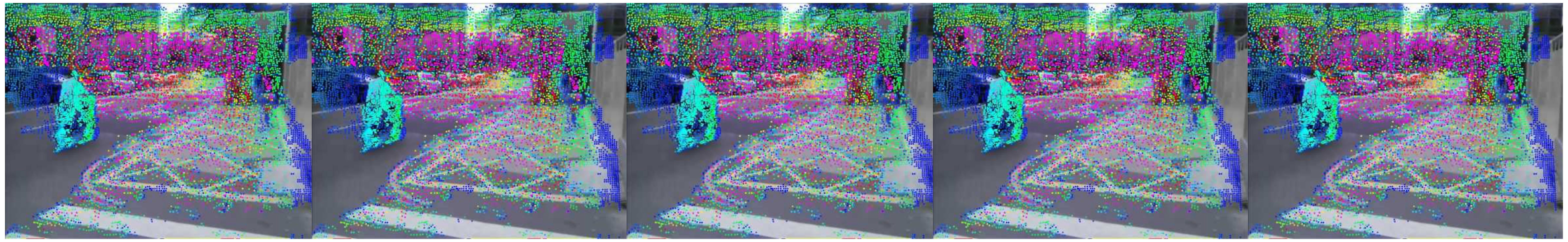
CPSC 425: Computer Vision



Lecture 15: Optical Flow

Menu for Today

Topics:

- **Stereo** recap, 1D vs 2D motion
- **Optical Flow**
- **Brightness** Constancy
- **Lucas Kanade**

Readings:

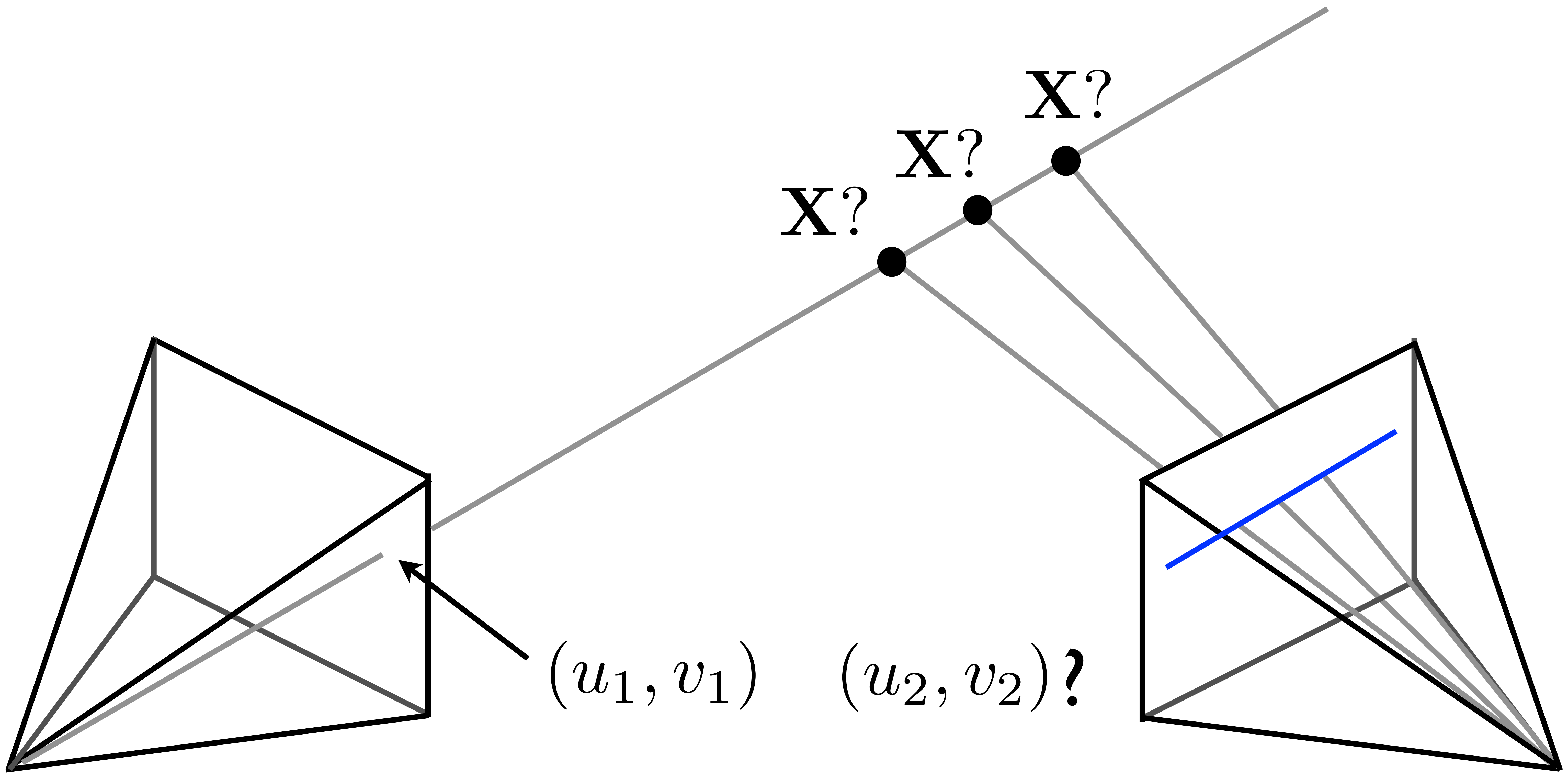
- **Today's** Lecture: Szeliski 12.1, 12.3-12.4, 9.3

Reminders:

- **Midterm results** will be available very soon (likely tomorrow)
- **Assignment 4**: RANSAC and Panoramas due **November 9th**

Epipolar Line

- How do we transfer points between 2 views? (non-planar)



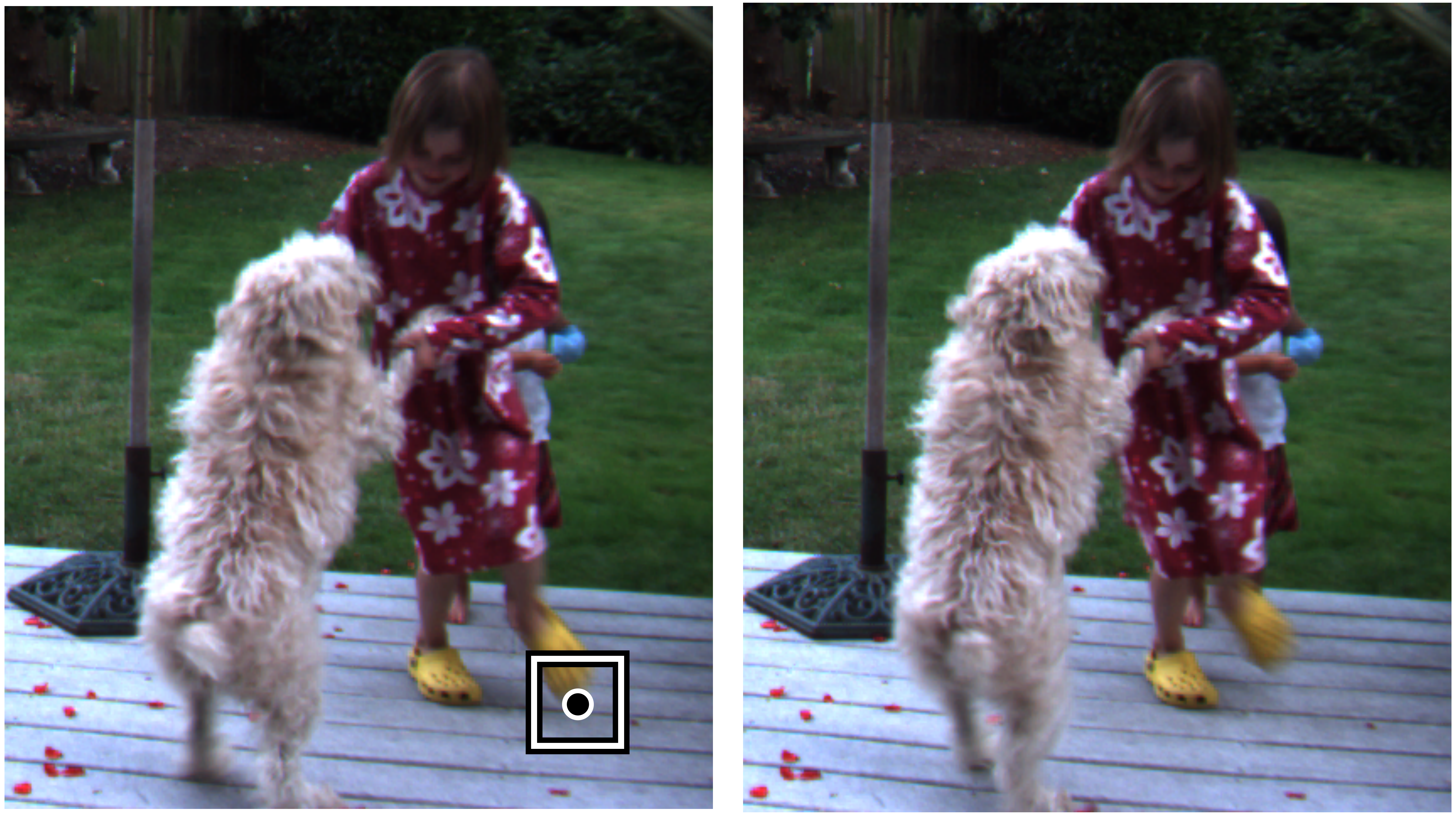
A point in image 1 gives a **line** in image 2

