Character Deformation

Affine Transformation

Affine Transformation

Linear transformation & translation

$$\vec{x}' = K\vec{x} + \vec{t}$$

Affine transformation

$$\begin{pmatrix} \vec{x}' \\ 1 \end{pmatrix} = \begin{bmatrix} K & \vec{t} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \vec{x} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{yz} & K_{zy} & K_{zz} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} & t_x \\ K_{yx} & K_{yy} & K_{yz} & t_y \\ K_{zx} & K_{zy} & K_{zz} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Properties of Affine Transformation

- Composite of two affine transformations makes an affine transformation

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

- Associative property: $(A_1A_2)A_3 = A_1(A_2A_3)$

- Inverse is also an affine transformation

$$\begin{bmatrix} K^{-1} & -K^{-1}\vec{t} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} K & \vec{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$$

Useful Property of Associative Law

Associative law for matrix: $A(BC) = (AB)C$

$$E\left(D\left(C\left(B(Ax)\right)\right)\right) = \underbrace{(EDCBA)}_{K}x$$

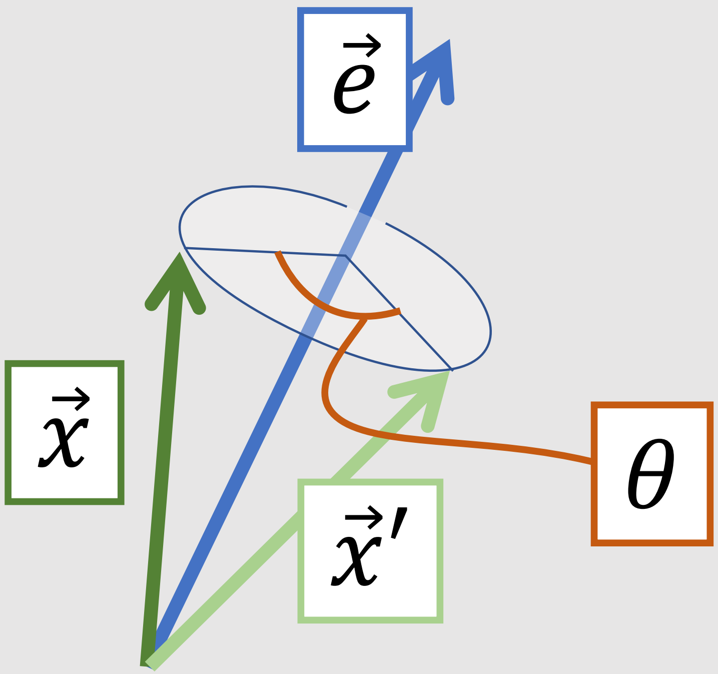
Precompute $K = EDCBA$ to efficiently compute Kx for various x

Articulated Rigid Body

Rotation Around Axis

- The rotation is parameterized by axis vector $\vec{e}$ and angle θ

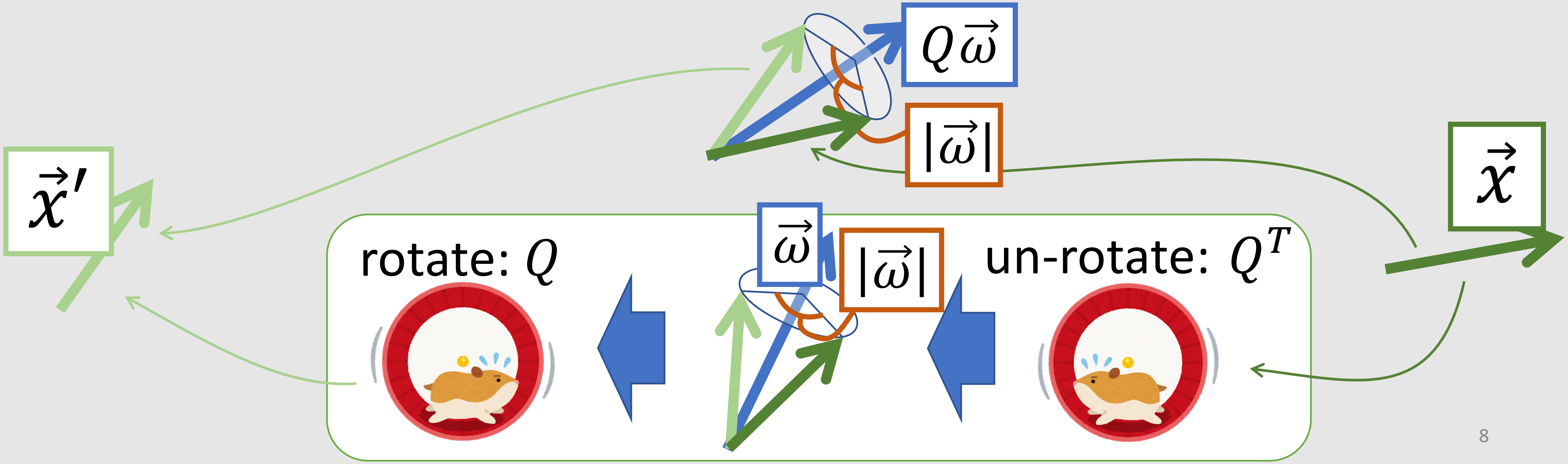
$$\vec{x}' = R(\underbrace{\theta\vec{e}}_{\vec{\omega}})\vec{x}$$



Rotation around Rotated Axis

- Axis is rotated with Q → un-rotate the object with Q^T then rotate around $\vec{\omega}$, then rotate back with Q

$$R(Q\vec{\omega}) = Q * R(\vec{\omega}) * Q^T$$



intrinsic rotation:
axis rotated

extrinsic rotation:
axis fix

$$R(Q\vec{\omega}) = Q * \boxed{R(\vec{\omega})} * Q^T$$

$$R(Q\vec{\omega}) * Q = Q * R(\vec{\omega})$$

matrix don't commute!

