GAMES 105
Fundamentals of Character Animation

Lecture 05:

# Data-driven Character Animation

**Libin Liu**

School of Intelligence Science and Technology
Peking University

GAMES105 课程交流

VCL @ PKU

# Outline

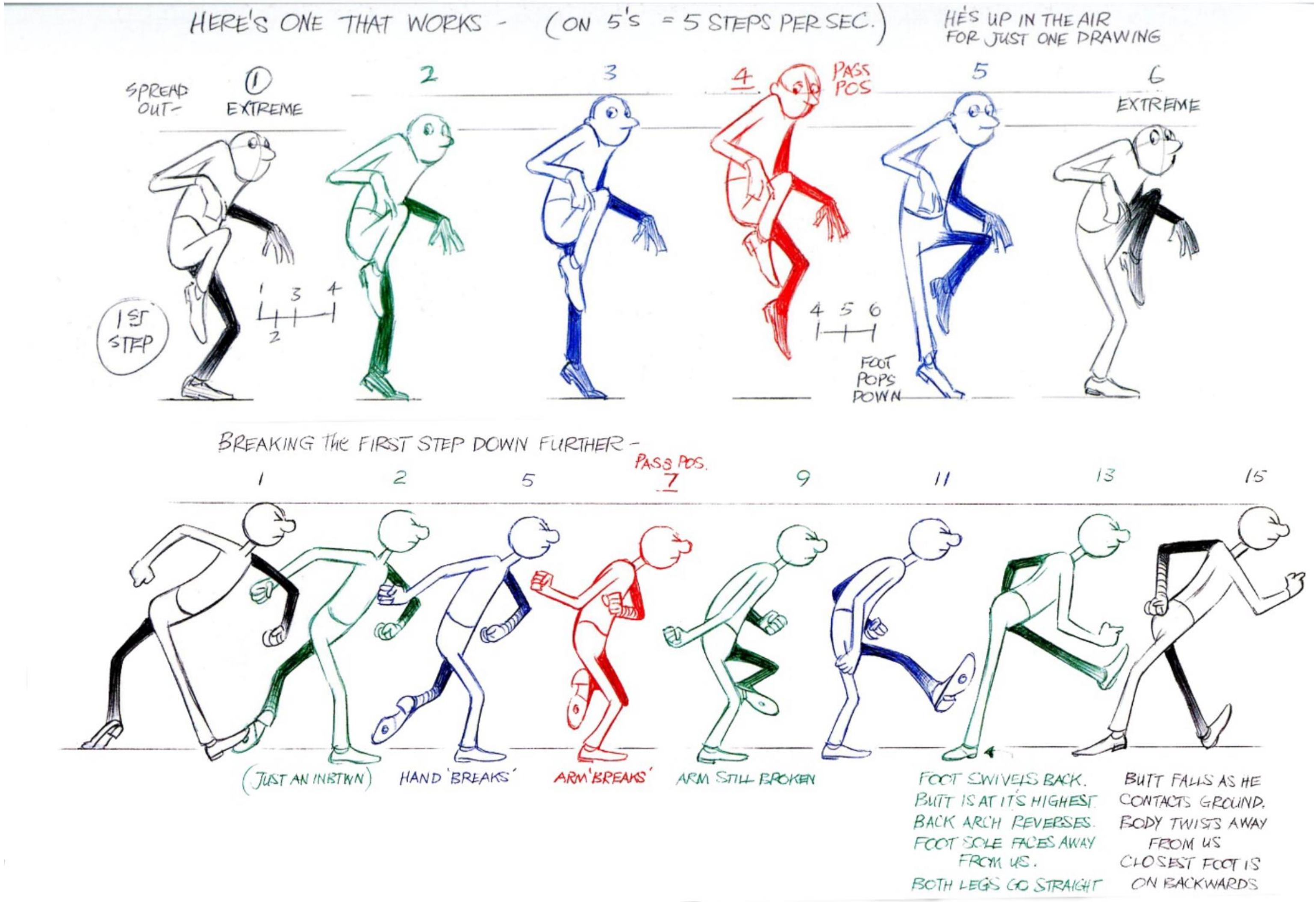
- Motion Capture
  - History and modern mocap systems

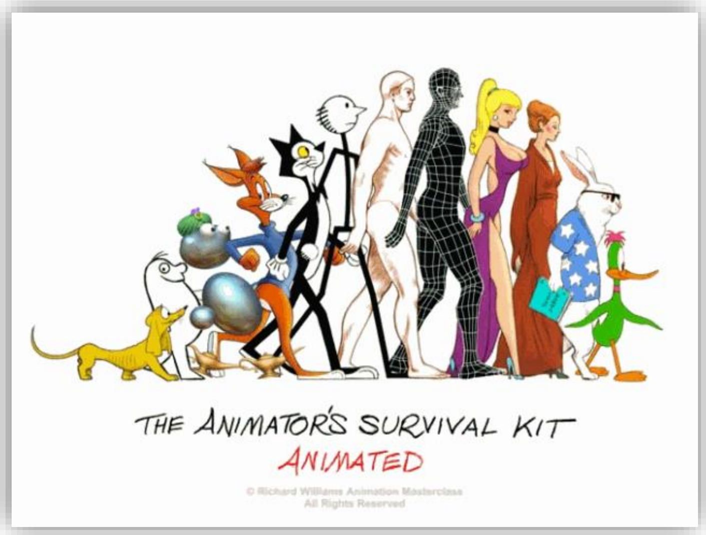
- Motion Synthesis
  - Motion retargeting
  - Motion transition
  - Motion graph

# Motion Capture

How to get motion data?

# Keyframe Animation


HERE'S ONE THAT WORKS - (ON 5'S = 5 STEPS PER SEC.)
HE'S UP IN THE AIR FOR JUST ONE DRAWING
SPREAD OUT-
1 EXTREME
2
3
4 PASS POS
5
6 EXTREME
1ST STEP
1 3 4
2
4 5 6
FOOT POPS DOWN
BREAKING THE FIRST STEP DOWN FURTHER -
1
2
5
PASS POS. 7
9
11
13
15
(JUST AN INBTWN)
HAND 'BREAKS'
ARM 'BREAKS'
ARM STILL BROKEN
FOOT SWIVELS BACK. BUTT IS AT IT'S HIGHEST. BACK ARCH REVERSES. FOOT SOLE FACES AWAY FROM US. BOTH LEGS GO STRAIGHT
BUTT FALLS AS HE CONTACTS GROUND. BODY TWISTS AWAY FROM US. CLOSEST FOOT IS ON BACKWARDS

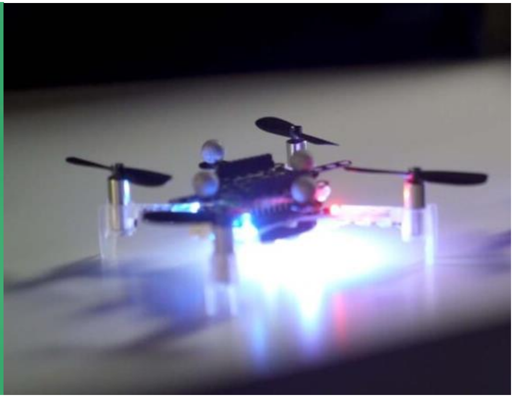

THE ANIMATOR'S SURVIVAL KIT
ANIMATED
© Richard Williams Animation Masterclass
All Rights Reserved

http://www.theanimatorssurvivalkit.com/

# Motion Capture

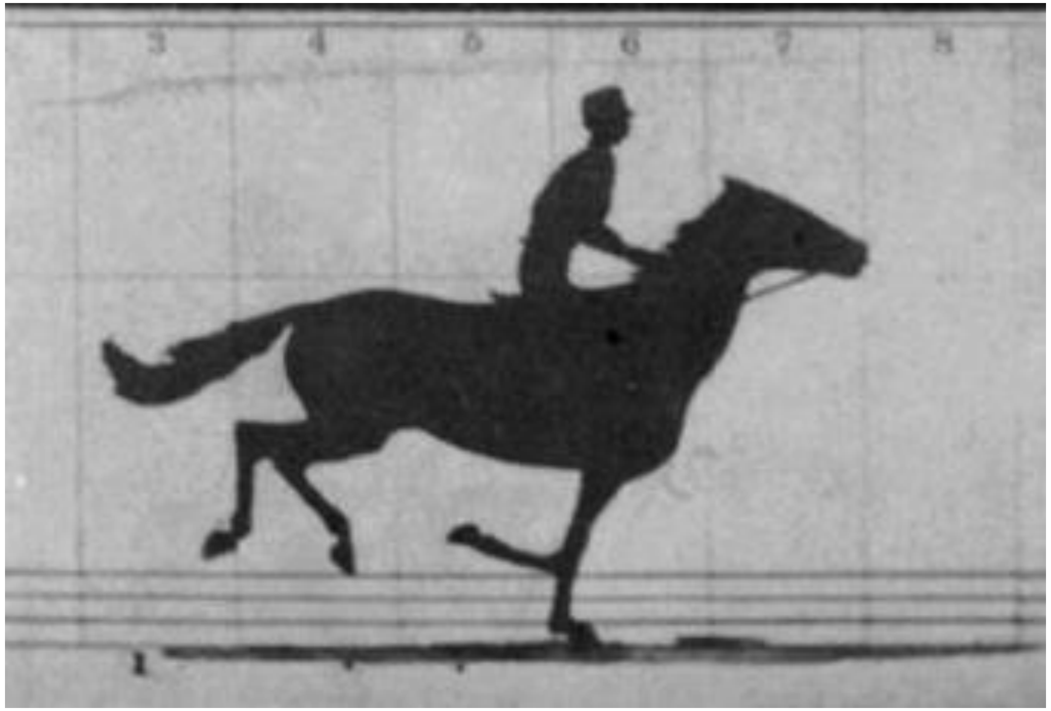
# Motion Capture

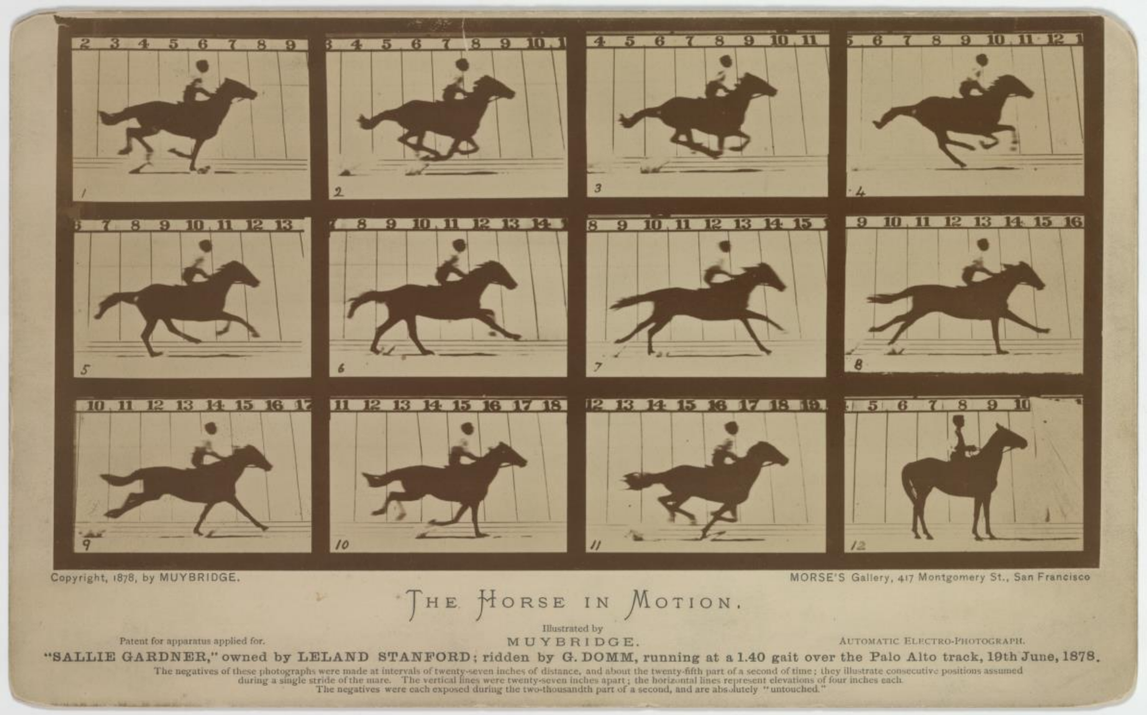
- Digitally recording movements of human, animation, and objects
  - Entertainment: games, films, virtual idols, metaverse
  - Sport: professional training, performance optimization
  - Medicine: orthopedics, injury diagnosis and therapy
  - Robotics: tracking and locating
  - ……



# The History of Mocap

- Study of motions using photography
  - Eadweard Muybridge (1830 –1904)
  - “*The Horse in Motion*” - 1878





Copyright, 1878, by MUYBRIDGE.

MORSE'S Gallery, 417 Montgomery St., San Francisco

THE HORSE IN MOTION.

Illustrated by
MUYBRIDGE.

Patent for apparatus applied for.

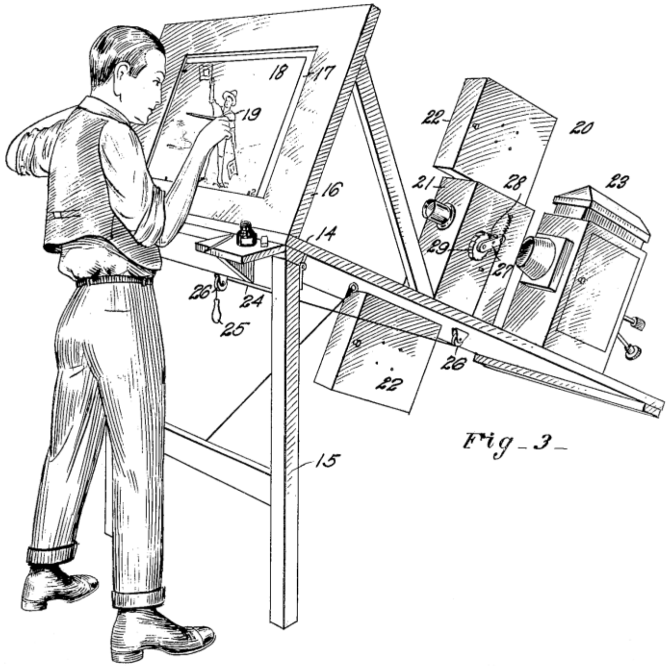
AUTOMATIC ELECTRO-PHOTOGRAPH.

“SALLIE GARDNER,” owned by LELAND STANFORD; ridden by G. DOMM, running at a 1.40 gait over the Palo Alto track, 19th June, 1878.

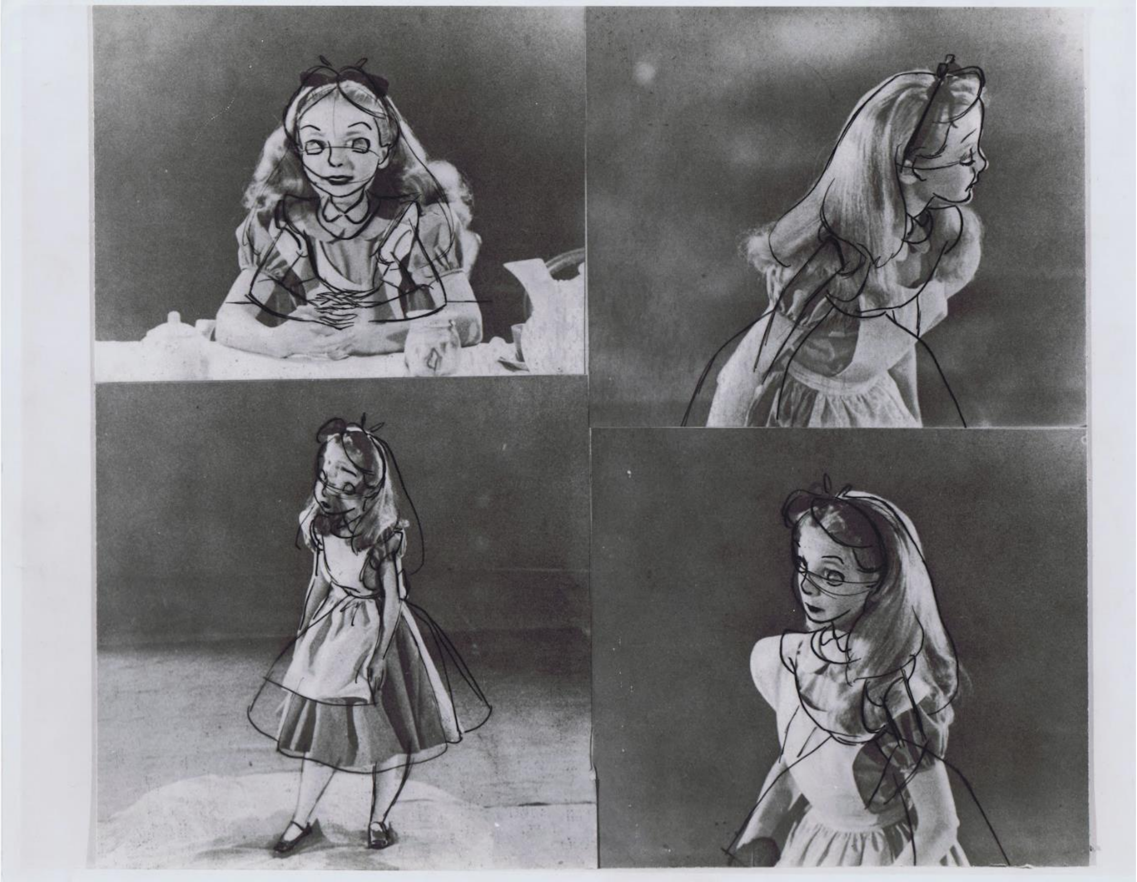
The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed during a single stride of the mare. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each. The negatives were each exposed during the two-thousandth part of a second, and are absolutely “untouched.”

# The History of Mocap

- Rotoscoping (~1914)
  - Max Fleischer (1883 – 1972)



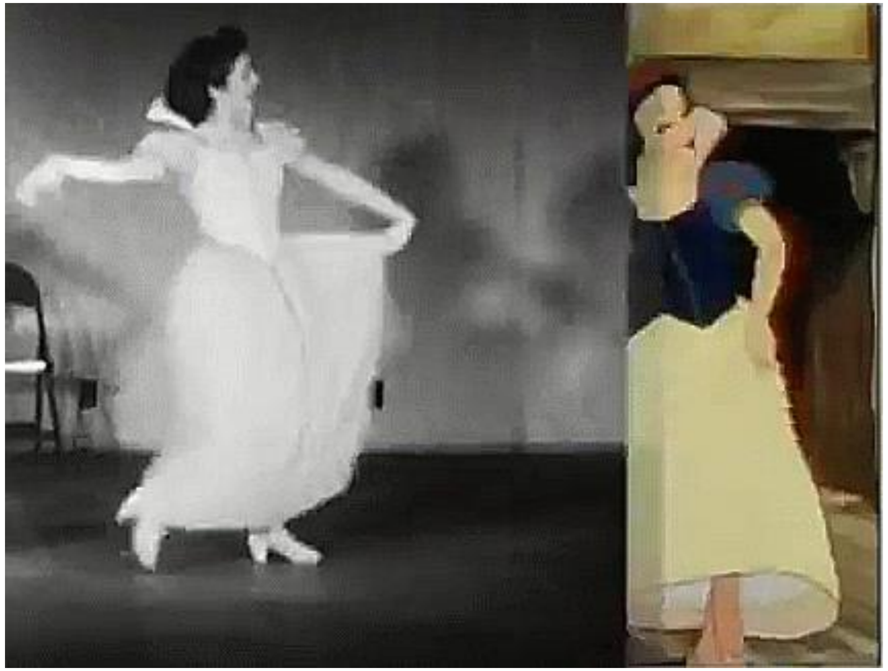
[US patent 1242674]



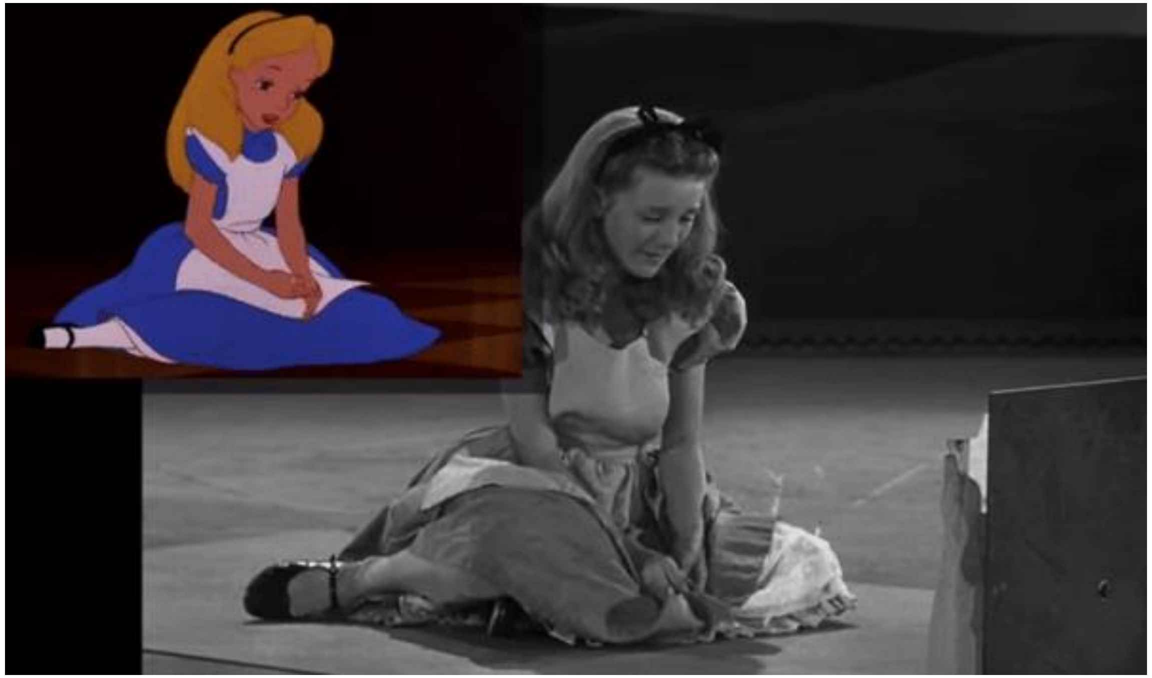
*Alice in Wonderland*, 1951

# The History of Mocap

- Rotoscoping (~1914)
  - Rotoscope in Animation



*Snow White and the Seven Dwarfs*, 1939



*Alice in Wonderland*, 1951

# The History of Mocap

- Rotoscoping (~1914)
  - Rotoscope in Animation

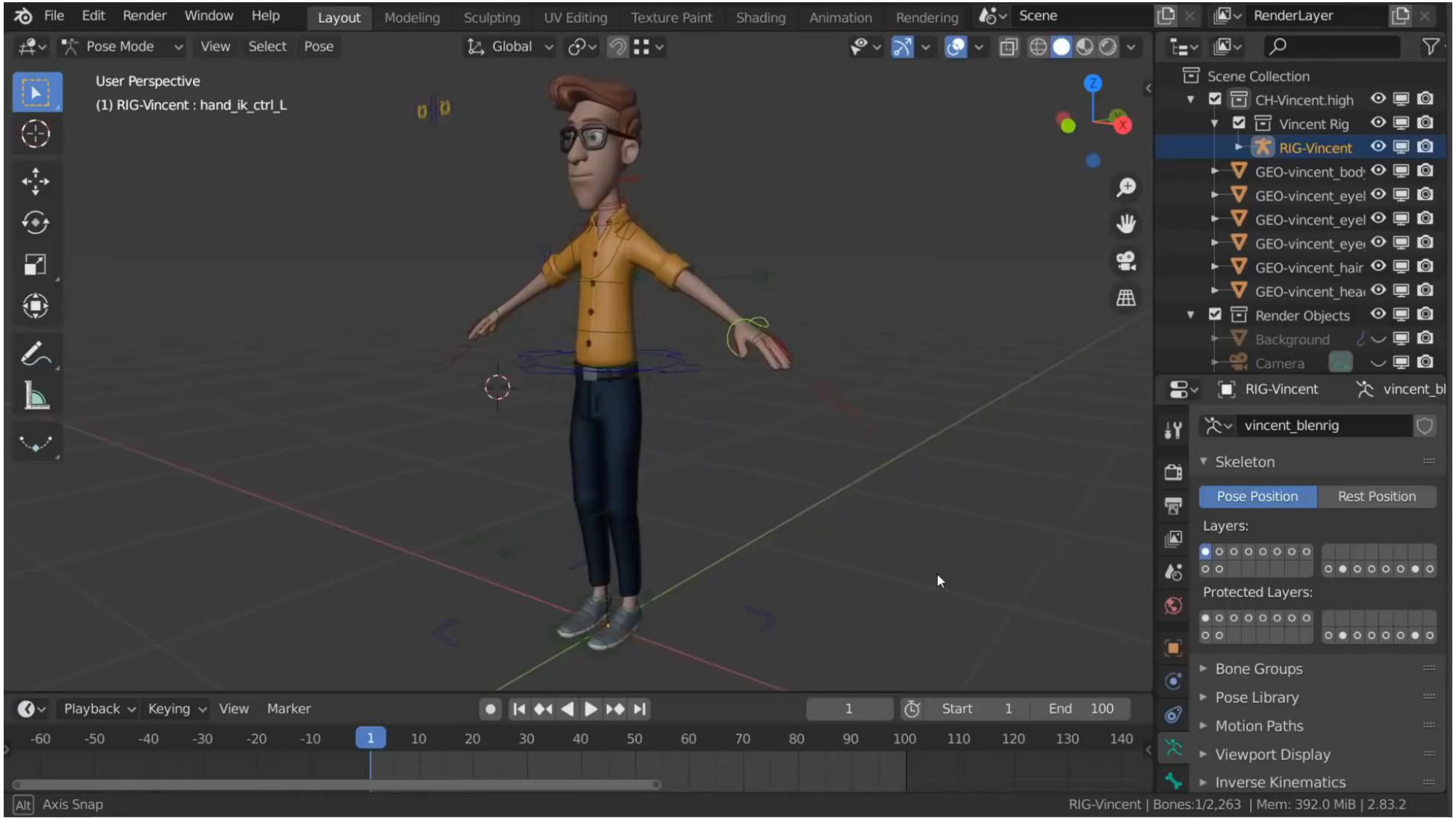
Snow White (1937)

# The History of Mocap

- Rotoscoping (~1914)
    - Rotoscope in Animation

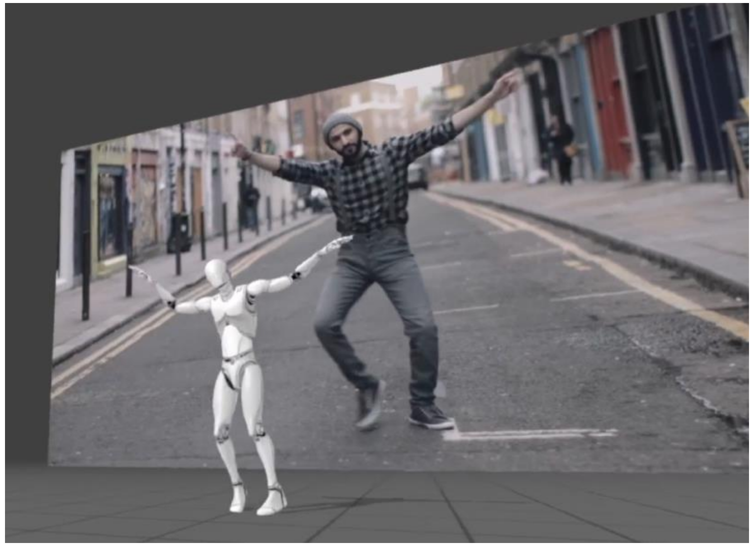
*The Princess of Iron Fan (铁扇公主)*, Wan brothers (万氏兄弟), 1941

