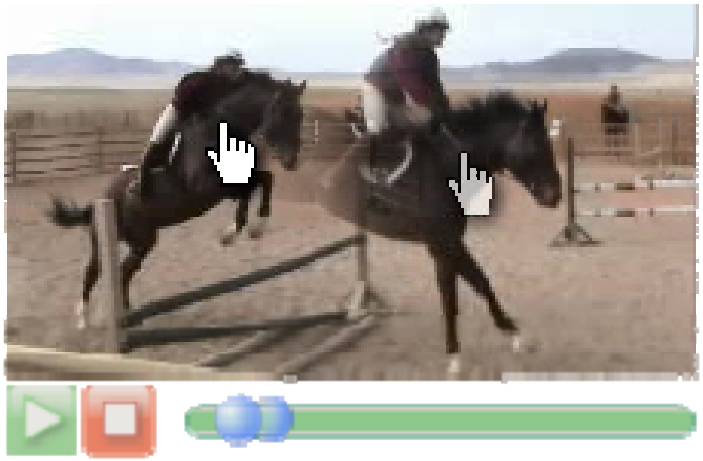
Video Browsing By Direct Manipulation



Pierre Dragicevic
Gonzalo Ramos
Jacobo Bibliowitcz
Derek Nowrouzezahrai
Ravin Balakrishnan
Karan Singh

dgp, University of Toronto
INRIA, France

Video Browsing
By Direct Manipulation

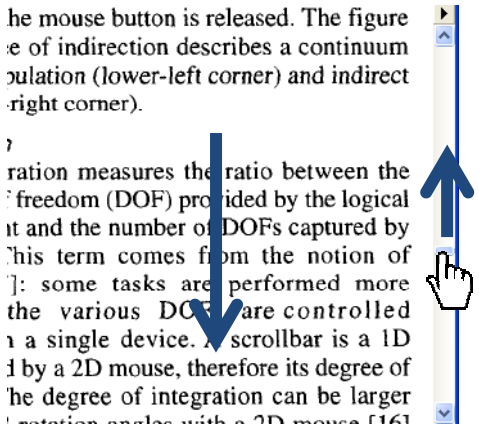
- Sample Scenario

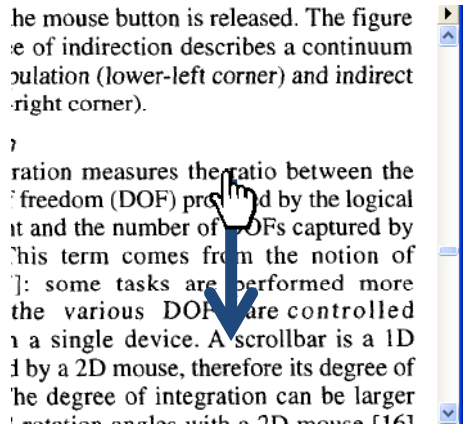
Direct Manipulation

- **Shneiderman (1974, 1982)**
  - Continuous representation of objects
  - Rapid, incremental, reversible actions
  - **Easy to learn, efficient, less errors**
- **Hutchins, Hollan & Norman (1985)**
  - **Low cognitive effort** needed to manipulate the system and evaluate the results
  - Feeling of **direct engagement**

Directness

- **Hutchins, Hollan & Norman (1985, cont'd)**
  - How well the **input** language of the GUI **matches** its **output** language
- **Michel Beaudouin-Lafon (2000)**
  - Degree of **indirection**
  - Degree of **compatibility**
  - Degree of **integration**





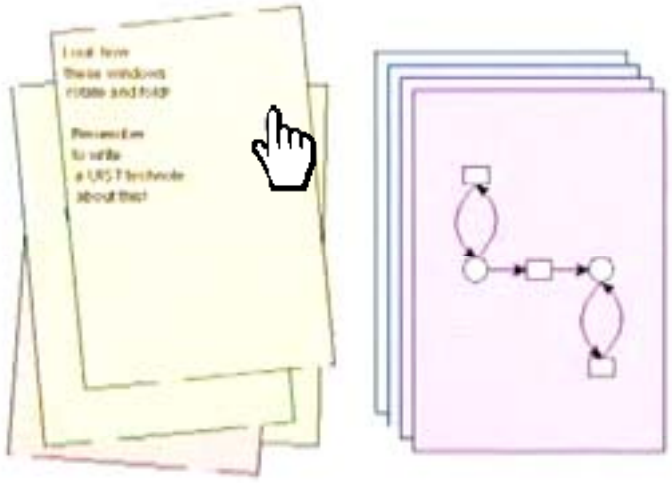
Direct Manipulation
As a Mapping Problem

- We already have a visual representation of objects of interest
- We want to be able to manipulate them
- Mapping user's gestures with existing visual motion

Direct Manipulation
As a Mapping Problem

- **User's Gesture Space:**
  - 2D mouse translations + buttons

- **Screen Motion Space:**
  - Any observable motion on the screen

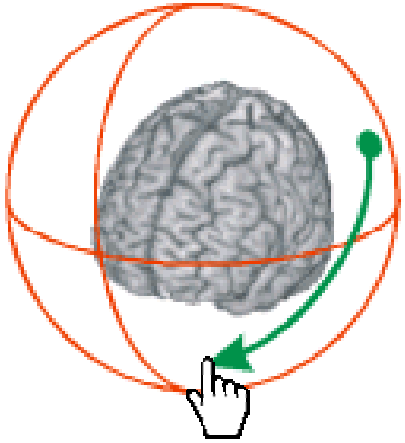


2D translations → 2D translations + 1D rotations [Beaudouin-Lafon 01]

Direct Manipulation
As a Mapping Problem

- **User's Gesture Space:**
  - 2D mouse translations + buttons

- **Screen Motion Space:**
  - Any observable motion on the screen

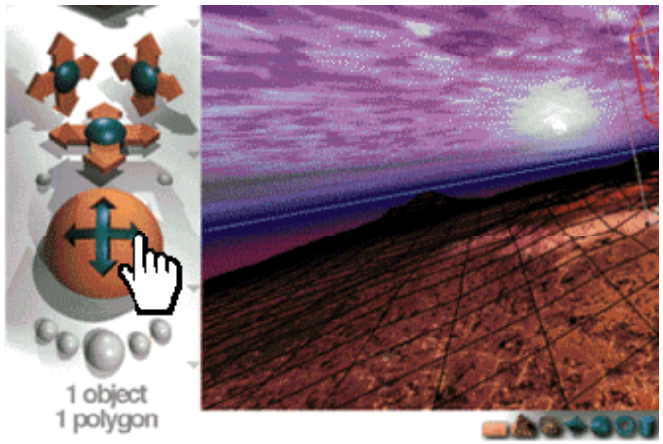


2D translations → 3D rotations [Shoemake 92]

Direct Manipulation
As a Mapping Problem

- User's Gesture Space:
  - 2D mouse translations + buttons

- Screen Motion Space:
  - Any observable motion on the screen



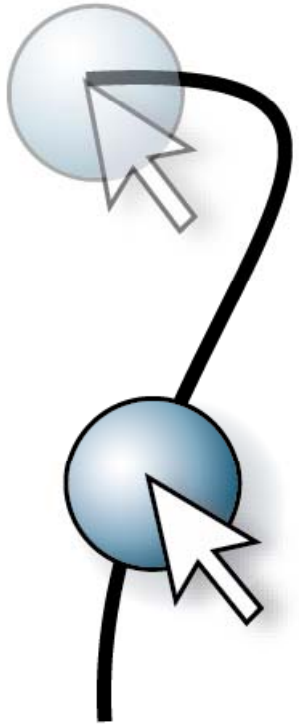
2D translations → 6DOF rigid motions

Direct Manipulation
As a Mapping Problem

- Control of High-Dimensional Motion Spaces:
  - Lots of Work
- Control of Low-Dimensional Motion Spaces:
  - Almost no Work
- 1-DOF Direct Manipulation

Curvilinear Dragging

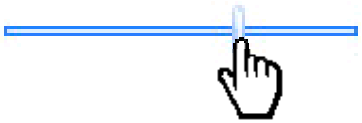
= Manipulating an object whose translation is constrained to a curve

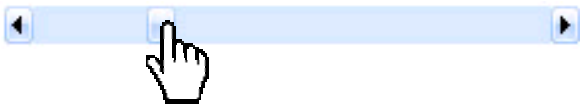


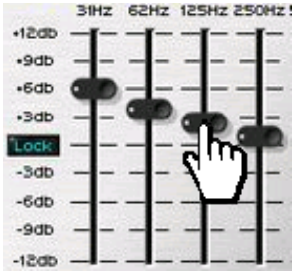
Curvilinear Dragging

= Manipulating an object whose translation is constrained to a curve

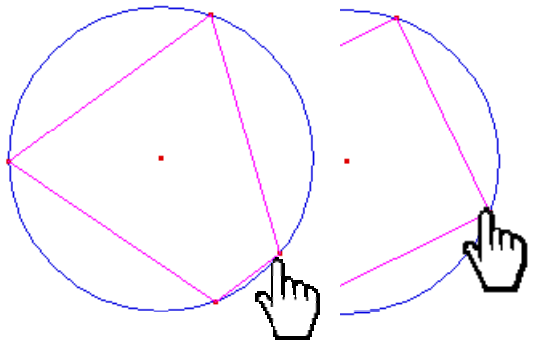
- Straight lines:



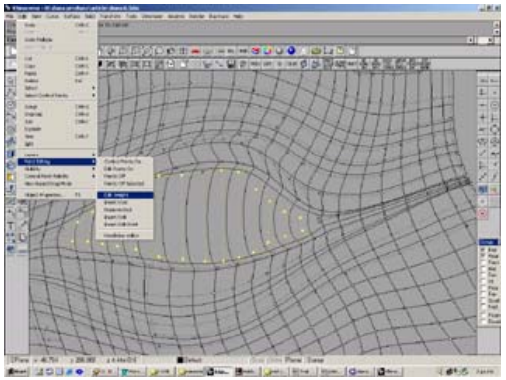




- Other:



