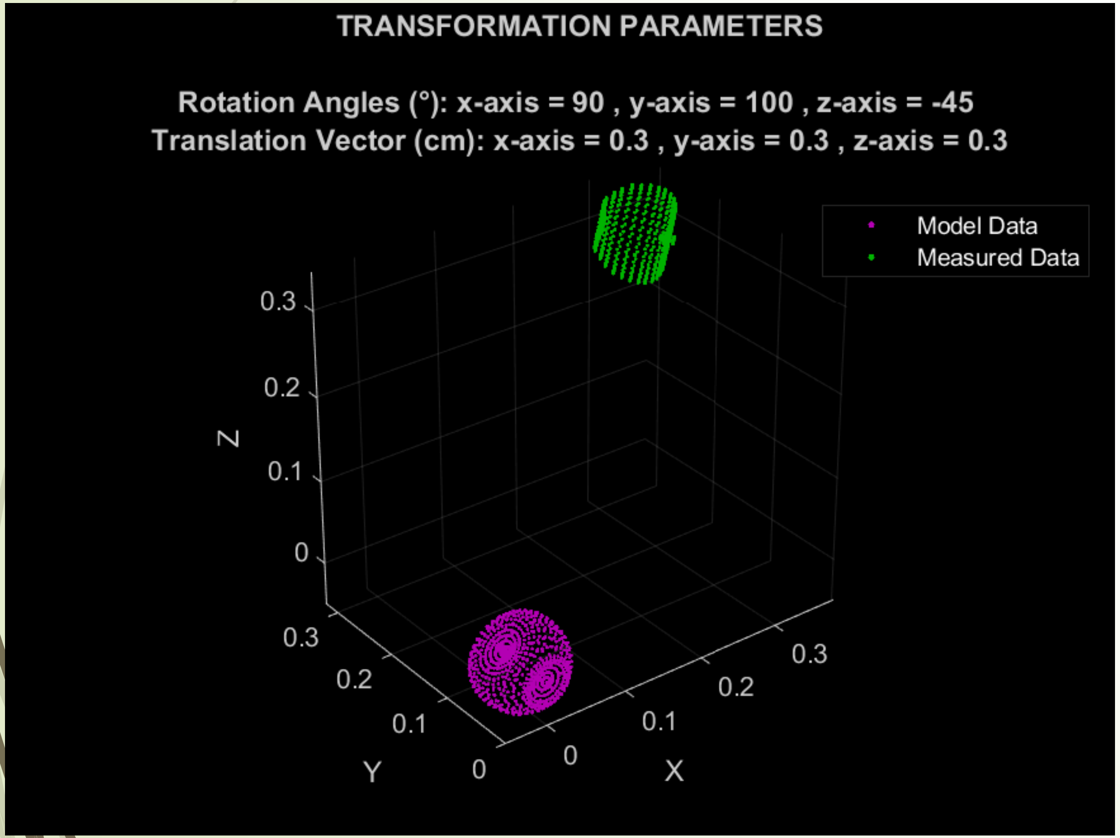
APPLE (Continuous)

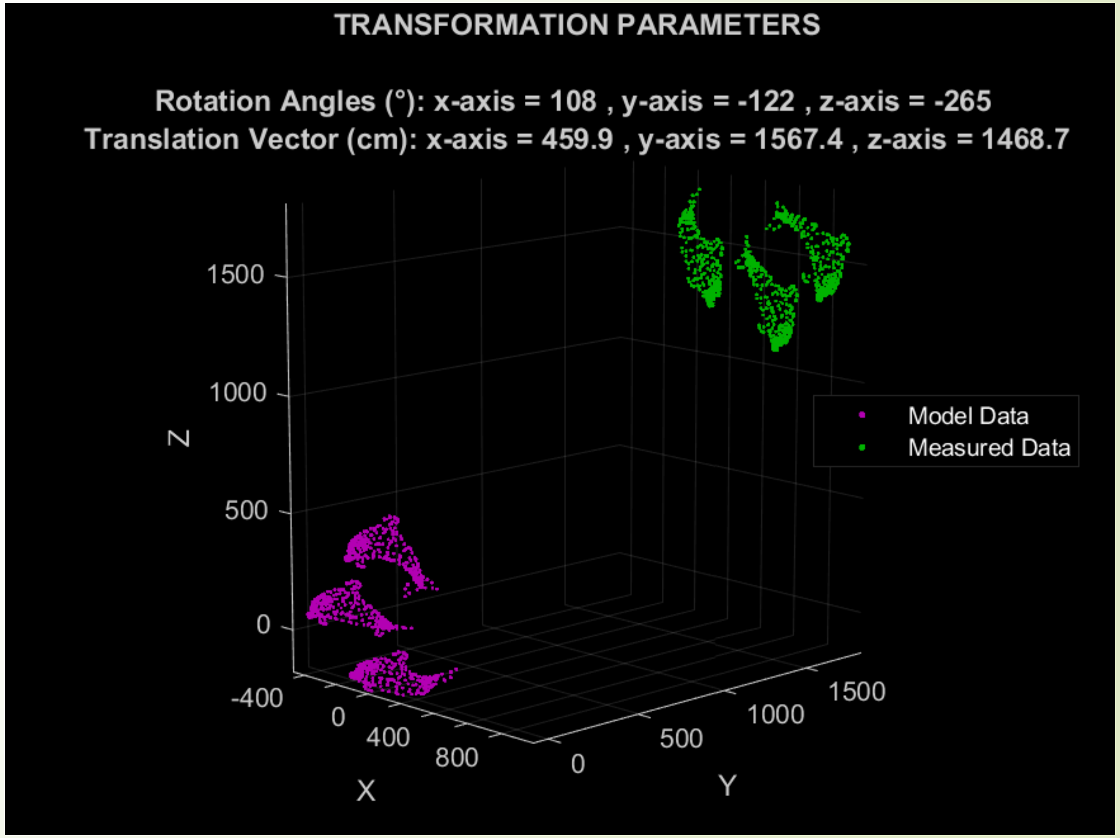


DOLPHINS (Non - Continuous)