# Rotoscoping in 3D



**How to Animate 3D Characters in 1 Minute** https://www.youtube.com/watch?v=TjJLIuFKA20

# Modern Mocap System



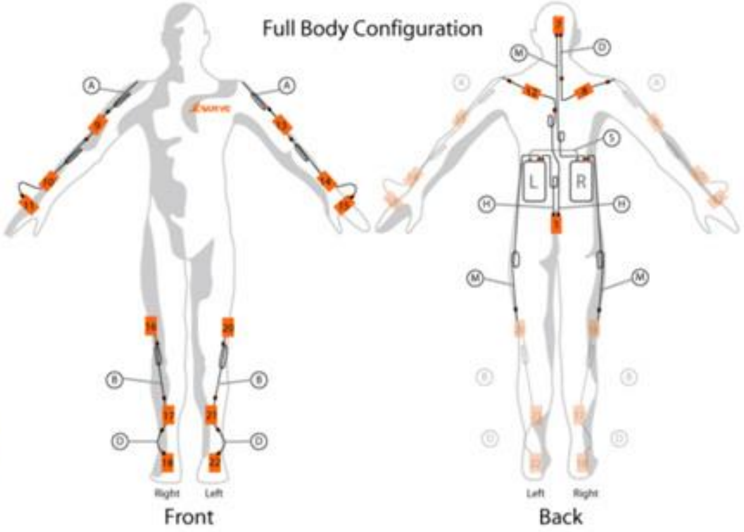
# Mechanical Mocap

- Exoskeleton

# Inertial Mocap

- Inertial Measurement Unit (IMU)
  - Accelerometers (3dof) + axis gyroscope (3dof)
- Optionally other sensors

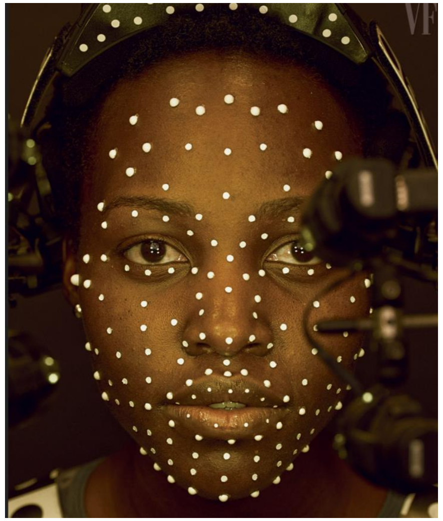
(a)



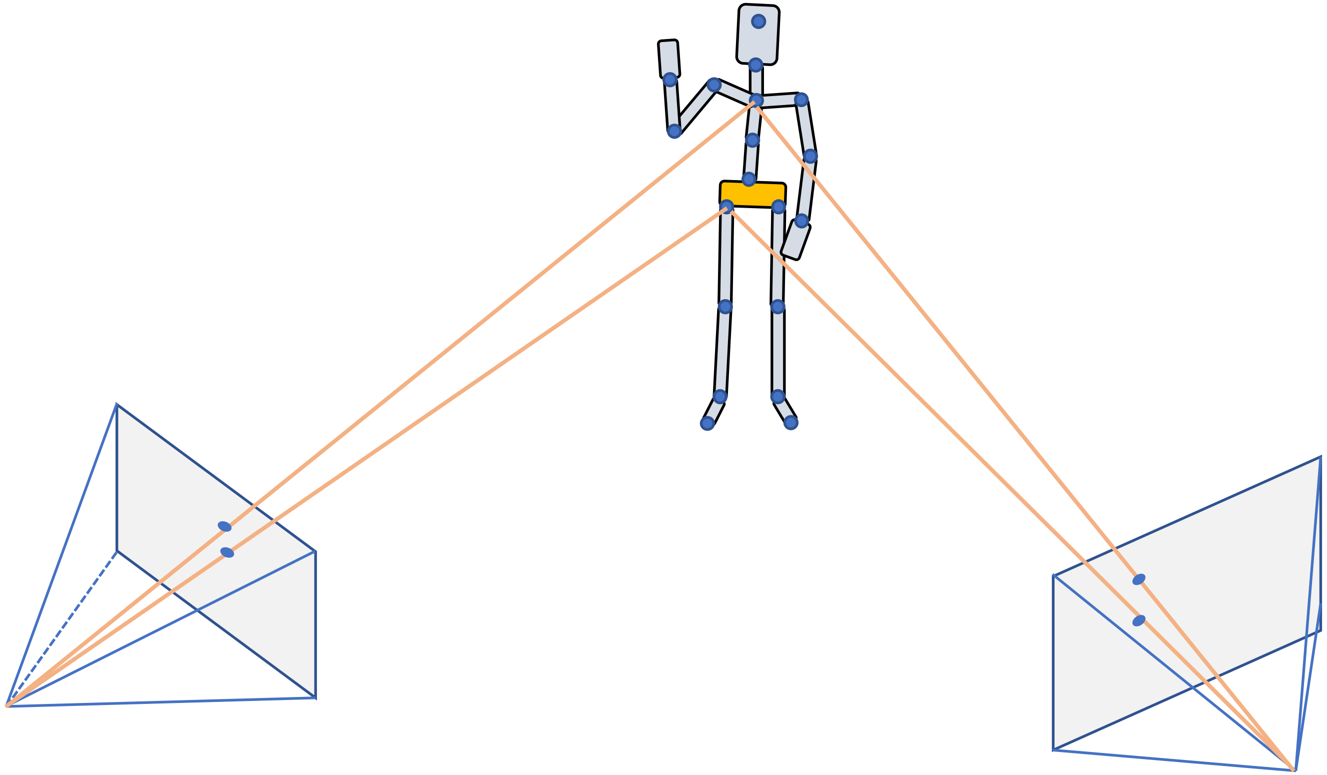
(b)

# Optical Mocap

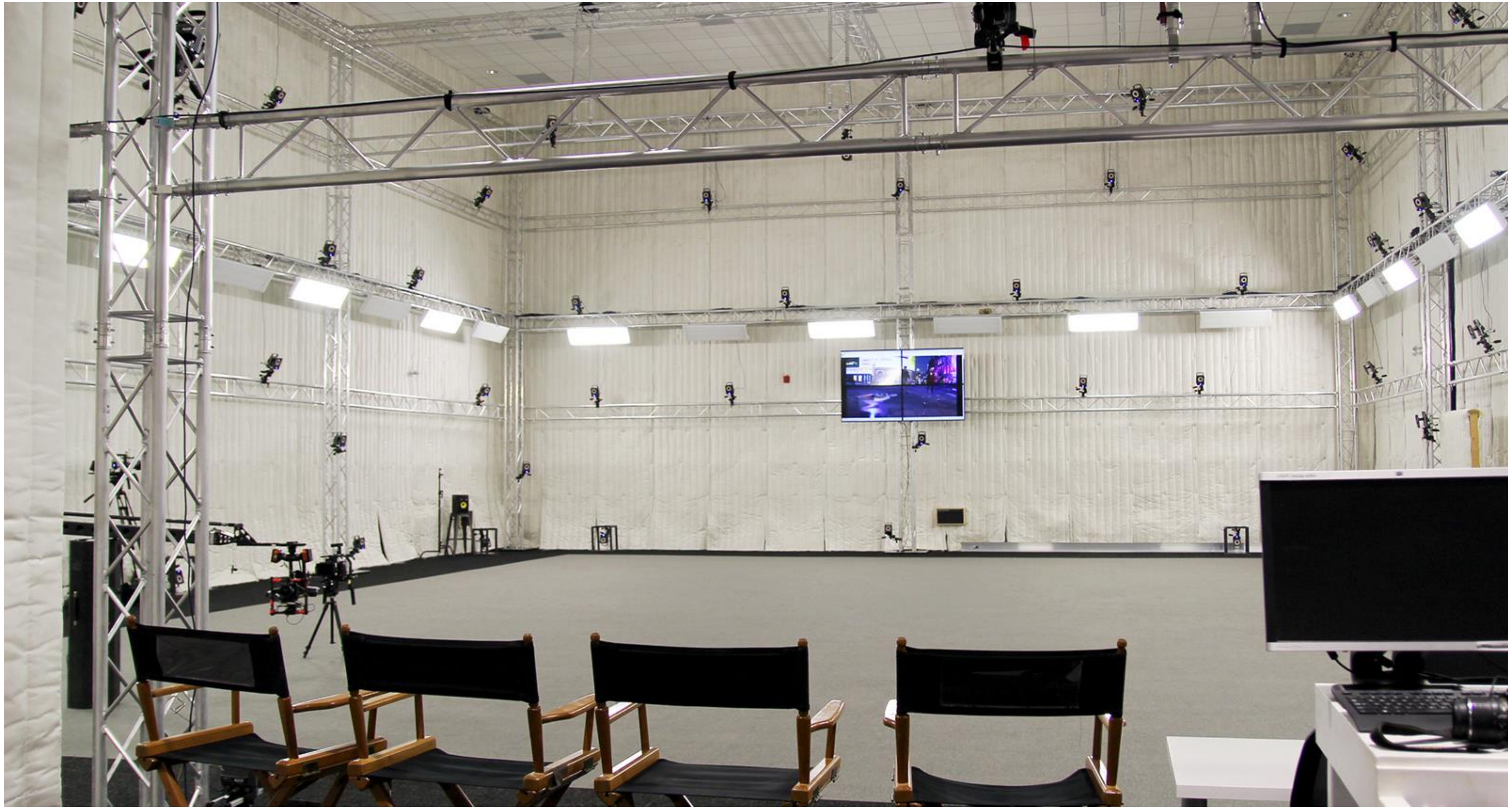
- Reflective/light-emitting markers
- Multi-view geometry
- Solve body motions based on marker positions



# Optical Mocap

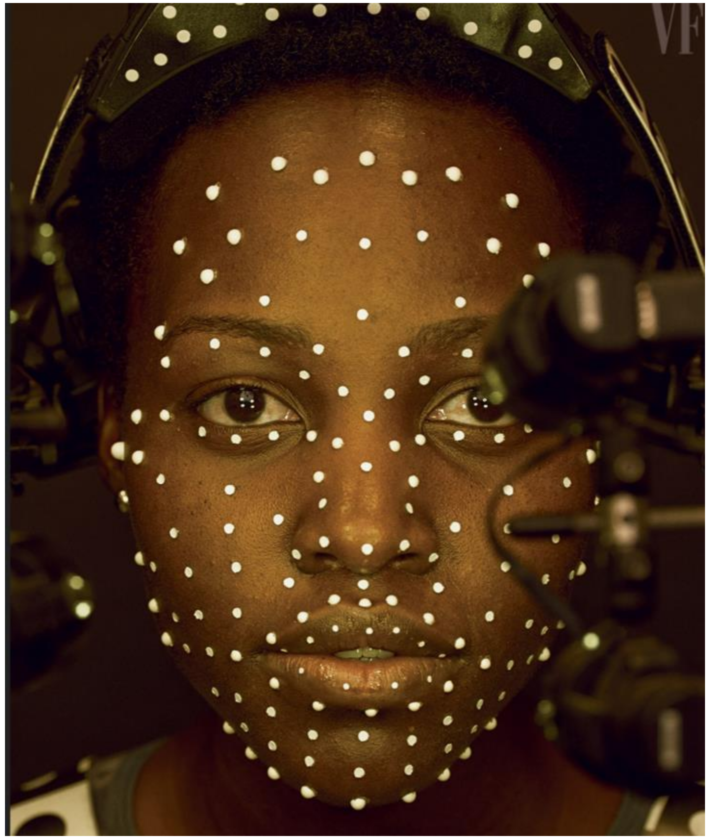


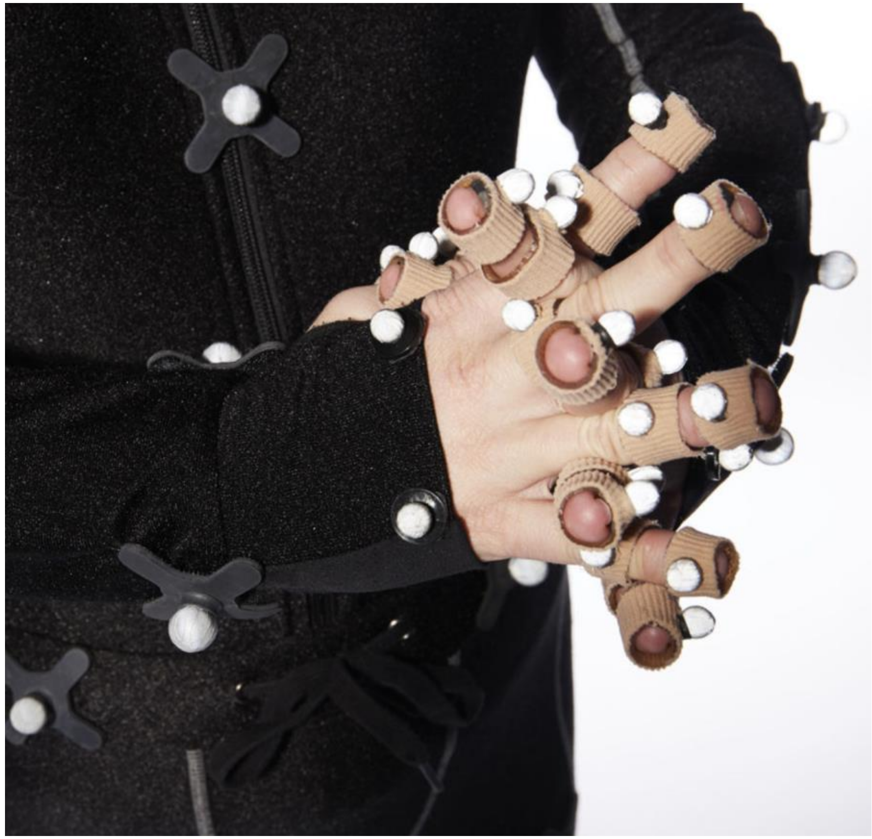
# Optical Mocap

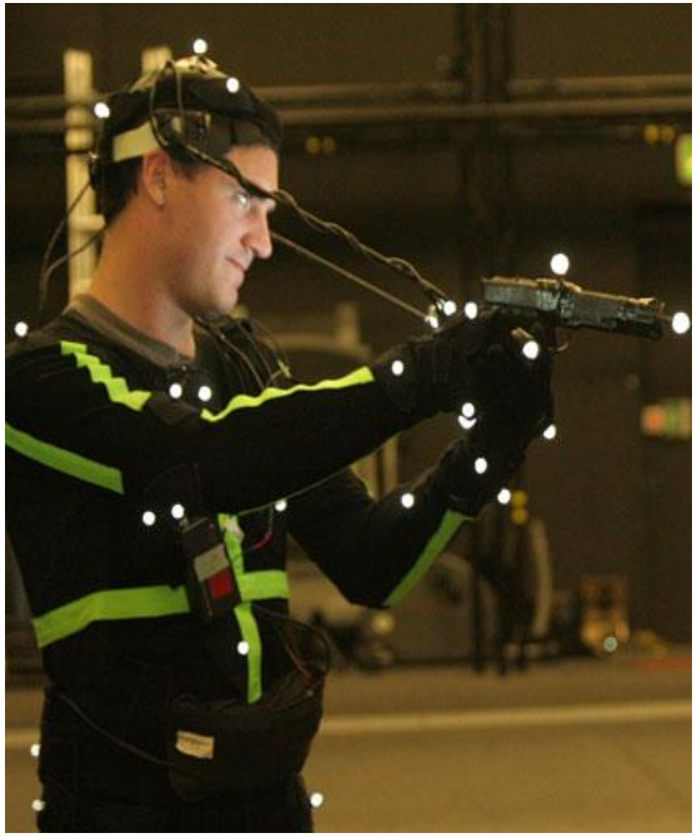


https://toronto.ubisoft.com/performance-capture/

# Optical Mocap



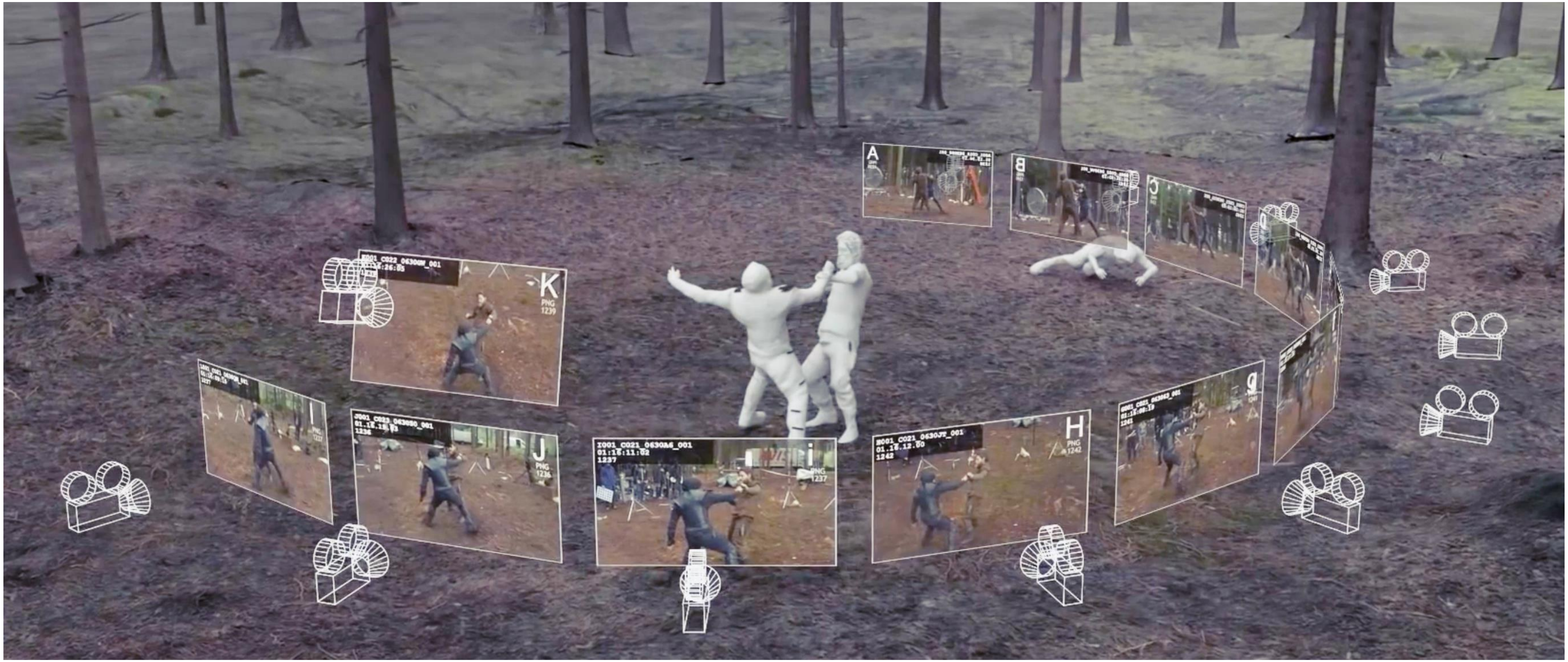




# Optical Mocap

[Holden 2018 - Robust Solving of Optical Motion Capture Data by Denoising]

# Markerless Mocap with Multiple Cameras

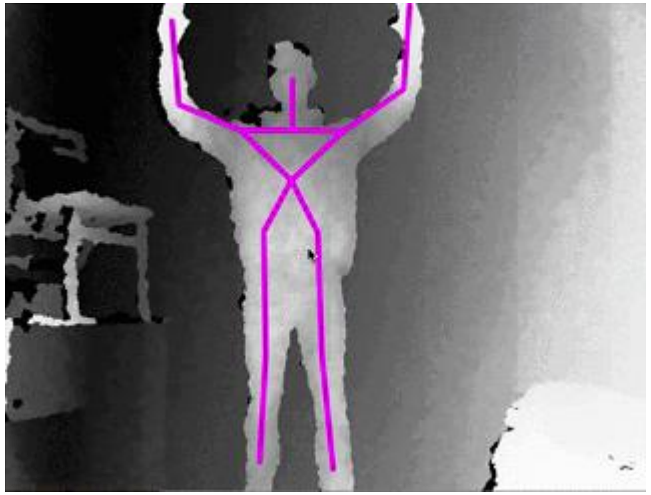


https://captury.com/

# Markerless Mocap with Multiple Cameras

https://www.theiamarkerless.ca

# Markerless Mocap with Depth Cameras

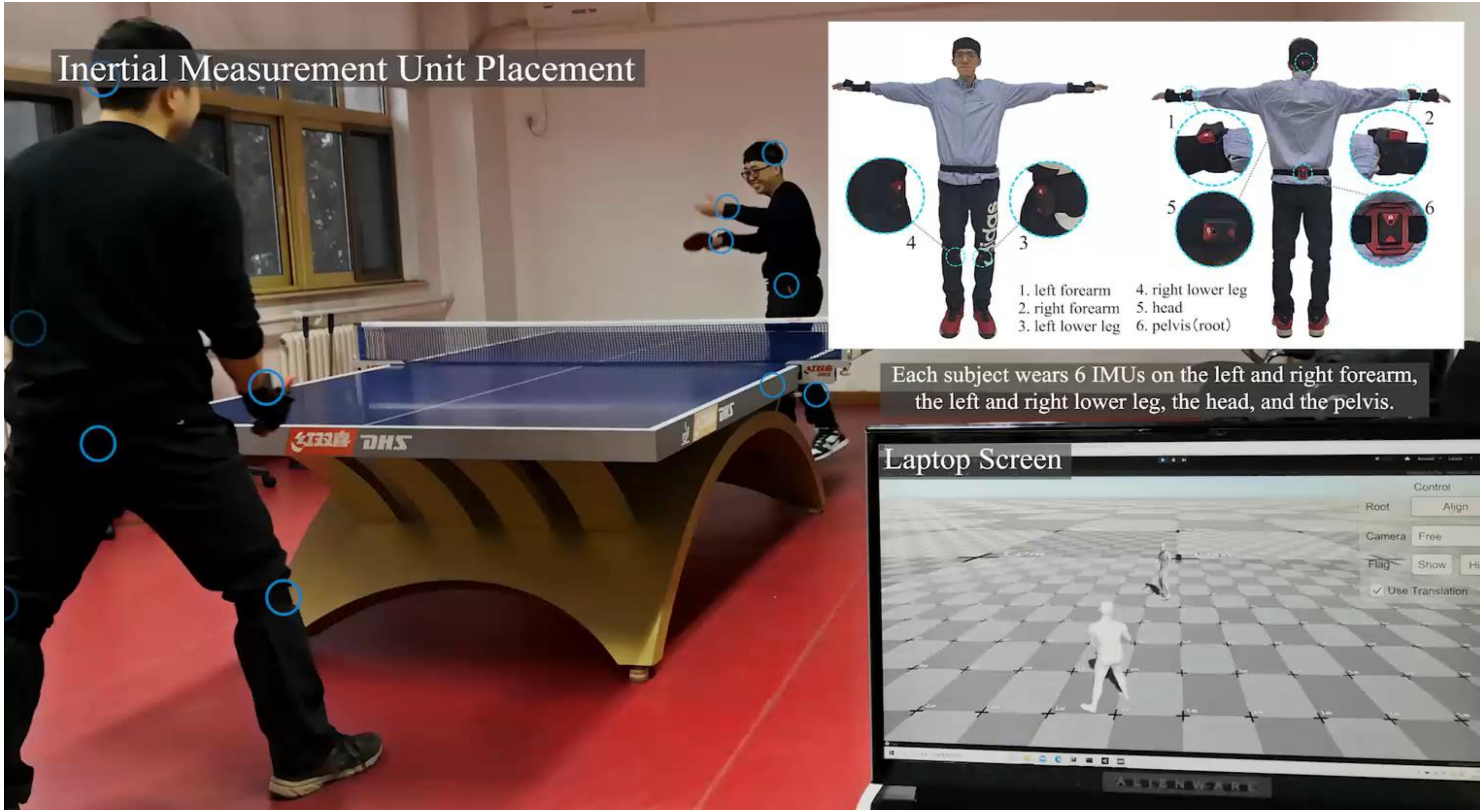


# Motion Estimation with Monocular Videos

[OpenPose, 2D Pose estimation]

[3D Video-based Pose estimation, source: DeepMotion Inc.]