Rotation around Rotated Axis: Robotic Arm

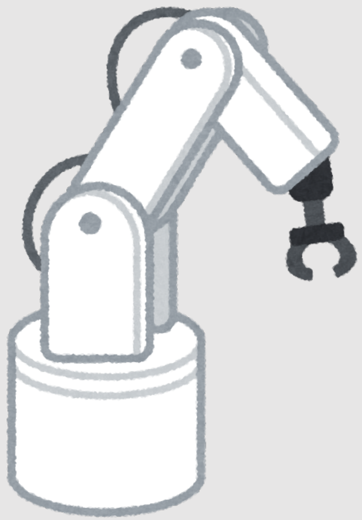
- Rotation of end-effector in a 4-link articulated body

intrinsic

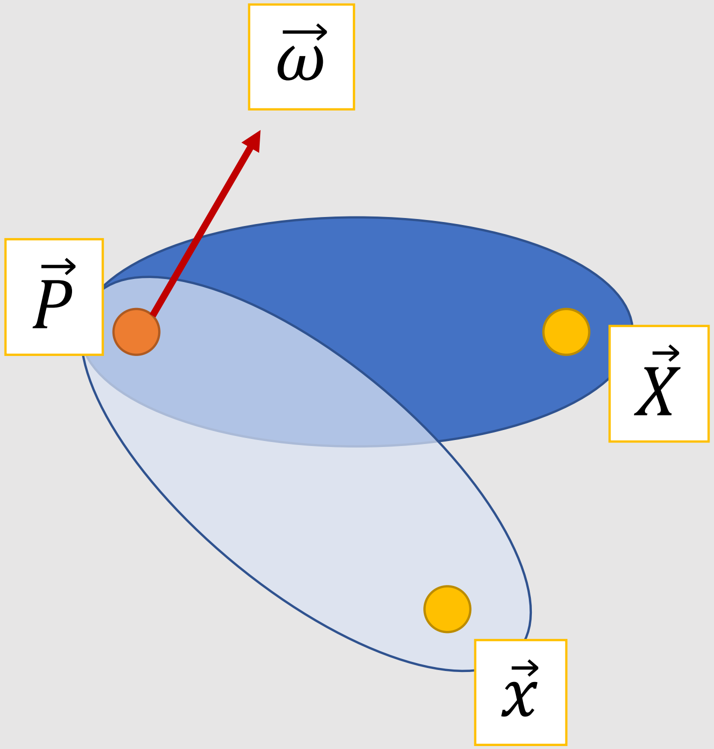
$$R_4(R_3R_2R_1\vec{\omega}_4) * R_3(R_2R_1\vec{\omega}_3) * R_2(R_1\vec{\omega}_2) * R_1(\vec{\omega}_1)$$

extrinsic

$$R_1(\vec{\omega}_1) * R_2(\vec{\omega}_2) * R_3(\vec{\omega}_3) * R_4(\vec{\omega}_4)$$



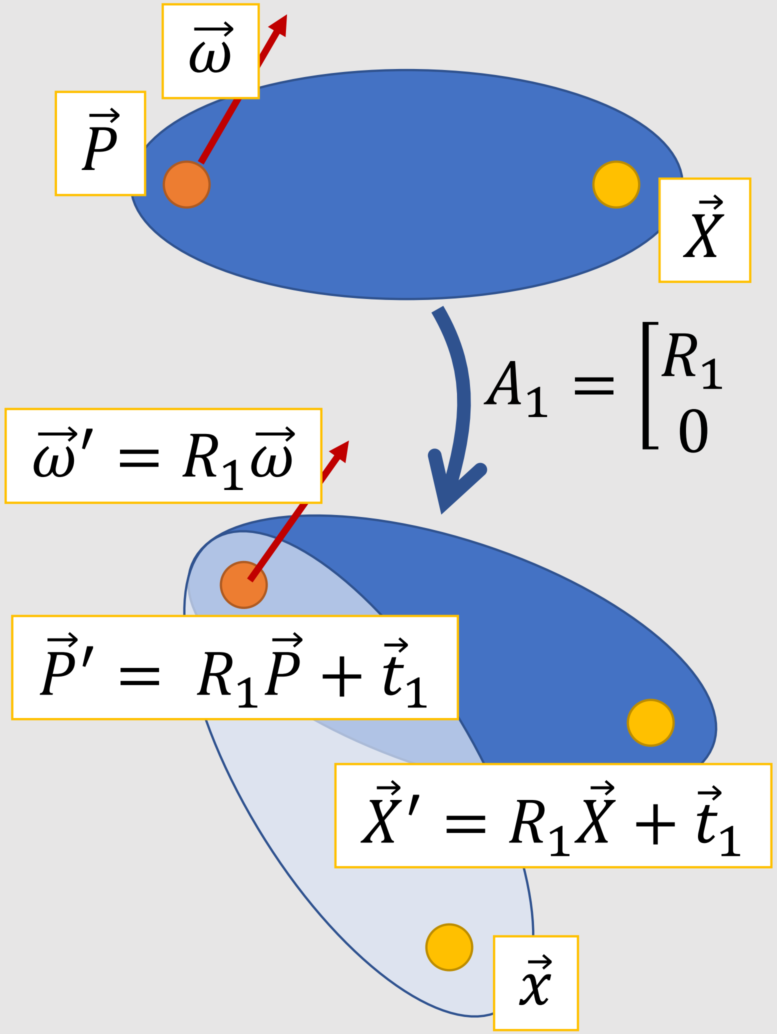
Affine Trans. of Rotation Around Fixed Point



$$\vec{x} = R(\vec{\omega})\left(\vec{X} - \vec{P}\right) + \vec{P}$$

$$\begin{pmatrix} \vec{x} \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} R(\vec{\omega}) & \vec{P} - R(\vec{\omega})\vec{P} \\ 0 & 1 \end{bmatrix}}_{A\left(\vec{\omega}, \vec{P}\right)} \begin{pmatrix} \vec{X} \\ 1 \end{pmatrix}$$

