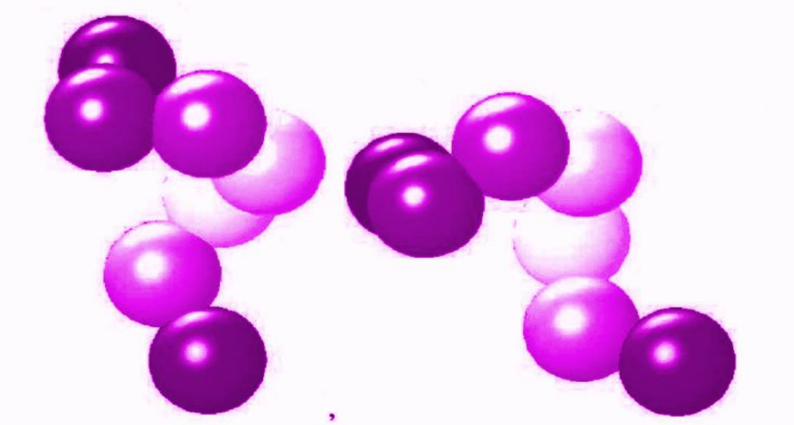
Yaw,Pitch and Roll

By: Chrismine and Anchal

Introduction

This project explores a fascinating and essential problem in computer science,robotics,aerospace,and molecular biology: **how can we tell if two 3D objects are the same, even when one has been rotated or moved**? To do this , Rotate one object using euler angles until it matches the other.

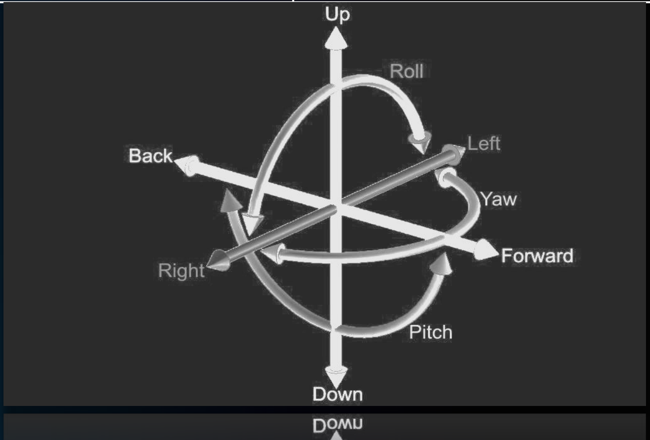


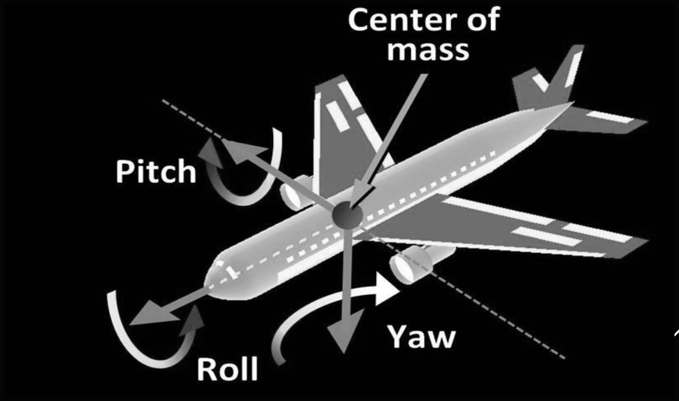
Think of it like comparing two LEGO models — if one is turned sideways or upside down, you wouldn't know it's the same until you rotate it back properly.

Main Idea

The three types of rotations — Yaw, Pitch, and Roll — allow us to rotate objects around the X, Y, and Z axes, just like in airplane flight control.

Yaw (ϕ)	Rotates around the z-axis , Like turning your head left or right (horizontal spin)
Pitch (θ)	Rotates around the y-axis , Like nodding your head up and down (tilting forward or backward)
Roll (ψ)	Rotates around the x-axis ,Like tilting your head side to side (rotating along your nose)



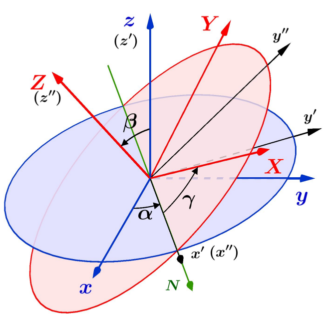


Approach

Explored both simple and advanced methods to match objects:

- First, using **Euler angles** and a brute-force approach,

- Then improving accuracy and speed through **Singular Value Decomposition (SVD)** in what's called the **Orthogonal Procrustes Problem**.