# Motion Estimation with Sparse Sensor



[Yi et al. 2021. TransPose: Real-time 3D Human Translation and Pose Estimation with **Six Inertial Sensors**]

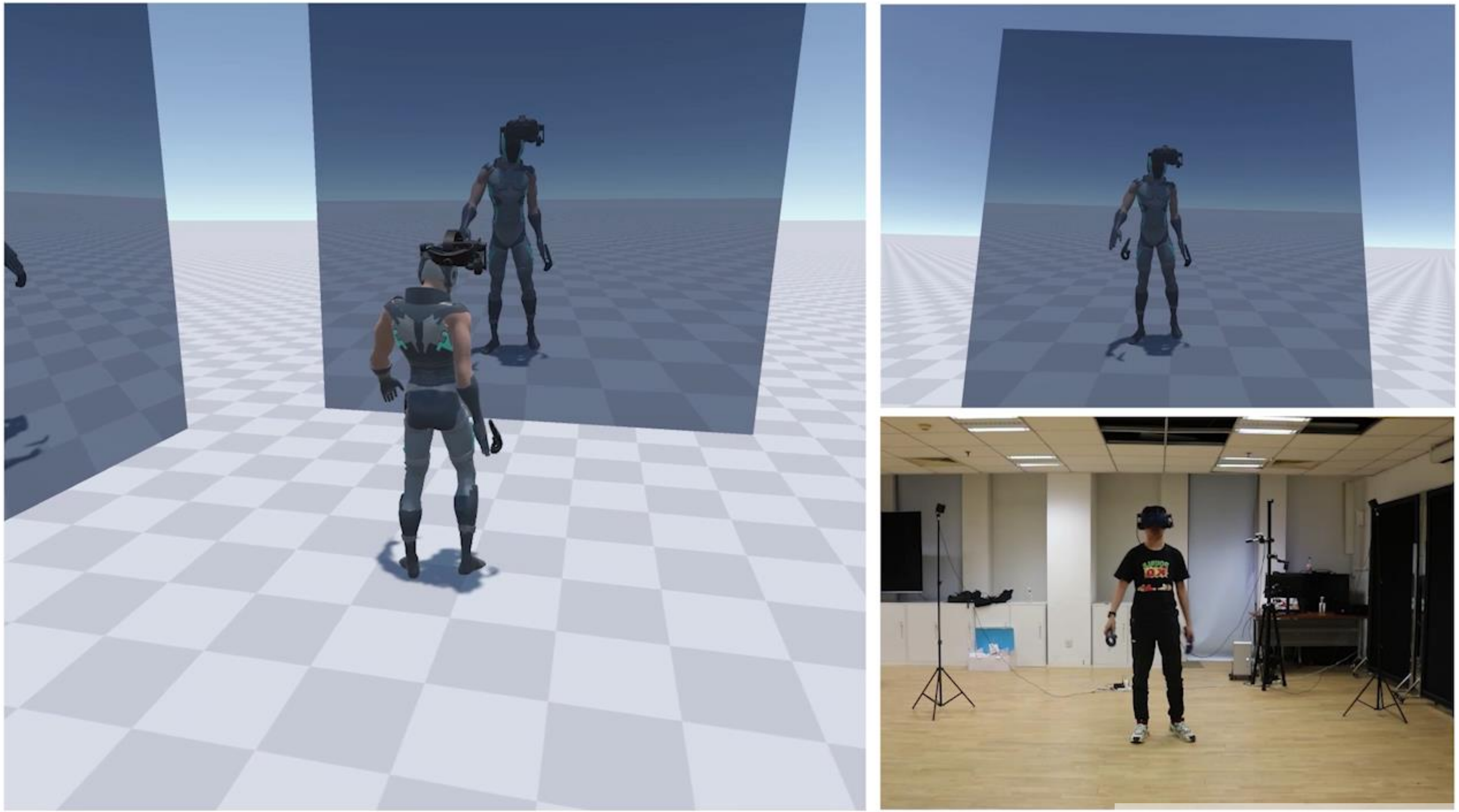
# Motion Estimation with Sparse Sensor

[Meta]

# Motion Estimation with Sparse Sensor

[Meta]

# Motion Estimation with Sparse Sensor



[Ye et al. 2022: Neural3Points]

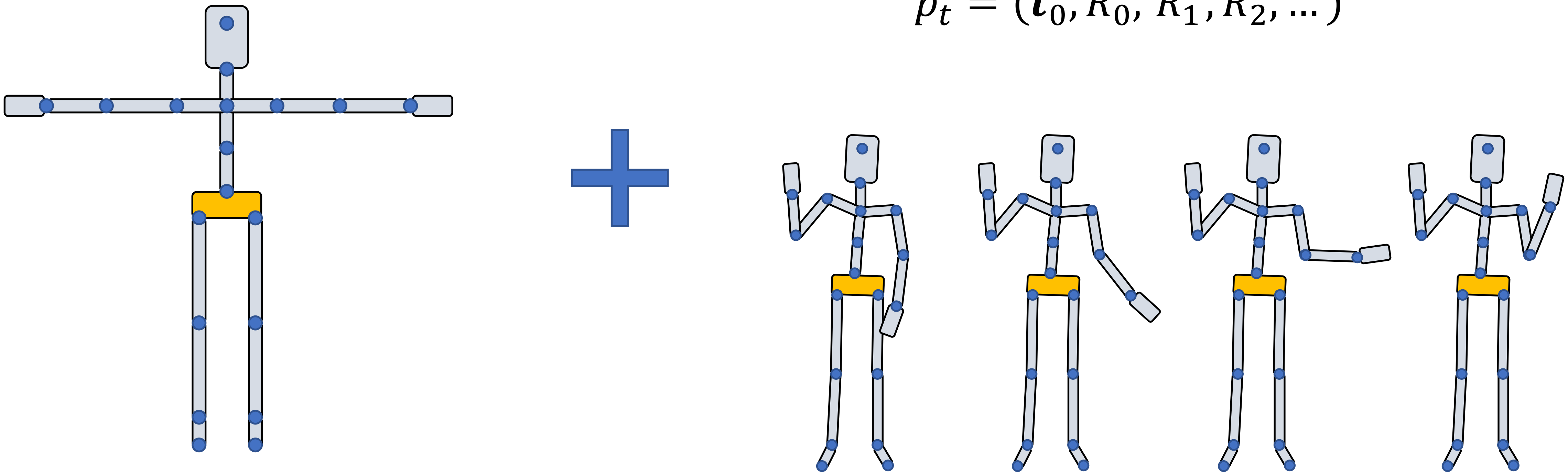
# Motion Synthesis

How to use motion data?

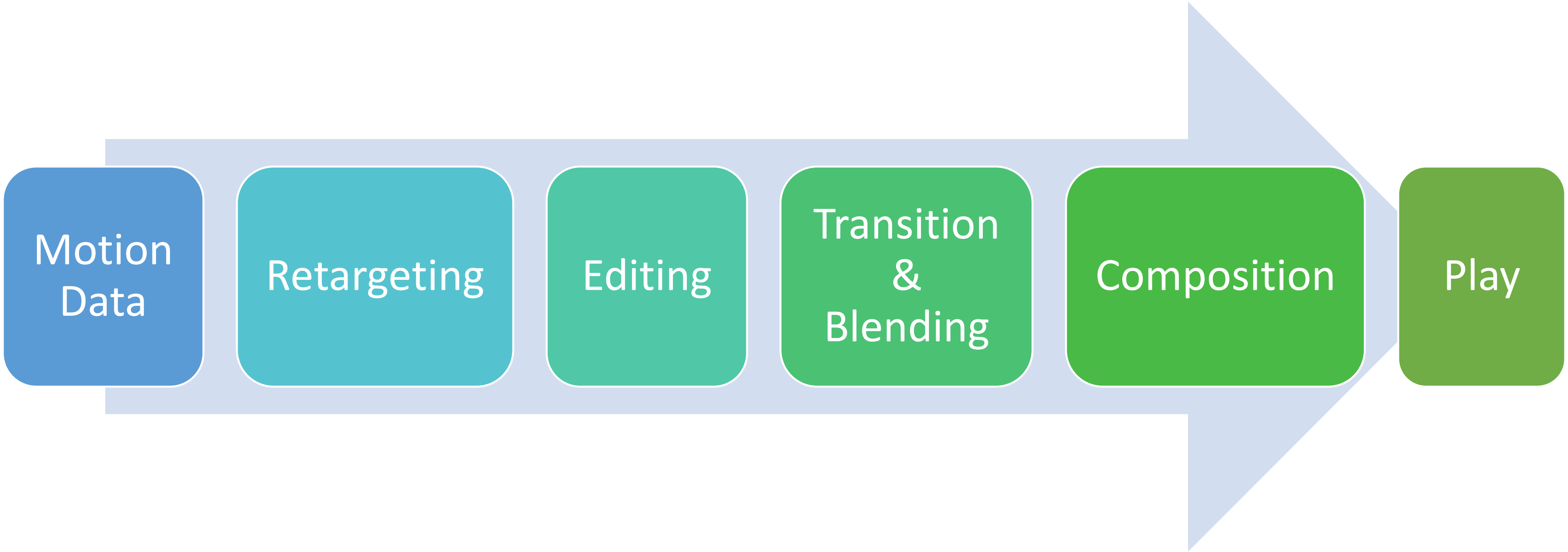
# Motion Data

$$\{p_t\}, t = 1, \ldots, N$$

$$p_t = (\boldsymbol{t}_0, R_0, R_1, R_2, \ldots)$$

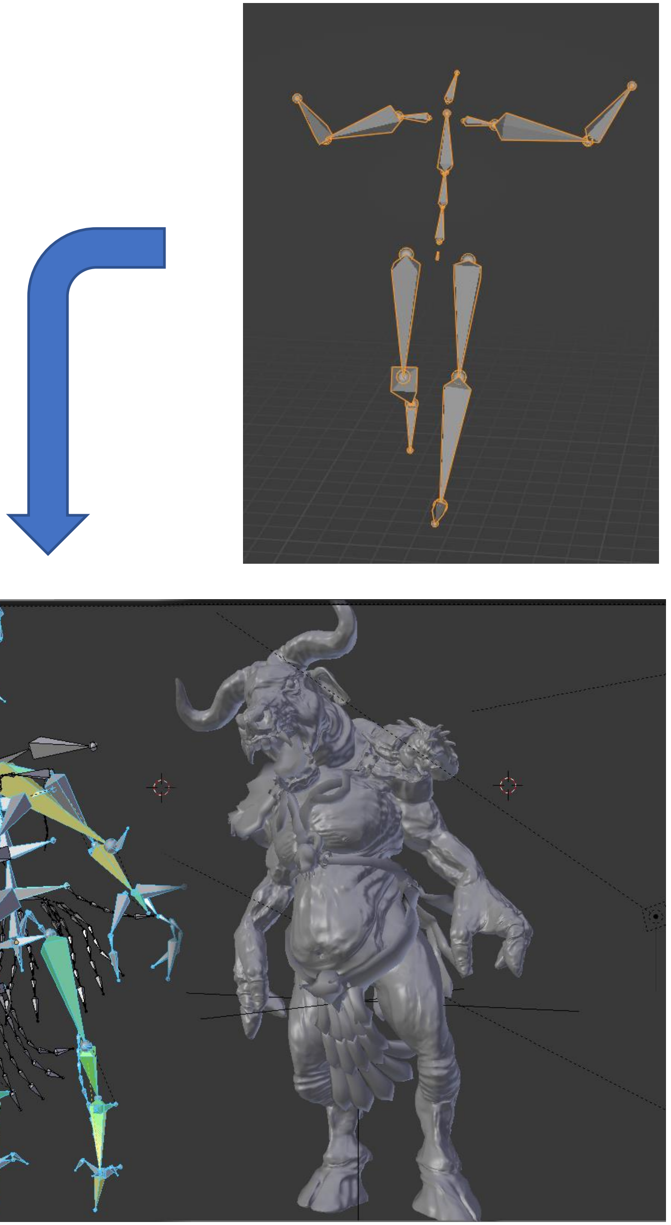


# Using Motion Data



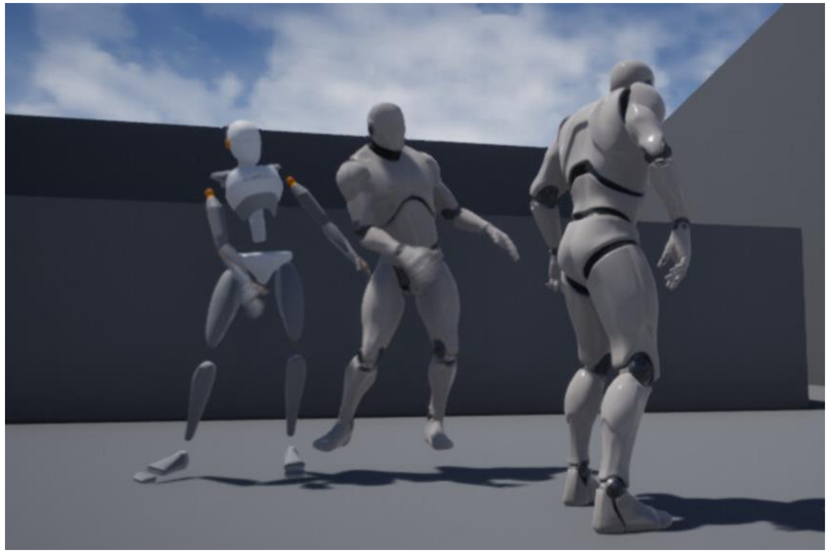
# Motion Retargeting

- Retarget a motion to drive a character with
  - Different number of bones
  - Different bone names
  - Different reference pose
  - Different bone ratios
  - Different skeletal structure
  - ……



# Motion Retargeting

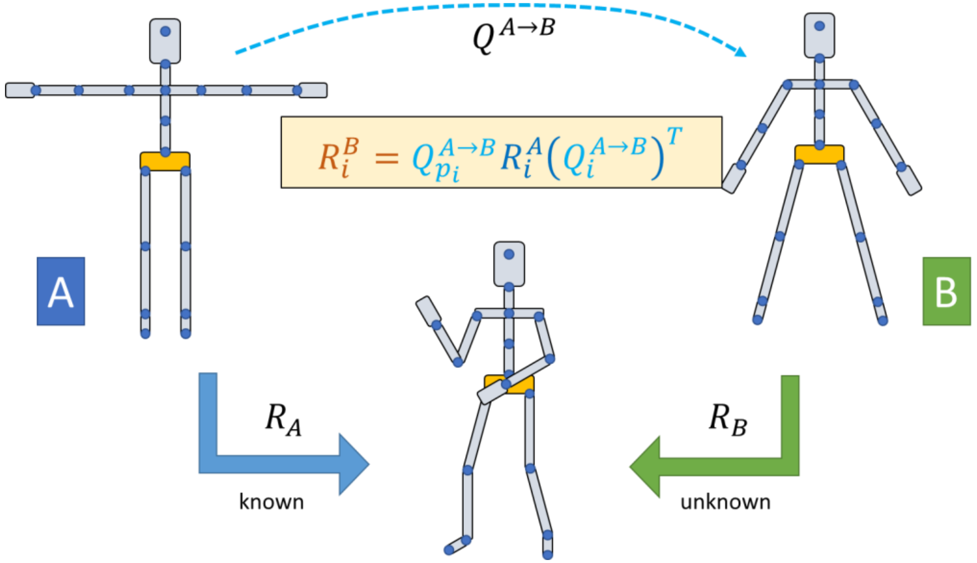
- Retarget a motion to drive a character with
  - Different number of bones
  - Different bone names
  - Different reference pose
  - Different bone ratios
  - Different skeletal structure
  - ……

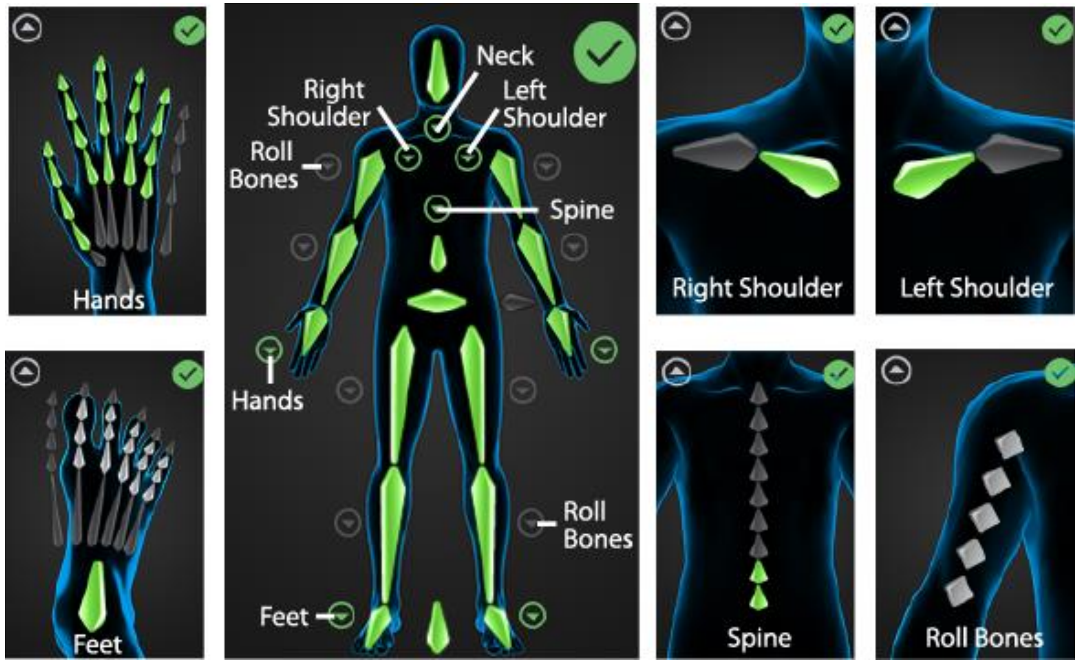


[Villegas et al. 2021, Contact-Aware Retargeting of Skinned Motion]

# Motion Retargeting

- A possible retargeting pipeline
  - Map bone names
  - Scale translations
  - Copy or retarget joint rotations to fix reference pose
  - Postprocessing with IK
    - Foot-skating
    - Self penetration
  - ……

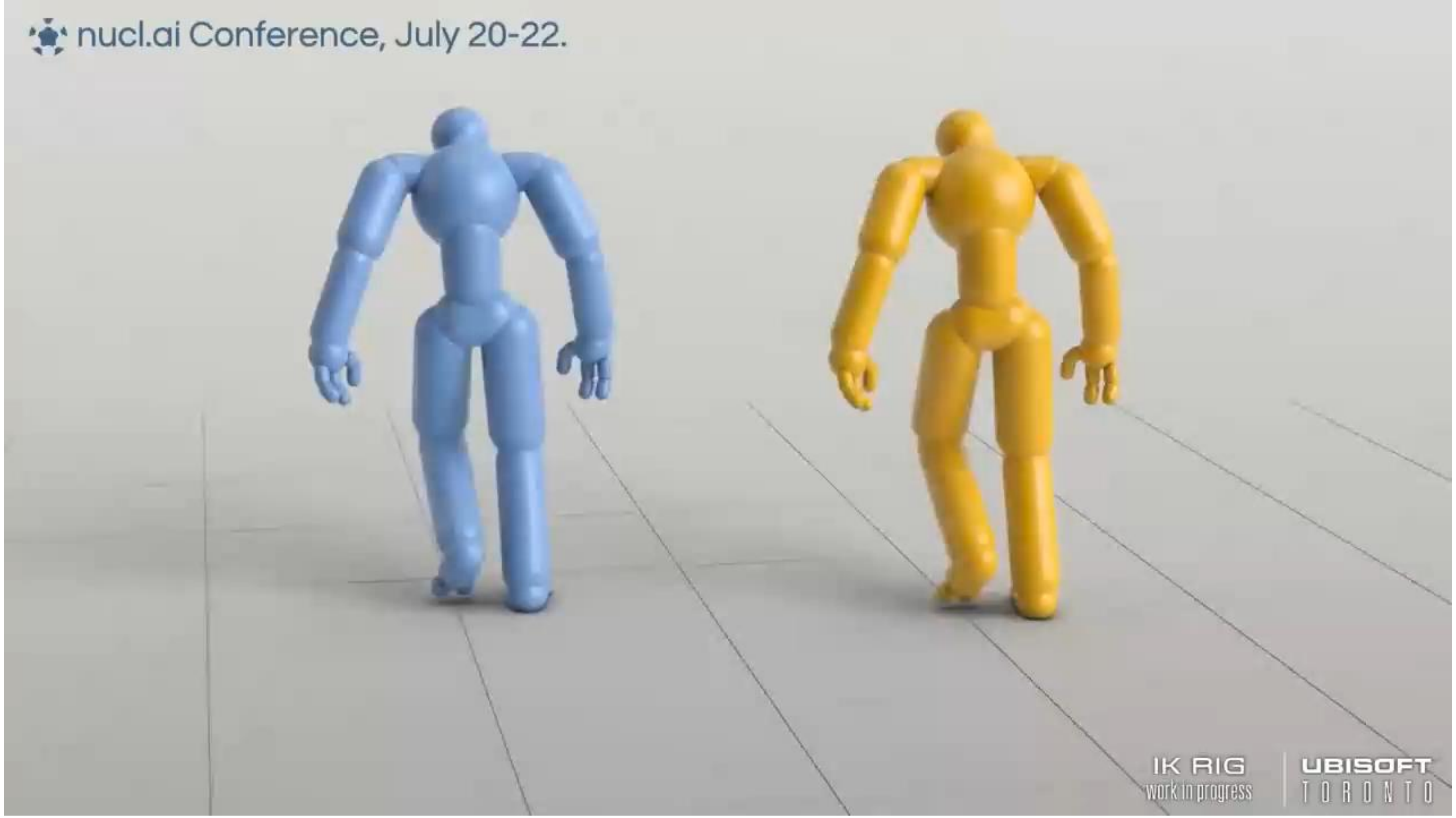

$Q^{A\to B}$
$R_i^B = Q_{p_i}^{A\to B} R_i^A \left(Q_i^{A\to B}\right)^T$
A
B
$R_A$
known
$R_B$
unknown



*HumanIK* in Autodesk Maya

# Motion Retargeting

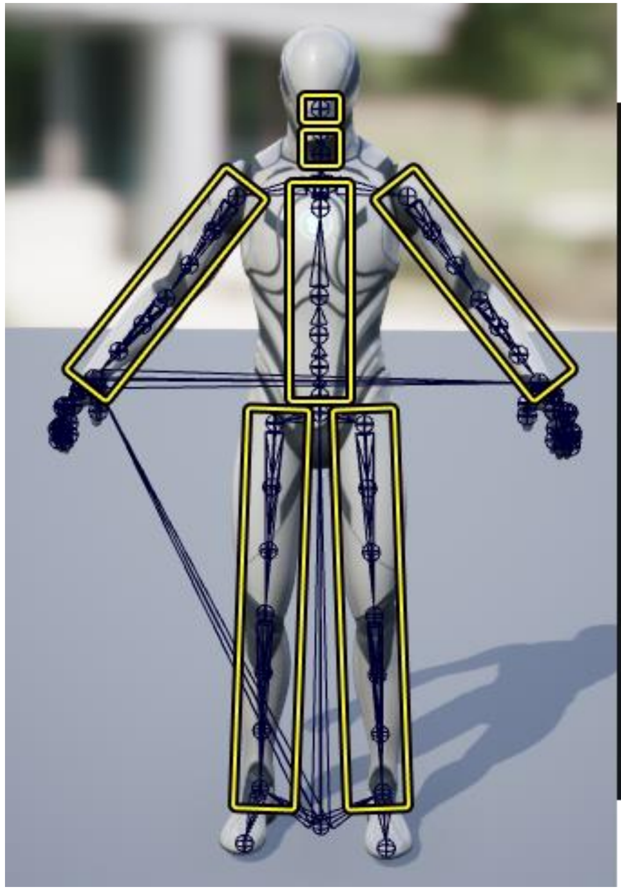
- IK Rig

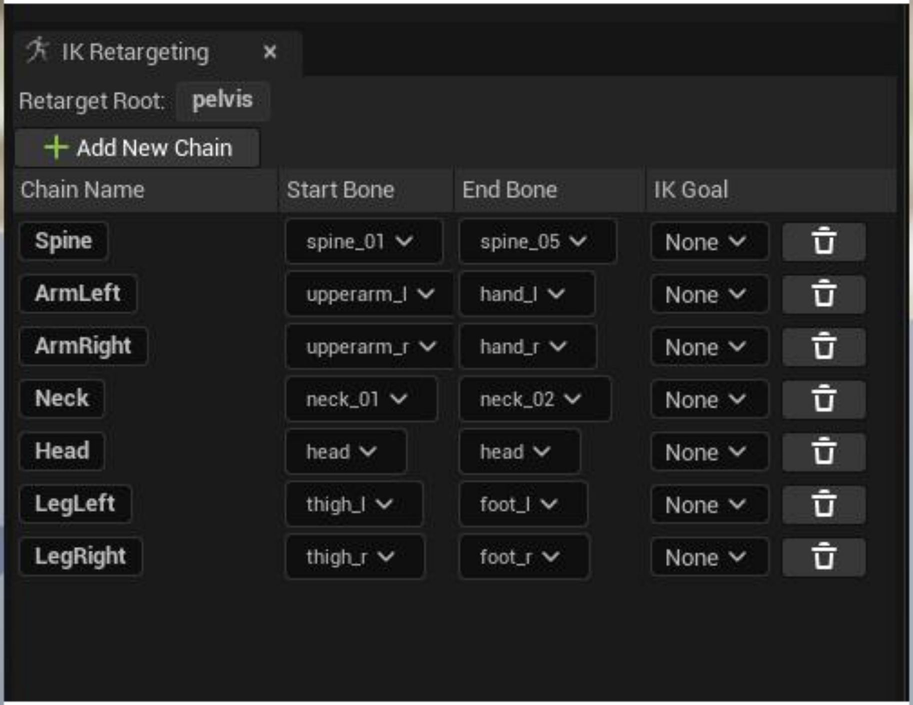


**nucl.ai Conference: Ubisoft Toronto "IK Rig" Prototype** https://www.youtube.com/watch?v=V4TQSeUpH3Q

# Motion Retargeting

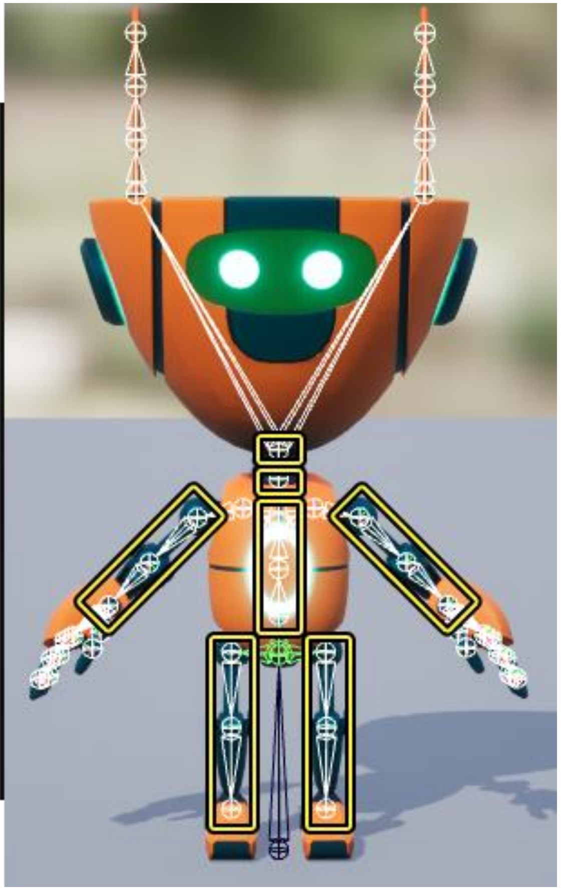
- IK Rig in Unreal





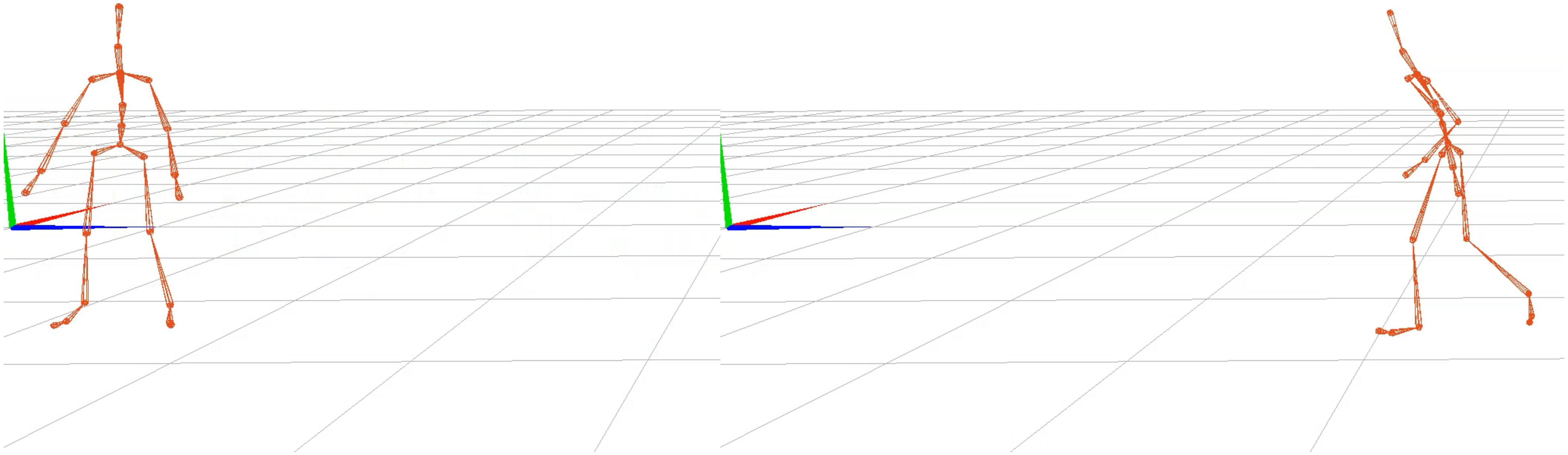
Retarget Root: pelvis

| Chain Name | Start Bone | End Bone | IK Goal |
|---|---|---|---|
| Spine | spine_01 | spine_05 | None |
| ArmLeft | upperarm_l | hand_l | None |
| ArmRight | upperarm_r | hand_r | None |
| Neck | neck_01 | neck_02 | None |
| Head | head | head | None |
| LegLeft | thigh_l | foot_l | None |
| LegRight | thigh_r | foot_r | None |

