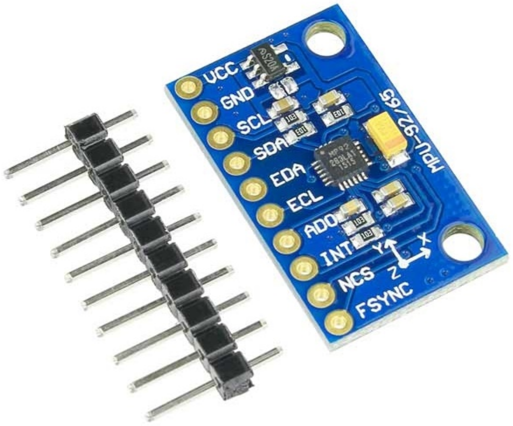
Inertial Motion Tracking using IMUs

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Inertial Measurement Unit (IMU)

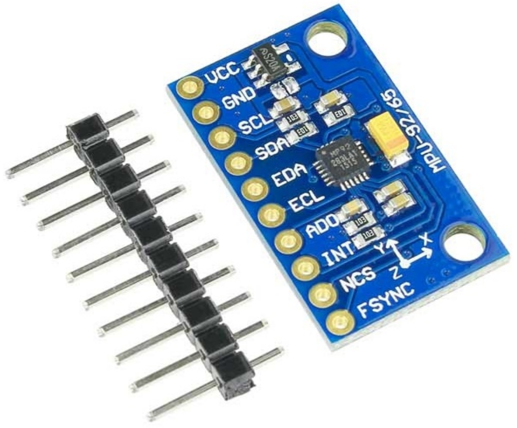


Accelerometer

Gyroscope

Magnetometer

Inertial Measurement Unit (IMU)



Accelerometer

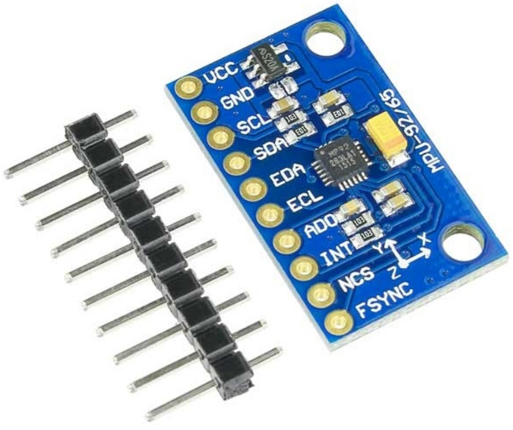
Gyroscope

Magnetometer

GPS

IMU

Inertial Measurement Unit (IMU)