Point Clouds do no carry colour information – colours are added for better visualization purposes

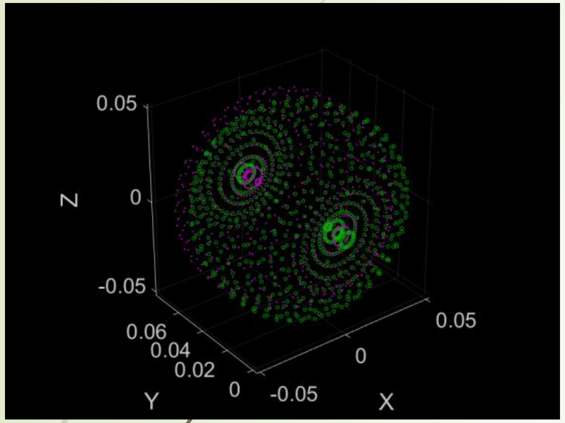
Problem 2b - Hard Transformations

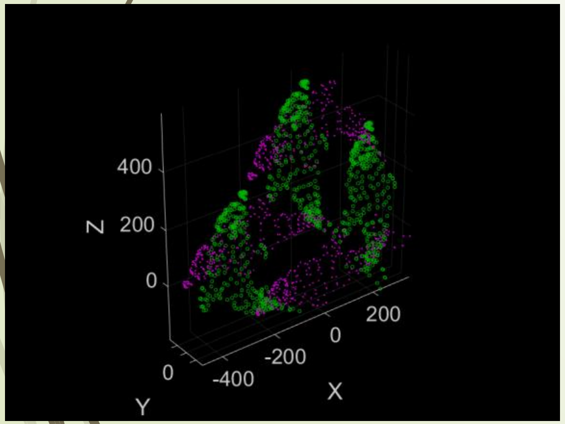




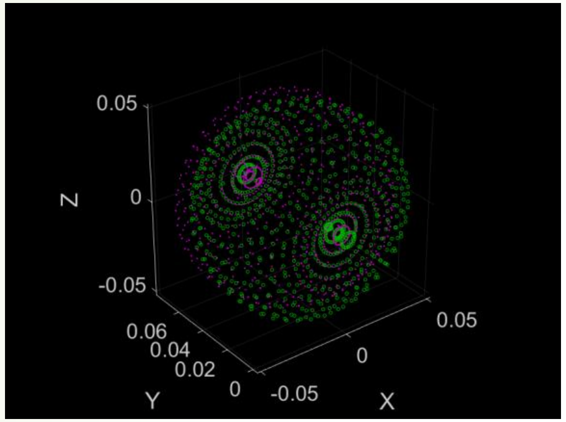
Problem 2b - Hard Transformations Results

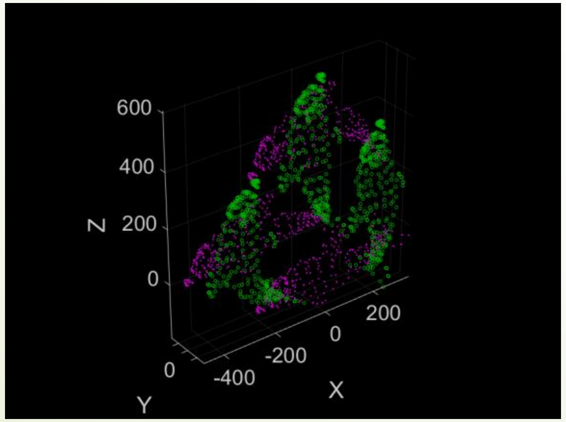
pointToPoint ICP
(MATLAB built-in function)



