Math primer for CISC/CMPE 330 part 2: Transformations

Matrices

An ordered table of numbers (or sub-tables)

columns

$$\underbrace{\mathbf{A}}_{3\times 3} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

Rows

$$\text{Identity}: \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix Addition (+)

Definition:

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} + \begin{bmatrix} x & u \\ y & v \end{bmatrix} = \begin{bmatrix} a+x & c+u \\ b+y & d+v \end{bmatrix}$$

Example:

$$\begin{bmatrix} 1 & 5 \\ 2 & 6 \end{bmatrix} + \begin{bmatrix} 0 & -2 \\ 1 & -7 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 3 & -1 \end{bmatrix}$$

Properties:

1. Closed result is always a matrix
2. Commutative A+B = B+A
3. Associative (A+B)+C = A+(B+C)
4. Identity A+0 = A (null matrix)
5. Inverse $A+A^{-}=0$ (negated matrix)

Matrix Multiplication (*)

Definition:

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} * \begin{bmatrix} x & u \\ y & v \end{bmatrix} = \begin{bmatrix} ax+cy & au+cv \\ bx+dy & bu+dv \end{bmatrix}$$

Example:

$$\begin{bmatrix} 1 & 5 \\ 2 & 6 \end{bmatrix} + \begin{bmatrix} 0 & -2 \\ 1 & -7 \end{bmatrix} = \begin{bmatrix} 5 & -37 \\ 6 & -46 \end{bmatrix}$$

Properties:

1.	Closed	result is always a matrix
2.	NOT Commutative	A*B != B*A
3.	Associative	(A*B)*C = A*(B*C)
4.	Identity	A*I = A (identity matrix)
5.	Inverse	$A*A^{-}=I$ (difficult to obtain!)

Distributive from either side

(A+B)C = AC + BC

C(A+B) = CA + CB

Vector is a special matrix

Examples:

v=[x,y,z] row vector, 1x3 matrix

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$ column vector, 2x1 matrix

Vector addition: just like adding matrices

Column vector

$$\begin{bmatrix} a & \cancel{c} \\ b & \cancel{d} \end{bmatrix} + \begin{bmatrix} x & \cancel{u} \\ y & \cancel{v} \end{bmatrix} = \begin{bmatrix} a+x & \cancel{c+u} \\ b+y & \cancel{d+v} \end{bmatrix}$$

Row vector

$$\begin{bmatrix} a & c \\ \cancel{b} & \cancel{d} \end{bmatrix} + \begin{bmatrix} x & u \\ \cancel{y} & \cancel{v} \end{bmatrix} = \begin{bmatrix} a+x & c+u \\ \cancel{b+y} & \cancel{d+v} \end{bmatrix}$$

Matrix*Vector Multiplication:
just like multiplying matrices

$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Column vector on the right

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$

$$= \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} ax_1 + by_1 \\ cx_1 + dy_1 \end{bmatrix}$$

$\boxed{\text{Result is a new column vector}}$

Matrix*Vector Multiplication(example)

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 5 & 1 \\ 2 & 3 \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$

$$= \begin{bmatrix} 5 & 1 \\ 2 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5*1+1*2 \\ 2*1+3*2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

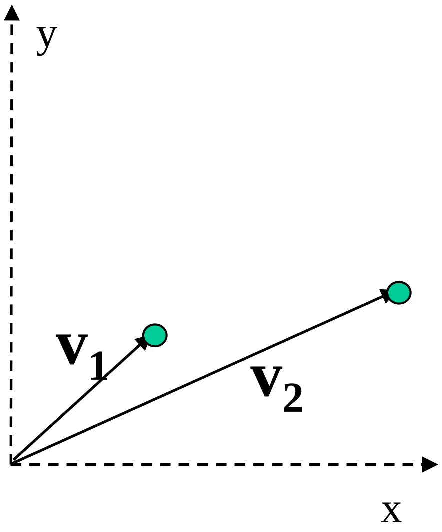
Matrices as transformations

Matrix-vector multiplication creates a new vector.
New vector means new location in space.
If all points of an object is multiplied by a matrix,
then the whole object assumes a new position (and may be new shape and size, too). This is called transformation.

$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$



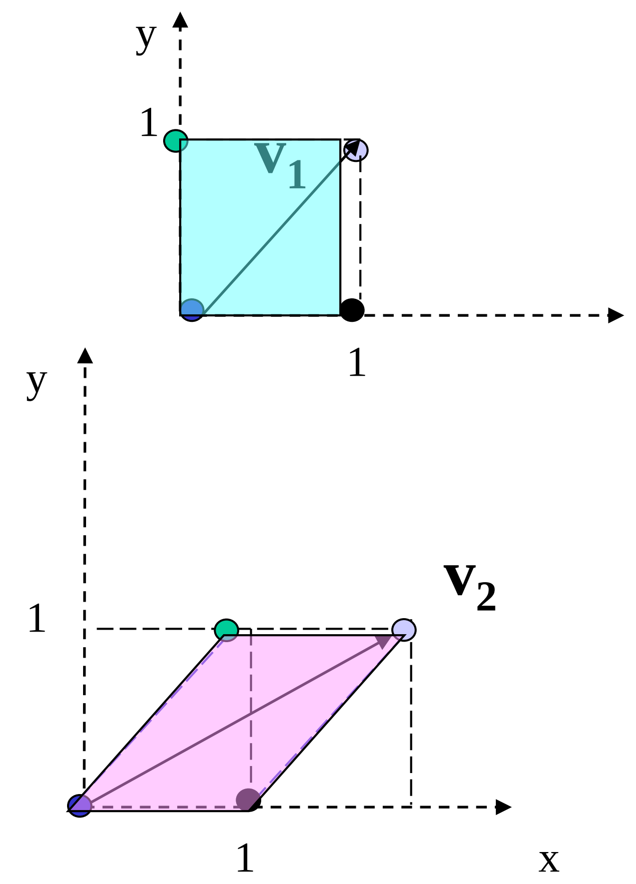
Matrices as transformations (examples)

Transform 4 points [0,0], [1,0],[1,1],[0,1] using the matrix M.

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad \mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



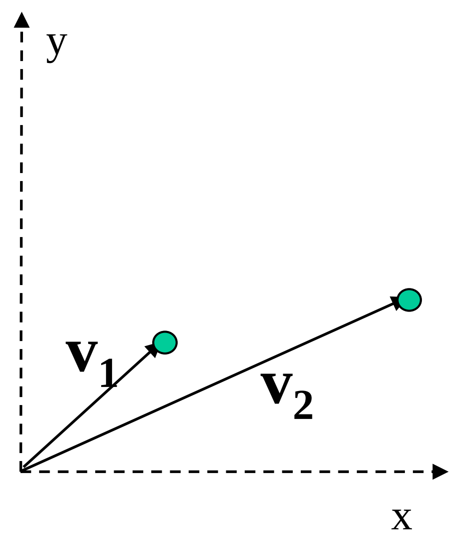
Notations & Conventions

In this course, typically …
For points we use column vectors.
For transformation matrices use square matrices.
To transform a point, we left multiply the column vector with the matrix.
The result is a new (transformed) column vector.

$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$



Notations & Conventions (cont'd)

Sometimes, it is more convenient to use row vectors for points. To transform a point, we right multiply the row vector with the transpose of the transformation matrix. The result is a new row vector.

$$\mathbf{v}_1 = \boldsymbol{M}\mathbf{v}_2 \quad \text{column vector format}$$

If transpose both sides, we receive the row vector format.

$$\mathbf{v}^*{}_1 = \mathbf{v}^*{}_2\,\boldsymbol{M}^* \quad \text{row vector format}$$

Remember to transpose the transformation matrix when going between column and row vector formats.

