Chapter 2
Math Fundamentals

Part 1

2.1 Conventions and Definitions
2.2 Matrices
2.3 Fundamentals of Rigid Transforms

Carnegie Mellon
THE ROBOTICS INSTITUTE

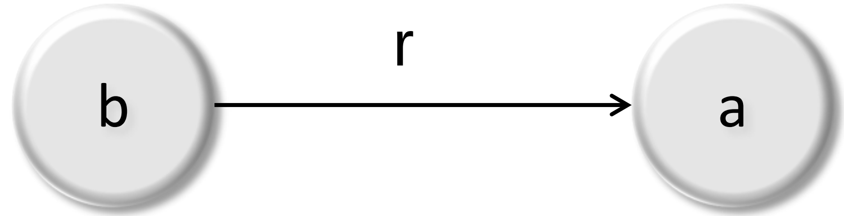
Outline

- 2.1 Conventions and Definitions
- 2.2 Matrices
- 2.3 Fundamentals of Rigid Transforms
- Summary

Outline

- 2.1 Conventions and Definitions
  - 2.1.1 Notational Conventions
  - 2.1.2 Embedded Coordinate Frames
- 2.2 Matrices
- 2.3 Fundamentals of Rigid Transforms
- Summary

Physical Quantities

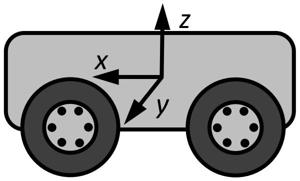


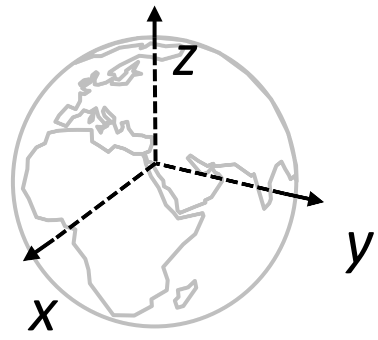
- Mechanics is about properties of / relations between objects.
- a is "r-related" to b
- r property of a relative to b r_a^b
- Example velocity (v) of robot (r) relative to earth (e): v_r^e
- Relationship is directional and (often) asymmetric. $r_a^b \neq r_b^a$

Properties ?

- “r’ is not quite a property of a.
  - “the” velocity of an object is not defined.
- It’s a property of a *relative* to b.
- a and b are real objects.
- In rare instances, we do not need a b.
  - unit vectors always of length 1.

$$r_a^b$$





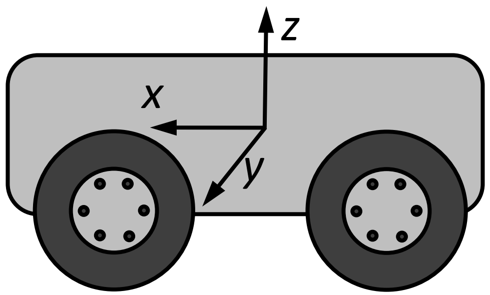
Vectors, Matrices, and Tensors

- With one exception (e.g. c_{light}) all require a datum (def'n of zero).
- May be scalars (density), vectors (velocity), tensors (_?_).
  - All are tensors of varying order.
- We write:

$$\rho_{ball} \qquad \vec{r}_{ball} \quad \vec{v}_{ball} \quad \vec{a}_{ball} \quad I_{ball}$$

- The vectors at least can be of 1, 2, or 3 dimensions.

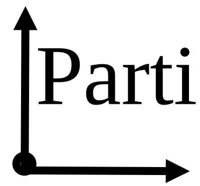
Frames of Reference and Coordinate Systems



- Objects of interest are real: wheels, sensors, obstacles.
- Abstract them by sets of axes <u>fixed to the body</u>.
- These axes:
  - Have a state of motion ⟵ Reference Frame
  - Can be used to express vectors. ⟵ Coordinate System
- Call them coordinate frames.

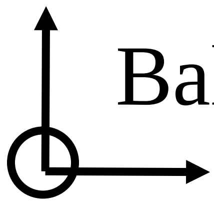
Coordinate Frames

- Points possess position but not orientation:



Particle (orientation undefined)

- Rigid Bodies possess position and orientation:



Ball (can rotate)

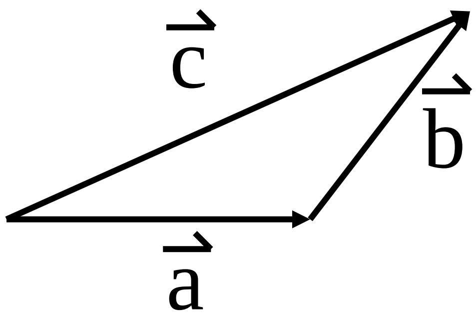
- A rigid body:
  - does not have one position.
  - does have one orientation

Outline

- 2.1 Conventions and Definitions
  - 2.1.1 Notational Conventions
  - 2.1.2 Embedded Coordinate Frames
- 2.2 Matrices
- 2.3 Fundamentals of Rigid Transforms
- Summary

Vectors and Coordinates

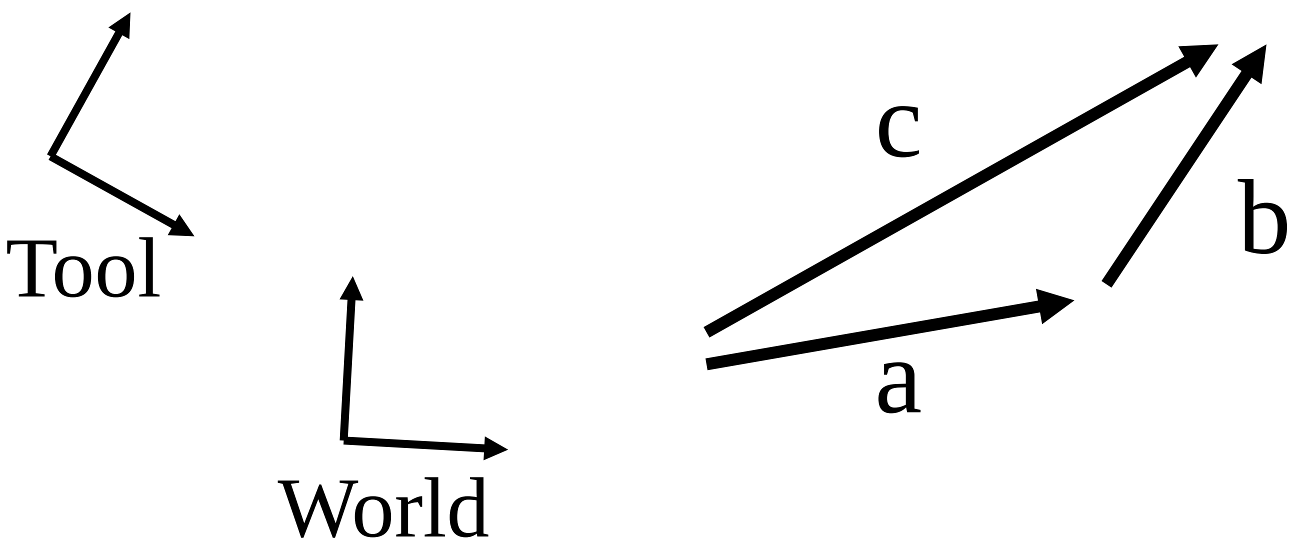
- Many laws of physics relate vectors and hold regardless of coordinate system.
- Notation:
  - $\underline{r}$ is expressed in some coordinate system.
  - $\vec{r}$ is coordinate system independent.



Vectors and Coordinates

- Vectors of physics are coordinate system independent:

$$\vec{c} = \vec{a} + \vec{b}$$



- Addition is defined geometrically.

Vectors and Coordinates

- Vectors of linear algebra are coordinate system dependent:

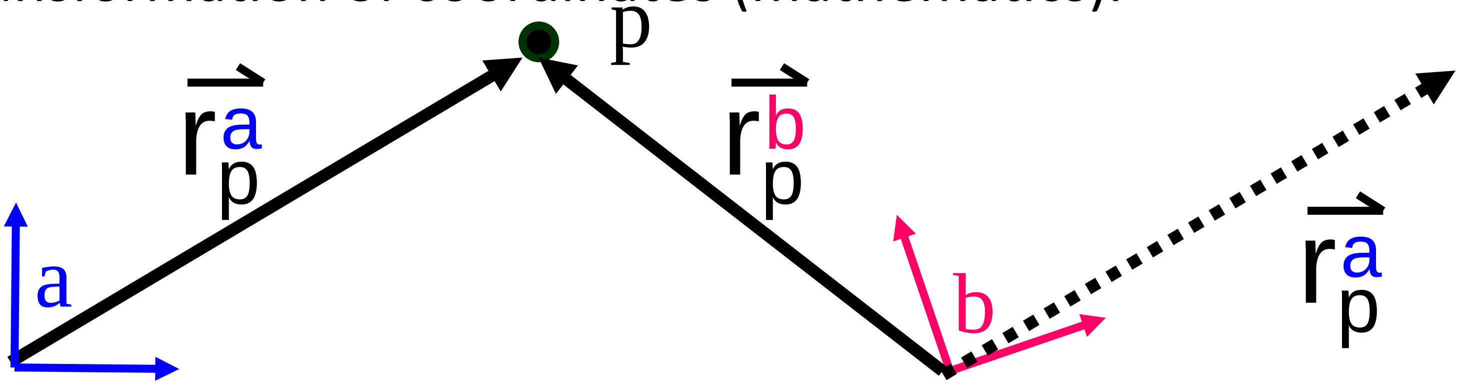
$$\underline{c} = \underline{a} + \underline{b}$$

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

- Addition is defined algebraically.
- A relation to physical vectors (directed line segments) requires a coordinate system.

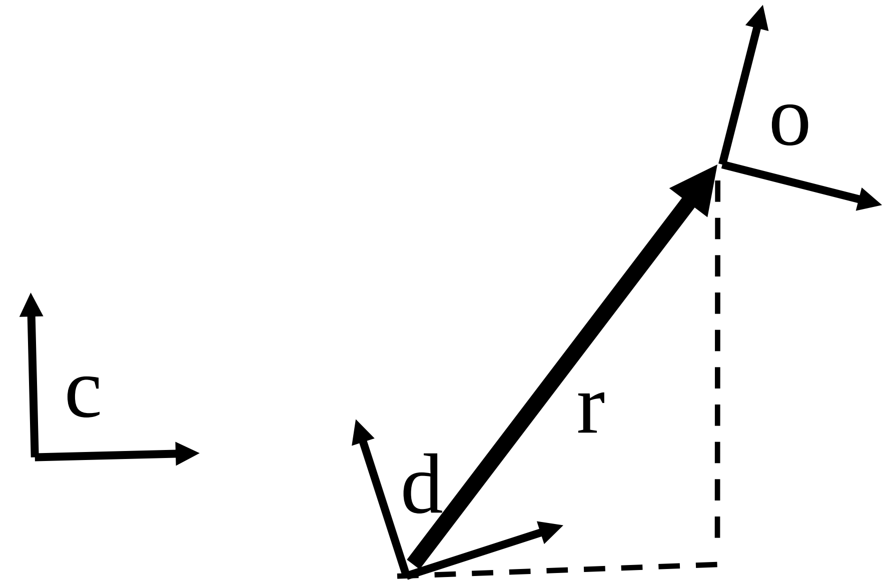
Free and Bound Transformations

- Distinguishes what happens when the reference frame changes.
- <u>Bound</u>: the vector may change and its expression may change.
  - Transformation of frame of reference (physics).
- <u>Free</u>: the vector remains the same and its expression may change
  - Transformation of coordinates (mathematics).