2-view Rigid Matching

- **1D search**, points constrained to lie along epipolar lines


2-view Non-Rigid Matching

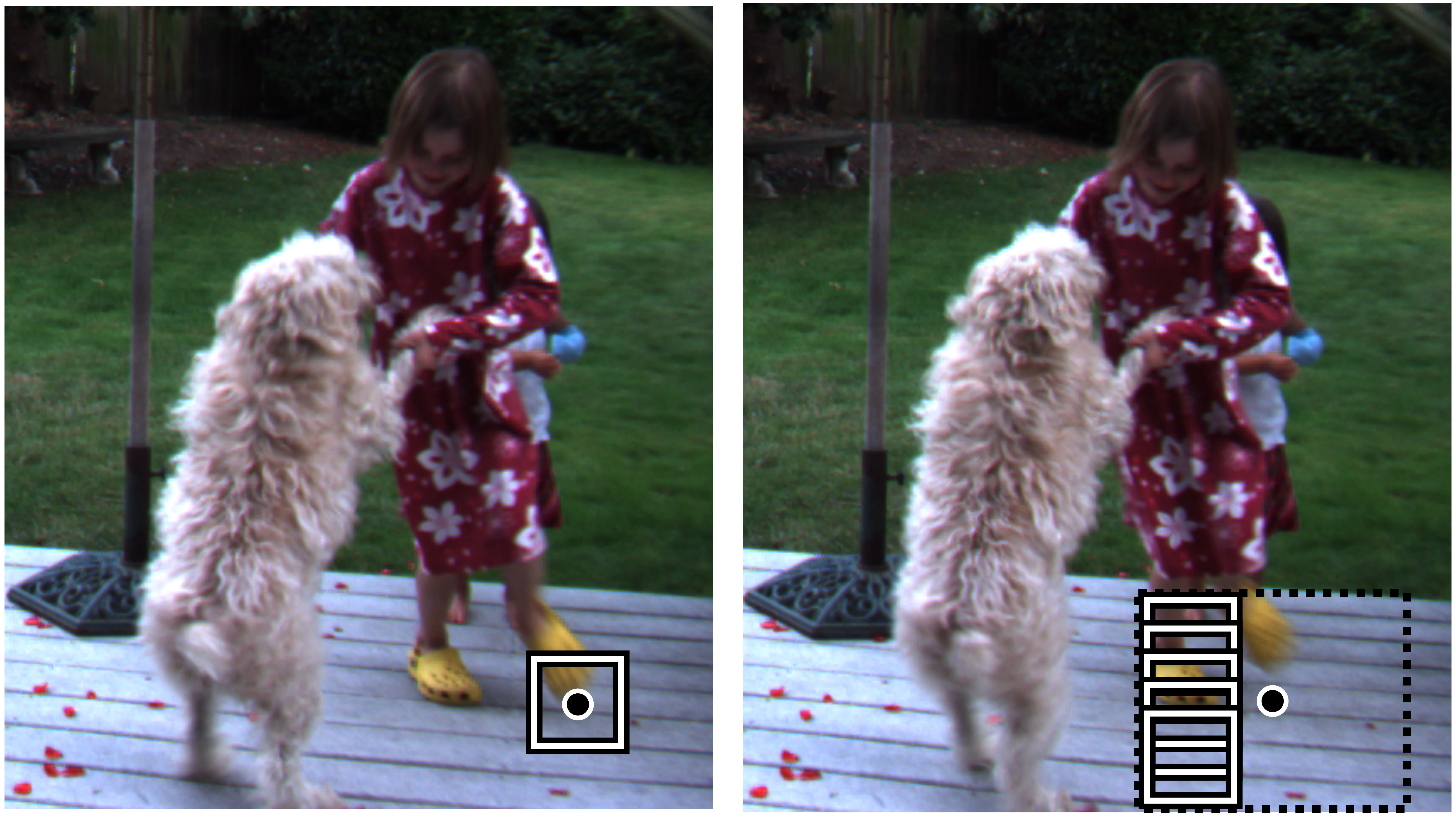
- **2D search**, points can move anywhere in the image



[vision.middlebury.edu/flow]

2-view Non-Rigid Matching

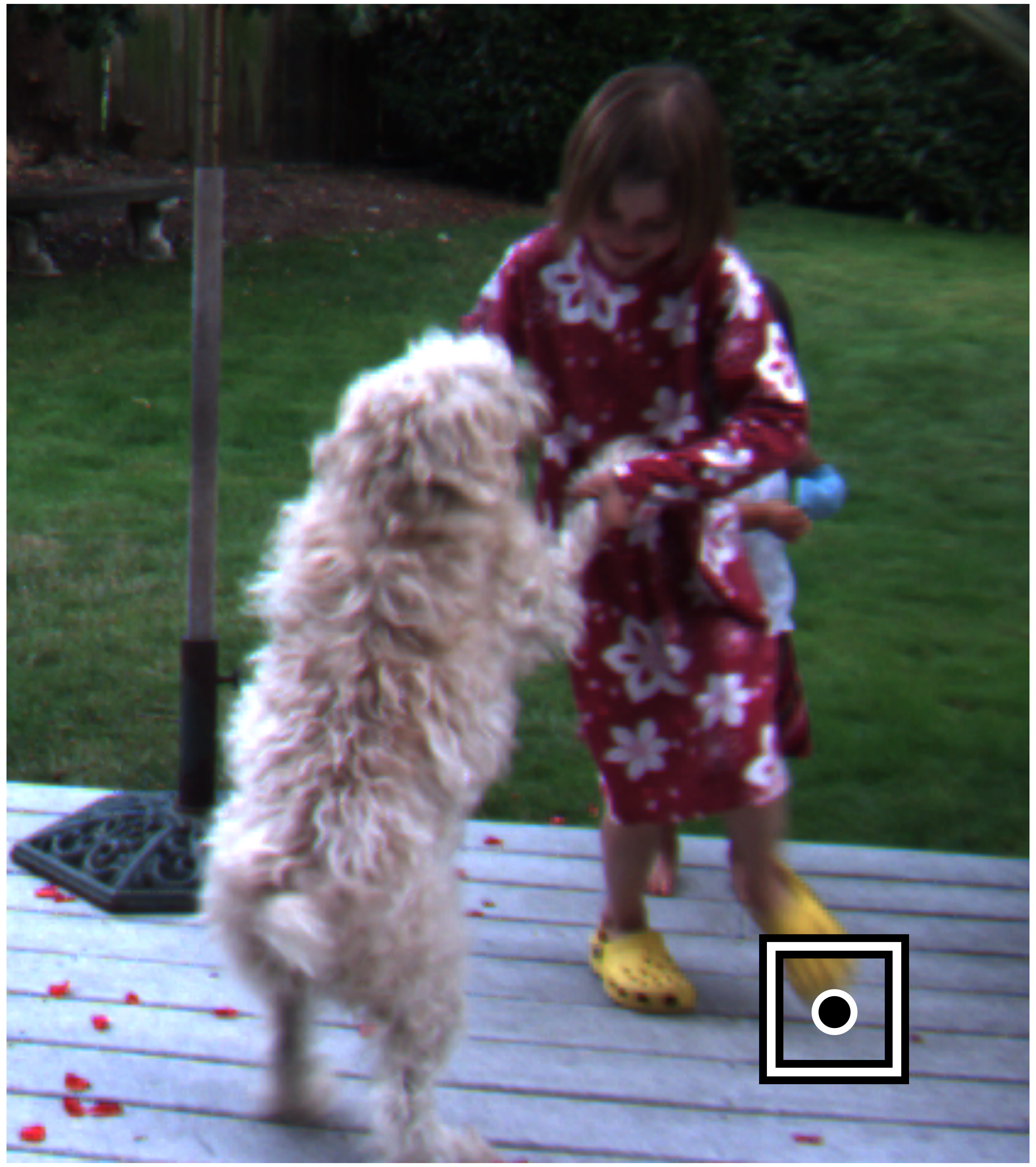
- **2D search**, points can move anywhere in the image



[vision.middlebury.edu/flow]