Scaling

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{S}\mathbf{v}_1$$

$$= \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5*1+0*2 \\ 0*1+3*2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

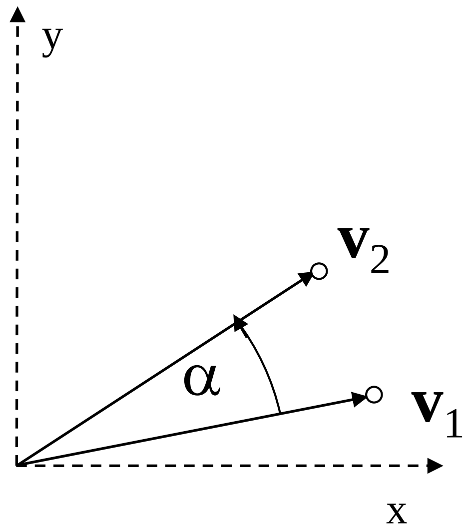
$$\mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \qquad \mathbf{S}^{-1} = \begin{bmatrix} 1/s_x & 0 \\ 0 & 1/s_y \end{bmatrix} \qquad \mathbf{S} * \mathbf{S}^{-1} = \mathbf{I}$$

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$ 3D scaling matrix

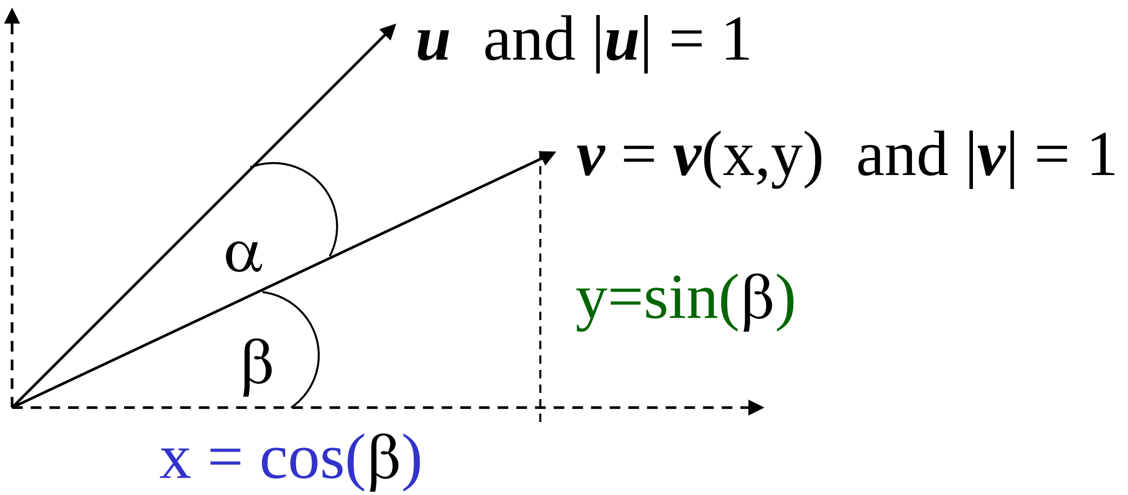
Rotation about the Z axis

$$\mathbf{v}_2 = \mathbf{R}_\alpha \mathbf{v}_1$$

$$\mathbf{v}_1 = \mathbf{R}_{-\alpha} \mathbf{v}_2$$

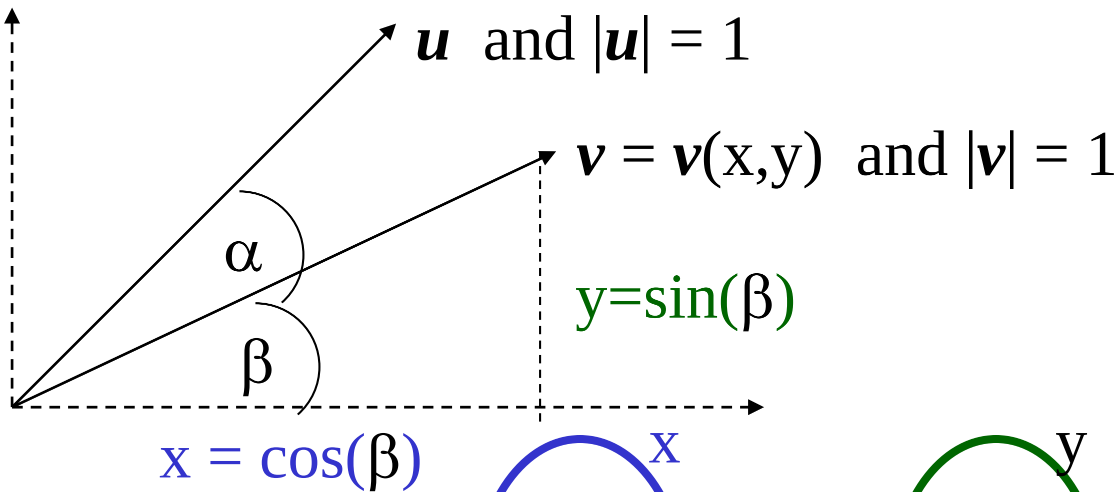


Z-axis Rotation Matrix



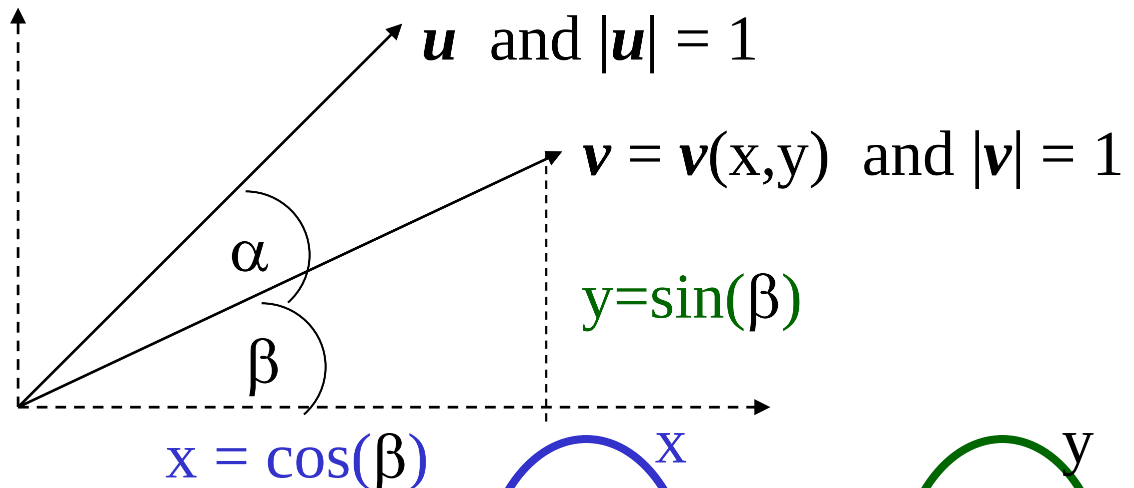
$$\bar{\boldsymbol{u}} = \begin{bmatrix} \cos(\alpha + \beta) \\ \sin(\alpha + \beta) \end{bmatrix} = \begin{bmatrix} \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \end{bmatrix} =$$

Z-axis Rotation Matrix



$$\bar{\boldsymbol{u}} = \begin{bmatrix} \cos(\alpha + \beta) \\ \sin(\alpha + \beta) \end{bmatrix} = \begin{array}{c} \qquad\qquad\qquad x \qquad\qquad\qquad\qquad\qquad y \\ \begin{bmatrix} \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \end{bmatrix} \end{array}$$

Z-axis Rotation Matrix



x y

$$\bar{\boldsymbol{u}} = \begin{bmatrix}\cos(\alpha+\beta)\\ \sin(\alpha+\beta)\end{bmatrix} = \begin{bmatrix}\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)\\ \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)\end{bmatrix} =$$

$$\begin{bmatrix}\cos(\alpha)\,x - \sin(\alpha)\,y\\ \sin(\alpha)\,x + \cos(\alpha)\,y\end{bmatrix} = \begin{bmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} \quad \bar{\boldsymbol{v}}$$

$$\boxed{R_\alpha = \begin{bmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{bmatrix}}$$

$$\boldsymbol{u} = \mathbf{R}_\alpha\,\boldsymbol{v}$$

Inverse of Z-axis Rotation Matrix

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \rightarrow R_{-\alpha} = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) \\ \sin(-\alpha) & \cos(-\alpha) \end{bmatrix} =$$

$$R_{-\alpha} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$R_\alpha^{-1} = R_{-\alpha}$ The proof: $R_\alpha * R_{-\alpha} = I$

Series of consecutive rotations

$$R_{(\alpha+\beta+\gamma)} = R_\alpha R_\beta R_\chi$$

Rotation Matrices around x, y and z

Rotation by α around z axis in 3D notation, as 3x3 matrix

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{v}_2 = R_a \boldsymbol{v}_1$$

When applied to vector $\boldsymbol{v}_1$, it does not change z coordinate. Try it!