Rotation After Rigid Transformation

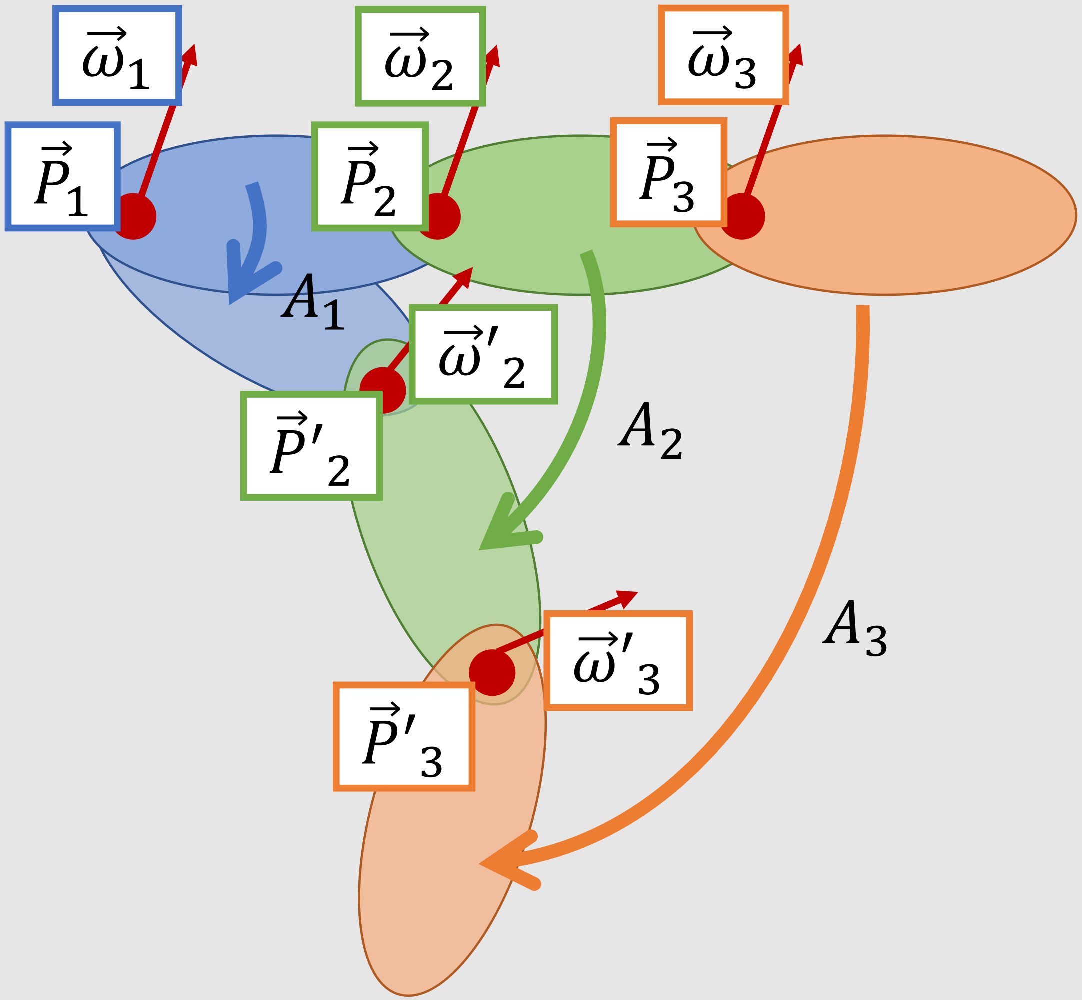


$$\begin{pmatrix} \vec{x} \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} R(\vec{\omega}') & \vec{P}' - R(\vec{\omega}')\vec{P}' \\ 0 & 1 \end{bmatrix}}_{A(\vec{\omega}', \vec{P}')} \underbrace{\begin{bmatrix} R_1 & \vec{t}_1 \\ 0 & 1 \end{bmatrix}}_{A_1} \begin{pmatrix} \vec{X} \\ 1 \end{pmatrix}$$

$$\begin{aligned} \vec{x} &= R(\vec{\omega}')(\vec{X}' - \vec{P}') + \vec{P}' \\ &= R(R_1\vec{\omega})(R_1\vec{X} + \vec{t}_1 - R_1\vec{P} - \vec{t}_1) + R_1\vec{P} + \vec{t}_1 \\ &= R(R_1\vec{\omega})R_1(\vec{X} - \vec{P}) + R_1\vec{P} + \vec{t}_1 \\ &= R_1 R(\vec{\omega})(\vec{X} - \vec{P}) + R_1\vec{P} + \vec{t}_1 \end{aligned}$$

$$\begin{pmatrix} \vec{x} \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} R_1 & \vec{t}_1 \\ 0 & 1 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} R(\vec{\omega}) & \vec{P} - R(\vec{\omega})\vec{P} \\ 0 & 1 \end{bmatrix}}_{A(\vec{\omega}, \vec{P})} \begin{pmatrix} \vec{X} \\ 1 \end{pmatrix}$$

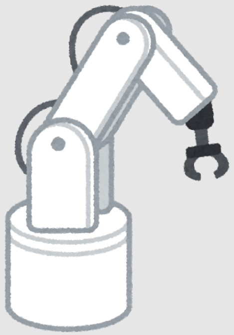
Affine Transformation of Articulated Body