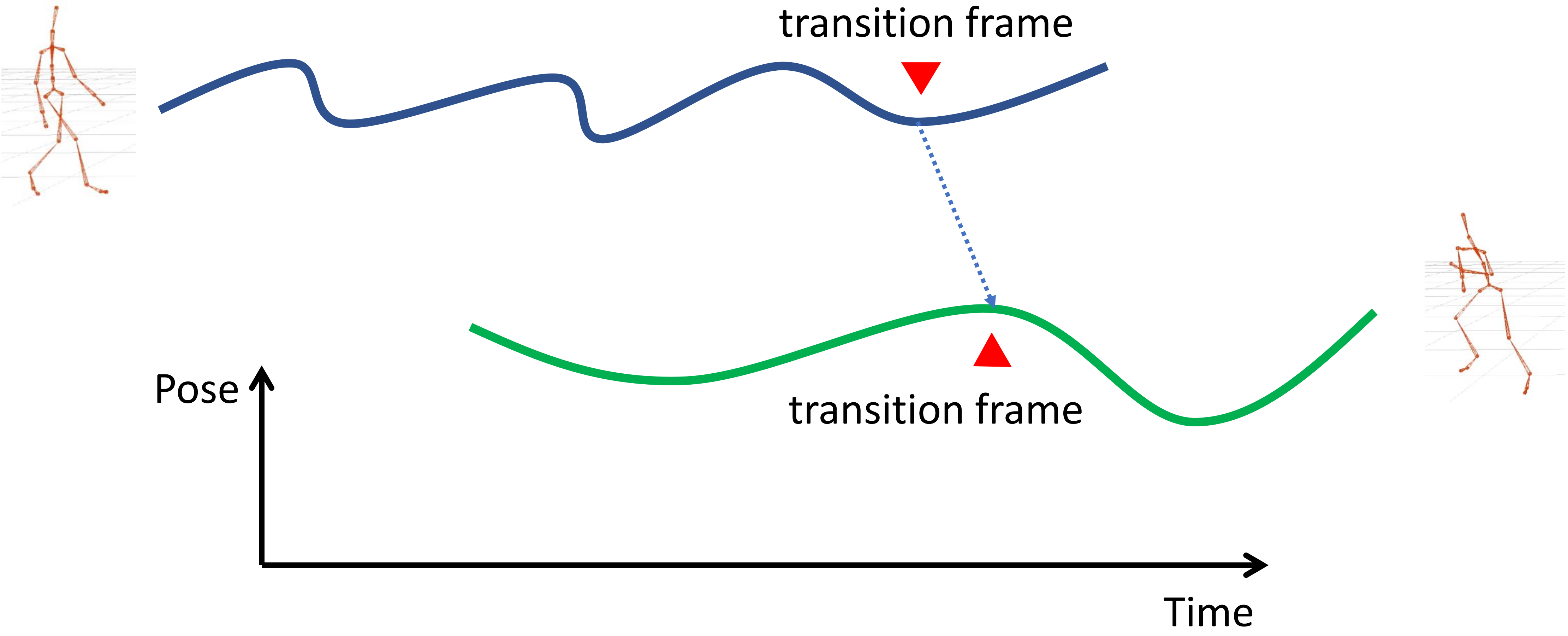


https://docs.unrealengine.com/5.0/en-US/ik-rig-animation-retargeting-in-unreal-engine/