Rotation matrices around x, y and z axes in 3D

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \qquad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \qquad R_z(\gamma) = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

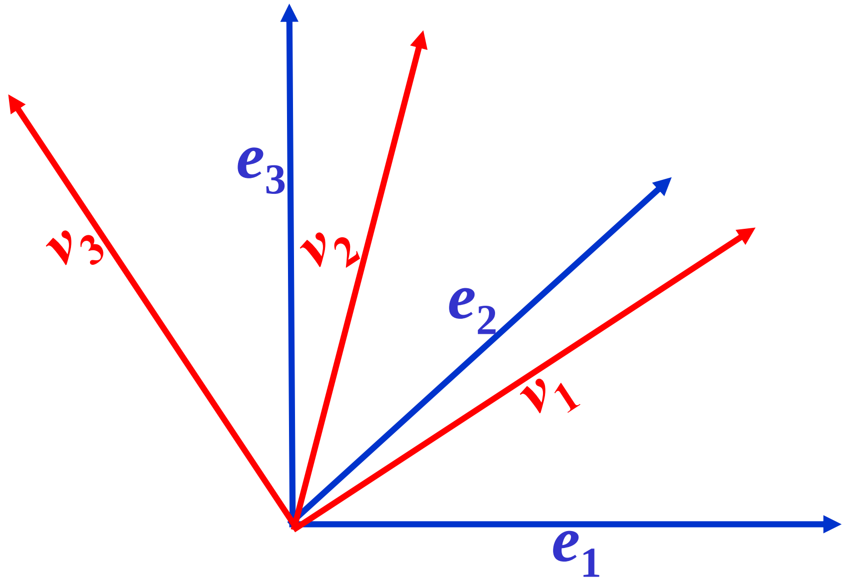
The Generalized Rotation Matrix

$e_1\, e_2\, e_3$ are orthonormal base vectors

$v_1\, v_2\, v_3$ are orthonormal base vectors

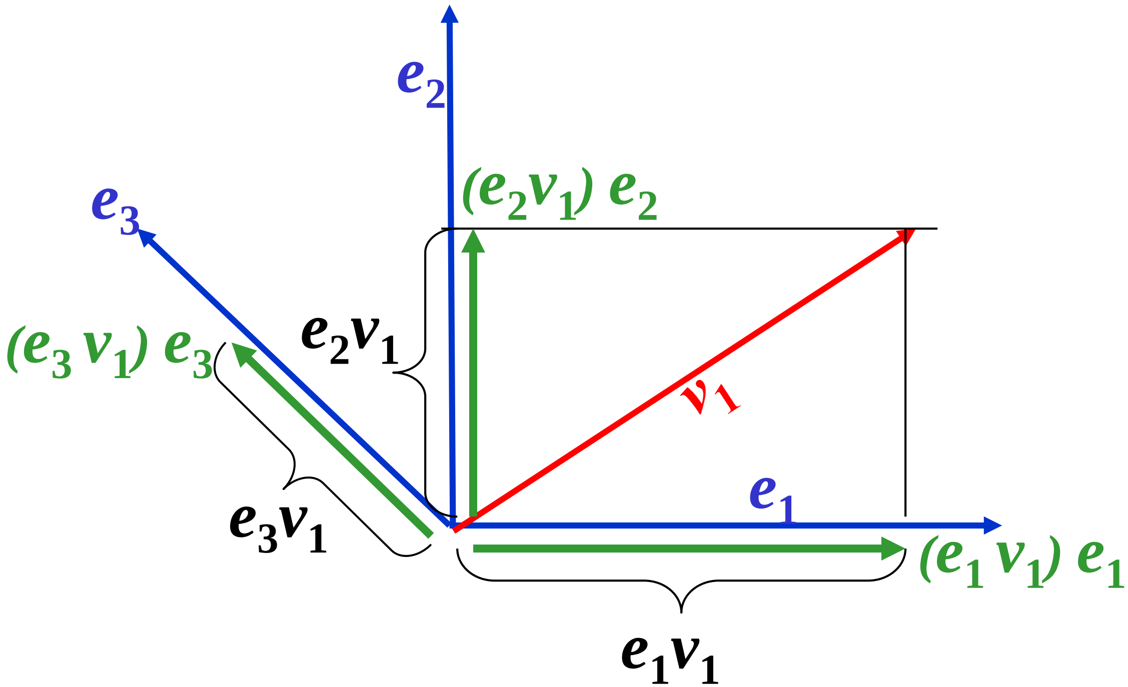
They are centered in a common point

We look for the matrix that rotates one orthonormal vector base to the other about this common point…



Generalized rotation matrix

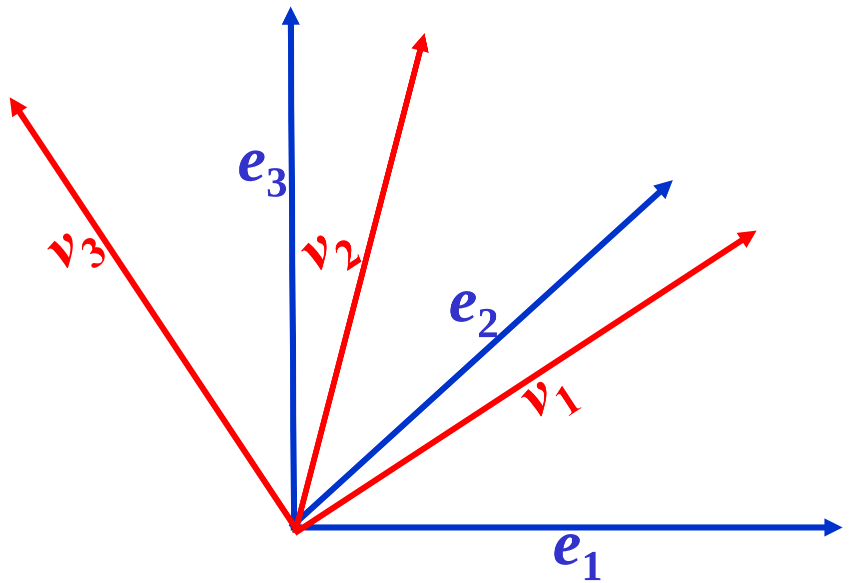
Express v_1 with the $e_1\, e_2\, e_3$ base vectors



$$v_1 = (e_1\, v_1)\, e_1 + (e_2\, v_1)\, e_2 + (e_3\, v_1)\, e_3 \qquad \textbf{(Eq. 1)}$$

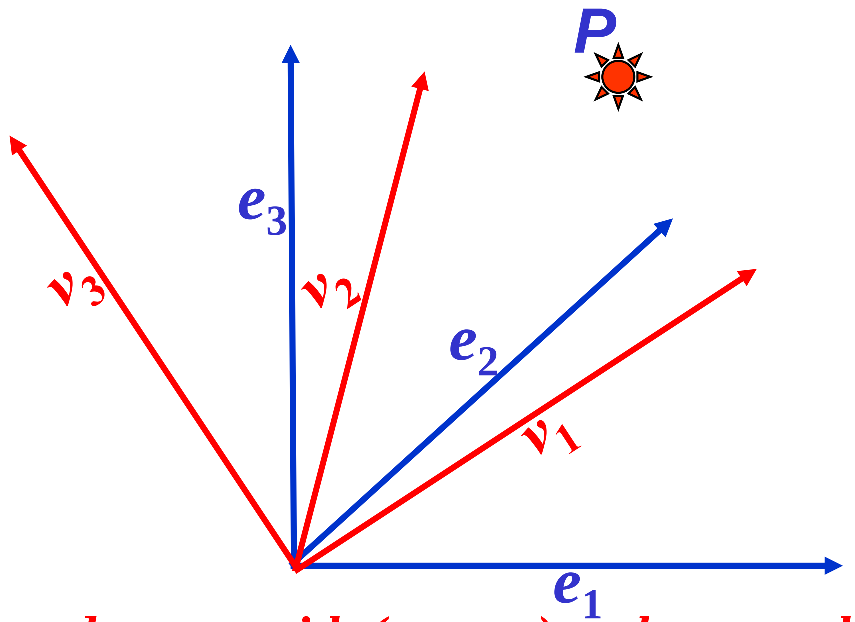
Generalized rotation matrix

Similarly, express $v_1\, v_2\, v_3$ vectors with $e_1\, e_2\, e_3$ base vectors



$$
\begin{aligned}
v_1 &= (e_1 v_1)\, e_1 + (e_2 v_1)\, e_2 + (e_3 v_1)\, e_3 \\
v_2 &= (e_1 v_2)\, e_1 + (e_2 v_2)\, e_2 + (e_3 v_2)\, e_3 \qquad \textbf{(Eq. 2)} \\
v_3 &= (e_1 v_3)\, e_1 + (e_2 v_3)\, e_2 + (e_3 v_3)\, e_3
\end{aligned}
$$

Generalized rotation matrix



(Expressed $v_1\, v_2\, v_3$ with $(e_1\, e_2\, e_3)$ orthonormal base vectors)

$$\boldsymbol{v}_1 = (\boldsymbol{e}_1 \boldsymbol{v}_1)\, \boldsymbol{e}_1 + (\boldsymbol{e}_2 \boldsymbol{v}_1)\, \boldsymbol{e}_2 + (\boldsymbol{e}_3 \boldsymbol{v}_1)\, \boldsymbol{e}_3$$

$$\boldsymbol{v}_2 = (\boldsymbol{e}_1 \boldsymbol{v}_2)\, \boldsymbol{e}_1 + (\boldsymbol{e}_2 \boldsymbol{v}_2)\, \boldsymbol{e}_2 + (\boldsymbol{e}_3 \boldsymbol{v}_2)\, \boldsymbol{e}_3 \qquad \textbf{(Eq. 3)}$$

$$\boldsymbol{v}_3 = (\boldsymbol{e}_1 \boldsymbol{v}_3)\, \boldsymbol{e}_1 + (\boldsymbol{e}_2 \boldsymbol{v}_3)\, \boldsymbol{e}_2 + (\boldsymbol{e}_3 \boldsymbol{v}_3)\, \boldsymbol{e}_3$$

