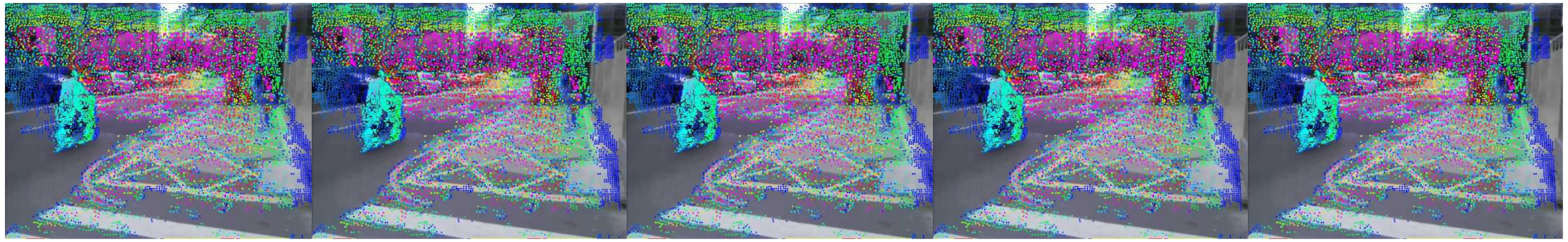
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 17: Optical Flow

Menu for Today

Topics:

— **Optical Flow**

Readings:

— **Today's** Lecture: Szeliski 15.1, 15.2

Reminders:

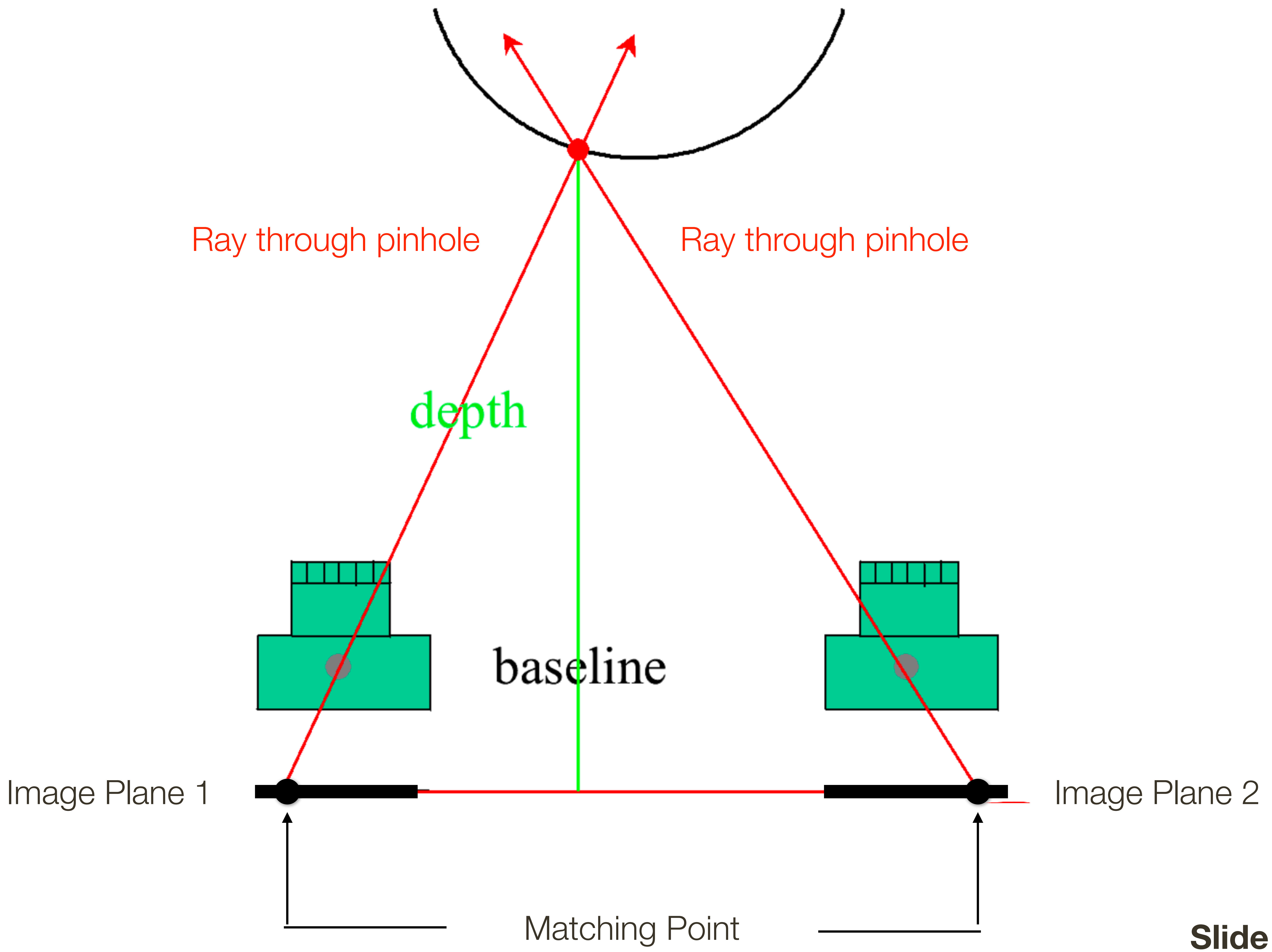
— Next week is a **Reading Break**

— **Assignment 4**: RANSAC and Panoramas due **November 8th**

— **Assignment 5:** Stereo and Optical Flow out **November 8th**

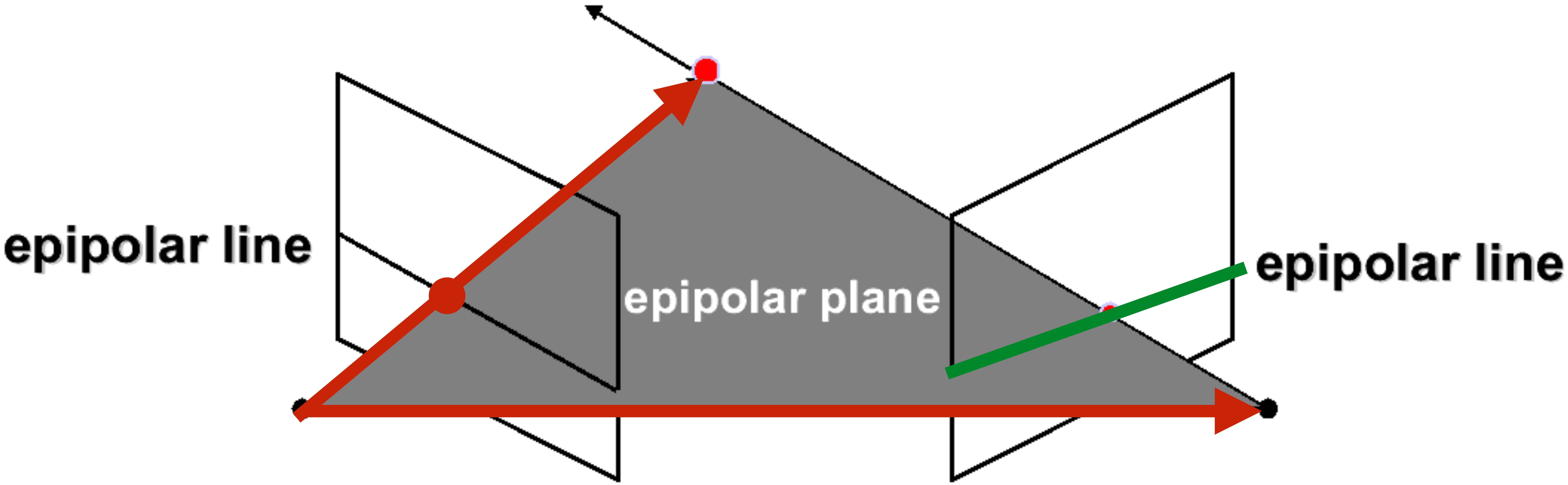
— **Quiz 4 and on** (after the Break)

Stereo Vision



Slide credit: Trevor Darrell

The **Epipolar** Constraint



Matching points lie along corresponding epipolar lines

Reduces correspondence problem to 1D search along conjugate epipolar lines

Greatly reduces cost and ambiguity of matching

Slide credit: Steve Seitz

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

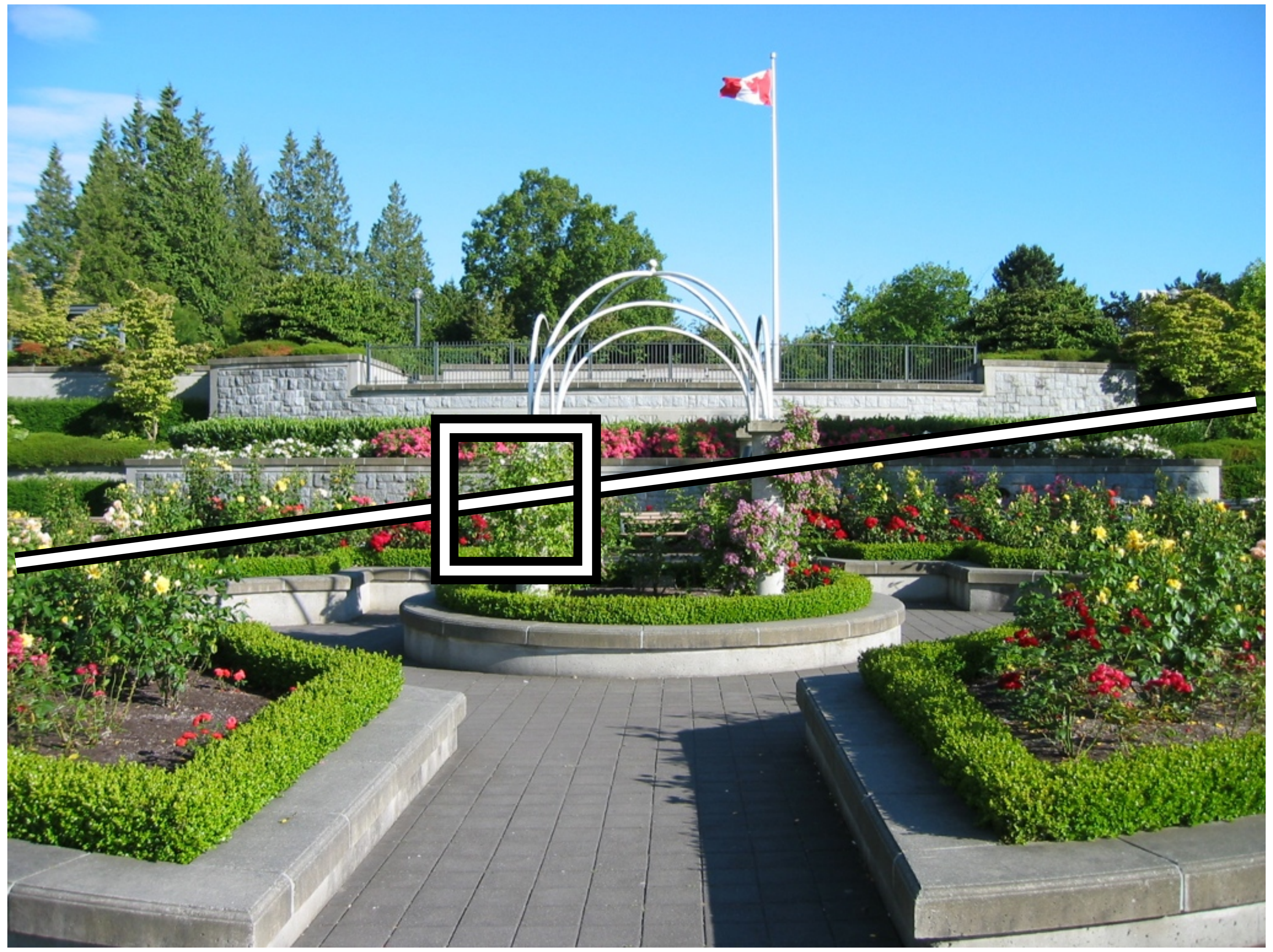
2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line



(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

2-view **Stereo**

Search over matches constrained to (epipolar) line

(reduces to 1d search)

Simplest Case: **Rectified** Images

Image planes of cameras are **parallel**

Focal **points** are at same height

Focal **lengths** same

Then, **epipolar lines** fall along the **horizontal scan lines** of the images

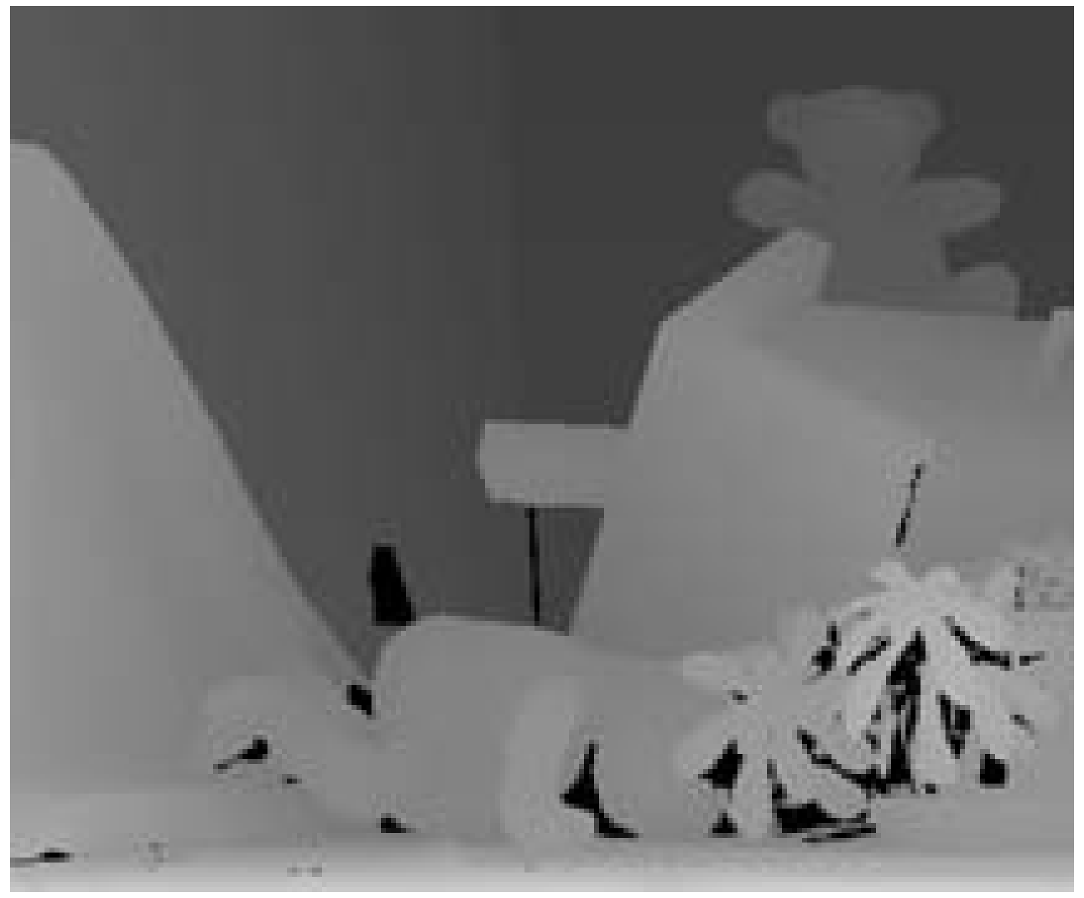
We assume images have been **rectified** so that epipolar lines correspond to scan lines

- Simplifies algorithms
- Improves efficiency

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal



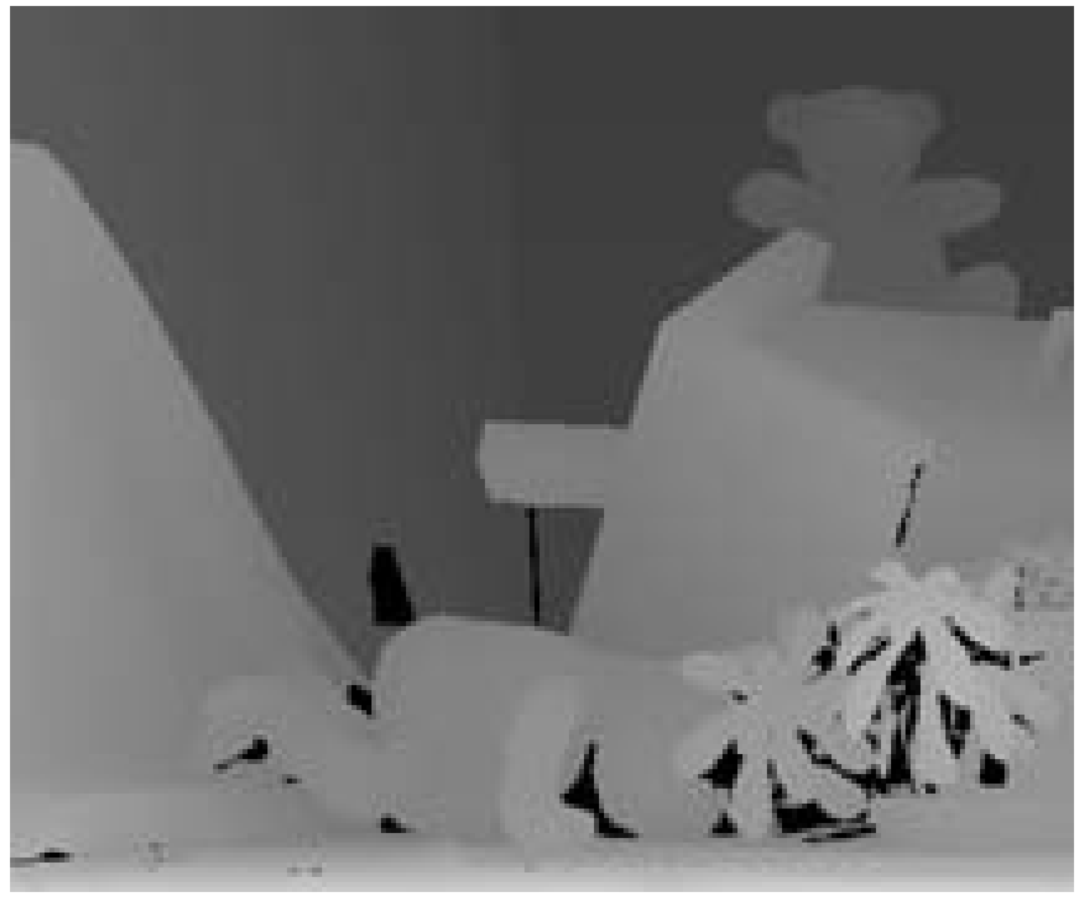


— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

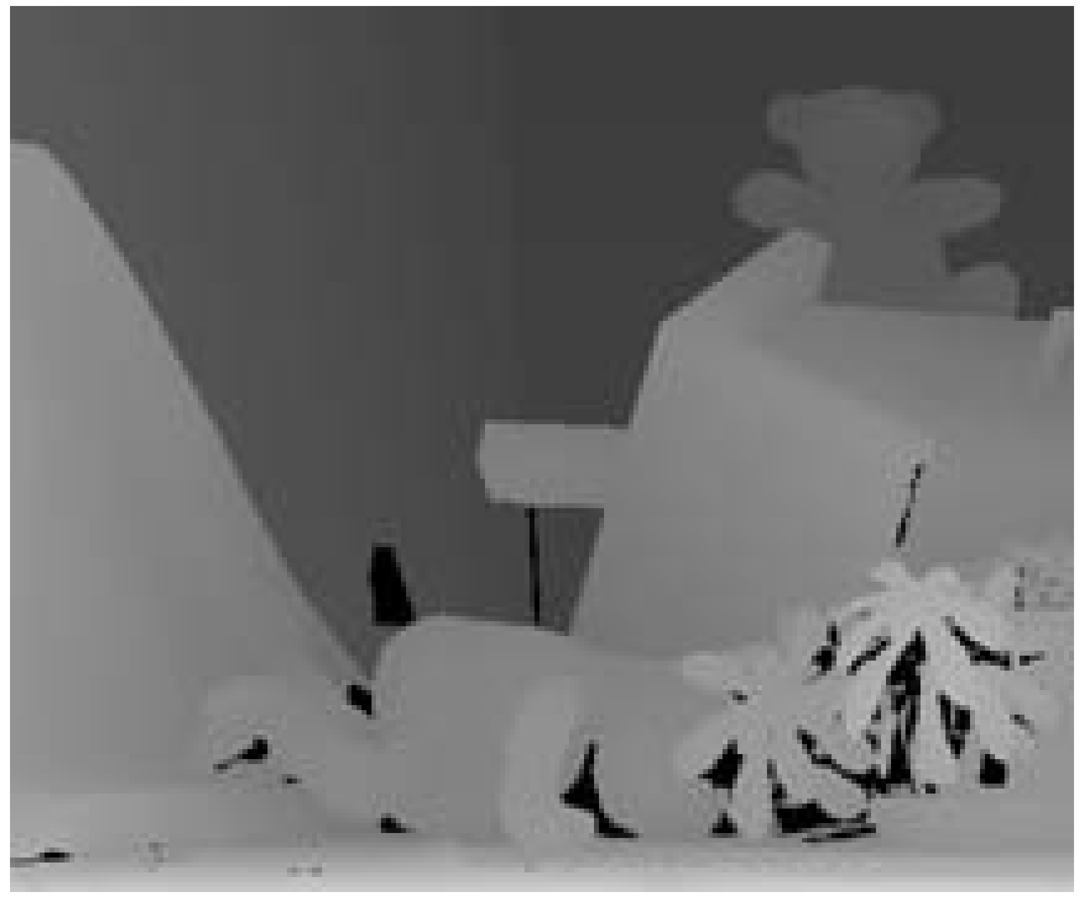
— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal





— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

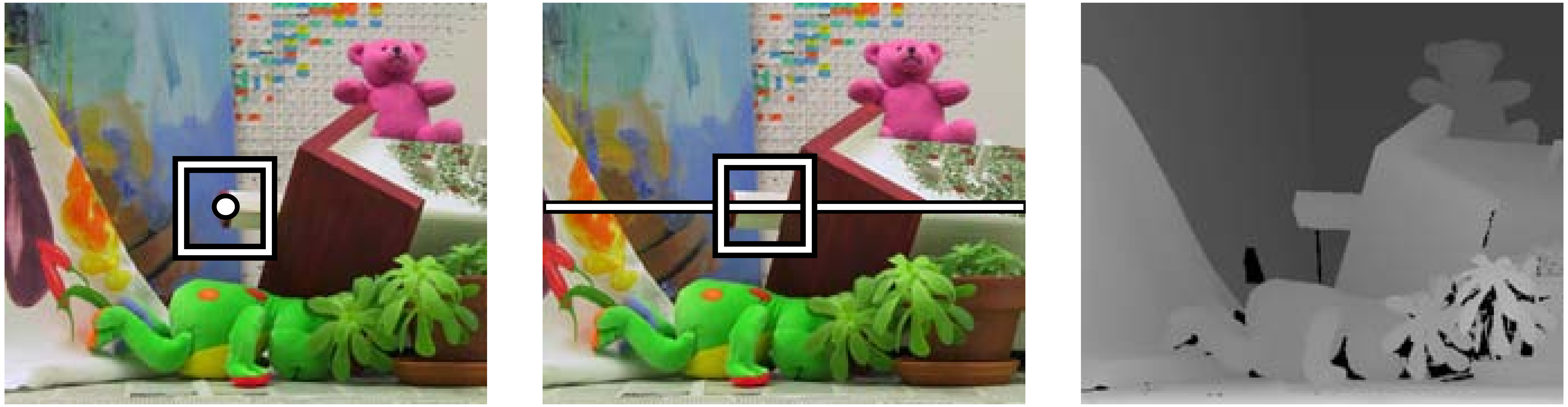
— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal



— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

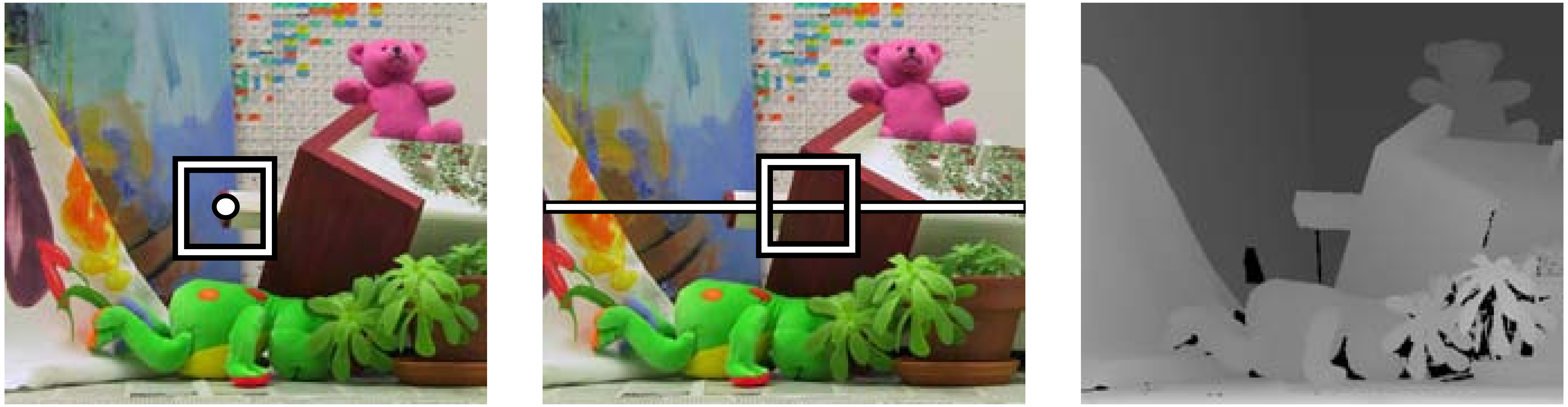
— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal

— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

Stereo Matching in Rectified Images

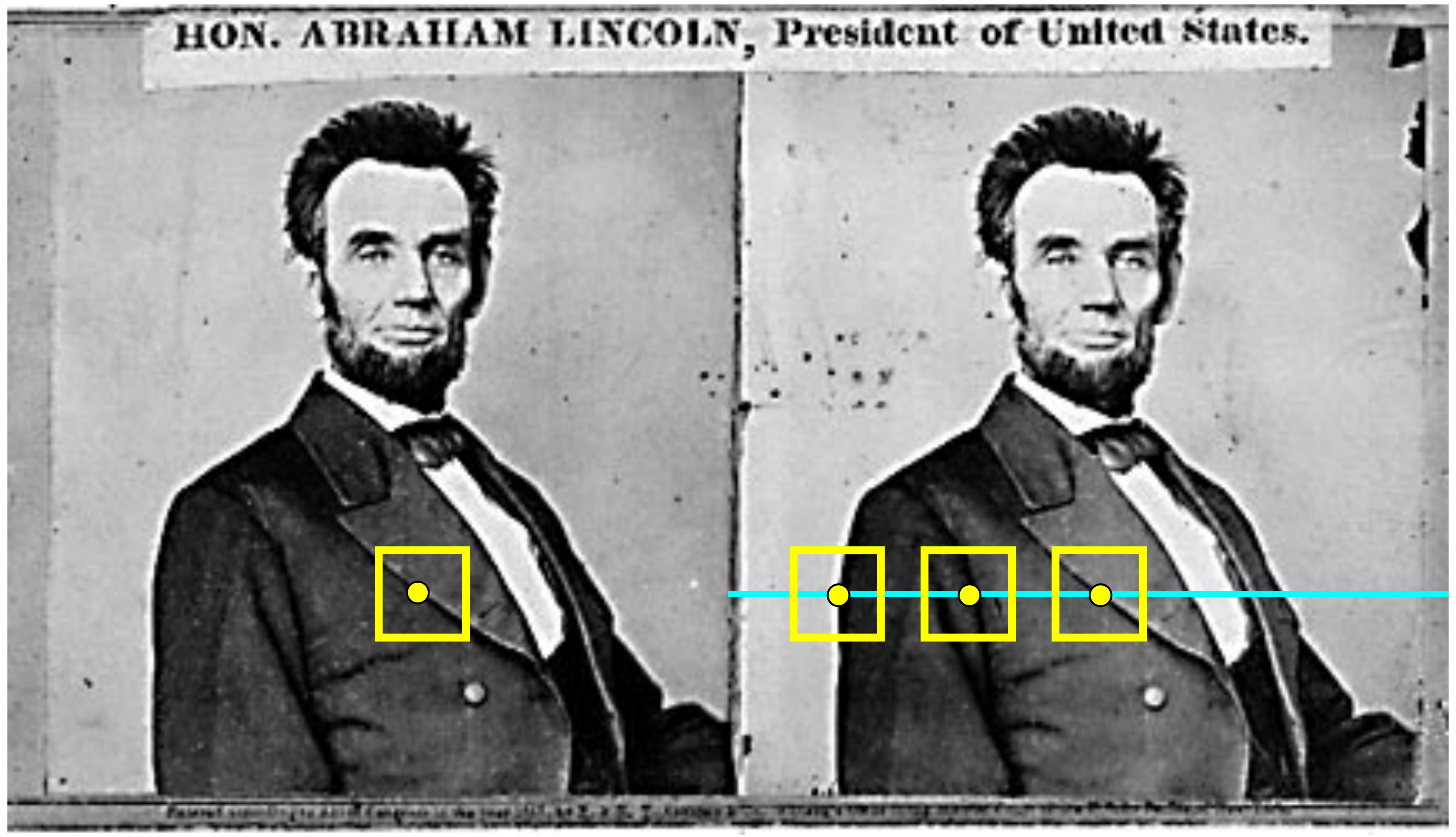
— In a standard stereo setup, where cameras are related by translation in the x direction, epipolar lines are horizontal



— Stereo algorithms search along scanlines for match

— Distance along the scanline (difference in x coordinate) for a corresponding feature is called **disparity**

(simple) **Stereo** Algorithm



1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity $Z = \frac{bf}{d}$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Block Matching: Sum of Squared (Pixel) Differences

$\mathbf{w}_L$ and $\mathbf{w}_R$ are corresponding $m \times m$ windows of pixels

Define the window function, $\mathbf{W}_m(x, y)$, by

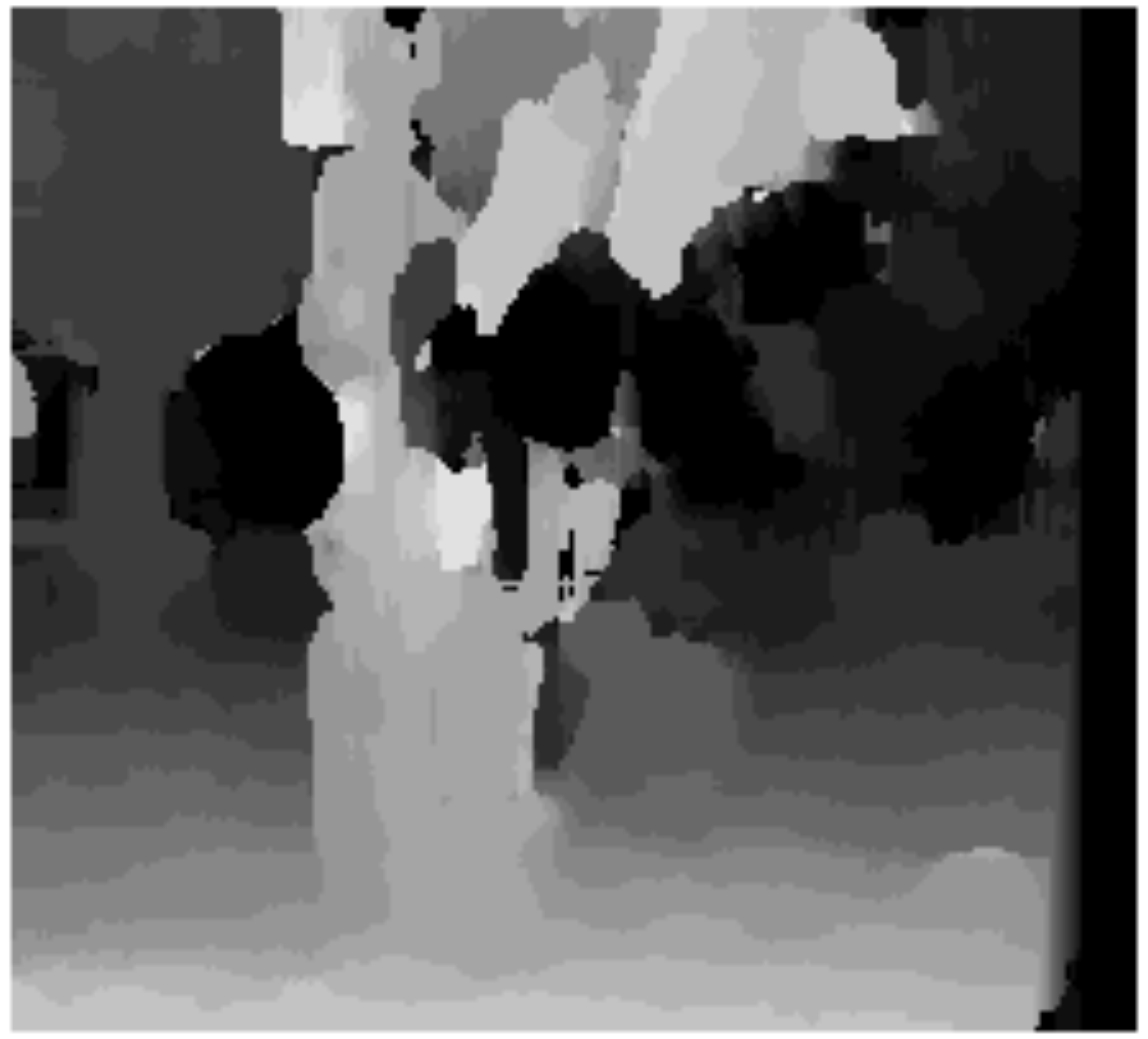
$$\mathbf{W}_m(x,y) = \left\{(u,v) \mid x - \frac{m}{2} \leq u \leq x + \frac{m}{2}, y - \frac{m}{2} \leq v \leq y + \frac{m}{2}\right\}$$

SSD measures intensity difference as a function of disparity:

$$C_R(x,y,d) = \sum_{(u,v)\in\mathbf{W}_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2$$

Effect of **Window Size**

W = 3



W = 20

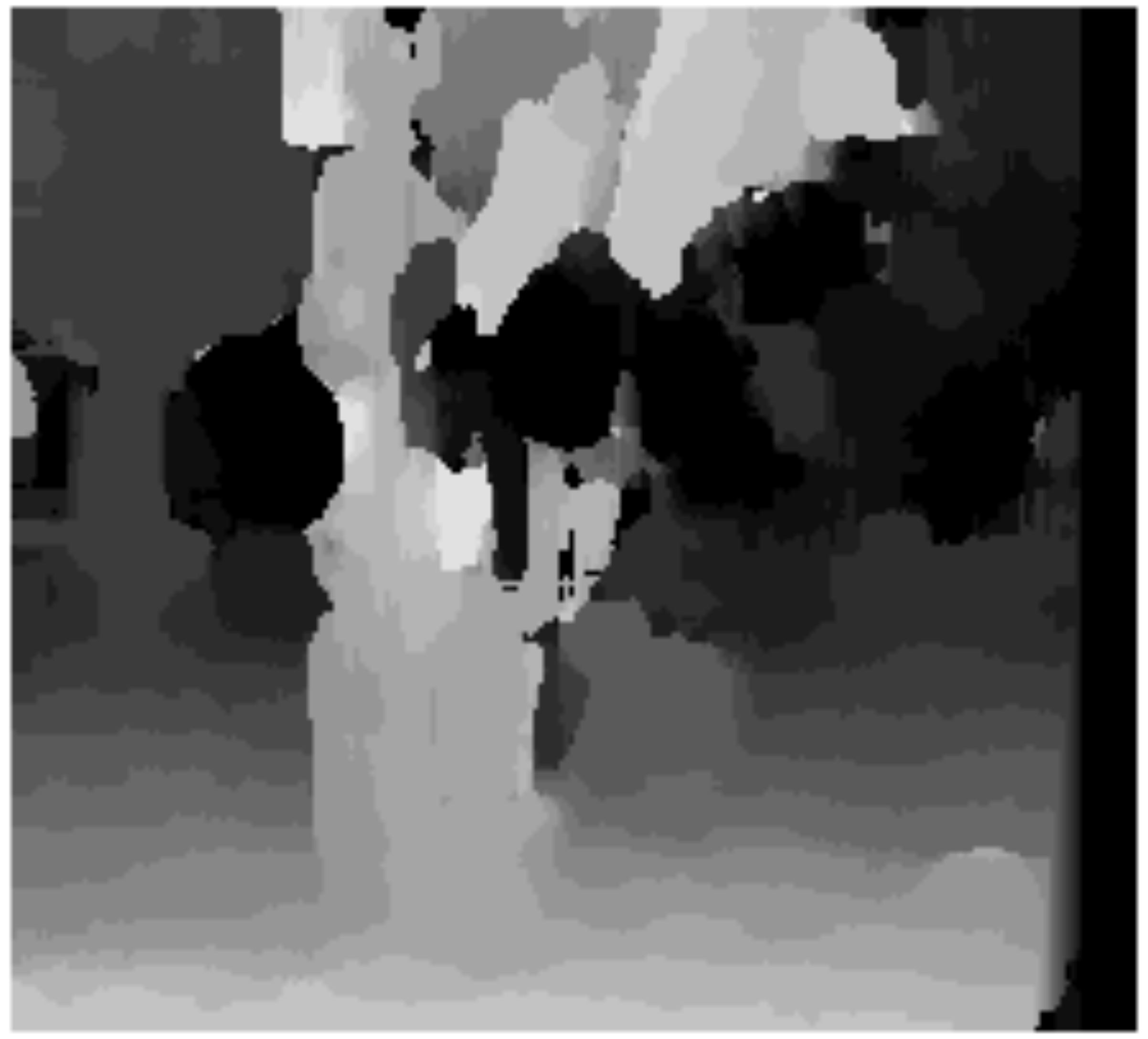
Smaller window
- \+ More detail
- \- More noise

Larger window
- \+ Smoother disparity maps
- \- Less detail
- \- Fails near boundaries

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Effect of **Window Size**

W = 3



W = 20

Note: Some approaches use an adaptive window size — try multiple sizes and select best match

Stereo Matching as **Energy Minimization**

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \lambda \underbrace{E_s(d)}_{\text{smoothness term}}$$

Want each pixel to find a good match in the other image

(block matching result)

Adjacent pixels should (usually) move about the same amount

(smoothness function)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Optical Flow

Problem:

Determine how objects (and/or the camera itself) move in the 3D world

Key Idea(s):

Images acquired as a (continuous) function of time provide additional constraint. Formulate motion analysis as finding (dense) point correspondences over time.

