**points vs vectors**

- we will think of vectors as motion between points $\vec{v}$
  - lives in a linear space $R^3$
  - addition and scalar multiplication have meaning
  - zero vector is no motion
  - cannot really translate motion
- we will think of points as a places $\tilde{p}$
  - lives in an affine space $A^3$
  - addition and scalar mul don't make sense
  - zero doesn't make sense
  - subraction does make sense, gives us motion

$$\tilde{p} - \tilde{q} = \vec{v}$$

  - Conversely point and motion gives a point

$$\tilde{q} + \vec{v} = \tilde{p}$$

**frames**

- basis is three vectors

$$\vec{v} = \sum_i c_i \vec{b}_i$$

- for affine space we will a frame
  - start with origin point $\tilde{o}$,
  - adding to it a linear combination of vectors using coordinates $c_i$

$$\tilde{p} = \tilde{o} + \sum_i c_i \vec{b} \quad = \quad \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} = \vec{\mathbf{f}}^t \mathbf{c}$$

**homogeneous coordinates**

- point is specified with homogeneous coordinate vector
  - four numbers
  - last one is always 1
- (if last coordinate is 0, we get a vector)

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 0 \end{bmatrix} = \vec{v}$$

**affine matrices**

- a we will call a matrix an "affine matrix" if it is of the form

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- we perform an affine transformation on a point by placing an affine matrix between a frame and a homogeneous coordinate vector, just like with linear transforms

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- the data types work iff the 4th row is $[0, 0, 0, 1]$

**Linear transforms**

- suppose i transform using a matrix of the form:

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

- this has the same effect on the coordinates as a 3by3 matrix multiply
- as if we applied a linear transform on the vectors connecting the origin to the point
- so we can use this to, say rotate a point about the origin

**translations**

- suppose i transform using a matrix of the form:

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

- we get A 3D translation by $[t_x, t_y, t_z]^t$

$$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \\ z' &= z + t_z \end{aligned}$$

- (a vector cannot be translated)

**rotations and translations**

- we can rotate point about the frame using matrix

$$R = \begin{bmatrix} k_x^2 v + c & k_x k_y v - k_z s & k_x k_z v + k_y s & 0 \\ k_y k_x v + k_z s & k_y^2 v + c & k_y k_z v - k_x s & 0 \\ k_z k_x v - k_y s & k_z k_y v + k_x s & k_z^2 v + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- composion of translation and rotation matrix gives us

$$TR = \begin{bmatrix} k_x^2 v + c & k_x k_y v - k_z s & k_x k_z v + k_y s & t_x \\ k_y k_x v + k_z s & k_y^2 v + c & k_y k_z v - k_x s & t_y \\ k_z k_x v - k_y s & k_z k_y v + k_x s & k_z^2 v + c & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- this is called a rigid body transform.
- so it is easy to factor a rbt matrix $M = TR$.

**frame is important**

- given point and matrix is not enough to specify mapping
- for example point $\tilde{p}$ and the matrix

$$\mathbf{S} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- the matrix is non-uniform scaling
- fix a frame $\vec{\mathbf{f}}^{\mathbf{t}}$
- in this frame $\tilde{p} = \vec{\mathbf{f}}^t \mathbf{c}$
- transform with matrix $\vec{\mathbf{f}}^{\mathbf{t}} \mathbf{c} \Rightarrow \vec{\mathbf{f}}^{\mathbf{t}} \mathbf{S} \mathbf{c}$
- the stretches by factor of two in direction $\vec{f_1}$

**other frame**

- pick some other frame $\vec{\mathbf{a}}^t$.
- relationship between bases $\vec{\mathbf{a}}^{\mathbf{t}} = \vec{\mathbf{f}}^{\mathbf{t}} \mathbf{M}$.
- express same point as $\tilde{p} = \vec{\mathbf{f}}^{\mathbf{t}} \mathbf{c} = \vec{\mathbf{a}}^{\mathbf{t}} \mathbf{M}^{-1} \mathbf{c} = \vec{\mathbf{a}}^{\mathbf{t}} \mathbf{d}$,
- use matrix $S$ we get $\vec{\mathbf{a}}^{\mathbf{t}} \mathbf{d} \Rightarrow \vec{\mathbf{a}}^{\mathbf{t}} \mathbf{S} \mathbf{d}$.
  - (equiv statement $\vec{\mathbf{a}}^{\mathbf{t}} \mathbf{M}^{-1} \mathbf{c} \Rightarrow \vec{\mathbf{a}}^{\mathbf{t}} \mathbf{S} \mathbf{M}^{-1} \mathbf{c}$.)
- the same point $\tilde{p}$ is stretched about direction $\vec{a}_1$

**SO**

- in general, if $\tilde{p} = \vec{\mathbf{a}}^t B D \mathbf{c}$
- then $\vec{\mathbf{a}}^t B Q D \mathbf{c}$ is the result of doing "$Q$ to $\tilde{p}$ wrt $\vec{\mathbf{a}}^t B$
- the point is transformed with respect to the the frame that appears immediately to the left of transformation matrix in the expression.
- call it the "left of" rule.