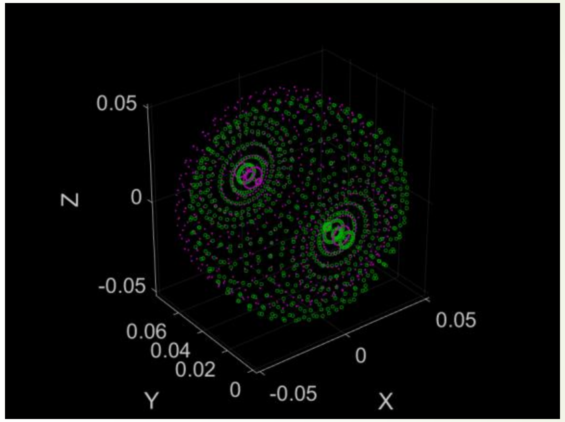


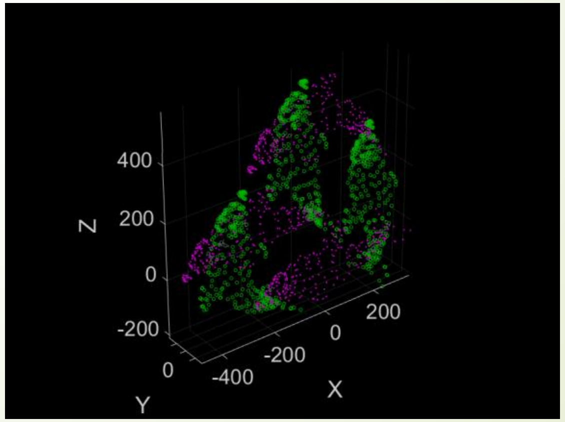
pointToPlane ICP
(MATLAB built-in function)



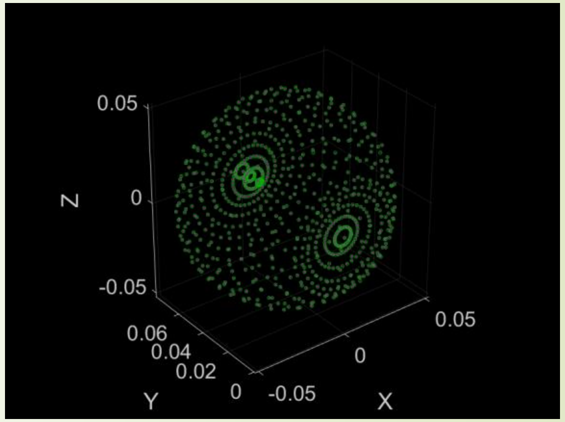


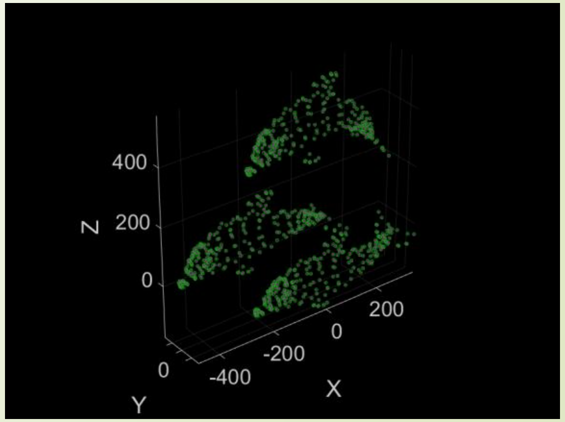
planeToPlane ICP
(MATLAB built-in function)





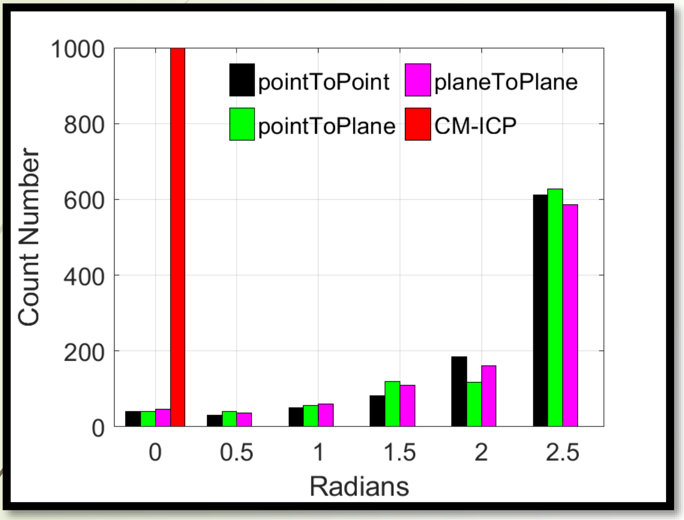
4D Spherical CM
(CM-ICP)