$$A_1 = A(\vec{\omega}_1, \vec{P}_1)$$

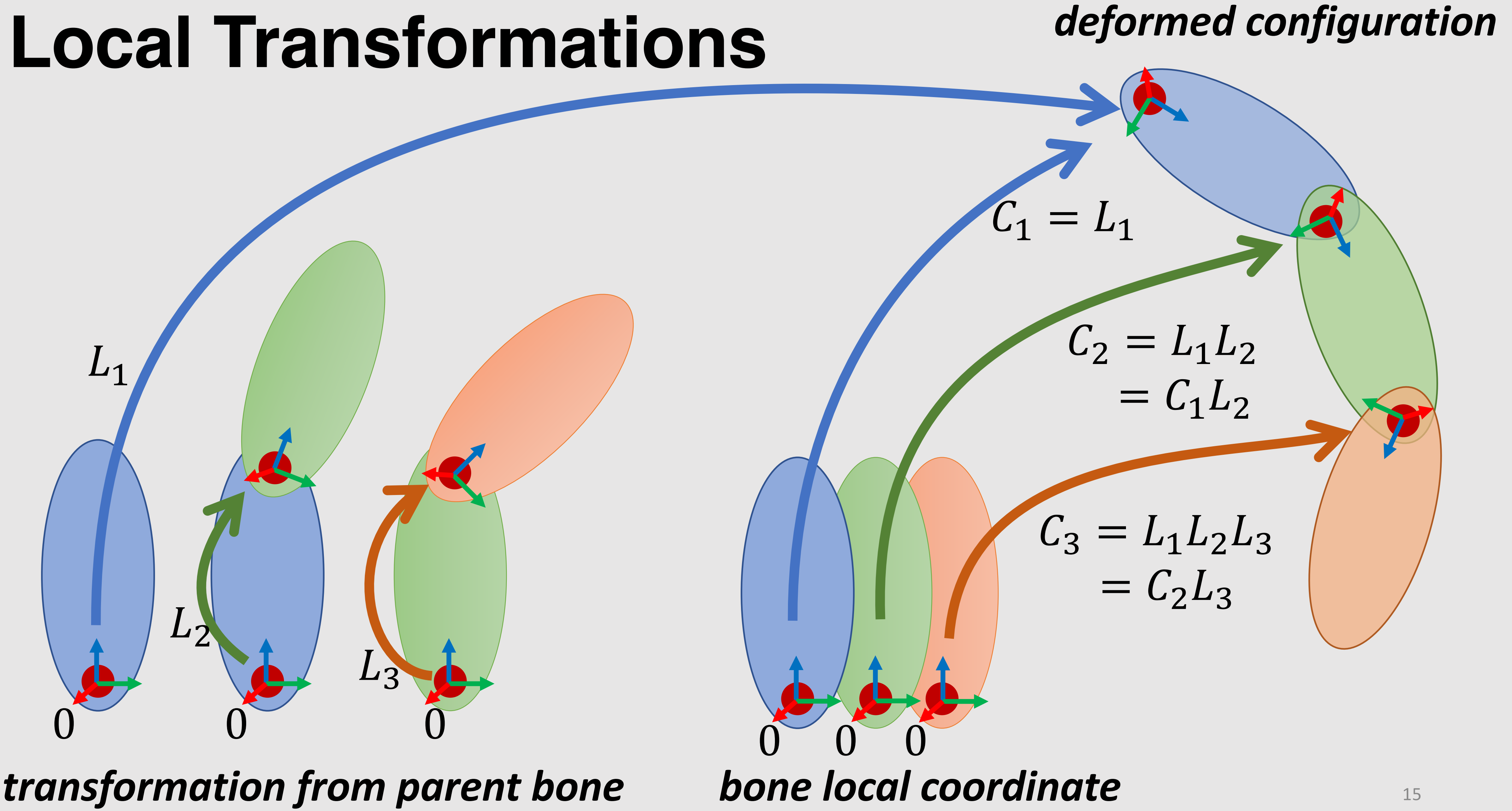
$$\begin{aligned} A_2 &= A(\vec{\omega}'_2, \vec{P}'_2)A_1 \\ &= A_1 A(\vec{\omega}_2, \vec{P}_2) \\ &= A(\vec{\omega}_1, \vec{P}_1)A(\vec{\omega}_2, \vec{P}_2) \end{aligned}$$

$$\begin{aligned} A_3 &= A(\vec{\omega}'_2, \vec{P}'_2)A_2 \\ &= A_2 A(\vec{\omega}_2, \vec{P}_2) \\ &= A(\vec{\omega}_1, \vec{P}_1)A(\vec{\omega}_2, \vec{P}_2)A(\vec{\omega}_3, \vec{P}_3) \end{aligned}$$

