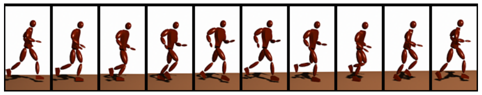
Motion Capture



Motion Capture in Movies

Motion Capture in Games

Magnetic Capture Systems

- Tethered
- Sensitive to metal
- Low frequency (60Hz)

Mechanical Capture Systems

- Any environment
- Measures joint angles
- Restricts the motion

Optical motion capture

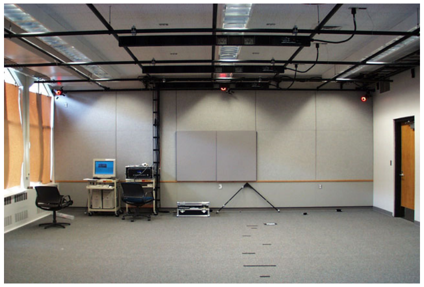
- Place markers on the actor

- Cameras can determine marker positions

Optical Capture Systems

- 8 or more cameras
- Restricted volume
- High Frequency (240Hz)
- Occlusions

How Does It Work?



8 cameras + 120 Hz + Special tape = Raw Point Data

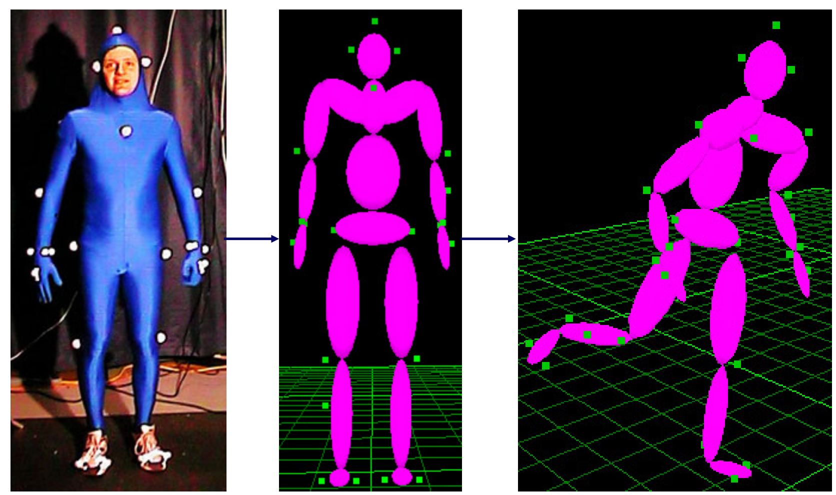
Optical motion capture process

1. Find the skeleton dimensions
   and exact marker positions on the body
2. Perform a motion trial
3. Compute marker positions from camera images
4. Identify and uniquely label markers
5. Calculate joint angles from maker paths

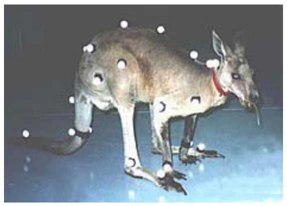
Optical motion capture process

1. **Find the skeleton dimensions**
   and exact marker positions on the body
2. Perform a motion trial
3. Compute marker positions from camera images
4. Identify and uniquely label markers
5. Calculate joint angles from maker paths

Problem Statement



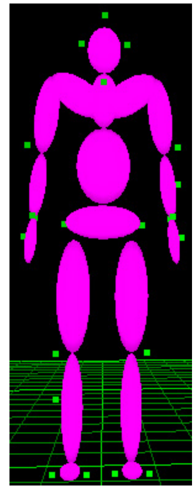
Automatic Calibration



Design Goals:

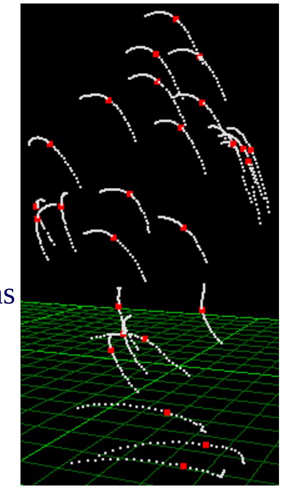
- Fully automatic
- Any skeleton
- Accurate

Input



Generic Skeleton

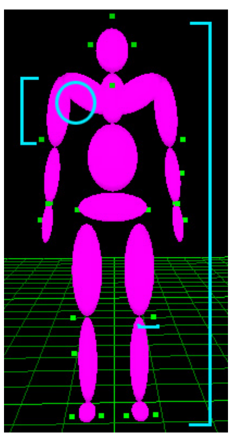
Actor's kinematics structure, and rough handle positions