Dense vs **Sparse** Matching

Sparse: correspondence / depth estimated at discrete feature points, e.g., SIFT feature matches

Dense: correspondence / depth estimated at all locations, e.g., using stereo matching algorithms

Dense vs **Sparse** Matching

Optical Flow

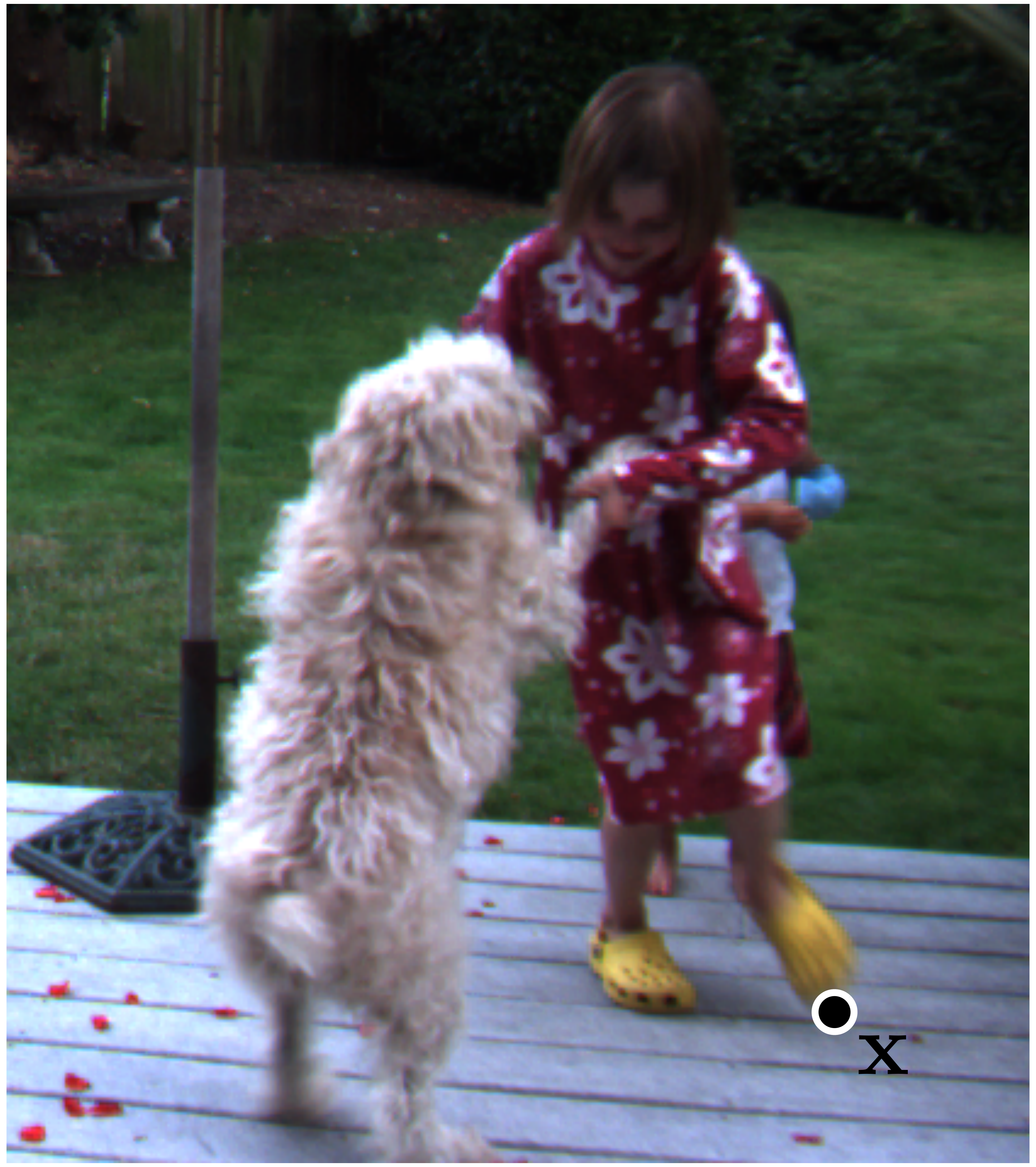
In this lecture we'll focus on

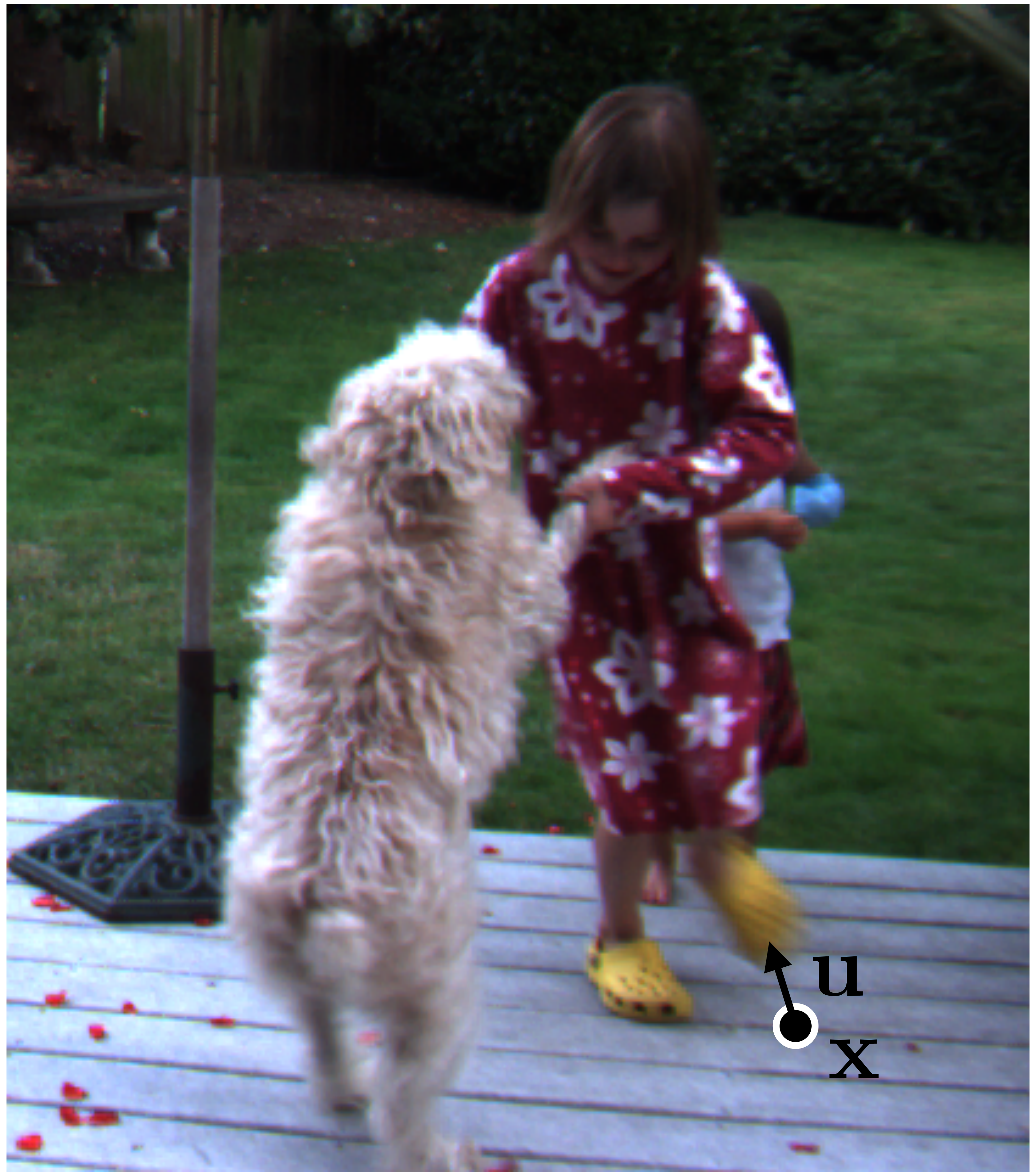
- **Dense flow** — compute correspondence / flow at every pixel
- **Short baselines** — assume small distances between frames, e.g., successive frames in a video

Wide baseline non-rigid matching algorithms do exist, but techniques are different (e.g., feature tracking)

[Z. Teed, Z. Deng, RAFT 2020]

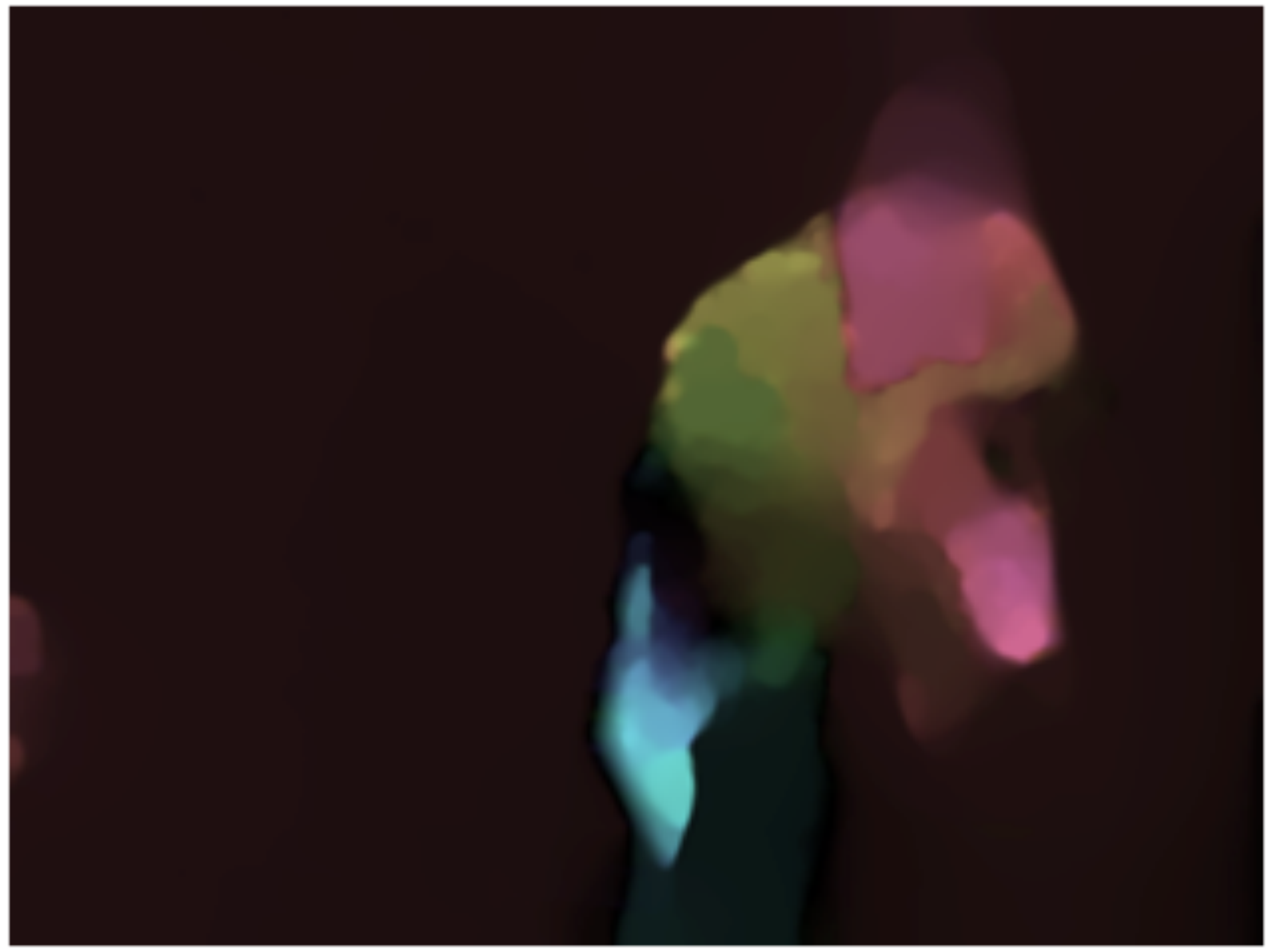
What is **Optical Flow**?

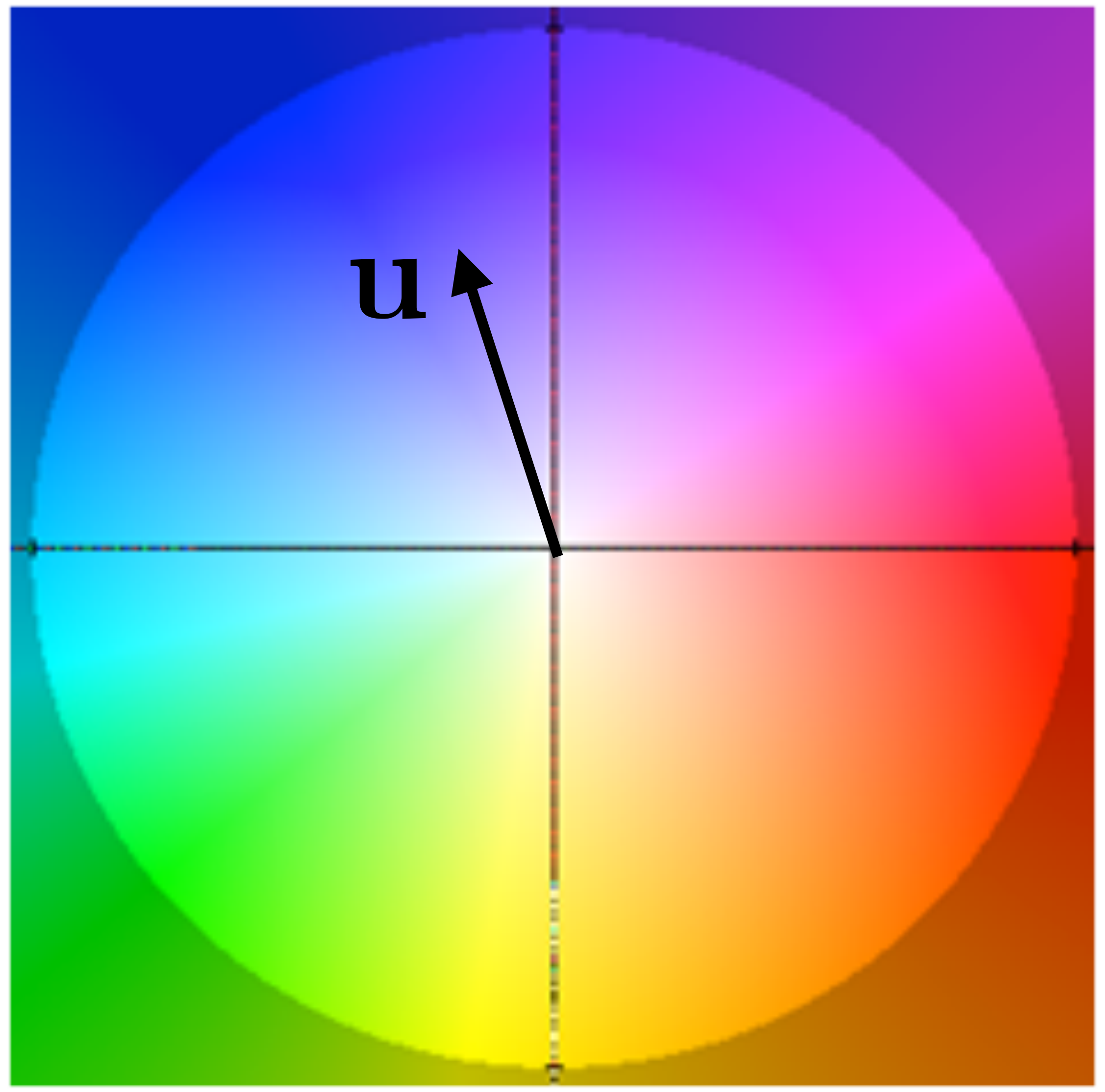




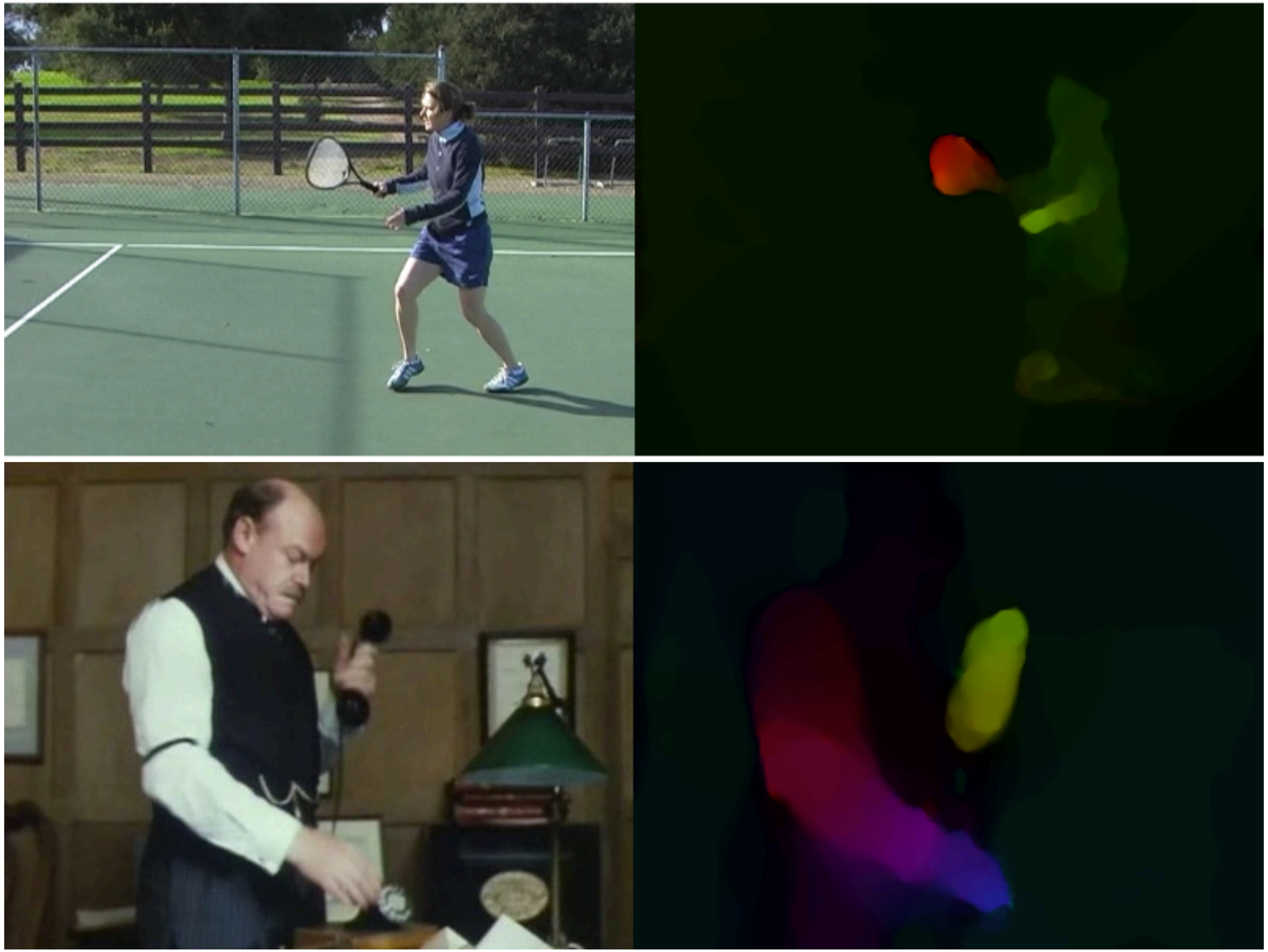
[vision.middlebury.edu/flow]

What is **Optical Flow**?





What is **Optical Flow**?



[Brox Malik 2011]

Optical Flow and 2D Motion

Optical flow is the apparent motion of brightness patterns in the image

Applications

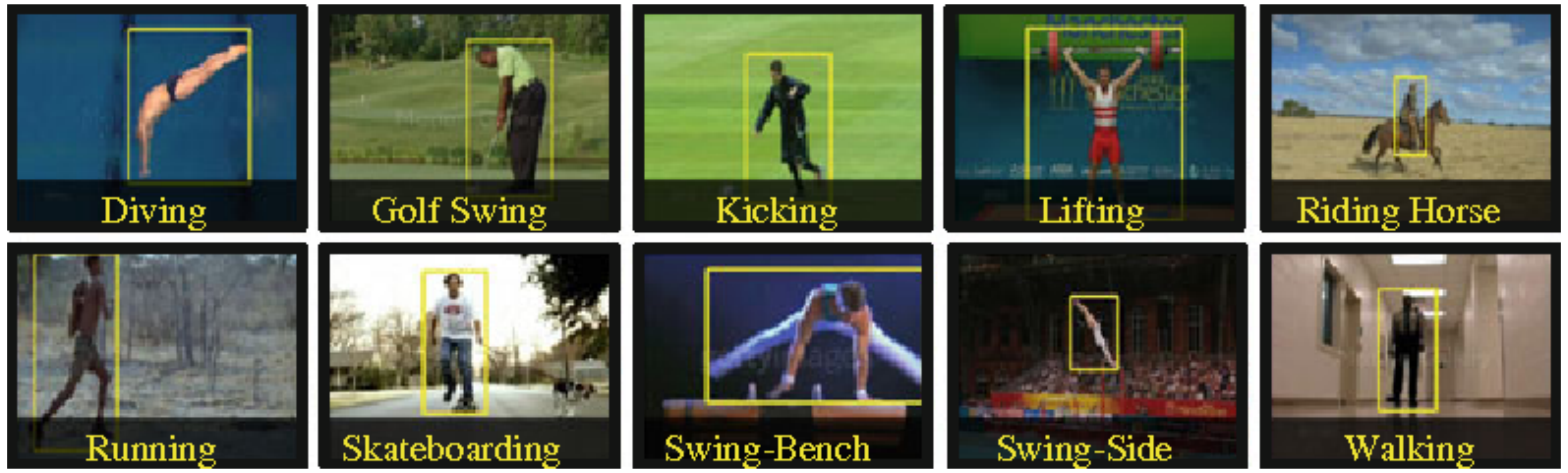
— image and video stabilization in digital cameras, camcorders

— motion-compensated video compression schemes such as MPEG

— image registration for medical imaging, remote sensing

— action recognition

— motion segmentation



Optical Flow and 2D Motion

Motion is geometric

Optical flow is radiometric

Usually we assume that optical flow and 2-D motion coincide ... but this is not always the case!

