

Computational Geometry

1. Intersect Two Lines



if I have three points and I found the Euclidean distance between P_2 and P_1 and P_2 and P_3 , which one will be a better error metric

In this case, you can calculate both distances:

1. Euclidean Distance between P_2 and P_1 (let's call it $D(P_2, P_1)$).
2. Euclidean Distance between P_2 and P_3 (let's call it $D(P_2, P_3)$).

Then, you can compare these two distances to determine which one is smaller. The one with the smaller value would represent the closer point to P_2 in terms of Euclidean distance. This can be useful, for example, in nearest neighbor search or clustering algorithms to identify the nearest point to a given reference point.

So, if your goal is to find the minimum of these two distances to assess which point is closer to P_2 , then you are using Euclidean distance as a metric to make that determination.

genCase 1

$$P_1 = [0, 0, 1] \quad P_2 = [1, 0, 0]$$

$$V_1 = [1, 0, 0] \quad V_2 = [0, 1, 0]$$

$$V_3 = \frac{V_1 \times V_2}{\|V_1 \times V_2\|} = \frac{\begin{matrix} x & y & z \\ [1, 0, 0] \end{matrix} \times \begin{matrix} x & y & z \\ [0, 1, 0] \end{matrix}}{\|V_1 \times V_2\|}$$

$$= 0 \times 0 - 1 \times 0, 0 \times 0 - 1 \times 0, 1 \times 1 - 0, 0$$

$$V_3 = [0, 0, 1]$$

$$P_matrix = \begin{bmatrix} P_{1x} - P_{2x} \\ P_{1y} - P_{2y} \\ P_{1z} - P_{2z} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$V_matrix = \begin{bmatrix} -V_{1x}, V_{2x}, V_{3x} \\ -V_{1y}, V_{2y}, V_{3y} \\ -V_{1z}, V_{2z}, V_{3z} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$t_matrix = V_matrix^{-1} \times P_matrix$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$L_1 = P_1 + t_1 v_1$$

$$L_2 = P_2 + t_2 v_2$$

$$\begin{aligned} L_1 &= [0, 0, 1] + 1 [1, 0, 0] \\ &= [1, 0, 1] \end{aligned}$$

$$L_2 = [1, 0, 0]$$

$$\text{Intersection} = \underline{L_1 + L_2} = [1, 0, 0.5]$$

general case $\frac{2}{2}$

$$P_1 = [0, 0, 0]$$

$$P_2 = [0, 1, 0]$$

$$V_1 = [1, 0, 0]$$

$$V_2 = [1, 0, 0]$$

$$V_3 = [0, 0, 0]$$

\therefore The result vector has no direction
and the two lines are parallel
and do not intersect

Fichtinger, Gabor. "Vector Geometry." Computer Integrated Surgery, Sep. 2023, Queen's University, Microsoft PowerPoint presentation.

"How to Find Distance between Two Points?" MATLAB Answers - MATLAB Central, www.mathworks.com/matlabcentral/answers/16848-how-to-find-distance-between-two-points. Accessed 8 Oct. 2023.

2. Intersect Line and Plane

.

Sources

Fichtinger, Gabor. "Vector Geometry." Computer Integrated Surgery, Sep. 2023, Queen's University, Microsoft PowerPoint presentation

MathLearnerMathLearner 16311 gold badge22 silver badges88 bronze badges, et al. "How Do I Verify That a Line Is Parallel to a Plane?" *Mathematics Stack Exchange*, 1 Nov. 1961, math.stackexchange.com/questions/1368461/how-do-i-verify-that-a-line-is-parallel-to-a-plane.