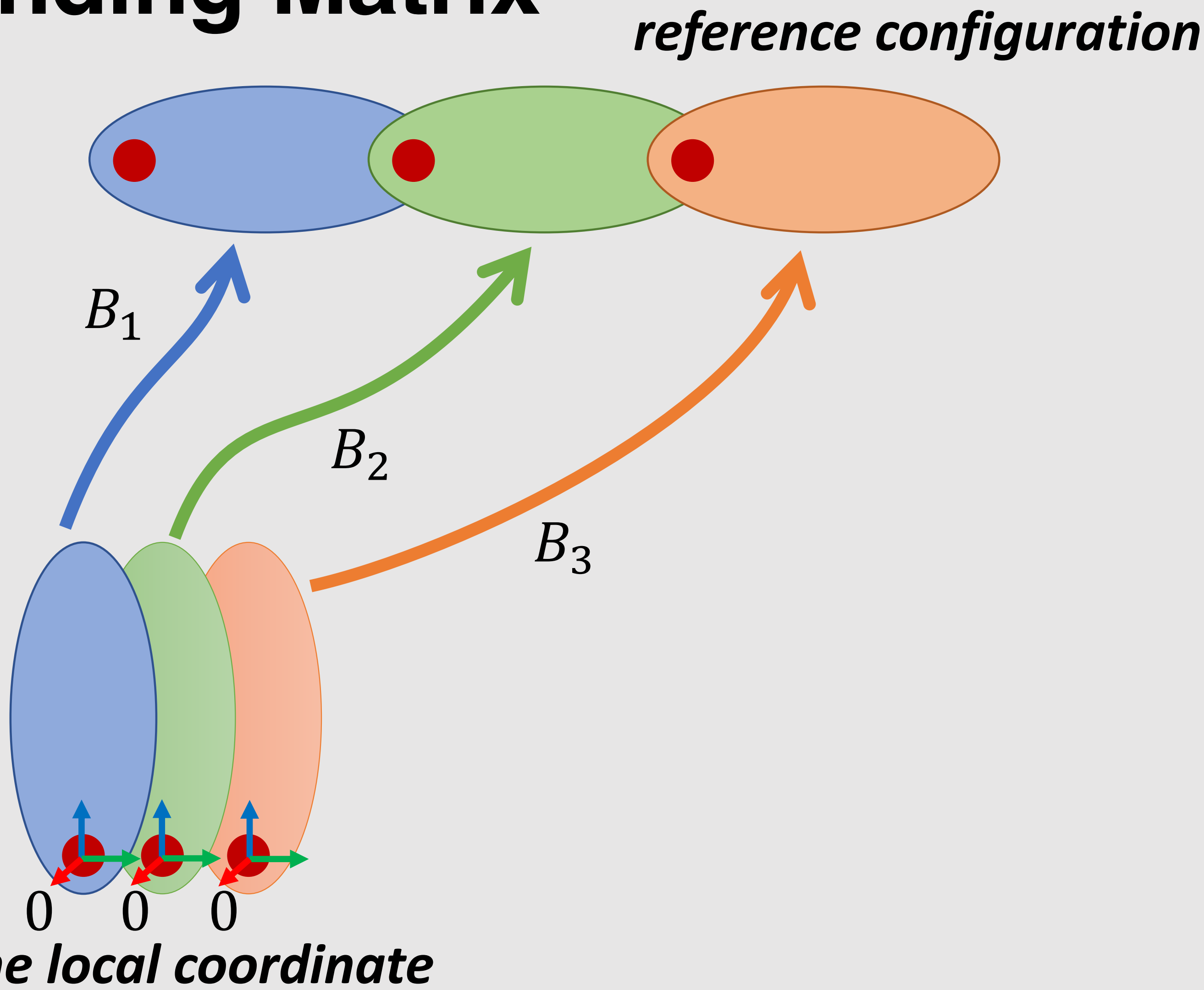


Binding Matrix

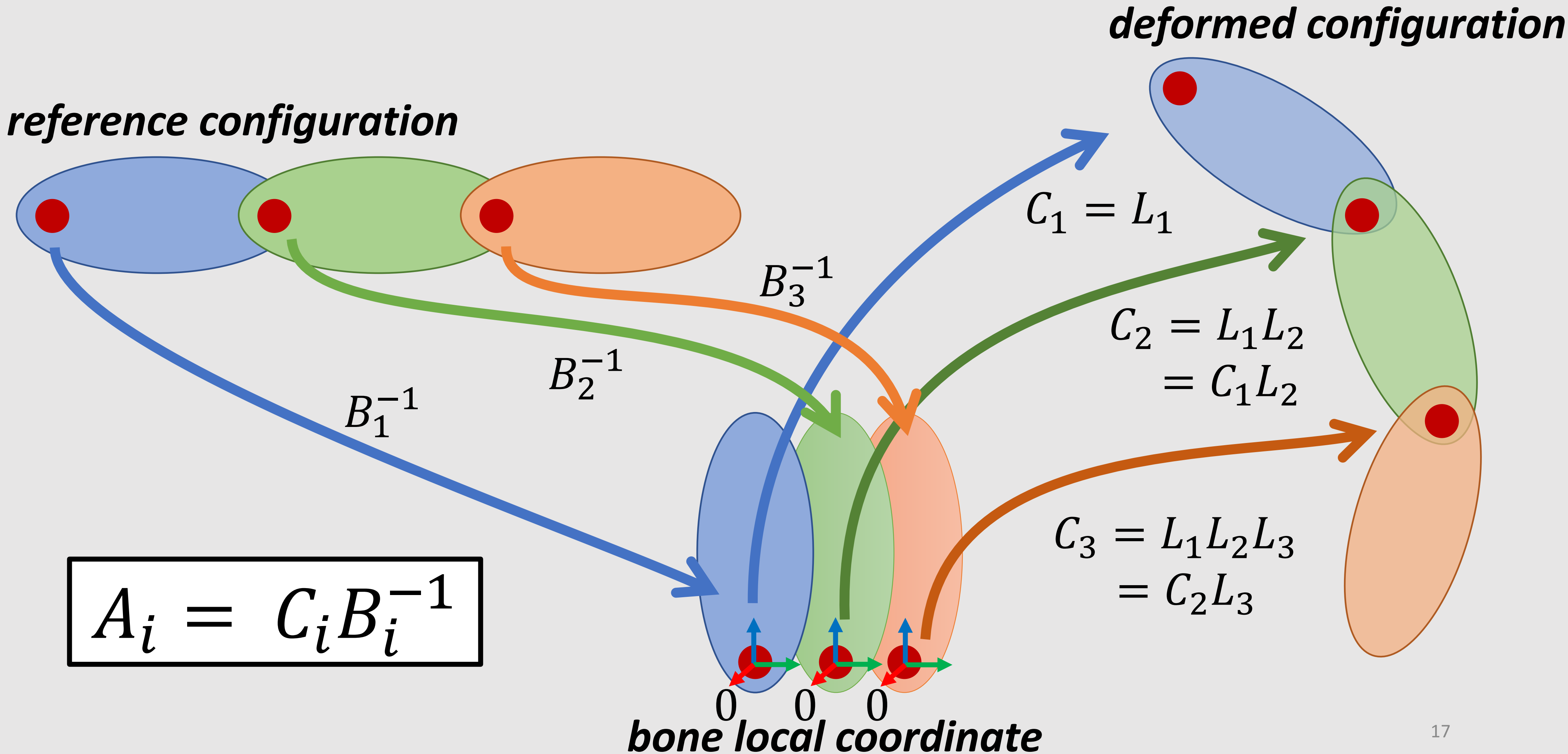
Local Transformations



Binding Matrix



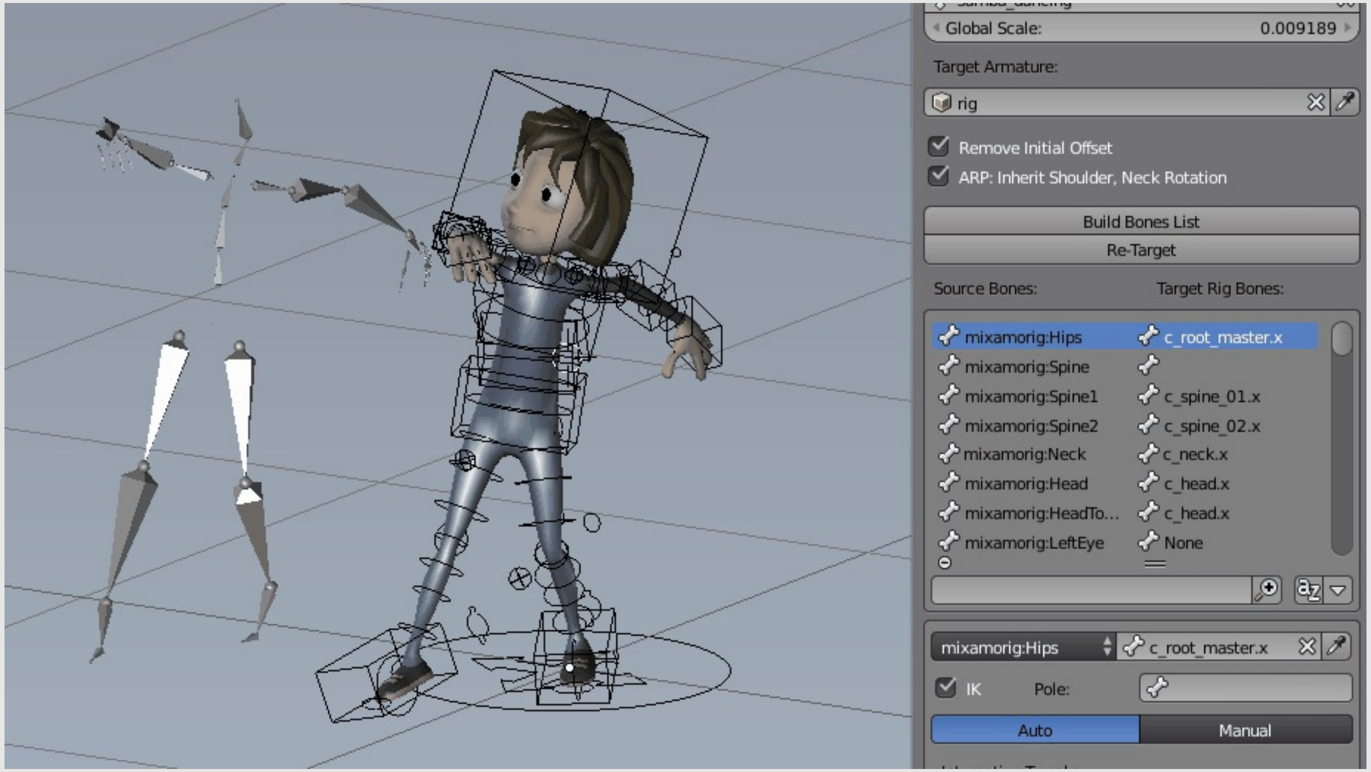
Inverse Binding Matrix



$$A_i = C_i B_i^{-1}$$

Linear Blend Skinning