Optical Flow and 2D Motion

Optical flow but **no motion** . . .

Optical Flow and 2D Motion

Optical flow but **no motion** . . .

. . . moving light source(s), lights going on/off, inter-reflection, shadows

Optical Flow and 2D Motion

Optical flow but **no motion** . . .

. . . moving light source(s), lights going on/off, inter-reflection, shadows

Motion but **no optical flow** . . .

Optical Flow and 2D Motion

Optical flow but **no motion** . . .

. . . moving light source(s), lights going on/off, inter-reflection, shadows

Motion but **no optical flow** . . .

. . . spinning sphere.

Optical Flow and 2D Motion

Here's a video example of a very skilled Japanese contact juggler working with a clear acrylic ball

Source: http://youtu.be/CtztrcGkCBw?t=1m20s

A key element to the illusion is motion without corresponding optical flow

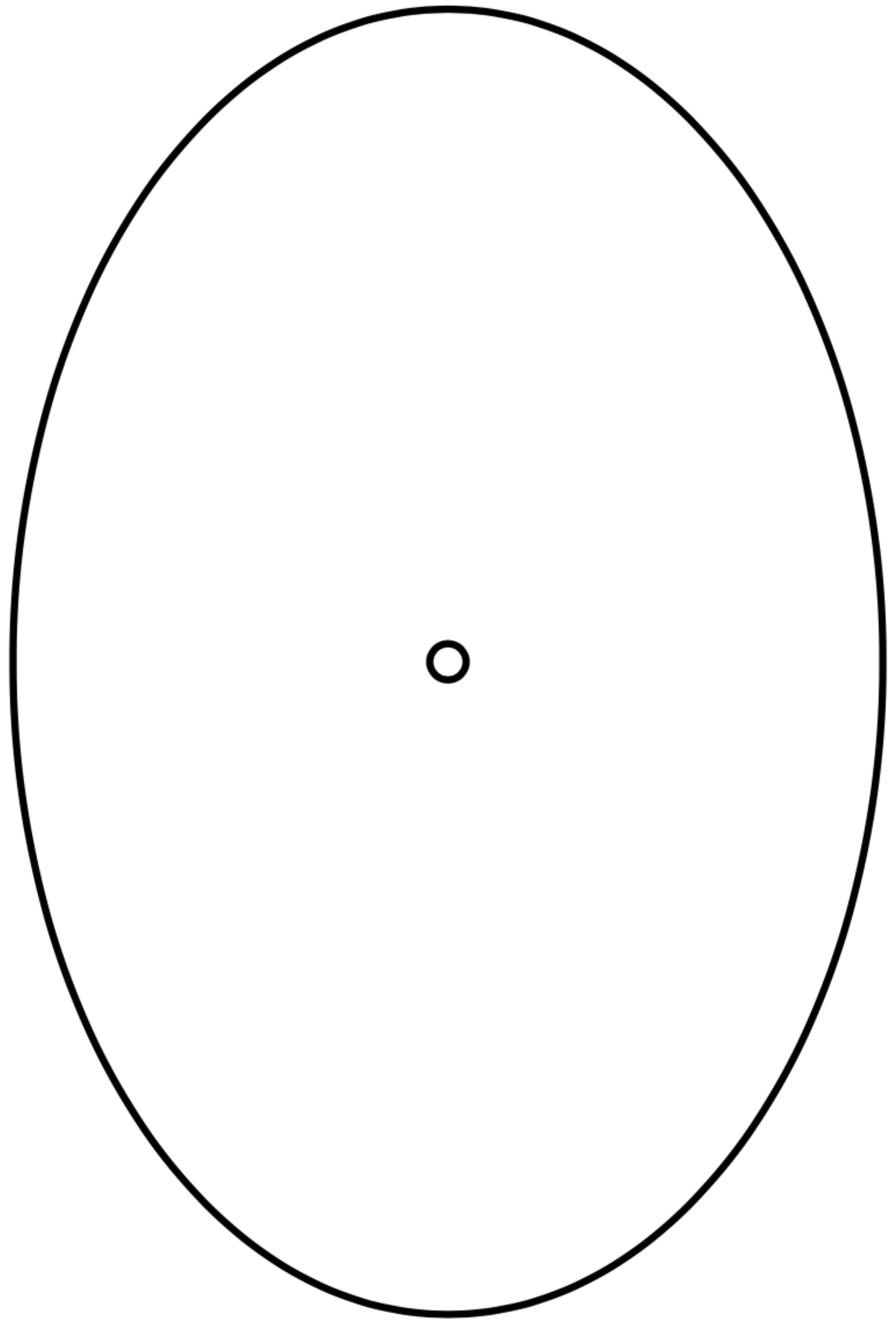
Optical Flow and 2D Motion

Here’s a video example of a very skilled Japanese contact juggler working with a clear acrylic ball

Source: http://youtu.be/CtztrcGkCBw?t=1m20s

A key element to the illusion is motion without corresponding optical flow

Example 1: Rotating Ellipse



Example 1: Three “Percepts”

1. **Veridical**:

— a 2-D rigid, flat, rotating ellipse

2. **Amoeboid**:

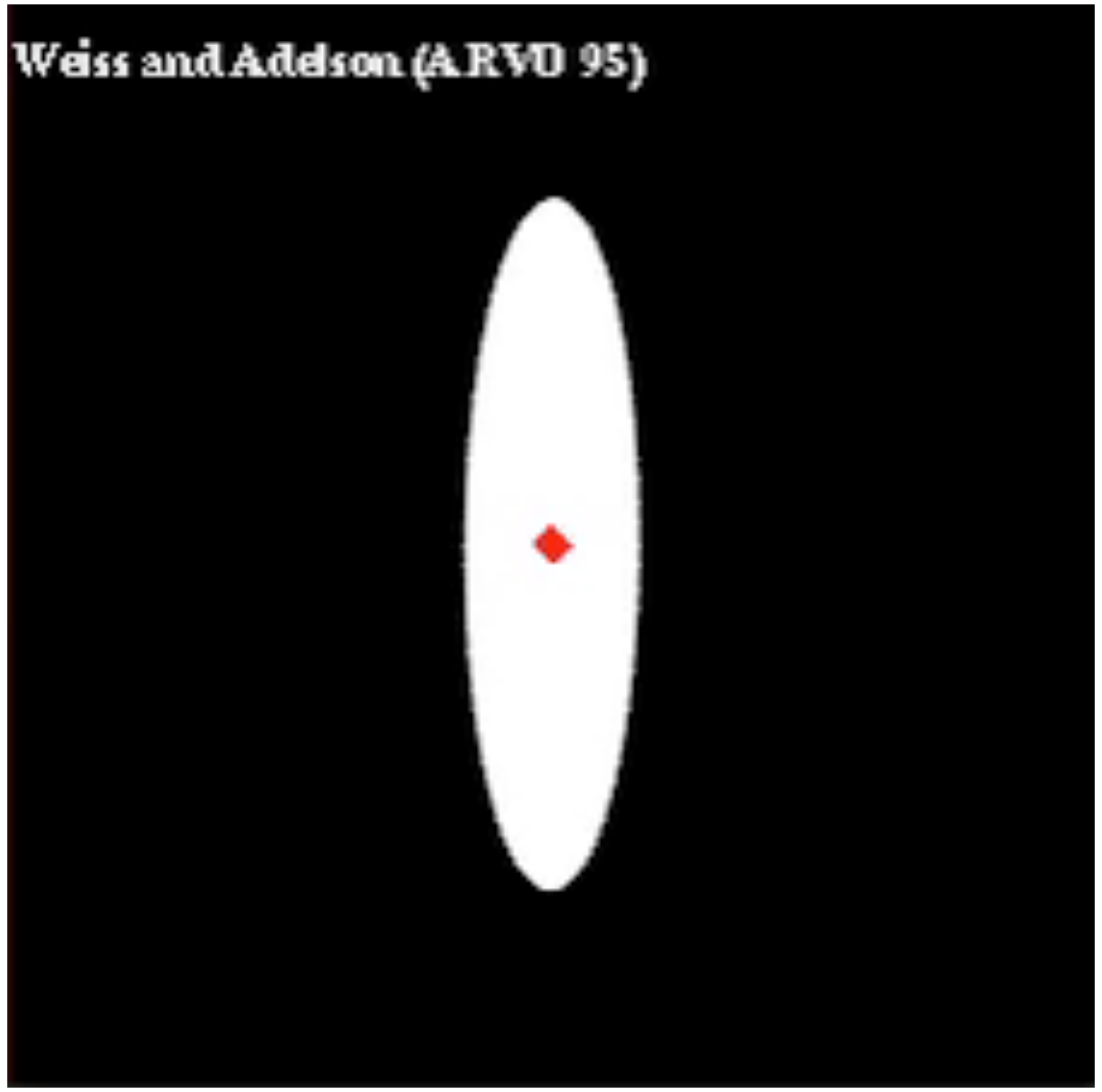
— a 2-D, non-rigid “gelatinous” smoothly deforming shape

3. **Stereokinetic**:

— a circular, rigid disk rolling in 3-D

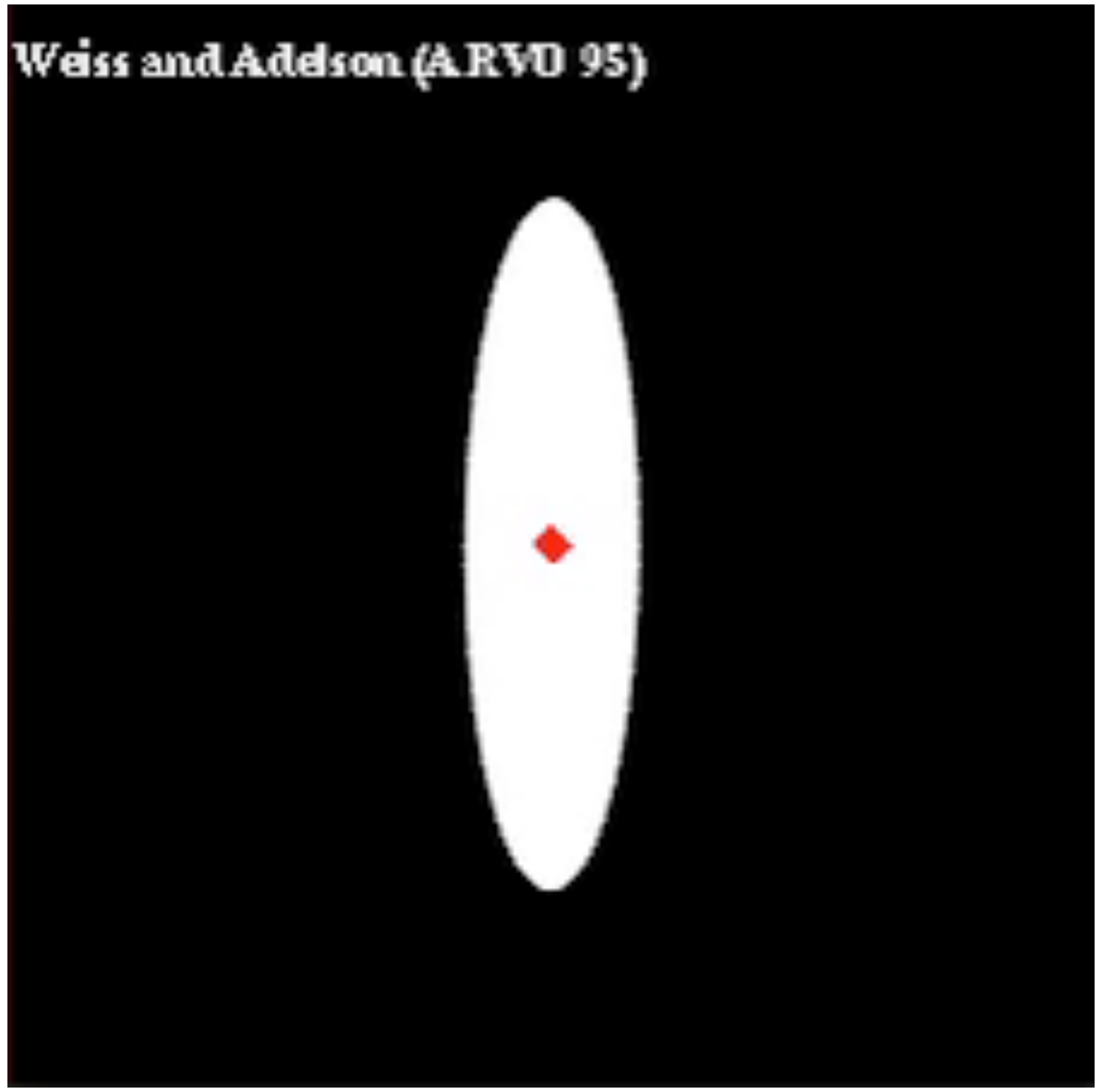
Example 1: Rotating Ellipse

A narrow ellipse oscillating rigidly about its center appears **rigid**



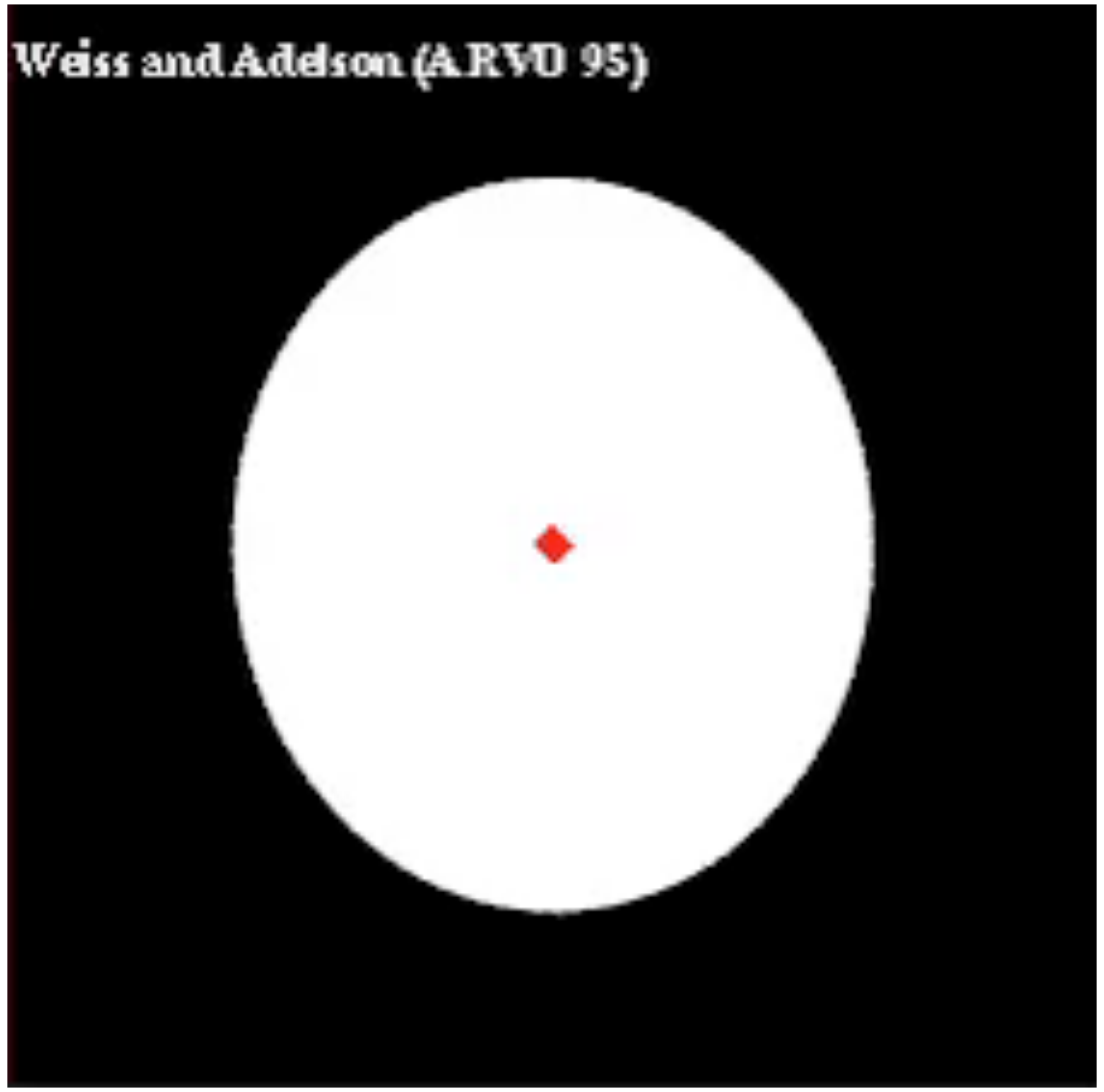
Example 1: Rotating Ellipse

A narrow ellipse oscillating rigidly about its center appears **rigid**



Example 1: Rotating Ellipse

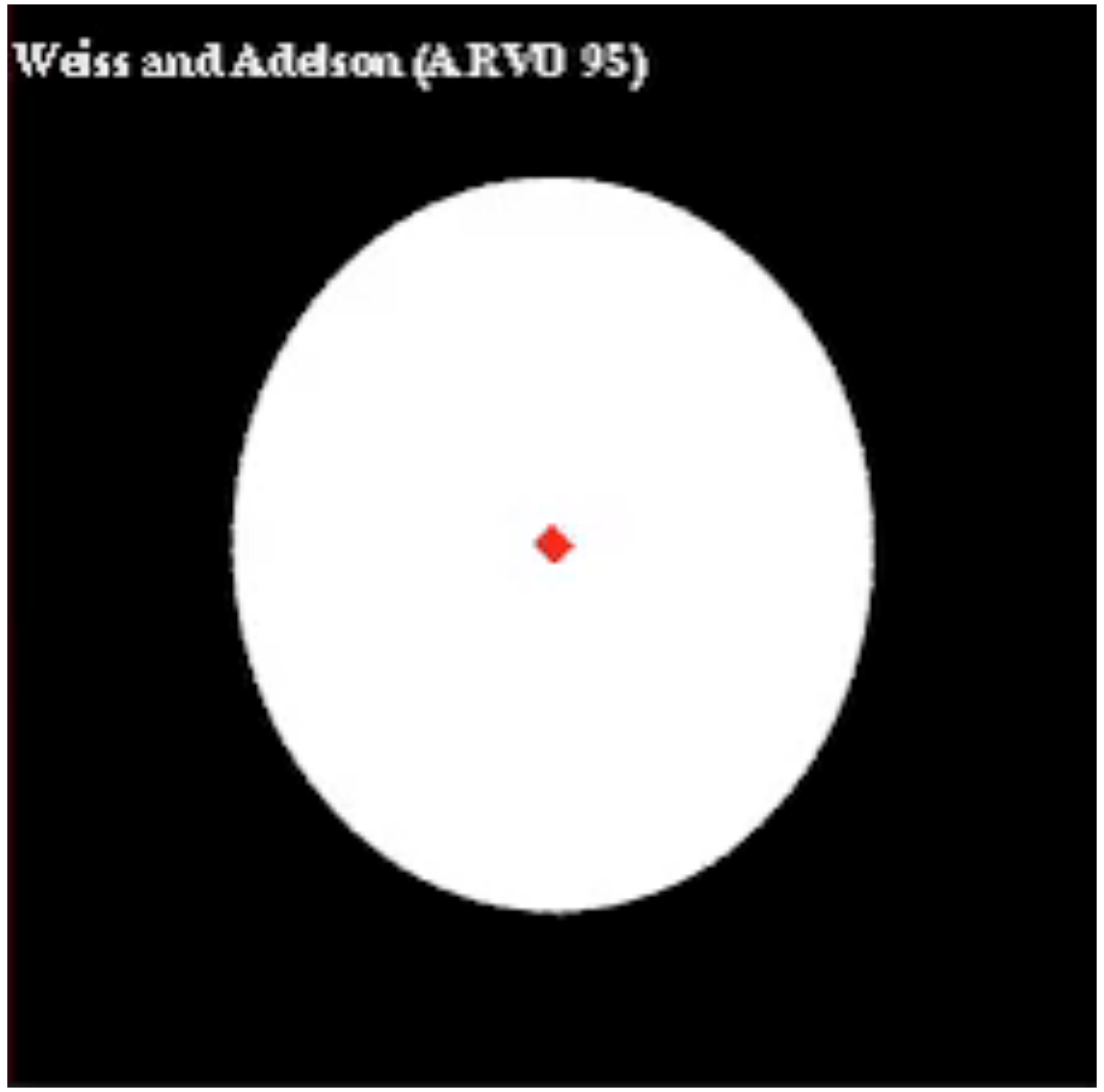
However, a fat ellipse undergoing the same motion appears **nonrigid**