Notational Conventions

$$ {}^{c}r_{o}^{d} $$



- r → relationship / property
- o → object to which property is attributed
- d → object serving as datum
- c → object providing the coordinate system

Box 2.1 Notation for Physical Quantities

- $\underline{r}_a$: the r property of a expressed in the default coordinate system associated with object a.
- $\vec{r}_a^{\,b}$: the r property of a relative to b in coordinate system independent form.
- $\underline{r}_a^{\,b}$: the r property of a relative to b expressed in the default coordinate system associated with object b.
- ${}^{c}\underline{r}_a^{\,b}$: the r property of a relative to b expressed in the default coordinate system associated with object c.

Box 2.1: Notation for Physical Quantities

We will use the following conventions for specifying physical quantities:

- r_a denotes the scalar r property of object a.
- r_n denotes the scalar n-th component of a vector or the n-th entity in a sequence.
- r_{ij} denotes the scalar ij-th component of a matrix or the ij-th entity in a sequence of order 2.
- $\vec{r}_a$ denotes the vector r property of object a expressed in coordinate system independent form.
- $\underline{r}_a$ denotes the vector r property of object a expressed in the default coordinate system associated with object a. Thus $\underline{r}_a = {}^a\underline{r}_a$.
- $\vec{r}_a^{\,b}$ denotes the vector r property of a relative to b in coordinate system independent form.
- $\underline{r}_a^b$ denotes the vector r property of a relative to b expressed in the default coordinate system associated with object b. Thus $\underline{r}_a^b = {}^b\underline{r}_a^b$.
- R_a^b denotes the matrix R property of a relative to b expressed in the default coordinate system associated with object b.
- ${}^c\underline{r}_a^b$ denotes the vector r property of a relative to b expressed in the default coordinate system associated with object c.

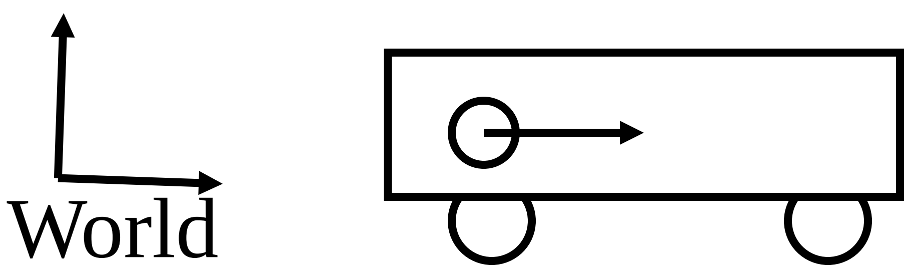
Sub/Super Scripts – Physics Vectors

- Leading <u>sub</u>scripts denote the frame/ object possessing the vector quantity:

$$\vec{v}_{ball}$$

- Leading <u>super</u>scripts denote the frame/object with respect to which the quantity is measured (i.e. the datum):

$$\vec{v}_{ball}^{world} = \vec{v}_{ball}^{train} + \vec{v}_{train}^{world}$$



Sub/SuperScripts – LA Vectors

- Leading <u>sub</u>scripts denote the object frame possessing the quantity:

$$\underline{v}_{wheel}$$

- Leading <u>super</u>scripts denote the datum (also the implied coordinate system within which the quantity is expressed).

$$^{world}\underline{v}_{wheel}$$

- Trailing superscripts denote the coordinate system, and leading denotes datum when necessary.

$$^{body}\underline{v}^{wheel}_{world}$$

Notational Conventions

- Position vectors:

$$r = \left[x\ y\ z\right]^T$$

- Also sometimes as $\underline{r}$ or as $\vec{r}$ to emphasize it is a vector.
- Matrices:

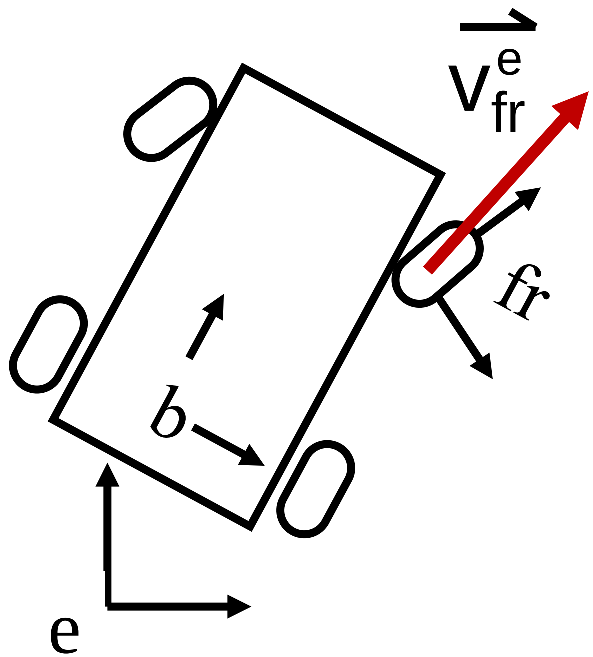
$$T = \begin{bmatrix} t_{xx} & t_{xy} & t_{xz} \\ t_{yx} & t_{yy} & t_{yz} \\ t_{zx} & t_{zy} & t_{zz} \end{bmatrix}$$

Why All The Fuss ?

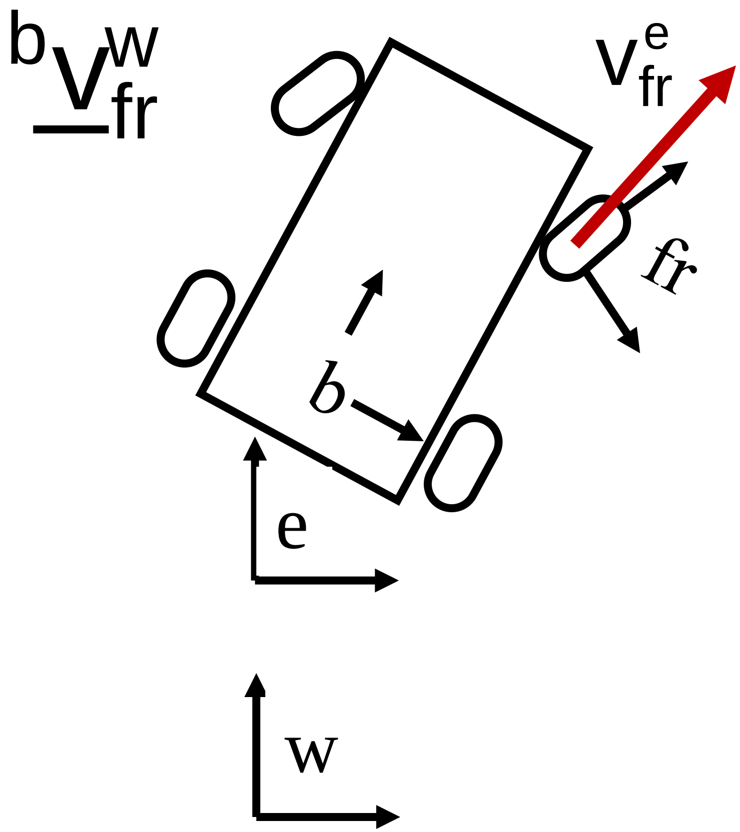
- Accelerometer: acceleration of the sensor wrt inertial space: $\vec{a}_s^i$
- Strapdown: acceleration of the sensor wrt inertial space referred to body coordinates: ${}^b\underline{a}_s^i$
- Nav Solution: Acceleration of the body wrt earth referred to earth coordinates: ${}^e\underline{a}_b^e$

Why All The Fuss ?

- WMR kinematics are much easier to do in the body frame.
- Velocity of the front right wheel wrt the earth ("world") frame: $\vec{v}_{fr}^{e}$
- Velocity of front right wheel wrt earth referred to body coordinates: ${}^{b}\underline{v}_{fr}^{e}$



Why All The Fuss ?



- Coordinate system may be unrelated to either the object or the datum.
  - So you need a third symbol to be precise.

Converting Coordinates

$$r^b = T_a^b r^a$$

- We will see later that T_a^b notation satisfies our conventions where it means the ‘T’ property of ‘object’ a wrt ‘object’ b.

Outline

- 2.1 Conventions and Definitions
- 2.2 <u>Matrices</u>
- 2.3 Fundamentals of Rigid Transforms
- Summary

Tensors

- For our purposes, these are multidimensional arrays:
- Consider T(I,j,k) to be a 3D "box" of numbers.
- Suppose it is 3X3X3, then there are three "slices" extending out of the page.

$$T_1 = \begin{bmatrix} 2 & 3.6 & 7 \\ 8 & -4.3 & 0 \end{bmatrix} \qquad T_2 = \begin{bmatrix} 3 & -5 & 12 \\ 4 & 2 & -1 \end{bmatrix} \qquad T_3 = \begin{bmatrix} 7 & 9.2 & 18 \\ 8-4 & 0 & 13 \end{bmatrix}$$

$$T[2, 1, 3] = T[2][1][3] = t_{213} = 12$$

Some Notation

- Block Notation:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

- Tensor Notation:

$$A = \left[a_{ijk} \right]$$

  – [..] means a set that can be arranged in a rectangle that is ordered in each dimension.

Operations

- Vector dot product: $\underline{a} \cdot \underline{b} = \sum_k a_k b_k$
  - Also written
- Matrix multiplication: $\underline{a}^T \underline{b}$
  - Dot product of i-th row and j-th column $C = AB = [c_{ij}] = \left[\sum_k a_{ik} b_{kj}\right]$
- Cross product $\underline{c} = \underline{a} \times \underline{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$
  - Also written: $\underline{c} = \underline{a} \times \underline{b} = \underline{a}^X \underline{b}$
  - 'Skew" matrix $\underline{a}^X = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$

Operations

- 2.2.1.7 Outer Product:

$$\underline{c} = \underline{a}\underline{b}^T = \begin{bmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{bmatrix}$$

- 2.2.1.8 Block Multiplication:

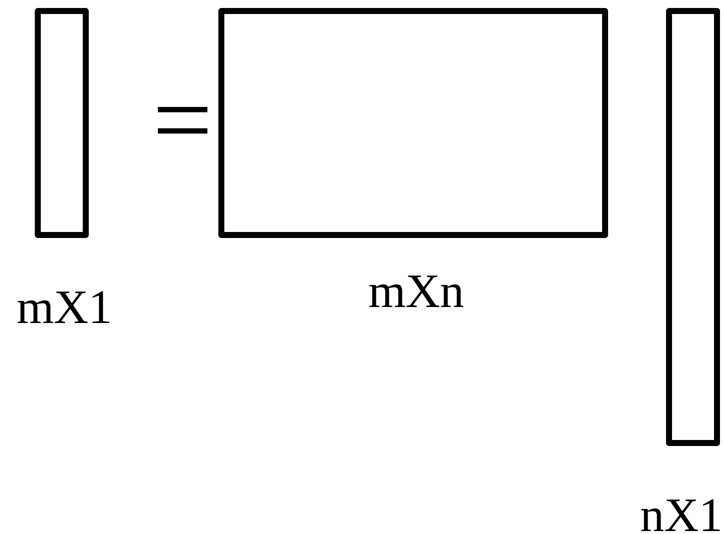
$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \qquad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

2.2.1.9 Linear Mappings

- Of course, this is:
  - "every element of $\underline{y}$ depends on every element of $\underline{x}$ in a linear manner".
- Two views:
  - "A operates on x": y is the list of projections of $\underline{x}$ on each row of A.
  - "x operates on A": y is a weighted sum of the columns of A. $\underline{x}$ is the weights.
- A turns $\underline{x}$ into $\underline{y}$ or $\underline{x}$ collapses A's rows to produce $\underline{y}$

$$\underline{y} = A\,\underline{x}$$



2.2.2 Matrix Functions …

- Function of a scalar:

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \dots \\ a_{21}(t) & a_{22}(t) & \dots \\ \dots & \dots & \dots \end{bmatrix} = \left[a_{ij}(t)\right]$$

- Function of a vector:

$$A(\underline{x}) = \begin{bmatrix} a_{11}(\underline{x}) & a_{12}(\underline{x}) & \dots \\ a_{21}(\underline{x}) & a_{22}(\underline{x}) & \dots \\ \dots & \dots & \dots \end{bmatrix} = \left[a_{ij}(\underline{x})\right]$$

Exponentiation

- Powers of matrices automatically commute:

$$A^3 = A(A^2) = (A^2)A = AAA$$

- Hence, we can define matrix “polynomials”:

$$Y = AX^2 + BX + C$$

- Not so useful in practice but:
  - $Y = aX^2 + bX + c$ (scalar coefficients) is super useful.