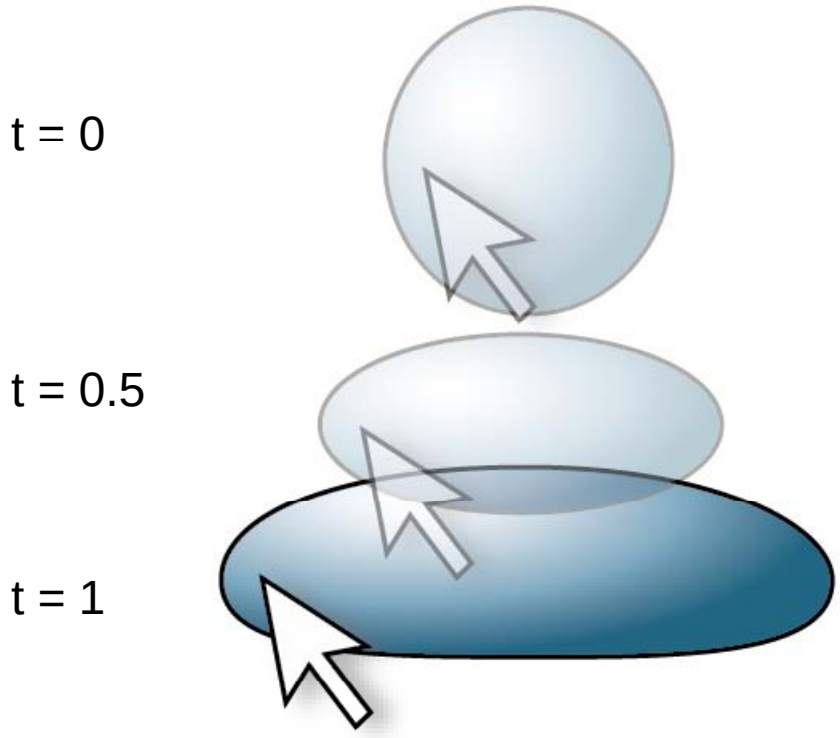
Cabri Géomètre



Autodesk Maya

Flow Dragging

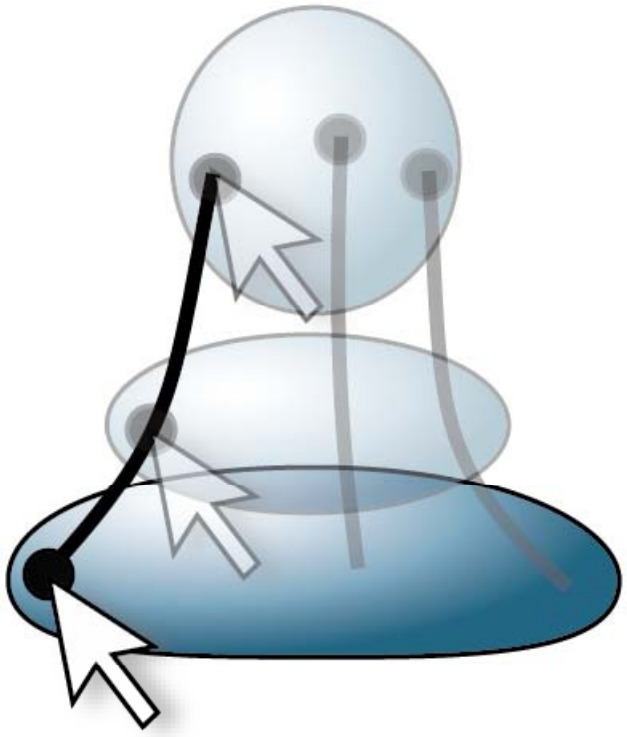
= Manipulating an object whose motion is constrained to a 1-D parameter



Flow Dragging

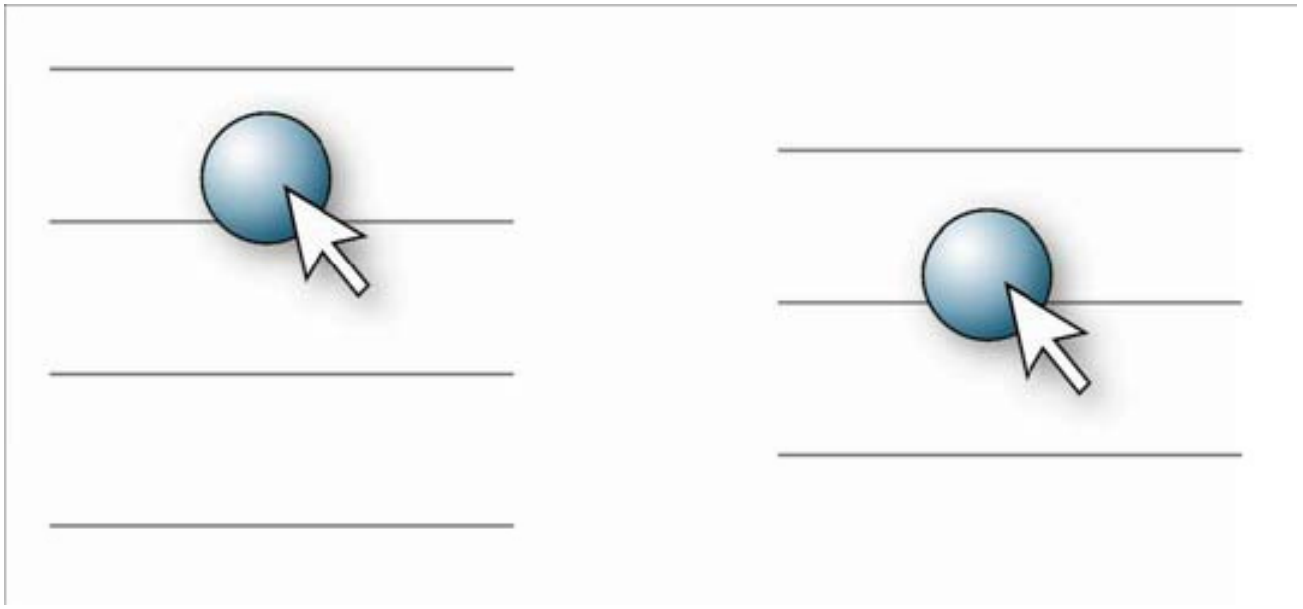
= Manipulating an object whose motion is constrained to a 1-D parameter

- Equivalent to curvilinear dragging on a family of curves

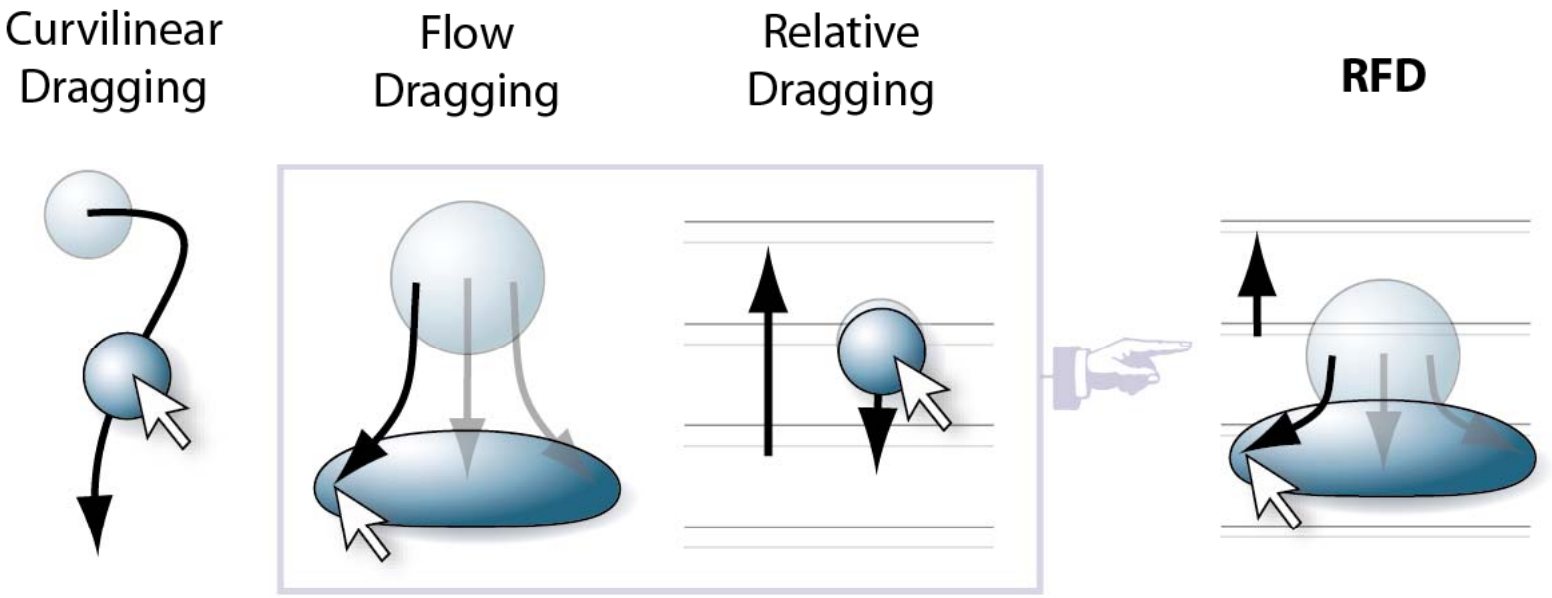


Relative Dragging

= Manipulating an object's relative motion ("induced motion")

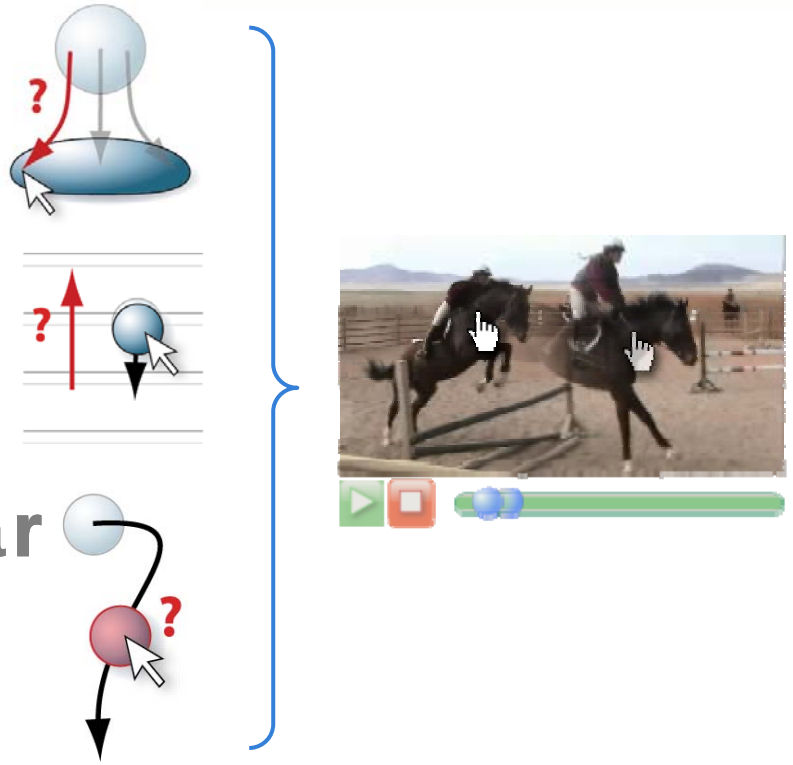


RFD
Relative Flow Dragging



Supporting RFD on Video Clips

- Extracting motions
  - Objects motion
  - Background motion
- Supporting Curvilinear Dragging