Express P in both coordinate systems…

$\boldsymbol{P} = \mathrm{r}\, \boldsymbol{e}_1 + \mathrm{s}\, \boldsymbol{e}_2 + \mathrm{t}\boldsymbol{e}_3$ (r, s, t are coordinates in the '***e***' system) **(Eq. 4)**

$\boldsymbol{P} = \mathrm{x}\, \boldsymbol{v}_1 + \mathrm{y}\boldsymbol{v}_2 + \mathrm{z}\boldsymbol{v}_3$ (x, y, z are coordinates in the '***v***' system) **(Eq. 5)**

Generalized rotation matrix

We replace v_1, v_2 and v_3 from Eq. 3 into Eq. 5:

$$\left\{\begin{aligned} \boldsymbol{v}_1 &= (\boldsymbol{e}_1 \boldsymbol{v}_1)\,\boldsymbol{e}_1 + (\boldsymbol{e}_2 \boldsymbol{v}_1)\,\boldsymbol{e}_2 + (\boldsymbol{e}_3 \boldsymbol{v}_1)\,\boldsymbol{e}_3 \\ \boldsymbol{v}_2 &= (\boldsymbol{e}_1 \boldsymbol{v}_2)\,\boldsymbol{e}_1 + (\boldsymbol{e}_2 \boldsymbol{v}_2)\,\boldsymbol{e}_2 + (\boldsymbol{e}_3 \boldsymbol{v}_2)\,\boldsymbol{e}_3 \\ \boldsymbol{v}_3 &= (\boldsymbol{e}_1 \boldsymbol{v}_3)\,\boldsymbol{e}_1 + (\boldsymbol{e}_2 \boldsymbol{v}_3)\,\boldsymbol{e}_2 + (\boldsymbol{e}_3 \boldsymbol{v}_3)\,\boldsymbol{e}_3 \end{aligned}\right.$$

$\boldsymbol{P} = x\,\boldsymbol{v}_1 + y\boldsymbol{v}_2 + z\boldsymbol{v}_3$ (x, y, z are coordinate values)

Eq. 5 becomes the following

$$\begin{aligned} \boldsymbol{P} = {} & x(\boldsymbol{e}_1 \boldsymbol{v}_1)\,\boldsymbol{e}_1 + x(\boldsymbol{e}_2 \boldsymbol{v}_1)\,\boldsymbol{e}_2 + x(\boldsymbol{e}_3 \boldsymbol{v}_1)\,\boldsymbol{e}_3 + \\ & y(\boldsymbol{e}_1 \boldsymbol{v}_2)\,\boldsymbol{e}_1 + y(\boldsymbol{e}_2 \boldsymbol{v}_2)\,\boldsymbol{e}_2 + y(\boldsymbol{e}_3 \boldsymbol{v}_2)\,\boldsymbol{e}_3 + \\ & z(\boldsymbol{e}_1 \boldsymbol{v}_3)\,\boldsymbol{e}_1 + z(\boldsymbol{e}_2 \boldsymbol{v}_3)\,\boldsymbol{e}_2 + z(\boldsymbol{e}_3 \boldsymbol{v}_3)\,\boldsymbol{e}_3 \end{aligned} \qquad \textbf{(Eq. 6)}$$

Group the e_1, e_2 and e_3 members together …

$$\begin{aligned} \boldsymbol{P} = {} & x(\boldsymbol{e}_1 \boldsymbol{v}_1)\,\boldsymbol{e}_1 + y(\boldsymbol{e}_1 \boldsymbol{v}_2)\,\boldsymbol{e}_1 + z(\boldsymbol{e}_1 \boldsymbol{v}_3)\,\boldsymbol{e}_1 + \\ & x(\boldsymbol{e}_2 \boldsymbol{v}_1)\,\boldsymbol{e}_2 + y(\boldsymbol{e}_2 \boldsymbol{v}_2)\,\boldsymbol{e}_2 + z(\boldsymbol{e}_2 \boldsymbol{v}_3)\,\boldsymbol{e}_2 + \\ & x(\boldsymbol{e}_3 \boldsymbol{v}_1)\,\boldsymbol{e}_3 + y(\boldsymbol{e}_3 \boldsymbol{v}_2)\,\boldsymbol{e}_3 + z(\boldsymbol{e}_3 \boldsymbol{v}_3)\,\boldsymbol{e}_3 \end{aligned} \qquad \textbf{(Eq. 7)}$$

Generalized rotation matrix

Group the scale factors for e_1 e_2 and e_3

$$P = (x(e_1 v_1) + y(e_1 v_2) + z(e_1 v_3))\ e_1 + (x(e_2 v_1) + y(e_2 v_2) + z(e_2 v_3))\ e_2 + (x(e_3 v_1) + y(e_3 v_2) + z(e_3 v_3))\ e_3 \quad \text{(Eq. 8)}$$

Remember, how P was expressed in the (e_1 e_2 e_3) system in Eq. 4

$P = r\, e_1 + s\, e_2 + t e_3$ (r, s, t are coordinate values)

In Eq.8 we recognize r, s, t as:

$$\begin{aligned} r &= x(e_1 v_1) + y(e_1 v_2) + z(e_1 v_3) \\ s &= x(e_2 v_1) + y(e_2 v_2) + z(e_2 v_3) \\ t &= x(e_3 v_1) + y(e_3 v_2) + z(e_3 v_3) \end{aligned} \quad \text{(Eq. 9)}$$

Generalized rotation matrix

Rearrange Eq 9. to column vector format

$$\begin{bmatrix} r \\ s \\ t \end{bmatrix} = \begin{bmatrix} x(\boldsymbol{e}_1 \boldsymbol{v}_1) + y(\boldsymbol{e}_1 \boldsymbol{v}_2) + z(\boldsymbol{e}_1 \boldsymbol{v}_3) \\ x(\boldsymbol{e}_2 \boldsymbol{v}_1) + y(\boldsymbol{e}_2 \boldsymbol{v}_2) + z(\boldsymbol{e}_2 \boldsymbol{v}_3) \\ x(\boldsymbol{e}_3 \boldsymbol{v}_1) + y(\boldsymbol{e}_3 \boldsymbol{v}_2) + z(\boldsymbol{e}_3 \boldsymbol{v}_3) \end{bmatrix} \quad \text{(Eq. 10)}$$

*Recognize a matrix*vector product*

$$\begin{bmatrix} r \\ s \\ t \end{bmatrix} = \begin{bmatrix} (\boldsymbol{e}_1 \boldsymbol{v}_1) & (\boldsymbol{e}_1 \boldsymbol{v}_2) & (\boldsymbol{e}_1 \boldsymbol{v}_3) \\ (\boldsymbol{e}_2 \boldsymbol{v}_1) & (\boldsymbol{e}_2 \boldsymbol{v}_2) & (\boldsymbol{e}_2 \boldsymbol{v}_3) \\ (\boldsymbol{e}_3 \boldsymbol{v}_1) & (\boldsymbol{e}_3 \boldsymbol{v}_2) & (\boldsymbol{e}_3 \boldsymbol{v}_3) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{(Eq. 11)}$$

Recognize Eq. 4: P in the (e_1,e_2,e_3) system

$R_{e \leftarrow v}$

Rotation matrix that takes P point from *(v_1,v_2,v_3) system to (e_1,e_2,e_3) system*

Recognize Eq. 5: P in the (v_1,v_2,v_3) system

Generalized rotation matrix

$\boldsymbol{R_{e \leftarrow v}}$ ***what is it good for??***

- *R*otates the *(v_1,v_2,v_3) base vectors to (e_1,e_2,e_3) base vectors*
- *Transforms* a point from *(v_1,v_2,v_3) coordinate system to (e_1,e_2,e_3) coordinate system (assuming the v and e coordinate systems are centered in a common point. (If not, we will use translation to bring their centers together...)*

Generalized rotation matrix

$$R_{e \leftarrow v} = \begin{bmatrix} (e_1 v_1) & (e_1 v_2) & (e_1 v_3) \\ (e_2 v_1) & (e_2 v_2) & (e_2 v_3) \\ (e_3 v_1) & (e_3 v_2) & (e_3 v_3) \end{bmatrix} \qquad \text{(Eq. 12)}$$

For the reverse rotation, simply swap the corresponding v and e vectors

$$R_{v \leftarrow e} = \begin{bmatrix} (v_1 e_1) & (v_1 e_2) & (v_1 e_3) \\ (v_2 e_1) & (v_2 e_2) & (v_2 e_3) \\ (v_3 e_1) & (v_3 e_2) & (v_3 e_3) \end{bmatrix} \qquad \text{(Eq. 13)}$$

Note that they are transposes of one another: $R^*_{v, \leftarrow e} = R_{e \leftarrow v}$