Arbitrary Functions of Matrices

- Recall the Taylor Series:

$$f(x) = f(0) + x\left\{\frac{df}{dx}\right\}_0 + \frac{x^2}{2!}\left\{\frac{d^2f}{dx}\right\}_0 + \frac{x^3}{3!}\left\{\frac{d^3f}{dx}\right\}_0 + \dots$$

- Taylor series for exponential function:

$$e^x = \exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

- Hence, <u>define</u> the matrix exponential as:

$$\exp(A) = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$$

2.2.3 Matrix Inversion & Inverse Mapping

- Matrix inverse defined s.t.:

$$A^{-1}A = I$$

- Therefore:

$$A^{-1}\underline{y} = A^{-1}A\underline{x} = \underline{x}$$

$$\boxed{\underline{x} = A^{-1}\underline{y}}$$

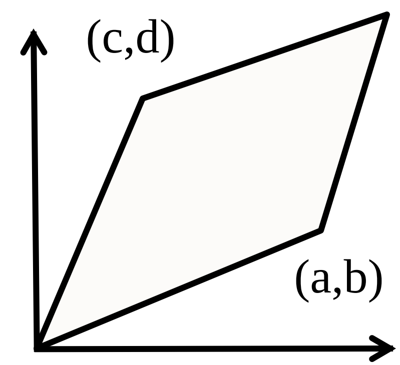
2.2.3.2 Determinant

- Scalar-valued function of a matrix.

$$\det(A) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - cb$$

- Matrix not invertible if distinct inputs map to same output.

- Determinant measures:
  - Volume spanned by rows of A.
  - Ratio of the volumes spanned by two input and the associated two output vectors.



2.2.3.3 Rank

- Rank = "dimension of largest invertible submatrix"
- Unlike determinant, defined for nonsquare matrices.
- A nonsquare matrix can have a rank no larger than the smaller of its two dimensions.
- The rank of a matrix product cannot exceed the minimum of the ranks of the two operands.
- For an mXn matrix A with (m <= n)
  - Rank = m → "is of full rank"
  - Rank < m → "is rank deficient"
- For an n X n matrix A:
  - Rank = n → "nonsingular", "invertible"
  - Rank < n → "singular", "noninvertible"

35

2.2.3.4 Positivity

- A square matrix A is called “positive definite” if:

$$\underline{x}^T A \underline{x} > 0 \qquad \forall \underline{x} \neq 0$$

- Matrix equivalent to positive scalars:
  - E.g. sum of two pos def. matrices is pos. def.
- Covariance, inertia, are always positive definite (in absence of bugs).
- $f(\underline{x})=\underline{x}^T A \underline{x}$ is a parabloid in n dimensions.

2.2.3.5 Homogeneous Linear Systems

- Special form of system:

$$A\underline{x} = \underline{0}$$

- Is called a homogeneous system.
- When A is nonsingular, the solution is, of course:

$$\underline{x} = A^{-1}\underline{0} = \underline{0}$$

- If A is singular, there are an infinite number of nonzero solutions and $\underline{x}$ is in the nullspace of A (see below for more on nullspace).

2.2.3.6 Eigenvalues and Eigenvectors

- The vector $\underline{e}$ is an eigenvector of A when it satisfies:

$$A\underline{e} = \lambda\underline{e}$$

- For some scalar λ called the eigenvalue associated with $\underline{e}$.
- To solve, rewrite first equation as:

$$(\lambda I \ - A)\underline{e} = \underline{0}$$

- Therefore, for this homogeneous system, nonzero $\underline{e}$ implies:

$$det(\lambda I \ - A) = 0$$

2.2.4 Subspaces

- Consider:

$$\underline{y} = A\underline{x} \;\; ; \;\; A \in \mathbb{R}^{m \times n}$$

- Range or columnspace of A, denoted $\mathrm{C}(A)$ is the set of all possible values for $\underline{y} \in \mathbb{R}^{m \times 1}$.
  - equivalently all possible linear combinations of columns of A
- The rowspace of A, denoted $\mathrm{R}(A)$ is the set of all vectors $\underline{x} \in \mathbb{R}^{n \times 1}$ for which $\underline{y} = A\underline{x} \neq \underline{0}$.
  - equivalently all possible linear combinations of the rowsof A.
- The nullspace of A, denoted $\mathrm{N}(A)$ is the set of all vectors $\underline{x} \in \mathbb{R}^{n \times 1}$ for which $\underline{y} = A\underline{x} = \underline{0}$.
  - Its rank is called nullity.

2.2.4 Rank Nullity Theorem

- Every vector in the rowspace is orthogonal to every vector in the nullspace:

$$R(A)\ \perp N(A)$$

- The union of these two subspaces of $\mathbb{R}^n$ is $\mathbb{R}^n$.

$$R(A) \cup N(A) = \mathbb{R}^n$$

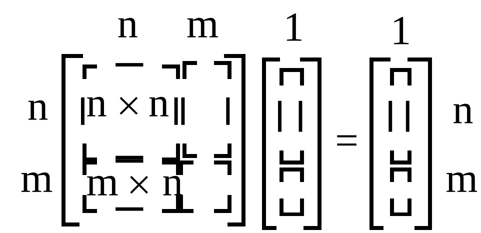
- The dimensions of these two sum to m:

$$rank(A) + nullity(A) = n$$

2.2.5.3 Blockwise Matrix Elimination

- Given:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} y_A \\ y_B \end{bmatrix}$$



- Assuming A is invertible, multiply first block row by CA^{-1}:

$$\begin{bmatrix} C & CA^{-1}B \\ C & D \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix} = \begin{bmatrix} CA^{-1}y_A \\ y_B \end{bmatrix}$$

- Subtract two blocks of m rows to produce:

$$(D - CA^{-1}B)x_B = y_B - CA^{-1}y_A$$

- Solve for x_B and substitute into original 1st equation to get x_A.

2.2.5.4 Matrix Inversion Lemma

- This is:

$$[A - BD^{-1}C]^{-1} = A^{-1} + A^{-1}B[D - CA^{-1}B]^{-1}CA^{-1}$$

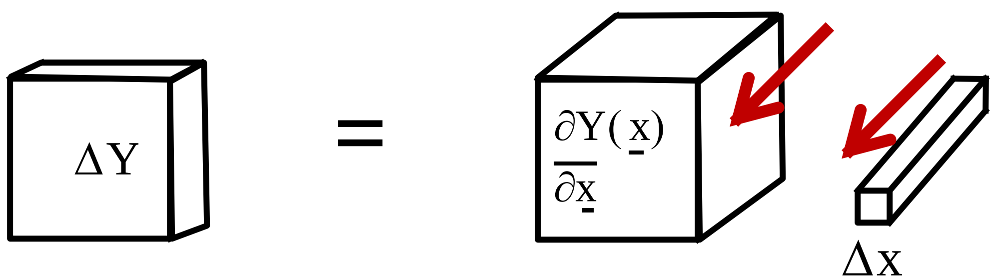
- Derived in the text using block matrix inversion.
- The matrix inversion on left is nXn. The one on the right is mXm.
  – Therefore, the inversion lemma is less work.
  – Often m=1 so its a lot less work.
  – Special case is the Sherman–Morrison formula often used to give a rank 1 update to an inverse.
- Kalman filter is based on this.

2.2.6.2 Expansion Operations

- Derivative of a matrix with respect to a vector is a 3rd order tensor.

$$\frac{\partial Y(\underline{x})}{\partial \underline{x}} = \left[\frac{\partial}{\partial x_k} y_{ij}(\underline{x})\right]$$

- Each k is a different matrix $[y_{ij}]$:
- Use this to perturb a matrix-valued function:



2.2.6.4 Product Rules - 1

- Derivative of a matrix product w.r.t a scalar:

$$\frac{\partial}{\partial x}C(x) = \frac{\partial}{\partial x}\{A(x)B(x)\} = \frac{\partial}{\partial x}\{A(x)\}B(x) + A(x)\frac{\partial}{\partial x}\{B(x)\}$$

- Example:

$$\frac{d\{\underline{x}(t)^T A \underline{x}(t)\}}{dt} = \dot{\underline{x}}^T A \underline{x} + \underline{x}^T A \dot{\underline{x}}$$

2.2.6.7 Product Rules - 2

- Derivative of a matrix product w.r.t a vector:

$$C(\underline{x}) = \frac{\partial\{A(\underline{x})B(\underline{x})\}}{\partial \underline{x}} = \frac{\partial}{\partial \underline{x}}A(\underline{x})B(\underline{x}) + A(\underline{x})\frac{\partial}{\partial \underline{x}}B(\underline{x})$$

- Examples:

$$\frac{\partial\{\underline{a}^T\underline{x}\}}{\partial \underline{x}} = \underline{a}^T \qquad \frac{\partial\{\underline{x}^T A\}}{\partial \underline{x}} = A^T$$

$$\frac{\partial\{\underline{x}\}}{\partial \underline{x}} = \frac{\partial\{I\underline{x}\}}{\partial \underline{x}} = I \qquad \frac{\partial\{\underline{x}^T\underline{a}\}}{\partial \underline{x}} = \underline{a}^T$$

$$\frac{\partial\{\underline{x}^T A\underline{x}\}}{\partial \underline{x}} = \underline{x}^T A^T + \underline{x}^T A \qquad \frac{\partial\{\underline{x}^T\}}{\partial \underline{x}} = \frac{\partial\{\underline{x}^T I\}}{\partial \underline{x}} = I$$

45

2.2.6.8 Names and Notation for Derivatives

Table 2.1: Notation for Derivatives

Symbol	Meaning	Symbol	Meaning
$\frac{\partial y}{\partial x}$, y_x	a partial derivative	$\frac{\partial Y}{\partial x}$, Y_x	a matrix partial derivative
$\frac{\partial \underline{y}}{\partial x}$, $\underline{y}_x$	a vector partial derivative	$\frac{\partial \underline{y}}{\partial \underline{x}}$, $\underline{y}_{\underline{x}}$	a Jacobian matrix
$\frac{\partial y}{\partial \underline{x}}$, $y_{\underline{x}}$	a gradient vector	$\frac{\partial Y}{\partial \underline{x}}$, $Y_{\underline{x}}$	an order 3 tensor
$\frac{\partial^2 y(\underline{x})}{\partial x^2}$, y_{xx}	a Hessian matrix		

Outline

- 2.1 Conventions and Definitions
- 2.2 Matrices
- <u>2.3 Fundamentals of Rigid Transforms</u>
  - <u>2.3.1 Definitions</u>
  - 2.3.2 Why Homogeneous Transforms
  - 2.3.3 Semantics and Interpretations
- Summary

2.3.1 Affine Transformation

- Most general linear transformation

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

- R's and t's are the transform constants
- Can be used to effect translation, rotation, scale, reflections, and shear.
- Preserves linearity but not distance (hence, not areas or angles).