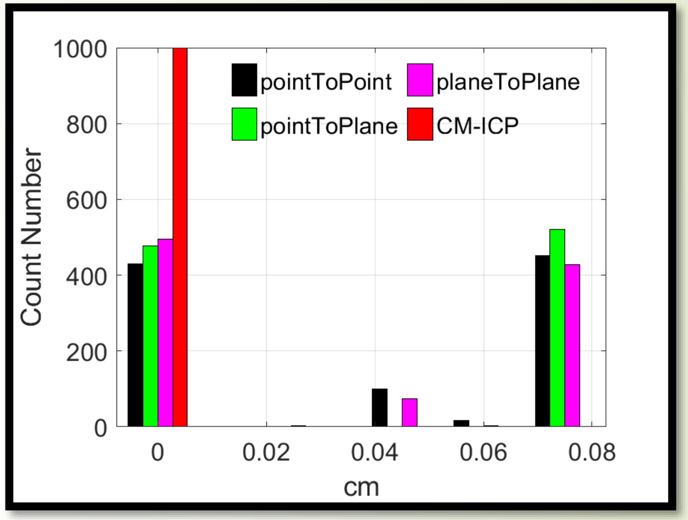


Problem 2b: 1000 Random Positions

Apple: Rotation Error

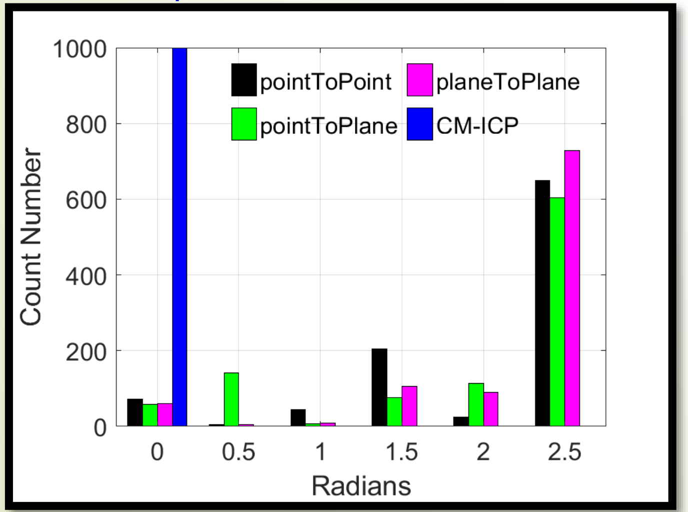


Apple: Translation Error

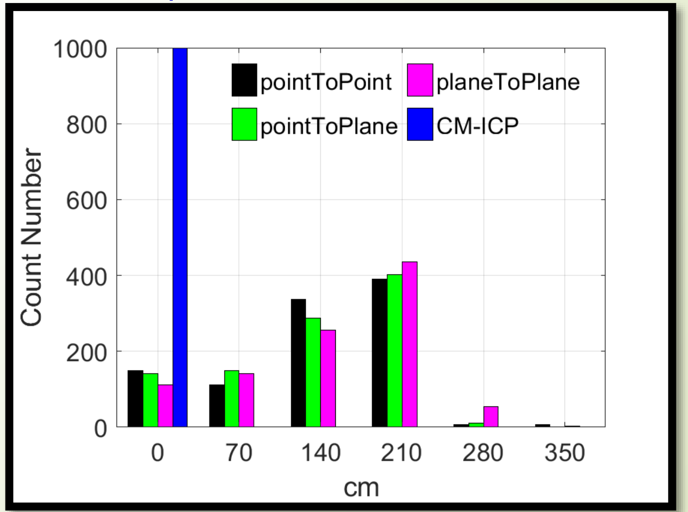


$$R_{apple} \in [-360°, 360°]$$
$$t_{apple} \in [0, 1]$$

Dolphins: Rotation Error

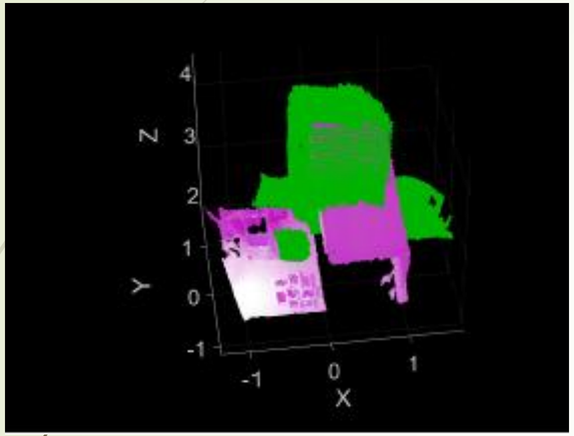


Dolphins: Translation Error

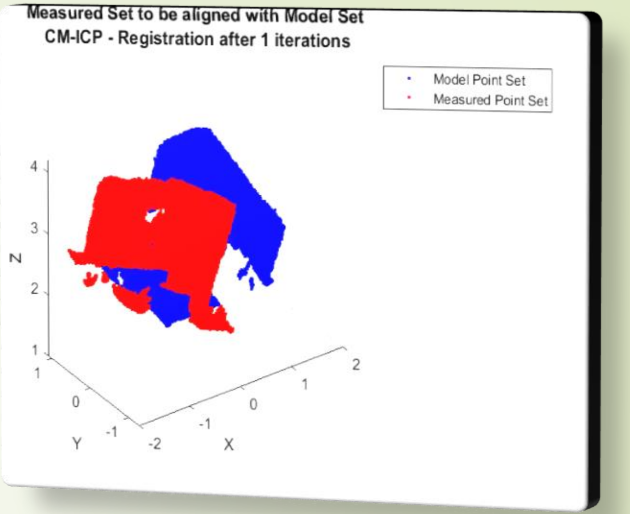


$$R_{dolphins} \in [-360°, 360°]$$
$$t_{dolphins} \in [0, 1500]$$

Problem 2b: Real Data (Room)

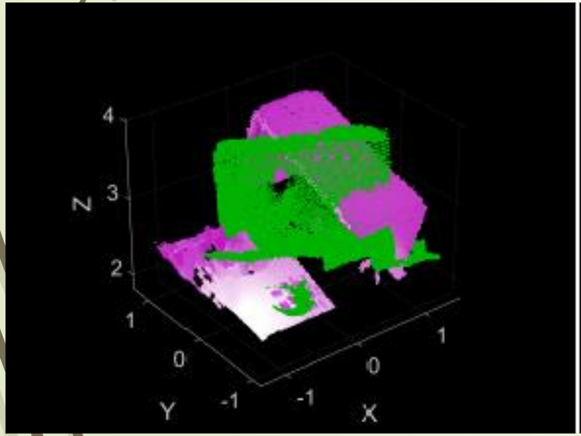


- Model Point Cloud (235306 points)
- Measured Point Cloud (236483 points)