- important: same rule used for transformations of frames
- in general, if $\vec{\mathbf{f}}^t = \vec{\mathbf{a}}^t BD$
- then $\vec{\mathbf{a}}^t BQD$ is the result of doing "$Q$ to $\vec{\mathbf{f}}^t$ wrt $\vec{\mathbf{a}}^t B$
- nb: our named matrices (say rotations) only do the expected thing when applied wrt an orthonormal frame.
  - so this will impact how we do non-uniform scales for our robots.

**tforms using auxiliary basis**

- we may want to tform a frame $\vec{\mathbf{f}}^t$ using lets say a rotation $R$, with respect to some auxiliary frame $\vec{\mathbf{a}}^t$.
- given

$$\vec{\mathbf{f}}^{\mathbf{t}}\mathbf{M} = \vec{\mathbf{a}}^{\mathbf{t}}$$

- The transformed coordinate system can then be expressed as

$$\begin{aligned} & \vec{\mathbf{f}}^{\mathbf{t}} \\ = & \ \vec{\mathbf{a}}^{\mathbf{t}}\mathbf{M}^{-\mathbf{1}} \\ \Rightarrow & \ \vec{\mathbf{a}}^{\mathbf{t}}\mathbf{R}\mathbf{M}^{-\mathbf{1}} \\ = & \ \vec{\mathbf{f}}^{\mathbf{t}}\mathbf{M}\mathbf{R}\mathbf{M}^{-\mathbf{1}} \end{aligned}$$

- using rewrites, transform using "left of" and more rewrites

**multiple transformations**

- using the "left of" rule
- example:
  - a rotation matrix $R$ rotating a point by $\theta$ degrees about origin
  - translation matrix $T$, translating the point by one unit in the direction of the first frame axis.
- given transform

$$\vec{\mathbf{f}}^t \Rightarrow \vec{\mathbf{f}}^t RT$$

**first way**

- break into 2 steps
- first just use $R$

$$\vec{\mathbf{f}}^t \Rightarrow \vec{\mathbf{f}}^t R = \vec{\mathbf{f}'}^t$$

- the frame $\vec{\mathbf{f}}^t$ is rotated about the frame $\vec{\mathbf{f}}^t$.
- The resulting frame can also be expressed as $\vec{\mathbf{f}'}^t$.
- then this resulting frame is now transformed using $T$

$$\vec{\mathbf{f}}^t R \Rightarrow \vec{\mathbf{f}}^t RT$$

- which is the same as

$$\vec{\mathbf{f}'}^t \Rightarrow \vec{\mathbf{f}'}^t T$$

- using "left of" rule, translation is done along the direction of the first frame axis of $\vec{\mathbf{f}'}^t$.

**other way**

- can interpret another way
- first apply the translation

$$\vec{\mathbf{f}}^{\mathbf{t}} \Rightarrow \vec{\mathbf{f}}^{\mathbf{t}}\mathbf{T}$$

- the frame $\vec{\mathbf{f}}^t$ is first translated one unit in the direction of the first basis axis of $\vec{\mathbf{f}}^t$.
- then rotate the resulting frame

$$\vec{\mathbf{f}}^{\mathbf{t}}\mathbf{T} \Rightarrow \vec{\mathbf{f}}^{\mathbf{t}}\mathbf{RT}$$

- $R$ is placed with $\vec{\mathbf{f}}^t$ immediately to its left,
  - so the rotation with respect to $\vec{\mathbf{f}}^t$.

**summary**

- first interp
  - Rotate with repsect to $\vec{\mathbf{f}}^t$
  - translate wrt $\vec{\mathbf{f}'}^t = \vec{\mathbf{f}}^t R$
  - left to right, wrt latest in reading (local)
- second interp
  - Translate wrt $\vec{\mathbf{f}}^t$
  - rotate wrt $\vec{\mathbf{f}}^t$
  - right to left, wrt original frame in reading (global)