Linear Blend Skinning is Industry Standard



Source “Auto-Rig Pro: Remap Update!
https://www.youtube.com/watch?v=-tuuijd2fCc

Linear Blend Skinning: Rest Pose

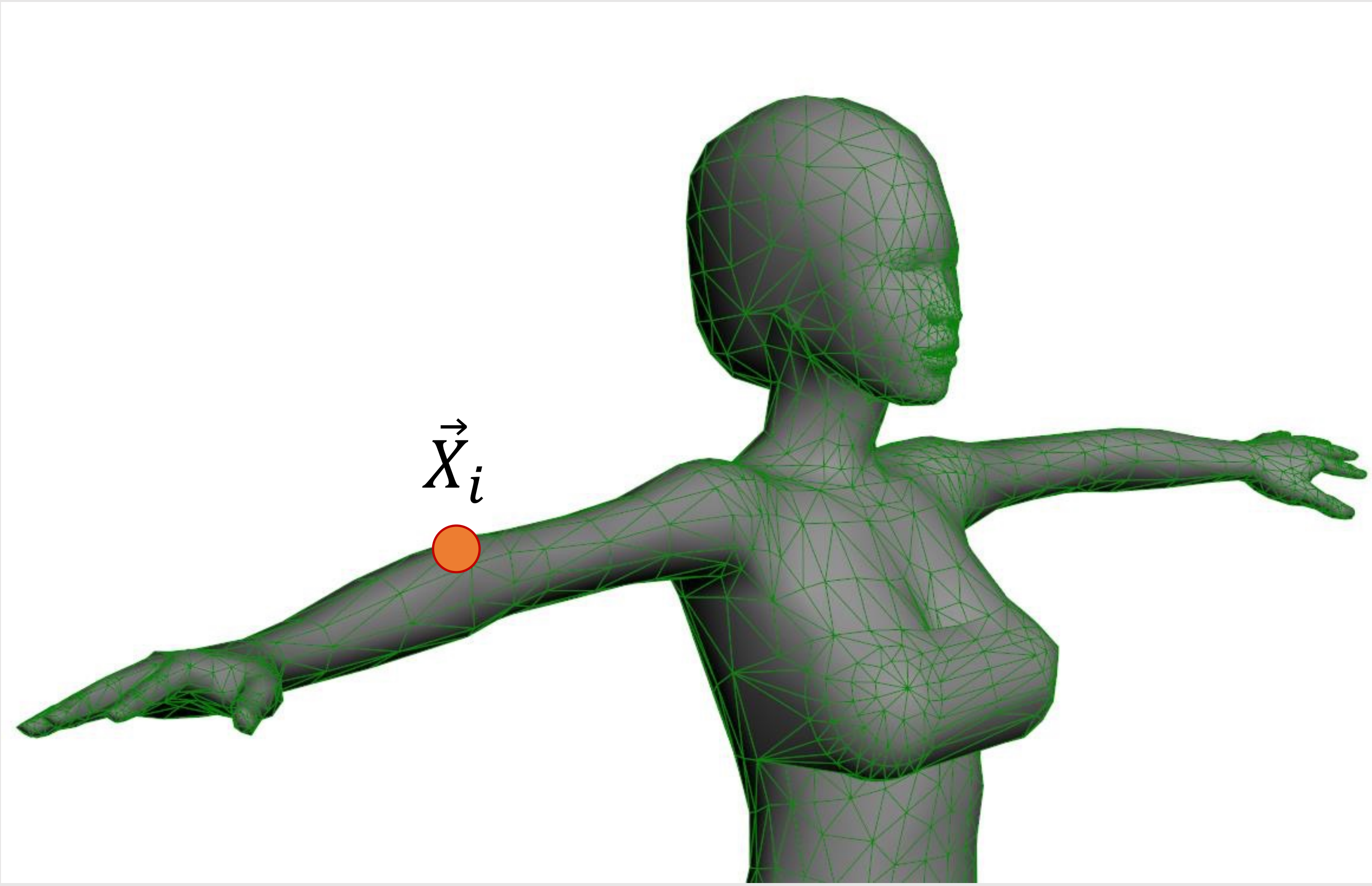


Image Credit: Ladislav Kavan (https://skinning.org/direct-methods-slides.pdf)