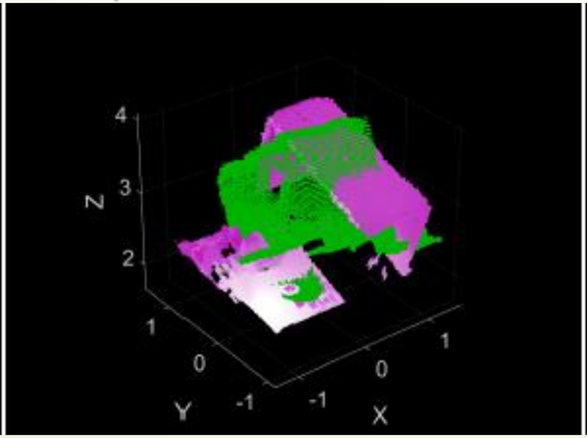


Results

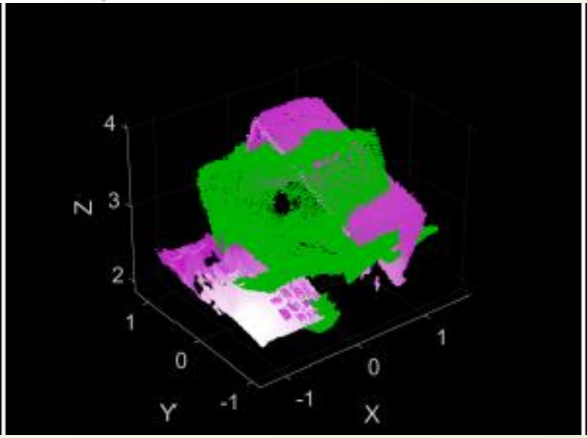
pointToPoint ICP



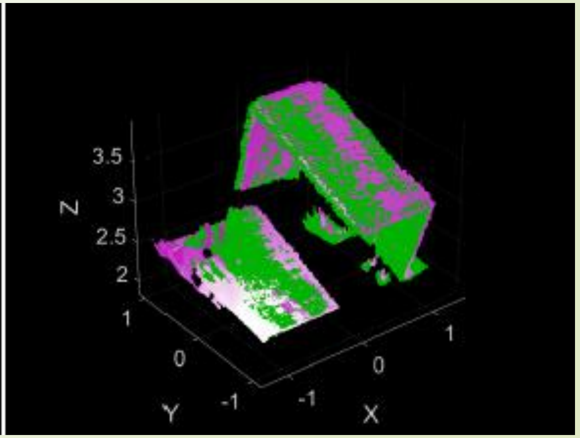
pointToPlane ICP



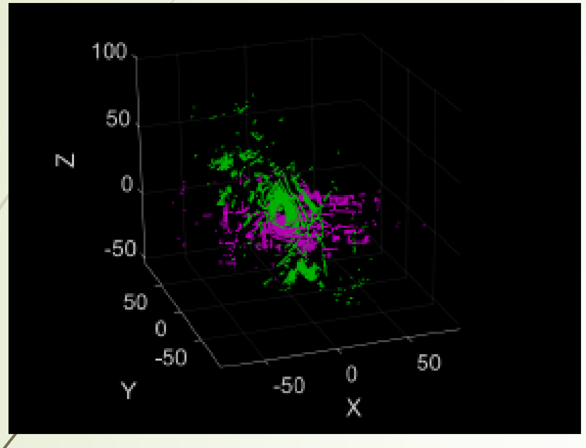
planeToPlane ICP



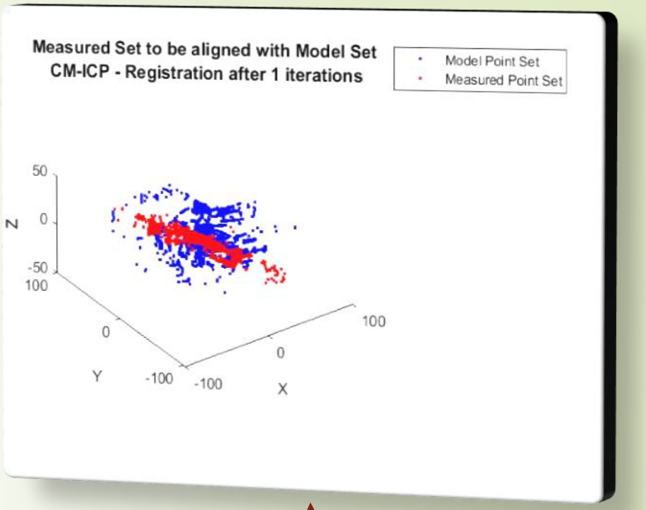
CM-ICP



Problem 2b: Real Data (Construction Road)

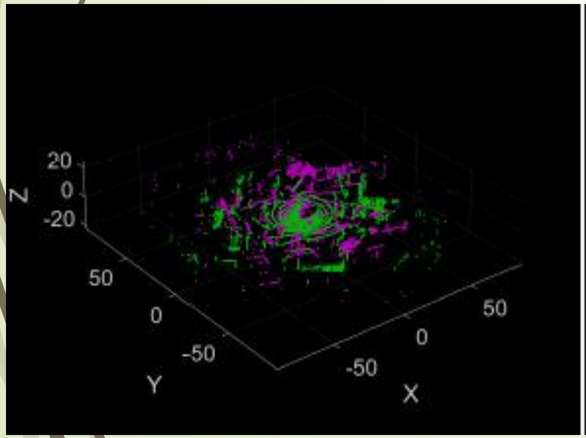


- Model Point Cloud (17851 points)
- Measured Point Cloud (17841 points)