2-view Non-Rigid Matching

- **2D search**, points can move anywhere in the image

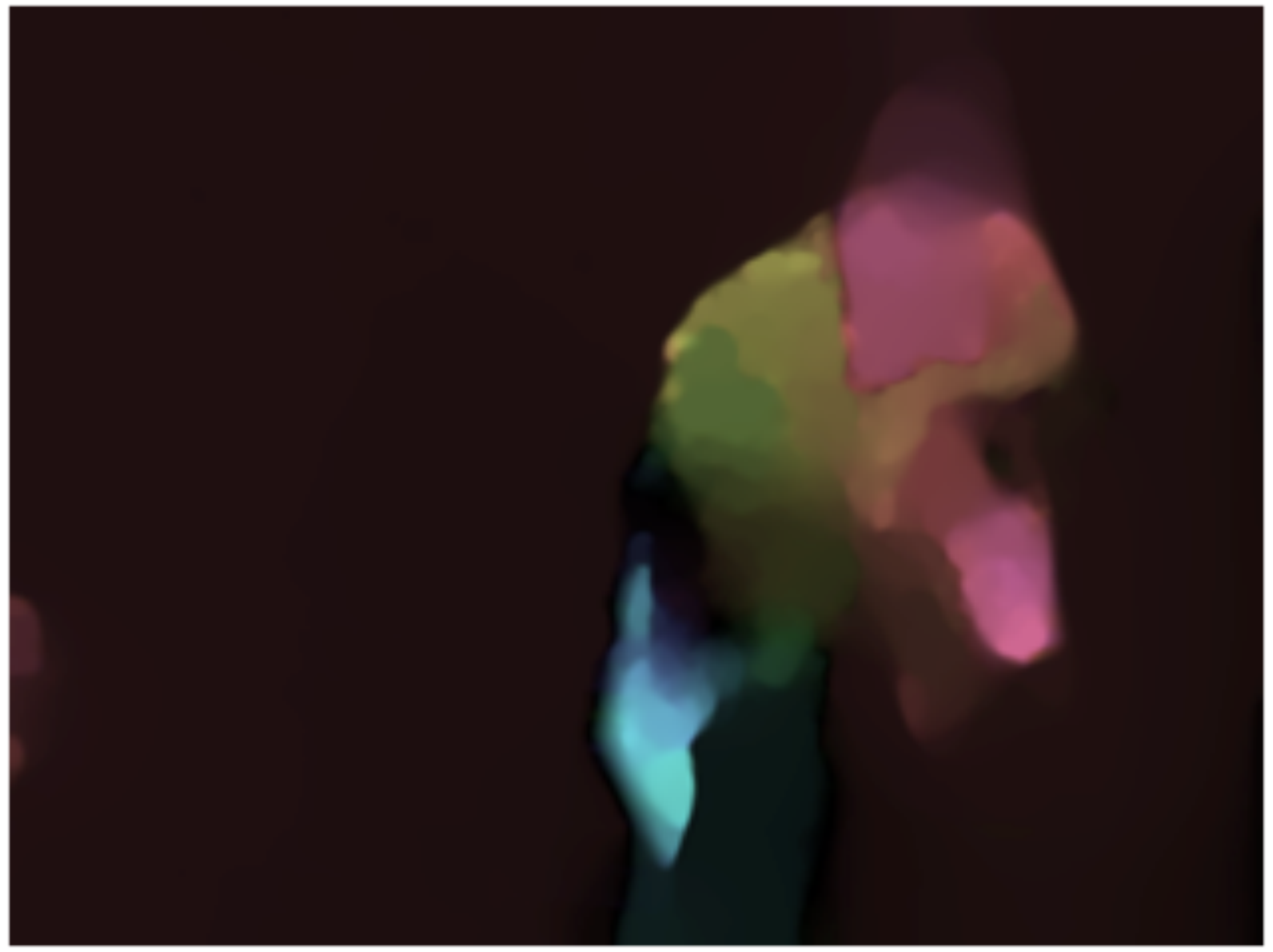
[vision.middlebury.edu/flow]

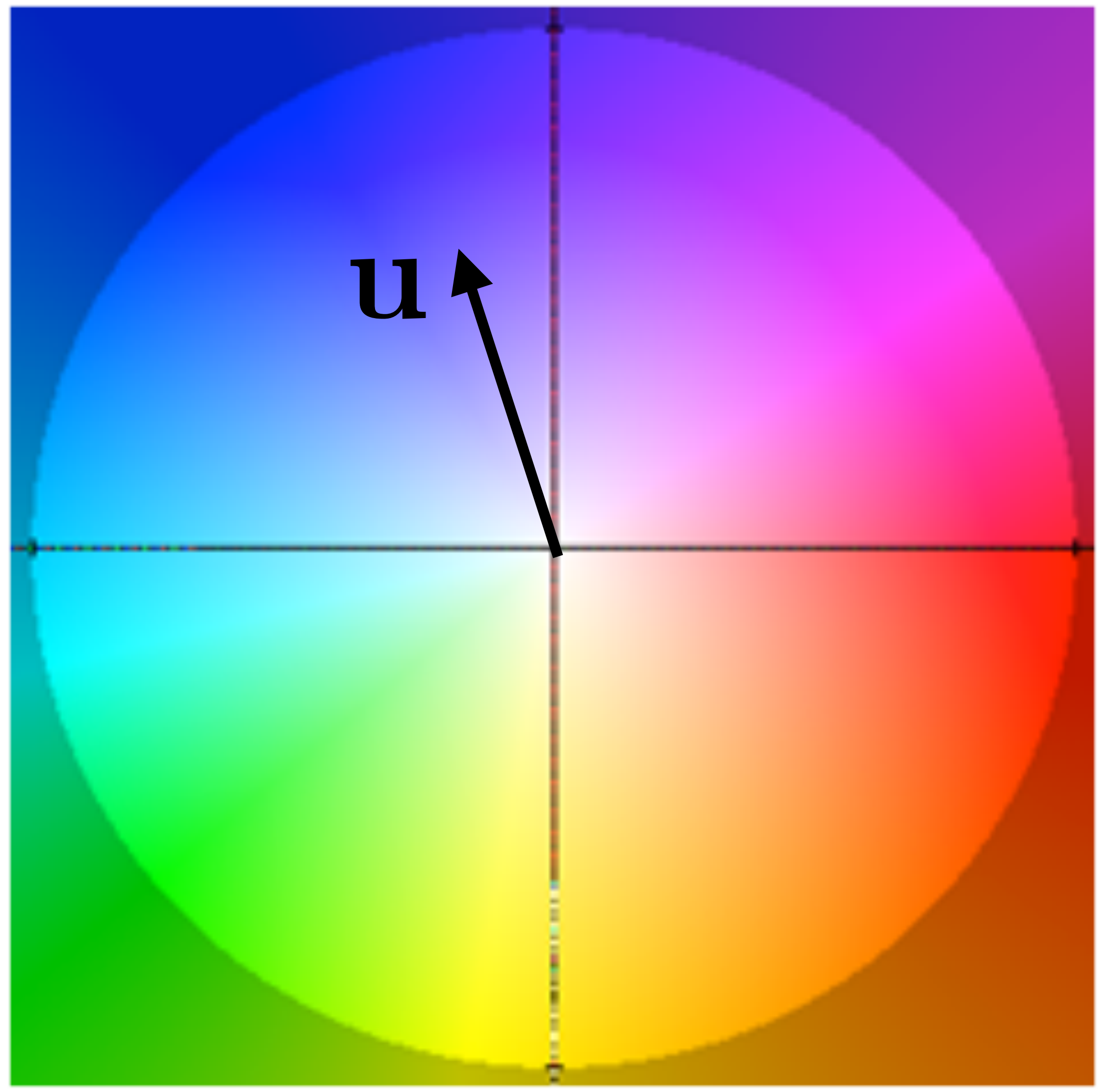
2-view Non-Rigid Matching

- **2D search**, points can move anywhere in the image

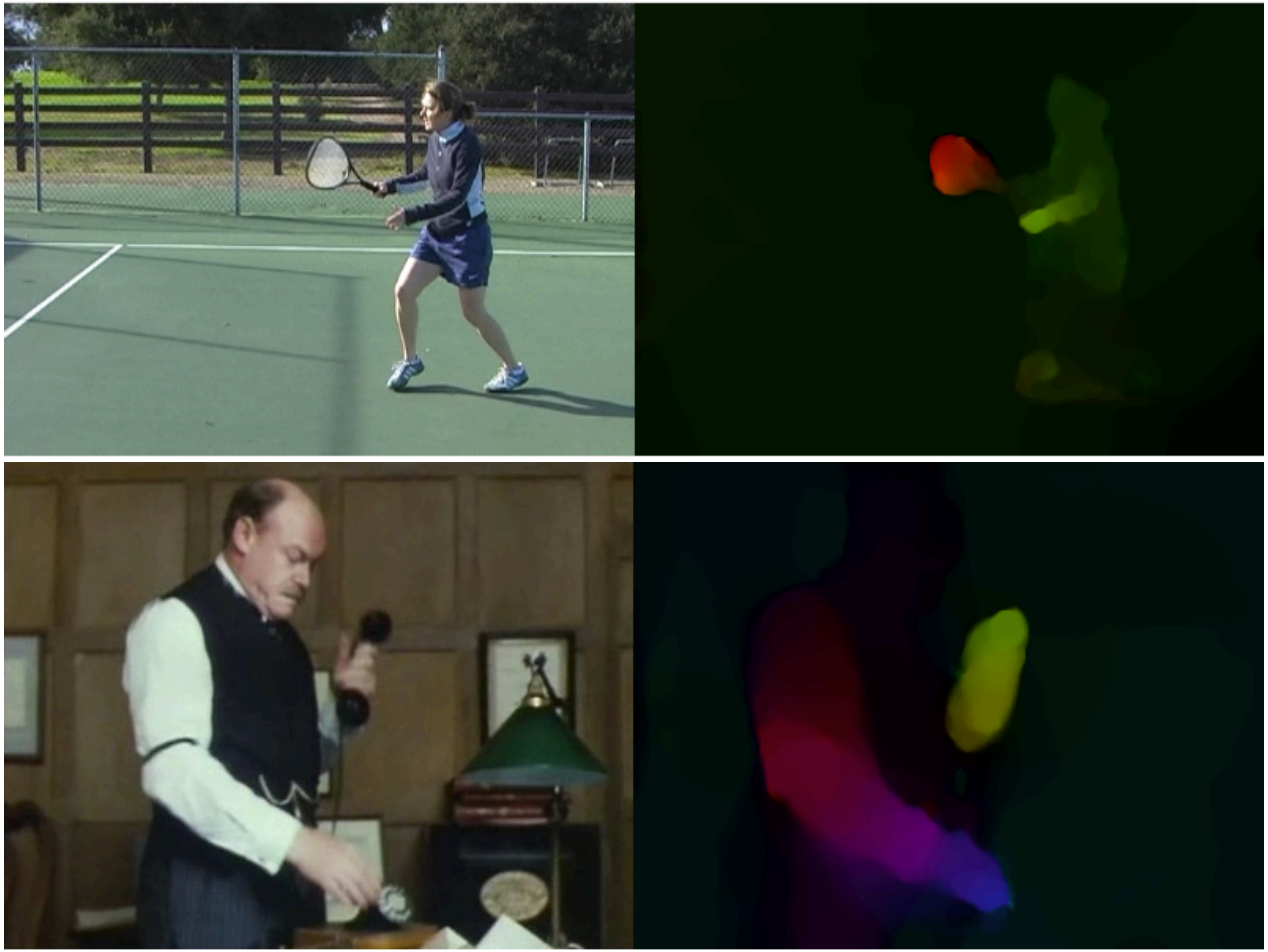
[vision.middlebury.edu/flow]