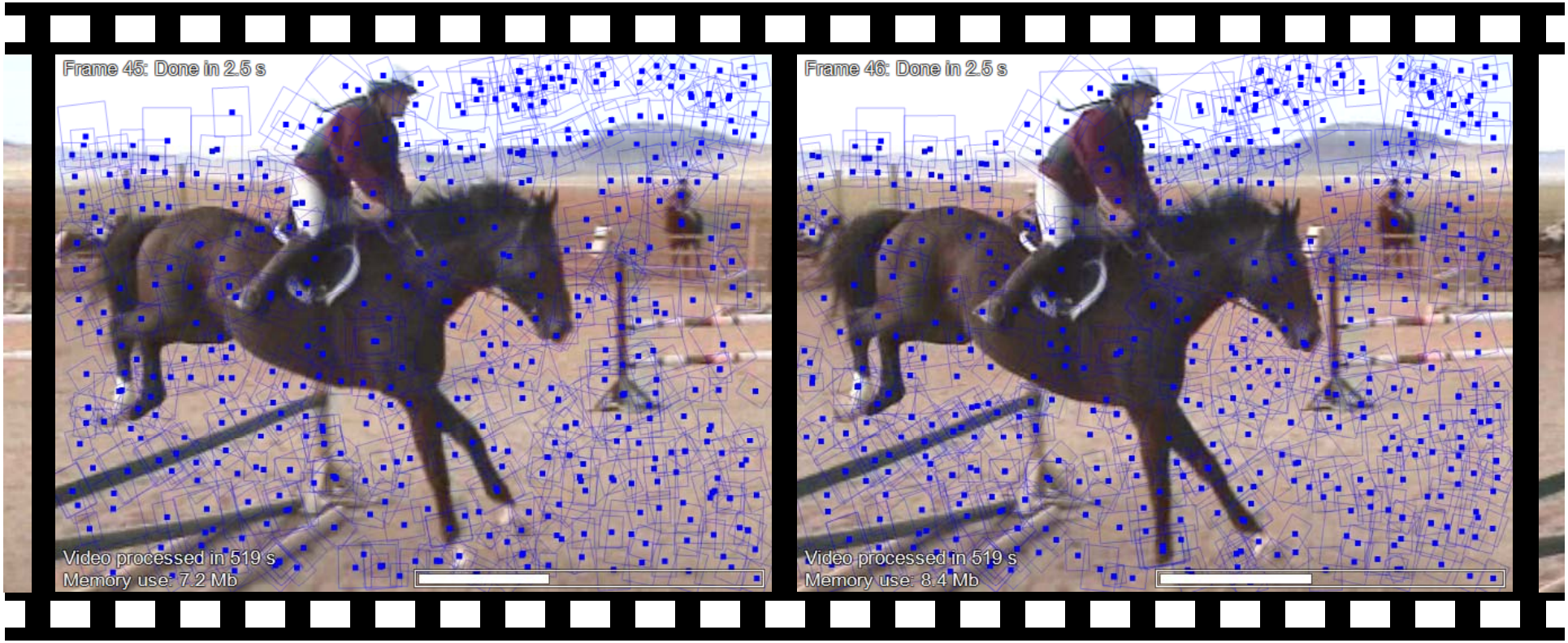
Extracting Motions from Videos

- Manual Annotation vs. Automatic
- Object tracking vs. Optical flow
  - Scale Invariant Feature Transforms (SIFT) [Lowe 2004]