Each Bone Has Affine Transformation

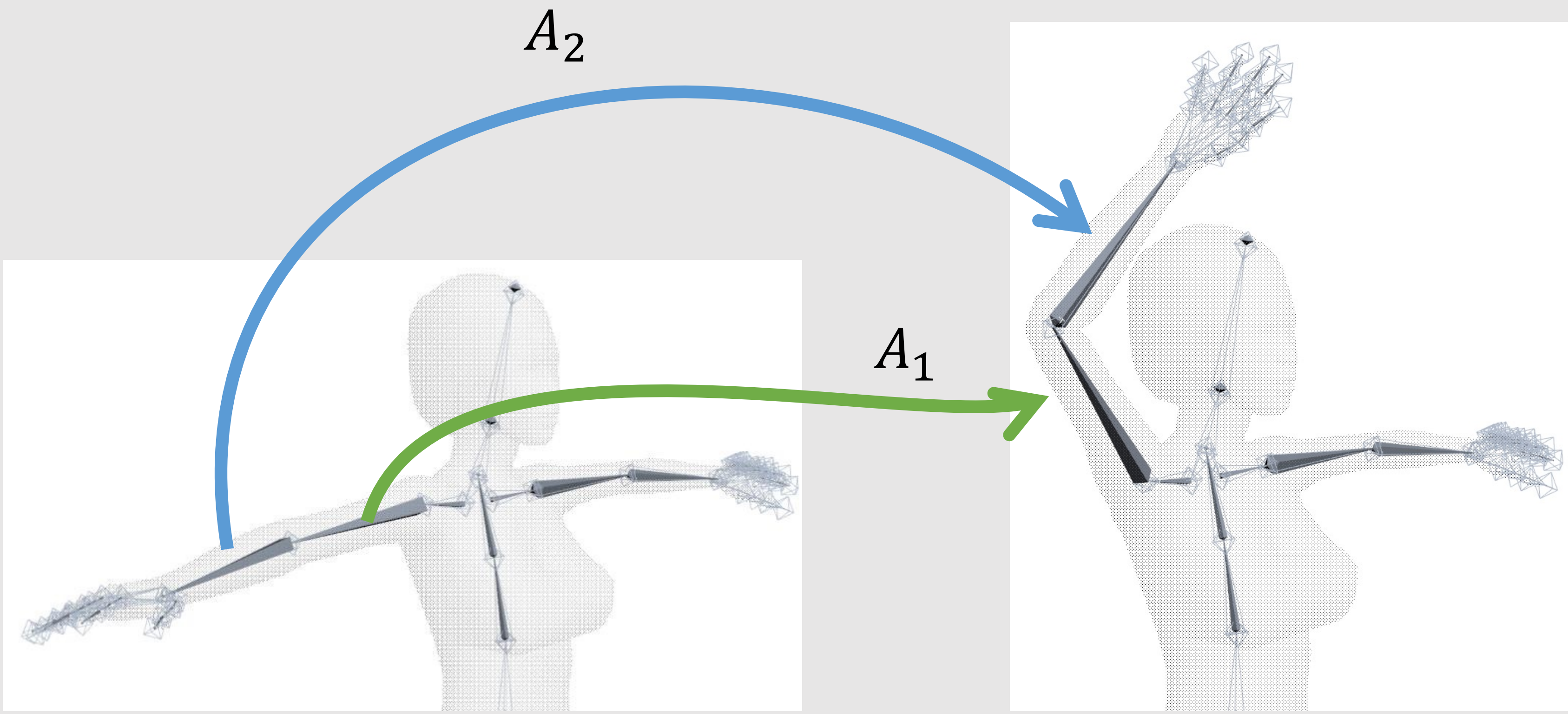
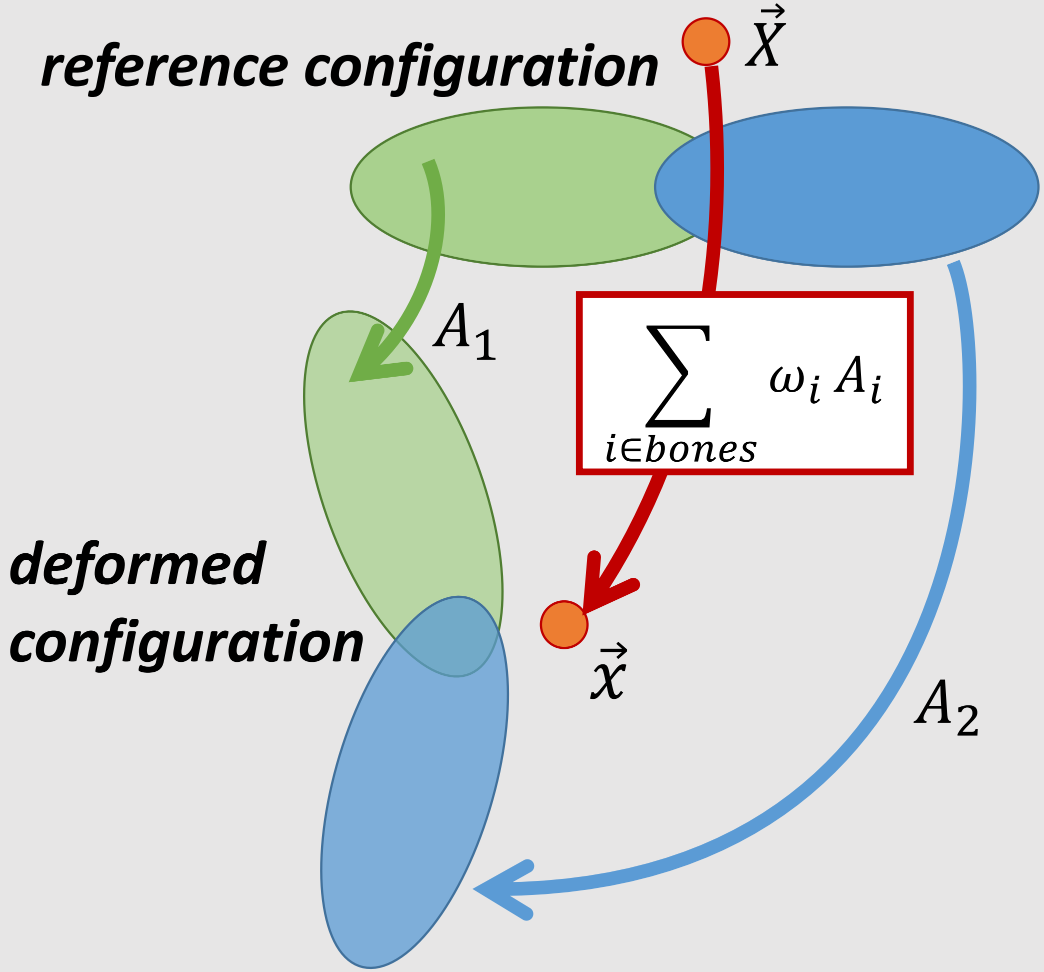