Video credits: Yair Weiss

Example 1: Rotating Ellipse

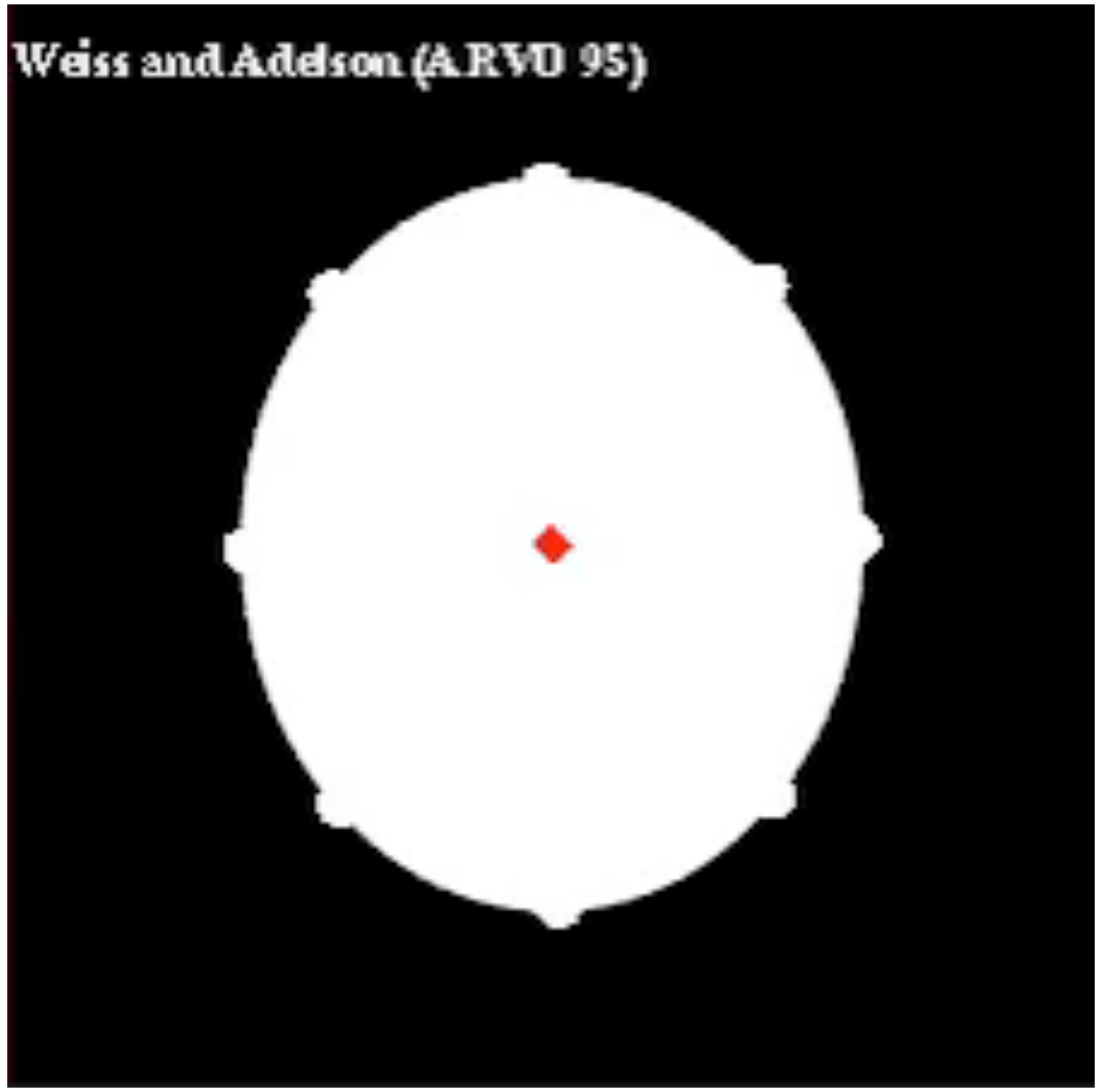
However, a fat ellipse undergoing the same motion appears **nonrigid**



Video credits: Yair Weiss

Example 1: Rotating Ellipse

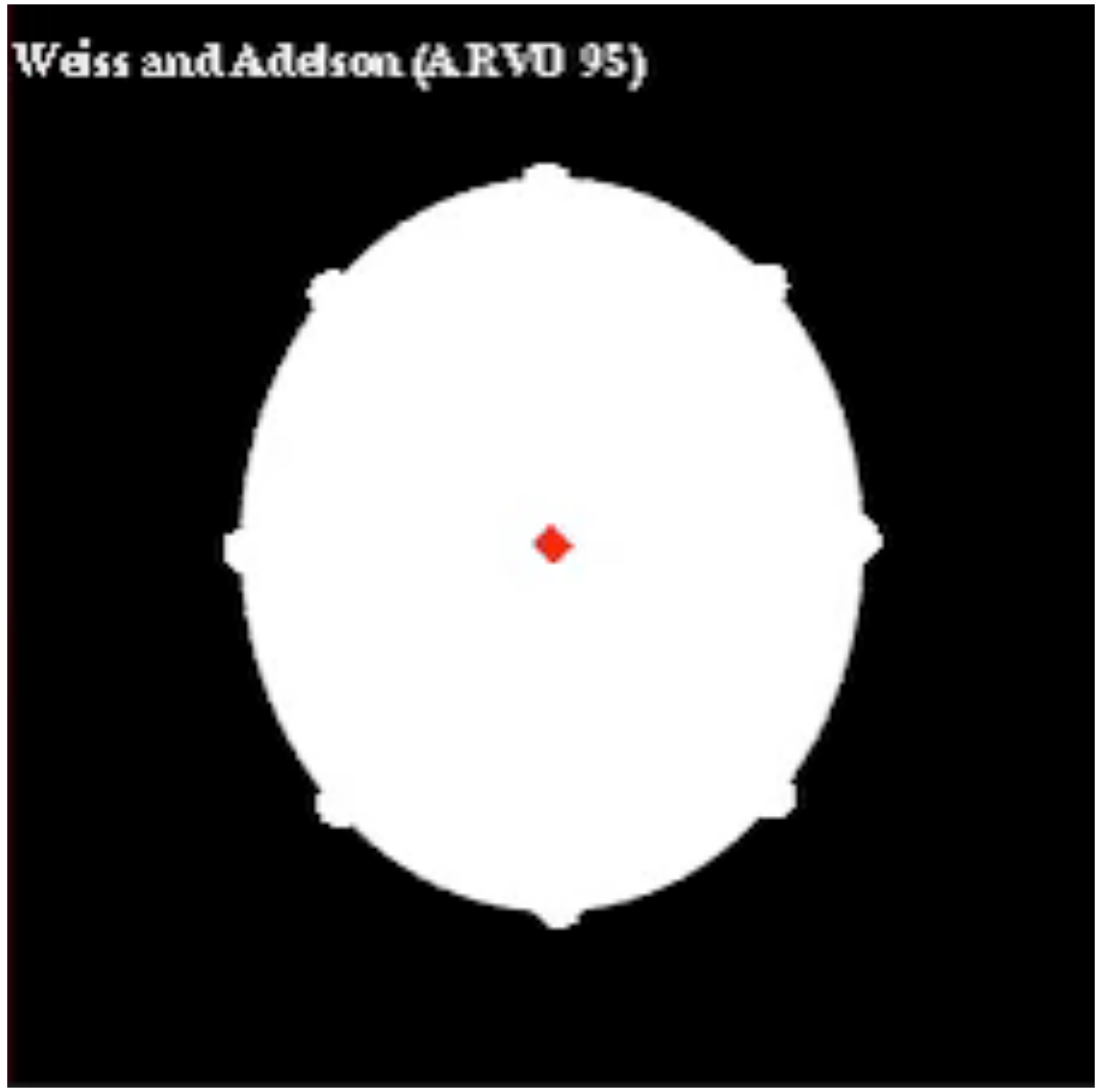
The apparent nonrigidity of a fat ellipse is not really a "visual illusion". A rotating ellipse or a nonrigid pulsating ellipse can cause the exact same stimulation on our retinas. In this sequence the ellipse contour is always doing the same thing, only the markers' motion changes.



Video credits: Yair Weiss

Example 1: Rotating Ellipse

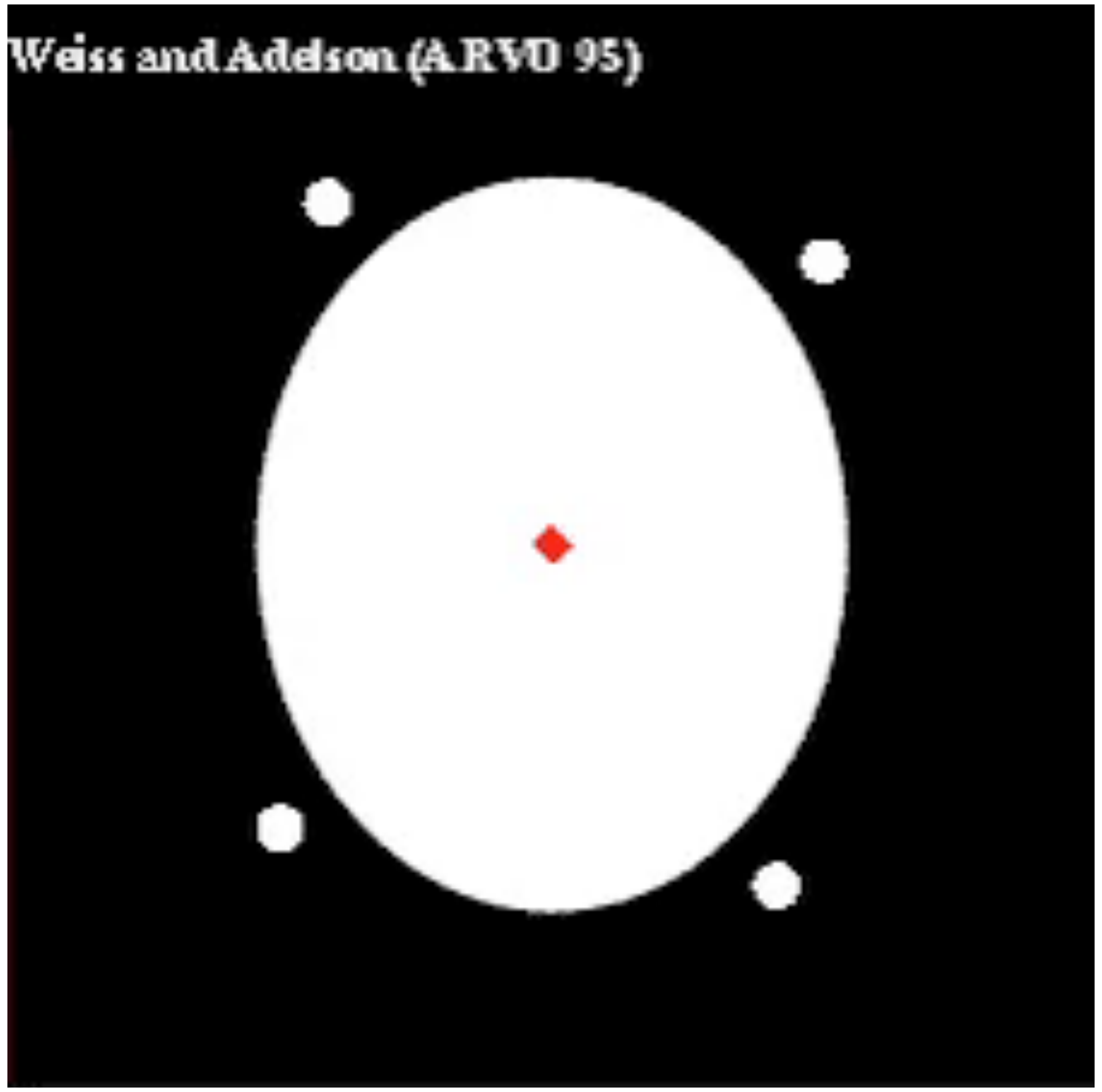
The apparent nonrigidity of a fat ellipse is not really a "visual illusion". A rotating ellipse or a nonrigid pulsating ellipse can cause the exact same stimulation on our retinas. In this sequence the ellipse contour is always doing the same thing, only the markers' motion changes.



Video credits: Yair Weiss

Example 1: Rotating Ellipse

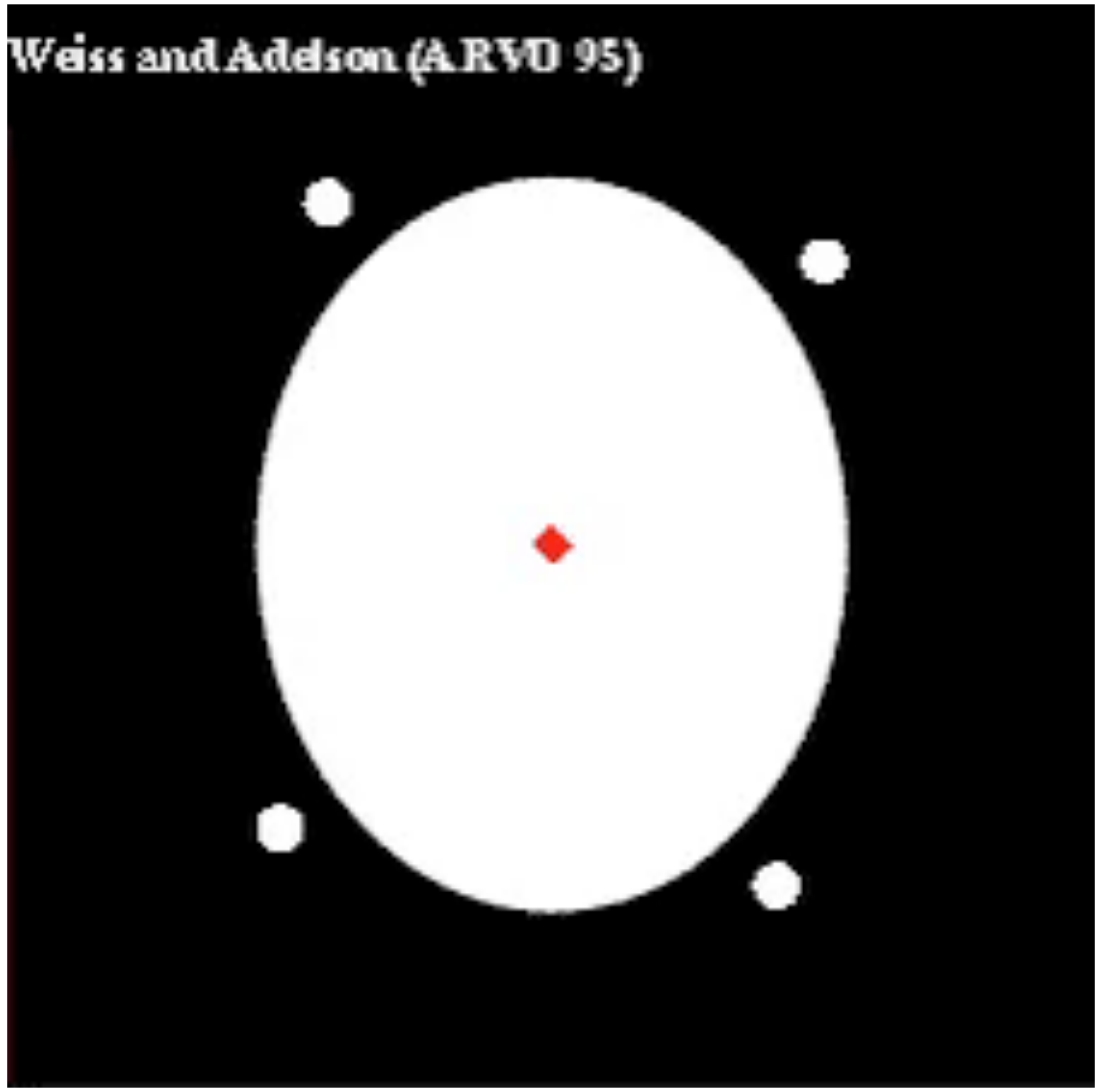
The ellipse's motion can be influenced by features not physically connected to the ellipse. In this sequence the ellipse is always doing the same thing, only the dots' motion changes.



Video credits: Yair Weiss

Example 1: Rotating Ellipse

The ellipse's motion can be influenced by features not physically connected to the ellipse. In this sequence the ellipse is always doing the same thing, only the dots' motion changes.



Video credits: Yair Weiss

Example: Flying Insects and Birds

Bees have very limited stereo perception. How do they fly safely through narrow passages?

Example: Flying Insects and Birds

Bees have very limited stereo perception. How do they fly safely through narrow passages?

A simple strategy would be to balance the speeds of motion of the images of the two walls. If wall A is moving faster than wall B, what should you (as a bee) do?