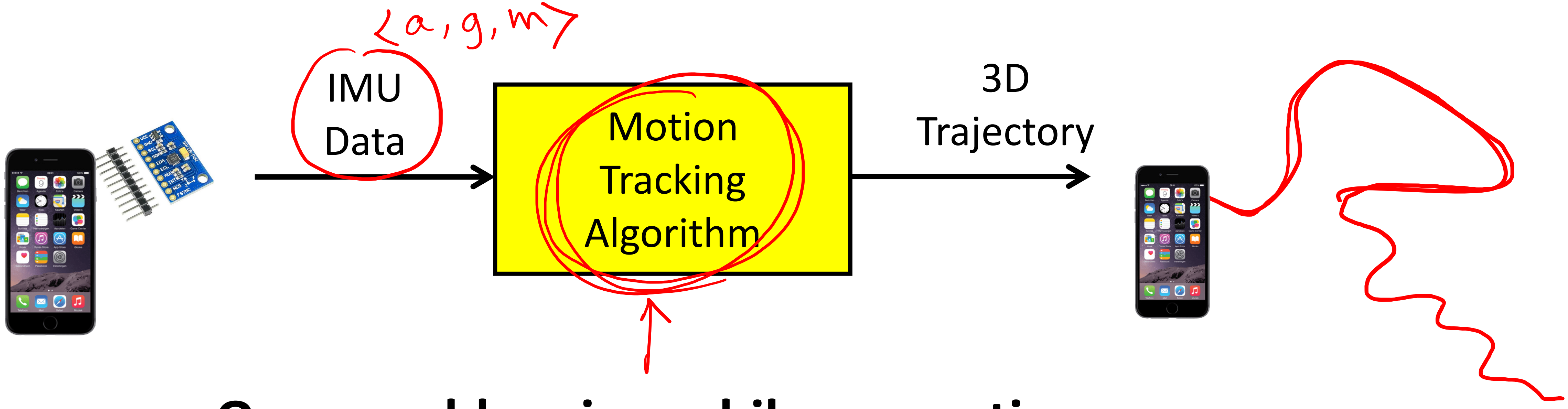
Accelerometer

Gyroscope

Magnetometer

Wide applications in motion tracking

Lot of work in inertial motion tracking



Open problem in mobile computing

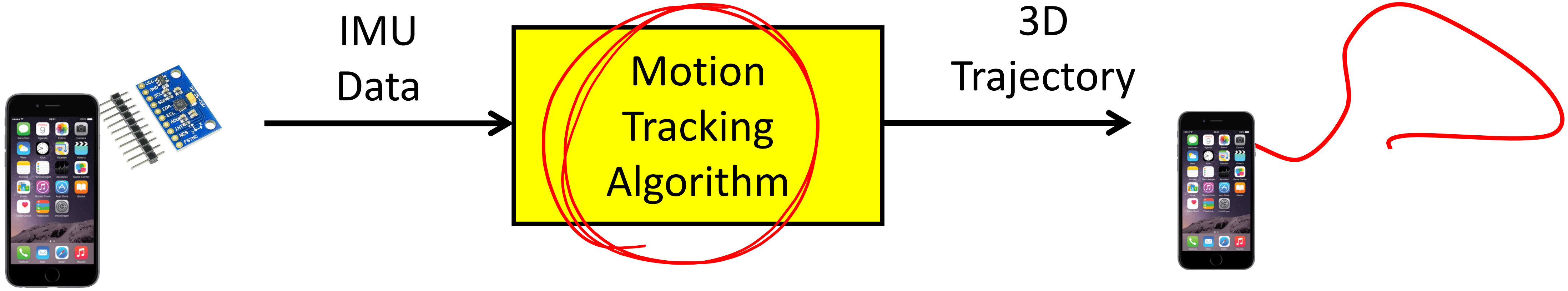
Lot of work in inertial motion tracking

IMU Data → **Motion Tracking Algorithm** → 3D Trajectory

Open problem in mobile computing

No one has the solution … but people making progress

Lot of work in inertial motion tracking



Open problem in mobile computing

No one has the solution … but people making progress

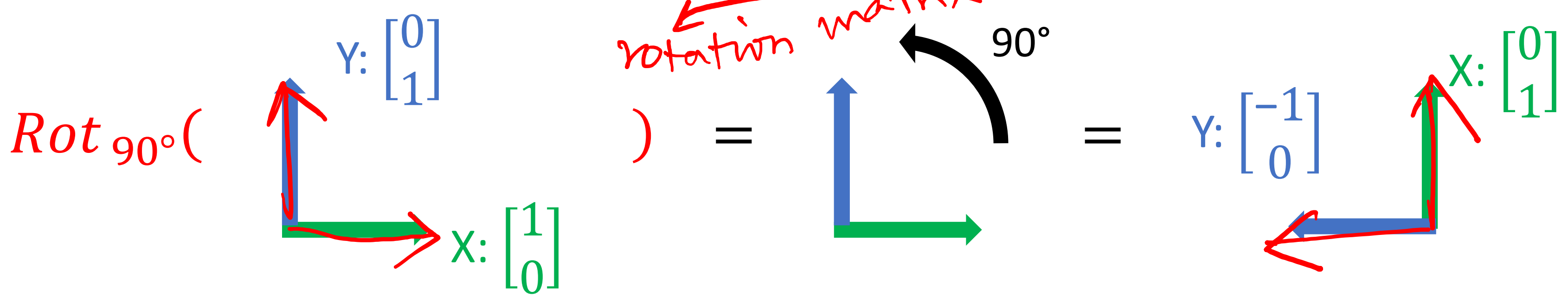
Let’s understand what’s the real difficulty here …