Image Credit: Ladislav Kavan (https://skinning.org/direct-methods-slides.pdf)

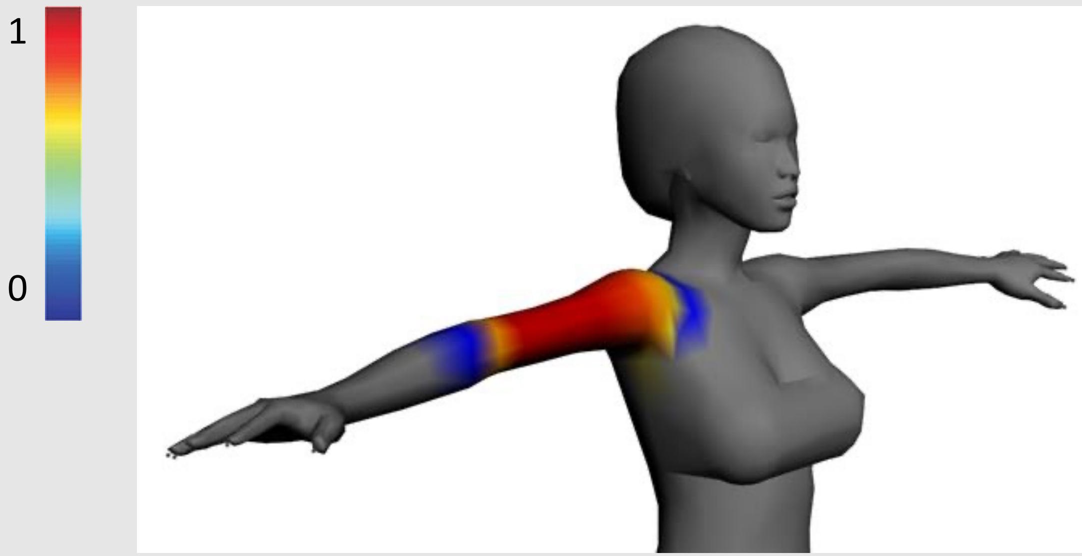
Linear Blend Skinning: Weighted Transform.



$$\begin{pmatrix} \vec{x} \\ 1 \end{pmatrix} = \sum_{b \in bones} \omega_b A_b \begin{pmatrix} \vec{X} \\ 1 \end{pmatrix}$$

Skinning Weight: Weight Defined on Vertices

Weights for upper arm: ω_1



Weights for lower arm: ω_2

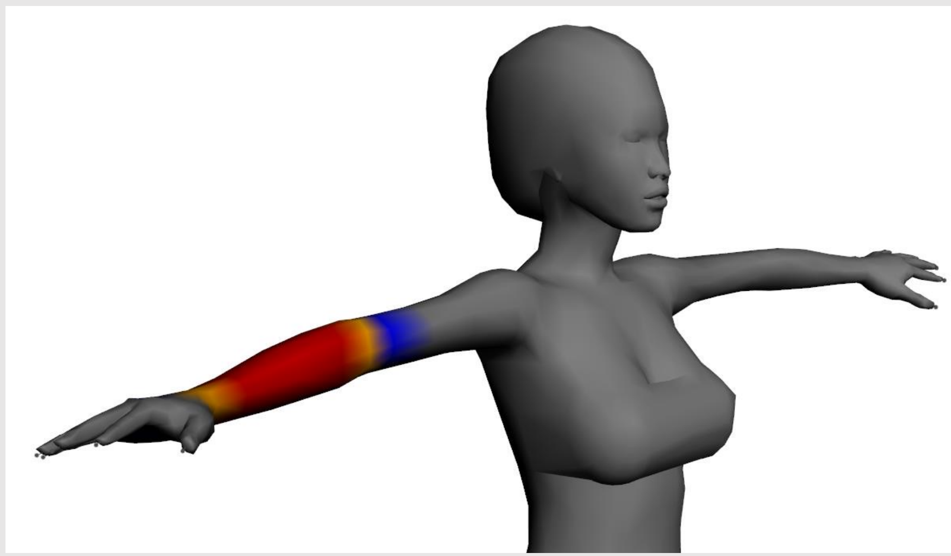


Image Credit: Ladislav Kavan (https://skinning.org/direct-methods-slides.pdf)

Reference

- Skinning: Real-time Shape Deformation, ACM SIGGRAPH 2014 Course, https://skinning.org/