**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

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 marker 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 marker 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:

\[ q \quad \text{actor state vector} \]
\[ \text{(joint bundle)} \]

\[ C(q) \quad \text{constraint functions that pull handles} \]

Then:

\[ \text{solve for } q \quad \text{such that } \quad C(q) = 0 \]
What’s a Constraint?

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

\[ C(q) = h(q) - d = 0 \]

Real-time Motion Capture

Motion capture as UI

- Map a “whiteboard space” anywhere
- 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