One Prerequisite Slide: Rotation Matrices

$Ax=b$ $\quad x \rightarrow \boxed{A} \rightarrow b$ $\quad$ Think of A as an operator or function on x

rotation matrix

- Rotation is a function



- Mathematically, rotation is a matrix

90° rotation

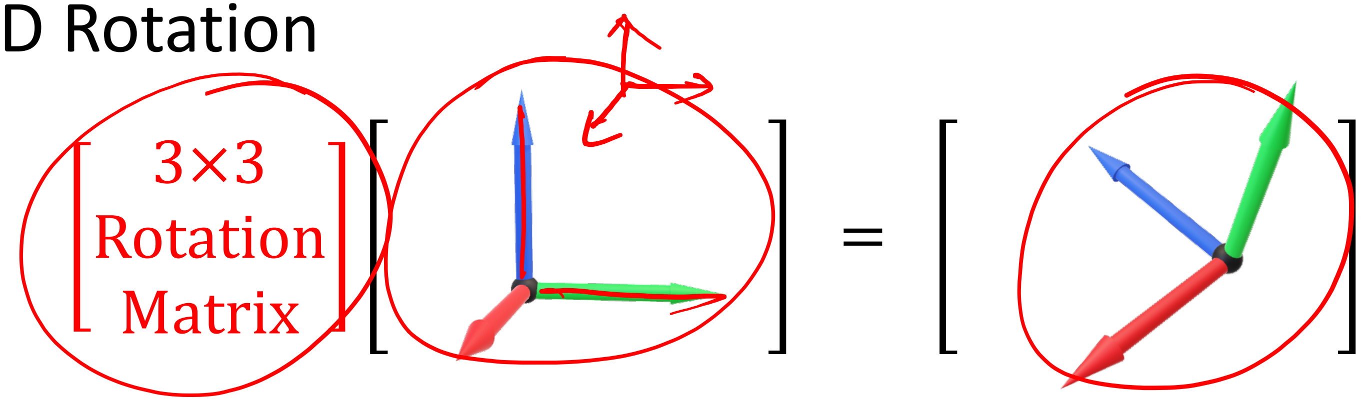
Rotates 90° cc.

$$\begin{bmatrix} \cos 90° & -\sin 90° \\ \sin 90° & \cos 90° \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$Ax = b$

b is 90° rotation of x.