Also accounted for **translation**, not just rotation, and tested our algorithms under **real-world conditions**, including noisy and imperfect data.

Challenge 01

Goal:

- Geometrical meaning of rotation ?
- prove what it means to rotate an object by matrix Q and that Q can be written using yaw (ϕ), pitch (θ), and roll (ψ)?

Explanation:

- Rotating a vector [x, y, z] using a matrix Q changes its direction in 3D space but keeps its length the same. This is because it's based on normalized rotation matrices.
- Yes, for any 3×3 orthogonal matrix Q, there exists a set of angles (ϕ, θ, ψ) such that Q = Qroll × Qpitch × Qyaw.
  This means: take a vector, rotate it around z (yaw), then y (pitch), then x (roll), and the result is the same as applying matrix Q.

Challenge 02

Goal:

- Find the best euler's angle that makes one object A look like another object B after rotation

Explanation:

- Imagine you have two sets of points in 3D:
  A = the original object (like a molecule)
  B = the target object (maybe the same object but rotated)
  Want to find the best set of angles — ϕ (yaw), θ (pitch), and ψ (roll) — that, when applied to A using rotation matrix $Q(\phi, \theta, \psi)$, makes Q^*A look as close as possible to B. This is done using a Gauss-Newton iteration of an initial guess in the co-ordinates of the rotation.

- Measuring Closeness by using the formula : $f(\phi,\theta,\psi) = \| B - Q(\phi, \theta, \psi) * A \|^2$
- calculate RMSD (Root Mean Squared Distance): $\mathrm{sqrt}(f/n)$

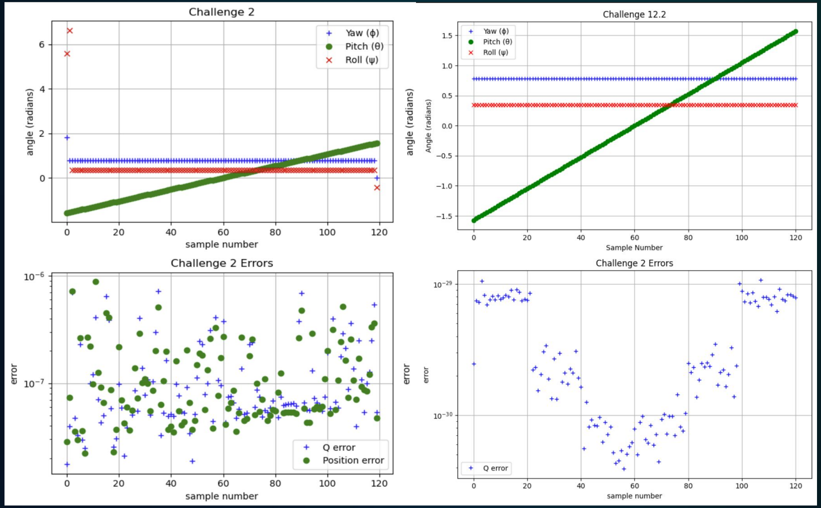
Note : Brute force is like trying every possible combination until you find the one that works best.
Least squares :Try to minimize the difference between the predicted values and the actual values.

Challenge 02

Procedure:

- Use a known set of angles: Yaw (ϕ) = π/4 , Roll (ψ) = π/9 and Try different values of pitch (θ) between −π/2 and π/2
- Compute Q and **B= Q(ϕ, θ, ψ) * A)**, where A is a matrix with coordinates
- Pretend you don't know the angles and try to find them using a solver
- For each trial of θ, find the best ϕ and ψ to minimize the difference
- Plot:The **angles found** and **The RMSD** value

Results:



Challenge 03

Goal:

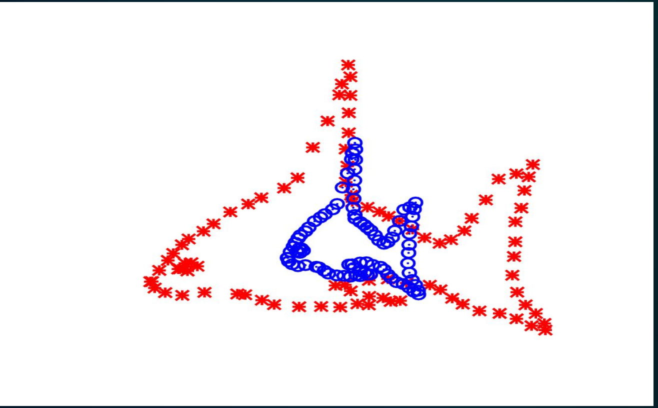
- Find a faster and smarter way to find the best rotation matrix Q that makes one object look like the other.