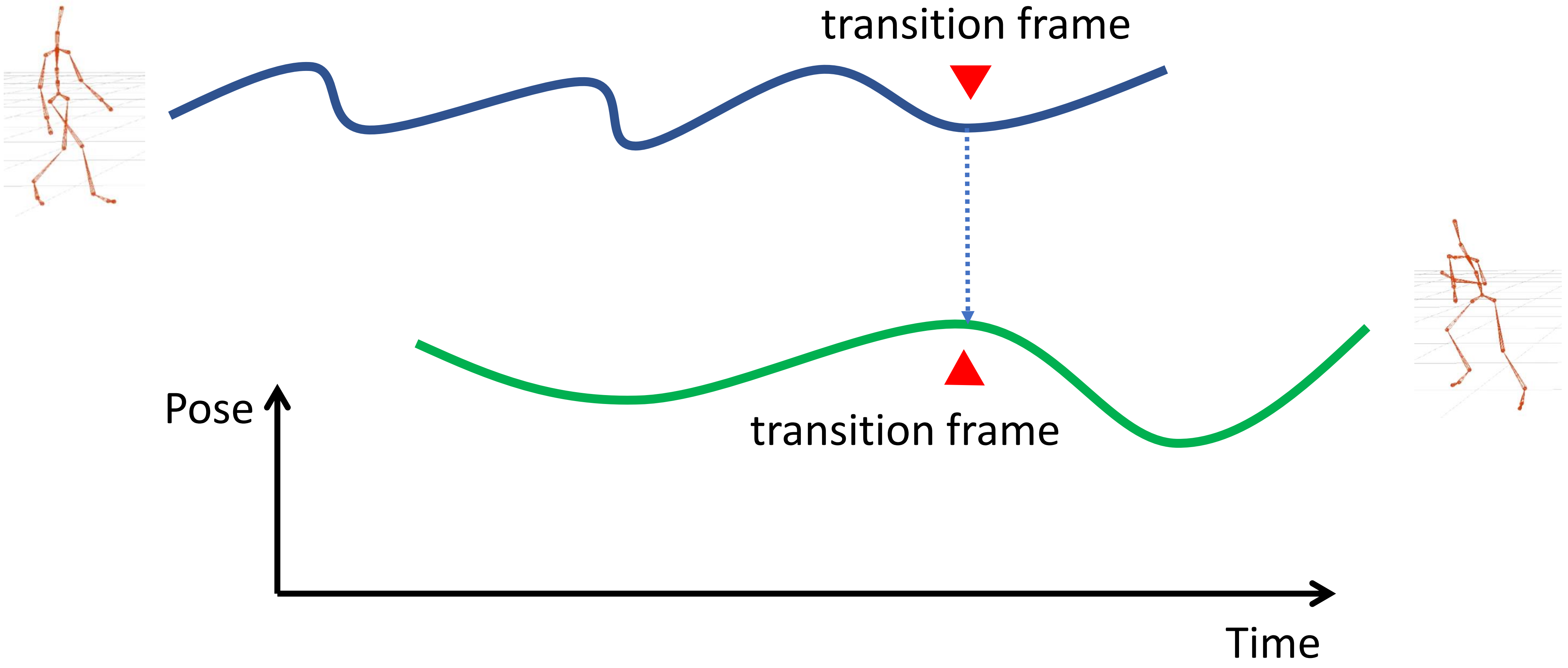
# Motion Transition



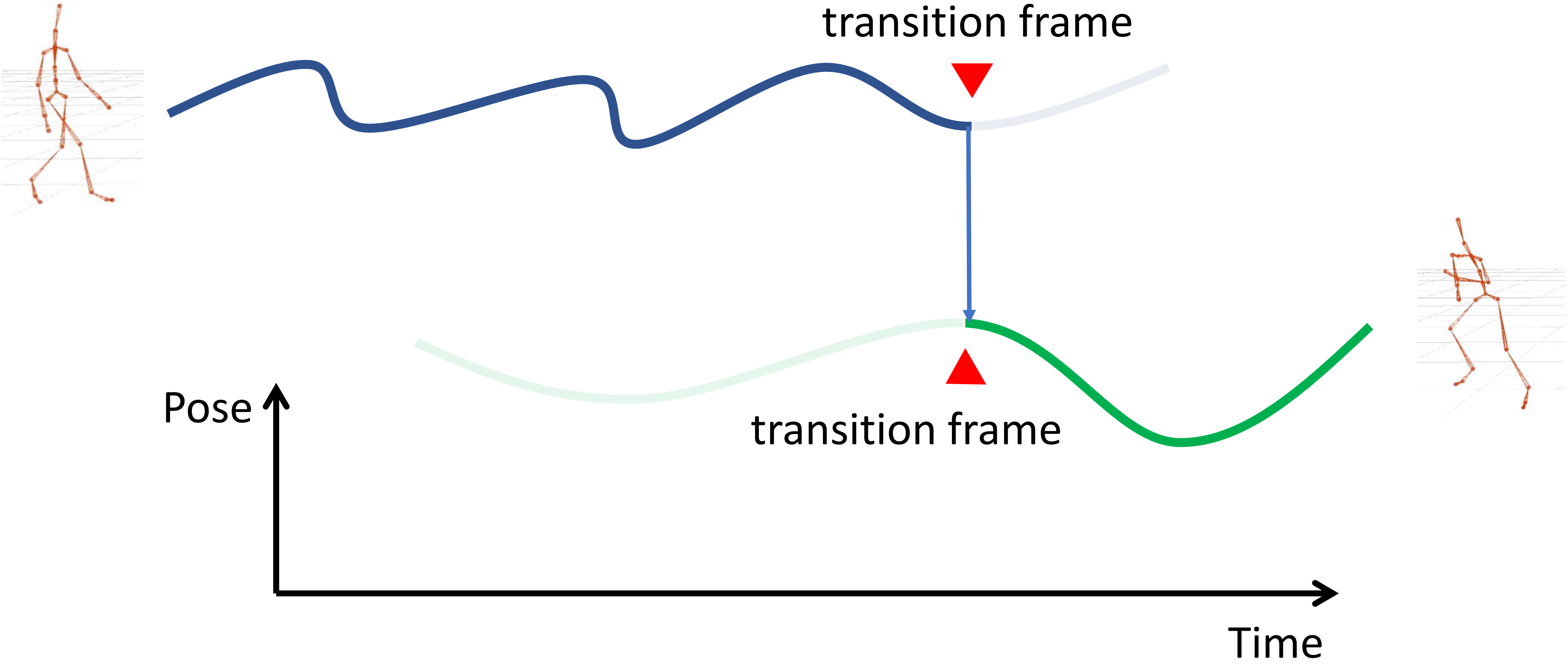
# Motion Transition



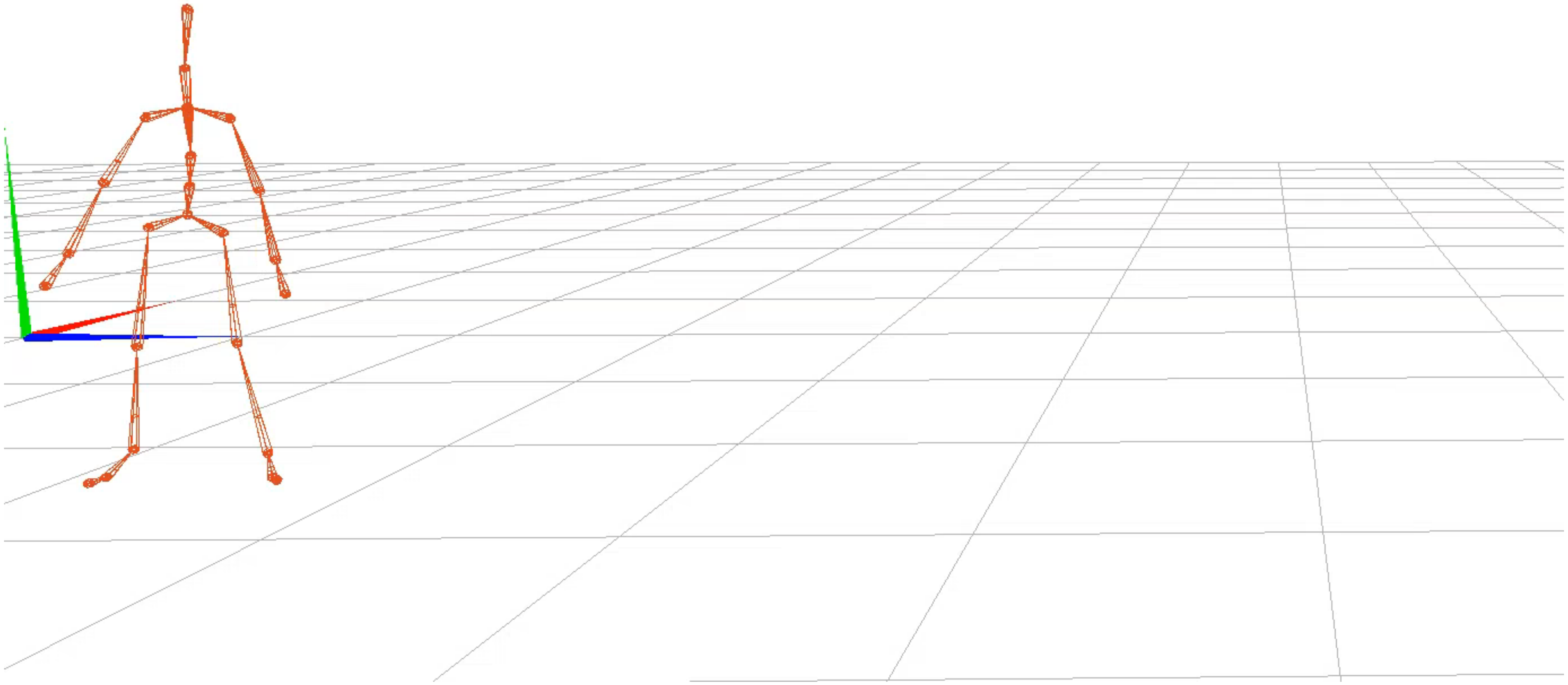
# Motion Transition



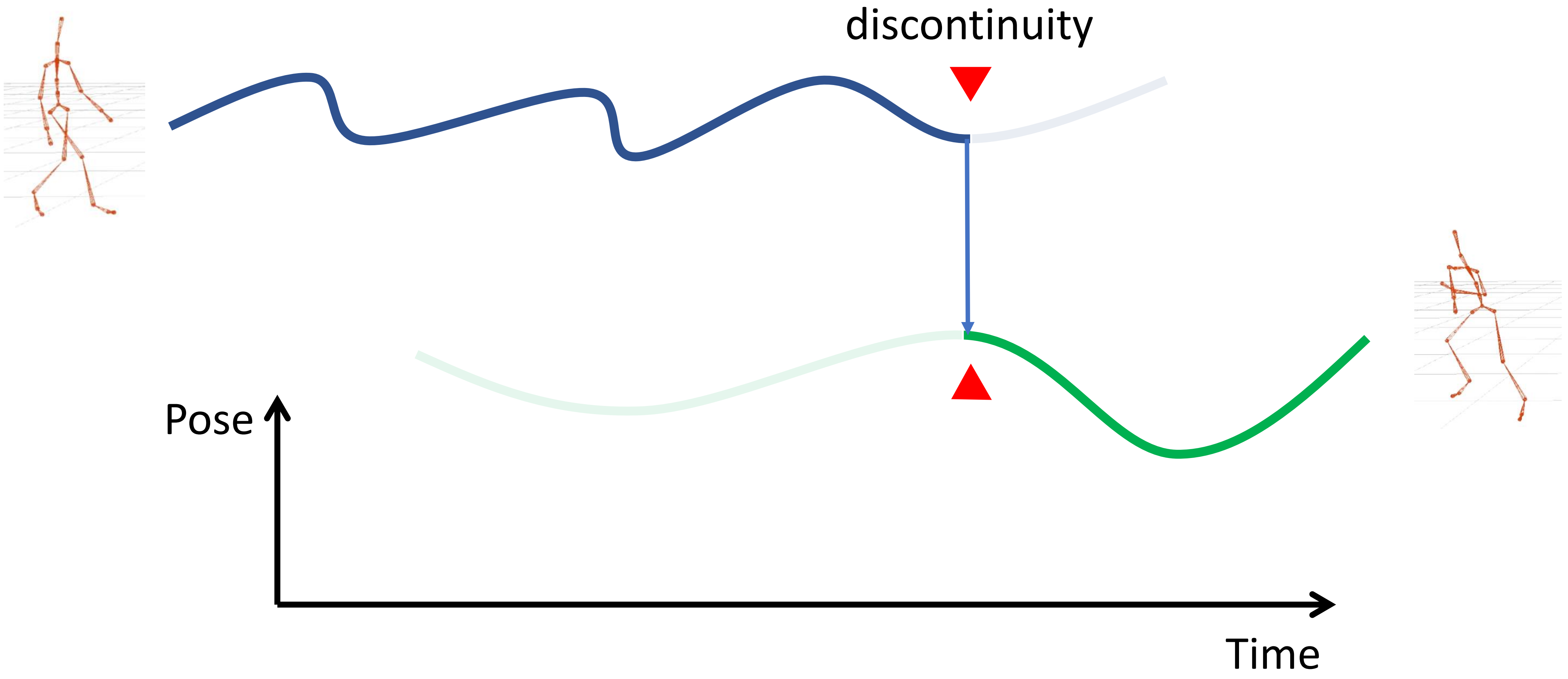
# Motion Transition



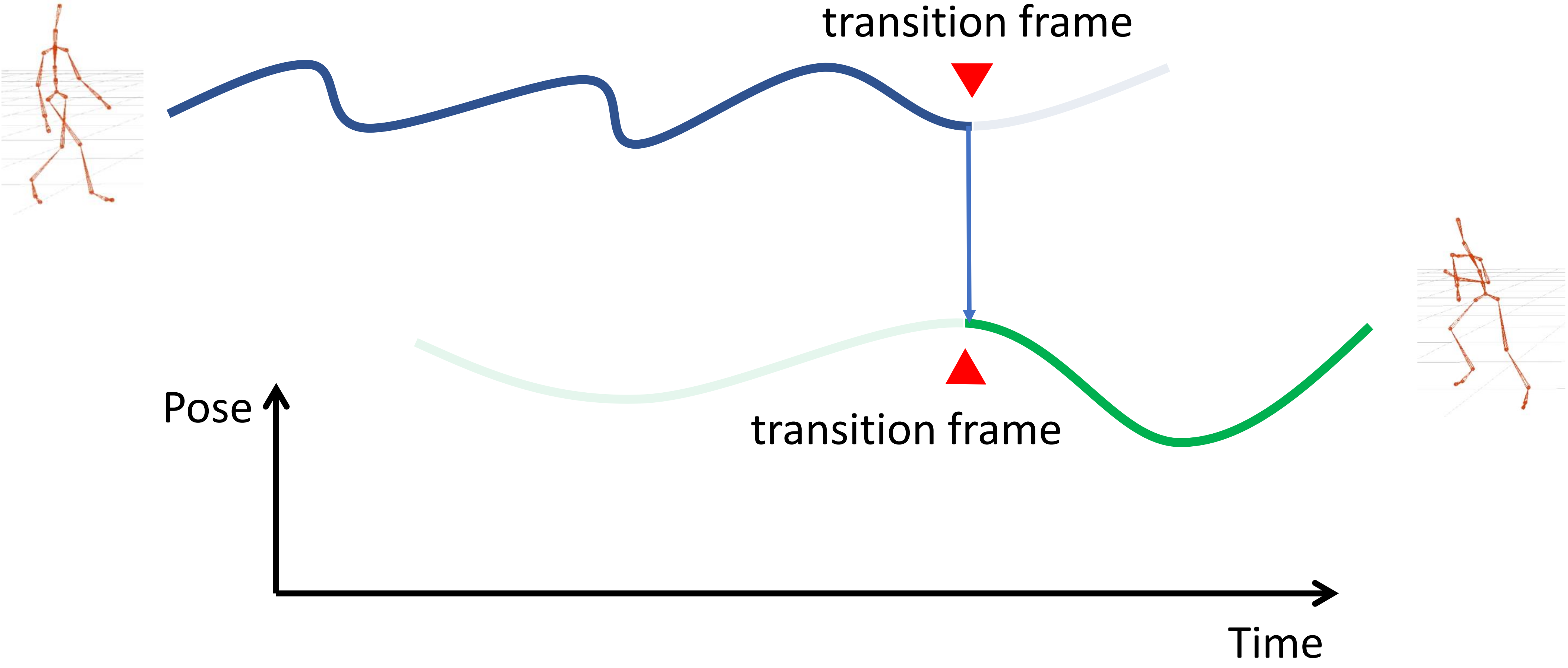
# Motion Transition



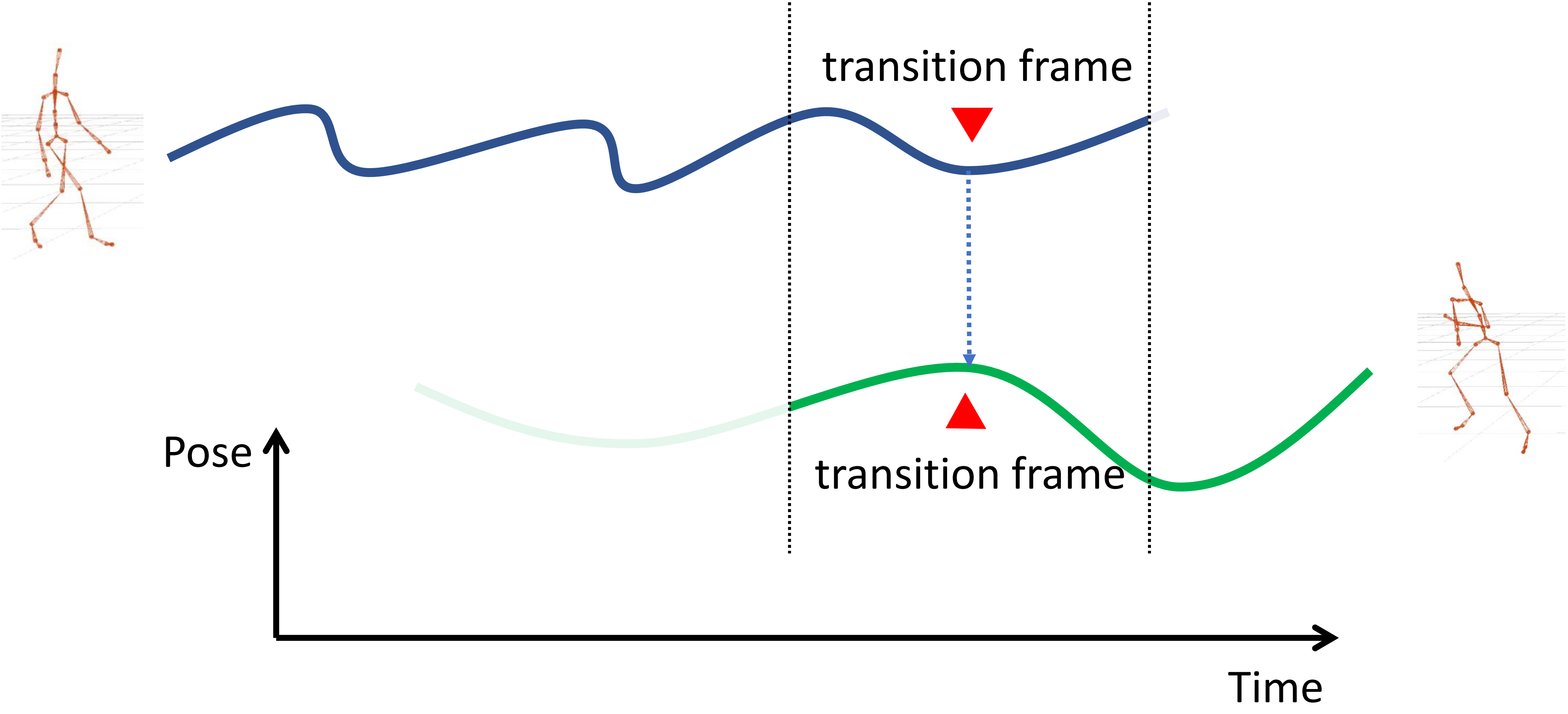
# Motion Transition



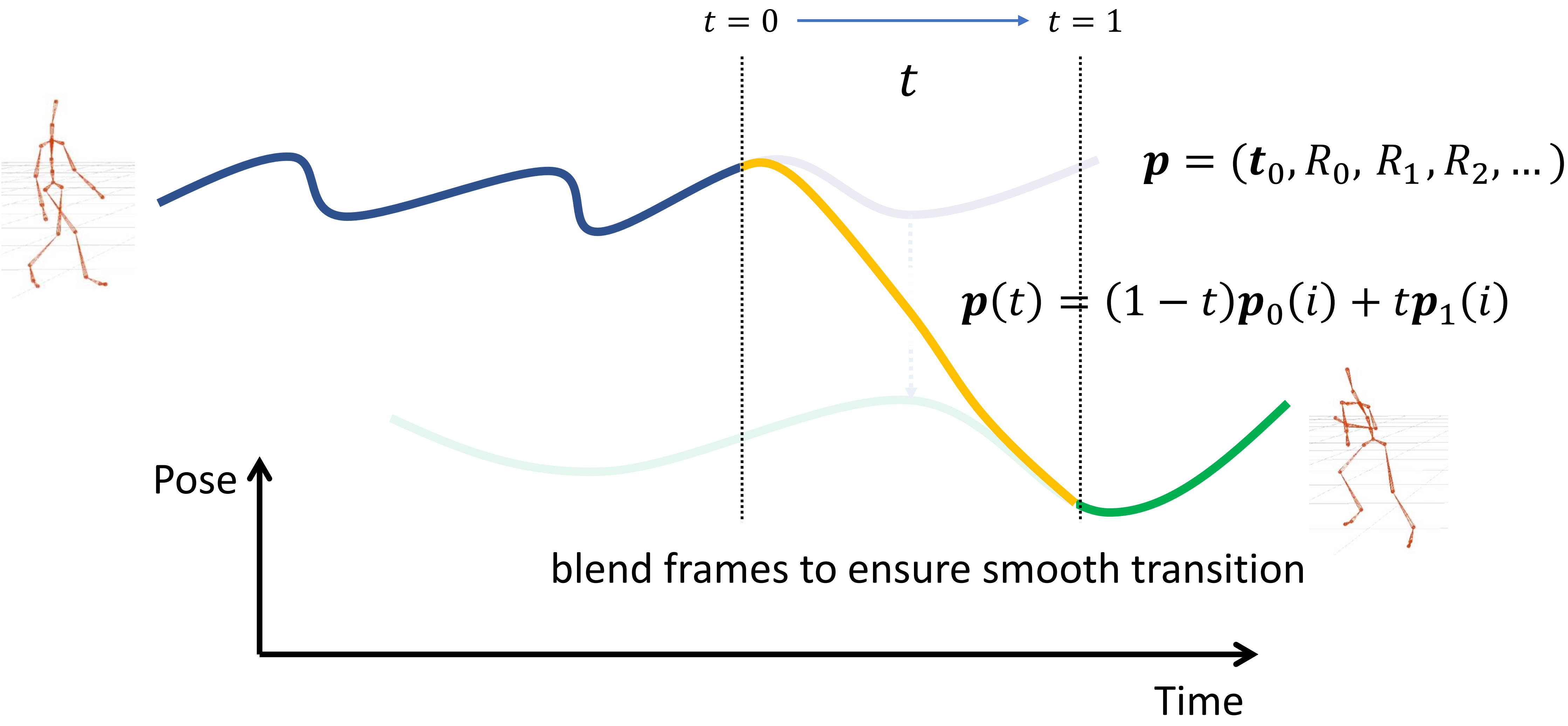
# Motion Transition



# Motion Transition



# Motion Transition

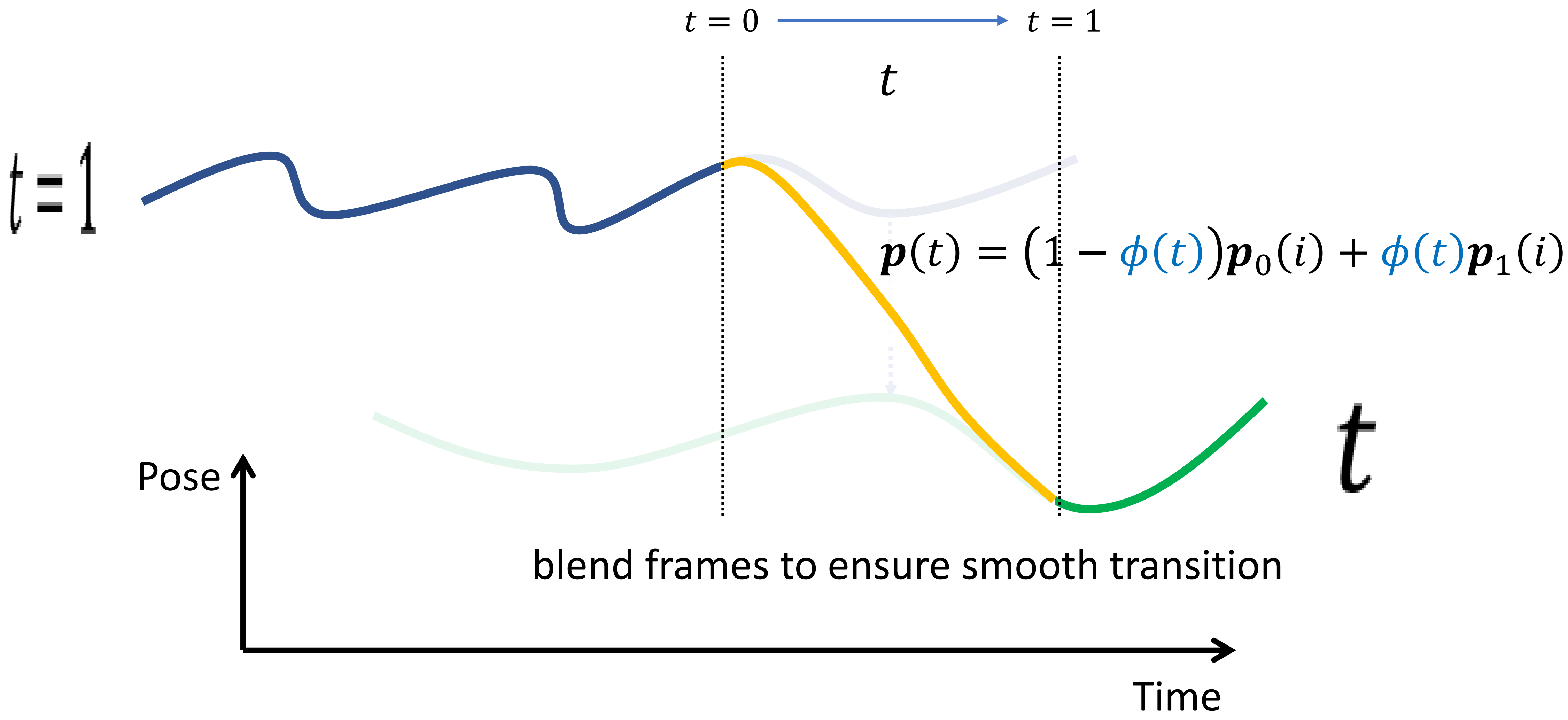


$$\boldsymbol{p} = (\boldsymbol{t}_0, R_0, R_1, R_2, \dots)$$

$$\boldsymbol{p}(t) = (1 - t)\boldsymbol{p}_0(i) + t\boldsymbol{p}_1(i)$$

blend frames to ensure smooth transition

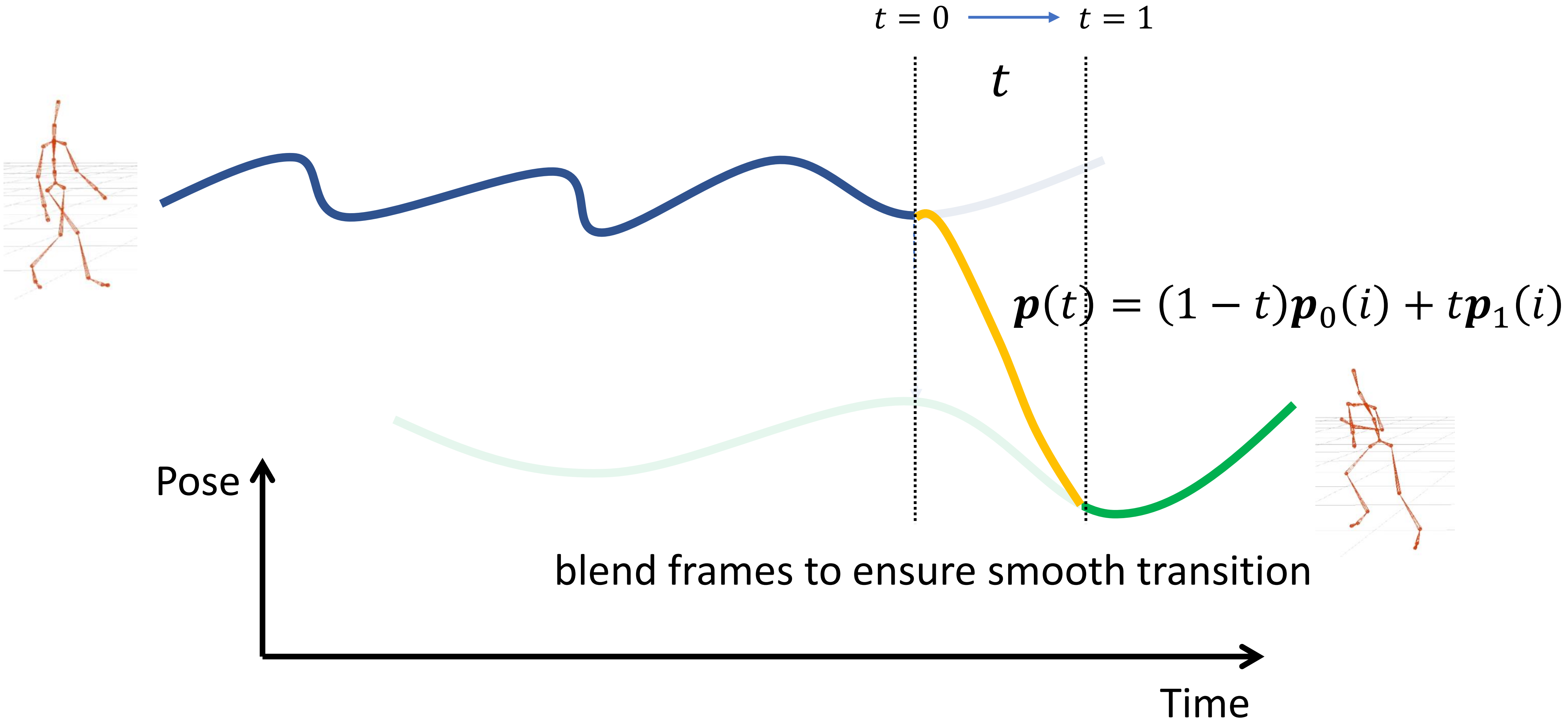
# Motion Transition



$$\boldsymbol{p}(t) = \big(1 - \phi(t)\big)\boldsymbol{p}_0(i) + \phi(t)\boldsymbol{p}_1(i)$$

blend frames to ensure smooth transition

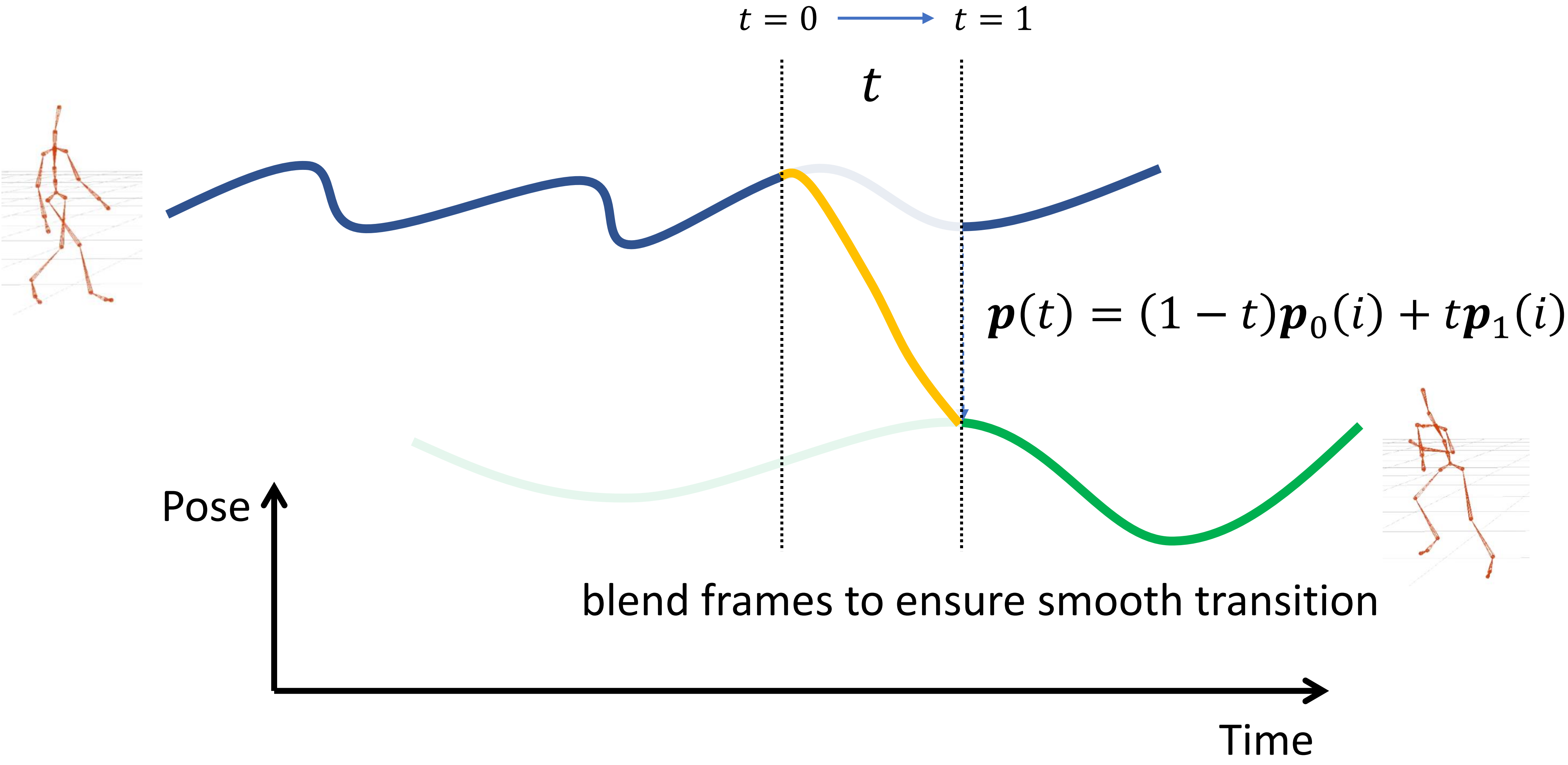
# Motion Transition



$$\boldsymbol{p}(t) = (1 - t)\boldsymbol{p}_0(i) + t\boldsymbol{p}_1(i)$$

blend frames to ensure smooth transition

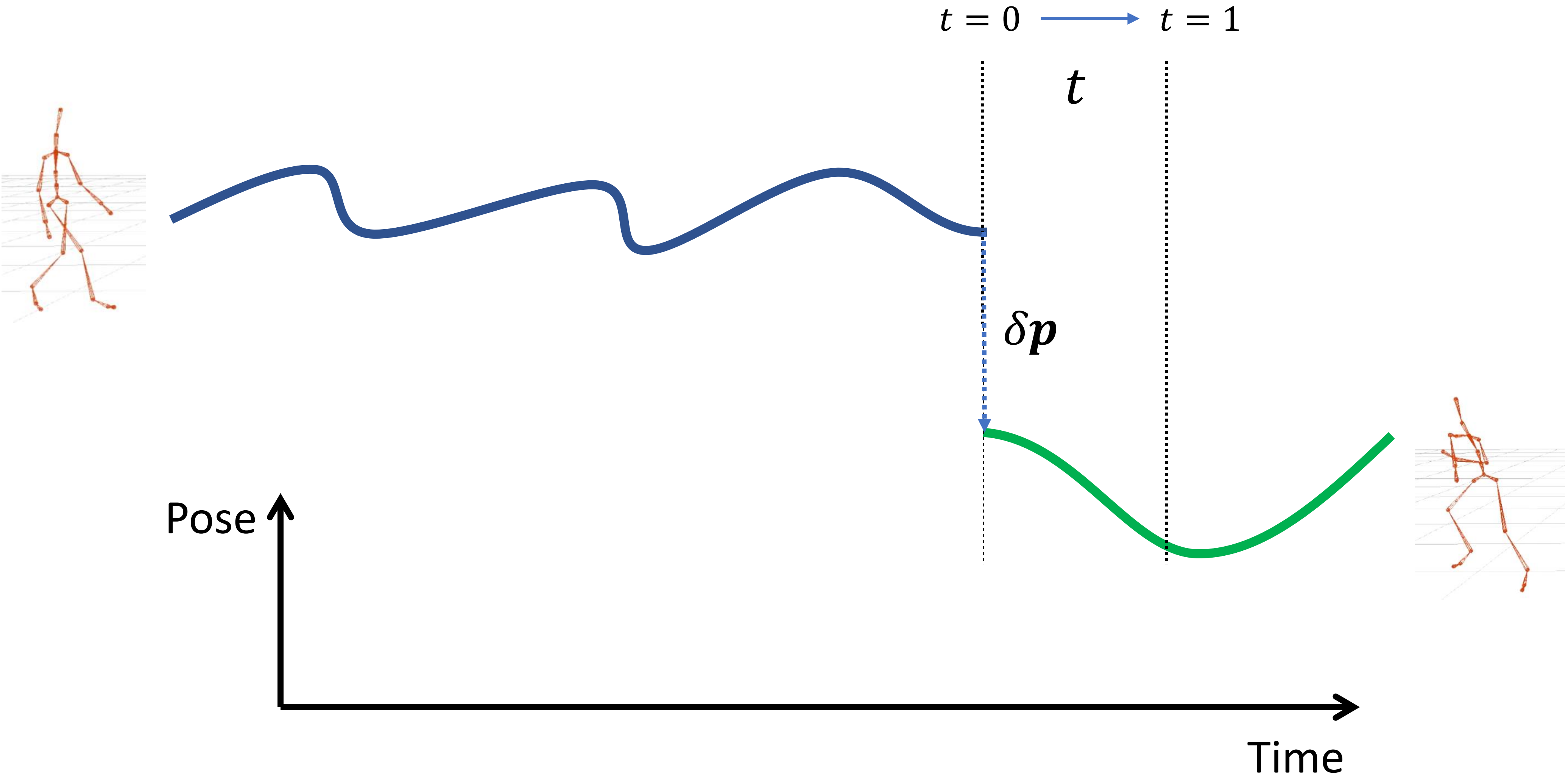
# Motion Transition



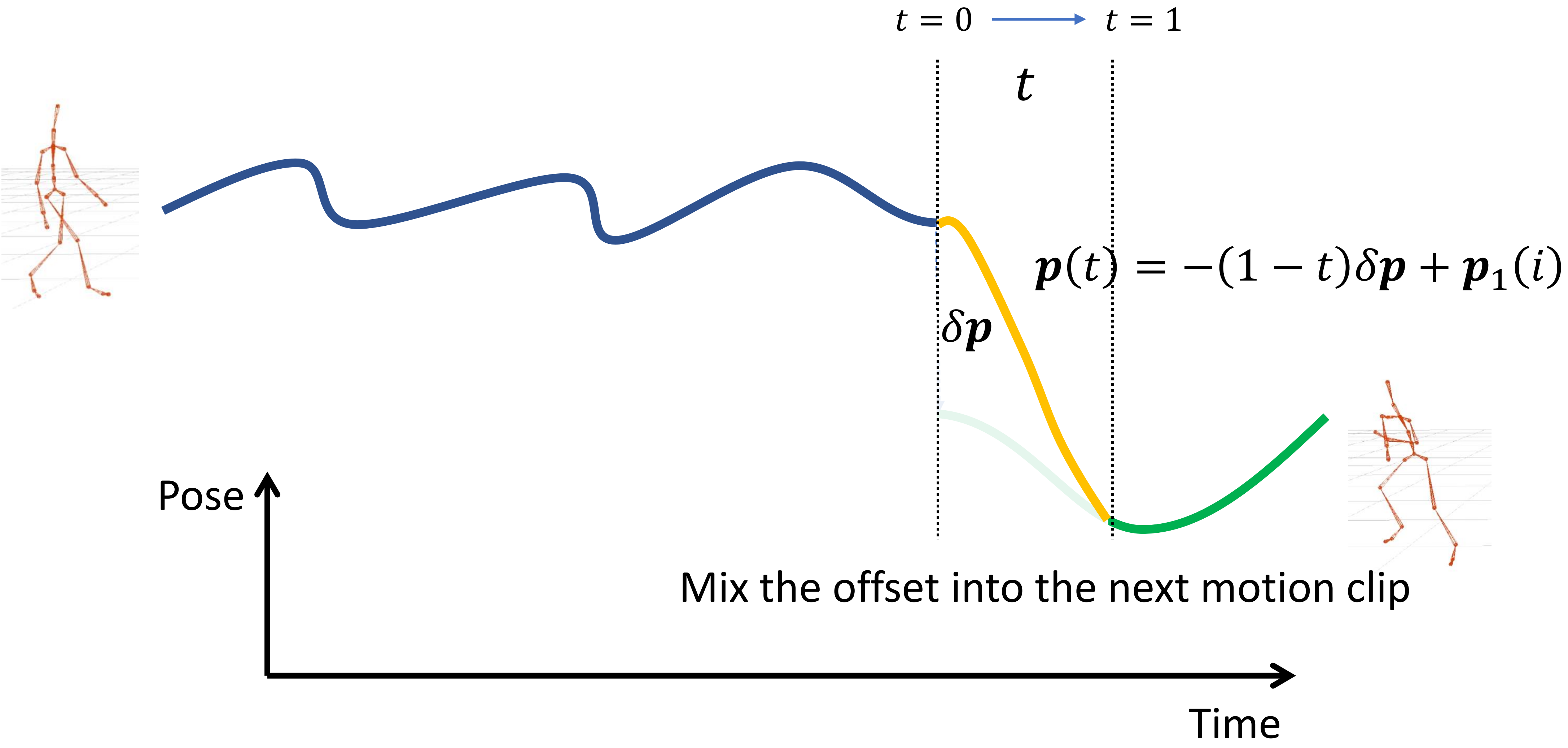
$$\boldsymbol{p}(t) = (1-t)\boldsymbol{p}_0(i) + t\boldsymbol{p}_1(i)$$

blend frames to ensure smooth transition

# Motion Transition



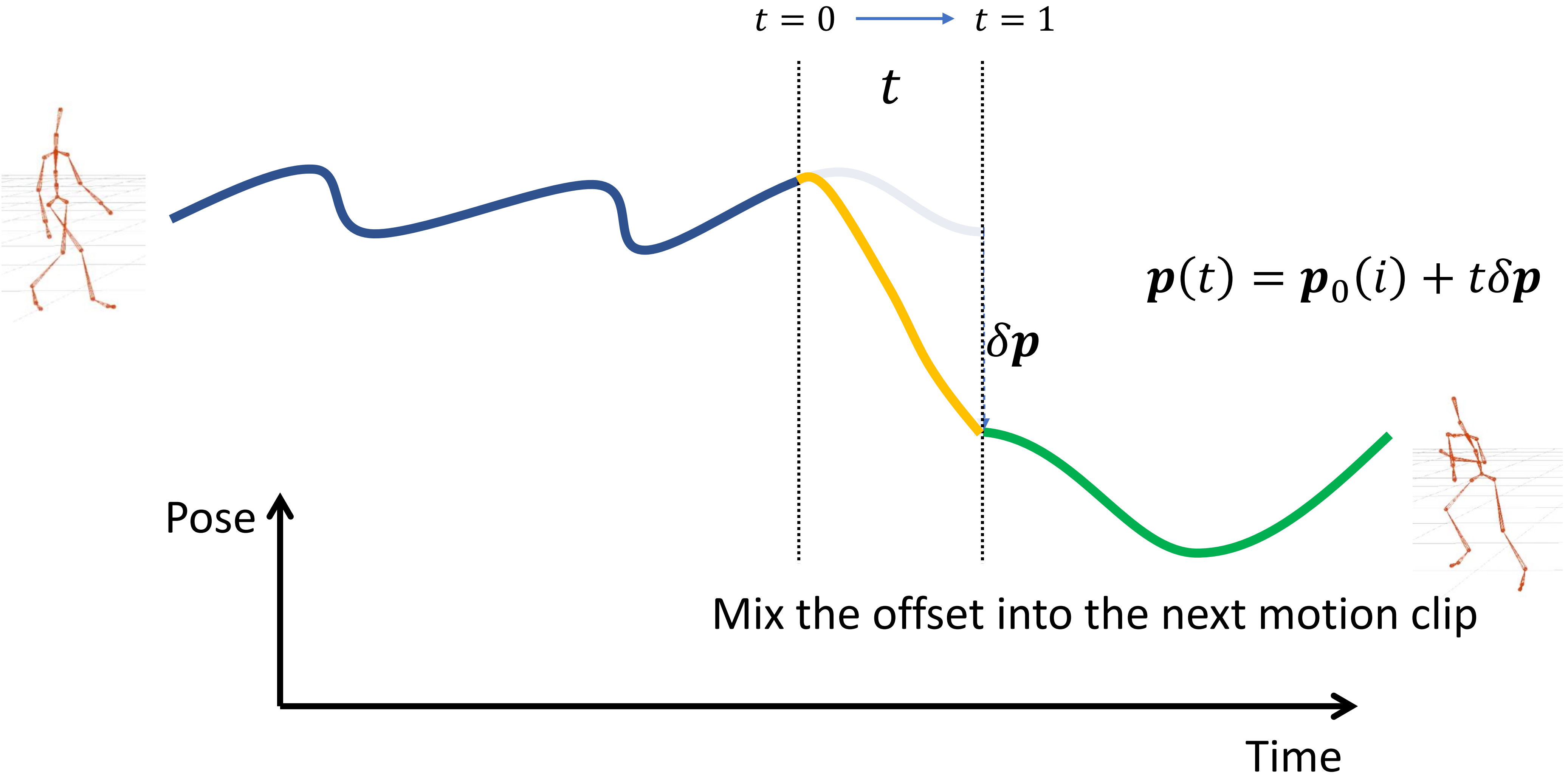
# Motion Transition



$$\boldsymbol{p}(t) = -(1-t)\delta\boldsymbol{p} + \boldsymbol{p}_1(i)$$

Mix the offset into the next motion clip

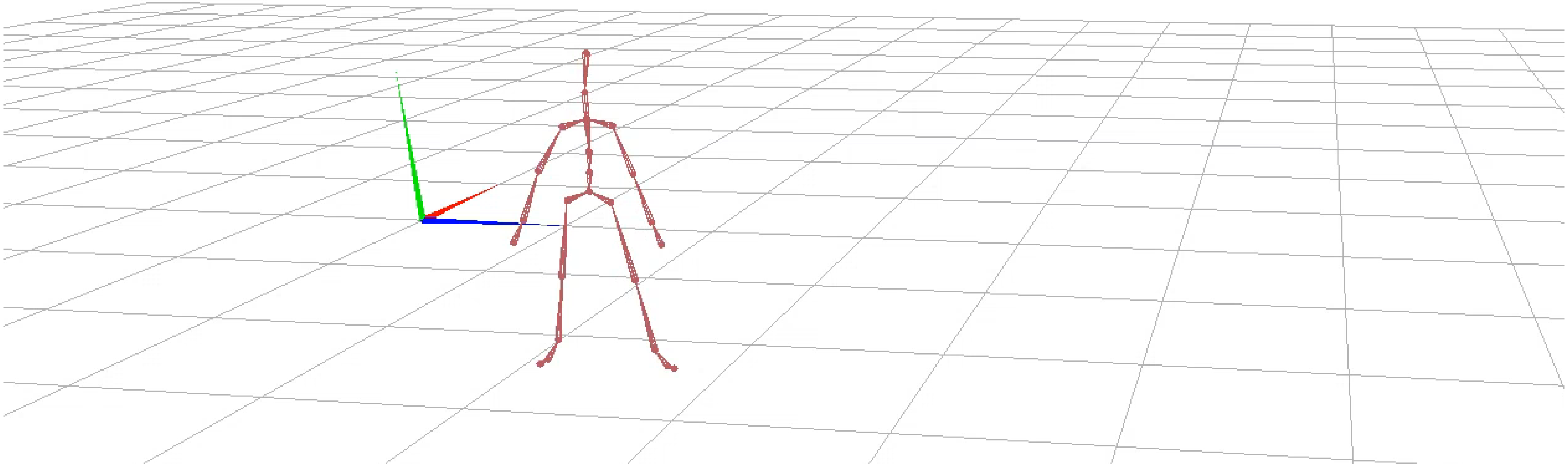
# Motion Transition



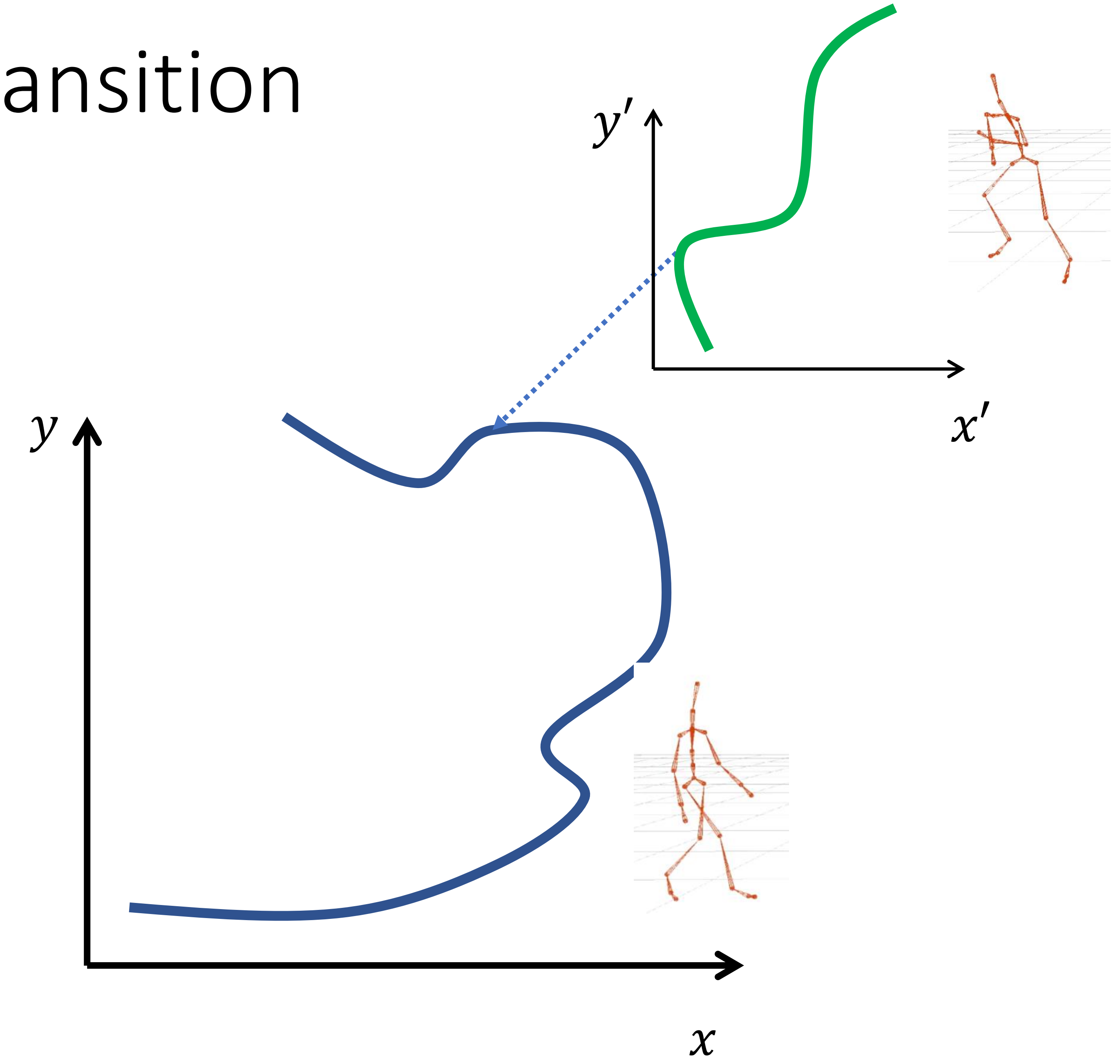
$$\boldsymbol{p}(t) = \boldsymbol{p}_0(i) + t\delta\boldsymbol{p}$$