Extracting Motions from Videos

Extracting Motions from Videos

1. Feature Detection [Lowe 2004]



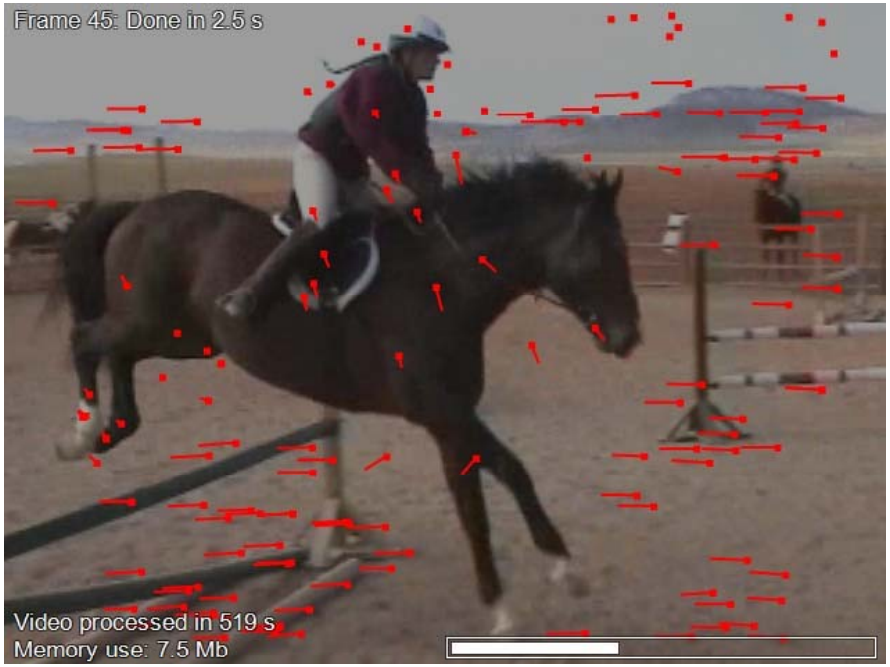
Extracting Motions from Videos

1. Feature Detection
2. Feature Matching [Lowe 2004]

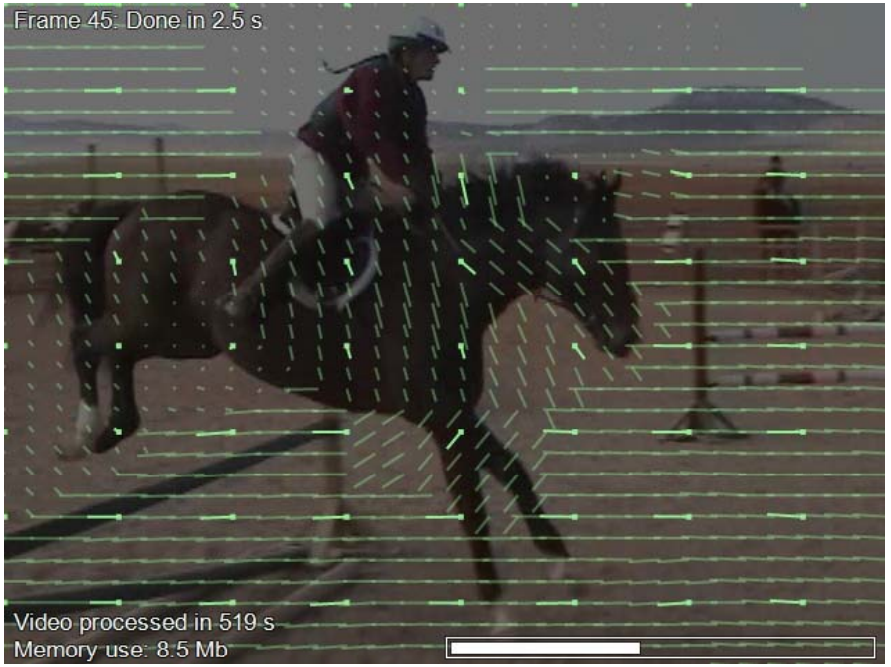
Feature Motions

Extracting Motions from Videos

1. Feature Detection
2. Feature Matching
3. **Interpolation**



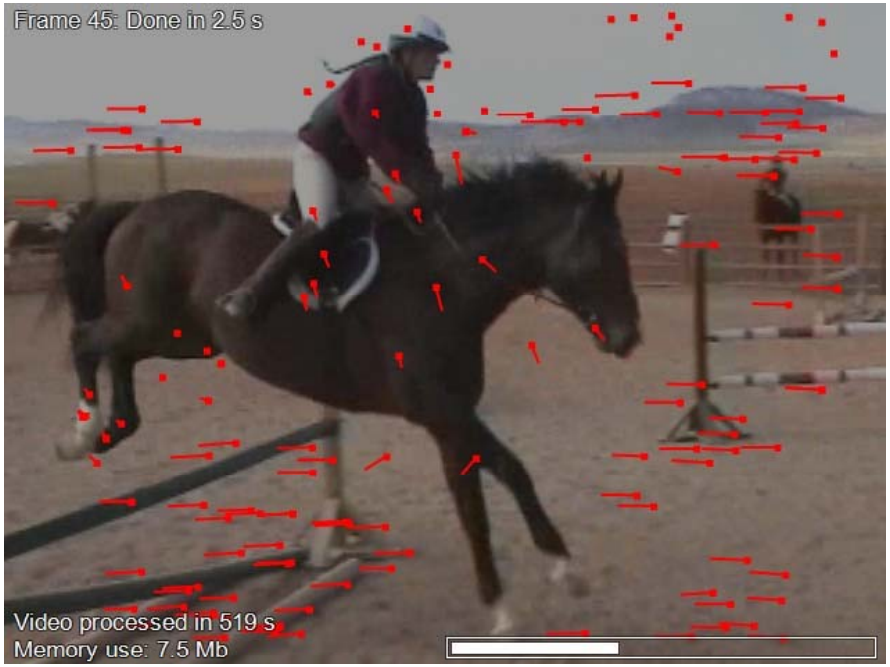
Feature Motions



Optical Flow

Extracting Motions from Videos

1. Feature Detection
3. Feature Matching
4. Interpolation
5. **Trajectory Construction**



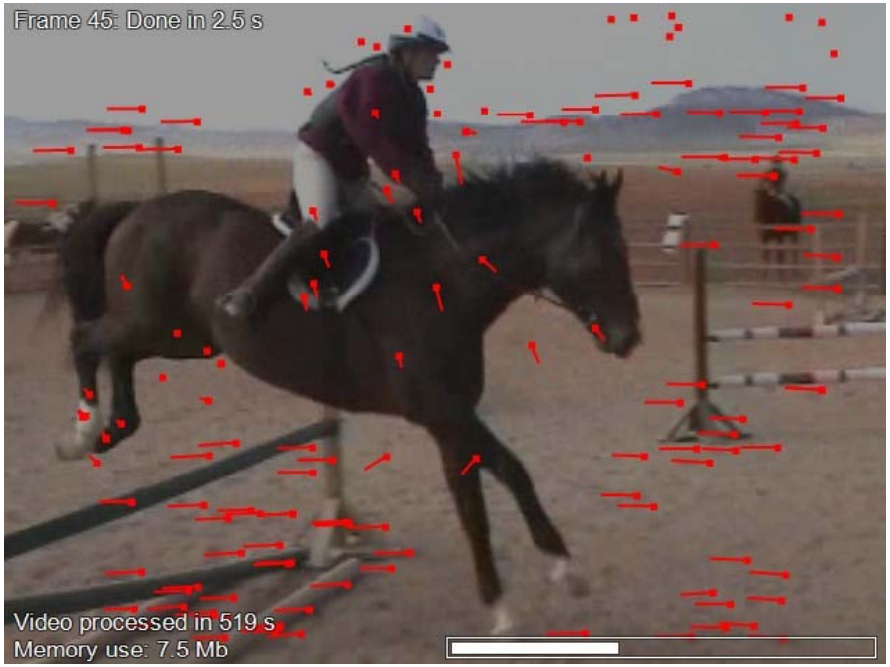
Feature Motions

“Pixel” Trajectories

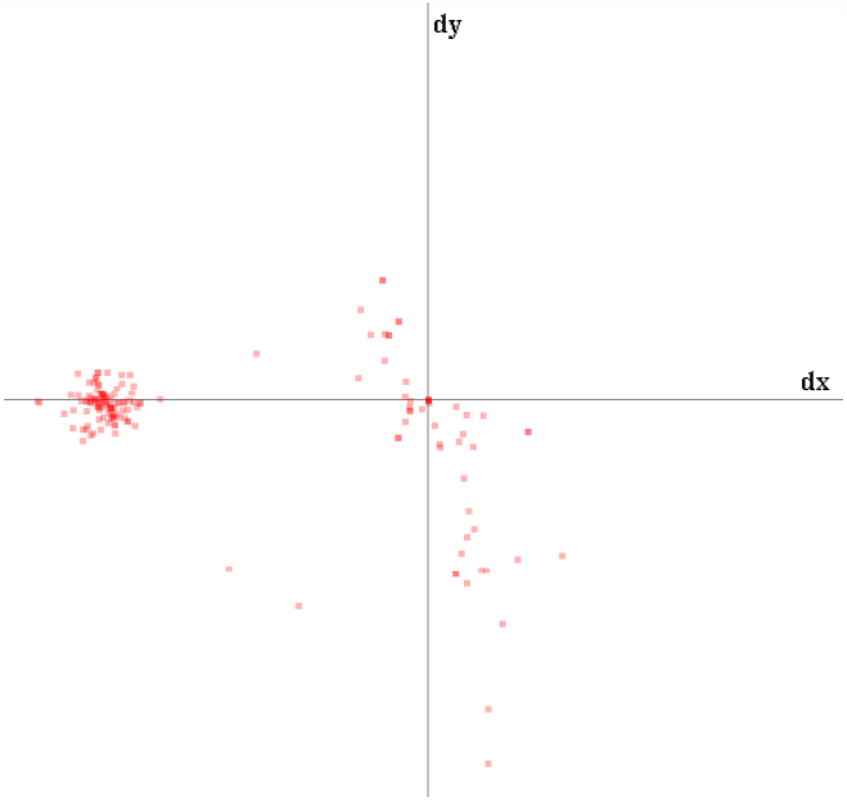
Extracting Motions from Videos

1. Feature Detection
3. Feature Matching
4. Interpolation
5. Trajectory Construction

2b. Background Motion Estimation



Feature Motions

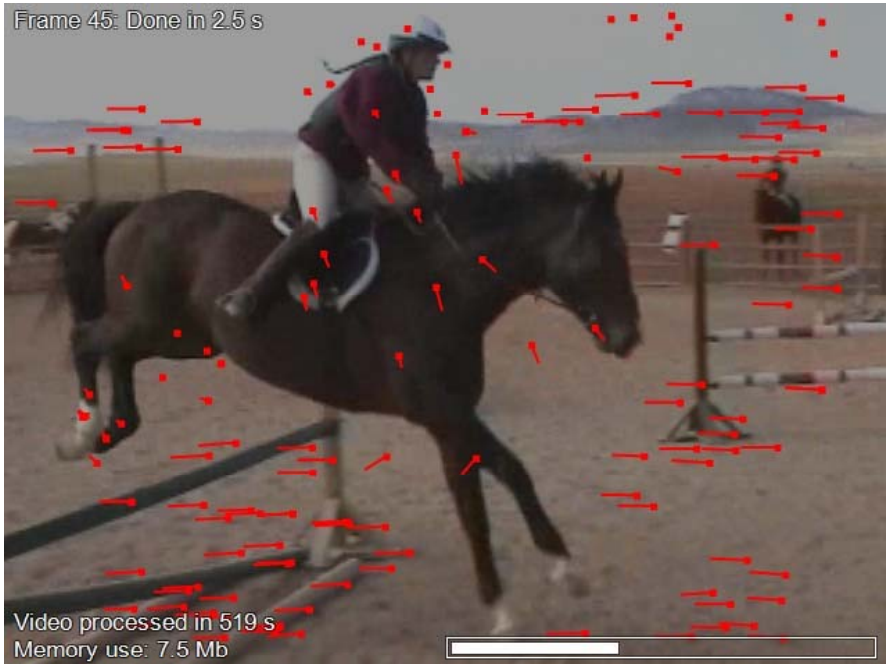


Feature Motions Plot

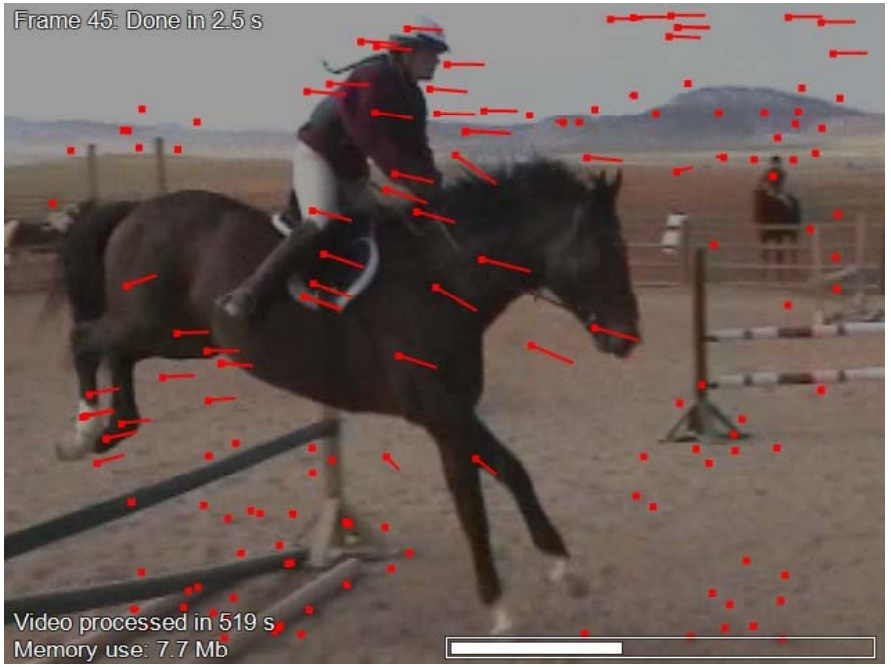
Extracting Motions from Videos

1. Feature Detection
2. Feature Matching
3. Interpolation
4. Trajectory Construction

{ 2b. Background Motion Estimation
{ **2c. Background Motion Compensation**



Absolute Feature Motions

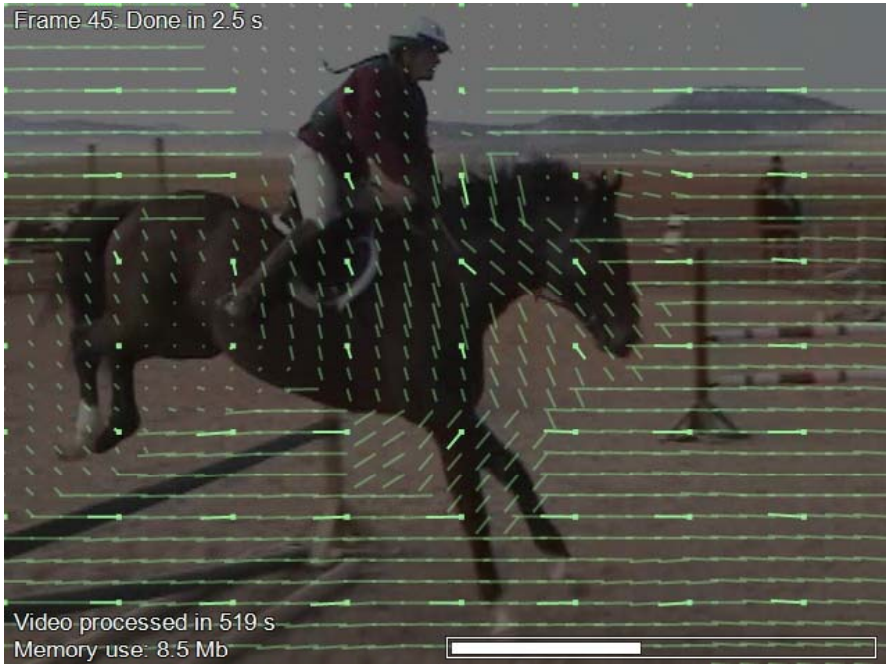


Relative Feature Motions

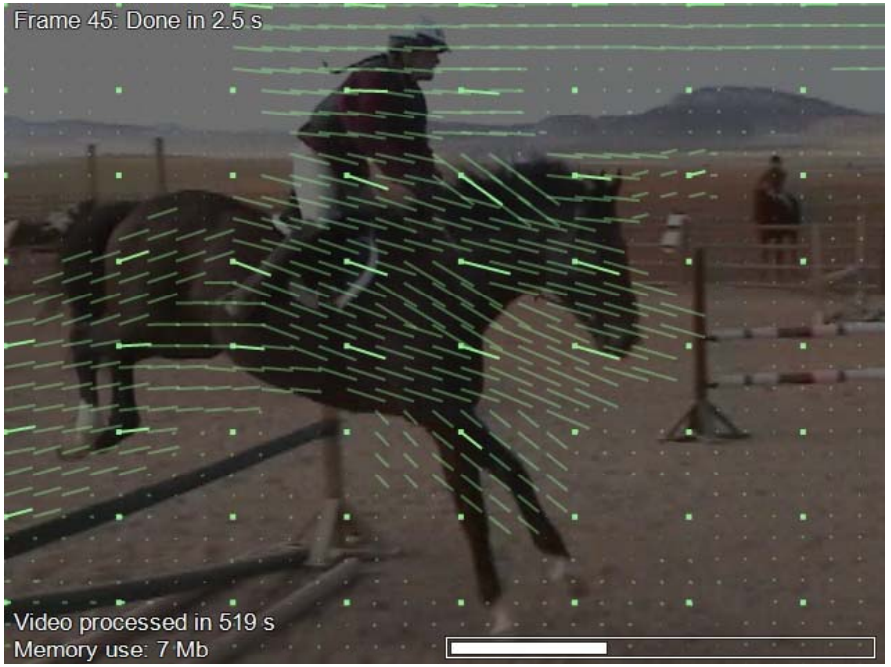
Extracting Motions from Videos

