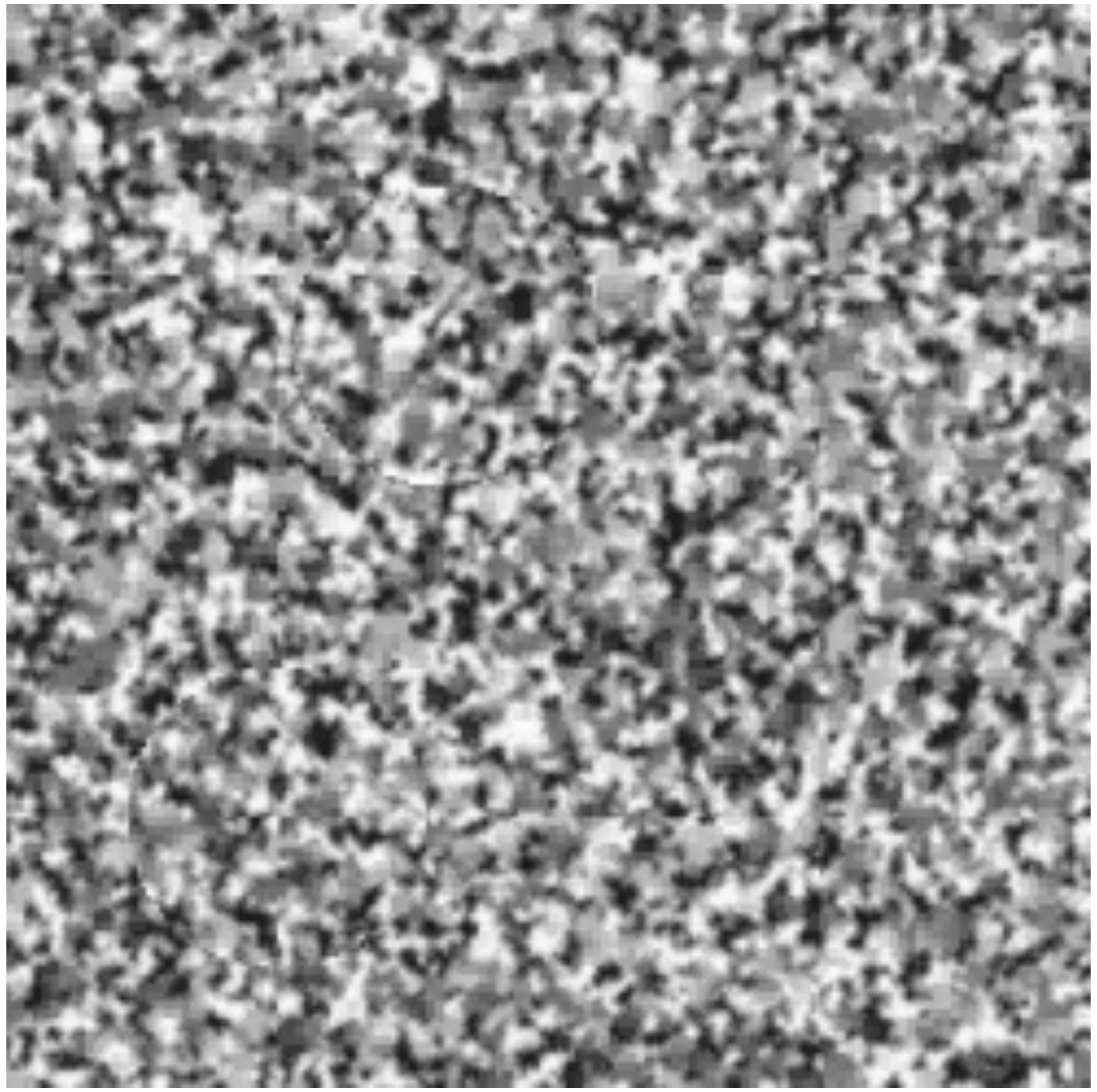
Lecture 23: Motion estimation

Most slides from S. Lazebnik, which are based on other slides from S. Seitz, R. Szeliski, M. Pollefeys