2.3.1 Homogeneous Transformation

- Set t1 = t2 = 0:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \text{nothing}$$

Homogeneous

- r's are the transform constants
- Can be used to effect rotation, scale, reflections, and shear (not translation).
- Preserves linearity but not distance (hence, not areas or angles).

2.3.1 Orthogonal Transformation

- Looks the same …

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

but:

$$\begin{aligned} r_{11}r_{12} + r_{21}r_{22} &= 0 \\ r_{11}r_{11} + r_{21}r_{21} &= 1 \\ r_{12}r_{12} + r_{22}r_{22} &= 1 \end{aligned}$$

- But:
  - Can be used to effect rotation.
  - Preserves linearity and distance (hence, areas and angles).

Outline

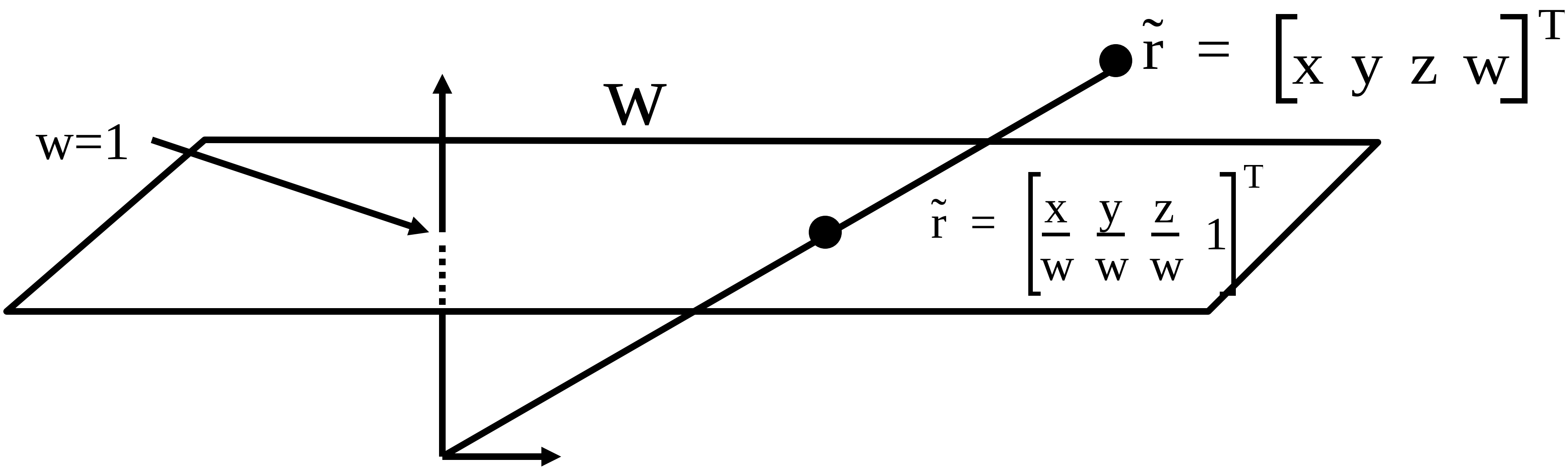
- 2.1 Conventions and Definitions
- 2.2 Matrices
- <u>2.3 Fundamentals of Rigid Transforms</u>
  - 2.3.1 Definitions
  - <u>2.3.2 Why Homogeneous Transforms</u>
  - 2.3.3 Semantics and Interpretations
- Summary

2.3.2 Homogeneous Coordinates

- Coordinates which are <u>unique up to a scale factor</u>. i.e

$$\underline{x} = 6\underline{x} = -12\underline{x} = 3.14\underline{x} = \text{same thing}$$

- The numbers in the vectors are not the same but we interpret them to mean the same thing (in fact, the thing whose scale factor is unity).



Pure Directions

- Its also possible to represent pure directions
  - Pure in the sense they "are everywhere" (i.e. have no position and cannot be moved).
- We use a scale factor of zero to get a pure direction:

$$d = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

- It will shortly be clear why this works.

Why Bother?

- Points in 3D can be rotated, reflected, scaled, and sheared with 3 X 3 matrices….

$$r' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = Tr = \begin{bmatrix} t_{xx} & t_{xy} & t_{xz} \\ t_{yx} & t_{yy} & t_{yz} \\ t_{zx} & t_{zy} & t_{zz} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} t_{xx}x + t_{xy}y + t_{xz}z \\ t_{yx}x + t_{yy}y + t_{yz}z \\ t_{zx}x + t_{zy}y + t_{zz}z \end{bmatrix}$$

- But not translated.

$$r' = r + \Delta r = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

- What 3X3 matrix is Trans(Δr)?

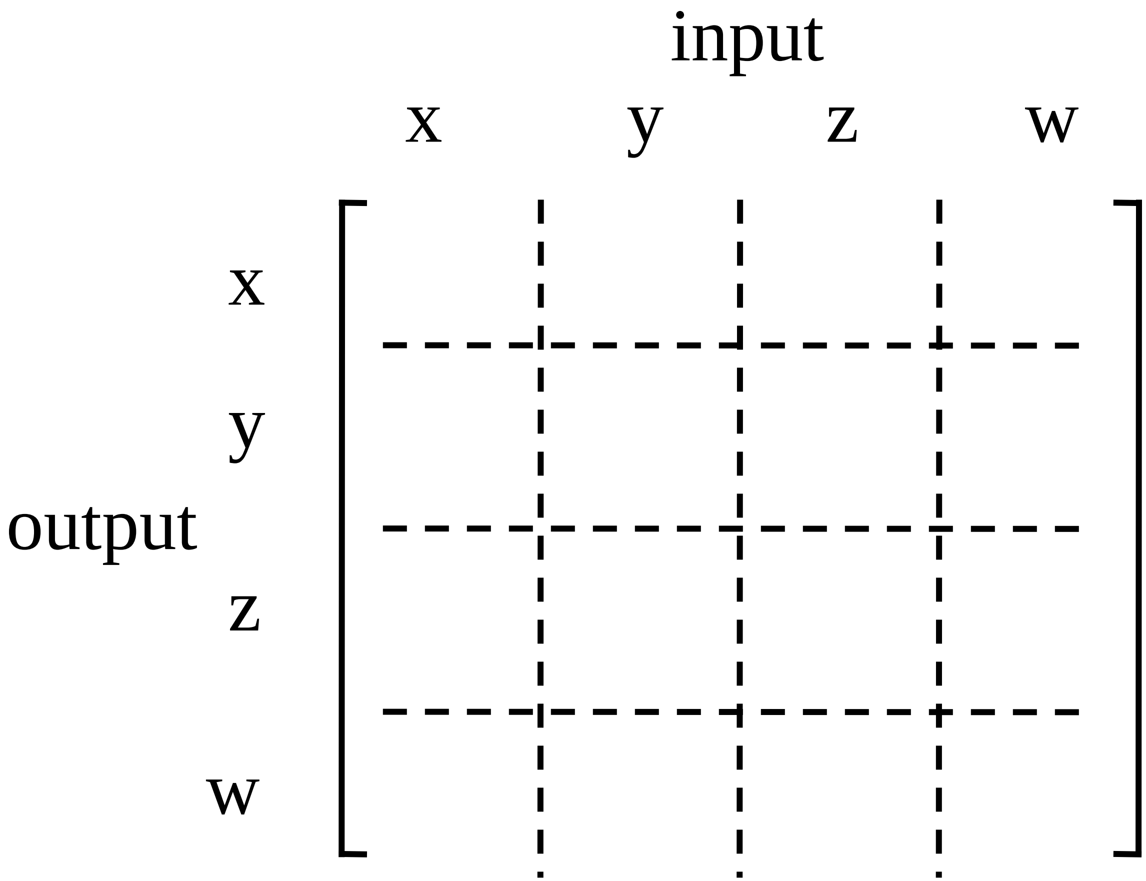
Trick: Move to 4D

$$r' = r + \Delta r = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathrm{Trans}(\Delta r) r$$

$$x = 1 \times x + \Delta x$$
$$y = 1 \times y + \Delta y$$
$$z = 1 \times z + \Delta z$$

- The scale factor in the vector is used to add a scaled amount of the 4th matrix column.

HT Matrix Format



Outline

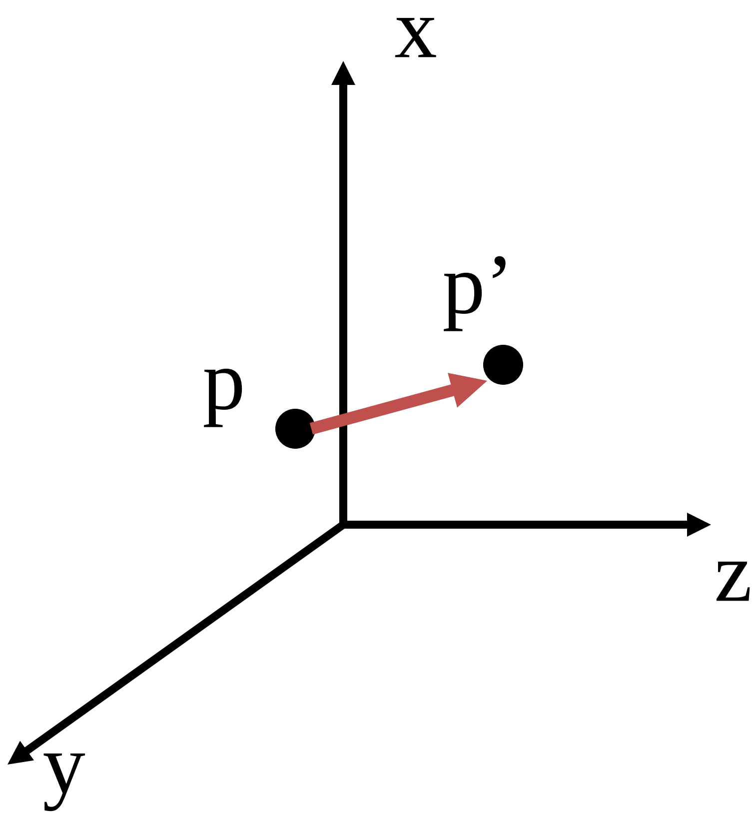
- 2.1 Conventions and Definitions
- 2.2 Matrices
- <u>2.3 Fundamentals of Rigid Transforms</u>
  - 2.3.1 Definitions
  - 2.3.2 Why Homogeneous Transforms
  - <u>2.3.3 Semantics and Interpretations</u>
- Summary