1. Feature Detection
2. Feature Matching
3. Interpolation
4. Trajectory Construction

{ 2b. Background Motion Estimation
{ **2c. Background Motion Compensation**



Absolute Optical Flow



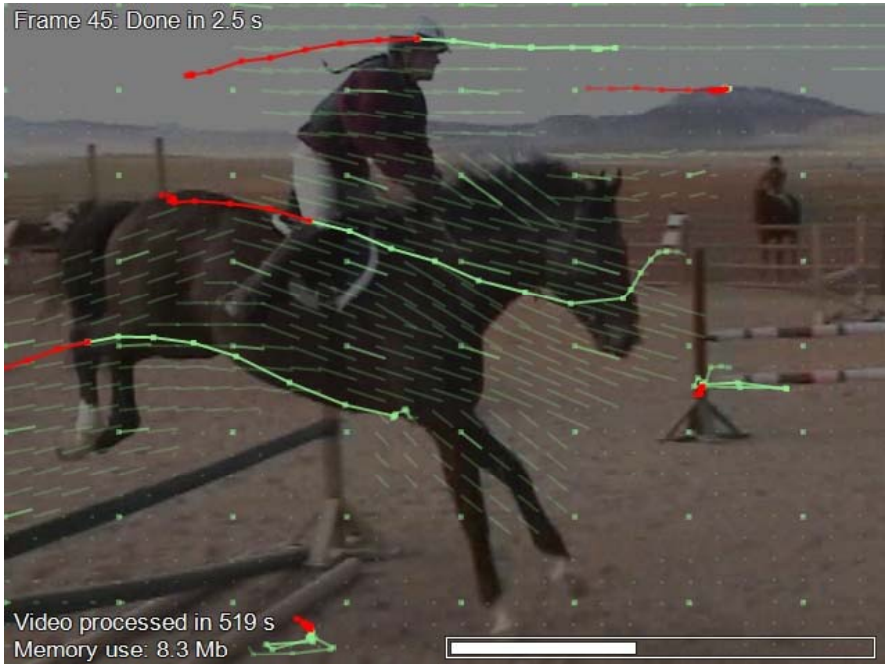
Relative Optical Flow

Extracting Motions from Videos

1. Feature Detection
2. Feature Matching
3. Interpolation
4. Trajectory Construction

{ 2b. Background Motion Estimation
{ **2c. Background Motion Compensation**

Absolute Trajectories



Relative Trajectories

Extracting Motions from Videos

1. Feature Detection
2. Feature Matching
3. Interpolation
4. Trajectory Construction

2b. Background Motion Estimation

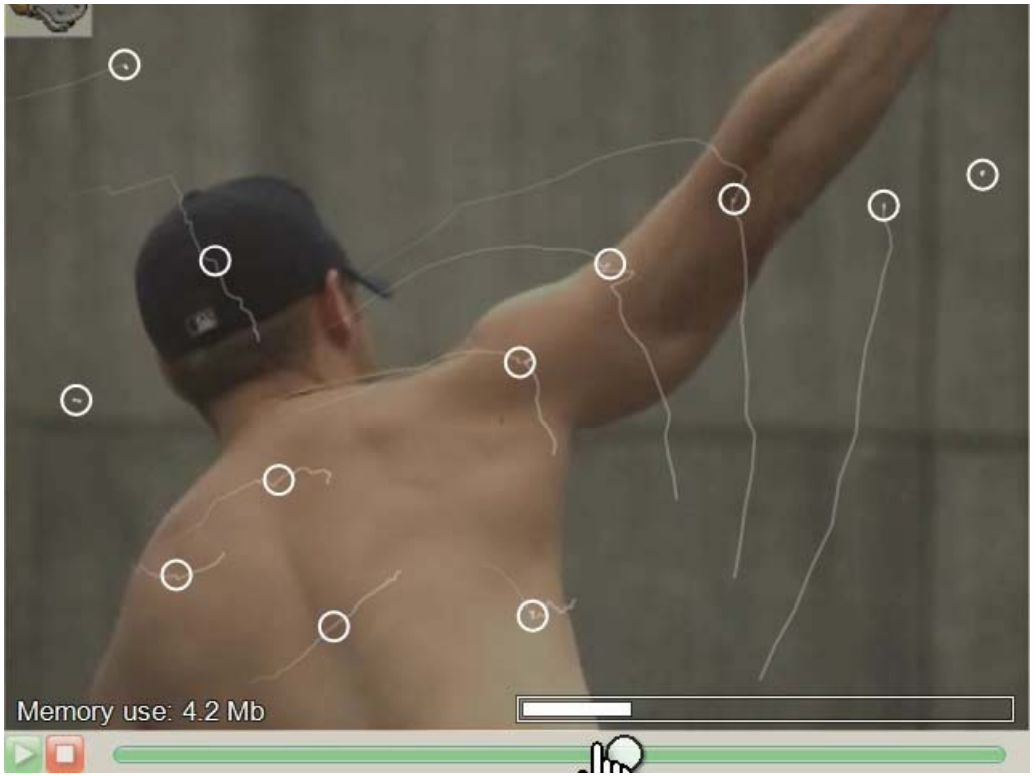
2c. Background Motion Compensation

Pre-computed (2-5 sec./frame)

Real time

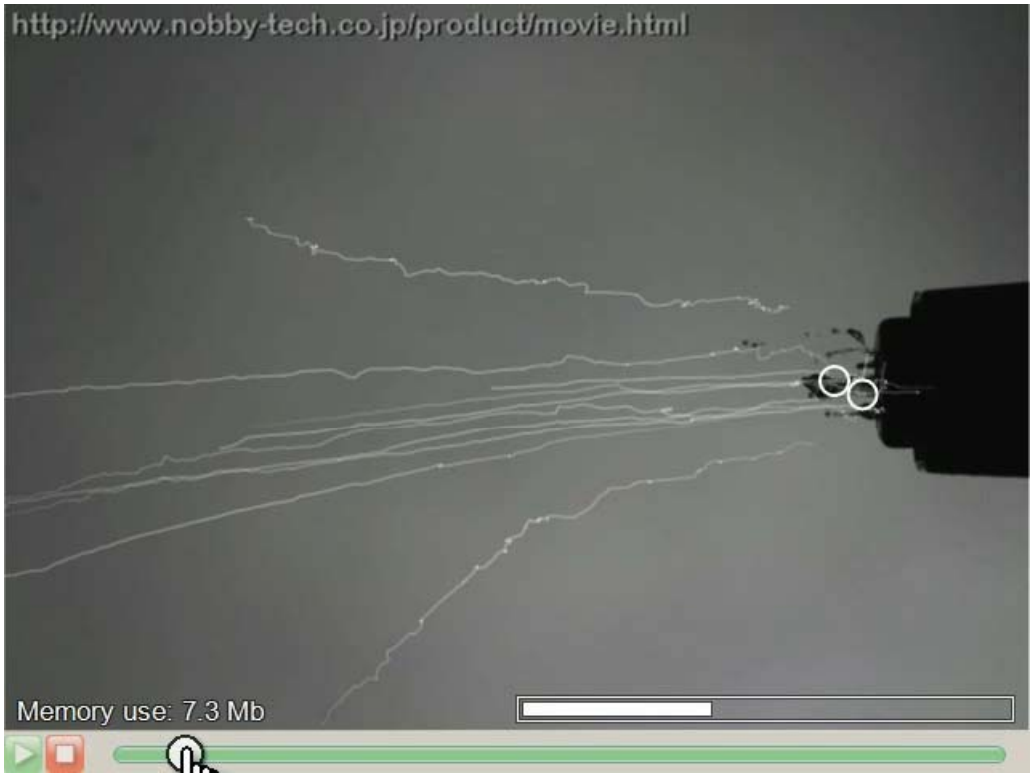
Extracting Motions from Videos

- Works especially well on:
  - High-quality videos
  - Large, continuous motions
  - Supports deformations



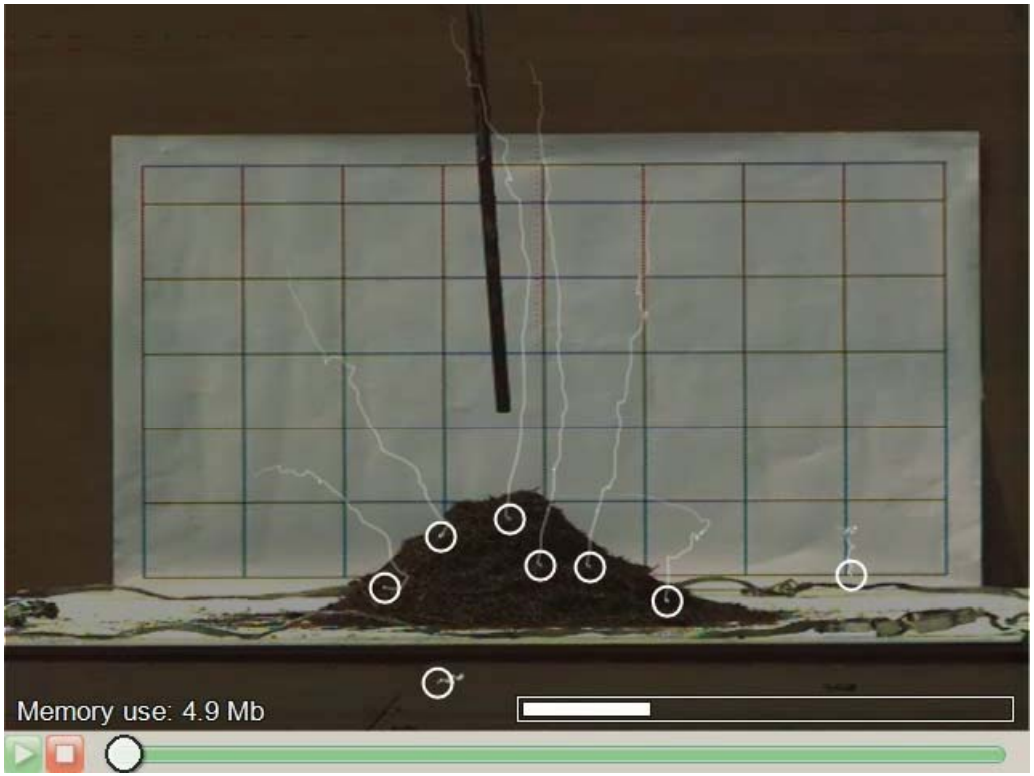
Extracting Motions from Videos

- Works especially well on:
  - High-quality videos
  - Large, continuous motions
  - Supports deformations



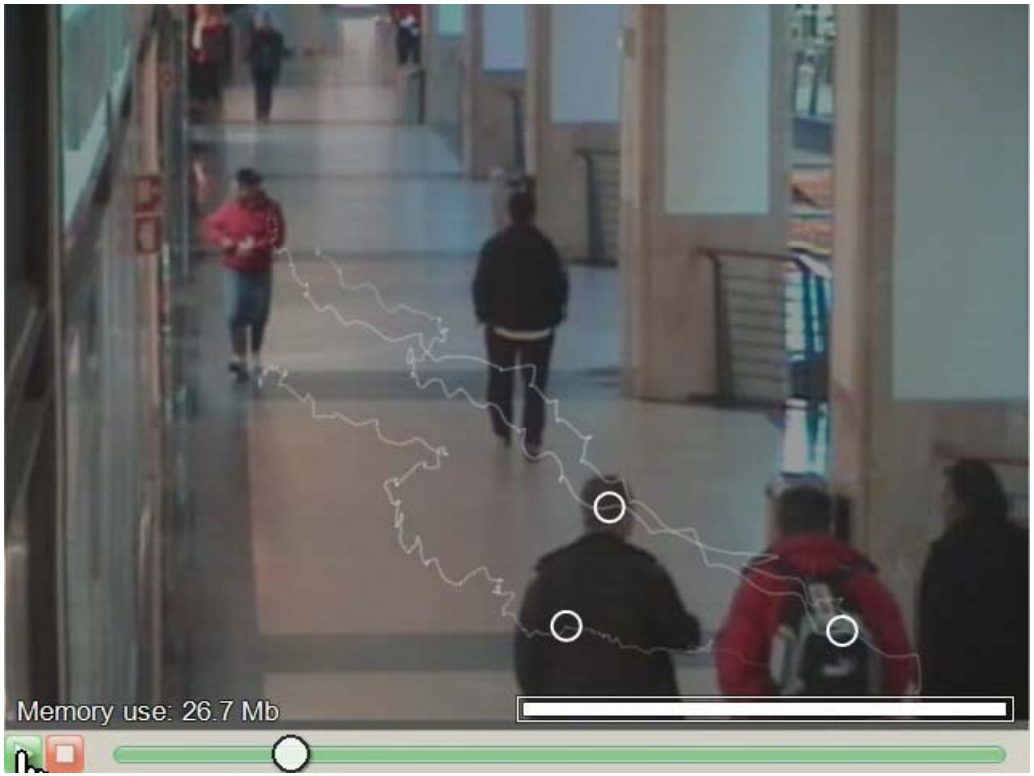
Extracting Motions from Videos

- Works especially well on:
  - High-quality videos
  - Large, continuous motions
  - Supports deformations