Announcements

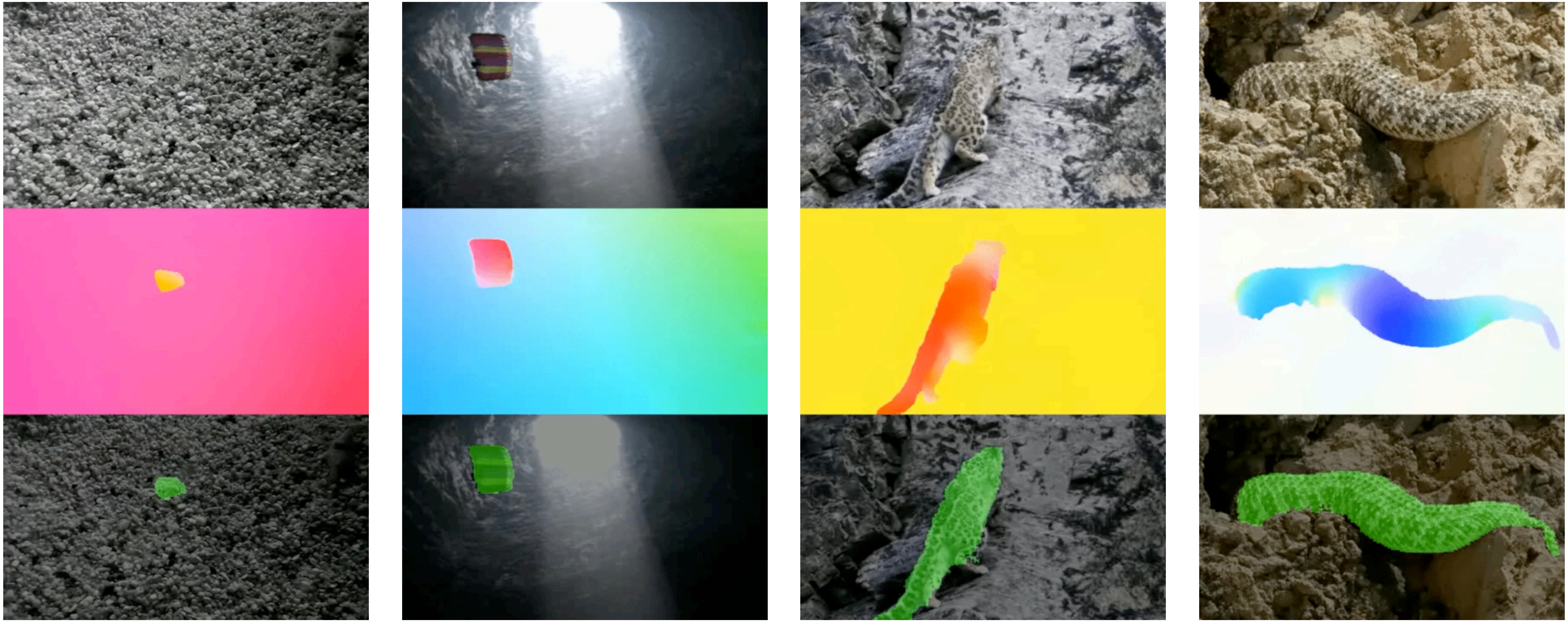
- One-day extension for PS7 due to download issues
- PS8: panorama stitching
- Discussion section: project office hours + geometry

Motion is a powerful perceptual cue



Source: S. Lazebnik

Motion is a powerful perceptual cue



[Yang et al., “Self-supervised Video Object Segmentation by Motion Grouping”, 2021]

Motion is a powerful perceptual cue

G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis",
Perception and Psychophysics 14, 201-211, 1973.

Source: S. Lazebnik

Optical flow

- **Optical flow** is the *apparent motion* of brightness patterns in the image
  - Can be caused by camera motion, object motion, or changes of lighting in the scene