Explanation:

- For any square matrix C , **trace(C^TC)= // C // 2**
- If CD is a square matrix , **trace(CD)=trace(DC)**
- We measure closeness with: // B–QA // ^2 (Frobenius Norm)

  Minimizing |B - QA|² is the same as maximizing: $\mathrm{trace}(A^T Q^T B)$

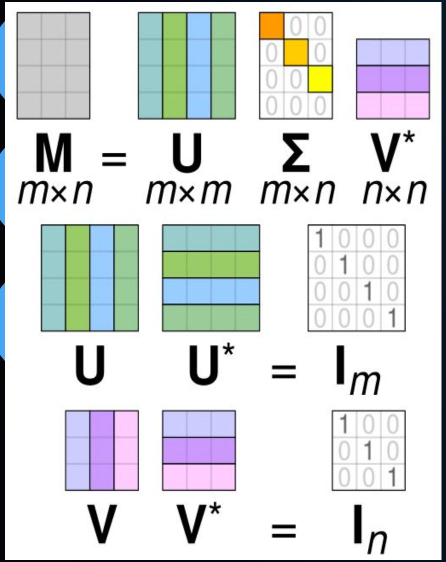
If you want two point clouds to match, it's easier to maximize alignment than minimize the difference.



Challenge03

Results:

- SVD (Singular Value Decomposition) — which is like breaking a big matrix into easier pieces.
- We are given a matrix A and a known rotation matrix Q constructed using Euler angles (yaw ϕ, pitch θ, roll ψ). We compute, B=QA
- Then, we calculate: $BA^T = U\Sigma V^T$
- Using this SVD, we find the optimal rotation $Q = UV^T$
- This computed Q turns out to be exactly the same as the original rotation matrix.
- U: An orthogonal matrix whose columns are called the left singular vectors
- V: An orthogonal matrix whose columns are the right singular vectors
- Σ: A diagonal matrix of singular values (non-negative numbers that represent “strength” or “importance” of each dimension)
- U rotates space so that the columns of BA^T align with the main directions of variation in B.
- V rotates space to align with the main directions of A.
- Σ scales these directions.



Conclusion:

The final error was nearly zero, proving the method is highly accurate and mathematically sound.

$$\|B - QA\|_F = 2.22 \times 10^{-16} \approx 0$$

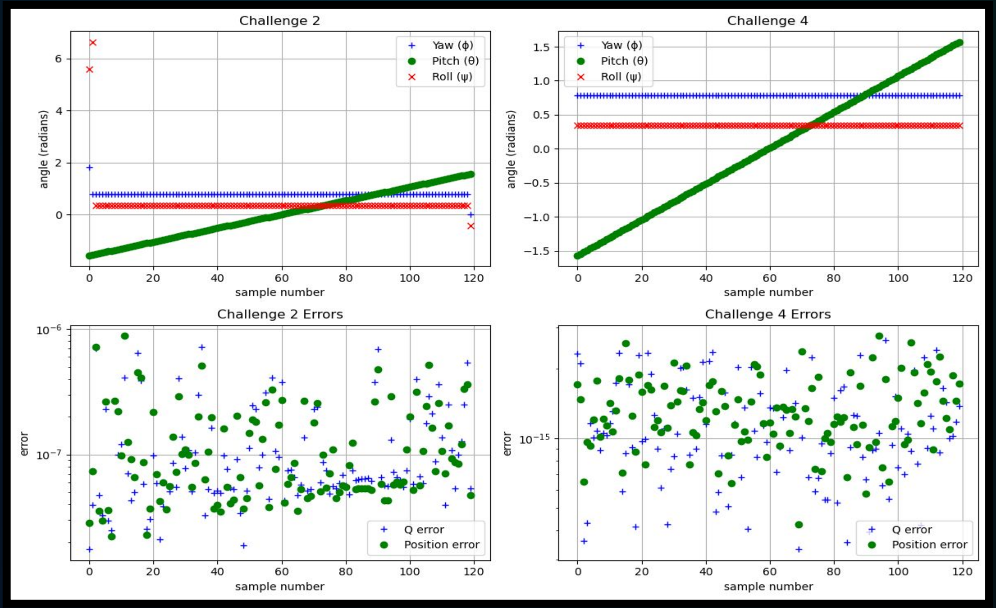
Challenge 04

Goal:

- Comparison between Nonlinear and SVD method

Results:

Challenge 4 shows that using SVD to compute rotation matrices is highly accurate, efficient, and reliable. Compared to the nonlinear method in Challenge 2, it gives much lower error, avoids instability, and consistently recovers the correct Euler angles. This makes SVD a superior method for precise 3D alignment tasks



Challenge 05

Goal: compute the optimal translation vector t that minimizes the Frobenius norm of the difference between matrix B and the transformed matrix QA+te^T , where:
Q is a given rotation matrix,
e is a vector of ones, and
t is the translation vector we want to find.

Explanation:

- The problem is framed as minimizing $\|B - QA - te^T\|_F^2$
- The solution involves taking the derivative of this expression with respect to t and setting it to zero.This leads to the result: $t = c_B - Qc_A$
- Where:cA and cB are the centroids (column-wise means) of matrices A and B, respectively.

Conclusion: incorporating translation significantly reduces alignment error. The optimal translation vector is simply the difference between the centroids of B B and the rotated A. This result is geometrically intuitive and reduces error to numerical precision.

$$t = \begin{bmatrix} 0.1692 \\ 1.1763 \\ 1.0587 \end{bmatrix} - \begin{bmatrix} 0.1516 \\ 1.1267 \\ 1.0363 \end{bmatrix} = \begin{bmatrix} 0.0176 \\ 0.0496 \\ 0.0224 \end{bmatrix}$$

Challenge 06

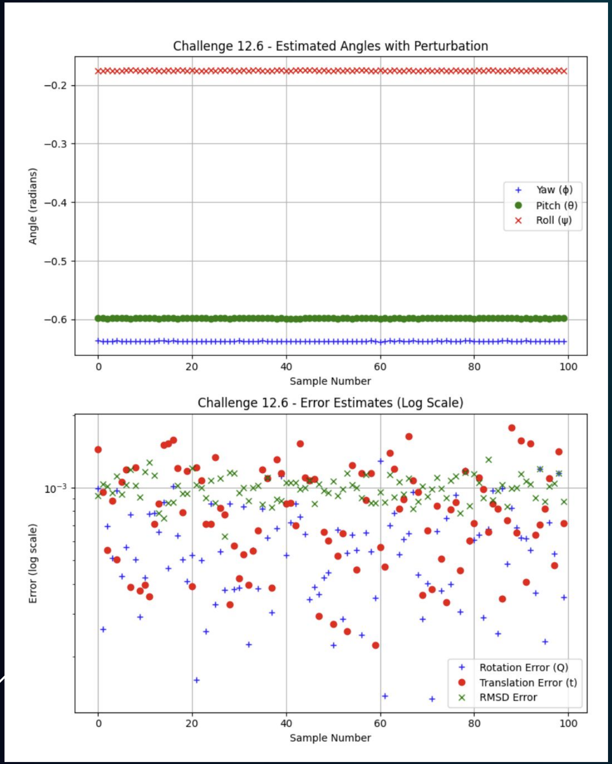
Goal:

- Apply the previous algorithm to the data in Challenge 02, using θ = π/4 and 20 randomly generated translations (t), adding a rotation uniformly distributed between -10^{-3} , 10^{3}

Explanation:

- Construct matrix $B = QA + te^T$ using the known rotation matrix Q and generated translation t.
- Recover the rotation Q' and translation t' using: $Q' = UV^\top, \quad t' = c_B - Q'c_A$
- Measure: Euler angle errors , Matrix error $\|Q - Q'\|_F$ and RMSD between B and $Q'A + t'e^T$
- Repeat the experiment by adding random **noise** (perturbation between 10^{-3} and 10^{-3} to each entry of A.

Challenge 06



Conclusion: SVD-based recovery of rotation and translation is extremely accurate under ideal conditions and remains robust even with small perturbations. This demonstrates the practical reliability of the Procrustes solution using SVD, making it suitable for real-world applications where measurement noise is inevitable .