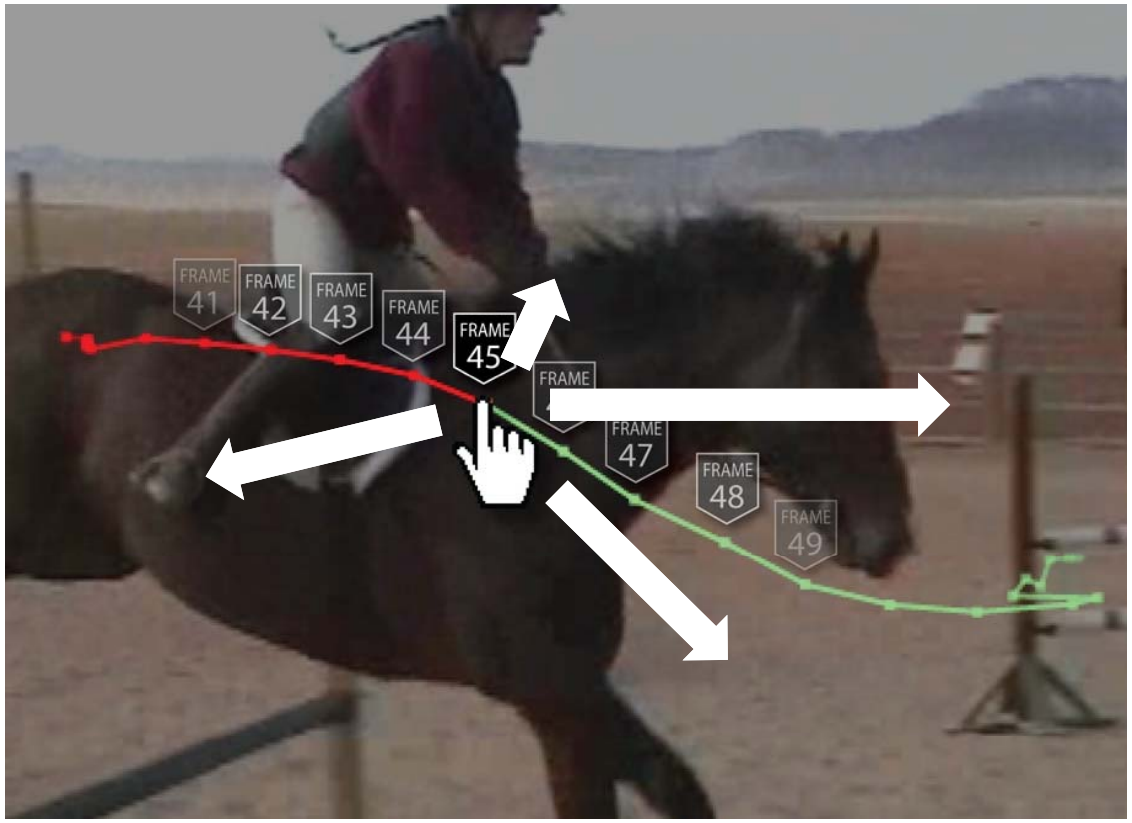
Extracting Motions from Videos
Limits

- Occlusion

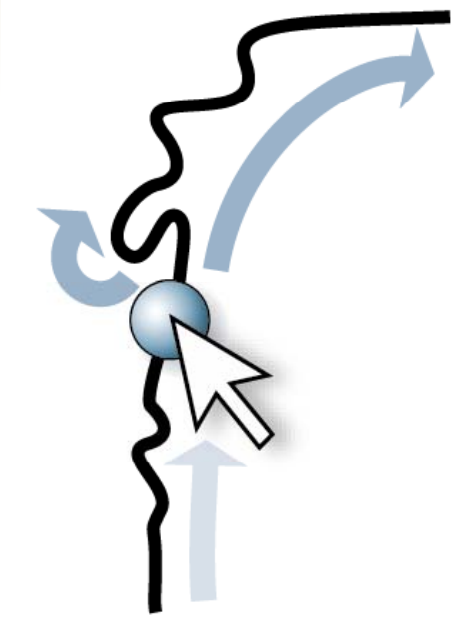


Extracting Motions from Videos

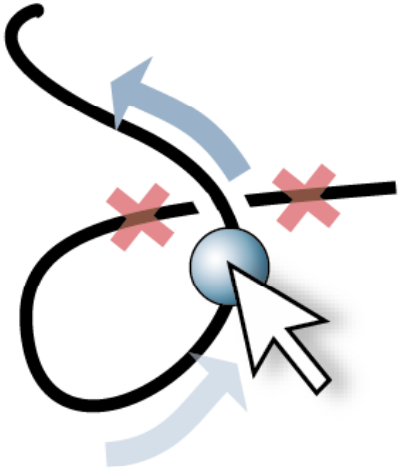
- **Inverse Computation:**
  - **Location on the curve -> Frame number**



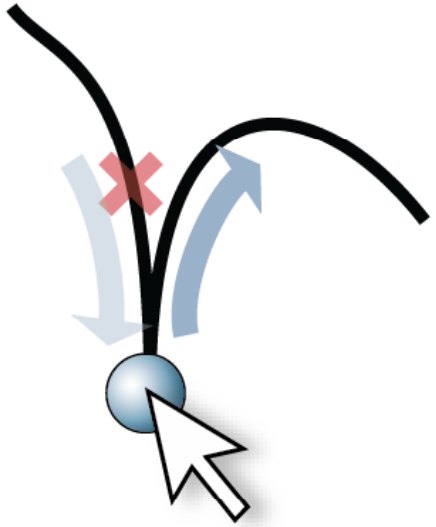
Curvilinear Dragging
Requirements



Multi-Scale Navigation



Arc-Length Continuity

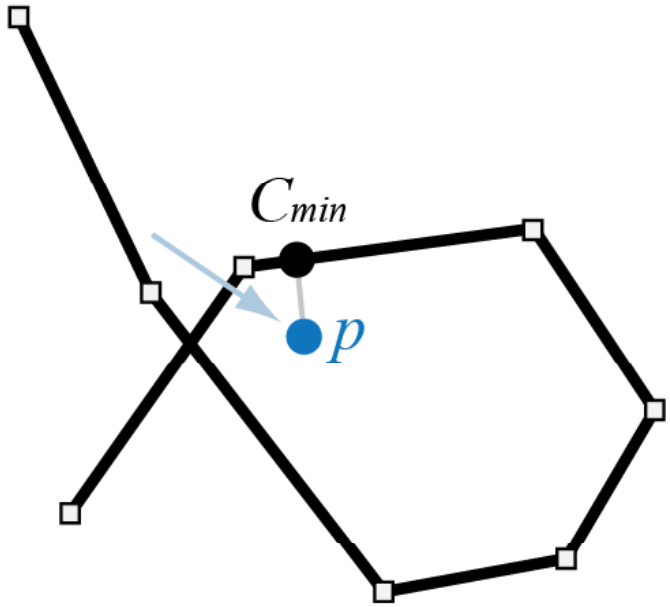


Directional Continuity

Curvilinear Dragging
Our Solution

- **The Closest Point Method**

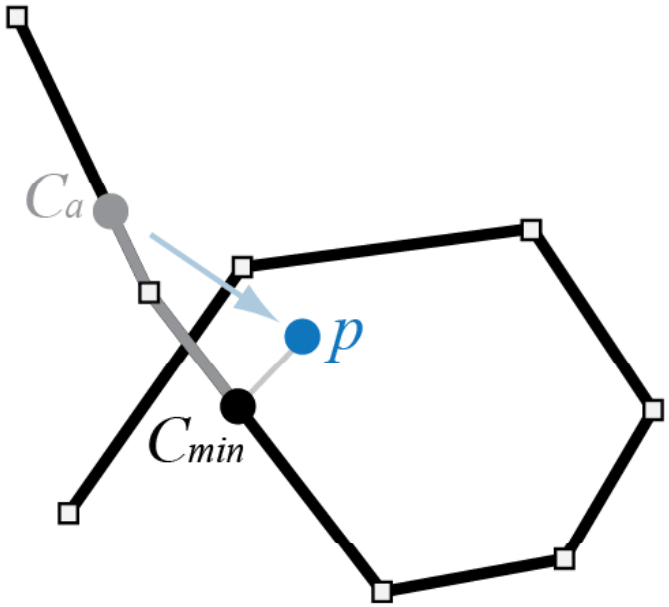
$$D = \sqrt{(p_x - C_x)^2 + (p_y - C_y)^2}$$



Curvilinear Dragging
Our Solution

- **The 3D Distance Method**

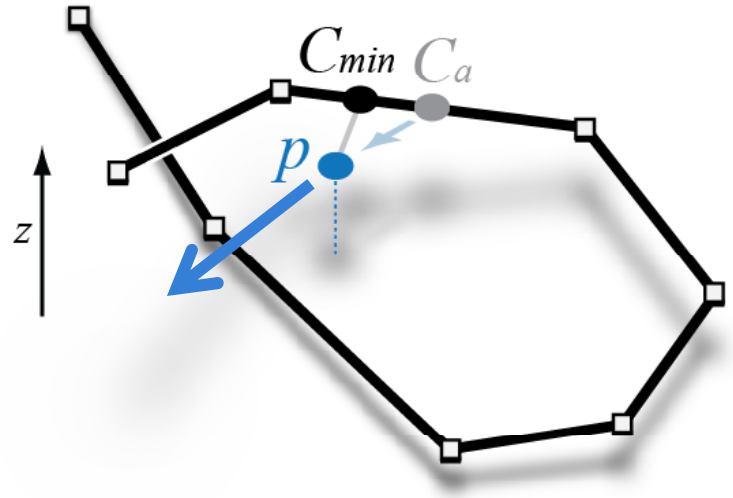
$$D = \sqrt{(p_x - C_x)^2 + (p_y - C_y)^2 + (k \cdot \overline{C_a C})^2}$$