Optical Flow: Example 1





Optical Flow: Example 2



[Brox Malik 2011]

Optical Flow

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

Problem:

Determine how objects (and/or the camera itself) move in the 3D world.
Formulate motion analysis as finding (dense) point correspondences over time.

Applications

— image and video stabilization in digital cameras, camcorders

— motion-compensated video compression schemes such as MPEG

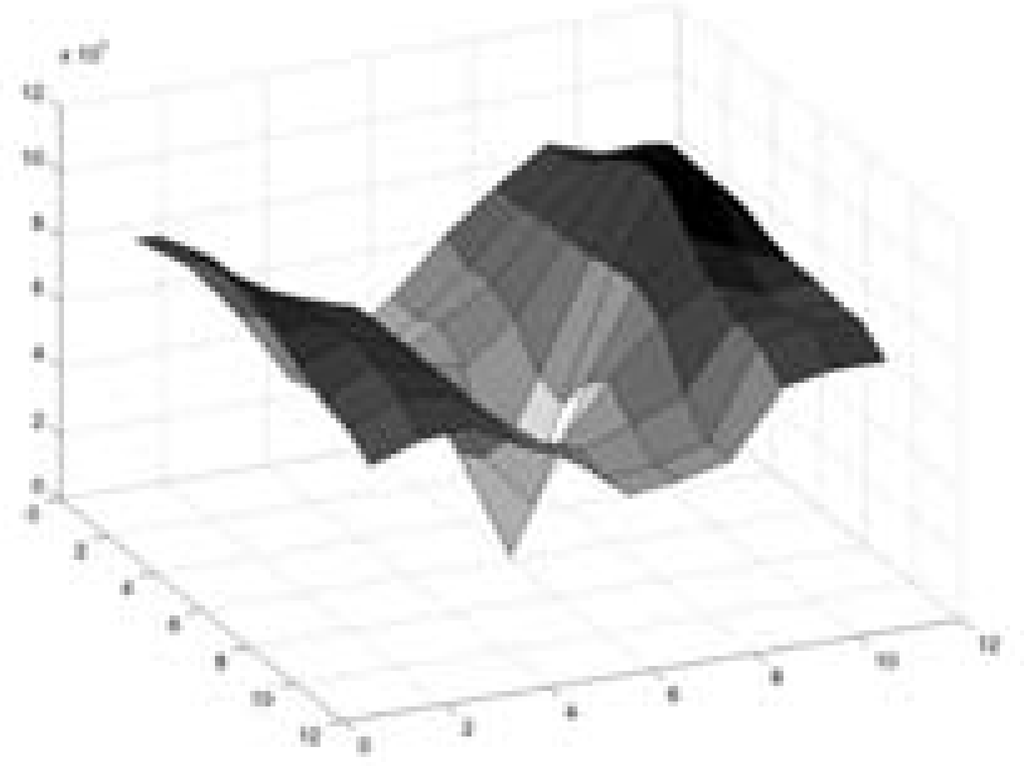
— image registration for medical imaging, remote sensing

Lucas Kanade

- The previous algorithm suggested a discrete search over displacements/flow vectors **u**
- We can do better by looking at the structure of the error surface:

$I_0(\mathbf{x})$

$I_1(\mathbf{x})$



$$e = |\mathbf{I}_1(\mathbf{x}+\mathbf{u}) - \mathbf{I}_0(\mathbf{x})|^2$$

15.1

Flow at a pixel

- Look at previous equation at a single pixel:

$$\frac{\partial I_1}{\partial \mathbf{x}}^T \Delta \mathbf{u} = I_0(\mathbf{x}) - I_1(\mathbf{x})$$

15.2

Optical Flow in 1D

- Consider a 1D function moving at velocity v

15.3

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

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
…

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Frame Differencing: Example

$t+1$

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

-

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

=

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

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

(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
…

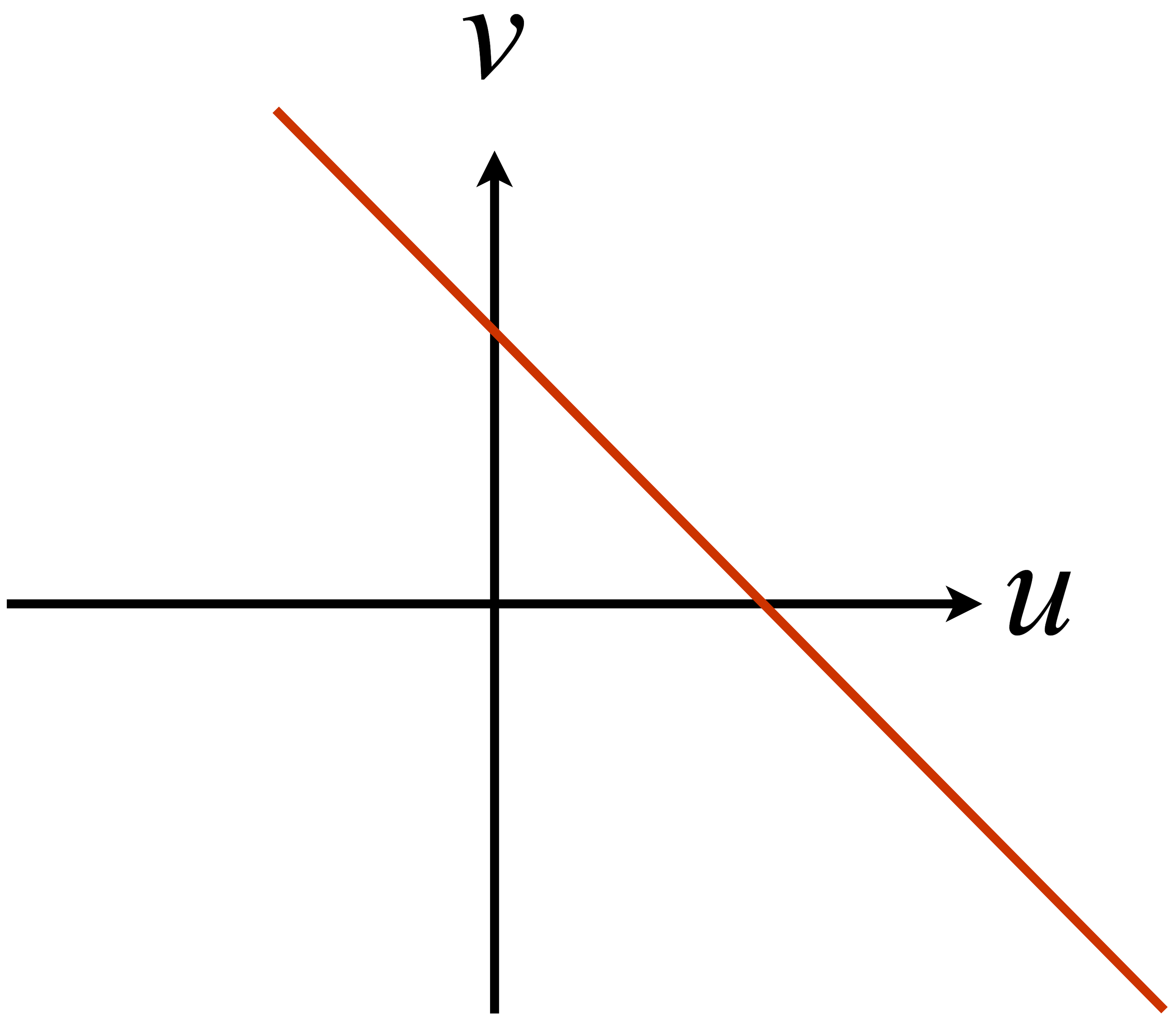
How do we solve for u and v?