Results:

We can thus see, Q can very easily be determined with small error using the SVD algorithm, including translations.

Challenge 07

Goal:

- Supposing all points in A lie on a line. Is there more than one choice of Q that minimizes ||B- QA||?
- Characterize degenerate cases for which Q is not well-defined
- Supposing true data (ϕ, $\theta = \pi/2$, ψ), describe an example where a small perturbation causes an increase in θ by 0.01

Results:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} \quad Q = \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$

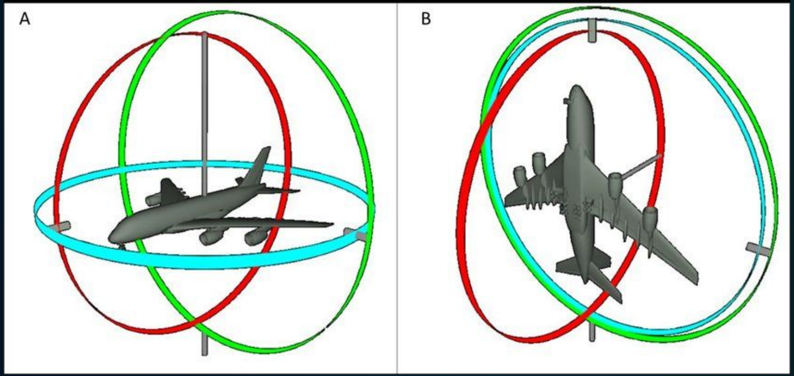
Challenge 07

Conclusion:

The sudden shift in theta, represents, axes aligning with each other during rotation, causing loss of one degree of freedom

$$\phi \approx 1.57 \quad \left(\approx \frac{\pi}{2}\right), \quad \theta \approx 0, \quad \psi \approx 0.436 \quad \left(\approx \frac{\pi}{9}\right)$$

$$\phi' \approx 1.58, \quad \theta' \approx -0.0034, \quad \psi' \approx 0.436$$



Summary

Challenge	Focus	Main Concept/Method	Key Outcome
12.1	Defining rotation via Euler angles	Rotation matrix as $Q = Q_{\text{roll}} Q_{\text{pitch}} Q_{\text{yaw}}$	Introduced Euler angles and matrix structure for 3D rotations
12.2	Recovering Euler angles with brute-force method	Nonlinear least squares	Sensitive to pitch values; less accurate with higher errors
12.3	Optimal rotation using SVD	Orthogonal Procrustes via SVD	Efficiently recovers $Q = UV^T$ with minimal error
12.4	Extracting Euler angles from computed Q	Trigonometric relations from rotation matrix	Recovered angles match original values with high precision
12.5	Finding optimal translation	Centroid alignment using: $t = c_B - Qc_A$	Ensures both rotation and position are aligned accurately
12.6	Sensitivity to noise and perturbations	SVD with noisy data and random translations	SVD remains robust; error increases only slightly with noise
12.7	Degenerate cases and gimbal lock	Rank deficiency and angular instability	Identified cases where Q is not unique or unstable near $\theta = \pm\pi/2$

Conclusion

This project showed that combining Euler angles with Singular Value Decomposition (SVD) offers a precise and reliable solution to the Orthogonal Procrustes problem—aligning two sets of 3D points.

SVD-based methods (Challenges 12.3–12.5) consistently produced accurate and stable results, even with noise or perturbations (Challenge 12.6).

In contrast, nonlinear least squares (Challenge 12.2) struggled with instability near vertical pitch angles.

Challenge 12.7 highlighted special cases like gimbal lock, where caution is needed.

RESOURCES

[1] Richard J. Hanson and Michael J. Norris, “Analysis of measurements based on the singular value decomposition,” SIAM J. Scientific and Statistical Computing,2(3):363-373, 1981.

[2] Kenichi Kanatani, “Analysis of 3-d rotation fitting,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(5):543-549, May 1994.

[3] D.W. Eggert and A. Lorusso and R.B. Fisher, “Estimating 3-d rigid body trans formations: a comparison of four major algorithms,” Machine Learning and Appli cations, 9:272-290, 1997.

[4] some national images from google web.

[5] https://www.cs.umd.edu/users/oleary/SCCS/SCCSanswers.pdf