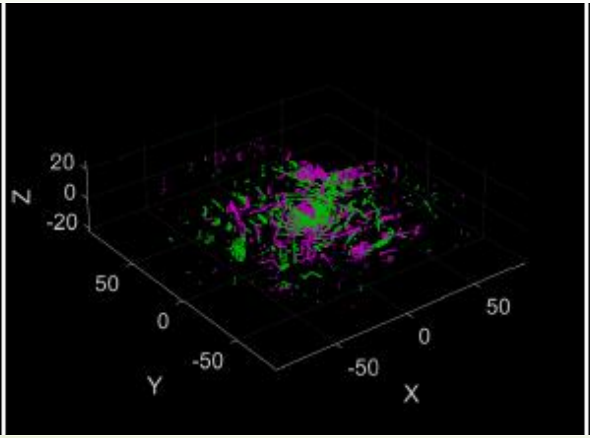


Results

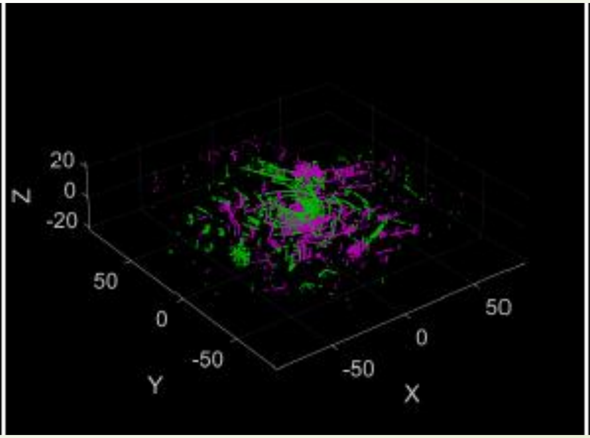
pointToPoint ICP



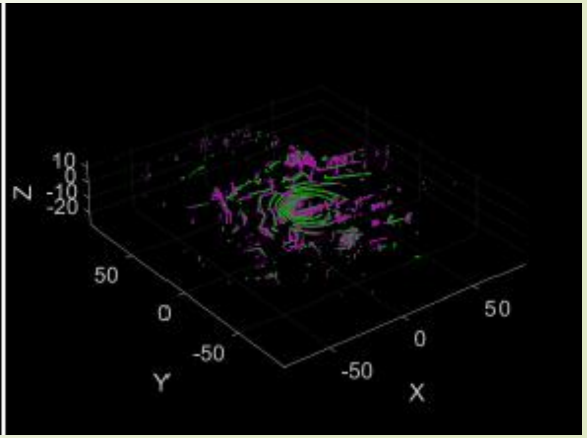
pointToPlane ICP



planeToPlane ICP



CM-ICP



Results So Far

Problem	Linear Algebra	Geometric Algebra	Result
3D Registration with Known Correspondences	SVD method	Closed-form, non-iterative solution	**Tie**
3D Registration with Unknown Correspondences (3D Solution)	Standard ICP and its variants	GA-ICP maintained accuracy of standard ICP	**Tie**
3D Registration with Unknown Correspondences (4D Solution 1DUp)	Standard ICP and its variants	CM-ICP outperformed standard ICP and its variants	**GA better**
3D Line Registration with Known Correspondences	**No well-established algorithms**	*(4D Solution 1DUp)*	**?**

Problem 3: 3D Line Registration with Known Correspondences

Aligning sets of 3D Lines is a complex task with no well established algorithms

Develop a new 3D Line registration method

Linear Algebra

Geometric Algebra

Problem 3: How? Methodology - Initial Setup

- For a point $p(x, y, z)$ in 3D Euclidean space, its equivalent in 4D Spherical space with curvature λ is defined by the transformation:

$$P = \left(\frac{2\lambda}{\lambda^2 + p^2}\right) p + \left(\frac{\lambda^2 - p^2}{\lambda^2 + p^2}\right) e_4, \quad P^2 = 1, \quad (e_4)^2 = 1$$

- Map the starting p_{start} and end points p_{end} of a line to 4D Spherical Space and wedge them together to construct a line L:

$$L = qe_{12} + re_{13} + se_{14} + te_{23} + ue_{24} + ve_{34}, \text{ with}$$

q, r, s, t, u, v *being scalar not independent coefficients*

- A line L can undergo transformation via a motor M:

$$L' = ML\tilde{M} = q'e_{12} + r'e_{13} + s'e_{14} + t'e_{23} + u'e_{24} + v'e_{34}$$

q', r', s', t', u', v' *being transformed scalar coefficients*

Problem 3 Methodology: From Bivectors to Vectors

- To start suppose that we have a set of 6 lines $\{L_i\}, i = 1, \ldots, 6$ which are rotated and translated via a motor M to $\{L_i'\}, i = 1, \ldots, 6$ s.t:

$$L_i^{'} = M L_i \tilde{M}$$

- The bivectors representing both $\{L_i\}$ and $\{L_i'\}$ can be treated as vector components in a 6D Euclidean space with Euclidean basis:

$$\{f_1, f_2, f_3, f_4, f_5, f_6\}$$

- The mapping of the original lines (bivectors) to vectors for the original and transformed lines is given by:

$$qe_{12} + re_{13} + se_{14} + te_{23} + ue_{24} + ve_{34} \longrightarrow qf_1 + rf_2 + sf_3 + tf_4 + uf_5 + vf_6$$

$$q'e_{12} + r'e_{13} + s'e_{14} + t'e_{23} + u'e_{24} + v'e_{34} \longrightarrow q'f_1 + r'f_2 + s'f_3 + t'f_4 + u'f_5 + v'f_6$$

- Call these vector representations v_i and v_i', and form the reciprocal vectors v^i and $v^{i'}$

Problem 3 Methodology: From Vectors Back to Bivectors

- Map the reciprocal vectors v^i and $v^{i'}$ back into bivectors within the 4D Spherical space that reverses the initial transformation

Details will be given in the paper

- If v^i is represented as $[q, r, s, t, u, v]$, L^i is constructed as:

$$qe_{12} + re_{13} + se_{14} + te_{23} + ue_{24} + ve_{34}$$

- So now we have $\{L_i\}, \{L_i'\}, \{L^i\}, \{L^{i'}\}, i = 1, \ldots, 6$
- Consider now reciprocal sets of line $\{L^i\}$ and $\{L^{i'}\}$ s.t:

$$L_i \cdot L^j = \delta_{ij}, \quad L_i' \cdot L^{j'} = \delta_{ij}$$

- Then it is easy to say that reciprocals transform correspondingly as:

$$L^{i'} = ML^i\tilde{M}$$

Methodology: Recovering the Motor

- The motor M is made up of scalar, bivector and quadvector (pseudoscalar, $I = e_1e_2e_3e_4$) parts which we can write as:

$$M = \alpha + B + \beta I$$

- It then follows that (*details will be given in the paper as it is not trivial*):

$$L_i M L^i = 6\alpha + 6\beta I - 2B = -2M + 8(\alpha + \beta I)$$

- Hence:

$$L_i M L^i \tilde{M} = L_i L^{i'} = -2 + 8(\alpha + \beta I)\tilde{M}$$

- We can derive a quantity as a key variable capturing the aggregate transformation effect:

$$X = \frac{1}{8}\left(\sum_i^6 (L_i \cdot L^{i'}) + 2\right)$$

Problem 3 Methodology: Recovering the Motor (cont.)

- We can multiply X by $\tilde{X}$ to form:

$$\begin{aligned} Y = X\tilde{X} &= M(\alpha + \beta I)(\alpha + \beta \tilde{I})\tilde{M} \\ &= M(\alpha^2 + \beta^2 + 2\alpha\beta I)\tilde{M} \\ &= \underbrace{\alpha^2 + \beta^2}_{u} + \underbrace{2\alpha\beta}_{v} I \end{aligned}$$

- Note that: $(\alpha + \beta)^2 = u + v \Rightarrow \alpha + \beta = \pm\sqrt{u + v}$
- Setting $w = \alpha + \beta$, we get $w = \pm\sqrt{u + v}$
- Then starting with the relations $u = \alpha^2 + \beta^2, v = 2\alpha\beta$ and rearranging:

$$\alpha^2 + \beta^2 - u = 0$$

- Since $\frac{v}{2} = \alpha\beta$, and setting $w = \alpha + \beta$ gives a quadratic equation:

$$\beta^2 - w\beta + v/2 = 0$$

Problem 3 Motor Solution

- Solving the quadratic equation and considering both (+, -) values of w, we have 4 distinct solutions (a set of two solutions for each sign of w)

- These solutions lead to four unique pairs of (a, b)

- After some algebra it can be shown that the motor is:

$$
\begin{aligned}
M &= \tfrac{1}{u}(a_2 - b_2 I)X, \quad \text{or} \\
M &= \tfrac{1}{u}(b_1 - a_1 I)X
\end{aligned}
$$

Note that we started with 6 lines, but this can also be extended to any number of lines