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Optical Flow **Constraint Equation**

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



We have one equation in the two unknown components of velocity u, v

Many possible solutions for u, v — need more constraints or prior knowledge to solve

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

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

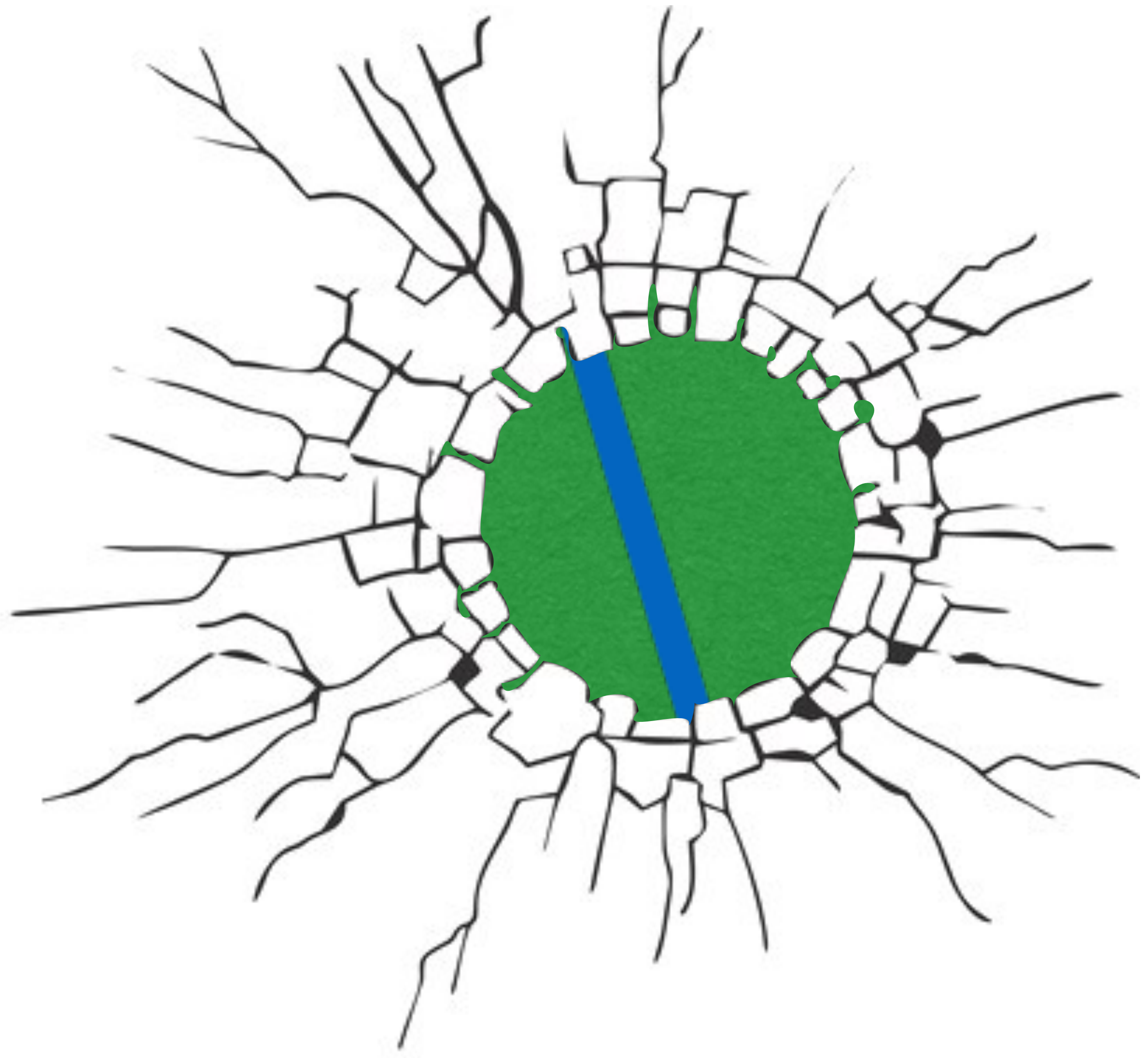
Flow Ambiguity

- Optical Flow Constraint:

$$\frac{\partial I}{\partial t} + \nabla I^T \mathbf{v} = 0$$

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

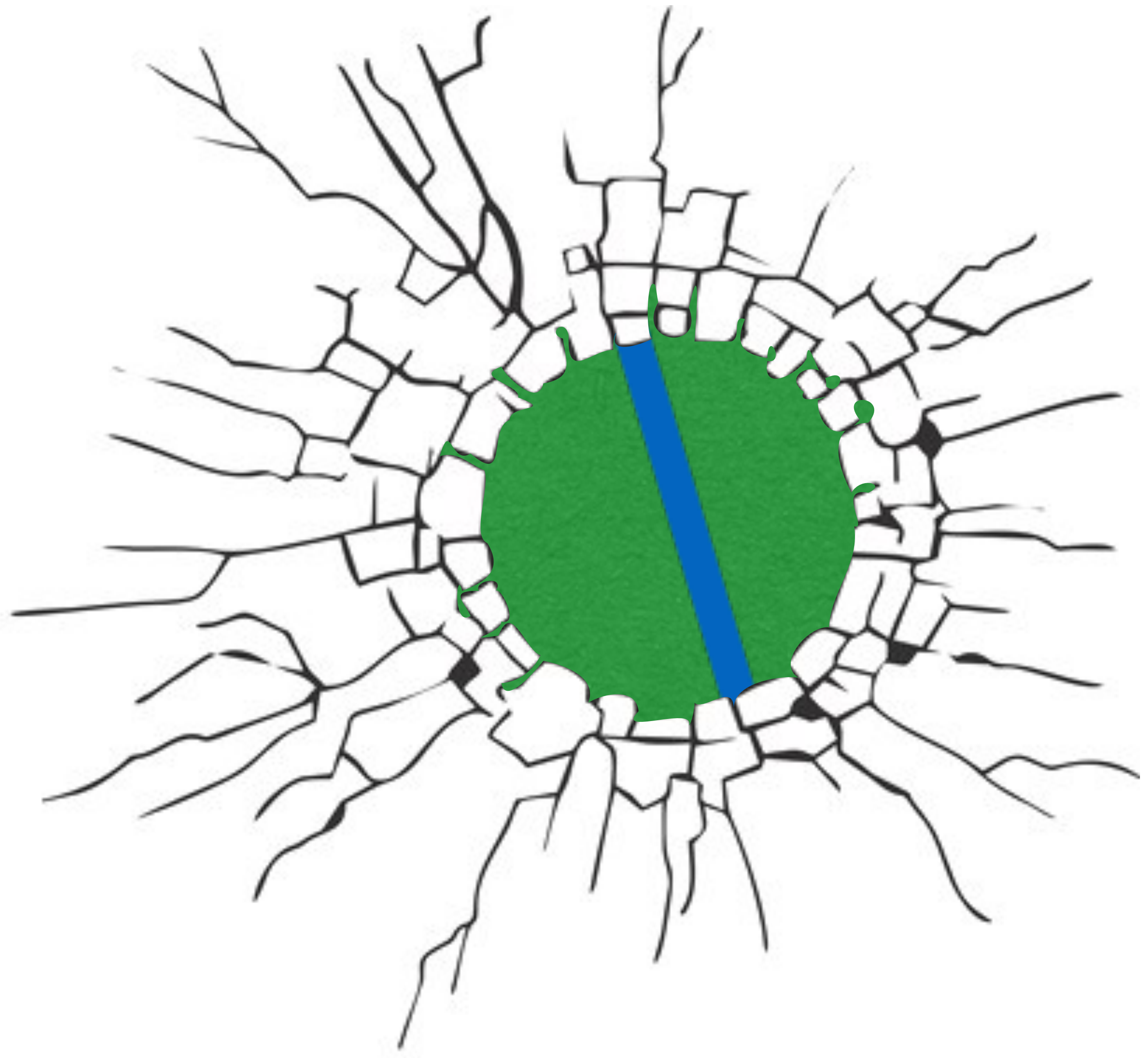
Aperture Problem



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

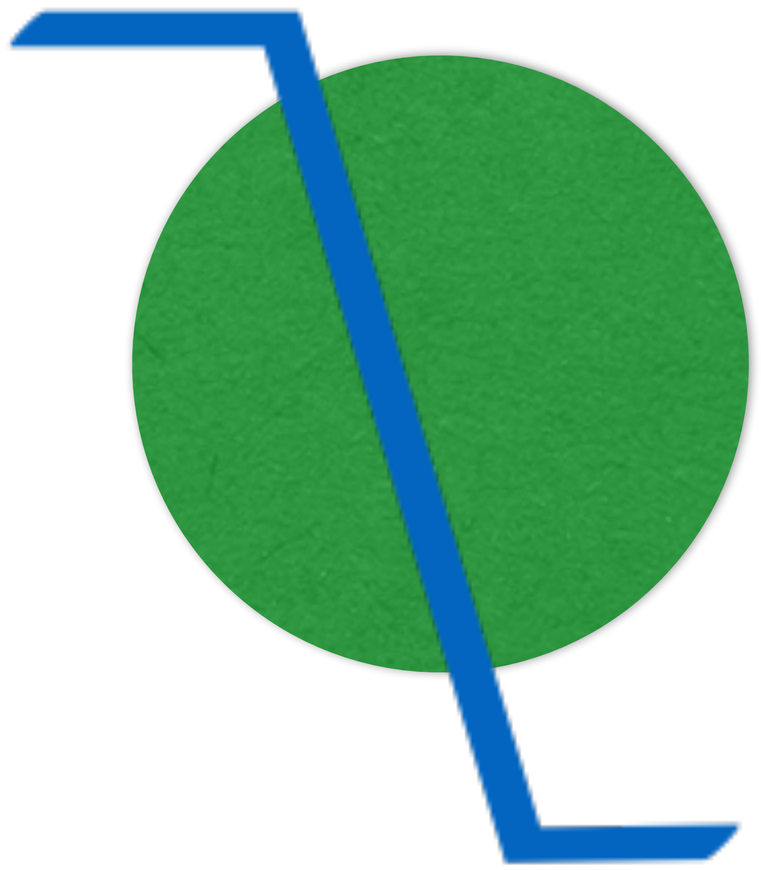


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

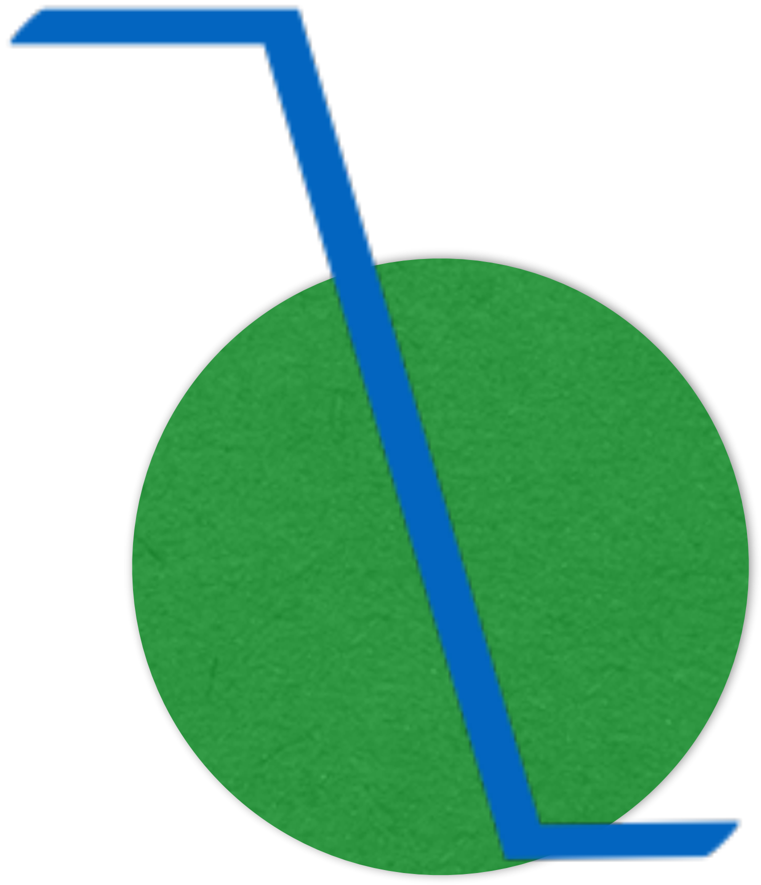


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

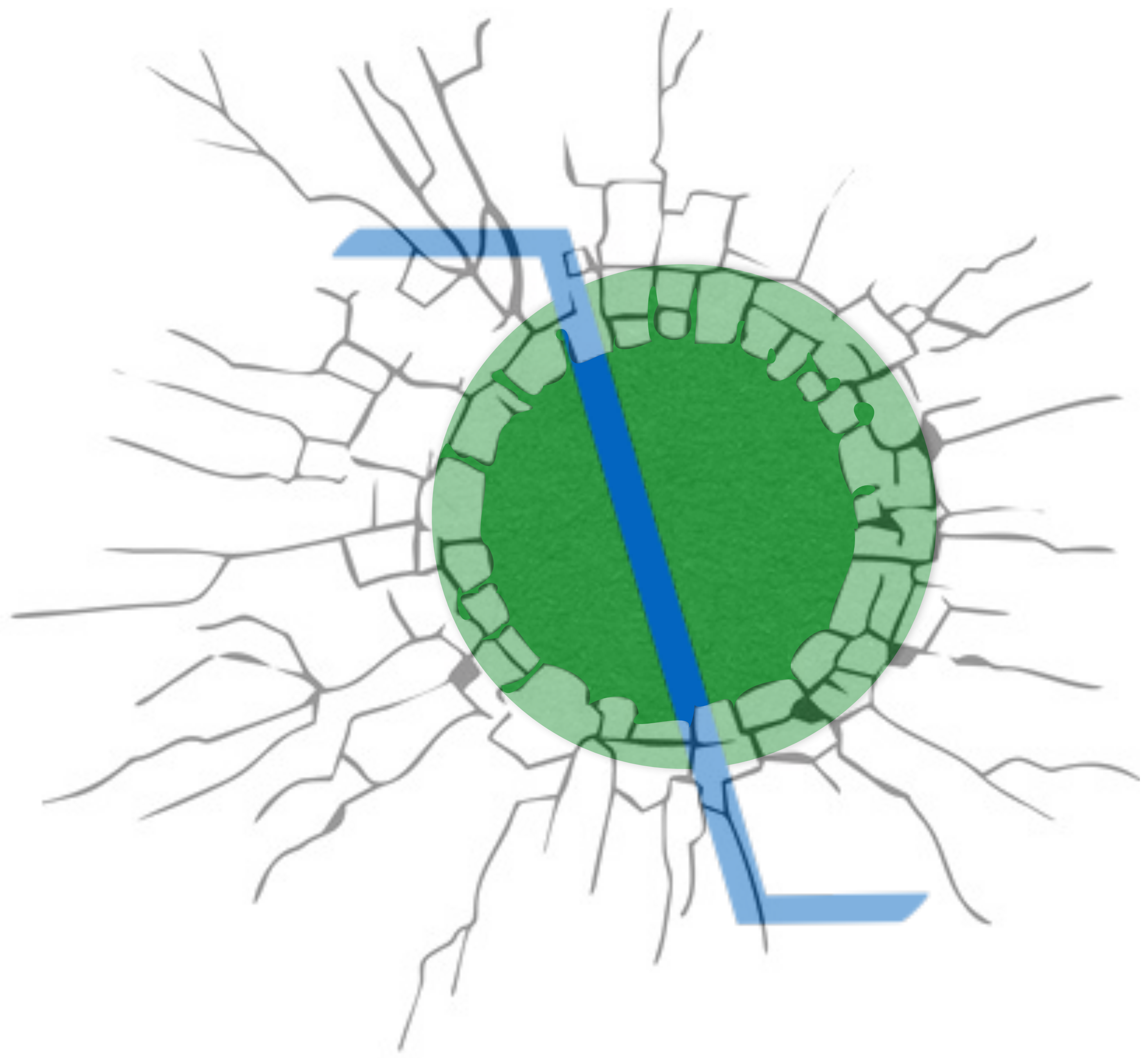


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Aperture Problem

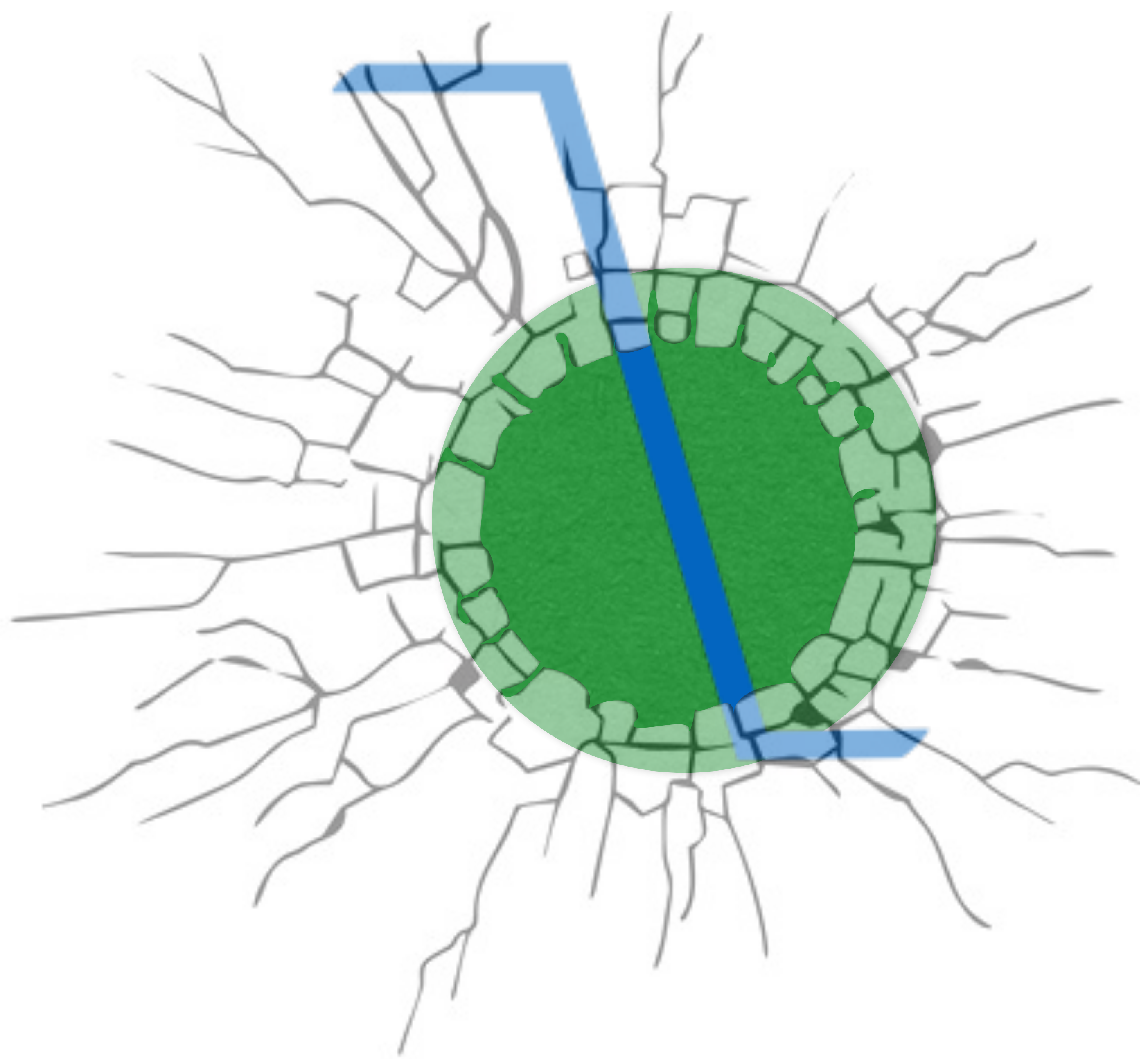