Example: Flying Insects and Birds

Bee strategy: Balance the optical flow experienced by the two eyes

Figure credit: M. Srinivasan

Example: Flying Insects and Birds

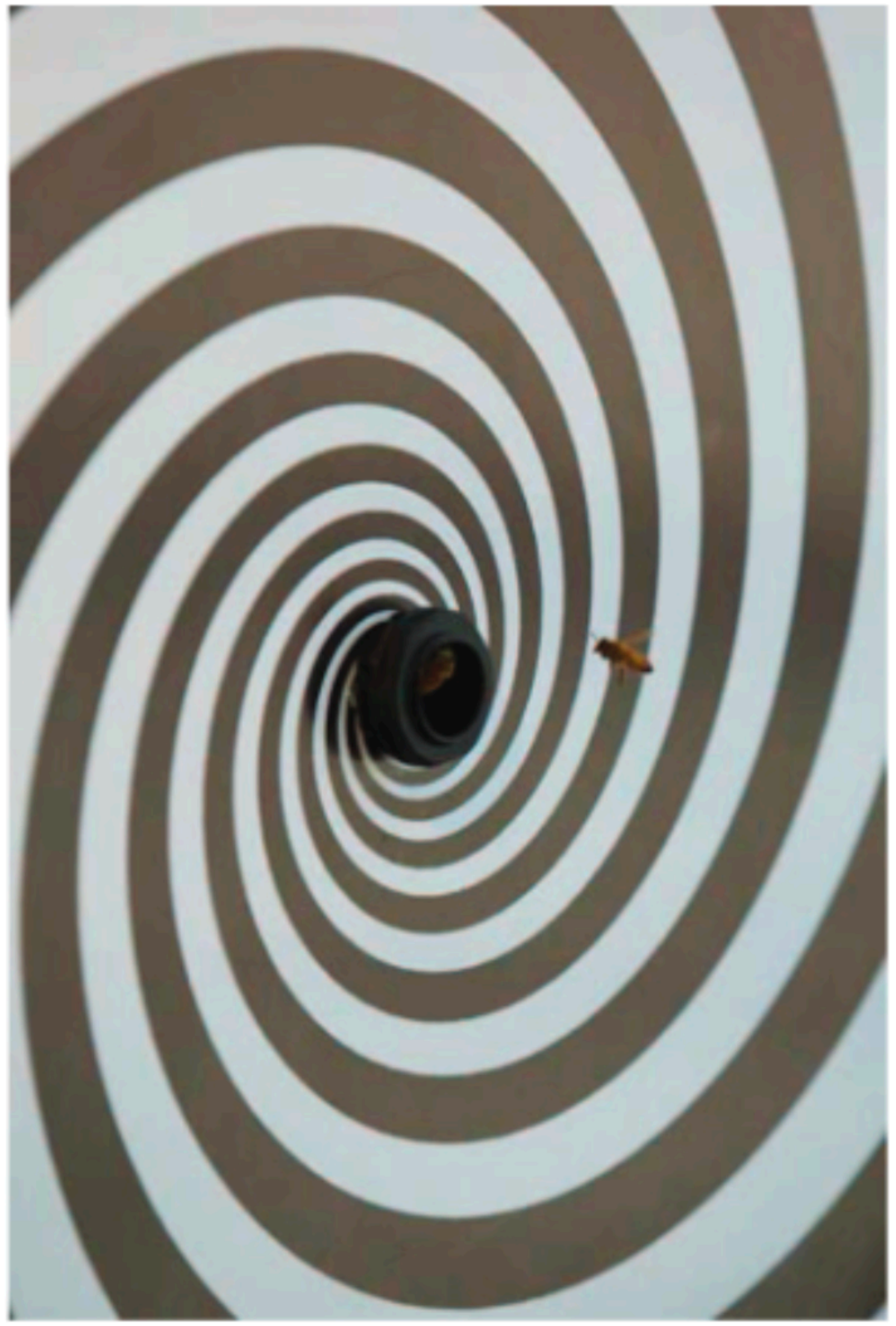
How do bees land safely on surfaces?

During their approach, bees continually adjust their speed to hold constant the optical flow in the vicinity of the target

— approach speed decreases as the target is approached and reduces to zero at the point of touchdown

— no need to estimate the distance to the target at any time

Example: Flying Insects and Birds



Bees approach the surface more slowly if the spiral is rotated to augment the rate of expansion, and more quickly if the spiral is rotated in the opposite direction

Figure credit: M. Srinivasan

Example: Flying Insects and Birds

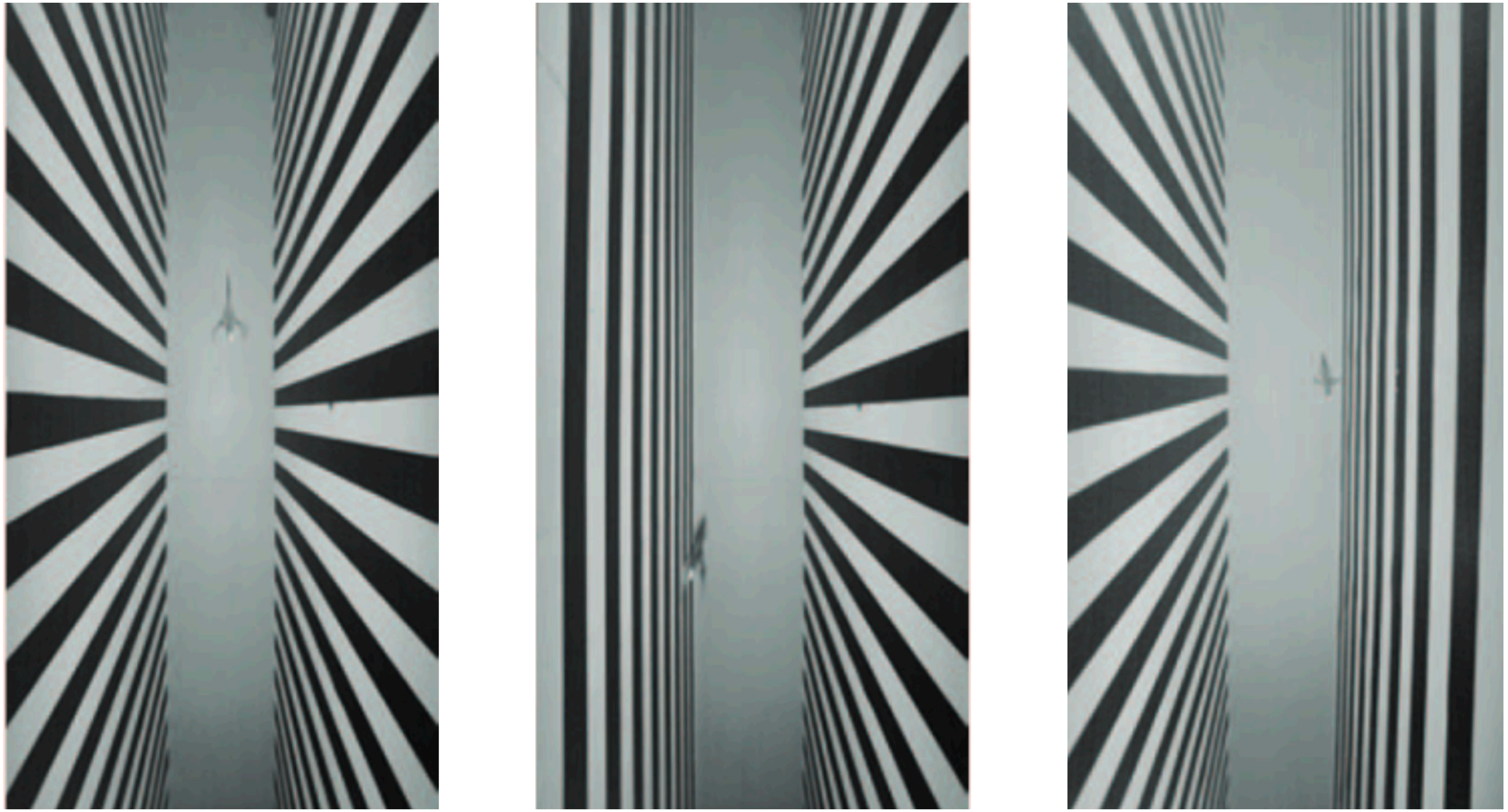


Figure credit: M. Srinivasan

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x, y, t)$$

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x(t), y(t), t)$$

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x(t), y(t), t)$$

Applying the **chain rule for differentiation**, we obtain

$$\frac{dI(x, y, t)}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t$$

where subscripts denote partial differentiation

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x(t), y(t), t)$$

Applying the **chain rule for differentiation**, we obtain

$$\frac{dI(x,y,t)}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t$$

where subscripts denote partial differentiation

Define $u = \dfrac{dx}{dt}$ and $v = \dfrac{dy}{dt}$. Then $[u, v]$ is the 2-D motion and the space of all such u and v is the **2-D velocity space**

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x(t), y(t), t)$$

Applying the **chain rule for differentiation**, we obtain

$$\frac{dI(x,y,t)}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t$$

where subscripts denote partial differentiation

Define $u = \dfrac{dx}{dt}$ and $v = \dfrac{dy}{dt}$. Then $[u, v]$ is the 2-D motion and the space of all such u and v is the **2-D velocity space**

Suppose $\dfrac{dI(x,y,t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x(t), y(t), t)$$

Applying the **chain rule for differentiation**, we obtain

$$\frac{dI(x,y,t)}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t$$

where subscripts denote partial differentiation

Define $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. Then $[u, v]$ is the 2-D motion and the space of all such u and v is the **2-D velocity space**

Suppose $\frac{dI(x,y,t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

Optical Flow **Constraint Equation**

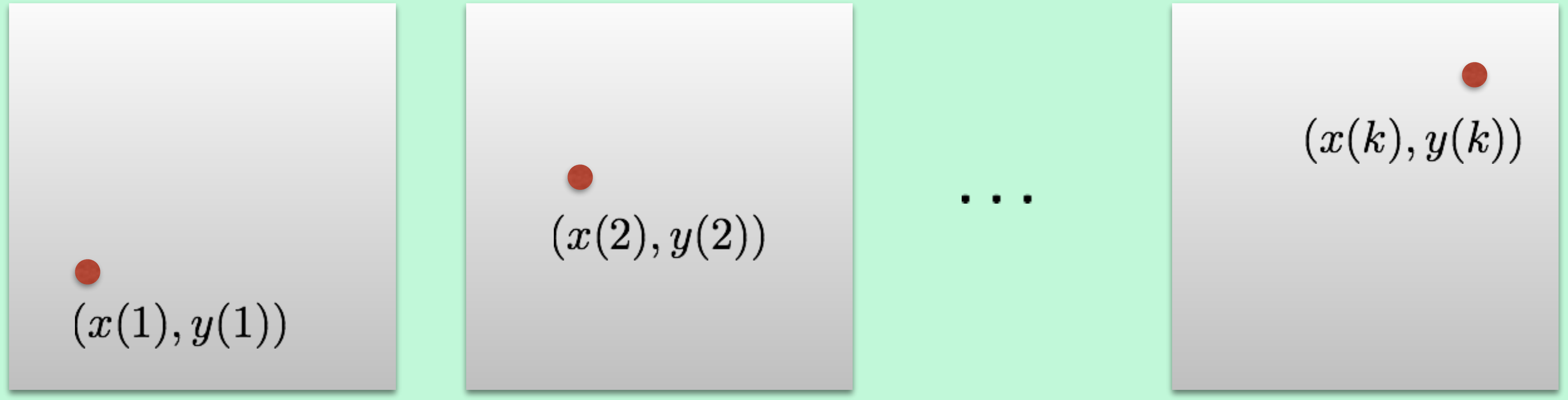
What does this mean, and why is it reasonable?

Suppose $\frac{dI(x,y,t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

Optical Flow **Constraint Equation**

Scene point moving through image sequence



What does this mean, and why is it reasonable?

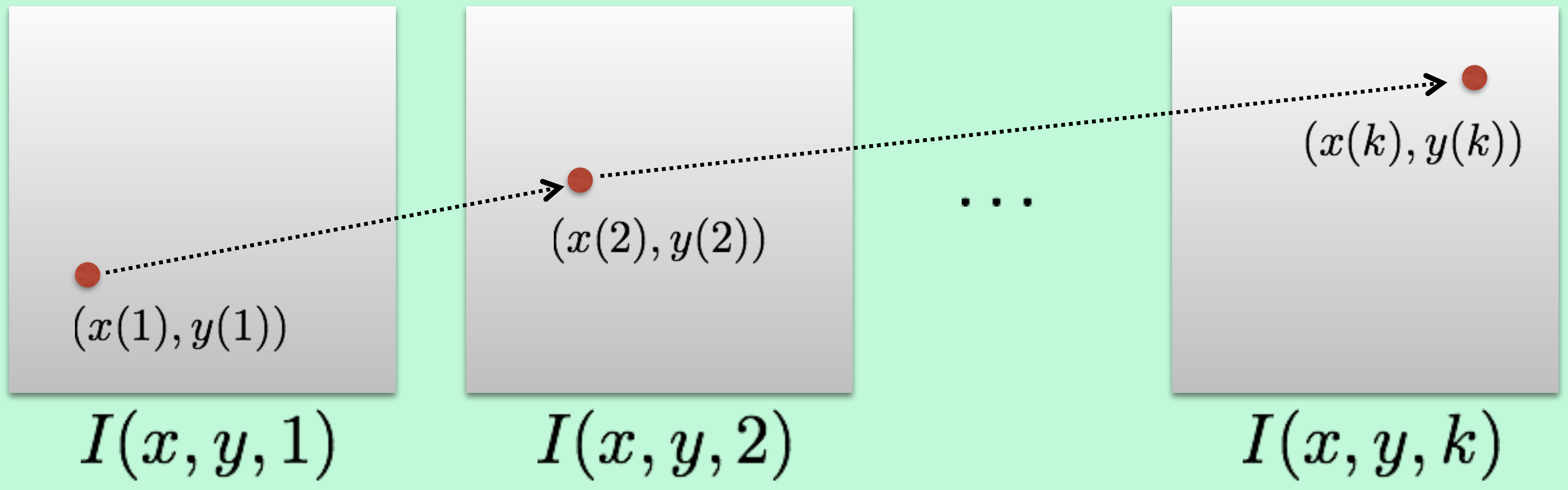
Suppose $\frac{dI(x,y,t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Optical Flow **Constraint Equation**

Scene point moving through image sequence



What does this mean, and why is it reasonable?

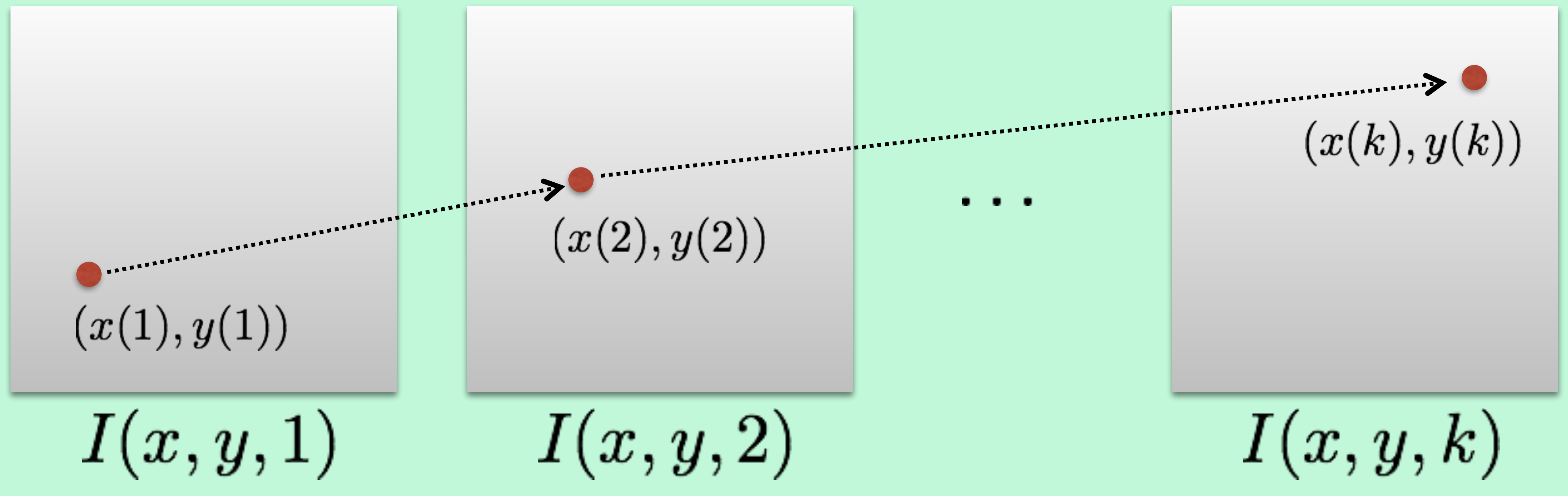
Suppose $\frac{dI(x,y,t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Optical Flow **Constraint Equation**

Brightness Constancy Assumption: Brightness of the point remains the same



$$I(x(t), y(t), t) = C$$

constant

What does this mean, and why is it reasonable?

Suppose $\frac{dI(x, y, t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

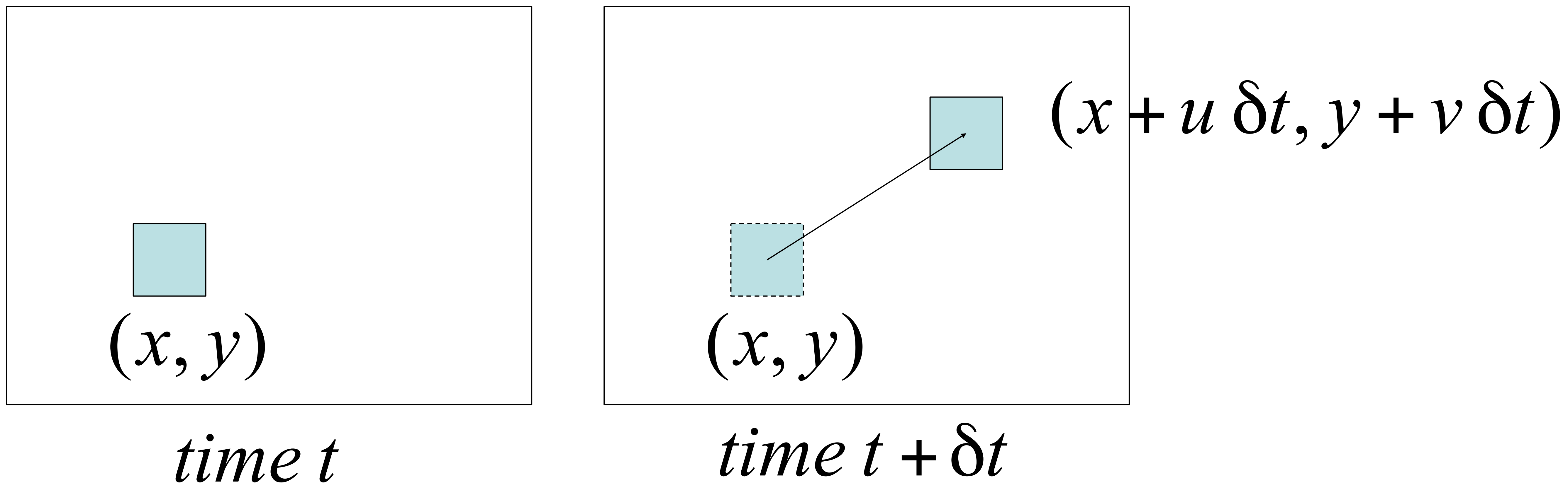
$$I_x u + I_y v + I_t = 0$$

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

For small space-time step, brightness of a point is the same



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

For small space-time step, brightness of a point is the same

Insight:
If the time step is really small,
we can *linearize* the intensity function
(and motion is really-small … think less than a pixel)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Multivariable Taylor Series Expansion

(First order approximation, two variables)

$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) - f_y(a,b)(y-b)$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Multivariable Taylor Series Expansion

(First order approximation, two variables)

$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) - f_y(a,b)(y-b)$$

$$I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x,y,t)$$ assuming small motion

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Multivariable Taylor Series Expansion

(First order approximation, two variables)

$$f(x, y) \approx f(a, b) + f_x(a, b)(x - a) - f_y(a, b)(y - b)$$

partial derivative

$$I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x, y, t)$$

assuming small motion

fixed point

cancel terms

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Multivariable Taylor Series Expansion

(First order approximation, two variables)

$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) - f_y(a,b)(y-b)$$

$$I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x,y,t)$$ assuming small motion

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$ cancel terms

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Multivariable Taylor Series Expansion

(First order approximation, two variables)

$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) - f_y(a,b)(y-b)$$

$$I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x,y,t)$$ assuming small motion

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

divide by δt

take limit $\delta t \to 0$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aside: Derivation of Optical Flow Constraint

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

Multivariable Taylor Series Expansion

(First order approximation, two variables)

$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) - f_y(a,b)(y-b)$$

$$I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x,y,t)$$ assuming small motion

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

divide by δt

take limit $\delta t \to 0$

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$ **Brightness Constancy Equation**

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Forward difference
Sobel filter
Scharr filter
…

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Forward difference
Sobel filter
Scharr filter
…

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Frame Differencing: Example

$$\underset{t+1}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 10 & 10 & 10 \\ 1 & 1 & 10 & 10 & 10 \\ 1 & 1 & 10 & 10 & 10 \end{bmatrix}} - \underset{t}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 10 & 10 & 10 & 10 \\ 1 & 10 & 10 & 10 & 10 \\ 1 & 10 & 10 & 10 & 10 \\ 1 & 10 & 10 & 10 & 10 \end{bmatrix}} = \underset{I_t = \frac{\partial I}{\partial t}}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -9 & -9 & -9 & -9 \\ 0 & -9 & 0 & 0 & 0 \\ 0 & -9 & 0 & 0 & 0 \\ 0 & -9 & 0 & 0 & 0 \end{bmatrix}}$$

(example of a forward temporal difference)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

$$I_x = \frac{\partial I}{\partial x}$$

-	0	0	0	-
-	0	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-

-1 0 1

$$I_y = \frac{\partial I}{\partial y}$$

-	-	-	-	-
0	0	0	0	0
0	9	9	9	9
0	0	0	0	0
0	0	0	0	0
-	-	-	-	-

-1
0
1

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	-9	-9	-9	-9
0	-9	0	0	0
0	-9	0	0	0
0	-9	0	0	0

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

How do you compute this?