Note that they are inverses of one another: $R^{-1}_{v \leftarrow e} = R_{e \leftarrow v}$

Direction cosigns in the generalized rotation matrix

Recognize here the direction cosines

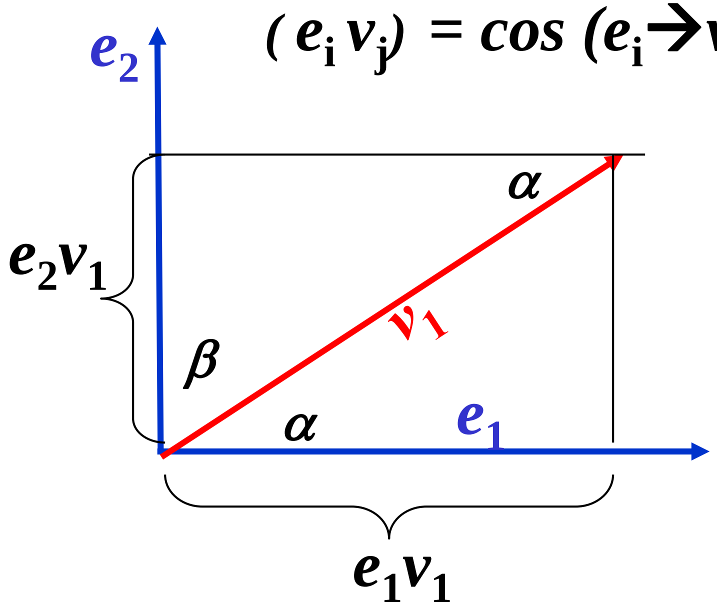
$$R_{e \leftarrow v} = \begin{bmatrix} (e_1 v_1) & (e_1 v_2) & (e_1 v_3) \\ (e_2 v_1) & (e_2 v_2) & (e_2 v_3) \\ (e_3 v_1) & (e_3 v_2) & (e_3 v_3) \end{bmatrix}$$

where

$$(e_1 v_1) = \cos(e_1 \rightarrow v_1) = \cos(\alpha)$$

$$(e_2 v_1) = \cos(e_2 \rightarrow v_1) = \cos(\beta)$$

$$(e_i v_j) = \cos(e_i \rightarrow v_j) = \cos(\beta_{ij})$$

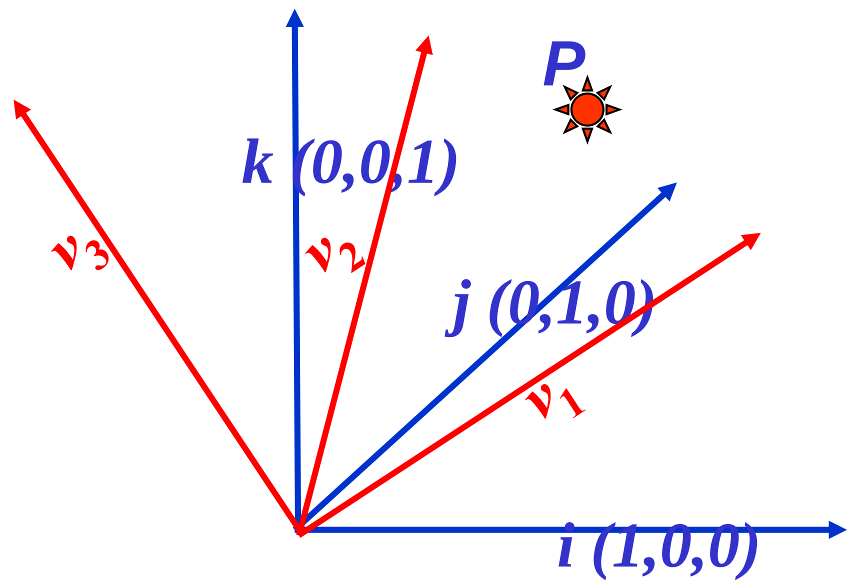


Rotation from *v* to home

$v_1\, v_2\, v_3$ are orthonormal base vectors

They are centered in 0,0,0 in home

We look for the matrix that rotates one orthonormal vector base to the other about this common 0,0,0 point…



$\boldsymbol{P} = r\,\boldsymbol{v_1} + s\,\boldsymbol{v_2} + t\boldsymbol{v_3}$ (r, s, t are coordinates in the '*v*' system) Eq 1

$\boldsymbol{P} = x\,\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$ (x, y, z are coordinates in the home system) Eq 2

Rotation from v to home

We write up v_1, v_2 and v_3 in the Home:

$$\begin{cases} v_1 = v_{1x}\, i + v_{1y}\, j + v_{1z}\, k \\ v_2 = v_{2x}\, i + v_{2y}\, j + v_{2z}\, k \\ v_3 = v_{3x}\, i + v_{3y}\, j + v_{3z}\, k \end{cases}$$

(Eq. (3))

Eq. 1 becomes the following

$$\begin{aligned} P = {} & r(v_{1x}\, i + v_{1y}\, j + v_{1z}\, k) + \\ & s(v_{2x}\, i + v_{2y}\, j + v_{2z}\, k) + \\ & t(v_{3x}\, i + v_{3y}\, j + v_{3z}\, k) \end{aligned}$$

(Eq. (4))

Group the ***i*****, j, k** *members together …*

$$\begin{aligned} P = {} & \boxed{(rv_{1x} + sv_{2x}\, t + tv_{3x})}\, i + \\ & \boxed{(rv_{1y} + sv_{2y} + tv_{3y})}\, j + \\ & \boxed{(rv_{1z} + sv_{2z} + tv_{3z})}\, k \end{aligned}$$

(Eq. 5)

We recognize x,y,z from Eq 2

Rotation from v to home

We recognized x,y,z from Eq2

$$x=(rv_{1x} + sv_{2x} + tv_{3x})$$
$$y=(rv_{1y} + sv_{2y} + tv_{3y})$$
$$z=(rv_{1z} + sv_{2z} + tv_{3z})$$

Arrange to matrix form

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} v_{1x} & v_{2x} & v_{3x} \\ v_{1y} & v_{2y} & v_{3y} \\ v_{1z} & v_{2z} & v_{3z} \end{bmatrix} \begin{bmatrix} r \\ s \\ t \end{bmatrix}$$

$R_{h \leftarrow v}$

Rotation matrix that takes P point from *(v_1,v_2,v_3) system to (i,j,k) home system*

Rotation from v to home

We recognize the v1,v2, v3 vectors as the columns

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} v_{1x} & v_{2x} & v_{3x} \\ v_{1y} & v_{2y} & v_{3y} \\ v_{1z} & v_{2z} & v_{3z} \end{bmatrix} \end{array} \begin{bmatrix} r \\ s \\ t \end{bmatrix}$$

$R_{h \leftarrow v}$

Rotation matrix that takes P point from *(v_1,v_2,v_3) system to (i,j,k) home system*

Rotation from home to v

Simply transpose…

We recognize the v1,v2, v3 vectors as the rows

$$\begin{bmatrix} r \\ s \\ t \end{bmatrix} = \begin{bmatrix} \boxed{v_{1x} \quad v_{1y} \quad v_{1Z}} \\ \boxed{v_{2X} \quad v_{2y} \quad v_{3Z}} \\ \boxed{v_{3X} \quad v_{3Y} \quad v_{3z}} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$R_{v \leftarrow h}$

Rotation matrix that takes P point from *(i,j,k) home to (*v_1*,*v_2*,*v_3*) system to*

Some typical vector transformations…

$\boldsymbol{p}_e = R\boldsymbol{p}_v + \boldsymbol{t}$ //$\boldsymbol{p}_v$ rotated by R and then translated by $\boldsymbol{t}$

$\boldsymbol{p}_e = R(\boldsymbol{p}_v + \boldsymbol{t})$ //$\boldsymbol{p}_v$ translated by $\boldsymbol{t}$ and then rotated by R

$\boldsymbol{p}_e = R_2(R_1\boldsymbol{p}_v + \boldsymbol{t}_1) + \boldsymbol{t}_2$ //$\boldsymbol{p}_v$ rotated by R_1, then translated by $\boldsymbol{t}_1$, then rotated by R_2 and finally translated by $\boldsymbol{t}_2$

Sadly, this formalism mixes 1x3 translation vectors with 3x3 rotation matrices and it looks even nastier for the inverse transformations:

$R^{-1}(\boldsymbol{p}_e - \boldsymbol{t}) = \boldsymbol{p}_v$

$R^{-1}\boldsymbol{p}_e - \boldsymbol{t} = \boldsymbol{p}_v$

$R_1^{-1}(R_2^{-1}(\boldsymbol{p}_e - \boldsymbol{t}_2) - \boldsymbol{t}_1) = \boldsymbol{p}_v$

We need to invent a new formalism to make this mixture disappear and transformations appear as "homogeneous" multiplication of matrices…

The Homogeneous 4x4 Translation Matrix

Let the translation vector be $[d_x, d_y, d_z]$. We construct the T 4x4 matrix:

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad v = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

We want to transform vector $[x,y,z]$. We construct a 4x1 vector by padding with 1.

$$Tv = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+d_x \\ y+d_y \\ z+d_z \\ 1 \end{bmatrix}$$

The Homogeneous 4x4 Rotation Matrix

Padding the 3x3 rotation matrix into a 4x4 matrix:

$$R = \begin{bmatrix} (\boldsymbol{e_1 v_1}) & (\boldsymbol{e_1 v_2}) & (\boldsymbol{e_1 v_3}) & \mathbf{0} \\ (\boldsymbol{e_2 v_1}) & (\boldsymbol{e_2 v_2}) & (\boldsymbol{e_2 v_3}) & \mathbf{0} \\ (\boldsymbol{e_3 v_1}) & (\boldsymbol{e_3 v_2}) & (\boldsymbol{e_3 v_3}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \qquad \boldsymbol{v} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Also padding the vector with 1

Now we can compute R***v***, and the result will be the rotated (x,y,z) vector, padded with 1

Some typical vector transformations…

In mixed format they were:

$\boldsymbol{p}_e = R\boldsymbol{p}_v + \boldsymbol{t}$ //$\boldsymbol{p}_v$ rotated by R and then translated by $\boldsymbol{t}$

$\boldsymbol{p}_e = R(\boldsymbol{p}_v + \boldsymbol{t})$ //$\boldsymbol{p}_v$ translated by $\boldsymbol{t}$ and then rotated by R

$\boldsymbol{p}_e = R_2 (R_1\boldsymbol{p}_v + \boldsymbol{t}_1)+\boldsymbol{t}_2$ //$\boldsymbol{p}_v$ rotated by $R_{1,}$ then translated by $\boldsymbol{t}_1$, then rotated by R_2 and finally translated by $\boldsymbol{t}_2$

***In homogeneous format the are*:**

$\boldsymbol{p}_e = TR\boldsymbol{p}_v$ //$\boldsymbol{p}_v$ rotated by R and then translated by T

$\boldsymbol{p}_e = RT\boldsymbol{p}_v$ //$\boldsymbol{p}_v$ translated by T and then rotated by R

$\boldsymbol{p}_e = T_2R_2 T_1R_1\boldsymbol{p}_v$ //p_v rotated by $R_{1,}$ then translated by T_1, then rotated by R_2 and finally translated by T_2

And the inverses

$\boldsymbol{p}_v = R^{-1}T^{-1}\boldsymbol{p}_e$ //$\boldsymbol{p}_e$ rotated translated by T^{-1} by and rotated R^{-1}

$\boldsymbol{p}_v = T^{-1}R^{-1}\boldsymbol{p}_e$ //$\boldsymbol{p}_e$ rotated R^{-1} and then translated by T^{-1}

$\boldsymbol{p}_v = R_1^{-1}T_1^{-1}R_2^{-1}T_2^{-1}\boldsymbol{p}_e$ //p_v translated by T_2^{-1}, rotated by R_2^{-1}, translated by T_1^{-1} and finally rotated by R_1^{-1}

Our transformations are homogeneous now – they only contain multiplies of 4x4 matrices, applied to a 4x1 (homogeneous) vector.

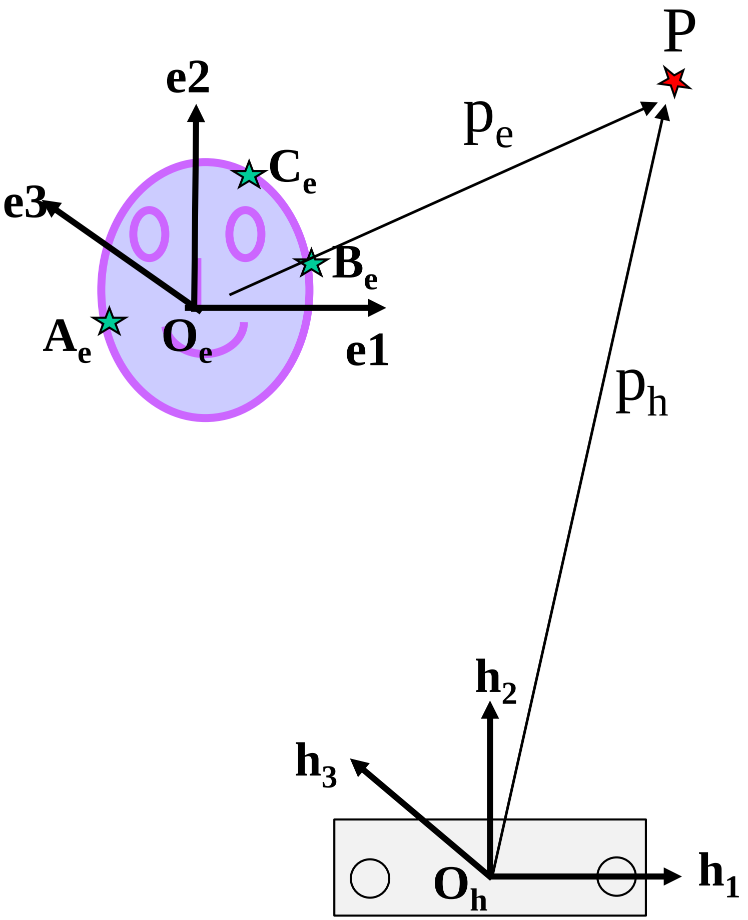
Some typical coordinate frame transformation scenarios …

1. Observe a fixed point from a moving body

2. Multiple observations of a moving body

3. Transform a body between two homes

1. Observe a fixed point from a moving frame



Observation (home) coordinate system

- A rigid body (such as the head) is moving during surgery
- ABC markers are affixed to the rigid head
- The markers are continuously localized (such as with optical stereo camera) in a home coordinate system
- There is a fixed point P in the room.
- In the home system, it is seen as p_h
- How is it seen the e system (p_e) ?
- In pose e, we compute the frame defined by the ABC markers:
  - e_1, e_2, e_3, O_e
- $p_v = \underbrace{R_{e\leftarrow h}\ T_{e\leftarrow h}}_{F_{e\leftarrow h}}\ p_h$

Typically in the home: $\mathbf{O_h} = (0,0,0)$, $\mathbf{h_1} = (1,0,0)$, $\mathbf{h_2} = (0,1,0)$, $\mathbf{h_3} = (0,0,1)$

Series of moving frames

Most real systems involve many (often dozens) of moving frames, where we must traverse from frame to frame. Here is an example with 3 frames (a,b,c)

$$\boldsymbol{p}_c = \overbrace{F_{c\leftarrow b} F_{b\leftarrow a}}^{F_{c\leftarrow a}} \boldsymbol{p}_a$$

We can undo them in reverse order:

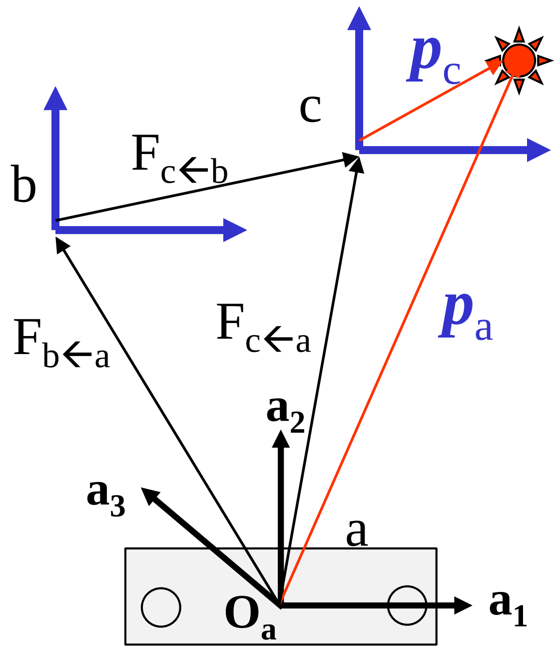
$$\overbrace{F_{a\leftarrow b} F_{b\leftarrow c}}^{F_{a\leftarrow c}} \boldsymbol{p}_c = \boldsymbol{p}_a$$