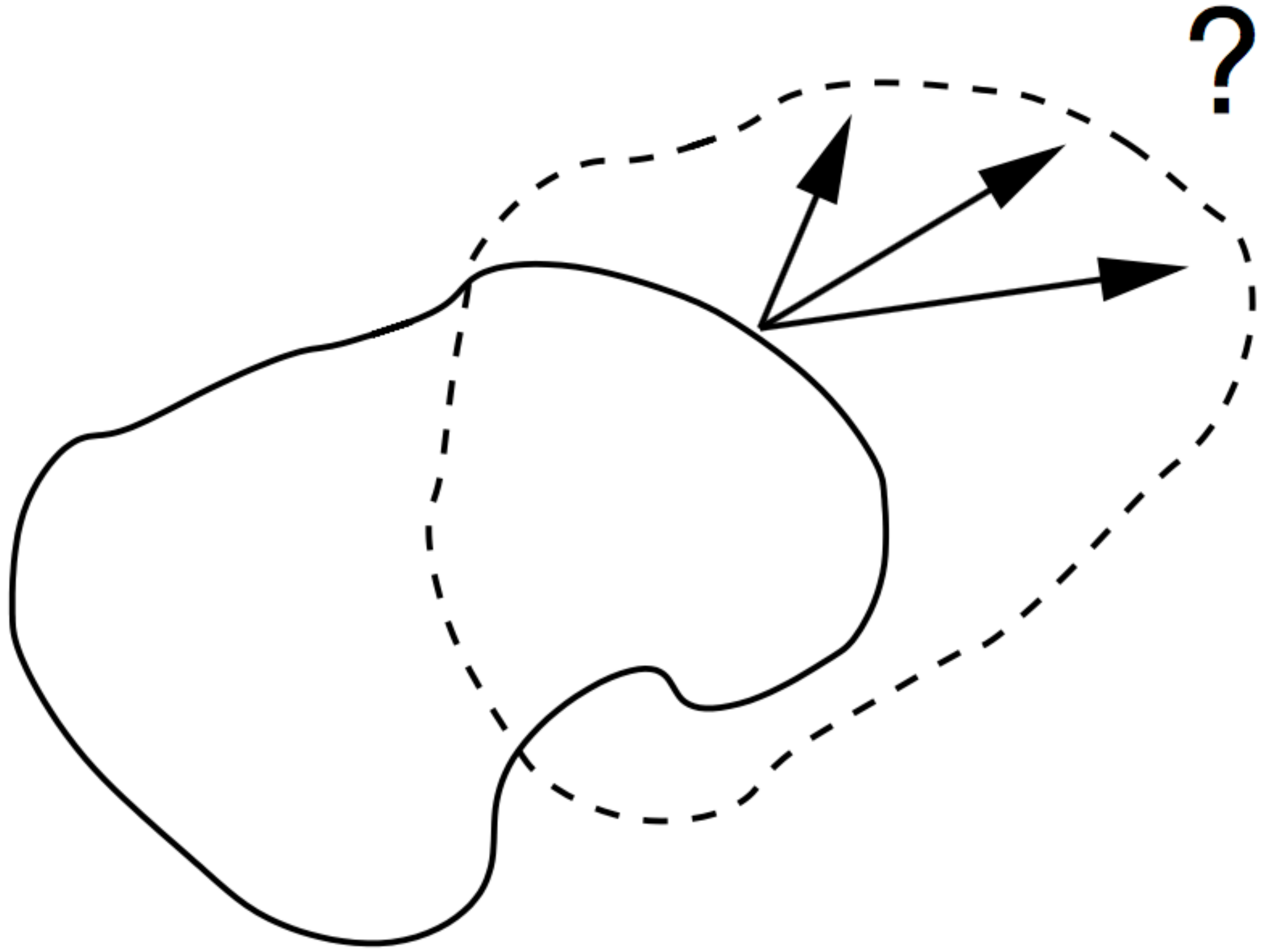


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

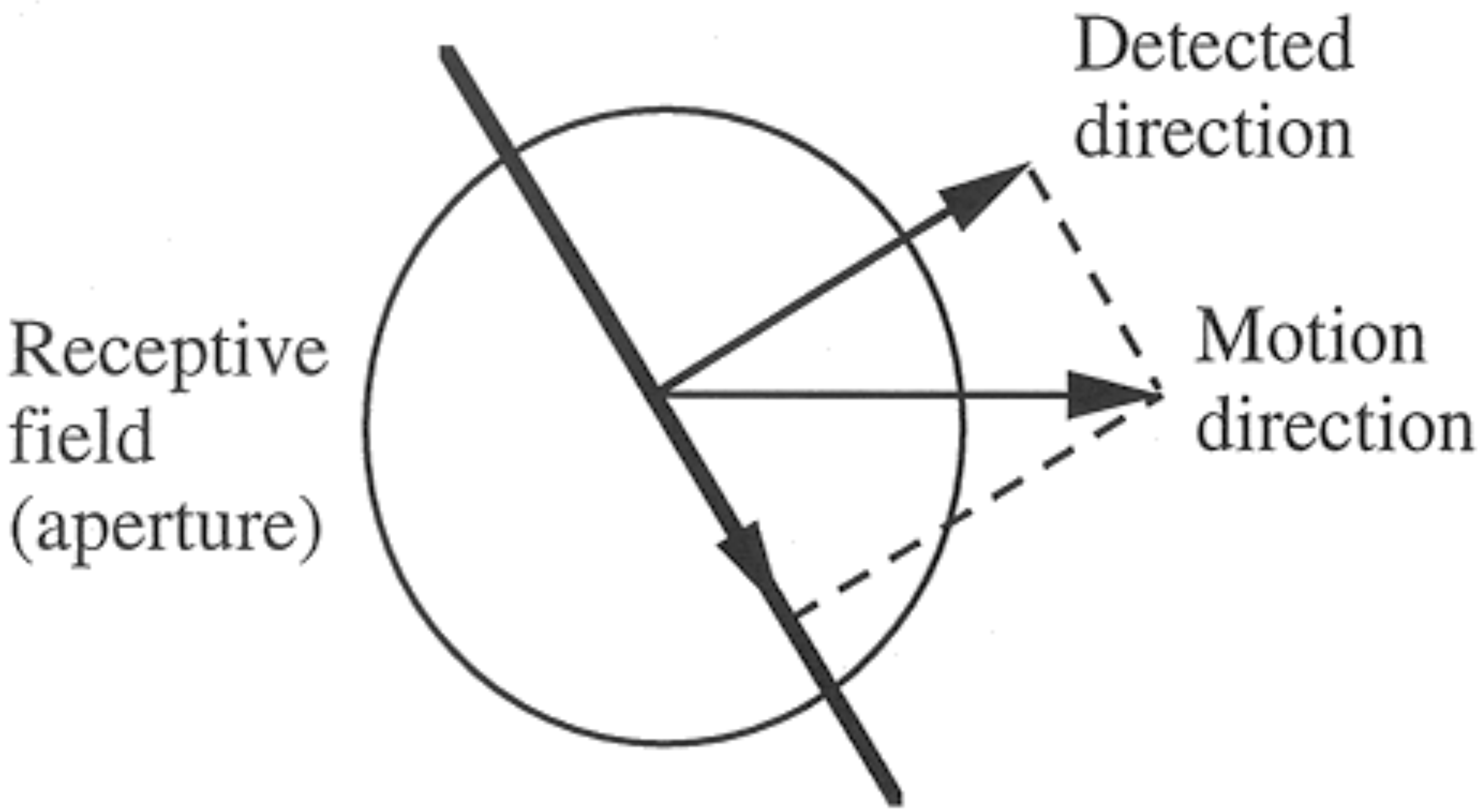
Aperture Problem



— 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

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 is

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

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

$$\mathbf{A}^T\mathbf{A} = \left[\begin{array}{cc} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & I_y^2 \end{array} \right]$$

Where have we seen this before?

Can this tell us something about where LK is likely to work well?