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

We need to solve for this!
(this is the unknown in the optical flow problem)

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

(u, v)

Solution lies on a line

Cannot be found uniquely
with a single constraint

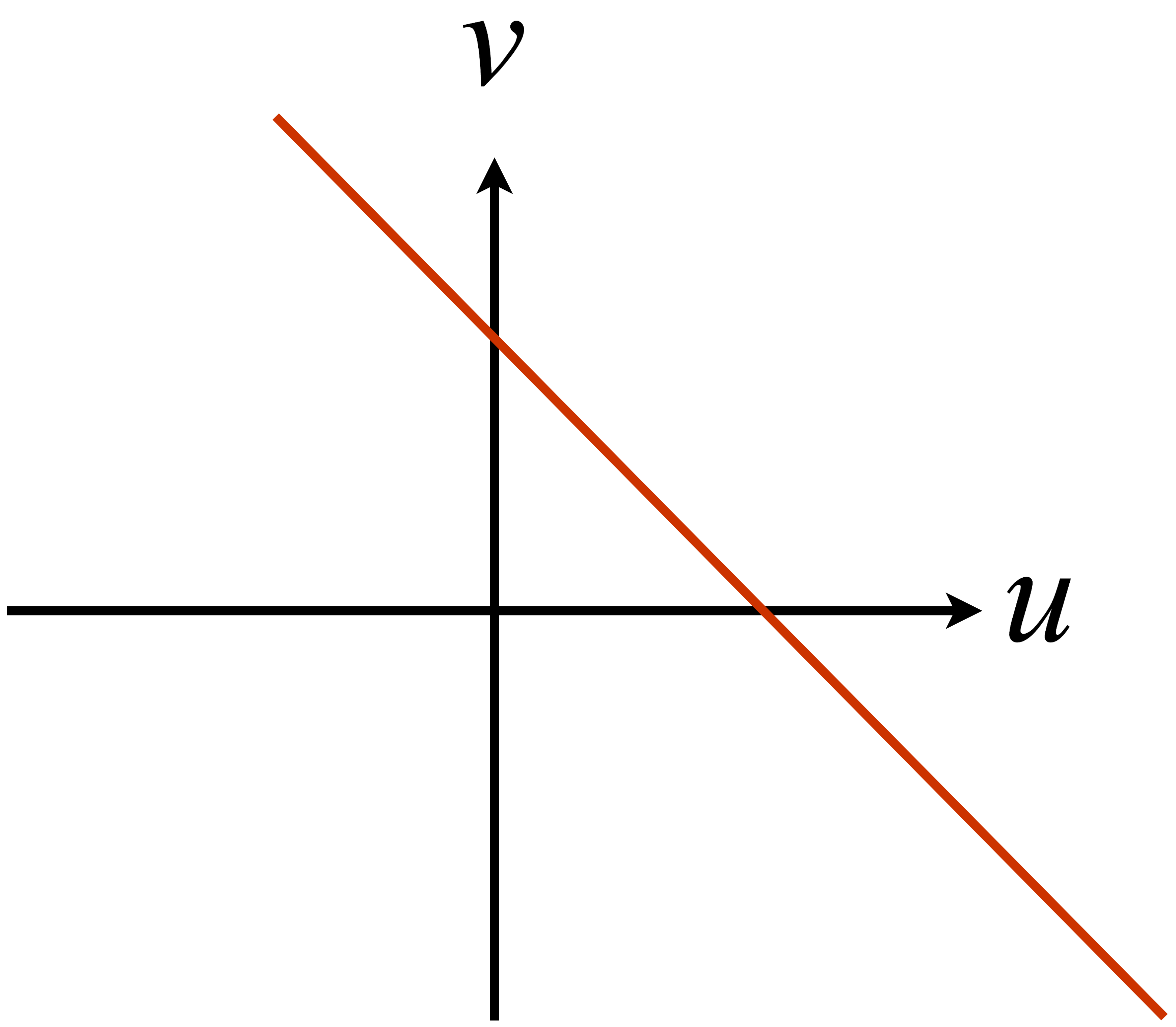
Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Optical Flow **Constraint Equation**

$$\boxed{I_x u + I_y v + I_t = 0}$$

many combinations of u and v will satisfy the equality



Equation determines a **straight line** in velocity space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

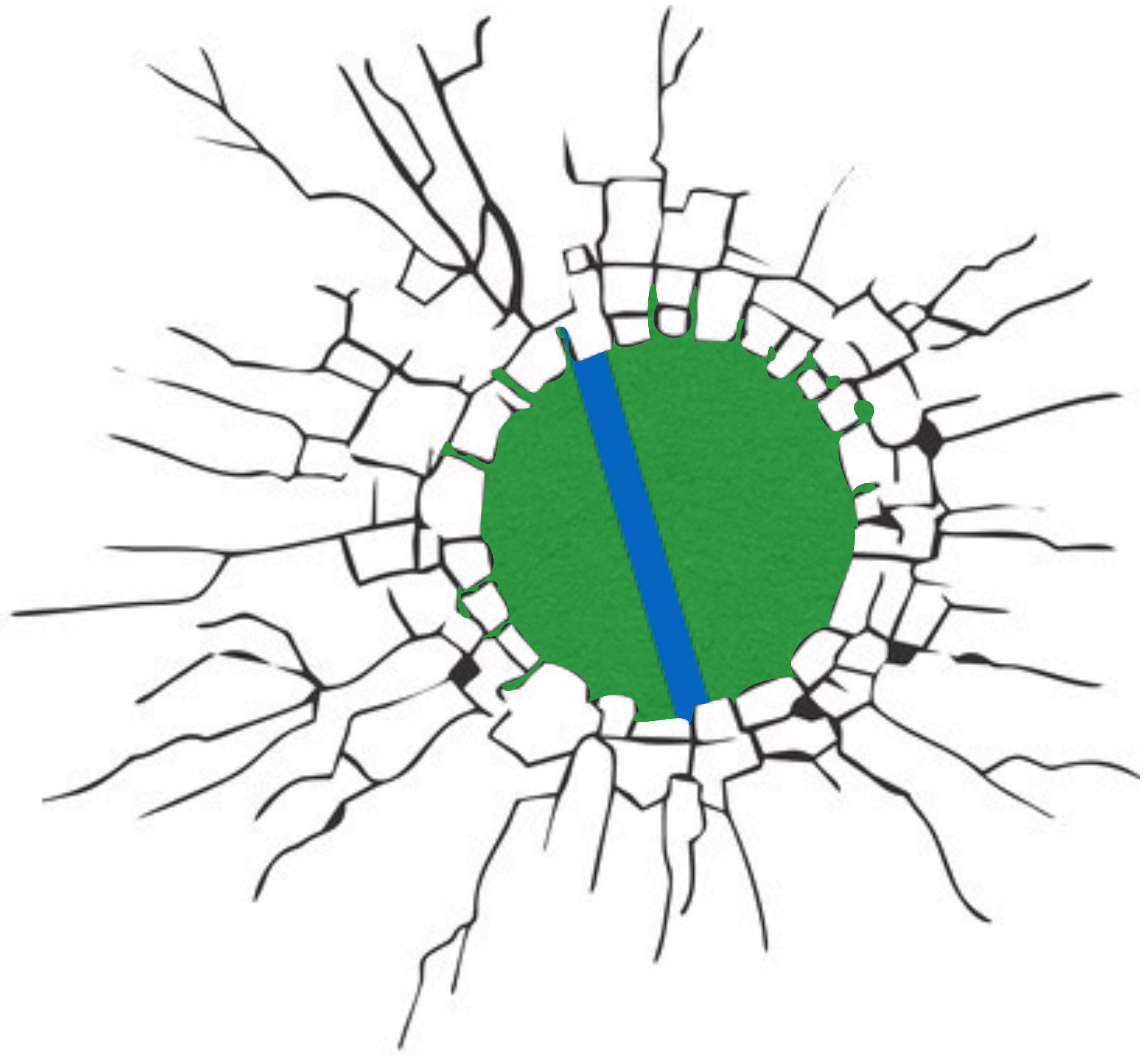
Flow **Ambiguity**

- The stripes can be interpreted as moving vertically, horizontally (rotation), or somewhere in between!
- The component of velocity parallel to the edge is unknown

Flow **Ambiguity**

- The stripes can be interpreted as moving vertically, horizontally (rotation), or somewhere in between!
- The component of velocity parallel to the edge is unknown

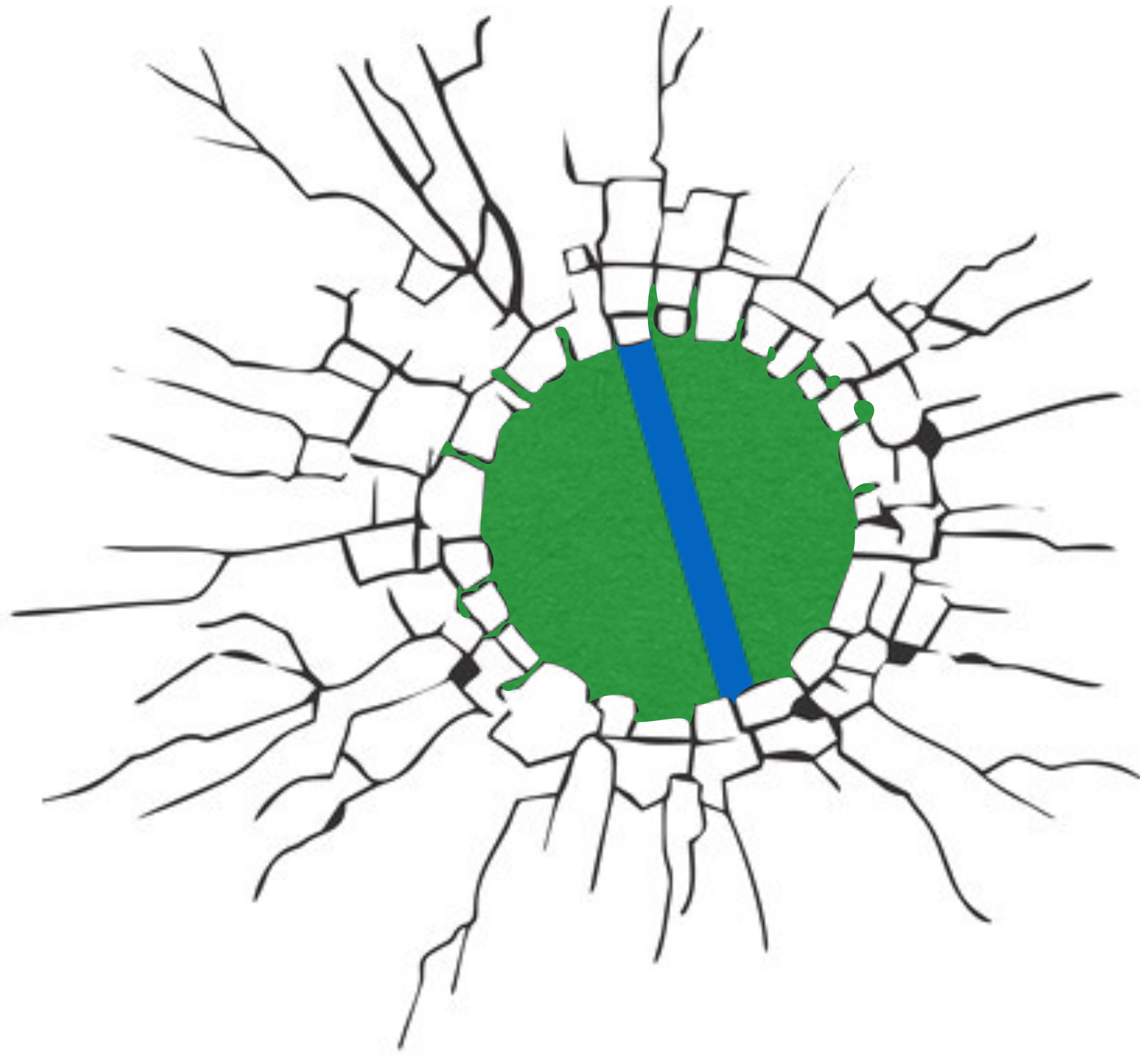
Aperture Problem



In which direction is the line moving?

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem



In which direction is the line moving?