Obviously…

$$F^{-1}_{a\leftarrow b} = F_{b\leftarrow a}$$

$$F^{-1}_{c\leftarrow b} = F_{b\leftarrow c}$$

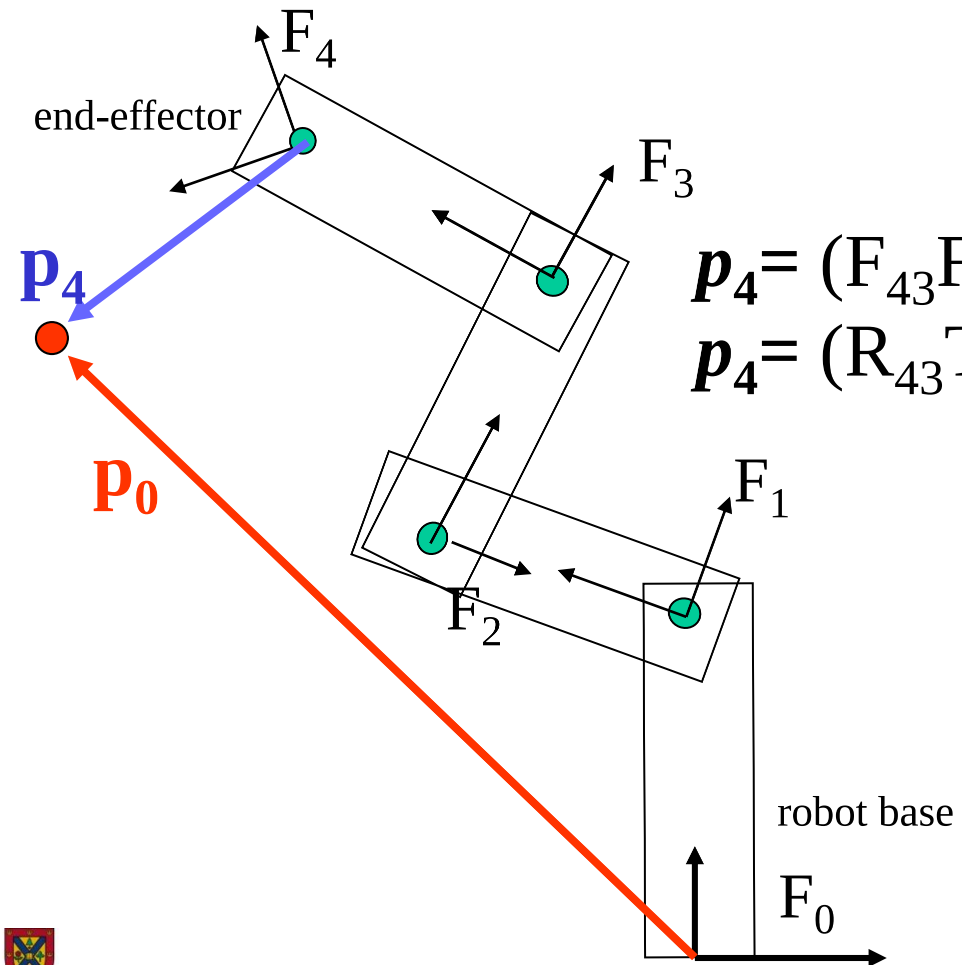
$$F^{-1}_{a\leftarrow c} = F_{c\leftarrow a}$$



Observation (home) coordinate system

Example of series of moving frame

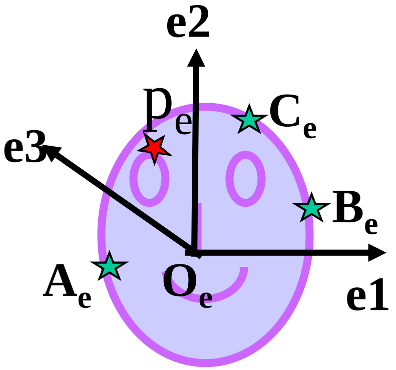
Four-joint serial robot, with rotation in each joint and constant translation between each two joints. We want to know the coordinates of a target point in end-effector frame if we know the target point in base frame.

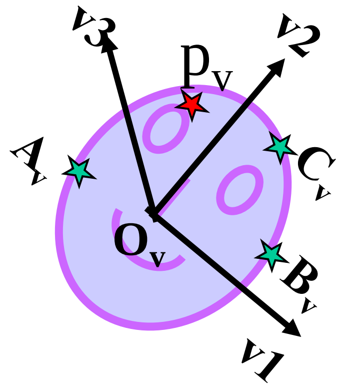


$$\boldsymbol{p_4} = (F_{43}F_{32}\ F_{21}F_{10}\ )\ \boldsymbol{p_0}$$

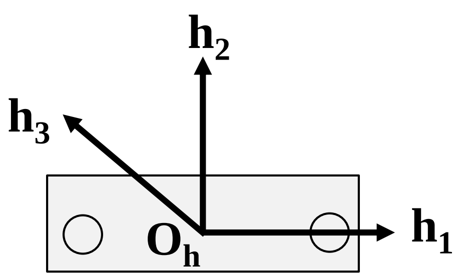
$$\boldsymbol{p_4} = (R_{43}T_{43}\ R_{32}T_{32}\ R_{21}T_{21}\ R_{01}T_{10}\ )\boldsymbol{p_0}$$

2. Multiple observations of a moving body





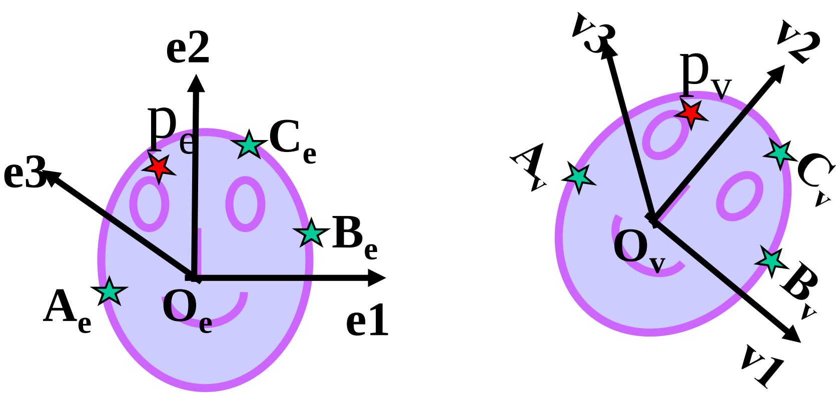
- A rigid body (such as the head) is moving during surgery
- ABC markers are affixed to the rigid head
- The markers are continuously localized (such as with optical stereo camera) in a home coordinate system
- We want to know the transformation that takes an arbitrary point of the head from pose-e (P_e) to pose v (P_v) .
- This transformation will be expressed as a series of translations and rotations…
- In each pose, we compute the frame defined by the ABC markers:
  - e_1, e_2, e_3, O_e
  - v_1, v_2, v_3, O_v



Observation (home) coordinate system

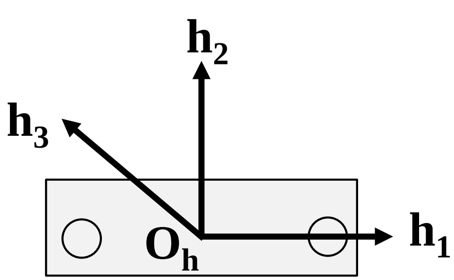
Typically in the home: $\mathbf{O_h} = (0,0,0)$, $\mathbf{h_1} = (1,0,0)$, $\mathbf{h_2} = (0,1,0)$, $\mathbf{h_3} = (0,0,1)$

2. Multiple observations of a moving body (cont'd)



Then we compute the following:

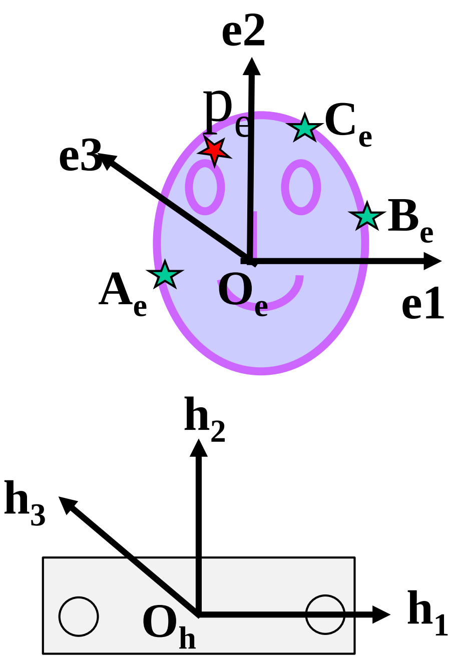
- $T_{h\leftarrow e}$ translation that takes the O_e center of the e frame to the O_h center of the home frame
- $T_{h\leftarrow v}$ translation that takes the O_v center of the v frame to the O_h center of the home frame.
- $R_{h\leftarrow e}$ rotation that rotates the e base vectors to home base vectors.
- $R_{h\leftarrow v}$ rotation that rotates the v base vectors to home base vectors.
- Then we can write:
- $R_{h\leftarrow v}\ T_{h\leftarrow v}\, p_v = R_{h\leftarrow e}\ T_{h\leftarrow e}\ p_e$
- $p_v = T_{v\leftarrow h} R_{v\leftarrow h}\ R_{h\leftarrow e} T_{h\leftarrow e}\ p_e$
- $R_{v\leftarrow e}$
- $p_v = F_{\ v\leftarrow e}\ p_e$