Curvilinear Dragging
Our Solution

- **The 3D Distance Method**

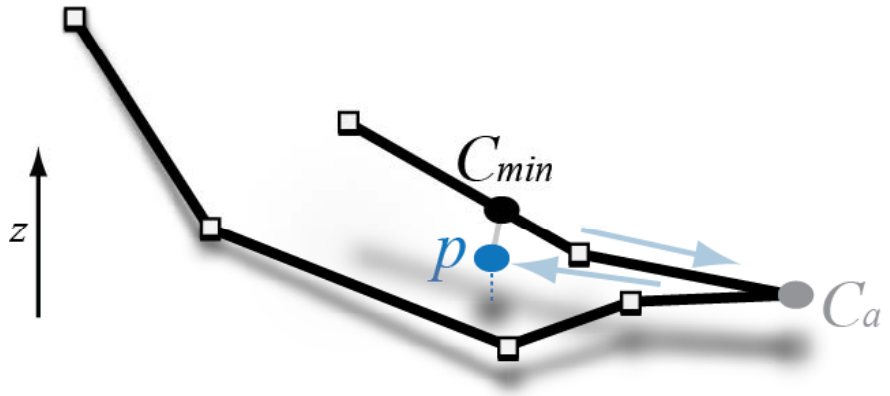
$$D = \sqrt{(p_x - C_x)^2 + (p_y - C_y)^2 + (k \cdot \overline{C_a C})^2}$$



Curvilinear Dragging
Our Solution

- **The 3D Distance Method**

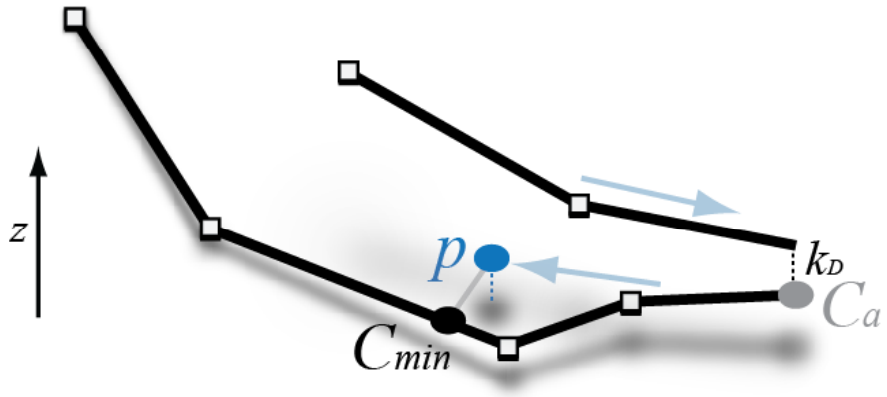
$$D = \sqrt{(p_x - C_x)^2 + (p_y - C_y)^2 + (k \cdot \overline{C_a C})^2}$$



Curvilinear Dragging
Our Solution

- **The 3D Distance Method**

$$D = \sqrt{(p_x - C_x)^2 + (p_y - C_y)^2 + (k \cdot \overline{C_a C})^2} + k_D$$



Curvilinear Dragging
Issues

- Extremely complex curves
  - Clip curves

- Sensitivity to noise
  - Low-pass filtering [Taubin 95]

- Direct Manipulation
- RFD
- Motion Extraction
- Curvilinear Dragging

- Video Player Design
- Related Work
- User Study

DimP

The Direct Manipulation Video Player

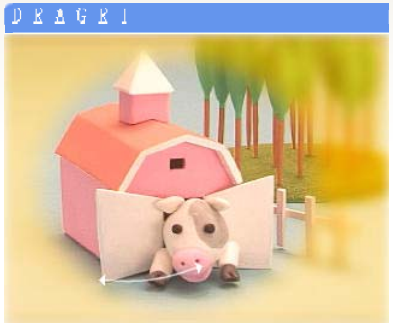
- Design / Demo!

Trajectory Visualization

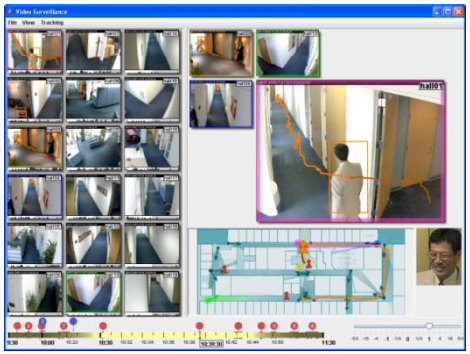
Background Stabilization

Dragging Video Content
Previous (and Current!) Work

- Similar systems
  - DRAGRI (2002)
  - Trailblazing
    - Kimber et al. ’07
  - DRAGON
    - Karrer et al.‘08



- Visual summaries
  - Schematic Storyboards
    - Goldman, et al. ‘06

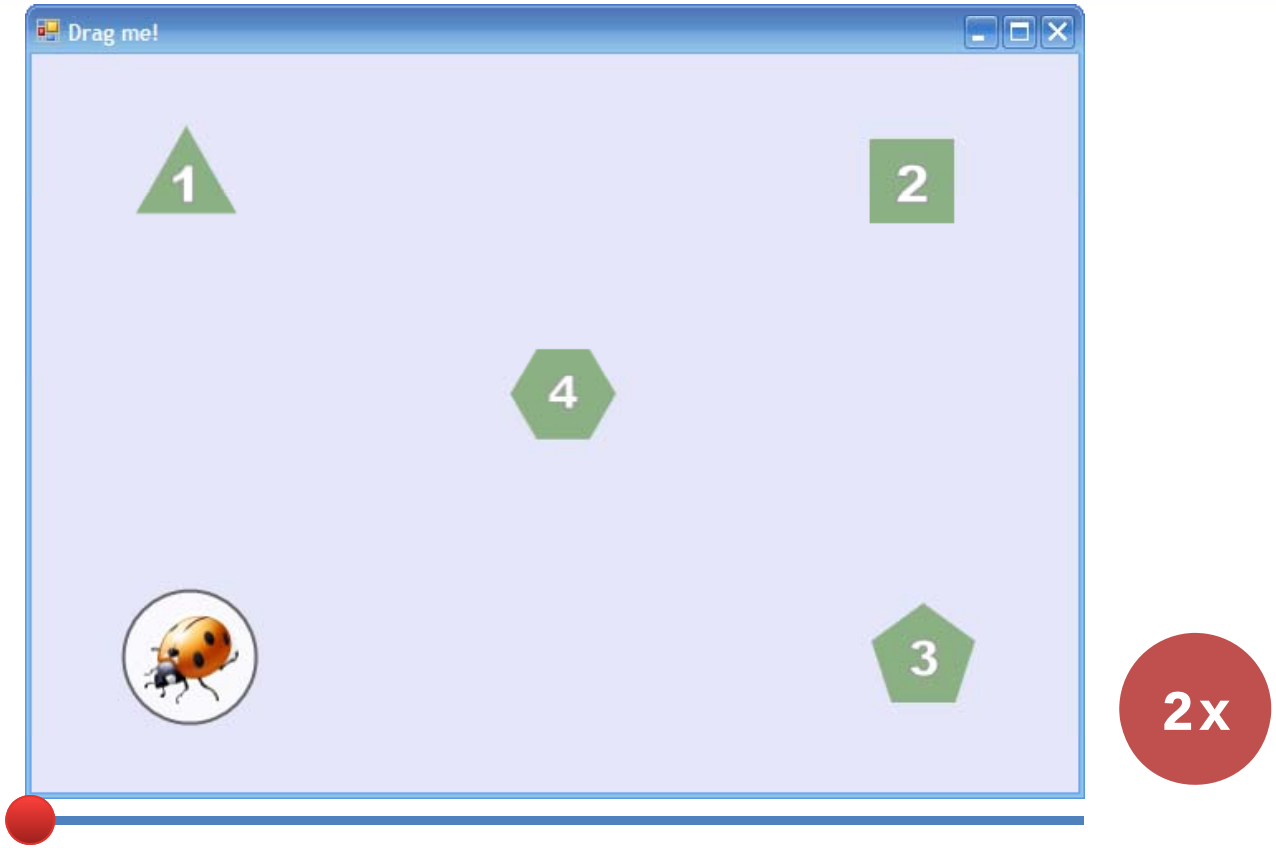


User Study: Questions

- Advantage in image-space tasks?
- Can users Relative Flow Drag?
- User's preference?

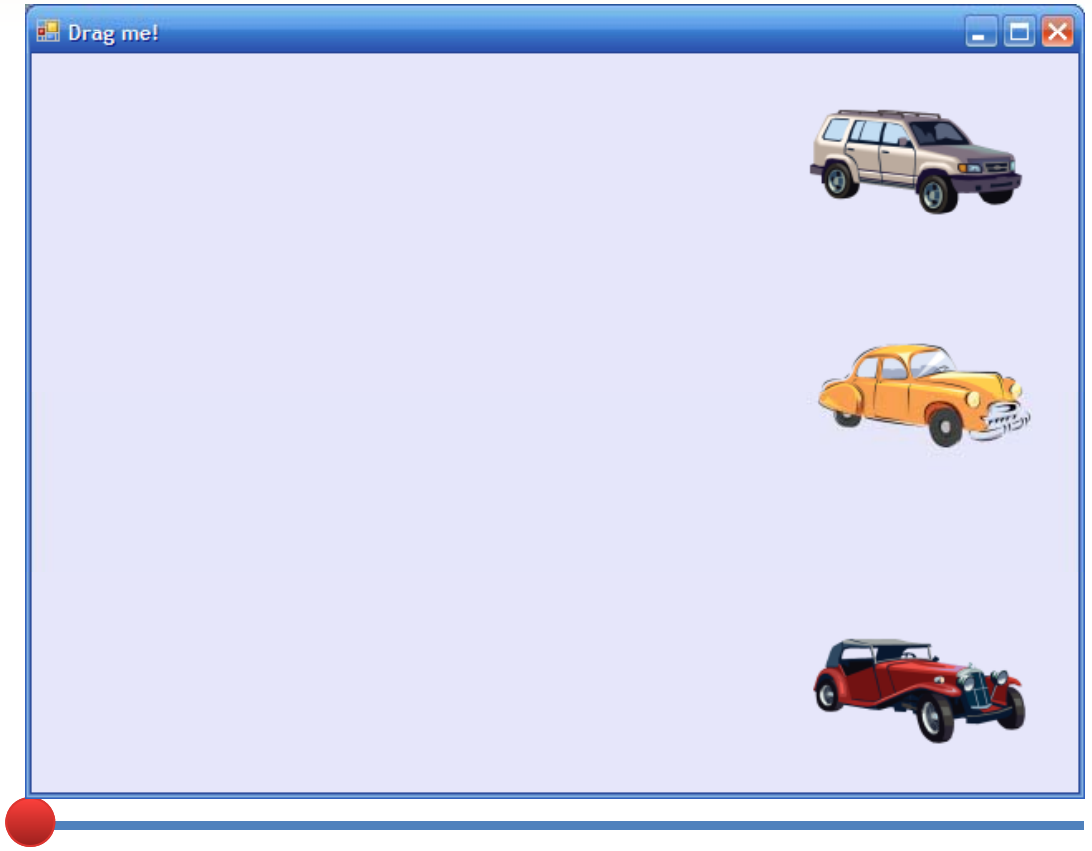
User Study: Ladybug Task

"Find the moment in the video when the ladybug passes over marker X"