Trig Function Shorthand

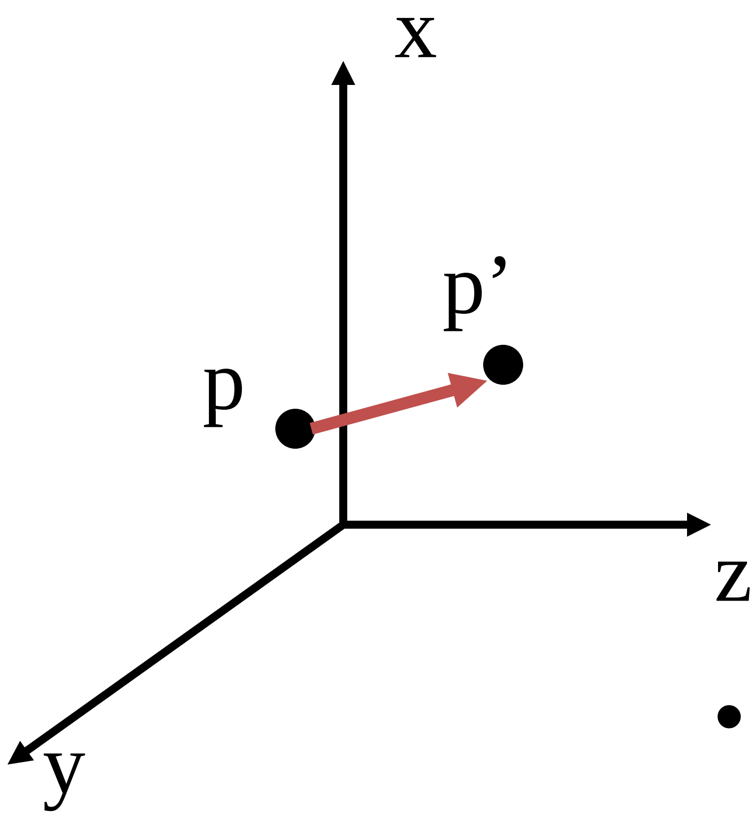
- $\sin(\theta) \rightarrow s\theta$
- $\cos(\theta) \rightarrow c\theta$
- $\tan(\theta) \rightarrow t\theta$
- $\sin(\theta_1)\cos(\theta_2) \rightarrow s\theta_1 c\theta_2 \rightarrow s1c2$
- $\sin(\theta_1+\theta_2) \rightarrow s\theta_1\theta_2 \rightarrow s12$

Operators

- Mapping:
  - Point → Point ’ (both expressed in same coordinates)



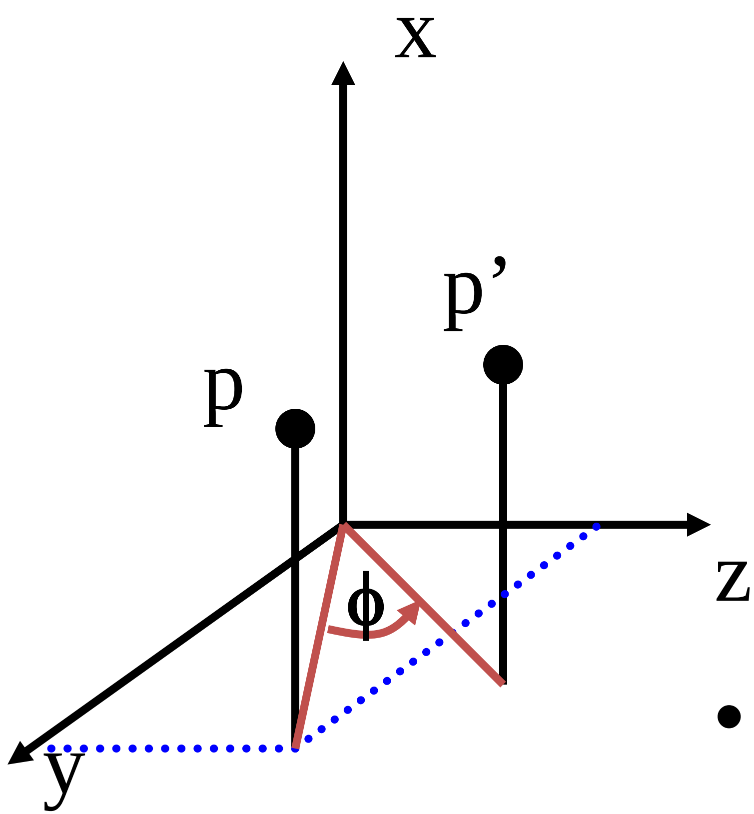
Operators



$$\text{Trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note <u>capital T</u> in Trans().

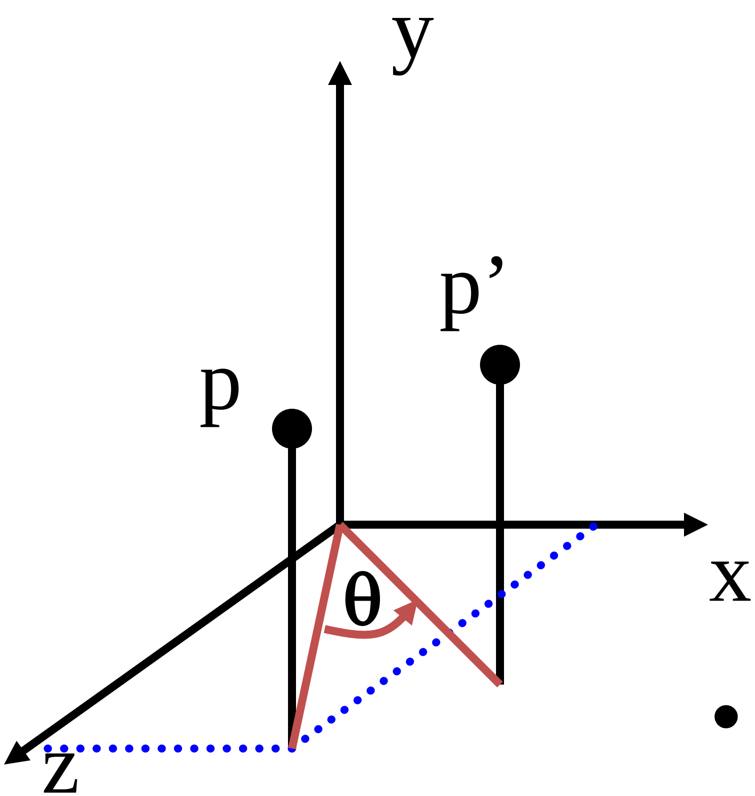
Operators



$$Rotx(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\phi & -s\phi & 0 \\ 0 & s\phi & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note <u>capital R</u> in Rotx().

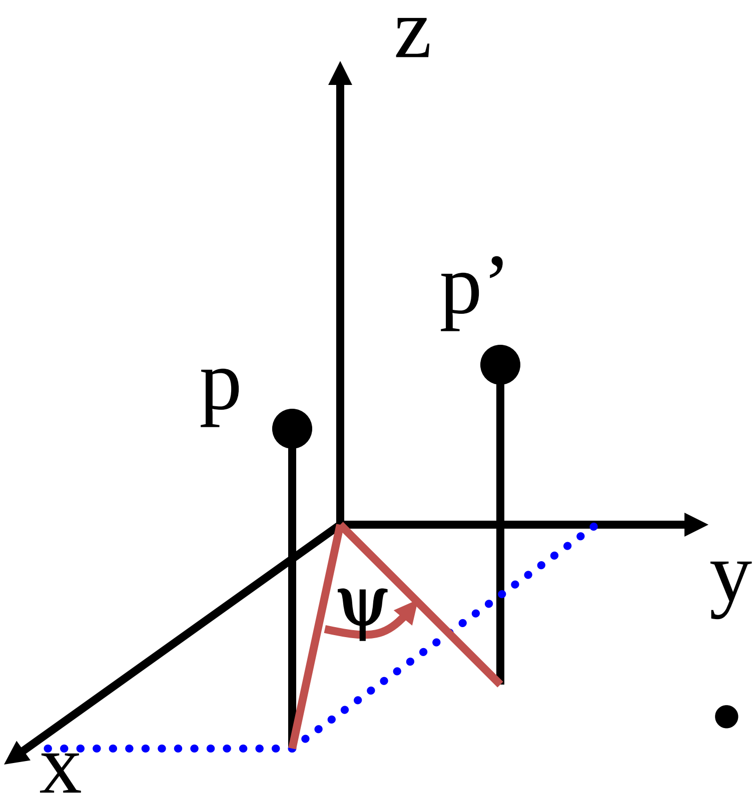
Operators



$$\text{Roty}(\theta) = \begin{bmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note <u>capital R</u> in Roty().

Operators



$$\mathrm{Rotz}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note <u>capital R</u> in Rotz().

Compound Operators

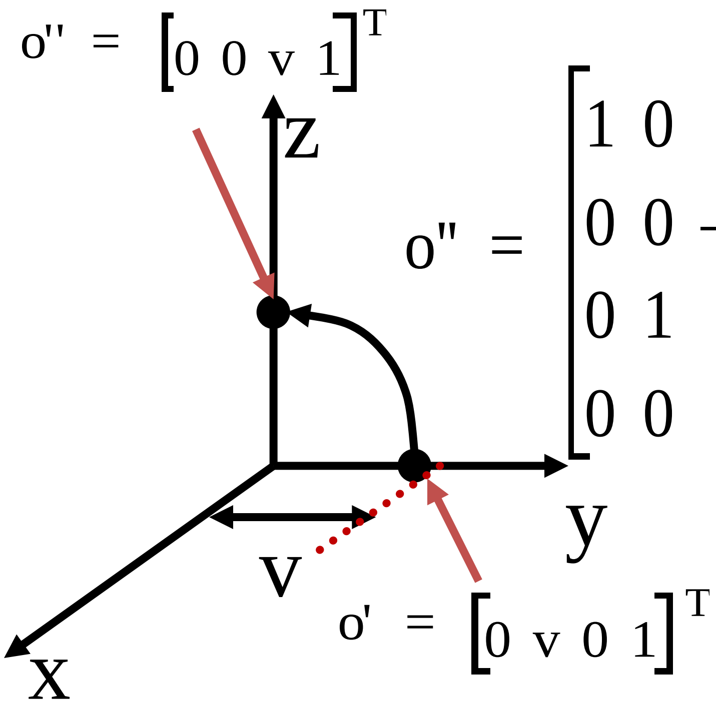
- Mapping:
  - Point → Point' (both expressed in same coordinates)
- Compound mapping:
  - Point' → Point'' (still in same coordinates)

- Operators have fixed axis compounding semantics.

Example: Operating on a Point

- A point at the origin is translated along the y axis by ‘v’ units and then the resulting point is rotated by 90 degrees around the x axis.

$$o'' = Rotx(\pi/2)Trans(0, v, 0)o$$



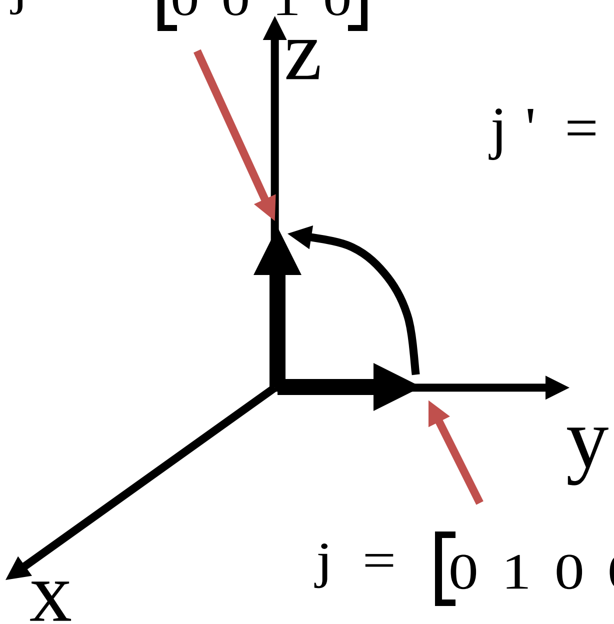
$$o'' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0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

Result

Example: Operating on a Direction

- The y axis unit vector is “translated” along the y axis by ‘v’ units and then rotated by 90 degrees around the x axis.