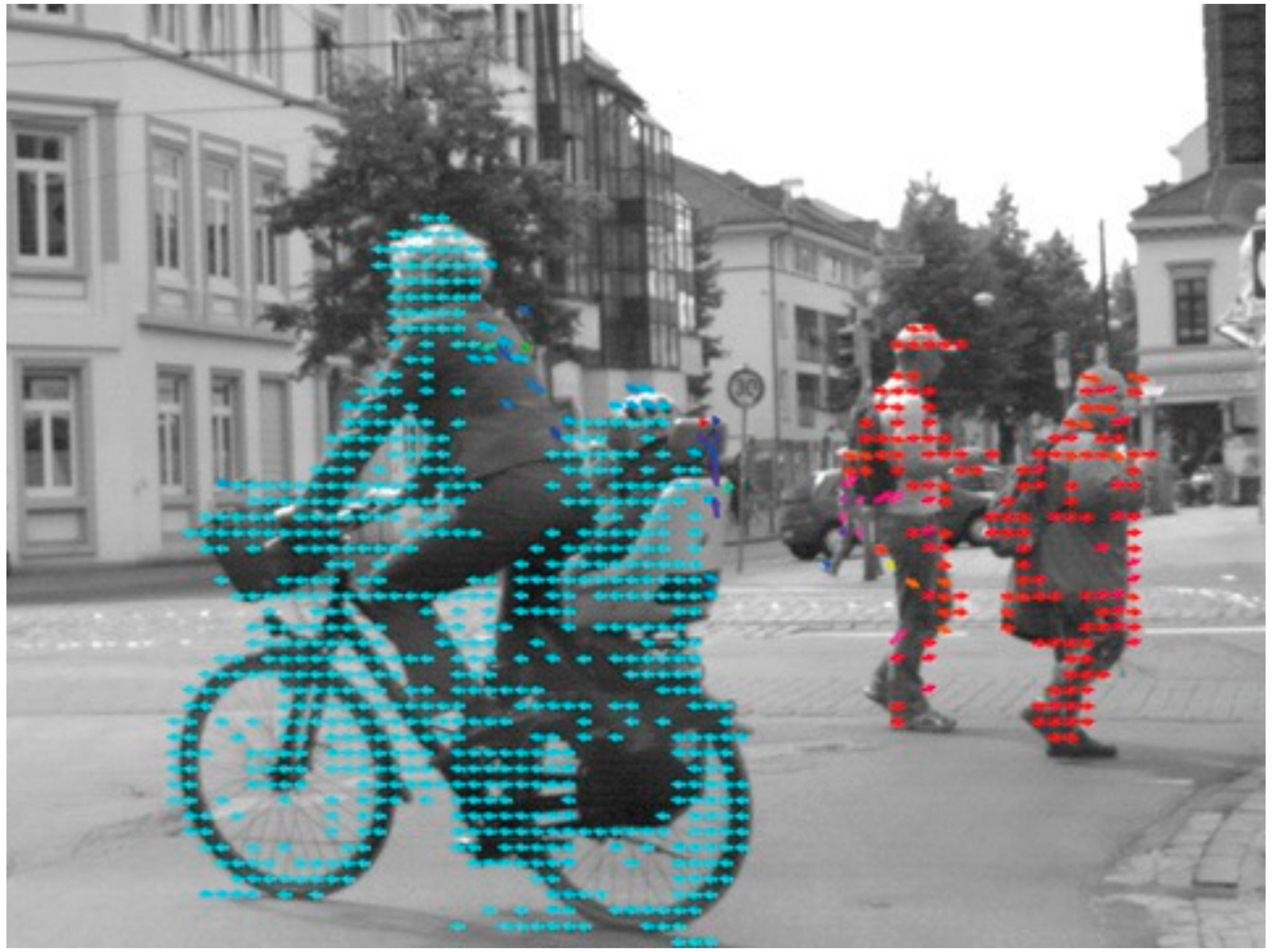


Image source