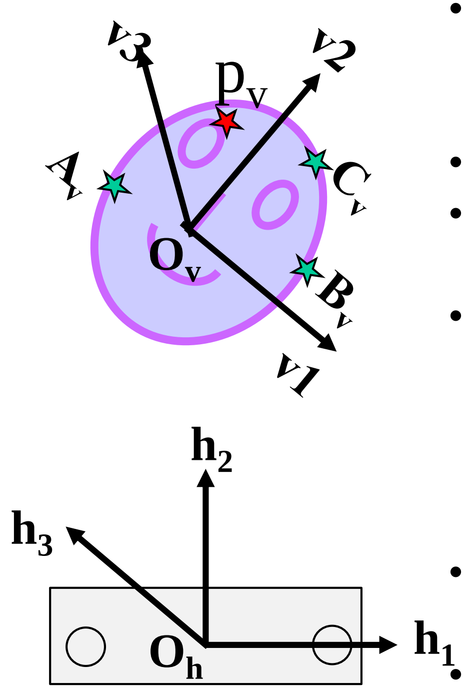
Observation (home) coordinate system

Typically in the home: $\mathbf{O_h} = (0,0,0)$, $\mathbf{h_1} = (1,0,0)$, $\mathbf{h_2} = (0,1,0)$, $\mathbf{h_3} = (0,0,1)$

3. Transform a body between two homes



Home1

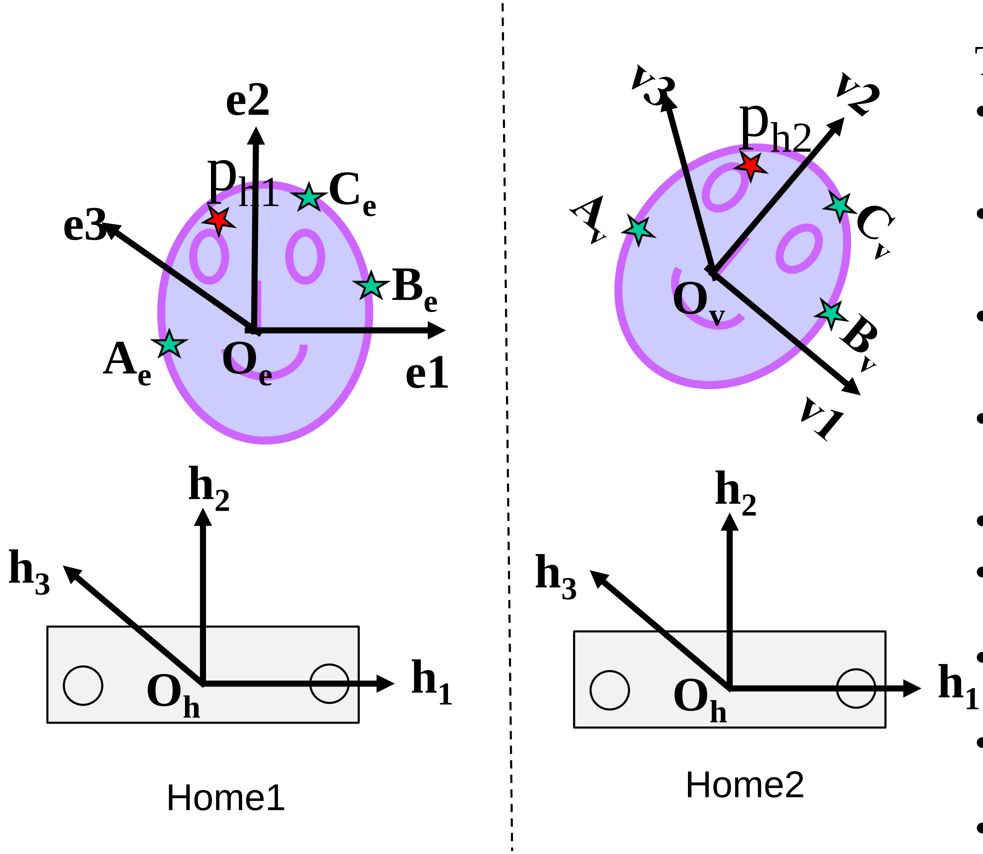


Home2

- A rigid body (such as the head) is observed in two home frames (such as two different imaging modalities at different times)
- ABC markers are affixed to the rigid head
- The markers are localized in each home frame
- We want to know the transformation that takes an arbitrary point of the head from one home to another. (Say, because we want to merge different imaging modalities of the same head.)
- This transformation will be expressed as a series of translations and rotations…
- In each home, we compute the frame defined by the ABC markers:
  - e_1, e_2, e_3, O_e
  - v_1, v_2, v_3, O_v

Typically in each home: $\mathbf{O_h} = (0,0,0)$, $\mathbf{h_1} = (1,0,0)$, $\mathbf{h_2} = (0,1,0)$, $\mathbf{h_3} = (0,0,1)$

3. Transform a body between two homes (cont'd)

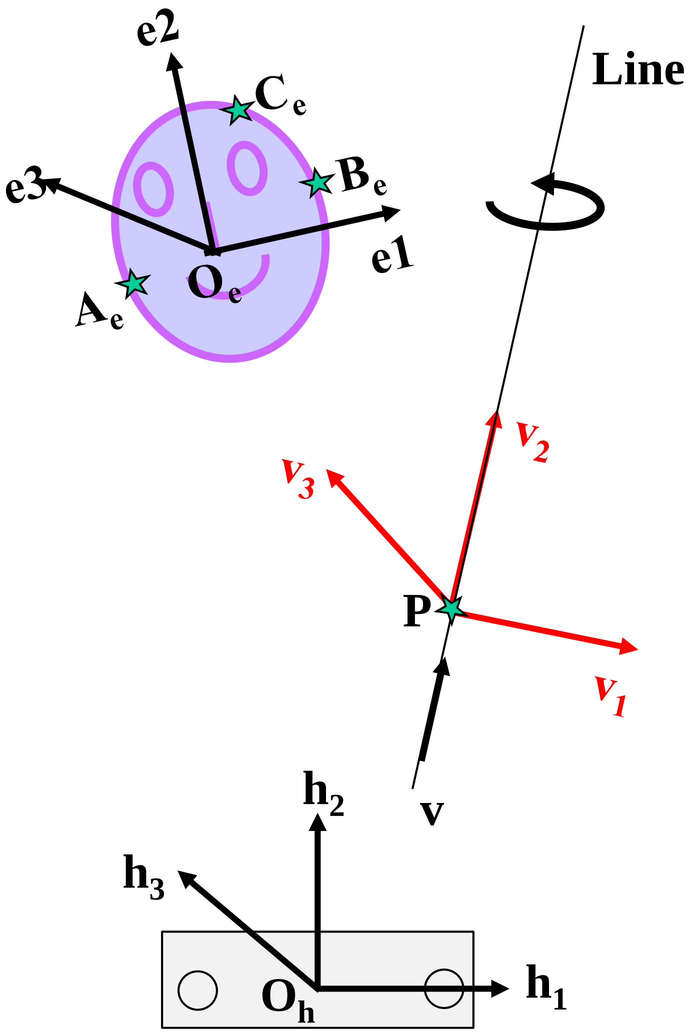


Then we compute the following:

- $T_{h1\leftarrow e}$ translation that takes the O_e center of the e frame to the O_h center of the home1 frame
- $T_{h2\leftarrow v}$ translation that takes the O_v center of the v frame to the O_h center of the home2 frame.
- $R_{h1\leftarrow e}$ rotation that rotates the e base vectors to home2 base vectors.
- $R_{h2\leftarrow v}$ rotation that rotates the v base vectors to home2 base vectors.
- Then we can write:
- $R_{h2\leftarrow v}\ T_{h2\leftarrow v}\ p_{h2} = R_{h1\leftarrow e}\ T_{h1\leftarrow e}\ p_{h1}$
- $p_{h2} = \underbrace{T_{v\leftarrow h2}\ R_{v\leftarrow h2}\ R_{h1\leftarrow e}\ T_{h1\leftarrow e}}_{F_{h2\leftarrow h1}}\ p_{h1}$
- $p_{h2} = F_{h2\leftarrow h1}\ p_{h1}$

Typically in each home: $\mathbf{O_h} = (0,0,0)$, $\mathbf{h_1} = (1,0,0)$, $\mathbf{h_2} = (0,1,0)$, $\mathbf{h_3} = (0,0,1)$

Rotation about an arbitrary line



Observation (home) coordinate system

- The rigid body is defined by ABC markers
- The line is defined by P fixed point and v direction vector
- We want to rotate the body by some angle about the line.

Method:

- Create a temporary frame, called line frame, that is centered in P on the line and in which one of the (v_1,v_2,v_3) base vectors is identical with the v direction vector of the line. (In the example we used v_2)
- Note that v_1,v_2 can be anywhere, so long (v_1,v_2,v_3) remains an orthonormal base !!!
- Transform the body from home frame to line frame
- In line frame, rotate the body about v_2 principal axis
- Transform the body from line frame back to home frame