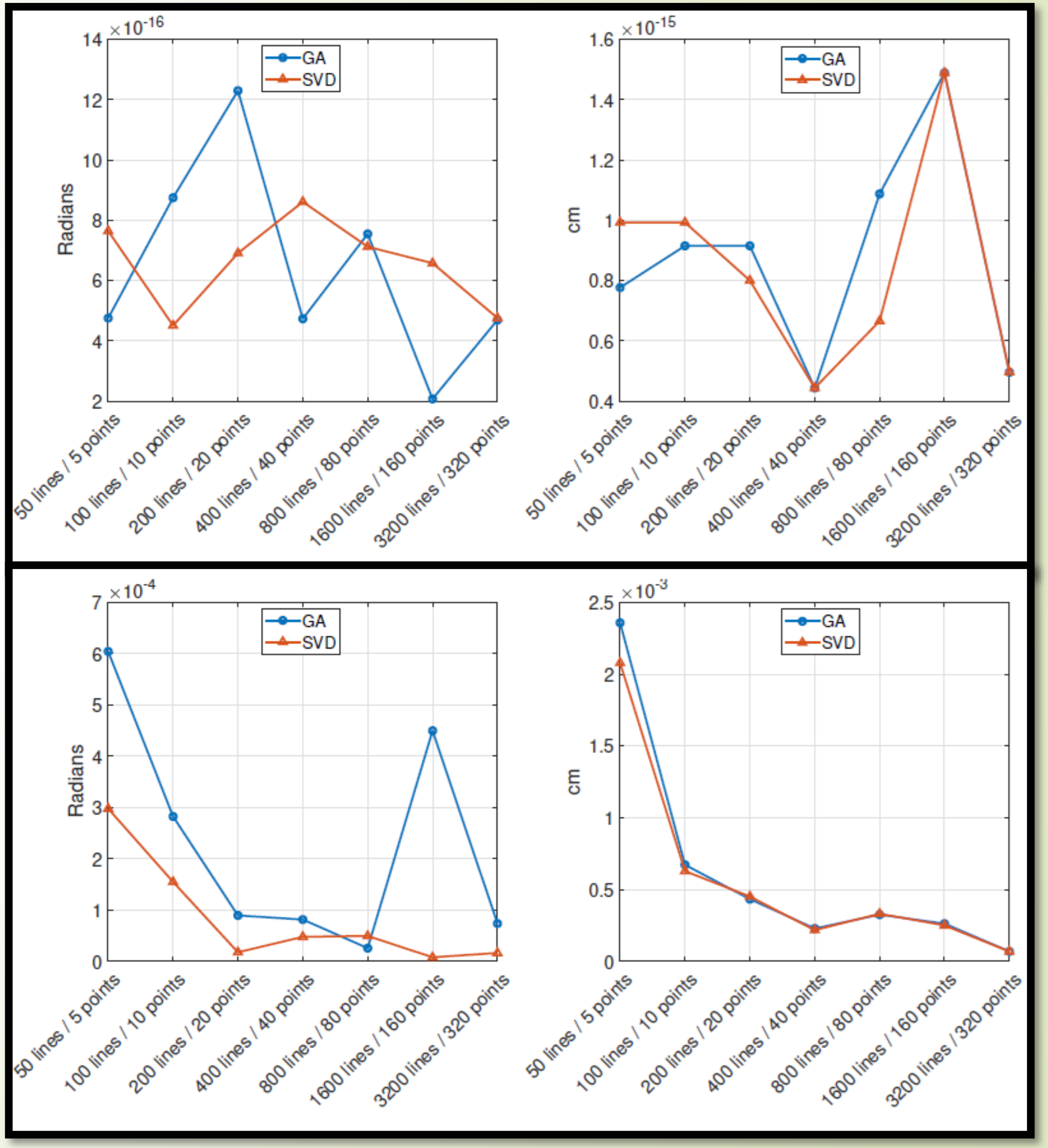
Problem 3: Setup Details

- **Lines and Points**: *Different numbers* of lines and points used in experiments
- **Transformation:** Rotate (x: -10°, y: -20°, z: -30°) and Translate (x: 1, y: 2, z: 3) cm
- **Scenarios:**
  - *No Noise*: Baseline performance measurement
  - *With Gaussian Noise ($\sigma = 0.01$)*: Added noise to test robustness

Comparison made with SVD point solution for benchmarking

Problem 3: Results

- Rotation Error:
  - **No Noise (Top Left)**: Both GA and SVD show very low errors ~10^{-16} radians
  - **With Noise (Bottom Left)**: GA shows higher initial error but improves with more data, SVD is consistent
- Translation Error:
  - **No Noise (Top Right)**: Errors around ~10^{-15} cm
  - **With Noise (Bottom Right)**: Converges to around 10^{-3}cm with more data for both methods.



Problem 3: Line Structures in Focus and Practical Considerations

Thank you for listening!

cm2128@cam.ac.uk

Any questions or comments?

- Lasenby, A., Lasenby, J., & *Matsantonis, C*. (*2022*). "Reconstructing a Rotor from Initial and Final Frames using Characteristic Multivectors: with applications in Orthogonal Transformations". **Mathematical Methods in the Applied Sciences**. 1-18
- *Matsantonis, C*. and Lasenby, J. (*2023*). "A Geometric Algebra Solution to the 3D Registration Problem". In: Iida, F., Maiolino, P., Abdulali, A., Wang, M. (eds) **Towards Autonomous Robotic Systems. TAROS 2023**. *Lecture Notes in Computer Science*, vol 14136. Springer, Cham
- *Matsantonis, C*. and Lasenby, J. (*2024*). "A Geometric Algebra Solution to the Absolute Orientation Problem". In: Araujo Da Silva, D.W.H., Hildenbrand, D., Hitzer, E. (eds) **Advanced Computational Applications of Geometric Algebra**. ICACGA 2022. *Springer Proceedings in Mathematics & Statistics*, vol 445. Springer, Cham
- *Matsantonis, C*. and Lasenby, J. (*2024*). "Characteristic Multivector Fitting of Two 3D Point Sets". (submitted to **IEEE Transactions on Pattern Analysis and Machine Intelligence**)
- *Matsantonis, C*., Lasenby, J., & Lasenby, A. (2024). "A Novel Line Alignment Algorithm using Geometric Algebra". In preparation for **Mathematical Methods in the Applied Sciences**