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

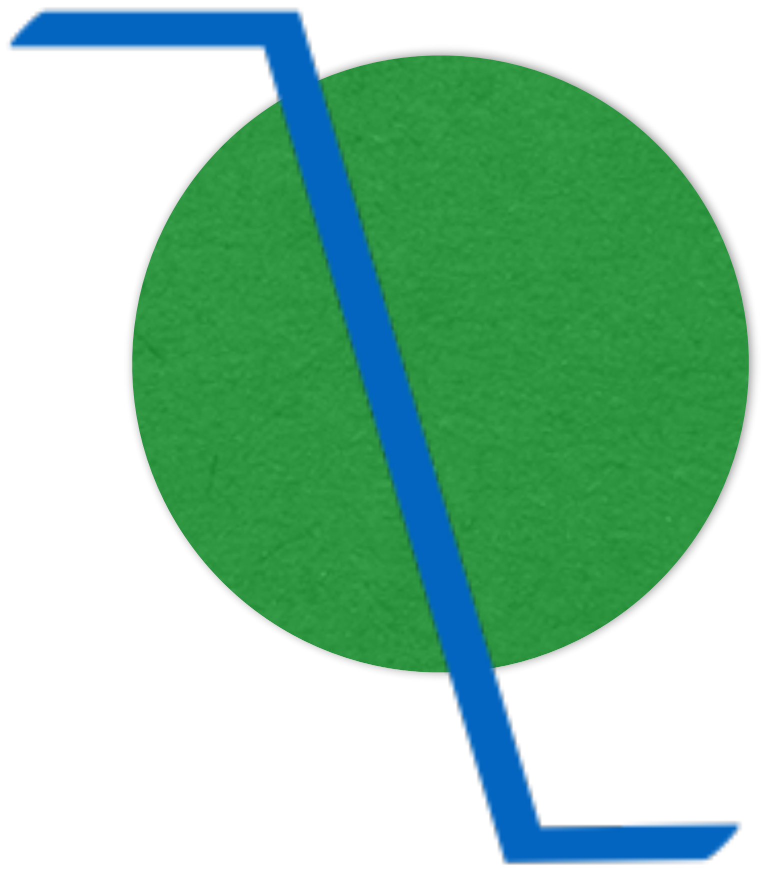


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

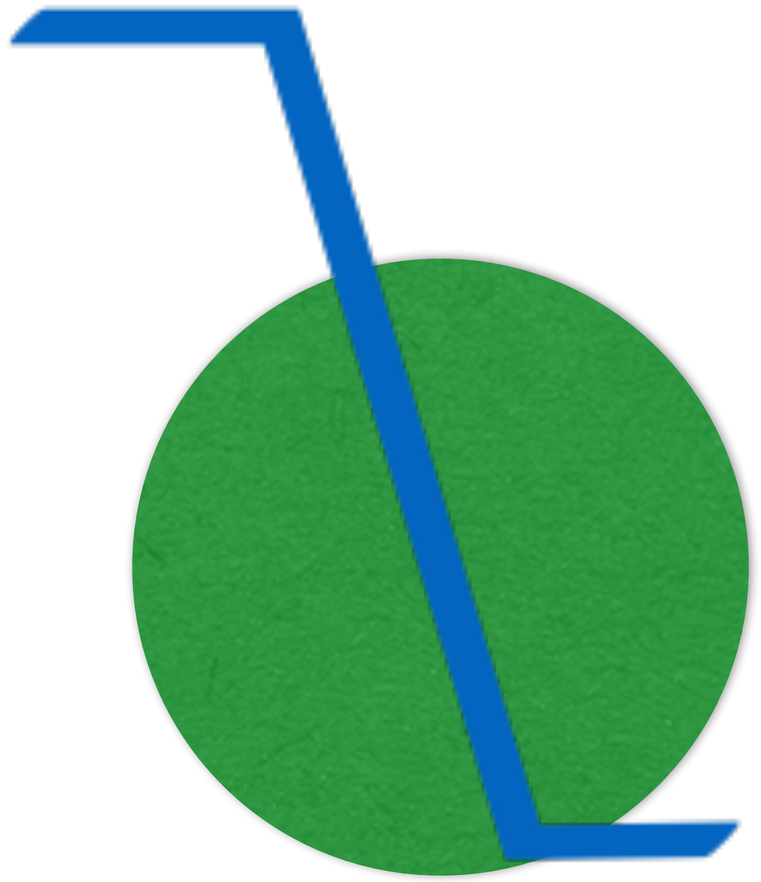


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

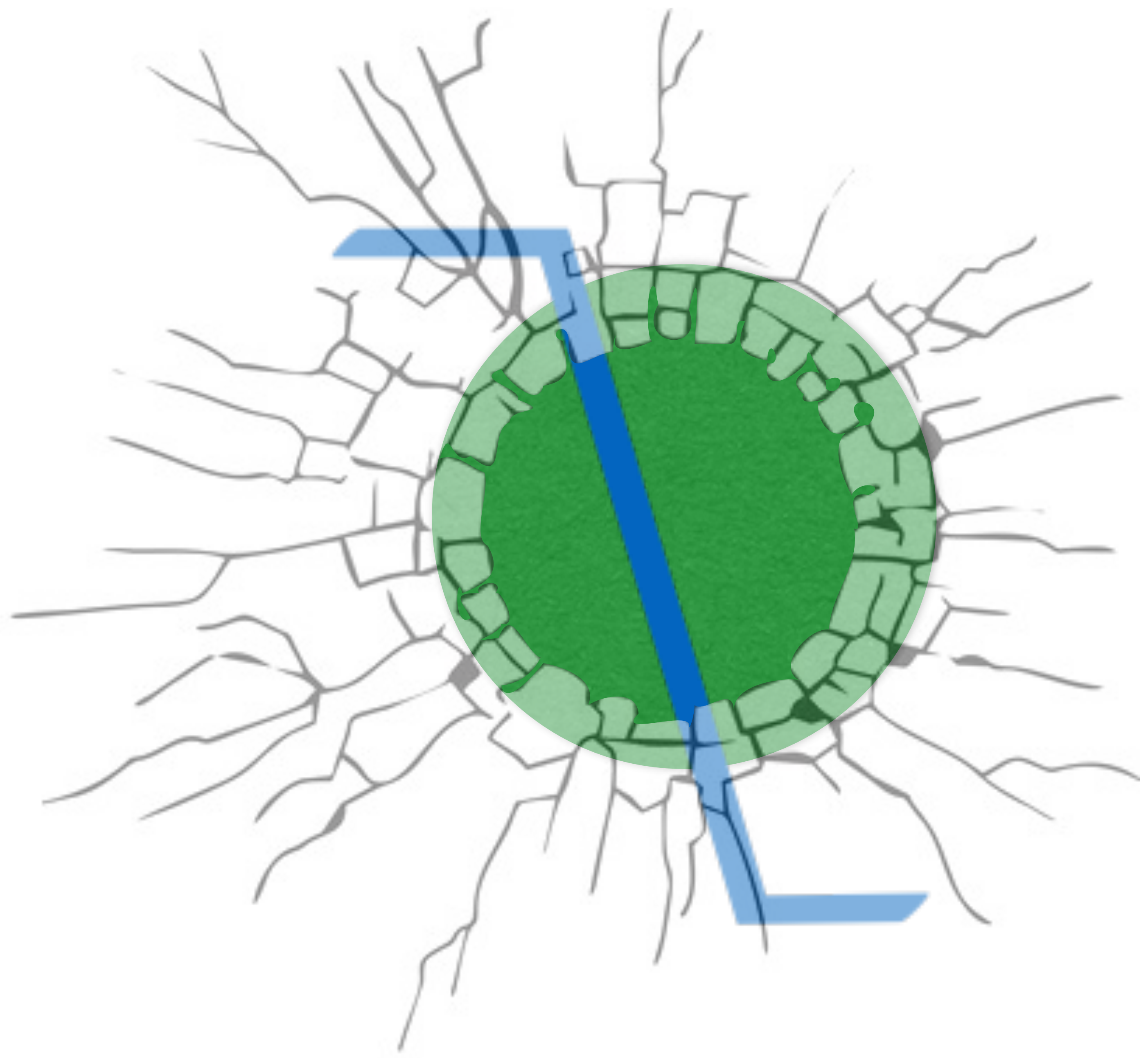


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

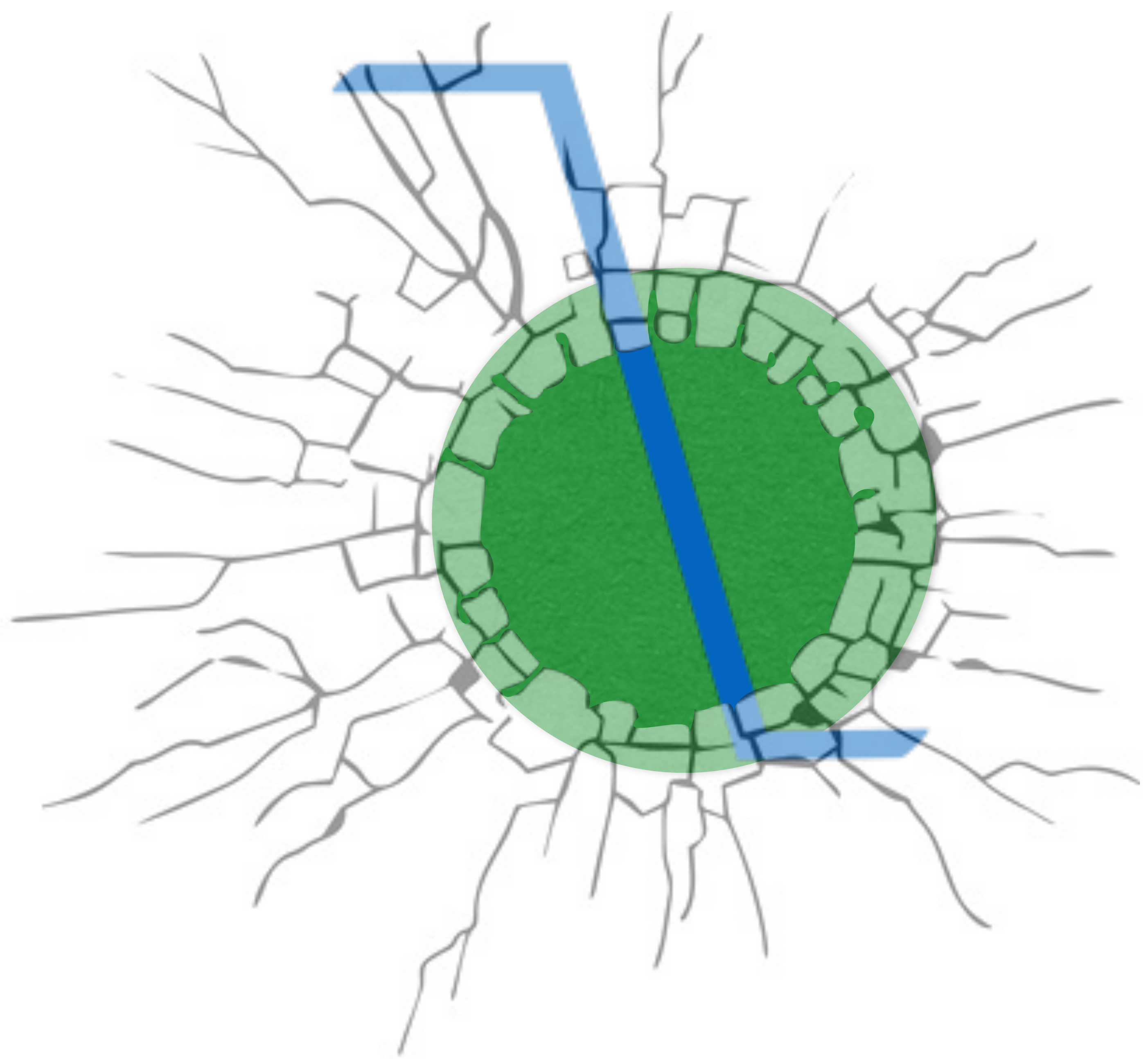
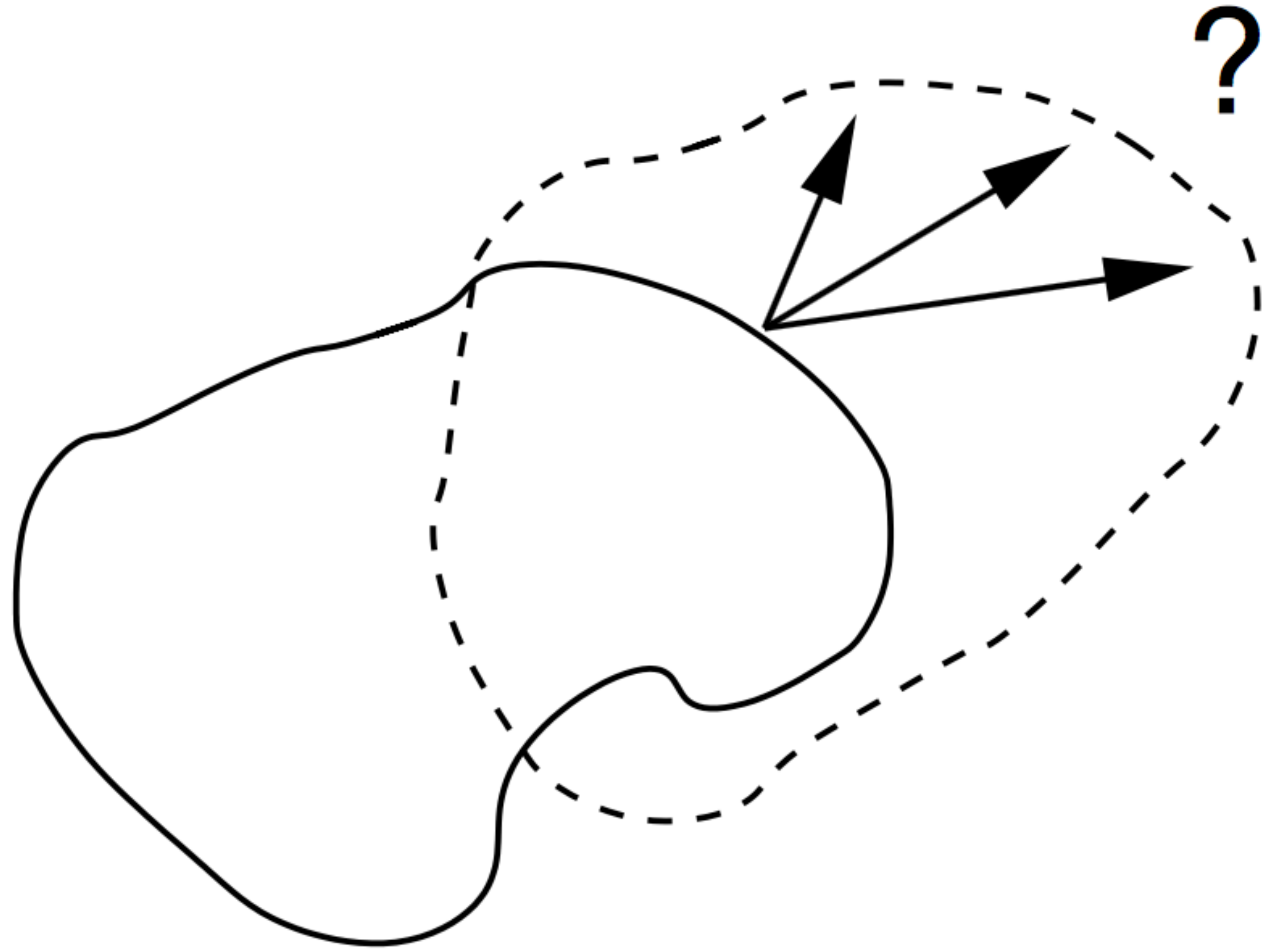


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

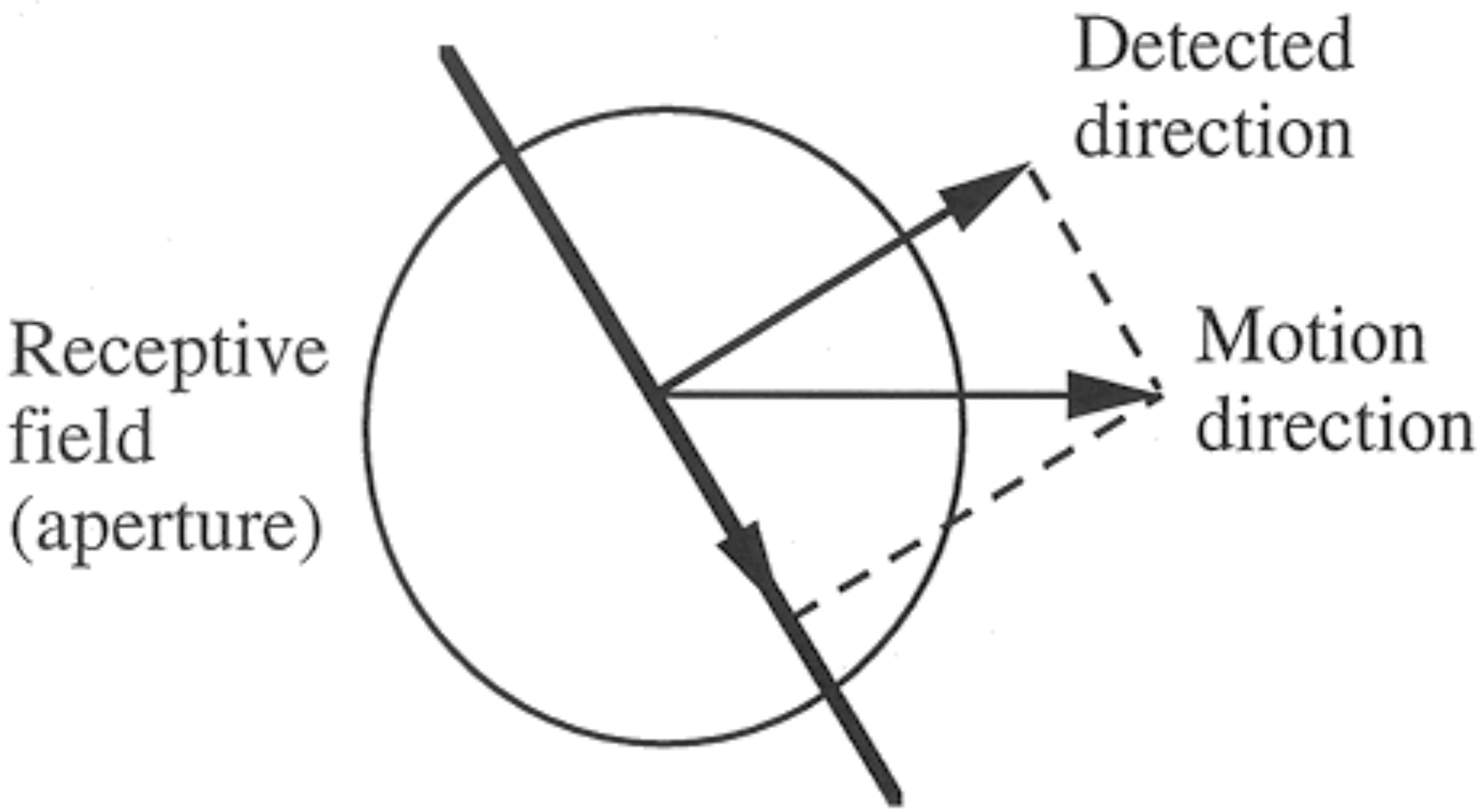
Aperture Problem



— Without distinct features to track, the true visual motion is ambiguous

— Locally, one can compute only the component of the visual motion in the direction perpendicular to the contour

Aperture Problem



— Without distinct features to track, the true visual motion is ambiguous

— Locally, one can compute only the component of the visual motion in the direction perpendicular to the contour

Lucas-Kanade

Observations:

1. The 2-D motion, $[u, v]$, at a given point, $[x, y]$, has two degrees-of-freedom

2. The partial derivatives, I_x, I_y, I_t, provide one constraint

3. The 2-D motion, $[u, v]$, cannot be determined locally from I_x, I_y, I_t alone

Lucas-Kanade

Observations:

1. The 2-D motion, $[u, v]$, at a given point, $[x, y]$, has two degrees-of-freedom

2. The partial derivatives, I_x, I_y, I_t, provide one constraint

3. The 2-D motion, $[u, v]$, cannot be determined locally from I_x, I_y, I_t alone

Lucas–Kanade Idea:

Obtain additional local constraint by computing the partial derivatives, I_x, I_y, I_t, in a window centered at the given $[x, y]$

Lucas-Kanade

Observations:

1. The 2-D motion, $[u, v]$, at a given point, $[x, y]$, has two degrees-of-freedom

2. The partial derivatives, I_x, I_y, I_t, provide one constraint

3. The 2-D motion, $[u, v]$, cannot be determined locally from I_x, I_y, I_t alone

Lucas–Kanade Idea:

Obtain additional local constraint by computing the partial derivatives, I_x, I_y, I_t, in a window centered at the given $[x, y]$

Constant Flow Assumption: nearby pixels will likely have same optical flow

Lucas-Kanade

Optical Flow Constraint Equation: $I_x u + I_y v + I_t = 0$

Suppose $[x_1, y_1] = [x, y]$ is the (original) center point in the window. Let $[x_2, y_2]$ be any other point in the window. This gives us two equations that we can write

$$I_{x_1} u + I_{y_1} v = -I_{t_1}$$
$$I_{x_2} u + I_{y_2} v = -I_{t_2}$$

and that can be solved locally for u and v as

$$\begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \end{bmatrix}^{-1} \begin{bmatrix} I_{t_1} \\ I_{t_2} \end{bmatrix}$$

provided that u and v are the same in both equations and provided that the required matrix inverse exists.

Lucas-Kanade

Optical Flow Constraint Equation: $I_x u + I_y v + I_t = 0$

Considering all n points in the window, one obtains

$$\begin{gathered} I_{x_1} u + I_{y_1} v = -I_{t_1} \\ I_{x_2} u + I_{y_2} v = -I_{t_2} \\ \vdots \\ I_{x_n} u + I_{y_n} v = -I_{t_n} \end{gathered}$$

which can be written as the matrix equation

$$\mathbf{A}\mathbf{v} = \mathbf{b}$$

where $\mathbf{v} = [u, v]^T$, $\mathbf{A} = \begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix}$ and $\mathbf{b} = -\begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \\ I_{t_n} \end{bmatrix}$

Lucas-Kanade

The standard least squares solution, $\bar{\mathbf{v}}$, to is

$$\bar{\mathbf{v}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

again provided that u and v are the same in all equations and provided that the rank of $\mathbf{A}^T\mathbf{A}$ is 2 (so that the required inverse exists)

Lucas-Kanade

The standard least squares solution, $\bar{\mathbf{v}}$, to is

$$\begin{bmatrix} I_{x_1} & I_{x_2} & \cdots & I_{x_n} \\ I_{y_1} & I_{y_2} & \cdots & I_{y_n} \end{bmatrix} \begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix}$$

$$\bar{\mathbf{v}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

again provided that u and v are the same in all equations and provided that the rank of $\mathbf{A}^T\mathbf{A}$ is 2 (so that the required inverse exists)

Lucas-Kanade

Note that we can explicitly write down an expression for $\mathbf{A}^T\mathbf{A}$ as

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & I_y^2 \end{bmatrix}$$

which is identical to the matrix $\mathbf{C}$ that we saw in the context of Harris corner detection

Lucas-Kanade

Note that we can explicitly write down an expression for $\mathbf{A}^T\mathbf{A}$ as

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & I_y^2 \end{bmatrix}$$

which is identical to the matrix $\mathbf{C}$ that we saw in the context of Harris corner detection

What does that mean?

Lucas-Kanade **Summary**

A dense method to compute motion, $[u, v]$ at every location in an image

Key Assumptions:

1. Motion is slow enough and smooth enough that differential methods apply (i.e., that the partial derivatives, I_x, I_y, I_t, are well-defined)

2. The optical flow constraint equation holds (i.e., $\frac{dI(x,y,t)}{dt} = 0$)

3. A window size is chosen so that motion, $[u, v]$, is constant in the window

4. A window size is chosen so that the rank of $\mathbf{A}^T\mathbf{A}$ is 2 for the window

Aside: Optical Flow Smoothness Constraint

Many methods trade off a ‘departure from the optical flow constraint’ cost with a ‘departure from smoothness’ cost.

The optimization objective to minimize becomes

$$E = \int \int (I_x u + I_y v + I_t)^2 + \lambda(|| \bigtriangledown u||^2 + || \bigtriangledown v||^2)$$

where λ is a weighing parameter.

Horn-Schunck Optical Flow

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ \overbrace{E_s(i,j)}^{\text{smoothness}} + \overbrace{\underbrace{\lambda}_{\text{weight}} E_d(i,j)}^{\text{brightness constancy}} \right\}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

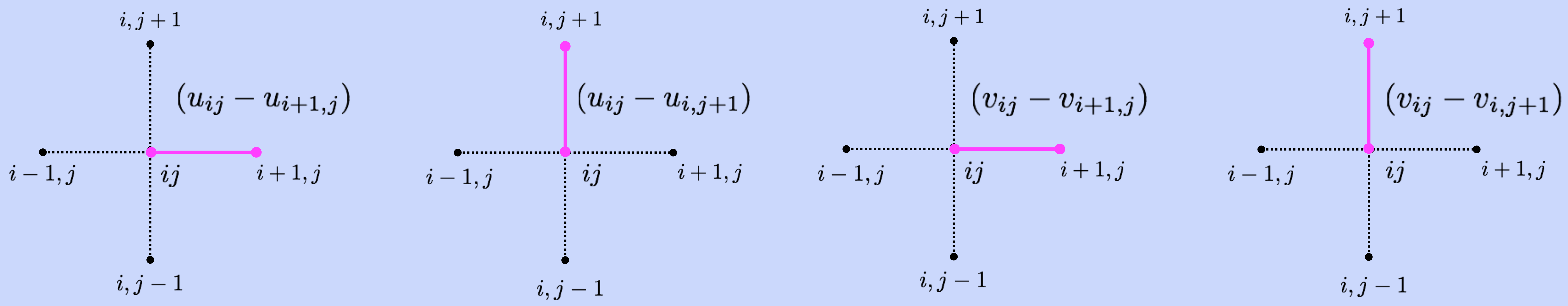
Horn-Schunck Optical Flow

Brightness constancy

$$E_d(i,j) = \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

Smoothness

$$E_s(i,j) = \frac{1}{4}\left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right]$$



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Optical Flow and 2D Motion

Motion is geometric, **Optical flow** is radiometric

Usually we assume that optical flow and 2-D motion coincide ... but this is not always the case!

Optical flow with **no motion:**

. . . moving light source(s), lights going on/off, inter-reflection, shadows

Motion with **no optical flow:**

. . . spinning cylinder, sphere.

Optical Flow **Summary**

Motion, like binocular stereo, can be formulated as a matching problem. That is, given a scene point located at (x_0, y_0) in an image acquired at time t_0, what is its position, (x_1, y_1), in an image acquired at time t_1?

Assuming image intensity does not change as a consequence of motion, we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

where $[u, v]$, is the 2-D motion at a given point, $[x, y]$, and I_x, I_y, I_t are the partial derivatives of intensity with respect to x, y, and t

Lucas–Kanade is a dense method to compute the motion, $[u, v]$, at every location in an image