Calibration Data

Initial path data that exercises all of the subject's DOFs

Independent Variables



DOFs

Bone lengths

Handle offsets

Global scale

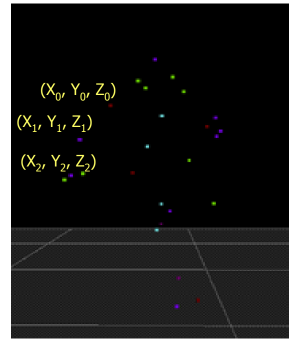
Optical motion capture process

1. Find the skeleton dimensions and exact marker positions on the body
2. Perform a motion trial
3. Compute marker positions from camera images
4. **Identify and uniquely label markers**
5. Calculate joint angles from maker paths

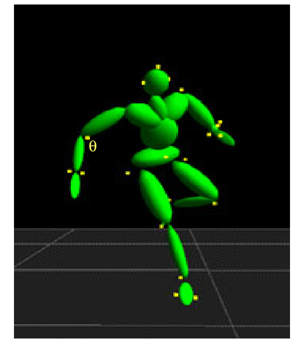
Optical motion capture process

1. Find the skeleton dimensions
   and exact marker positions on the body
2. **Perform a motion trial**
3. **Compute marker paths from camera images**
4. Identify and uniquely label markers
5. Calculate joint angles from maker paths

Marker Identification

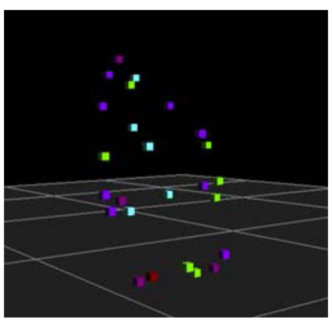


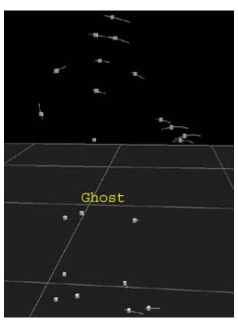
At each frame, motion capture gives us a set of points

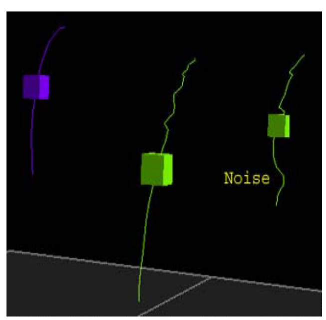


We would like something more intuitive

Marker Identification Problems





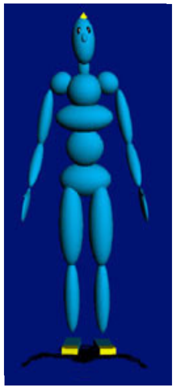


Making sense of raw data…

Optical motion capture process

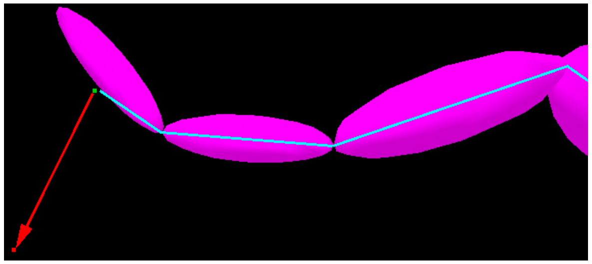
1. Find the skeleton dimensions
   and exact marker positions on the body
2. Perform a motion trial
3. Compute marker positions from camera images
4. Identify and uniquely label markers
5. **Calculate joint angles from maker paths**

IK Problem Definition



1. Create a handle on body
   - position or orientation
2. Pull on the handle
3. IK figures out how joint angles should change

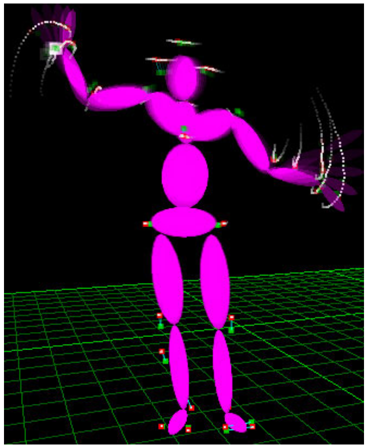
Inverse Kinematics



Inputs:
An articulated skeleton with handles. Desired positions for handles.

Outputs:
Joint angles that move handles to desired positions.

Inverse Kinematics (con't)



We are solving IK on a complex model (~50 DOFs and 30 handles).

Motion capture data often contains missing markers.

Many different formulations for IK problem, would like to use one that is best for motion capture data.

More Formally

Let:

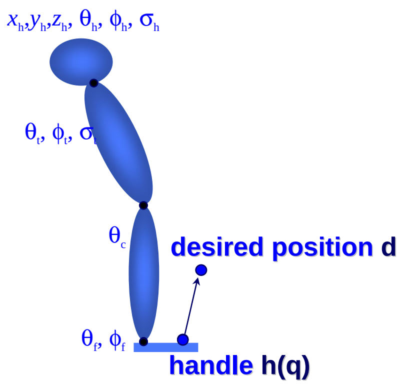
q	**actor *state vector* (joint bundle)**
C(q)	**constraint functions that pull handles**

Then:

solve for q such that C(q) = 0

What’s a Constraint?