General Test Cases

gen case 1

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad n = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \quad p = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad v = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

* check if line and plane are parallel

$$v \cdot n = (1 \times 2) + (1 \times -1) + (1 \times 1) = 2$$

$$p - A = \begin{bmatrix} 1 - 0 \\ 2 - 0 \\ 3 - 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$(A - p) \cdot n = (1 \times 2) + (2 \times -1) + (3 \times 1) = 3$$

$$t = \frac{3}{2}$$

$$L = p + tv = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \frac{3}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3/2 \\ 3/2 \\ 3/2 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1.5 \\ 1.5 \\ 1.5 \end{bmatrix}$$

gen case 9

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad n = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad p = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$v = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

* check if line and plane are orthogonal

$$v \cdot n = 0$$

$$\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= (1 \times 1) + (-1 \times 1) + (0 \times 1) = 0$$

\therefore The line and plane are parallel and there is no intersection

gen case 3

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad n = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \quad p = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad r = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

* check if line and plane are parallel

$$r \cdot n = (1 \times 2) + (2 \times -1) + (1 \times 1) = 1$$

$$A - p = \begin{bmatrix} 1 - 0 \\ 2 - 0 \\ 3 - 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$(A - p) \cdot n = (1 \times 2) + (2 \times -1) + (3 \times 1) = 3$$

$$t = (A - p) \cdot n / r \cdot n = \frac{3}{1} = 3$$

$$L = p + tn = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 3 \end{bmatrix}$$

20271708

3. Intersect Line and Ellipsoid

The image below showcases how I derived the polynomial for the Intersection of a line and a Ellipsoid.

B

Equation of line

$$l = p + tv \rightarrow \begin{matrix} p_x + tv_x \\ p_y + tv_y \\ p_z + tv_z \end{matrix}$$

Equation of Ellipsoid

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

Expand

$$\frac{(p_x + tv_x)^2}{a^2} + \frac{(p_y + tv_y)^2}{b^2} + \frac{(p_z + tv_z)^2}{c^2} = 1$$

$$\frac{(p_x^2 + 2p_x tv_x + t^2 v_x^2)}{a^2} + \frac{(p_y^2 + 2p_y tv_y + t^2 v_y^2)}{b^2} + \frac{(p_z^2 + 2p_z tv_z + t^2 v_z^2)}{c^2} = 1$$

Collect like terms and simplify

$$(v_x^2 + v_y^2 + v_z^2) t^2 + 2(p_x v_x + p_y v_y + p_z v_z) t + (p_x^2 + p_y^2 + p_z^2 - a^2 b^2 c^2)$$

$$\text{let } (v_x^2 + v_y^2 + v_z^2) = A$$

$$\text{let } 2(p_x v_x + p_y v_y + p_z v_z) = B$$

$$\text{let } (p_x^2 + p_y^2 + p_z^2 - a^2 b^2 c^2) = C$$

$$\text{So, } A t^2 + B t + C$$

Sources

“Ellipsoid.” *Wikipedia*, Wikimedia Foundation, 11 Sept. 2023, en.wikipedia.org/wiki/Ellipsoid.

Fichtinger, Gabor. “Vector Geometry.” Computer Integrated Surgery, Sep. 2023, Queen’s University, Microsoft PowerPoint presentation.

The Discriminant: Equal Roots - Pearson, www.pearson.com/content/dam/one-dot-com/one-dot-com/uk/documents/subjects/mathematics/Worksheets/Chapter2/Example-13-Chapter-2-The-discriminant-equal-roots.pdf. Accessed 7 Oct. 2023.

4. Intersect Sphere and Cylinder




(4) Intersect-Sphere-and-Cylinder




- Compute the number of intersection points (0, 1 infinite) between a sphere and an infinite cylinder. The sphere is given by its center (C) and radius (R). The cylinder is given by its radius (r), a point on its central axis (P) and the direction vector of its central axis (v). Explain your approach in comment or on paper.
- Input: (C, R), (r, P, v)
- Output: number of intersections (0, 1, infinite)
- Testing: Sketch up on paper at least three test cases with easy-to-see solutions (a.k.a. ground-truth). Examine if your program produces the expected output.

What steps should i follow to solve this?