- Same for 3D Rotation



MUSE: Our Goal is 3D Localization

Initial Location

IMU = Accel + Gyro + Mag

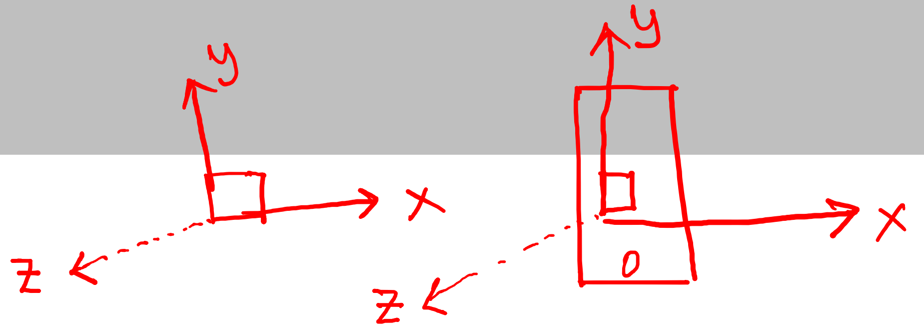
Algorithm

3D

Trajectory

Map

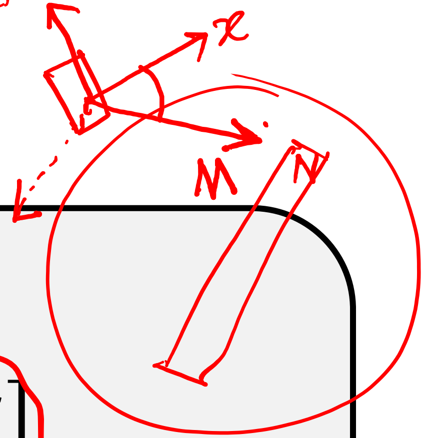
Let's Understand the Inputs



Initial Location $\xrightarrow{[x_0 \quad y_0 \quad z_0]^T}$ **Algorithm**

IMU = Accel + Gyro + Mag $\longrightarrow$ **Algorithm**

angular velocity of IMU around the x axis $= \frac{d\theta}{dt} \Big|_{\text{around } x \text{ axis}}$



Zoom into IMU data:

acceleration along IMU's x direction

200 Hz

$$\text{Accel.} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \qquad \text{Gyro.} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \qquad \text{Mag.} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

(Linear Acceleration) (Angular Velocity) (Earth's Magnetic North)

Earth's mag field projected to x, y, z axes.

Delta rotation in unit time

Can we solve localization with these inputs?

Initial Location $\xrightarrow{[x_0 \quad y_0 \quad z_0]^T}$ **Algorithm** $\xrightarrow{?}$ $\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$

IMU = Accel + Gyro + Mag $\longrightarrow$ **Algorithm**

Zoom into IMU data:

Accel. $= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ (Linear Acceleration)

Gyro. $= \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}$ (Angular Velocity)

Delta rotation in unit time

Mag. $= \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$ (Earth's Magnetic North)

One possibility is:

$\int acc.dt = velocity$

$\int velocity = displacement$

Initial Location $\xrightarrow{[x_0 \;\; y_0 \;\; z_0]^T}$

IMU $\xrightarrow{\text{Accel., Gyro., Mag.}}$

$$\boxed{\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \iint_0^t (Accel.)\, dt^2} \longrightarrow \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$$

where is the plane on earth's map.

Zoom into IMU data:

Accel. $= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ (Linear Acceleration)

Gyro. $= \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}$ (Angular Velocity)

Delta rotation in unit time

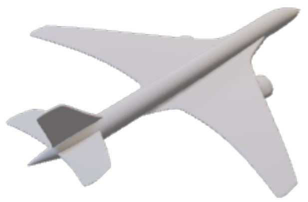
Mag. $= \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$ (Earth's Magnetic North)

But there is one BIG problem: Accel. $= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ is in local reference frame

But there is one BIG problem: Accel. $= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ is in local reference frame

But there is one BIG problem: Accel. $= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ is in local reference frame

As an analogy



Chicago Airport →

Gate 33

Accelerating →

a_x: Frontwards

a_y: Rightwards

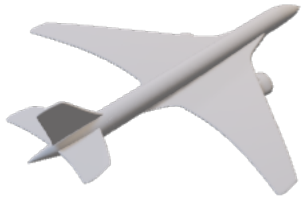
a_z: Upwards

3D → Trajectory

Map

But there is one BIG problem: Accel. $= \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ is in local reference frame