$$j' = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$$



$$j = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$$

$$j' = \mathrm{Rotx}(\pi/2)\mathrm{Trans}(0, v, 0)\ j$$

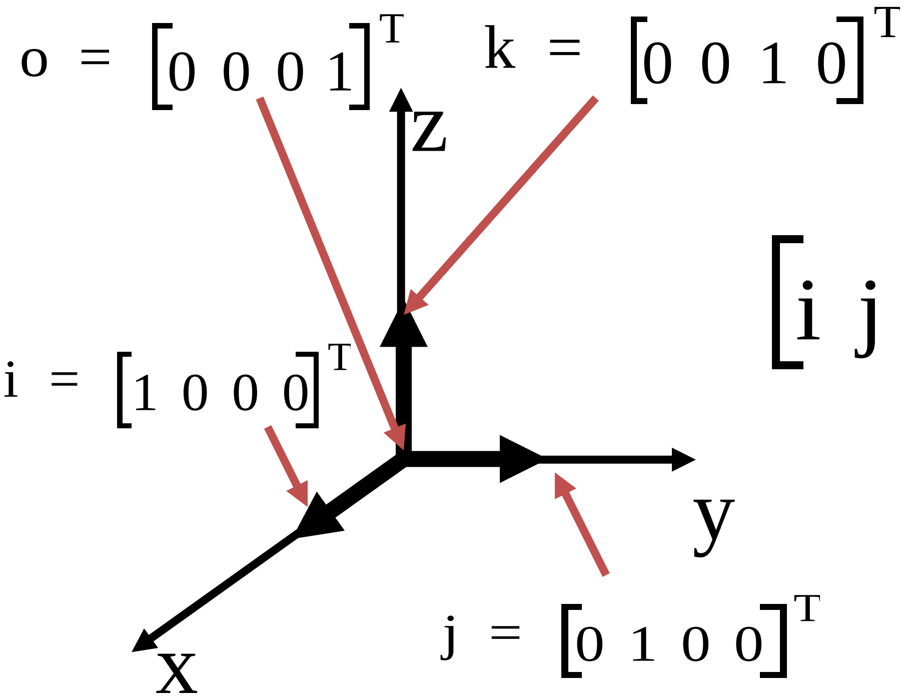
$$j' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 &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

Same!

- Having a zero scale factor disables translation.

HTs as Coordinate Frames

- The columns of the identity HT can be considered to represent 3 directions and a point – the coordinate frame itself.



3 directions / point

$$\begin{bmatrix} i & j & k & o \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

Example: Operating on a Frame

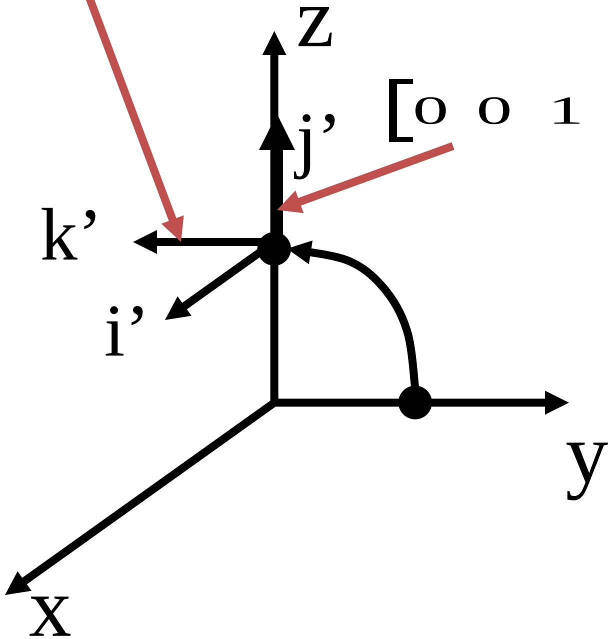
- Each resulting column of this result is the transformation of the corresponding column in the original identity matrix …

.

$$\begin{bmatrix}0 & -1 & 0 & 0\end{bmatrix}^T$$

$$I' = Rotx(\pi/2)Trans(0,v,0)I$$

$$\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}^T$$



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

Epiphany 1: HTs are Operators, and Operands, and Displacements.

- 1) The operator that moves points as desired also moves axes in the same way.
- But because the columns of an input matrix are treated independently in matrix multiplication…
- 2) … the operator also moves entire frames (4 columns) in one shot when you express them as a matrix.
  - Note that the HT matrix can now be either operator or operand.
- As Operand: But because frames can be embedded to track the motions of rigid bodies.
  - We can use this idea to computationally track the position and orientation of rigid bodies….
- As Operator: Every orthogonal matrix can be viewed as a displacement in translation and rotation.
  - Can be visualized as one set of axes located with respect to another set.

Epiphany 2 : Operator = Transformed Old Frame

- Notice the result is:

$$\text{Rotx}(\pi/2) * [Trans(0, v, 0) * I]$$

- Which is:

$$[\text{Rotx}(\pi/2) * Trans(0, v, 0)] * I$$

- Which is the compound operator:

$$[\text{Rotx}(\pi/2) * Trans(0, v, 0)]$$

- When the operand is the identity, the result is also the operator itself.

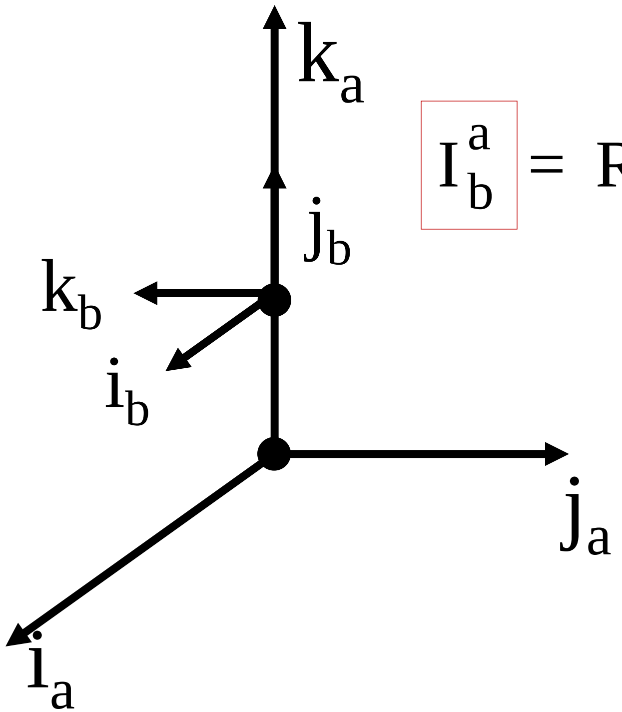
Epiphany 3 : Operator = Transform

- Because the operator, like all operators, expresses the new frame in the coordinates of the original frame….
- 1) The operator has columns that express the new axes in the coordinates of the old ones.
- Because $\underline{y} = A\underline{x} = \underline{a}_1 x_1 + \underline{a}_2 x_2 + \cdots$ (where $\underline{a}_1$ is 1st col of A)…
- And because the columns of the operator are the transformed unit vectors and origin….
- Then the operator [Rotx($\pi/2) * Trans(0, v, 0)$] must convert coordinates from the primed frame to the unprimed frame.
- See below for more..

Conversion of Basis

- Operator: the result is the movement of frame a's unit vectors ("basis") to those of frame b.

- So the result must express the unit vectors of frame b in coordinates of frame a.

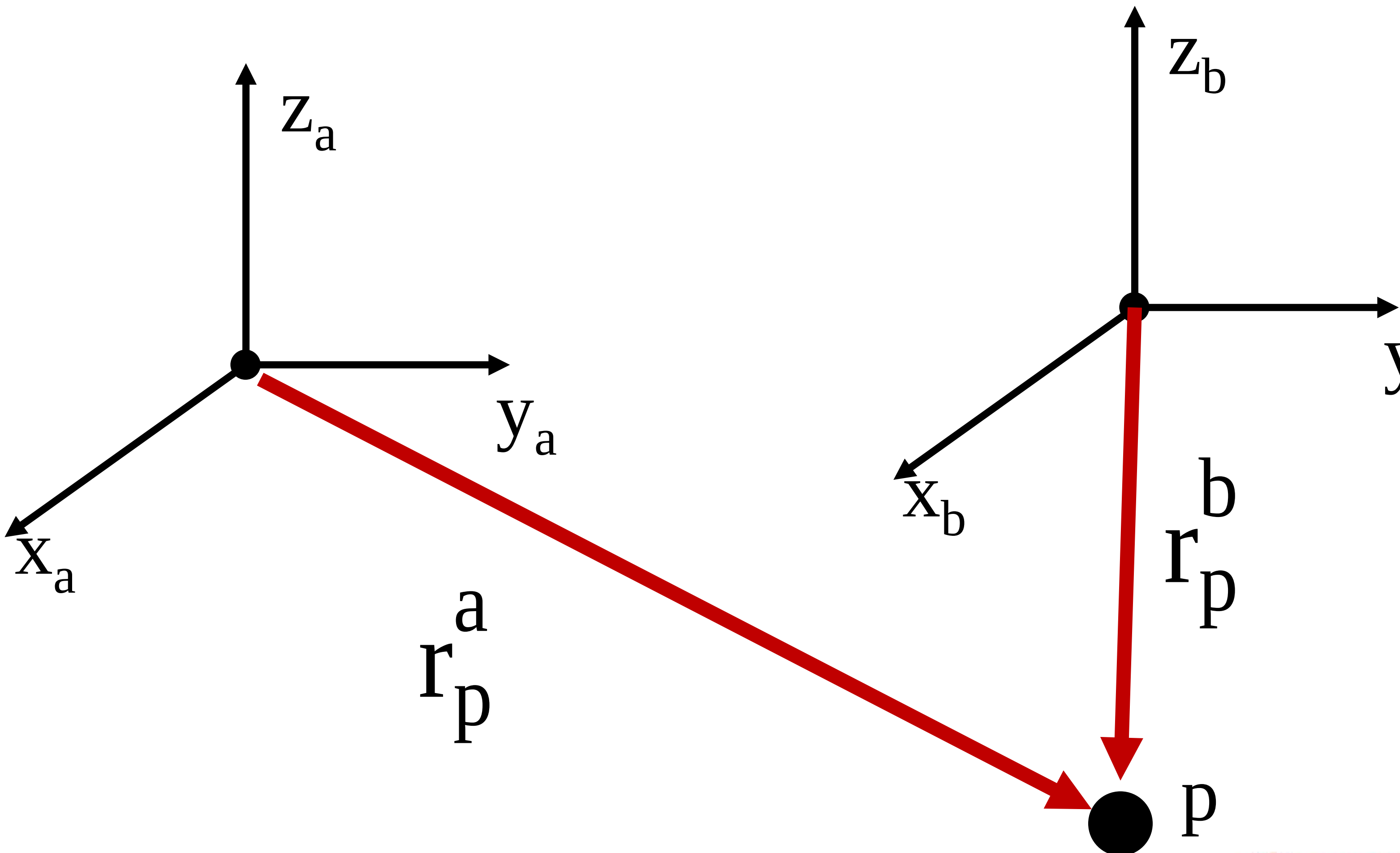


$$I_b^a = \text{Rotx}(\pi/2)\text{Trans}(0, v, 0)I_a^a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- We have converted coordinates of the basis b from frame b to frame a.

Converting Frames of Reference

- Converting frames is about expressing the same physical point with respect to a new origin and set of unit vectors.



Converting Coordinates

- Consider a general point expressed relative to frame b in the coordinates of frame b.

By Definition: $$r_p^b = x_p^b({}^b\dot{i}_b) + y_p^b({}^b\dot{j}_b) + z_p^b({}^b k_b) + {}^b o_b$$

- The unit vectors can be expressed in any coordinate system we like. Choose a.