To compute the number of intersection points between a sphere and an infinite cylinder, follow these steps:

1. **Calculate Vector CP:** Compute the vector from the center of the sphere (C) to a point on the central axis of the cylinder (P). This vector represents the direction from the sphere's center to the cylinder's axis.
2. **Projection onto Cylinder's Central Axis:** Find the projection of this vector (CP) onto the cylinder's central axis (v). This projection tells you how far along the direction of the cylinder's axis the sphere's center is located.
3. **Closest Point on Cylinder's Central Axis:** Determine the closest point (P') on the cylinder's central axis to the sphere's center. This is where the projection from step 2 intersects the axis.
4. **Compute Distance:** Calculate the distance between the closest point (P') on the cylinder's central axis and the sphere's center (C). This distance is essential for determining the relative positions of the sphere and the cylinder.
5. **Check Intersection Cases:**
 - If the distance calculated in step 4 is greater than the sum of the sphere's radius (R) and the cylinder's radius (r), there are no intersections. The result is 0 intersections.
 - If the distance is less than the absolute difference between the sphere's radius (R) and the cylinder's radius (r), it means the sphere is completely contained within the cylinder. In this case, there are infinite intersections.
 - If the distance is equal to the sphere's radius (R), it indicates a tangential intersection point. There is one intersection.
 - If the distance is less than the sphere's radius (R), it implies two intersections. There are two points where the sphere's surface intersects the cylinder.

 Regenerate

Sources

5. Reconstruct Sphere

```

A = [2 * points, ones(size(points, 1), 1)];
B = sum(points.^2, 2);

% Solve the linear system of equations using the least squares method.
params = lsqr(A, B);

% Extract the center and radius from the parameters.
C = -params(1:3);
t = (C(1)^2) + (C(2)^2) + (C(3)^2) + params(4);
R = sqrt(t);

end

```

5) Reconstruct-Sphere

- Reconstruct the best fitting sphere from a set of points. Explain your approach in comments. The sphere will be defined by center point (C) and radius (R).


- Input: [points-in]

- Output: C, R

Page 2 of 4

- Testing: Generate points on a general sphere (other than the canonical unit sphere), use 20×20 surface patches, reconstruct, and examine the result.

Using the code and info above, write code to plot a sphere and generate random points

 Regenerate



To test the `reconstructSphere` function, you can create a set of 3D points that are approximately located on a sphere. Here's an example with randomly generated points distributed on a sphere's surface:



matlab

Copy code

```
% Generate random points on a sphere's surface
numPoints = 100;
radius = 5; % Adjust the radius as needed
theta = 2 * pi * rand(numPoints, 1);
phi = pi * rand(numPoints, 1);
x = radius * sin(phi) .* cos(theta);
y = radius * sin(phi) .* sin(theta);
z = radius * cos(phi);

% Combine the x, y, and z coordinates into a matrix of points
points = [x, y, z];

% Call the reconstructSphere function to fit a sphere to the generated points
[center, fittedRadius] = reconstructSphere(points);

% Display the results
fprintf('True Sphere Radius: %f\n', radius);
fprintf('Fitted Sphere Center: [%f, %f, %f]\n', center(1), center(2), center(3));
fprintf('Fitted Sphere Radius: %f\n', fittedRadius);
```