$\mathbf{q}=[\, x_h, y_h, z_h, \theta_h, \phi_h, \sigma_h, \theta_t, \phi_t, \sigma_t, \theta_c, \theta_f, \phi_f \,]$



- Can be rich, complicated
- But most common is very simple:
- Position constraint just sets difference of two vectors to zero:

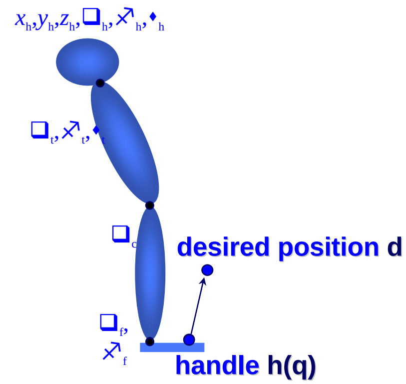
C(q) = h(q) - d = 0

Constraint derivatives

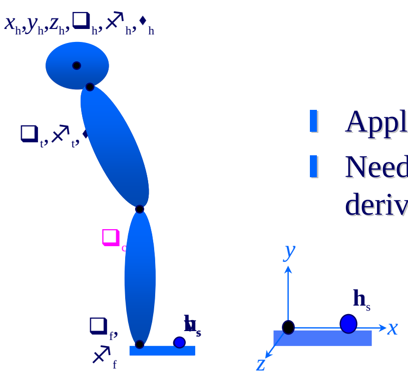
$\mathbf{q}=[\, x_h, y_h, z_h, ❑_h, ↗_h, ♦_h, ❑_t, ↗_t, ♦_t, ❑_c, ❑_f, ↗_f \,]$

C(q) = h(q) - d = 0

$$\frac{\partial C(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial h(\mathbf{q})}{\partial \mathbf{q}}$$



Computing Derivatives

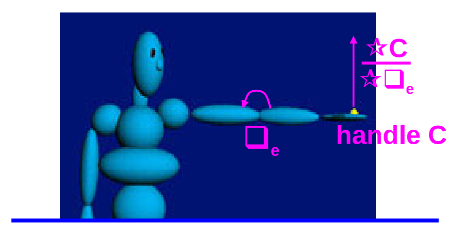


- Apply the chain rule
- Need to know how to compute derivatives for each transformation

$$h_w = \mathbf{T}(x_h, y_h, z_h)\mathbf{R}(\theta_h, \phi_h, \sigma_h)\ \mathbf{TR}(\theta_t, \phi_t, \sigma_t)\ \ \mathbf{T}\mathbf{R}(\theta_c)\mathbf{TR}(\theta_f, \phi_f)\ h_s$$

$$\frac{\partial h_w}{\partial \theta_c} = \mathbf{T}(x_h, y_h, z_h)\mathbf{R}(\theta_h, \phi_h, \sigma_h)\ \mathbf{TR}(\theta_t, \phi_t, \sigma_t)\ \ \mathbf{T}\frac{\partial \mathbf{R}(\theta_c)}{\partial \theta_c}\ \mathbf{TR}(\theta_f, \phi_f)\ h_s$$

Jacobian Matrix



☆C / ☆q

		θ_e	
		0	
		1	
		0	

- Can compute Jacobian for each constraint / handle
- Value of Jacobian depends on current state
- Jacobian linearly relates joint angle velocity to constraint velocity

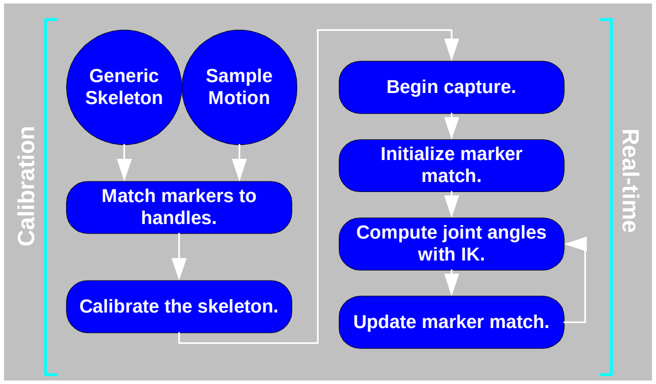
Unconstrained Optimization

- Minimize $G'(q) = G(q) + \sum_i w_i C_i(q)^2$
- Move in the direction of the objective function gradient:

$$\frac{\partial G'}{\partial q} = \frac{\partial G}{\partial q} + 2\sum_i w_i C_i \frac{\partial C_i}{\partial q}$$

$$q = q_o + \alpha \frac{\partial G'}{\partial q}$$

Real-time Motion Capture



Motion capture as UI

- Use acting for animation interface

Motion Transformation

- Start with a mocap sequence
- Edit it to fit the needs of the animation
- Try to be as close to the original motion as possible