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

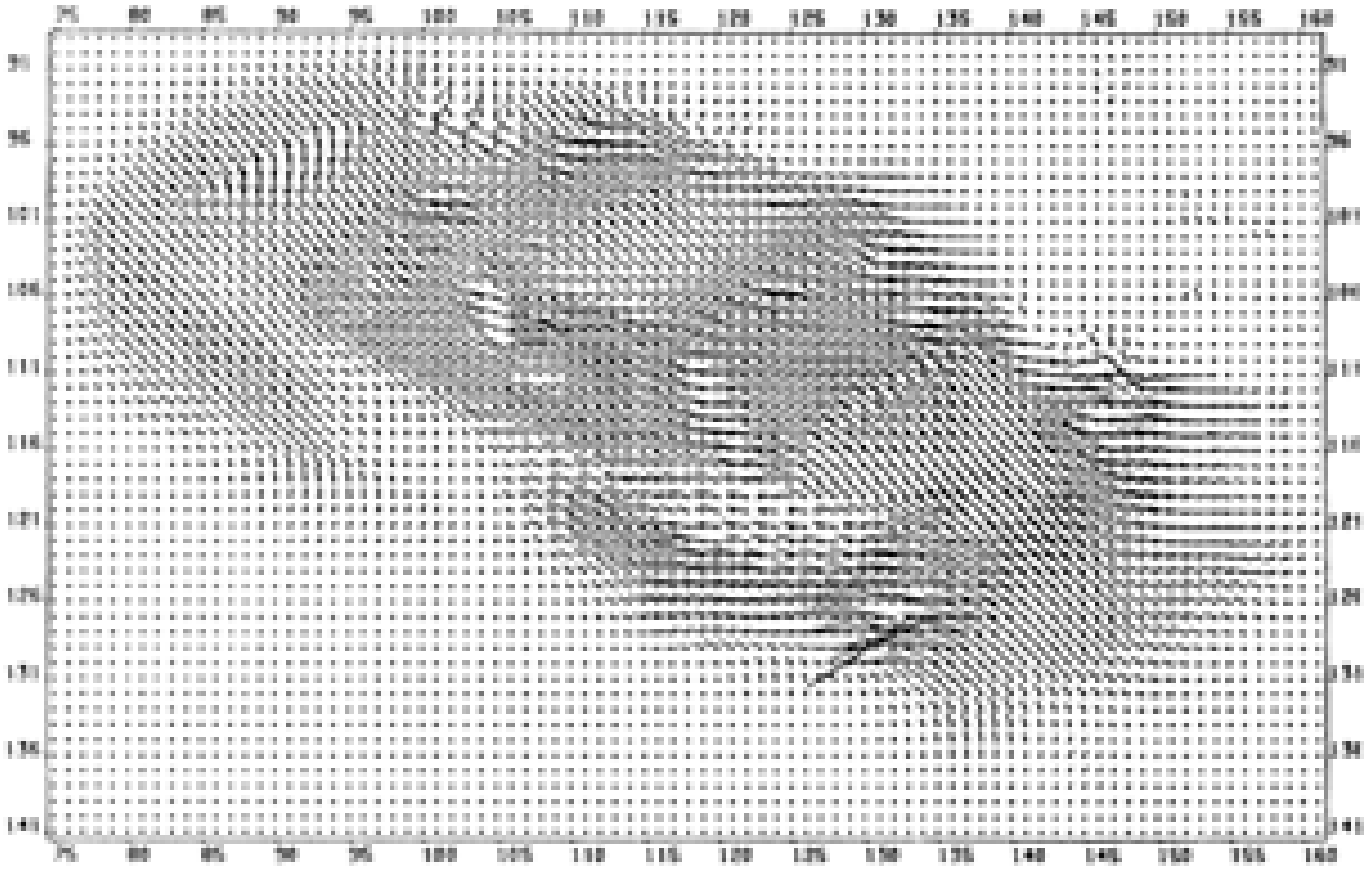
4. Windows are chosen s.t. that the rank of $\mathbf{A}^T\mathbf{A}$ is 2

Optical Flow **Smoothness Priors**

The optical flow equation gives **one constraint per pixel,** but we need to solve for 2 parameters u, v

Lucas Kanade adds constraints by **adding more pixels**

An alternative approach is to make assumptions about the **smoothness of the flow field**, e.g., that there should not be abrupt changes in flow



Optical Flow **Smoothness Priors**

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

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

e.g., the Horn Schunck objective function penalises the magnitude of velocity:

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

[Horn Schunck 1981, Szeliski p395]

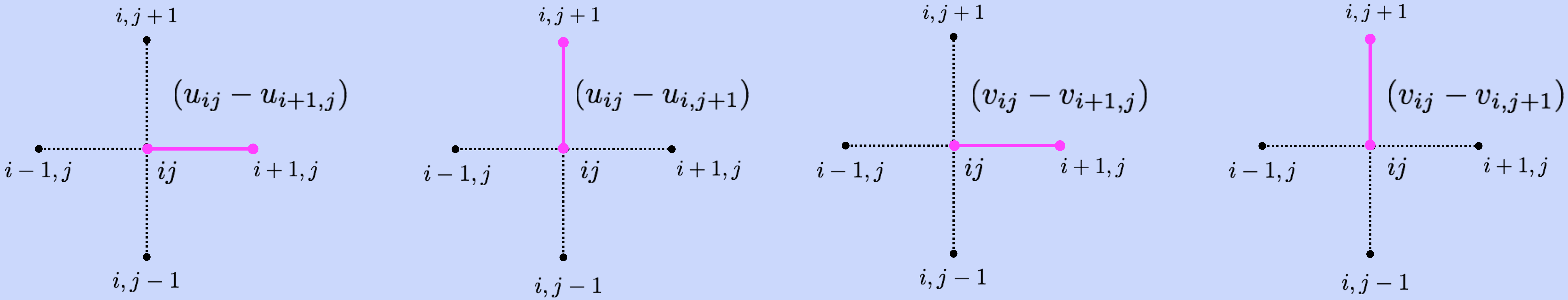
Horn-Schunck Optical Flow

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)

Brightness Constancy

- All the methods presented in this lecture have relied on the assumption that

$$I_1(\mathbf{x} + \mathbf{u}) \approx I_0(\mathbf{x})$$

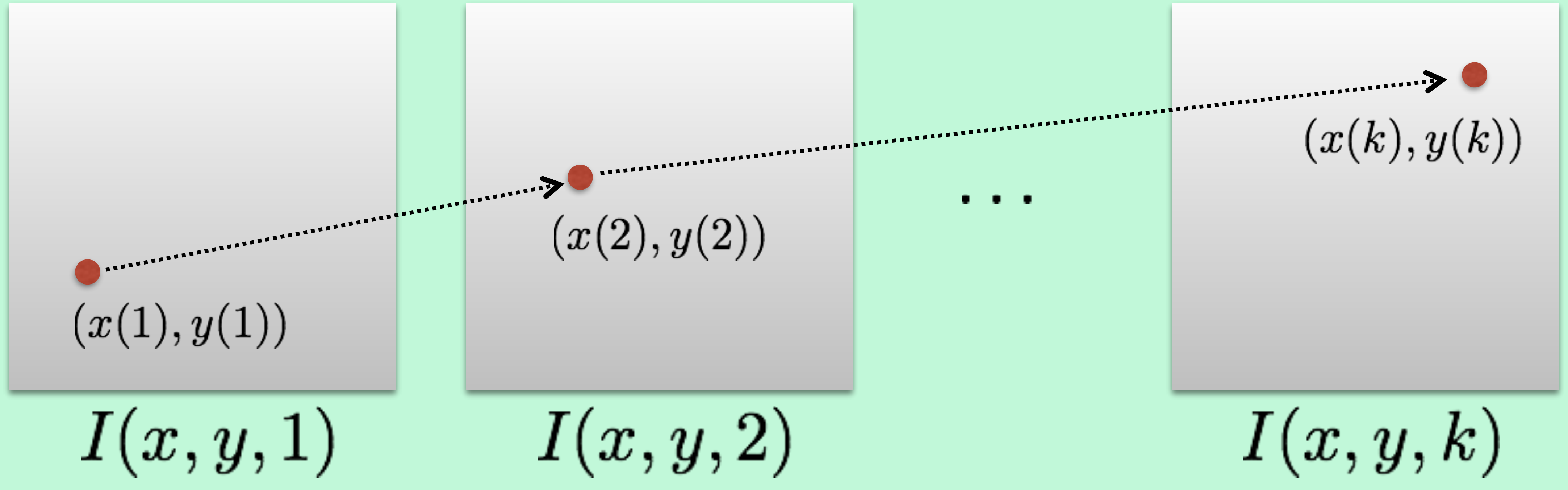
- This is called the **brightness constancy** assumption
- Taylor expansion for small motion at a single pixel → optical flow constraint

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

- Horn-Schunk = optical flow constraint + smoothing over **u**
- Lucas-Kanade = optical flow constraint over patches assuming **u** is constant/slowly varying over patch

Optical Flow **Constraint Equation**

Brightness Constancy Assumption: Brightness of the point remains the same



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

constant

15.4 What does this mean, and why is it reasonable?

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

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

Image 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

Menu for Today

Topics:

- **Stereo** recap, 1D vs 2D motion
- **Optical Flow**
- **Brightness** Constancy
- **Lucas Kanade**

Readings:

- **Today's** Lecture: Szeliski 12.1, 12.3-12.4, 9.3

Reminders:

- **Midterm results** will be available very soon (likely tomorrow)
- **Assignment 4**: RANSAC and Panoramas due **November 9th**