Origin change

$$r_p^a = x_p^b({}^a\dot{i}_b) + y_p^b({}^a\dot{j}_b) + z_p^b({}^a k_b) + o_b^a$$

$$r_p^a = x_p^b(I_b^a i_b) + y_p^b(I_b^a j_b) + z_p^b(I_b^a k_b) + I_b^a o_b$$

Converting Coordinates

- This is:

$$r_p^a = I_b^a[x_p^b({}^b i_b) + y_p^b({}^b j_b) + z_p^b({}^b k_b) + {}^b o_b] \quad \underbrace{}_{} \; r_p^b$$

- Or more simply:

$$r_p^a = I_b^a \, r_p^b$$

- Because I_b^a converts the coordinates of the basis, it converts to coordinates of an arbitrary vector too:
  - because an arbitrary vector is just a linear combination of the basis vectors.
  - and matrices are linear operators

Operator/Transform Duality

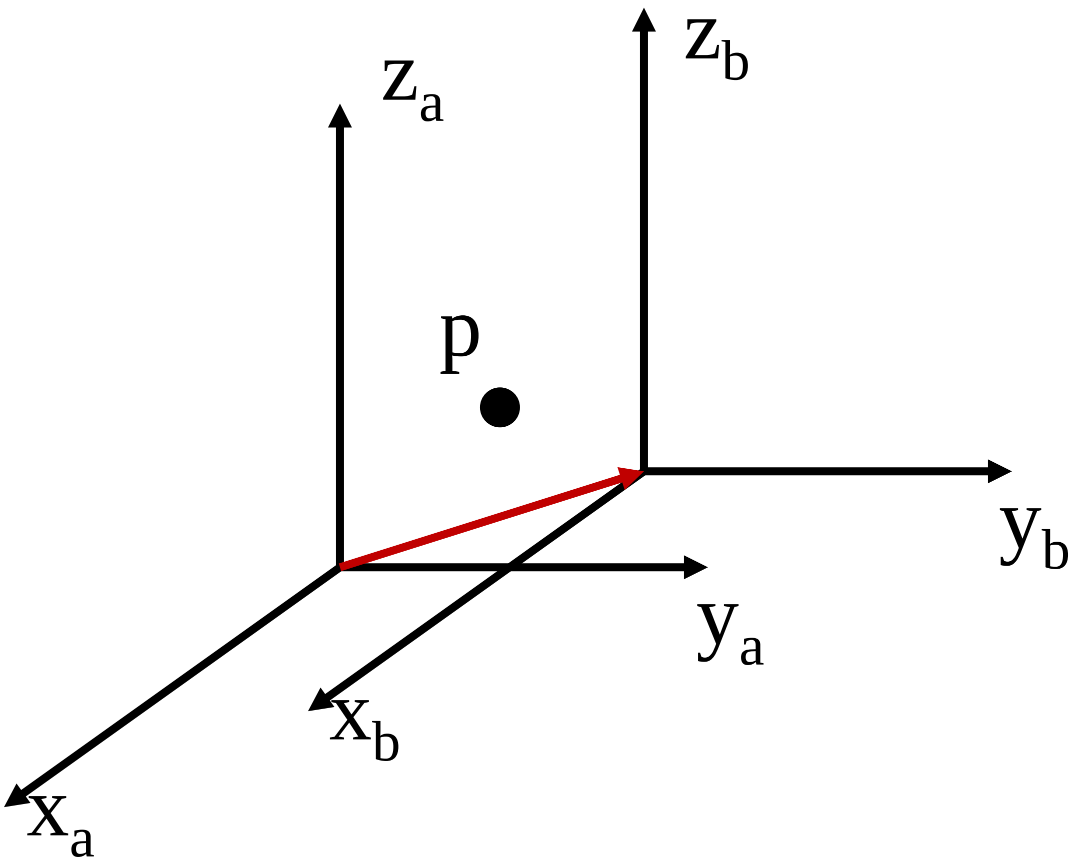
- The homogeneous transform that moves frame ‘a’ into coincidence with frame ‘b’ (operator) also converts the coordinates (transform) of points in the opposite direction - from frame ‘b’ to frame ‘a’.
- Because, of the *opposite direction* semantics, its sometimes more convenient to use the matrices which convert coordinates from frame ‘a’ to frame ‘b’.
  – These are just the matrix inverses.

Aligning Operations

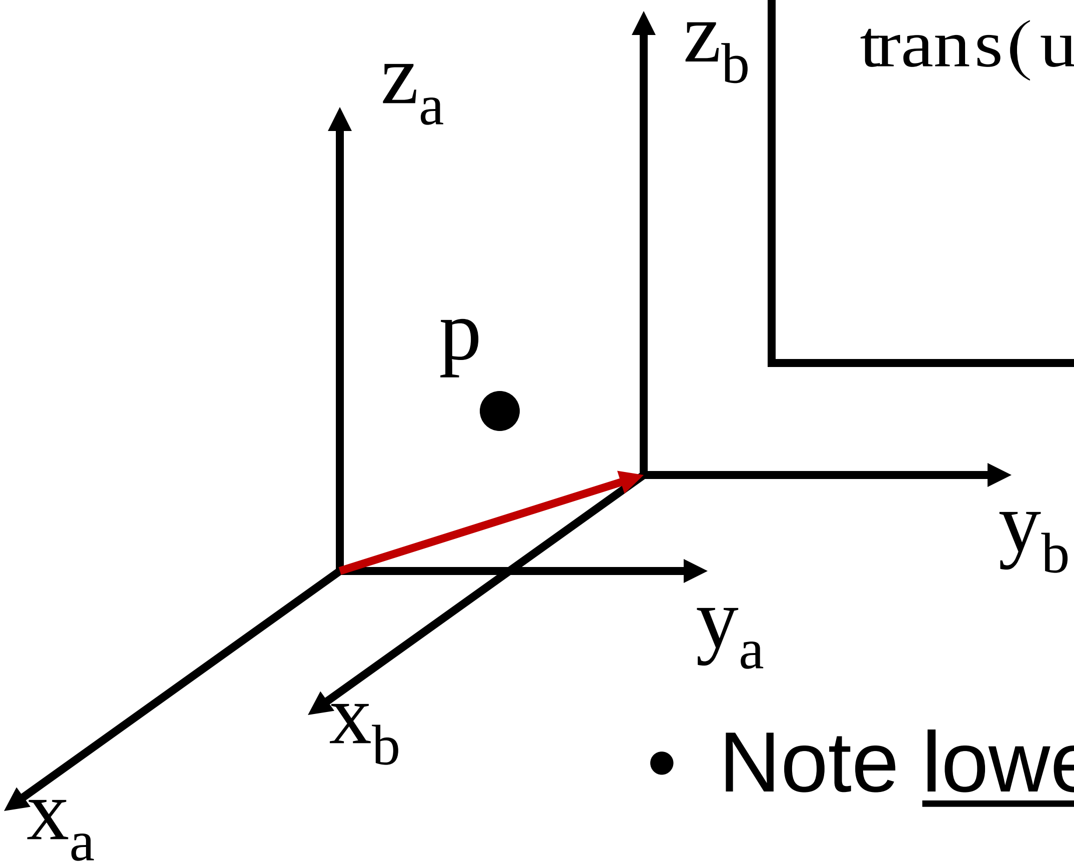
- We can ask of any two frames:
  - What operations, applied to one frame, bring it into coincidence with the other.
- To formulate the aligning operations
- ……. is equivalent to formulating the coordinate transformation.
- One of the biggest ideas in 3D kinematics.

Transforms

- Mapping:
  - Point [a] → Point [b] (same physical point)
  - Think now of moving frame a into coincidence with frame b.



Transforms



$$\mathrm{trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & -u \\ 0 & 1 & 0 & -v \\ 0 & 0 & 1 & -w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

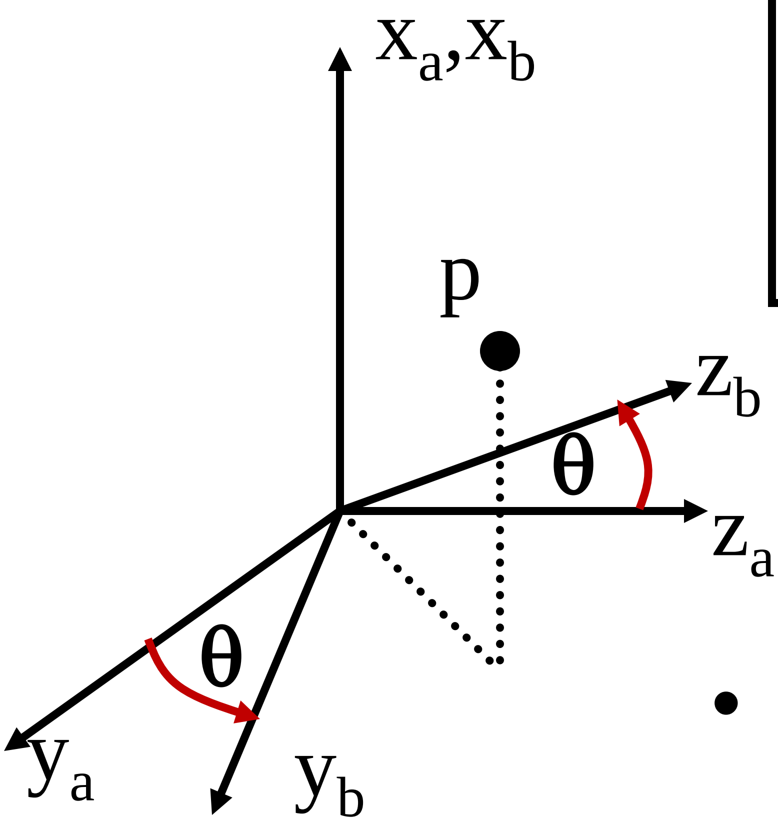
(u,v,w) are defined as the position of the origin <u>**of b**</u> <u>**wrt a**</u> expressed in frame a coords.

- Note <u>lowercase t</u> in trans().

Note

- Phi and theta may need to be swapped on the following slides.

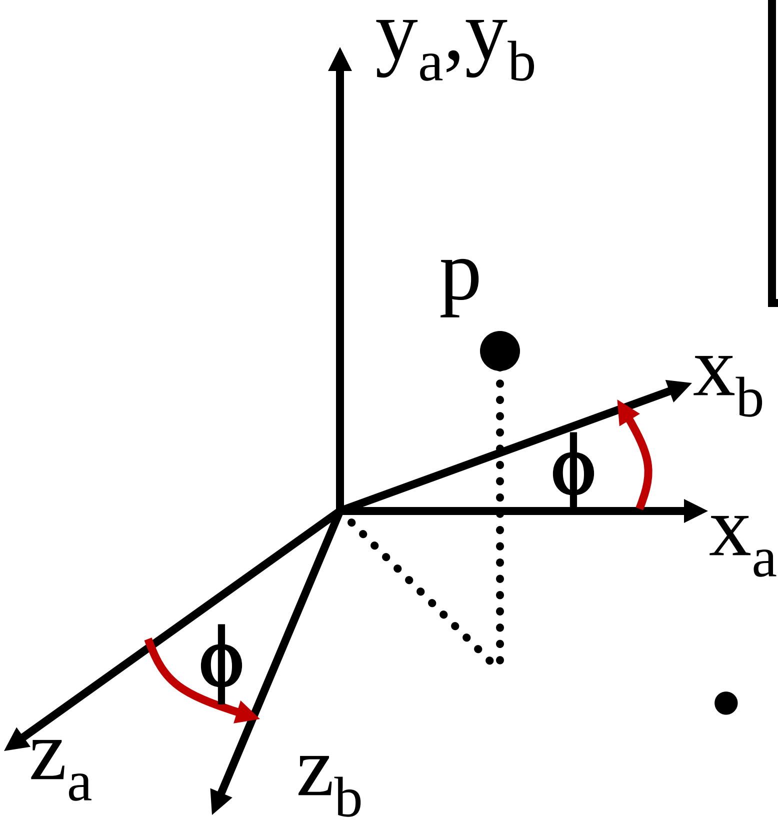
Transforms



$$\text{rotx}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & s\theta & 0 \\ 0 & -s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note <u>lowercase r</u> in rotx().