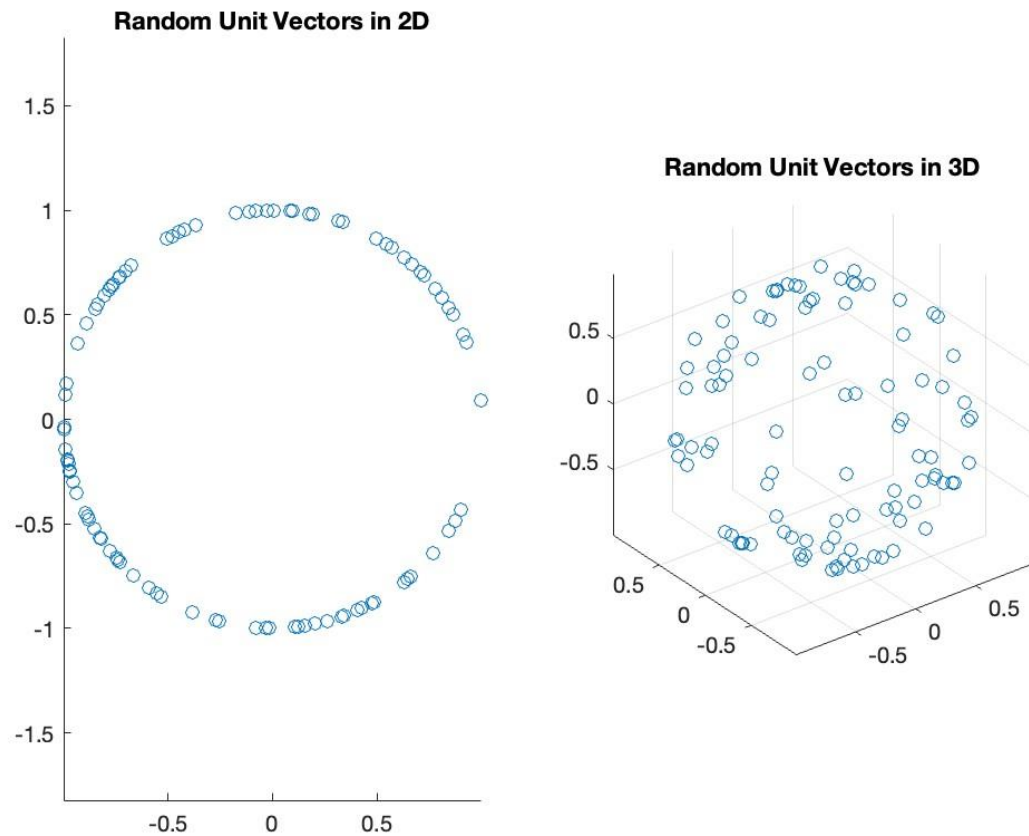
```
% Plot the generated points and the fitted sphere (requires the MATLAB Curve
scatter3(points(:, 1), points(:, 2), points(:, 3), 'b. ');
hold on;
[xSphere, ySphere, zSphere] = sphere;
xSphere = xSphere * radius + center(1);
ySphere = ySphere * radius + center(2);
zSphere = zSphere * radius + center(3);
surf(xSphere, ySphere, zSphere, 'FaceAlpha', 0.5);
axis equal;
xlabel('X');
ylabel('Y');
zlabel('Z');
title('Fitted Sphere');
```

This code generates random points on the surface of a sphere with a given radius and then uses the **'reconstructSphere'** function to fit a sphere to those points. It also plots the generated points and the fitted sphere using MATLAB's plotting capabilities.

Sources

- “Sphere Fit (Least Squared).” *MathWorks*,
www.mathworks.com/matlabcentral/fileexchange/34129-sphere-fit-least-squared.
 Accessed 8 Oct. 2023.
- “Lsqr.” *Solve System of Linear Equations - Least-Squares Method - MATLAB*,
www.mathworks.com/help/matlab/ref/lsqr.html. Accessed 8 Oct. 2023. Jekel, Charles.
- “Least Squares Sphere Fit.” *Charles Jekel - Jekel.Me - Least Squares Sphere Fit*,
jekel.me/2015/Least-Squares-Sphere-Fit/. Accessed 8 Oct. 2023.
- Admin. “Least Square Method.” *BYJUS*, BYJU’S, 24 June 2021, byjus.com/maths/least-square-method/.

6. Generate Random Unit Vector



(6) Generate-Random-Unit-Vector

- Generate a unit vector in random direction in 2D or in 3D.
- Input: 2 or 3 (for the dimension)
- Output: vector
- Testing: Create the following two ground-truth tests: run the function several times for 2D and 3D, and plot resulting points on the canonical unit circle and unit sphere, respectively, and inspect the results.

what steps should i take to solve this



Certainly! Here are the steps taken in the provided MATLAB function to generate a random unit vector in either 2D or 3D:



1. Check the Input Dimension:
 - Check if the input `'dim'` is equal to 2 or 3 to determine the desired dimension of the unit vector.
2. Generate Random Components:
 - If `'dim'` is 2, generate a random angle between 0 and 2π and use it to compute the x and y components of the 2D unit vector.
 - If `'dim'` is 3, generate two random angles (theta and phi) within the appropriate ranges to represent the spherical coordinates. Use these angles to compute the x, y, and z components of the 3D unit vector.
3. Normalize the Vector:
 - After generating the components, normalize the vector by dividing it by its norm (magnitude), ensuring that it becomes a unit vector.
4. Return the Unit Vector:
 - Return the computed unit vector as the output of the function.
5. Handle Invalid Inputs:
 - Display an error message if the input dimension is not 2 or 3, indicating that only these dimensions are supported.