Mix the offset into the next motion clip

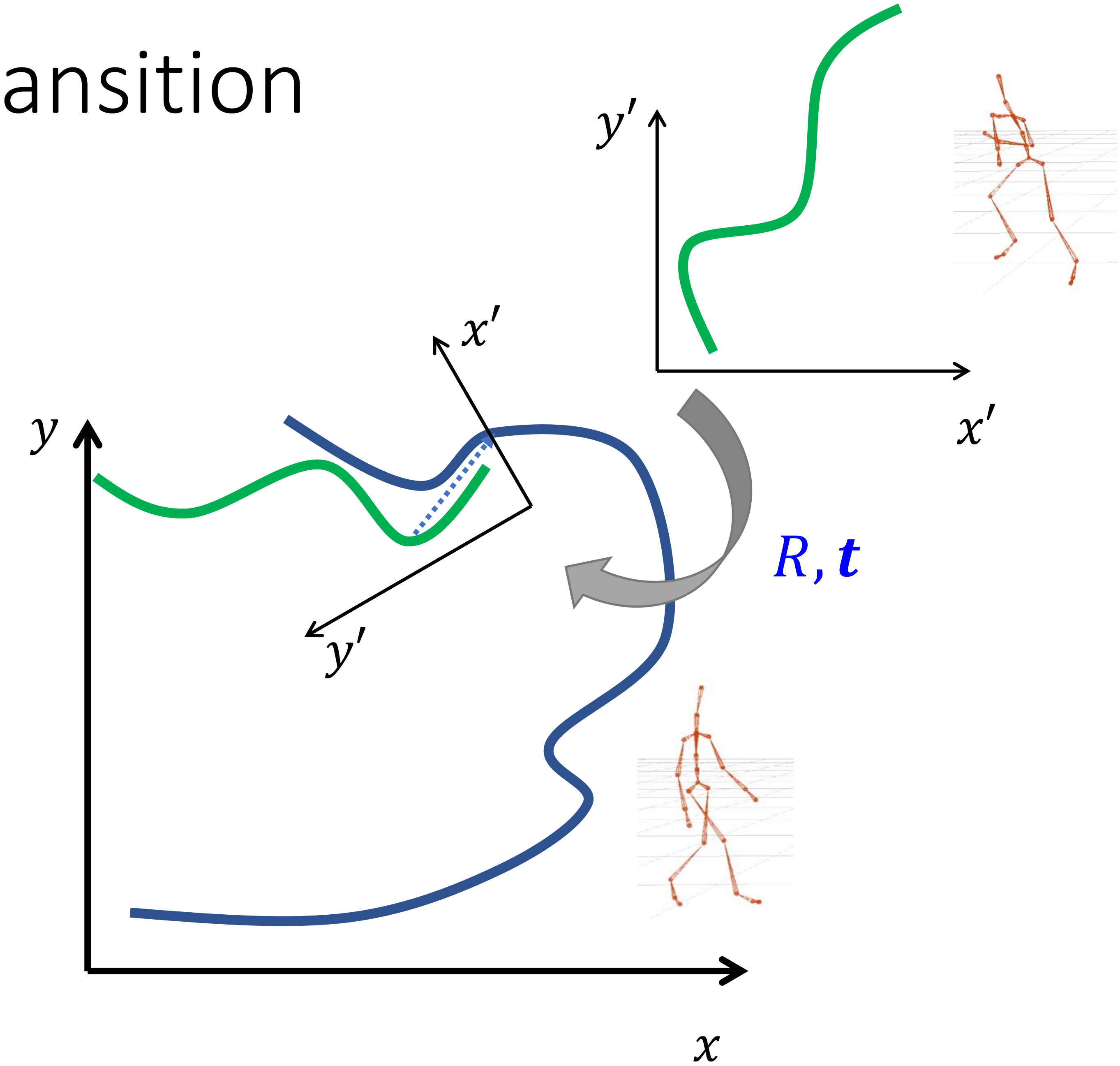
# Motion Transition



# Motion Transition

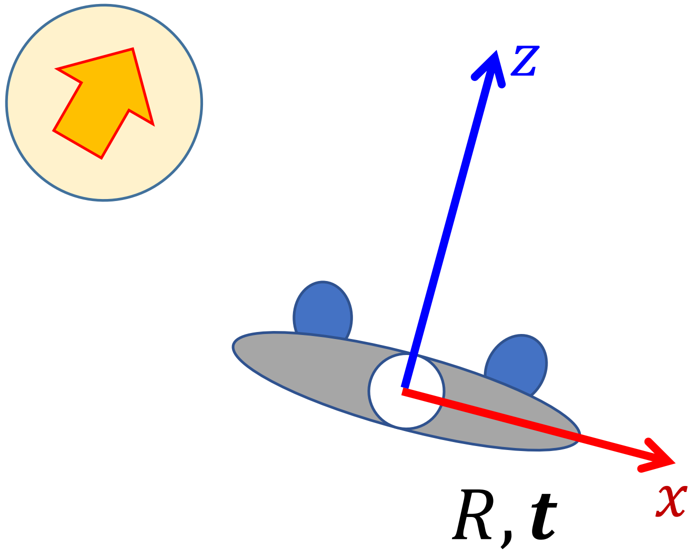


# Motion Transition



# “Facing Frame”

- A special coordinate system that moves horizontally with the character with one axis pointing to the “facing direction” of the character



$$R = \theta \boldsymbol{e}_y$$

$$\boldsymbol{t} = (t_x, 0, t_z)$$

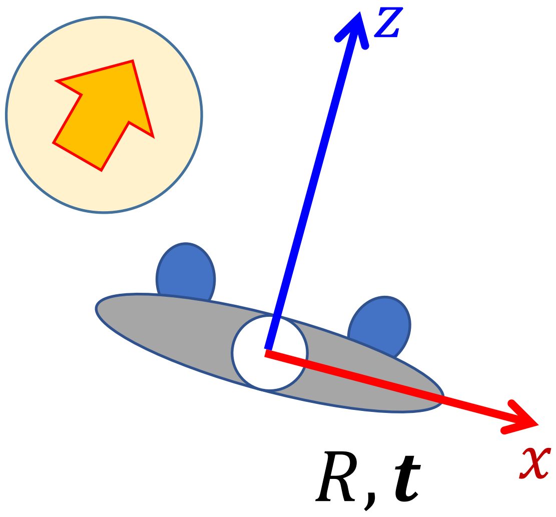
# "Facing Frame"

- A special coordinate system that moves horizontally with the character with one axis pointing to the "facing direction" of the character

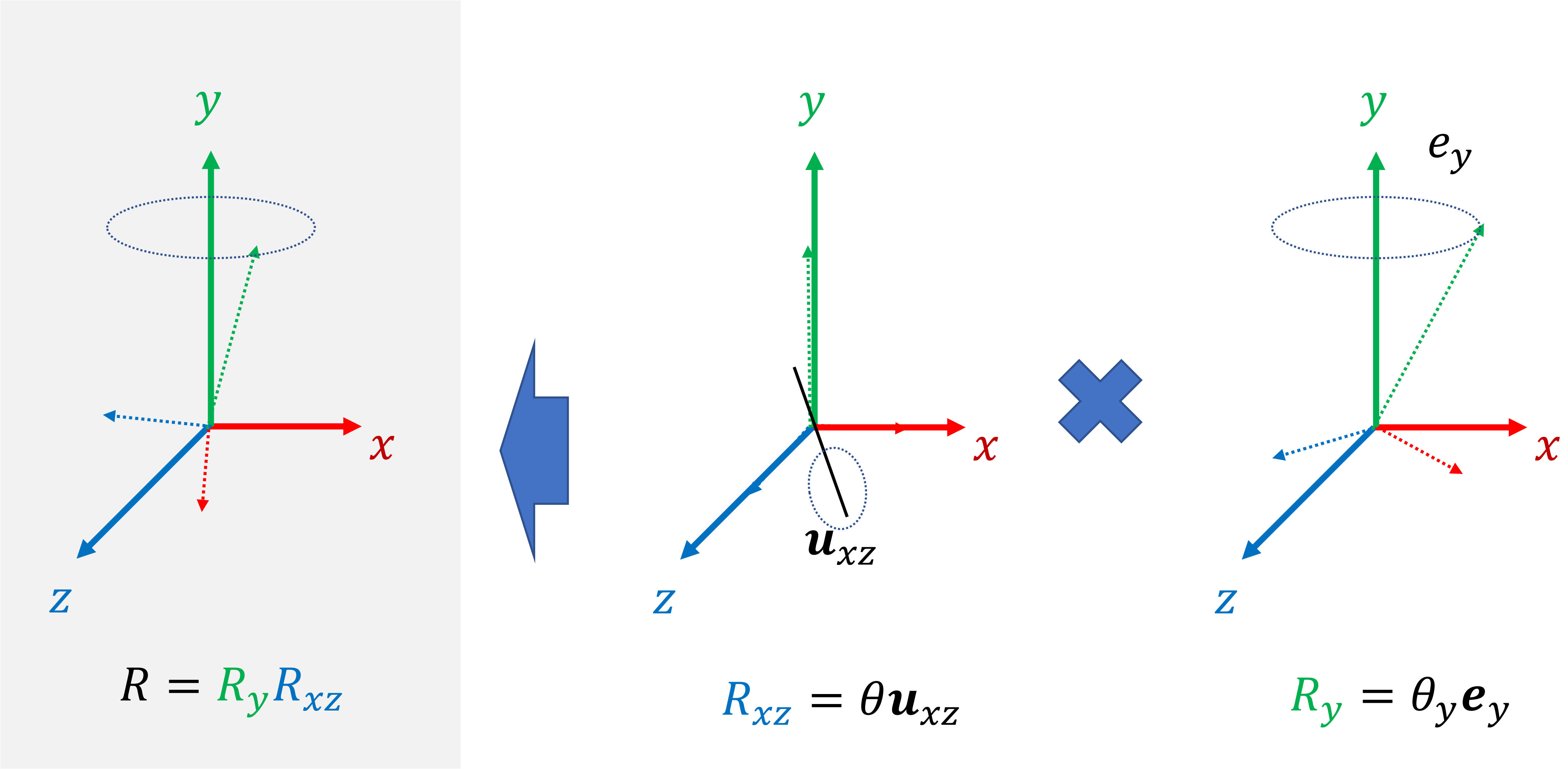
- Possible definitions of $R$
  - $R$ is the y-rotation that aligns the z-axis of the global frame to the heading direction
  - $R$ is the y-rotation that aligns x-axis of the global frame to the average direction of the vectors between shoulders and hips
  - Decomposition root rotation as $R_0 = R_y R_{xz}$