Transforms

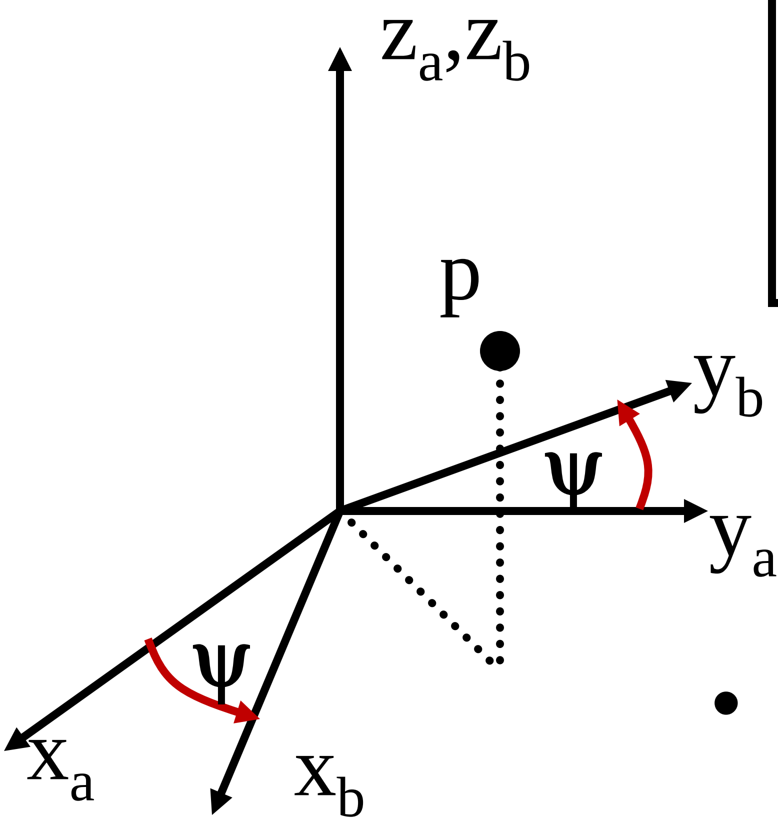


$$\mathrm{roty}(\phi) = \begin{bmatrix} c\phi & 0 & -s\phi & 0 \\ 0 & 1 & 0 & 0 \\ s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note <u>lowercase r</u> in roty().

Transforms

$$rotz(\psi) = \begin{bmatrix} c\psi & s\psi & 0 & 0 \\ -s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



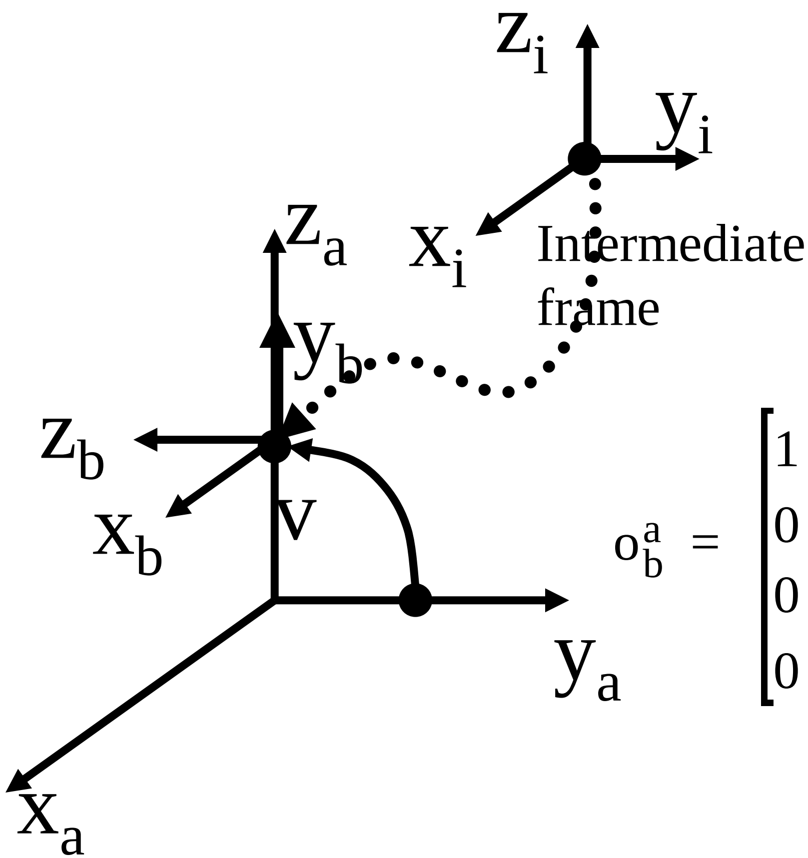
- Note <u>lowercase r</u> in rotz().

Compound Transforms

- Mapping:
  - Point [a] → Point [b] (same physical point)
  - Result expressed in frame b
- Compound mapping
  - Point [b] → Point [c] (same physical point)
  - Result now expressed in frame c.
- Transforms have moving axis compounding semantics.
  - Result is not expressed in “the original frame” but rather in the last one.

Example: Compound Transformation

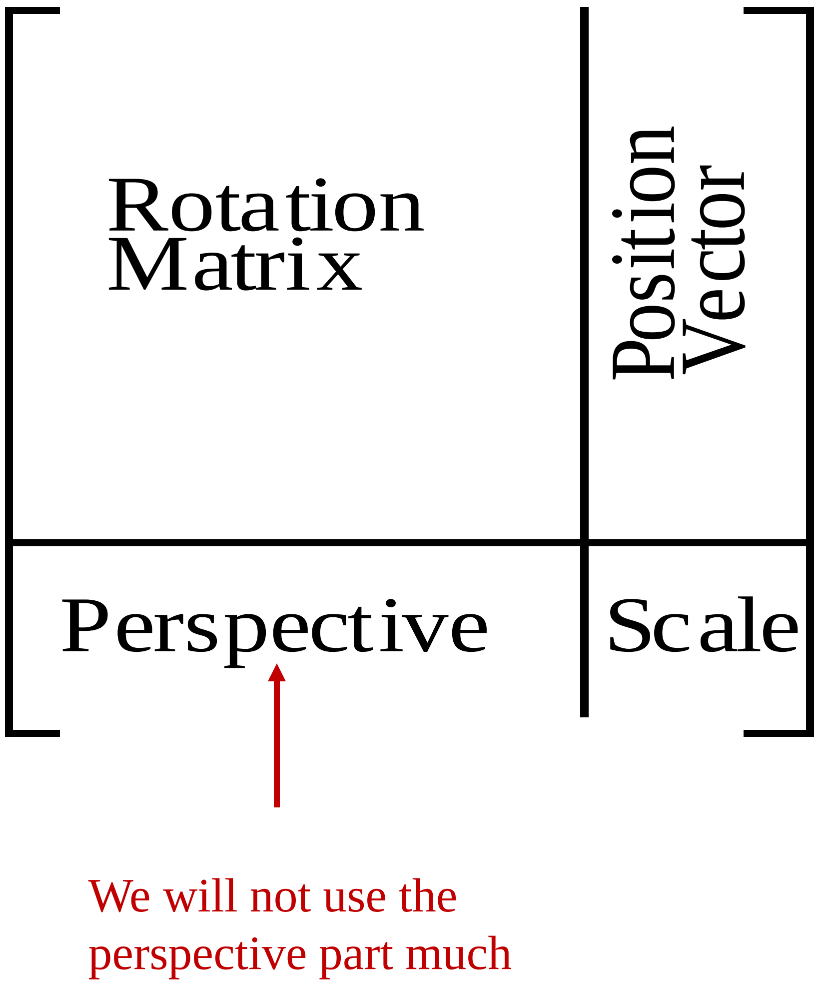
- The origin of frame b has coordinates $[0\ 0\ v\ 1]^T$ in frame a. Prove it.
- Takes two fundamental operations. Compound transforms to convert $o^b \rightarrow o^a$.



$$o_b^a = \text{trans}(0, 0, -v)\,\text{rotx}(-\pi/2)\,o_b^b$$

$$o_b^a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v \\ 1 \end{bmatrix}$$

Format of HTs



Inverse of a HT

$$\left[\begin{array}{ccc|c} & R & & \underline{p} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]^{-1} = \left[\begin{array}{ccc|c} & R^T & & -R^T\underline{p} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Duality Theorem

- Note that transforms and operators of the same name are (matrix) inverses:

$$\text{trans}(0, 0, v) = \text{Trans}(0, 0, v)^{-1}$$

$$\text{rotx}(\theta) = \text{Rotx}(\theta)^{-1}$$

$$\text{roty}(\phi) = \text{Roty}(\phi)^{-1}$$

$$\text{rotz}(\psi) = \text{Rotz}(\psi)^{-1}$$

- This implies that a sequence of transforms in one order (say, left to right) is identical to the same sequence of operators in the opposite order.
- The latter view is traditional in robotics.
- Without loss of generality, we will use operators only from now on.

Outline

- 2.1 Conventions and Definitions
- 2.2 Matrices
- 2.3 Fundamentals of Rigid Transforms
  - 2.3.1 Definitions
  - 2.3.2 Why Homogeneous Transforms
  - 2.3.3 Semantics and Interpretations
- Summary

Summary

- Tensors are just matrices with 3 or more indices.
- Arbitrary functions on matrices can be defined.
- The row space is orthogonal to the null space.
- Gaussian Elimination can be performed blockwise to express solution to big problems in terms of solutions to small problems.
- Matrix valued functions can be differentiated with respect to scalars and vectors.
  - Layout of resulting tensor is implicit in the latter case.

Summary

- An operator formed from an orthogonal HT preserves distances and hence is rigid.
- Homogeneous Transforms are:
  - Operators
  - Transforms
  - Frames
- They can be both the things that operate on other things and the things operated upon.

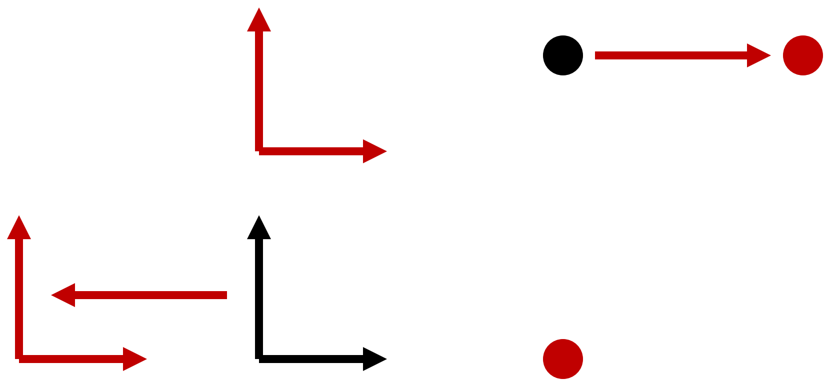
91

Summary

- Such a 4X4 operator matrix has these properties.
  - It rotates and/or translates points and directions and hence rotates and translates coordinate frames.
  - Its columns represent the unit vectors and origin of the result of operating on a coordinate frame expressed in the coordinates of the original frame.
  - It converts coordinates of points and directions from the result to the original frame.

Summary

- Everything is relative. There is no way to distinguish moving a point “forward” from moving the coordinate system “backward”.



- In both cases, the resulting (red) point has the same relationship to the redframe.