User Study: Car Task

“Find the moment in the video when car X starts moving ”

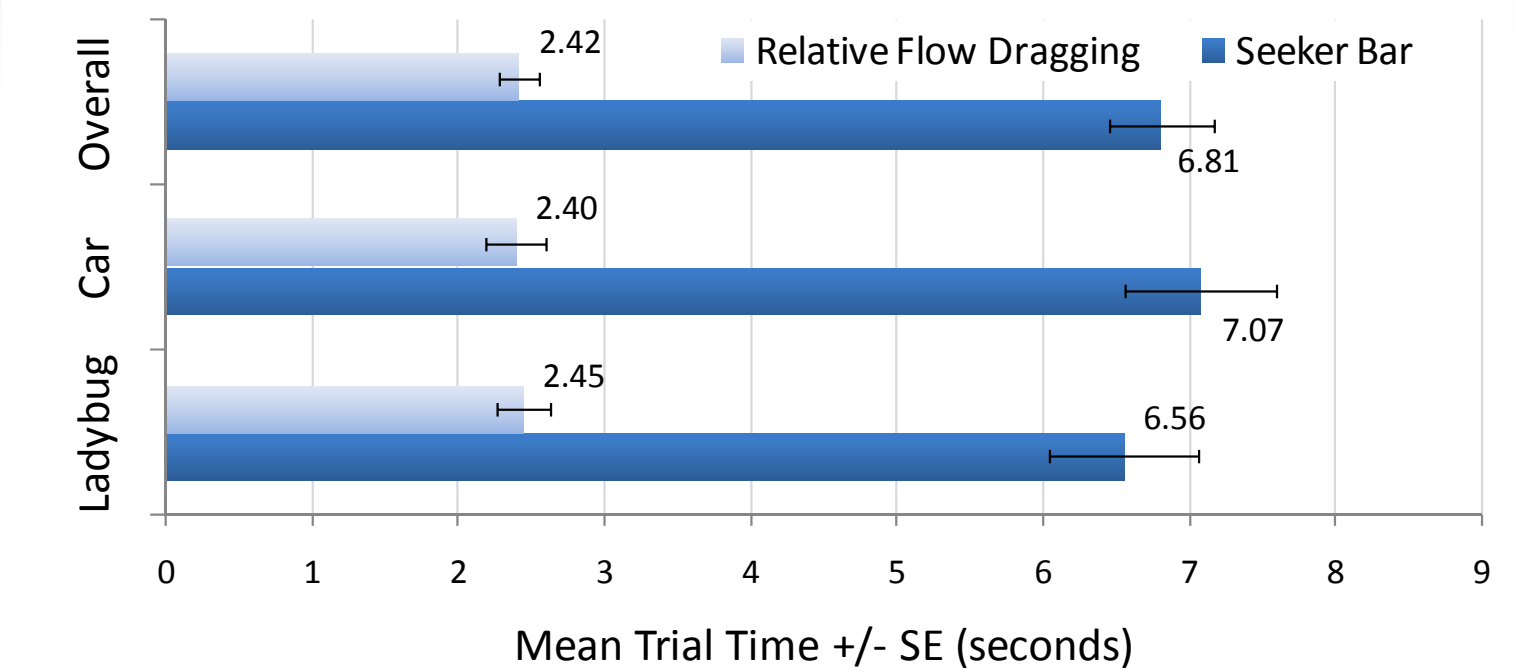


3x

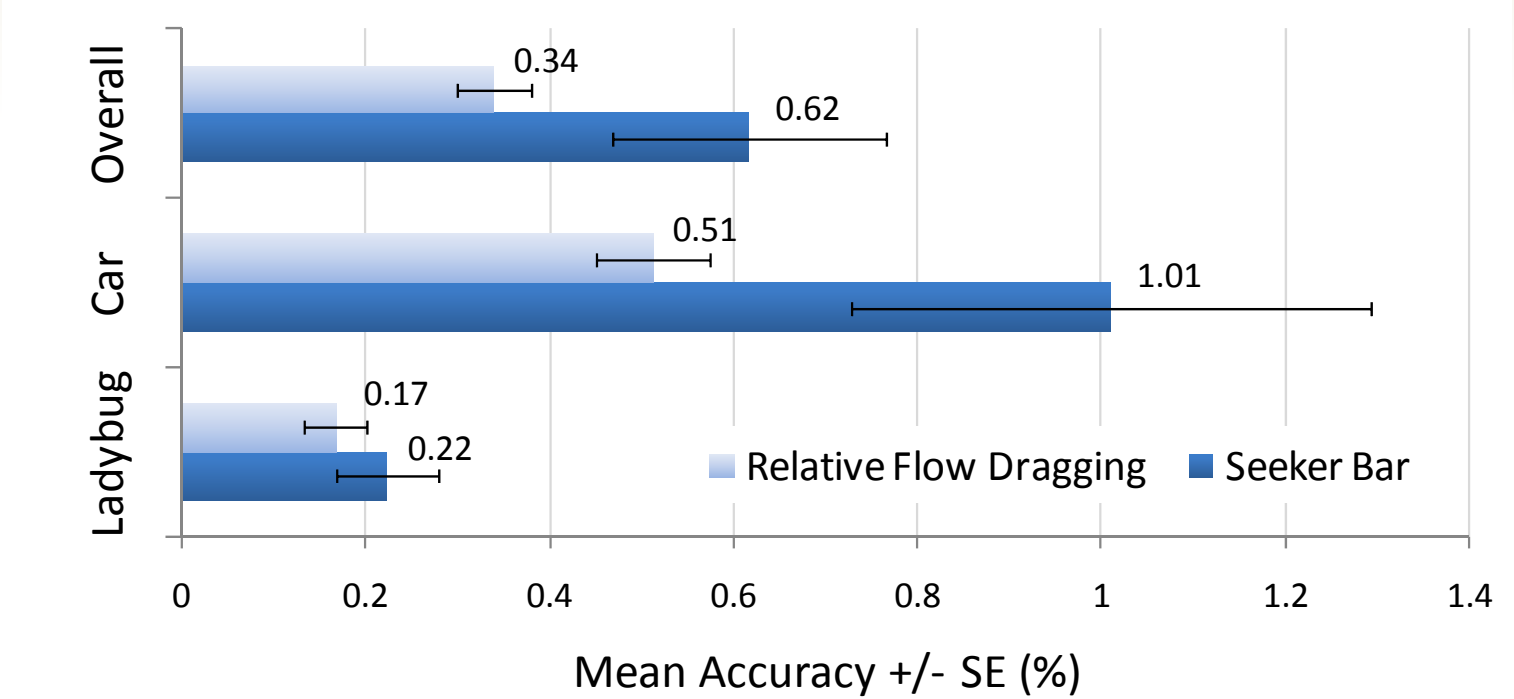
User Study: Design

- Within Subjects
  - 16 Participants
  - 2 Techniques (RFD, Seeker)
  - 2 Tasks (Ladybug, Car)
  - 3 Targets
- Measured Time, Accuracy
- Solicited User's Preference

User Study: Time

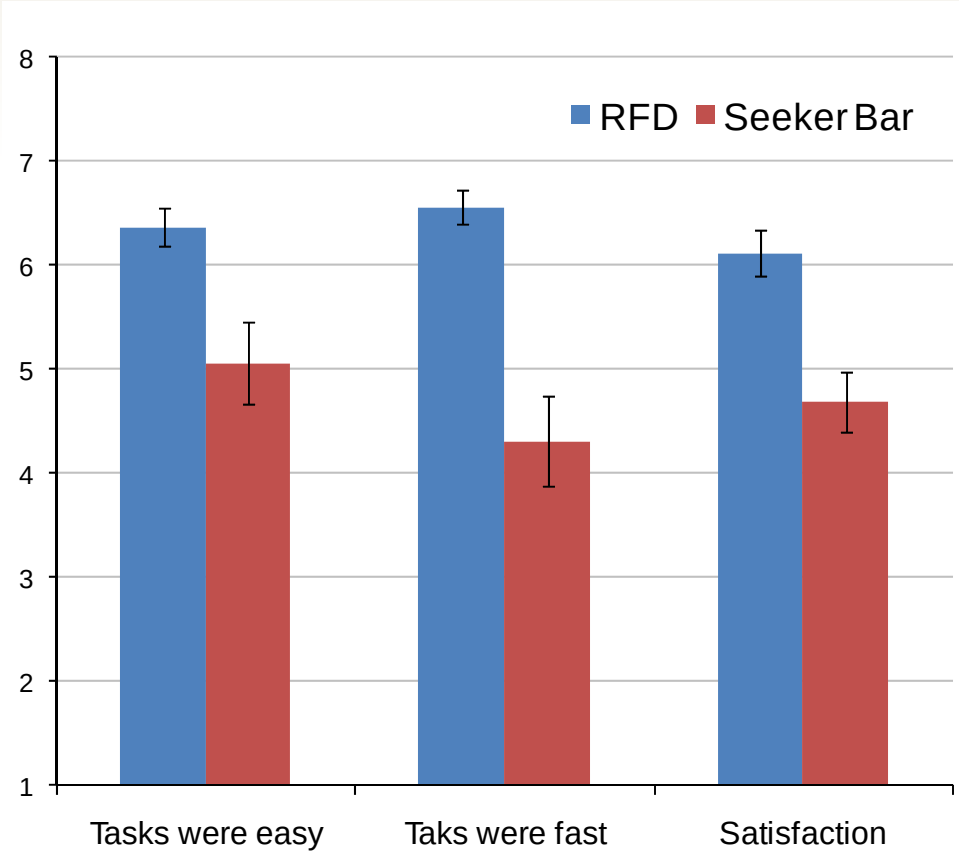


User Study: Accuracy



User Study: Preferences

(using a 7-point Likert Scale)



User Study: Take Aways

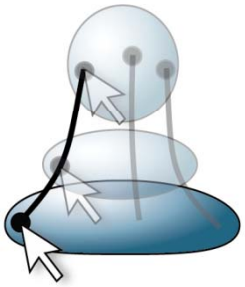
- People (Relative Flow) Drag well
- For space-centric tasks, RFD is
  - Faster
  - As precise or more than Seeker Bar
  - As preferred or more than Seeker Bar

Conclusion
& Future Work

- A new way of browsing videos
- Advantages of direct manipulation:
  - Radically different user experience
  - In some cases, more efficient
- Complementary of current tools
- New classes of DM techniques:
  - Curvilinear dragging
  - Flow dragging
  - Relative dragging
- Can be applied to other applications

Conclusion
& Future Work

- **Flow Dragging in a 3D Animation Tool**
  (Autodesk Maya Extension by Karan Singh)



Animation Timeline →

Thank You!

Thank You!

Video Browsing By Direct Manipulation

Pierre Dragicevic
Gonzalo Ramos
Jacobo Bibliowitcz
Derek Nowrouzezahrai
Ravin Balakrishnan
Karan Singh

dgp, University of Toronto
INRIA, France

www.aviz.fr/dimp