Sources

Fichtinger, Gabor. "Vector Geometry." Computer Integrated Surgery, Sep. 2023, Queen's University, Microsoft PowerPoint presentation.

"Scatter3." *3-D Scatter Plot - MATLAB*,
www.mathworks.com/help/matlab/ref/scatter3.html. Accessed 8 Oct. 2023.

7. Generate Orthonormal Frame

Sources

Fichtinger, Gabor. "Vector Geometry." Computer Integrated Surgery, Sep. 2023, Queen's University, Microsoft PowerPoint presentation.

gen case 1

$$A = [0, 0, 0] \quad B = [1, 0, 0] \quad C = [0, 1, 0]$$

$$Oe = \frac{A+B+C}{3} = \frac{[0, 0, 0] + [1, 0, 0] + [0, 1, 0]}{3}$$

$$= \frac{[1, 1, 0]}{3} = [0.333, 0.333, 0]$$

$$e_1 = \frac{B-A}{\|B-A\|} = \frac{[1, 0, 0] - [0, 0, 0]}{\sqrt{1^2 + 0^2 + 0^2}} = \frac{[1, 0, 0]}{1}$$

$$\therefore e_1 = [1, 0, 0]$$

$$C-A = [0, 1, 0]$$

$$e_3 = e_1 \times (C-A) = \begin{matrix} x & y & z \\ [1, 0, 0] \end{matrix} \times \begin{matrix} x & y & z \\ [0, 1, 0] \end{matrix}$$

$$= [0 \times 0 - 1 \times 0, 0 \times 0 - 1 \times 0, 1 \times 1 - 0 \times 0]$$

$$= [0, 0, 1]$$

$$e_3 = \frac{e^2}{\|e^2\|} = \frac{[0, 0, 1]}{\sqrt{0^2 + 0^2 + 1^2}} = \frac{[0, 0, 1]}{1}$$

$$e_3 = [0, 0, 1]$$

$$\begin{aligned} e_2 &= e_3 \times e_1 = \begin{matrix} x & y & z \\ 0 & 0 & 1 \end{matrix} \wedge \begin{matrix} x & y & z \\ 1 & 0 & 0 \end{matrix} \\ &= [0 \times 0 - 0 \times 1, 1 \times 1 - 0 \times 0, 0 \times 0 - 1 \times 0] \\ &= [0, 1, 0] \\ e_2 &= [0, 1, 0] \end{aligned}$$

genCase 2

$$A = [1, 0, 1] \quad B = [0, 0, 1] \quad C = [0, 1, 1]$$

$$e_1 = \frac{B-A}{\|B-A\|} = [-1, 0, 0]$$

$$C-A = [-1, 1, 0]$$

$$e_3 = \frac{e_1 \times (C-A)}{\sqrt{(C-A) \cdot (C-A)}} = \frac{[-1, 0, 0] \times [-1, 1, 0]}{\sqrt{-1^2 + 1^2 + 0^2}}$$

$$e_3 = [0, 0, -1]$$

$$e_2 = e_3 \times e_1 = [0, 0, -1] \times [-1, 0, 0]$$

$$e_2 = [0, 1, 0]$$

$$oe_2 = \frac{A+B+C}{3} = \frac{[1, 0, 1] + [0, 0, 1] + [0, 1, 1]}{3}$$

$$= \frac{[1, 1, 3]}{3} = [0.333, 0.333, 1]$$

given

$$A = [-1, 0, 0] \quad B = [0, 0, 1] \quad C = [-1, 0, 1]$$

$$OC = \frac{[-1, 0, 0] + [0, 0, 1] + [-1, 0, 1]}{3}$$

$$OC = [-0.667, 0, 0.667]$$

$$e_1 = \frac{B-A}{\|B-A\|} = \frac{[0, 0, 1] - [-1, 0, 0]}{\sqrt{1^2 + 0^2 + 1^2}}$$

$$= [0.71, 0, 0.71]$$

$$C-A = [-1, 0, 1] - [-1, 0, 0] = [0, 0, 1]$$

$$e_3 = \frac{e_1 \times (C-A)}{\|e_1 \times (C-A)\|} = \frac{[0.71, 0, 0.71] \times [0, 0, 1]}{\|e_1 \times (C-A)\|}$$