As an analogy



Chicago Airport →

Accelerating →

a_x: Frontwards

a_y: Rightwards

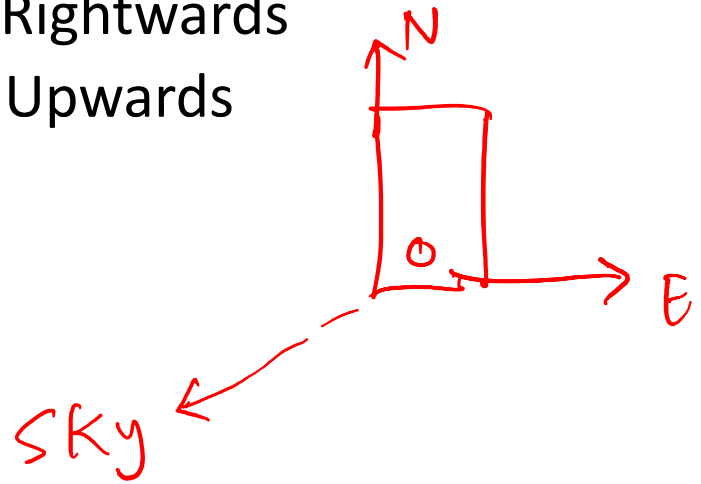
a_z: Upwards

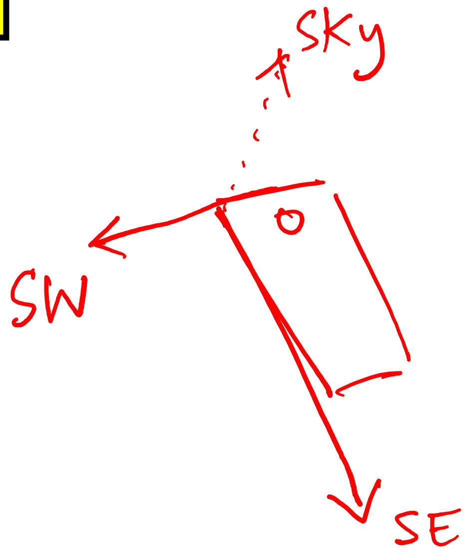
What is
- Frontwards
- Rightwards
- Upwards

w.r.t. the earth?

→ 3D Trajectory

Map





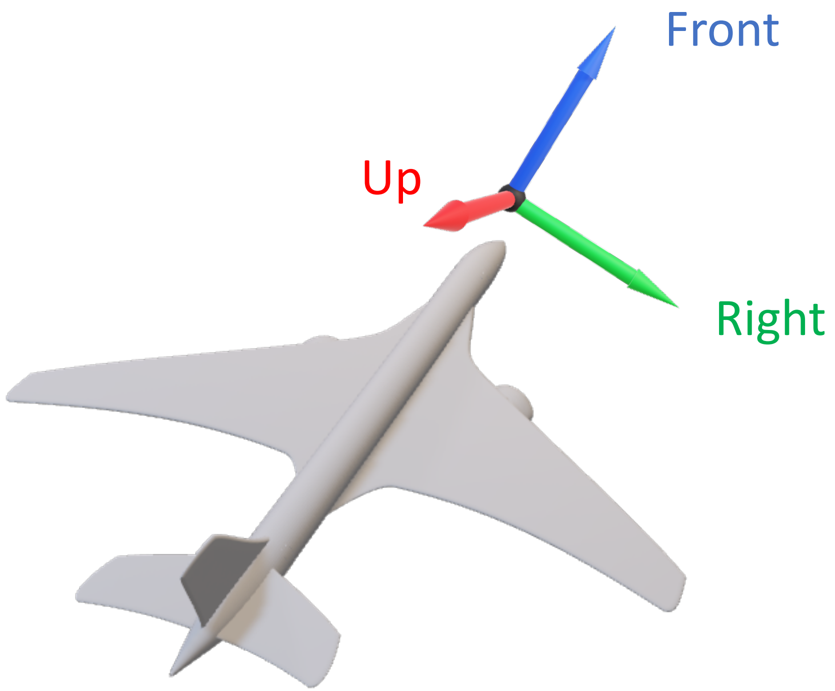
This makes orientation necessary

This makes orientation necessary

- What is 3D orientation?

Orientation is the **3D rotation needed** to make:

Plane's Own $\begin{bmatrix}\text{Frontwards}\\ \text{Rightwards}\\ \text{Upwards}\end{bmatrix} \rightarrow \begin{bmatrix}\text{Northwards}\\ \text{Eastwards}\\ \text{Vertical}\end{bmatrix}$



This makes orientation necessary

- What is 3D orientation?

Orientation is the **3D rotation needed** to make:

Plane's Own $\begin{bmatrix} \text{Frontwards} \\ \text{Rightwards} \\ \text{Upwards} \end{bmatrix} \rightarrow \begin{bmatrix} \text{Northwards} \\ \text{Eastwards} \\ \text{Vertical} \end{bmatrix}$