$$R = \theta \boldsymbol{e}_y$$

$$\boldsymbol{t} = (t_x, 0, t_z)$$



# Rotation Decomposition



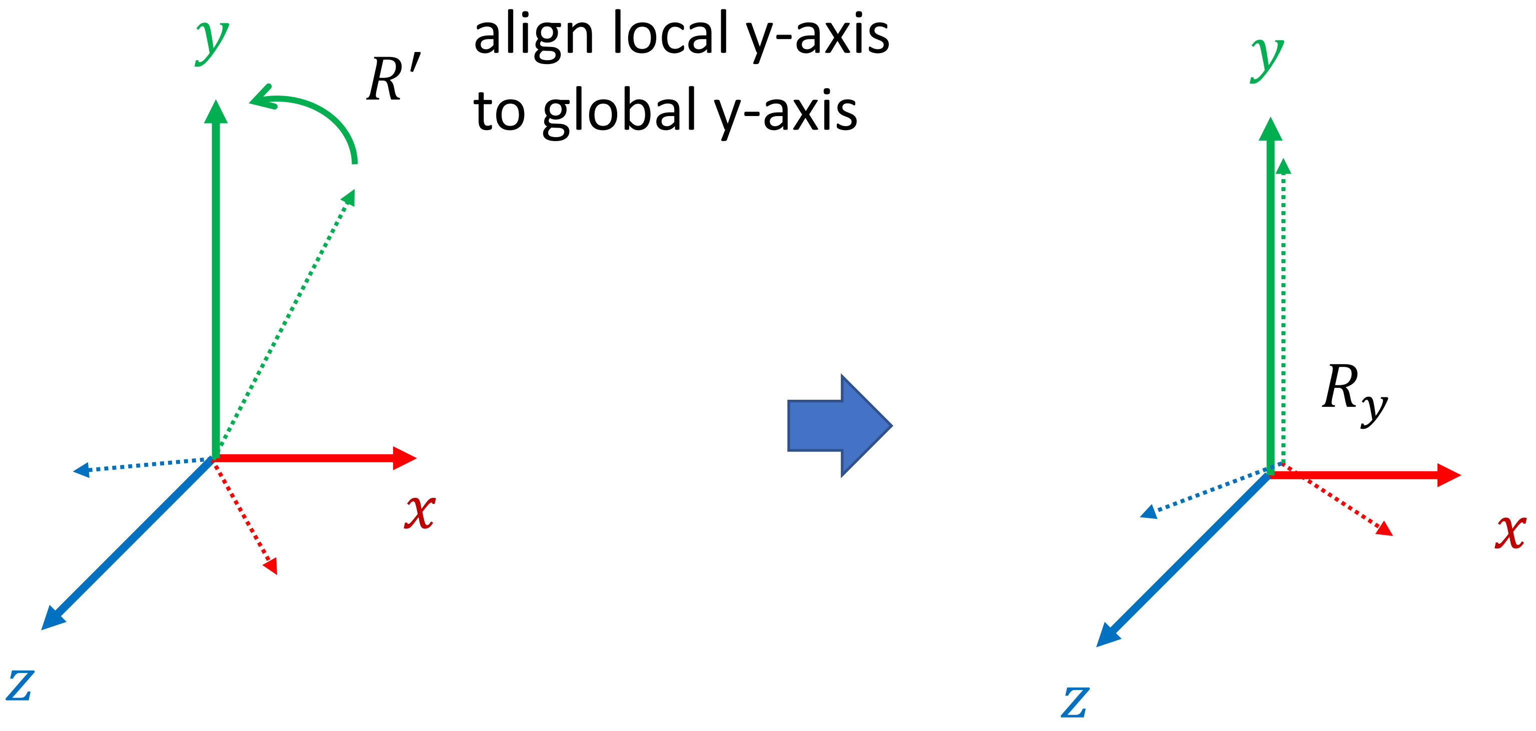
$$R = R_y R_{xz}$$

$$R_{xz} = \theta \boldsymbol{u}_{xz}$$

$$\boldsymbol{u}_{xz} = (u_x, 0, u_z)$$

$$R_y = \theta_y \boldsymbol{e}_y$$

# Rotation Decomposition

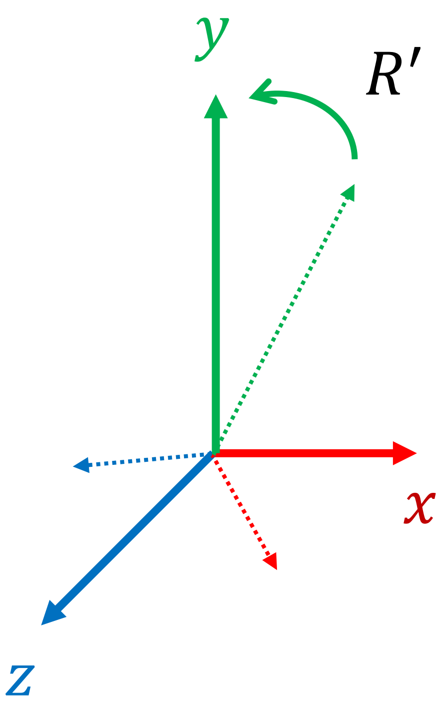


align local y-axis
to global y-axis

$$R = R_y R_{xz}$$

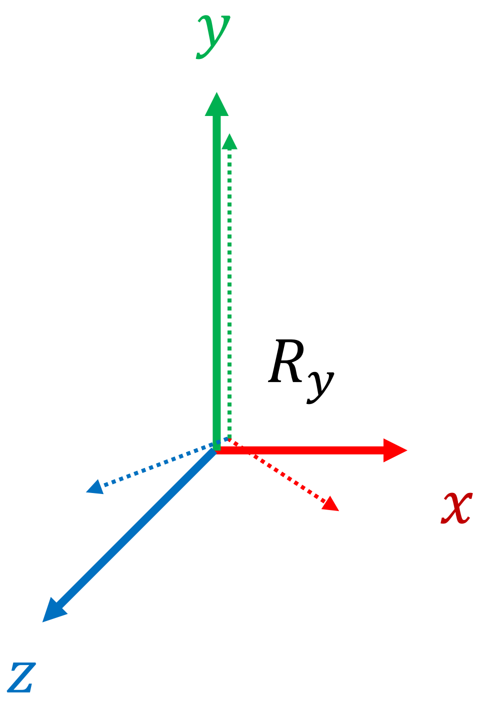
$$R_y = R' R$$

# Rotation Decomposition



align local y-axis
to global y-axis

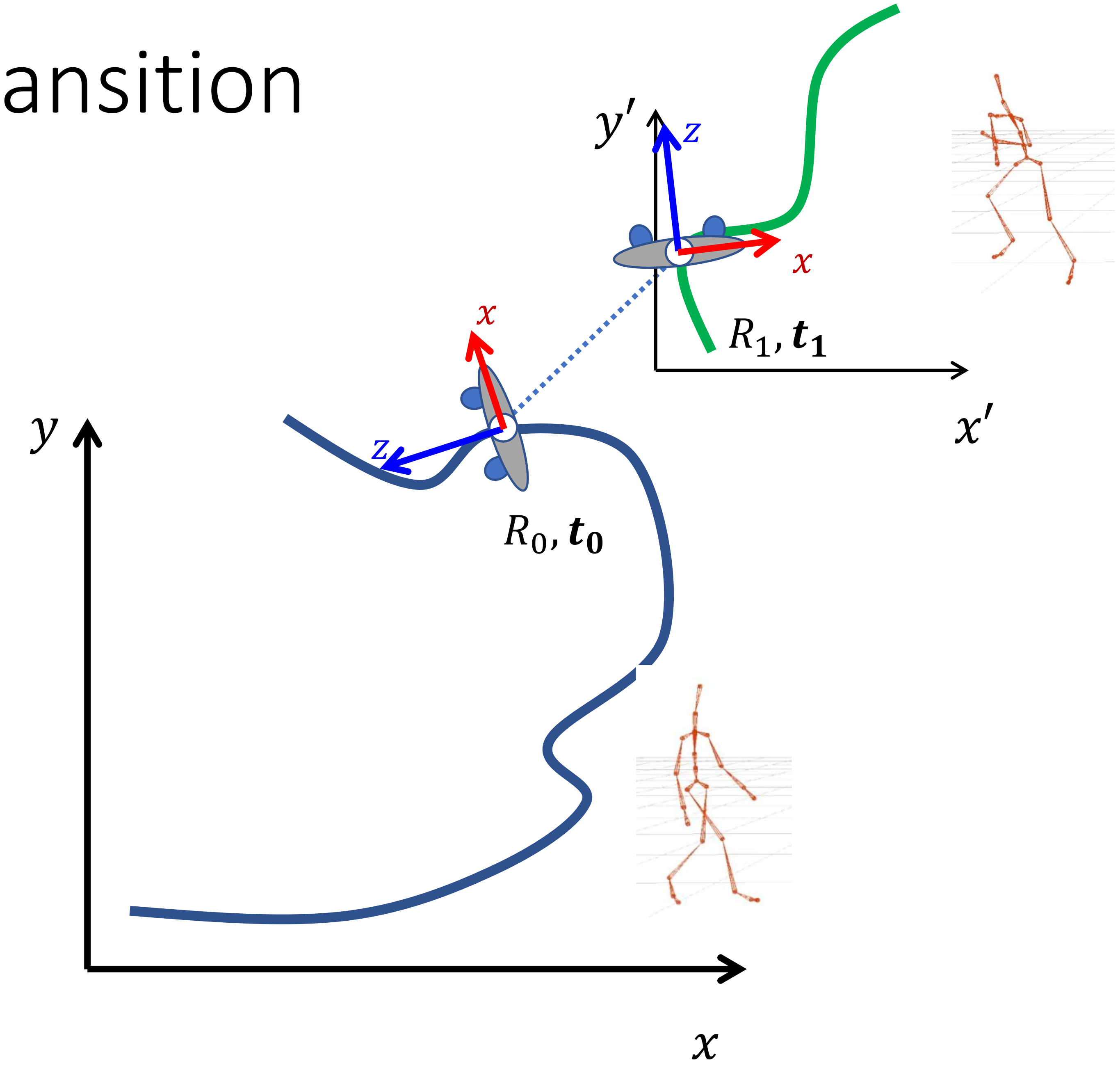
$$R = R_y R_{xz}$$



$$R_y = R' R$$

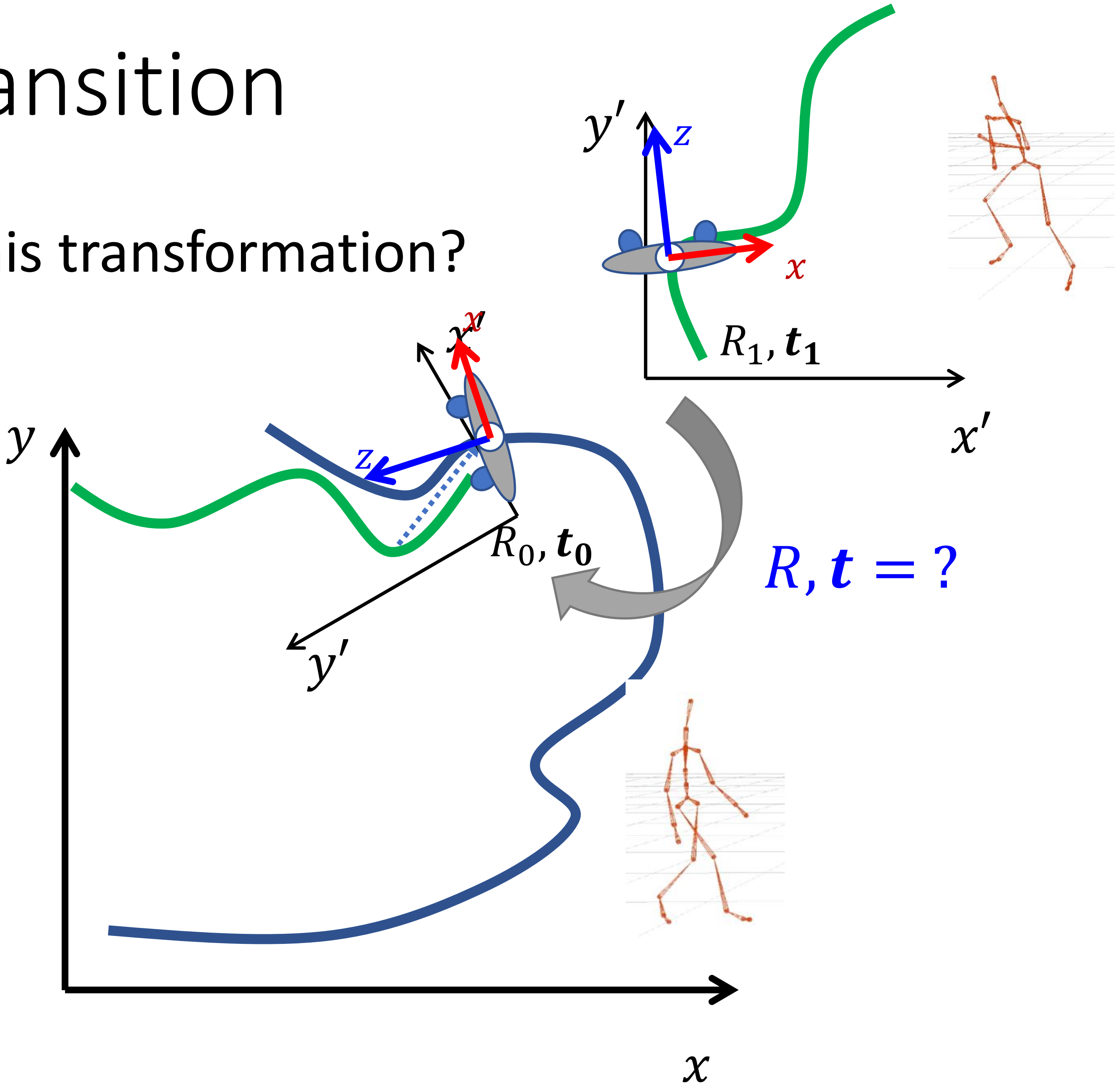
$$R_{xz} = R_y^T R$$

# Motion Transition



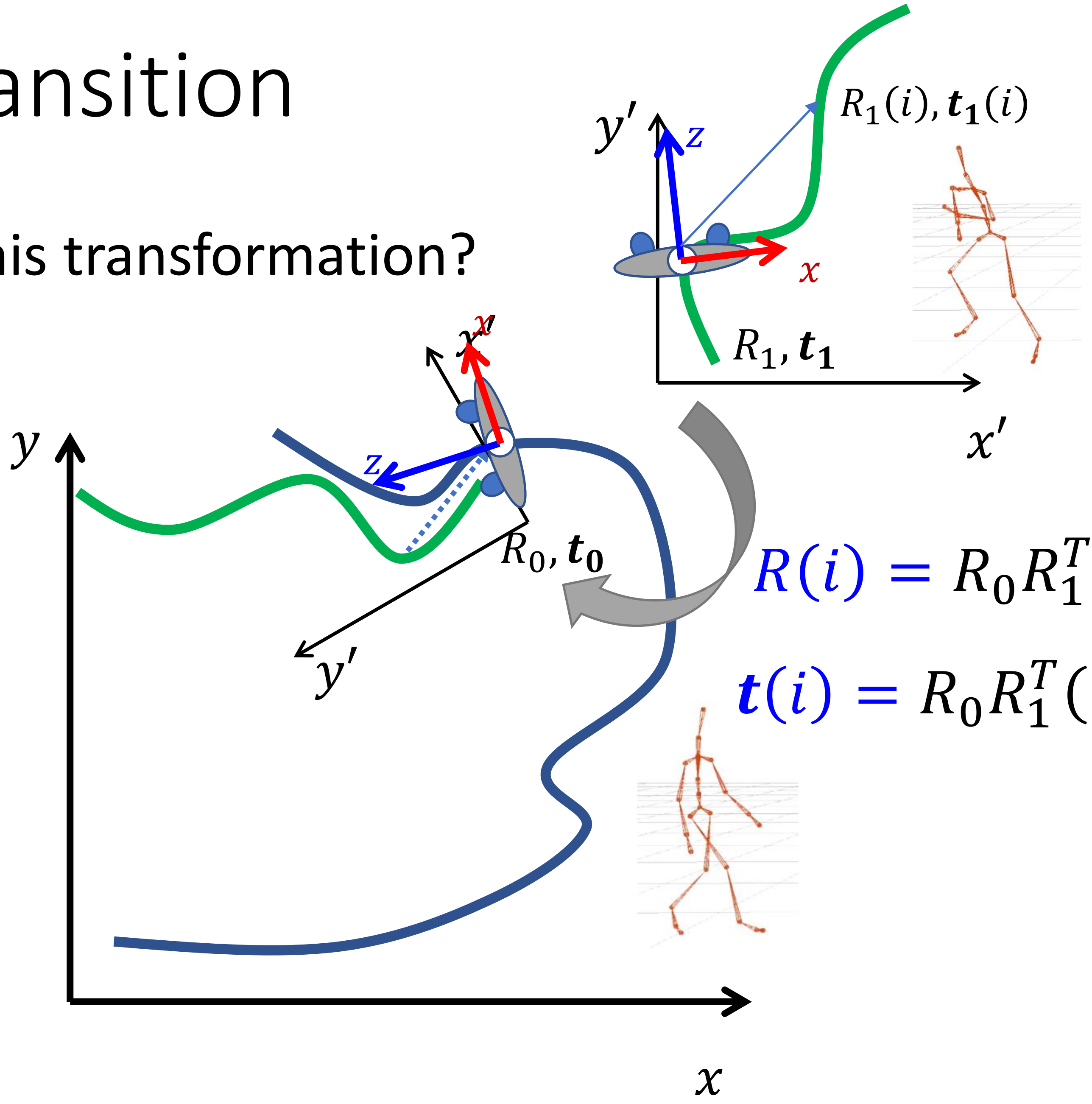
# Motion Transition

- How to compute this transformation?



# Motion Transition

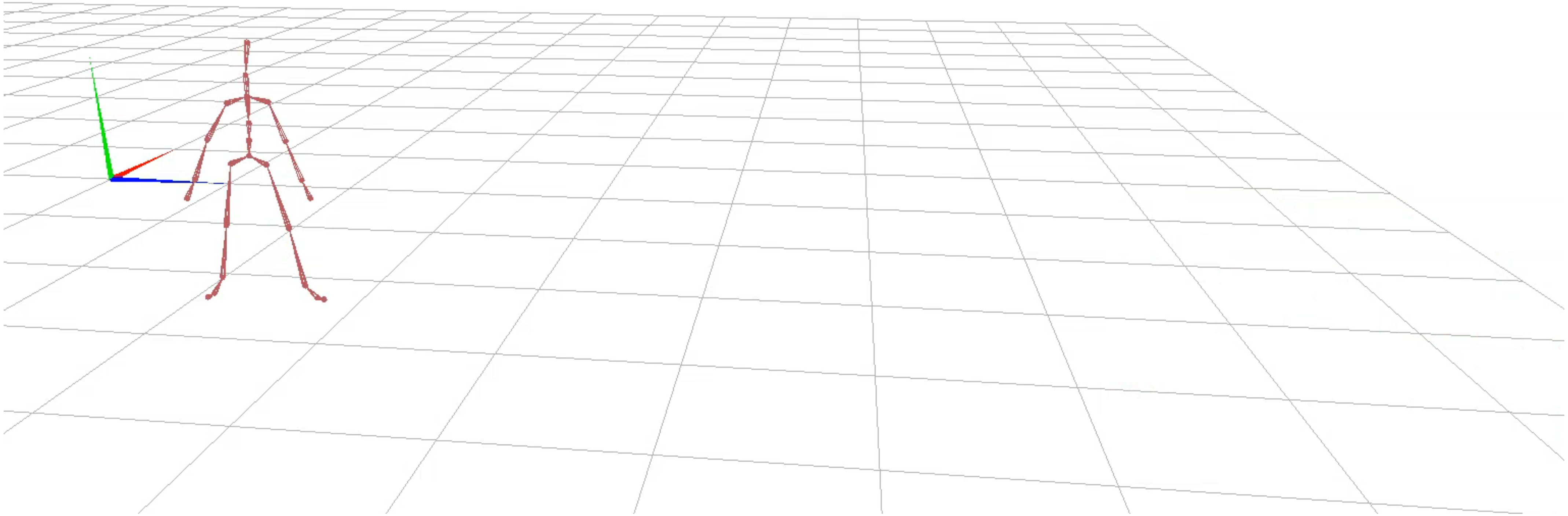
- How to compute this transformation?



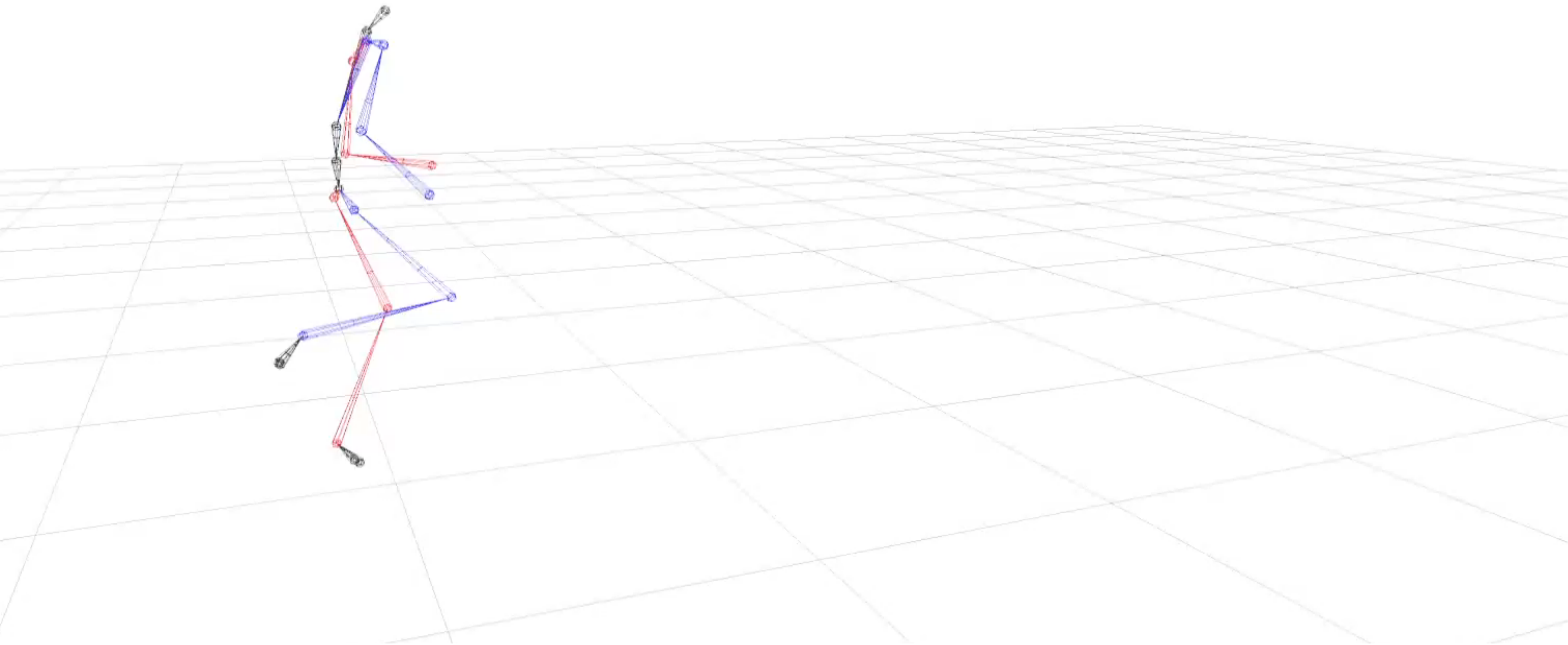
$$R(i) = R_0 R_1^T R_1(i)$$

$$\boldsymbol{t}(i) = R_0 R_1^T (\boldsymbol{t}_1(i) - \boldsymbol{t}_1) + \boldsymbol{t}_0$$

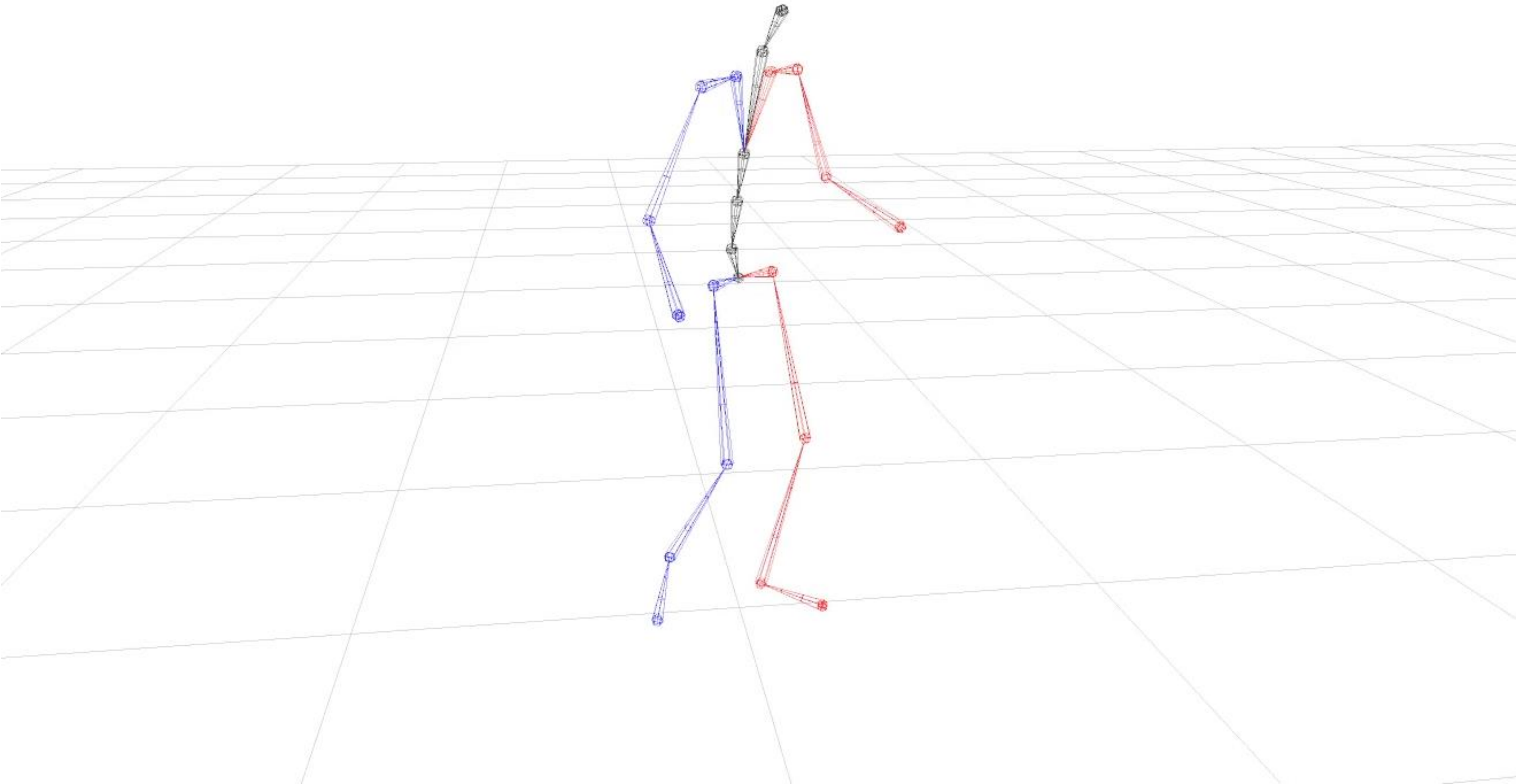
# Motion Transition



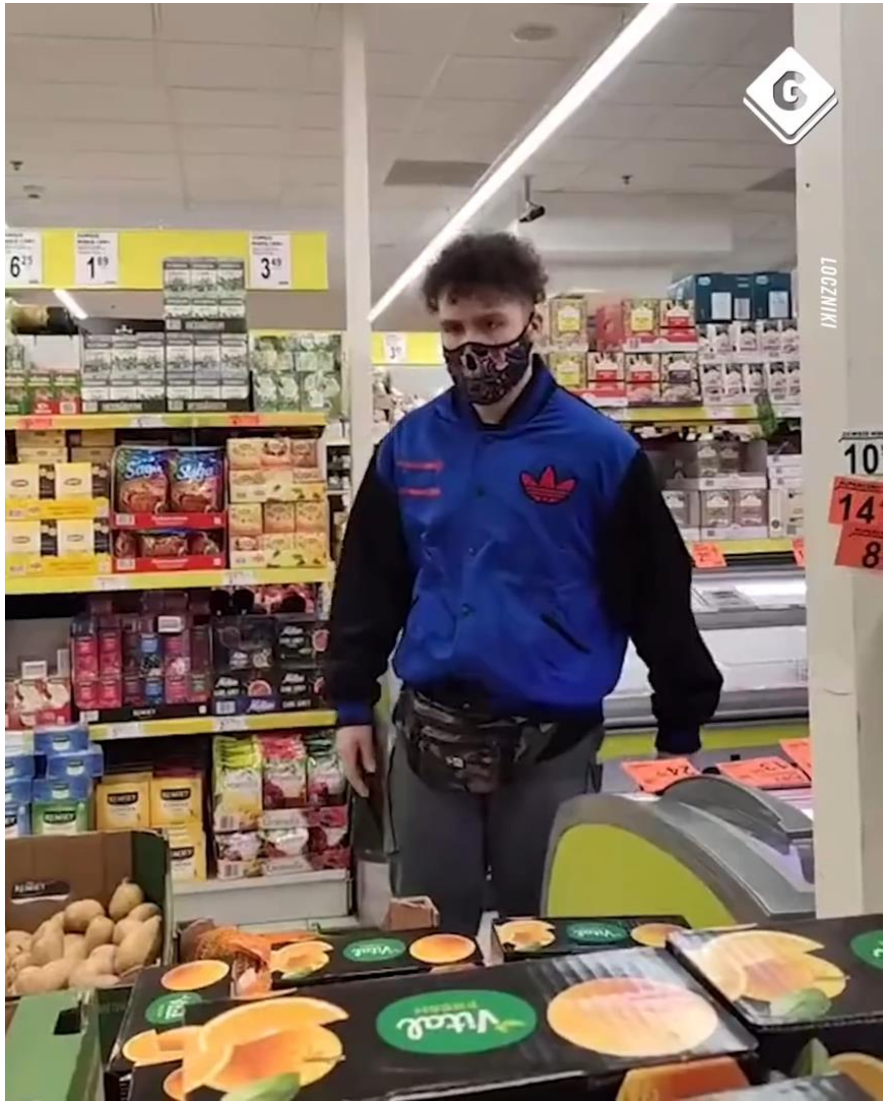
# Motion Transition



# Motion Data without Root Translation

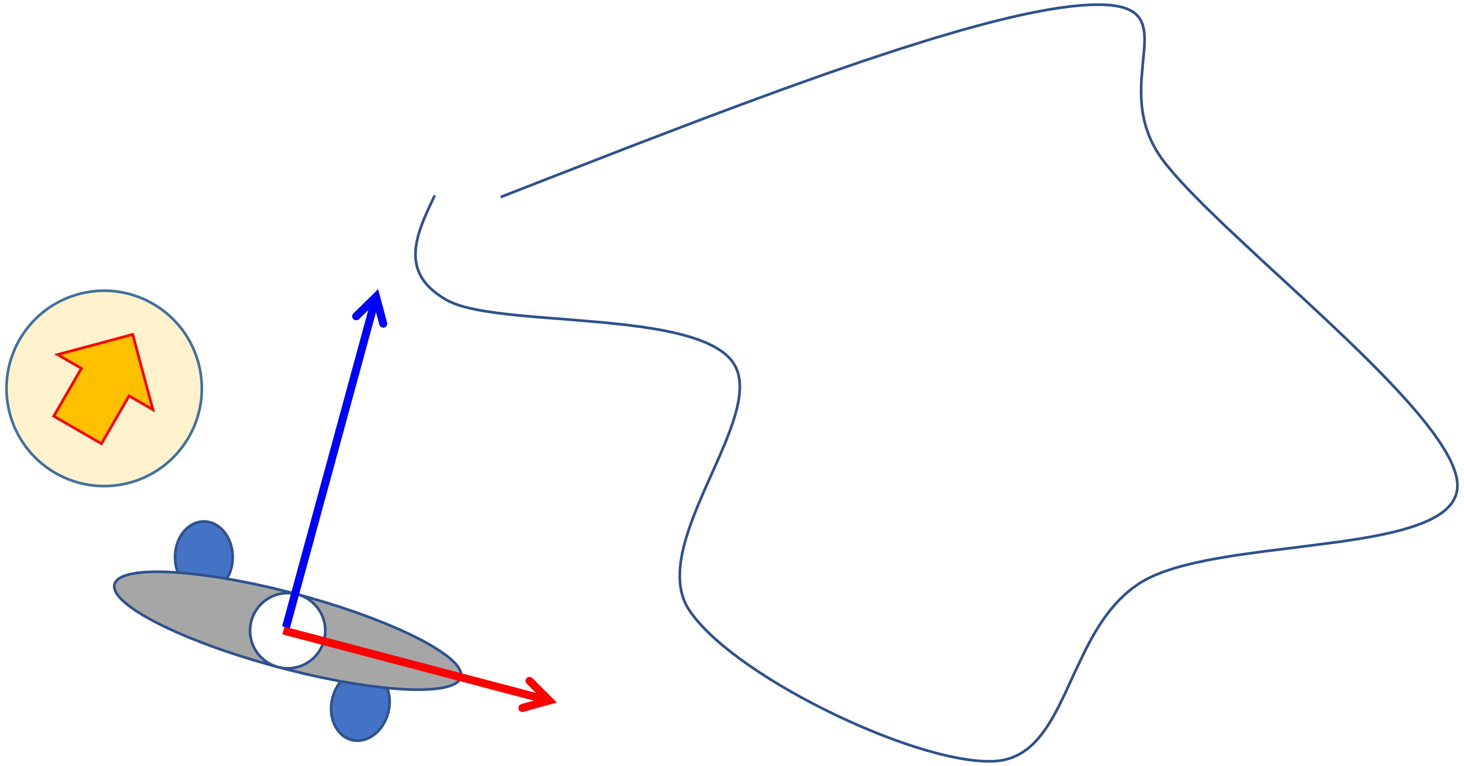


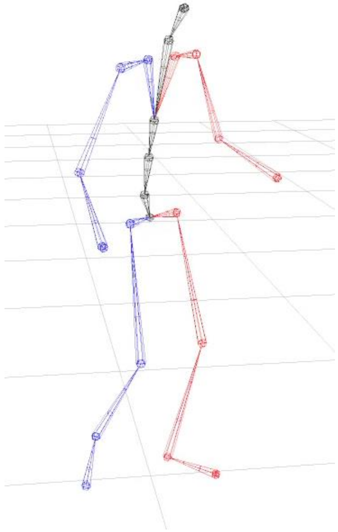
# Motion Data without Root Translation



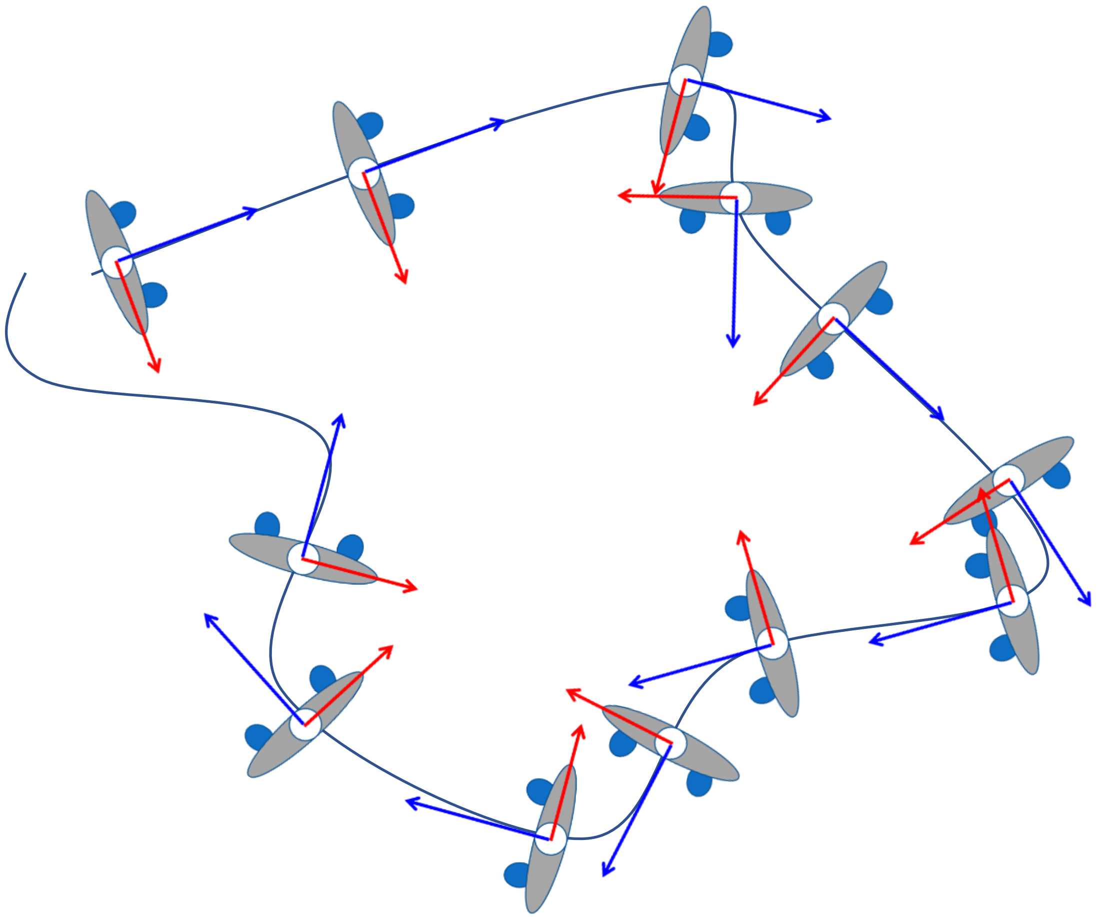
**Let's bring back gta characters to real life**
https://www.youtube.com/watch?v=DeutKhta1Uo

# Path Fitting



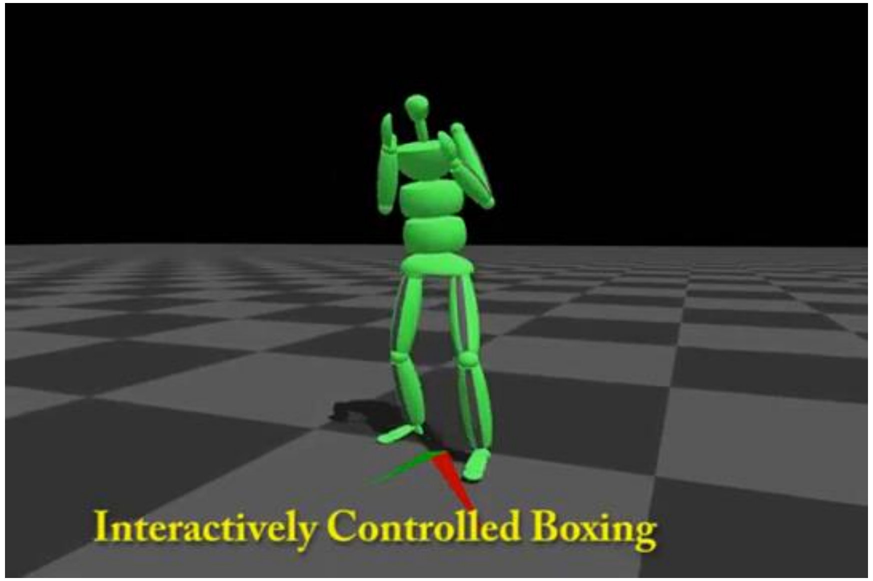


# Path Fitting

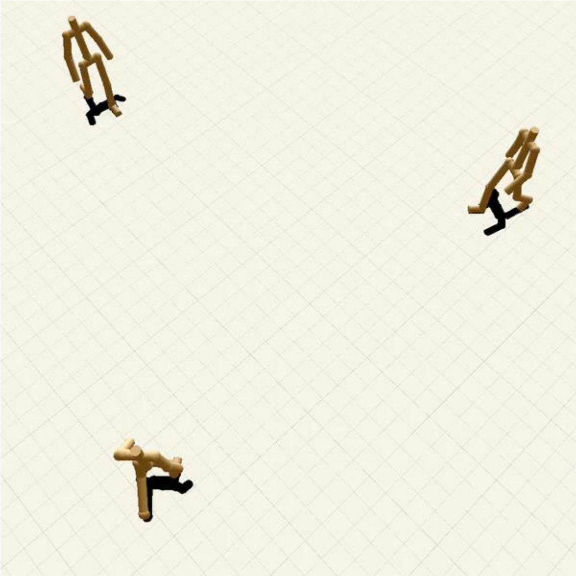


# Motion Composition

- Computationally generating motions according to
  - User control
  - Objects in the same environment
  - Movements of other characters
  - ……