$$= \frac{[0, -0.71, 0]}{\sqrt{0^2 + 0.71^2 + 0^2}} \text{ or } [0, -1, 0]$$

$$= [0, -1, 0]$$

8. Rotation About Frame Axis

general case - 1

$$p = [0; 0; 1] \quad \theta = \pi/2 \quad \text{axis} = y$$

$$\sin(\pi/2) = 1 \quad \cos(\pi/2) = 0$$

$$\begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \times 0 + 0 \times 0 + 1 \times 1 \\ 0 \times 0 + 1 \times 0 + 0 \times 1 \\ -1 \times 0 + 0 \times 0 + 0 \times 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

general Case 2

$$p = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\theta = \pi$$

$$\text{axis} = x$$

$$s = \sin(\pi) = 0$$

$$c = \cos(\pi) = -1$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \times 1 + 0 \times 0 + 0 \times 0 \\ 0 \times 1 + -1 \times 0 + 0 \times 1 \\ 0 \times 1 + 0 \times 0 + -1 \times 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

general case 3

$$p = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\theta = \frac{\pi}{2}$$

$$\text{axis} = z$$

$$s = \sin(\pi) = 1$$

$$c = \cos(\pi) = 0$$

$$\begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \times 0 + -1 \times 1 + 0 \times 1 \\ 1 \times 0 + 0 \times 1 + 0 \times 1 \\ 0 \times 0 + 0 \times 1 + 1 \times 1 \end{bmatrix}$$

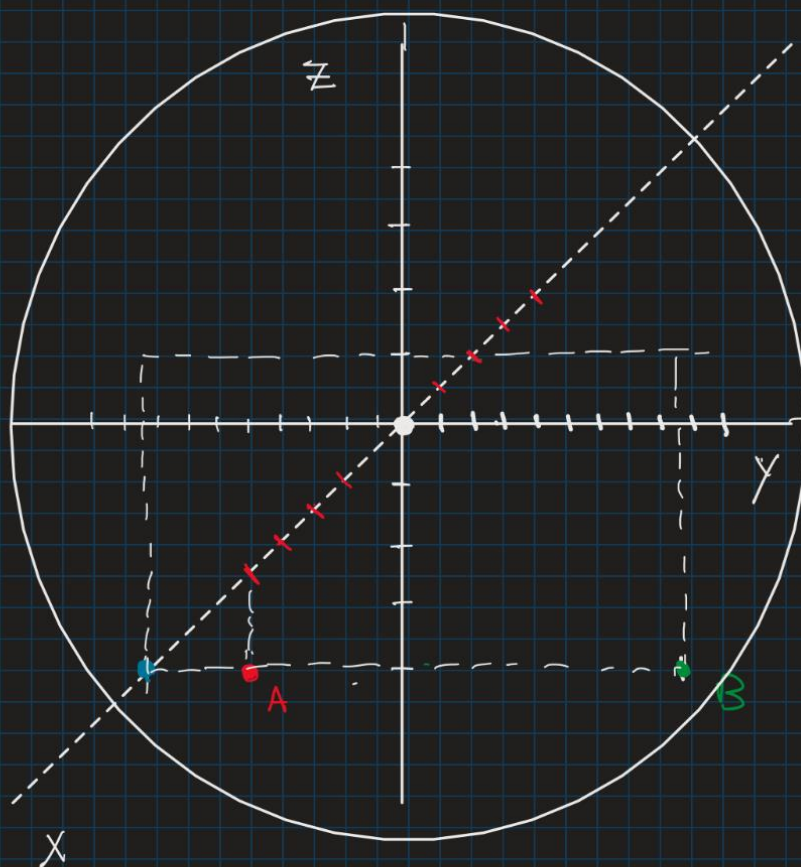
$$= \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Sources

“Rotation Matrix.” *Wikipedia*, Wikimedia Foundation, 29 Sept. 2023, en.wikipedia.org/wiki/Rotation_matrix.

9. Target Registration Error Simulation

10. Target Registration Error Simulation



$$A = (100, 0, -300)$$

$$B = (-50, 86.6, -300)$$

$$C = (-50, -86.6, -300)$$