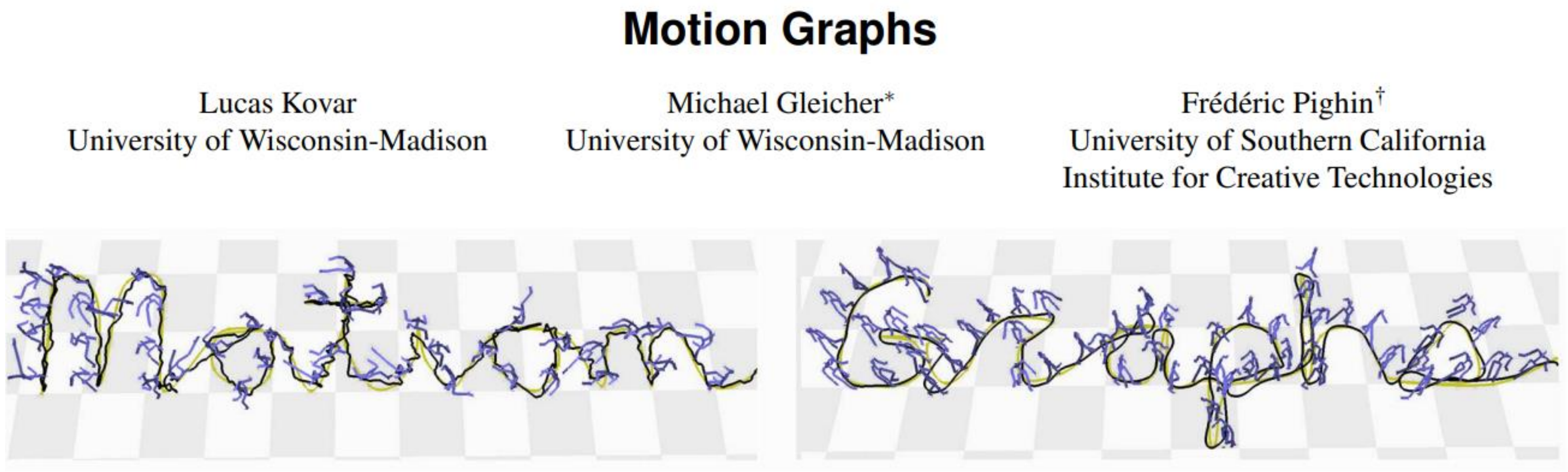


[Heck and Gleicher 2007, Parametric Motion Graphs]



[Treuille et al. 2007, Near-optimal character animation with continuous control]

# Motion Graphs

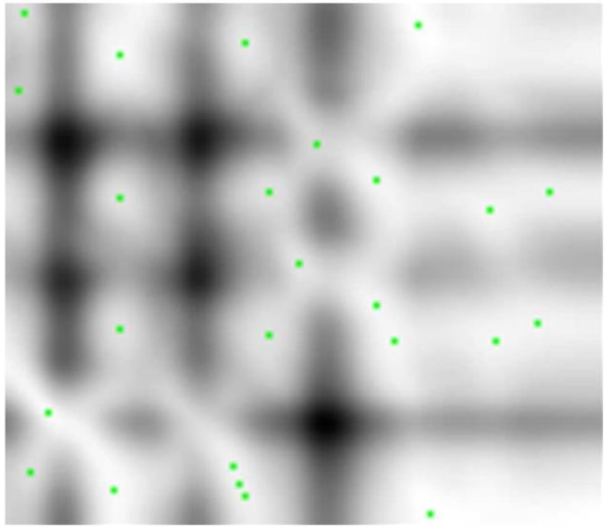


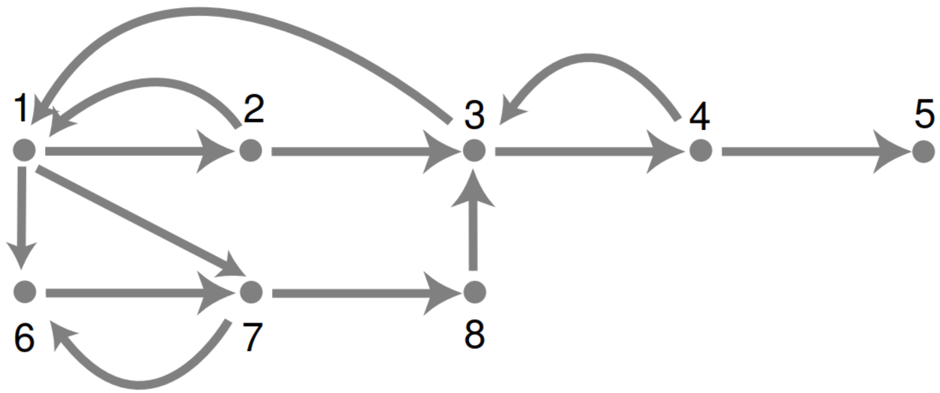
## Motion Graphs

Lucas Kovar
University of Wisconsin-Madison

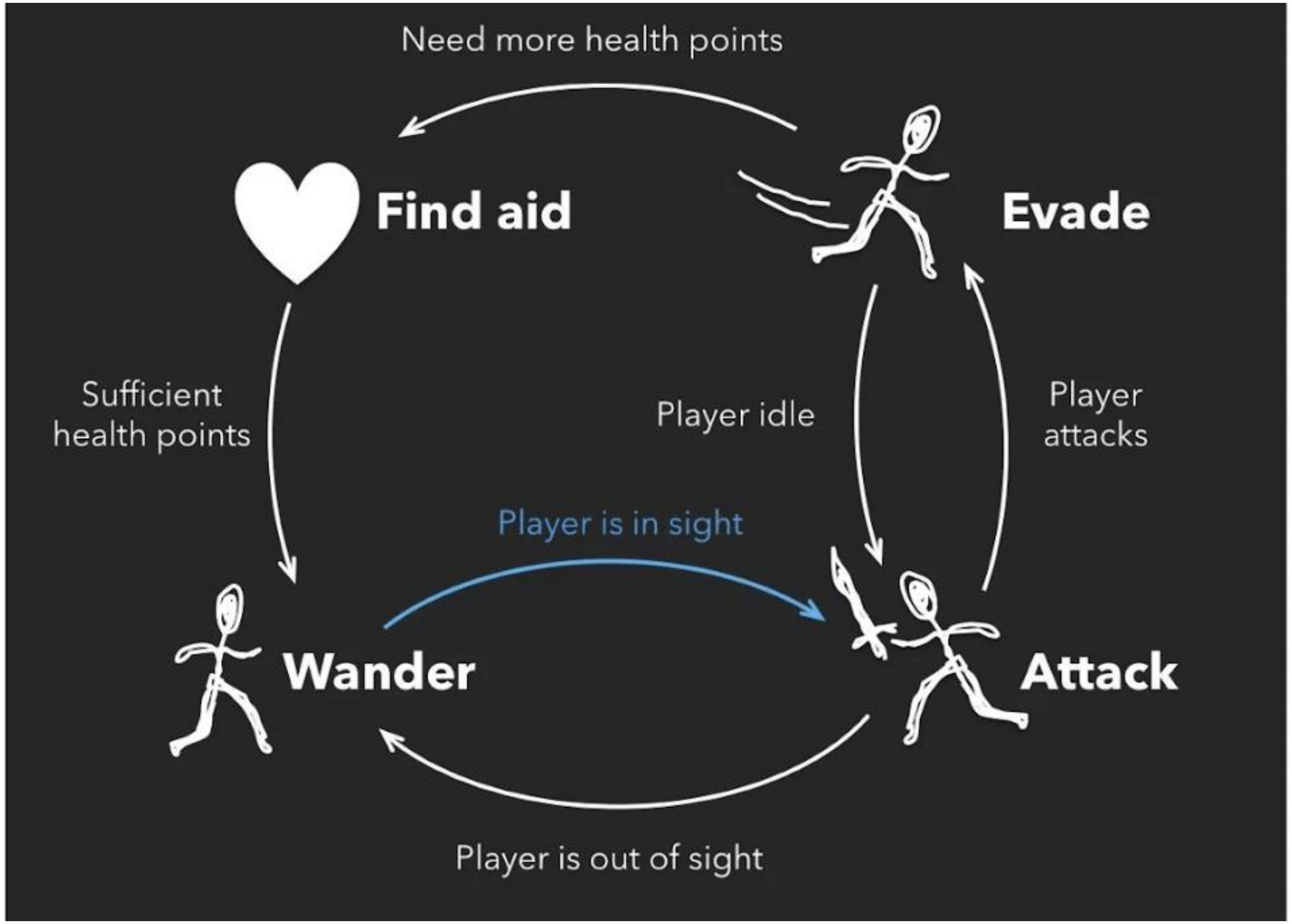
Michael Gleicher*
University of Wisconsin-Madison

Frédéric Pighin†
University of Southern California
Institute for Creative Technologies

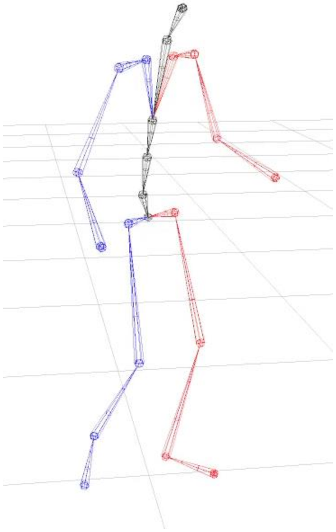


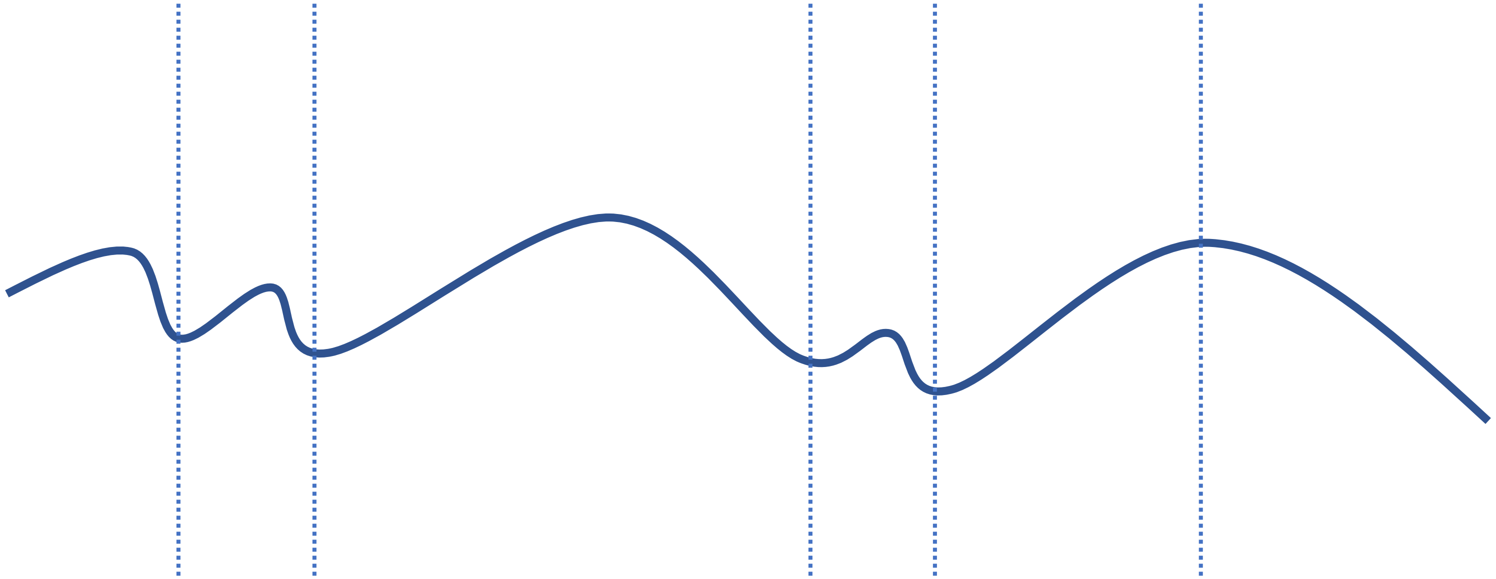


# Motion Graphs



# Segment Motion Data

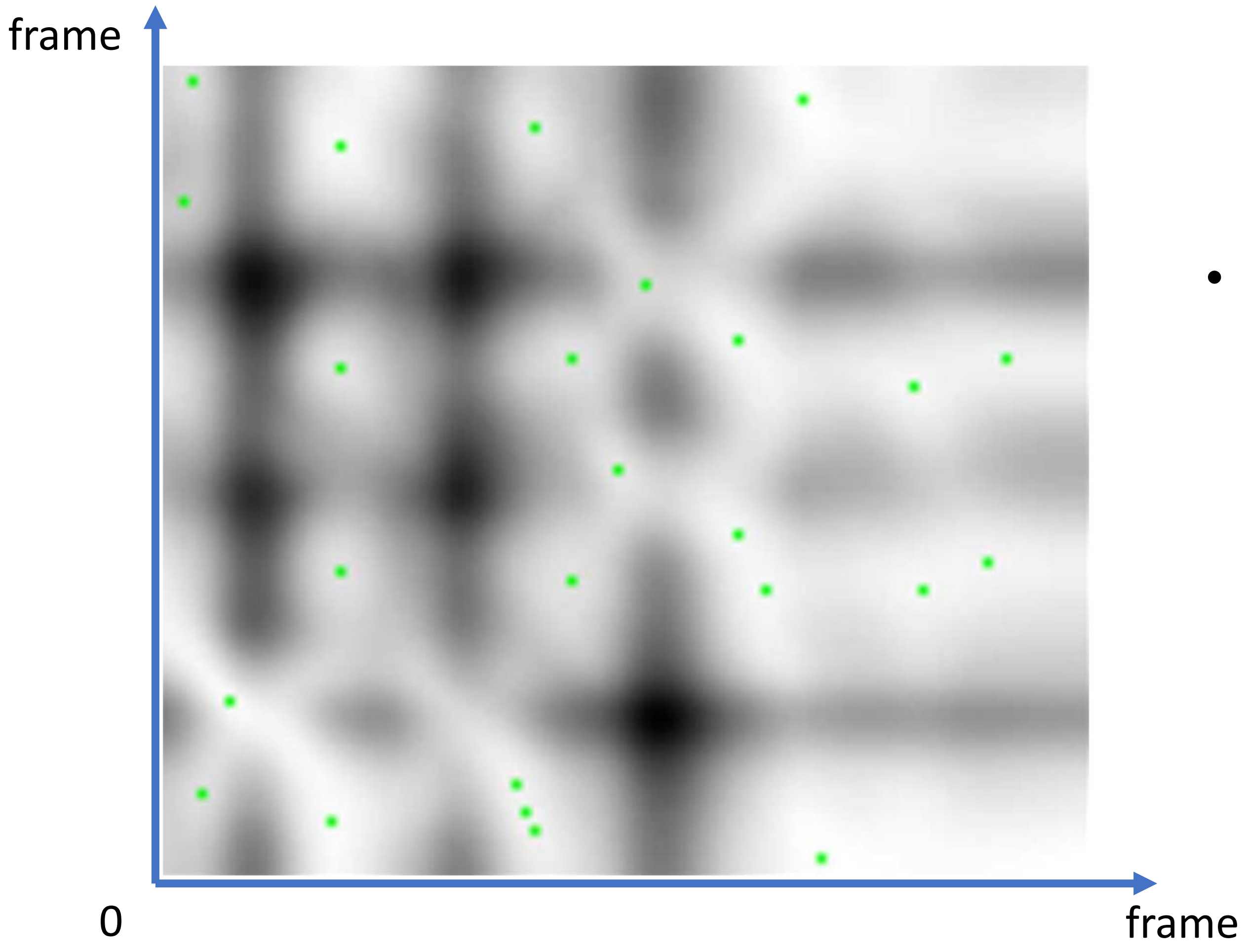




turn walk forward fall getup walk backward stop

# Segment Motion Data

- Where to segment?

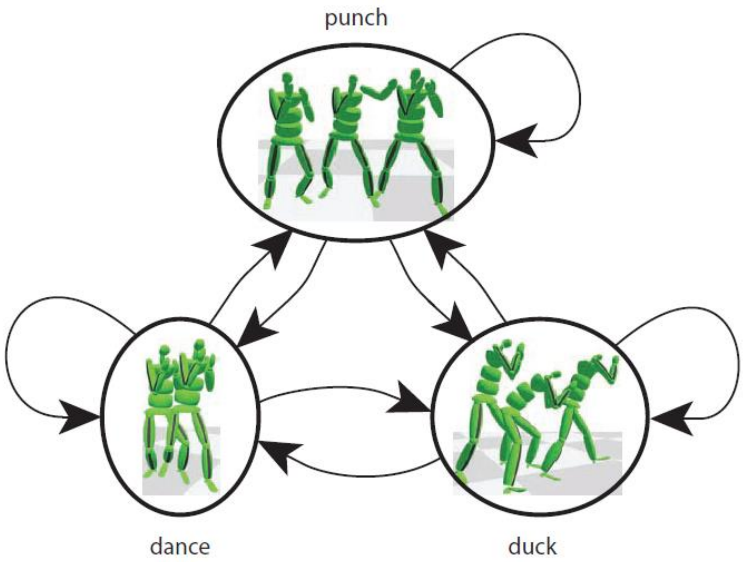


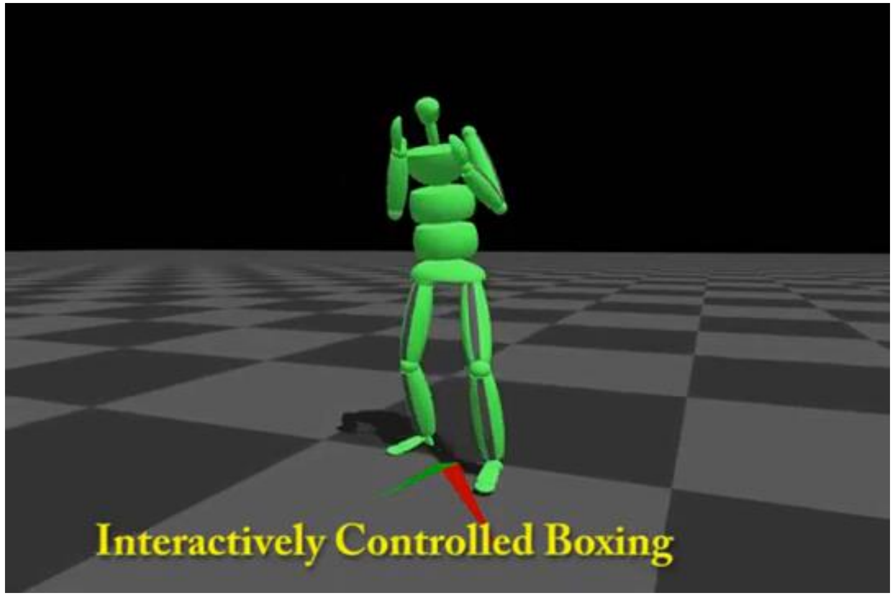
- Distance map
  - Each pixel represents the difference between a pair of poses
  - Local minima are potential transition point

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. ***Motion graphs***. *ACM Trans. Graph.* 21, 3 (July 2002),

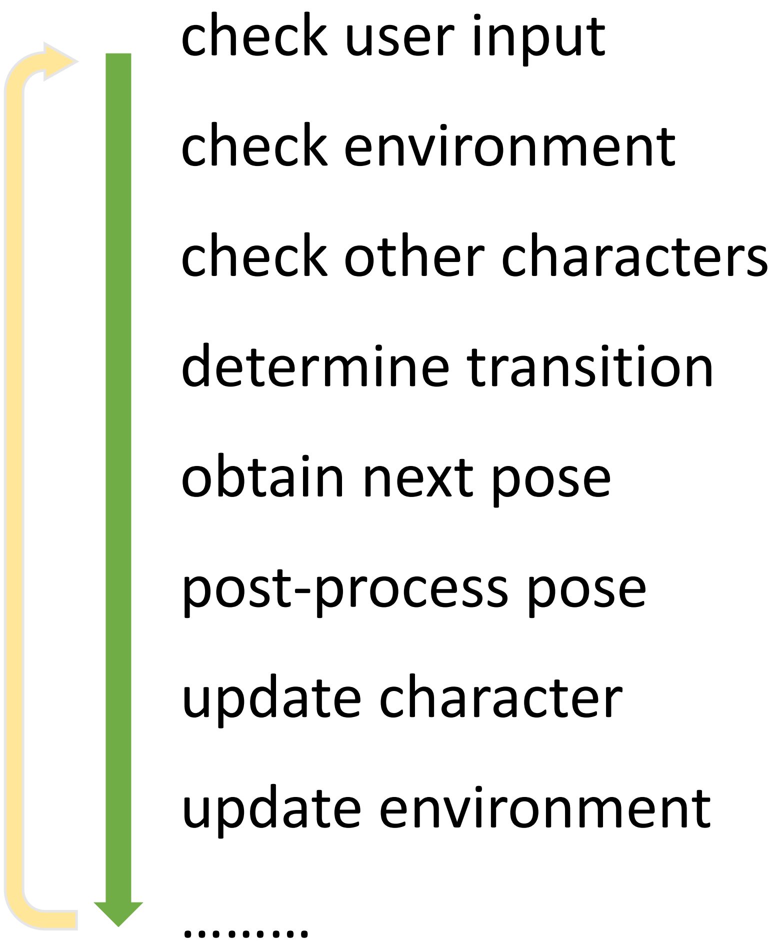
# Motion Synthesis

- State-machines
  - Nodes represent motion clips
  - Edges represent potential transitions
  - Transitions are triggered when necessary
    - User input
    - Clip end
  - Check immediate connections for the next clip
    - May need deeper search





# Interactive Animation Pipeline



check user input

check environment

check other characters

determine transition

obtain next pose

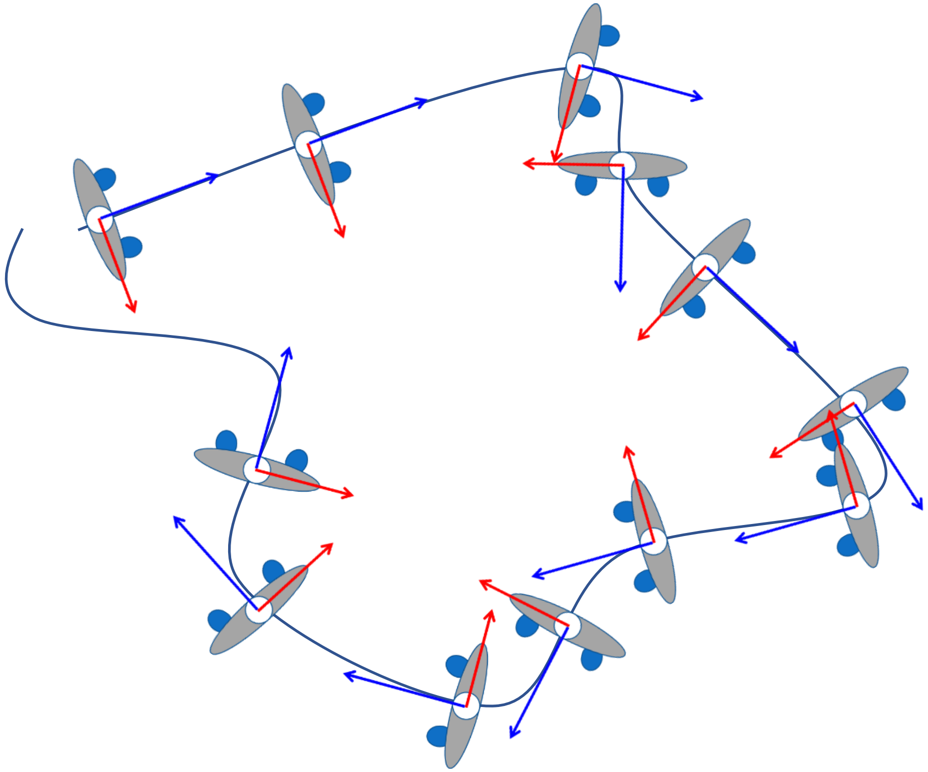
post-process pose

update character

update environment

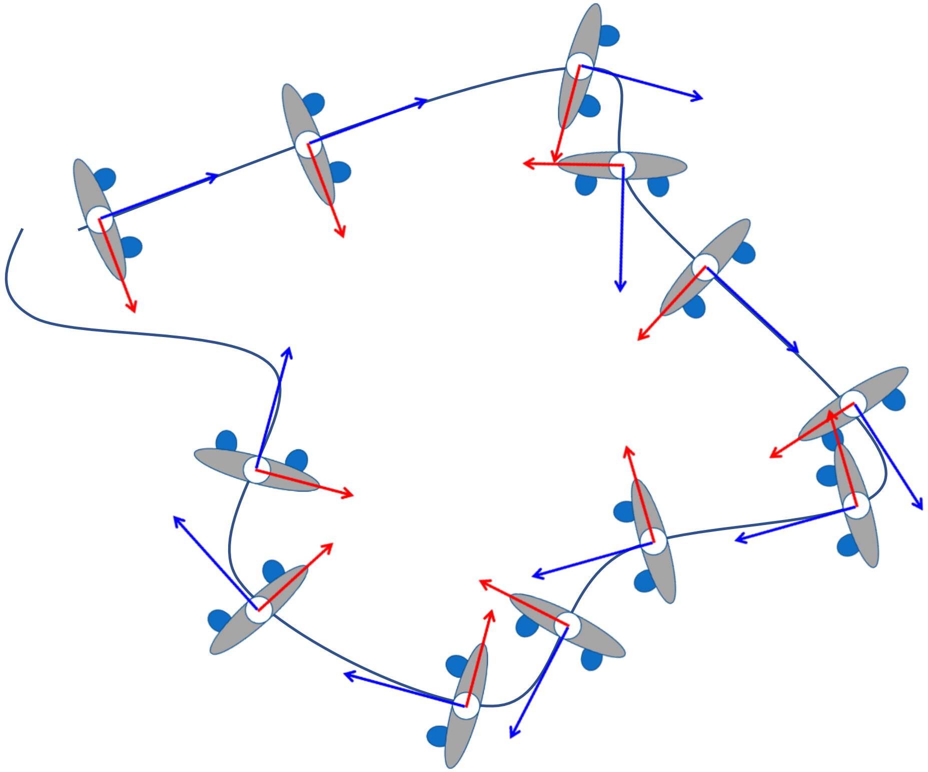
………

# Example: Path Fitting



[Motion Graphs]

# Example: Path Fitting



[Motion Graphs]

# Motion Matching?

- Clip → Pose
- Short clip →
  “Raw” and long motion data

UBISOFT
TORONTO
@KrisZadziuk
- Motion Matching -
The Future of Games Animation... Today
- Kristjan Zadziuk - Animation Director - Ubisoft Toronto -

# Outline

- Motion Capture
    - History and modern mocap systems

- Motion Synthesis
    - Motion retargeting
    - Motion transition
    - Motion graph

# Questions?