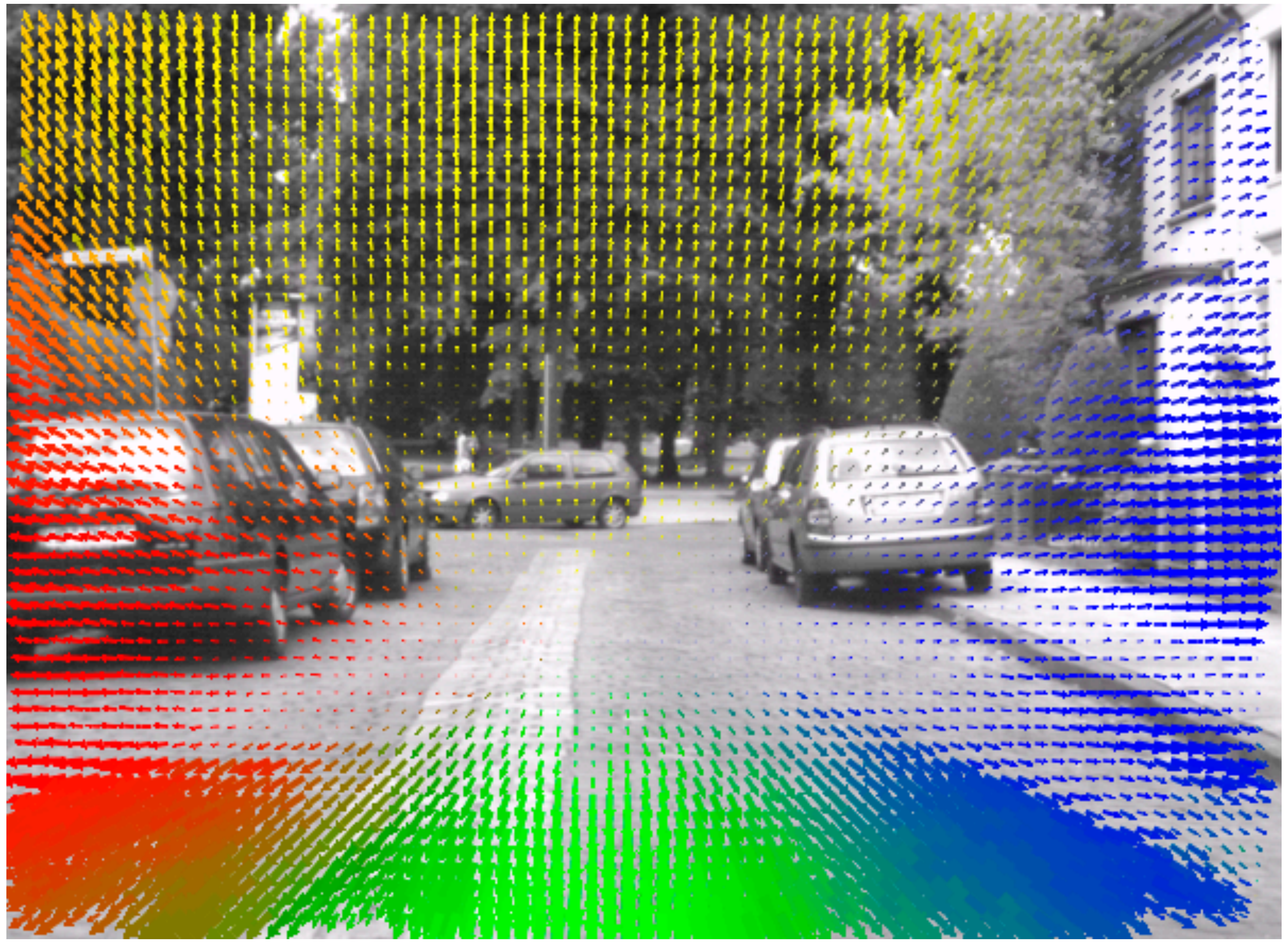
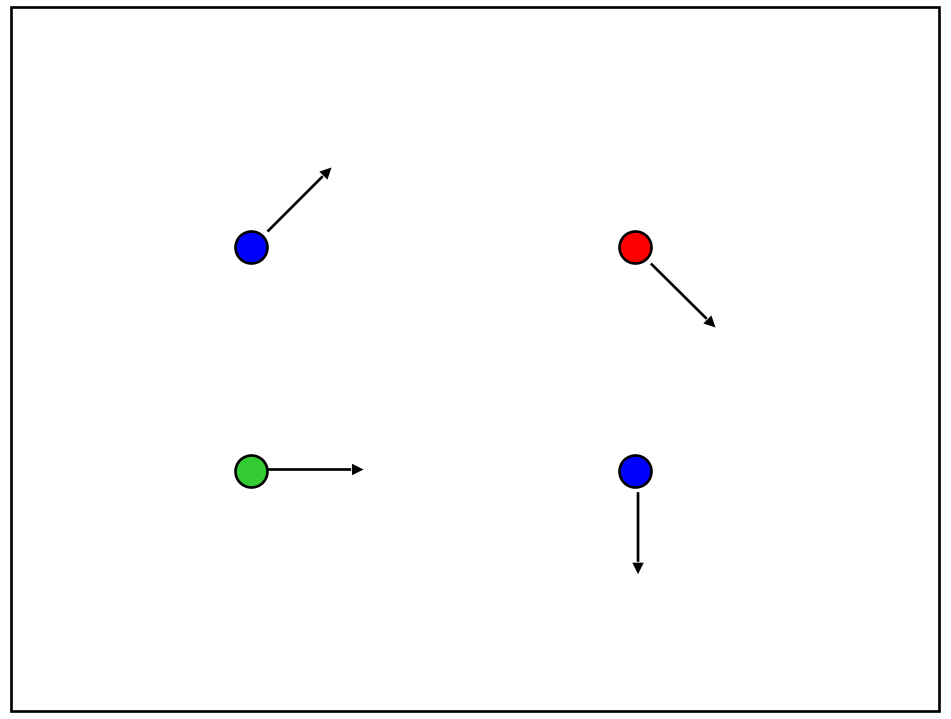


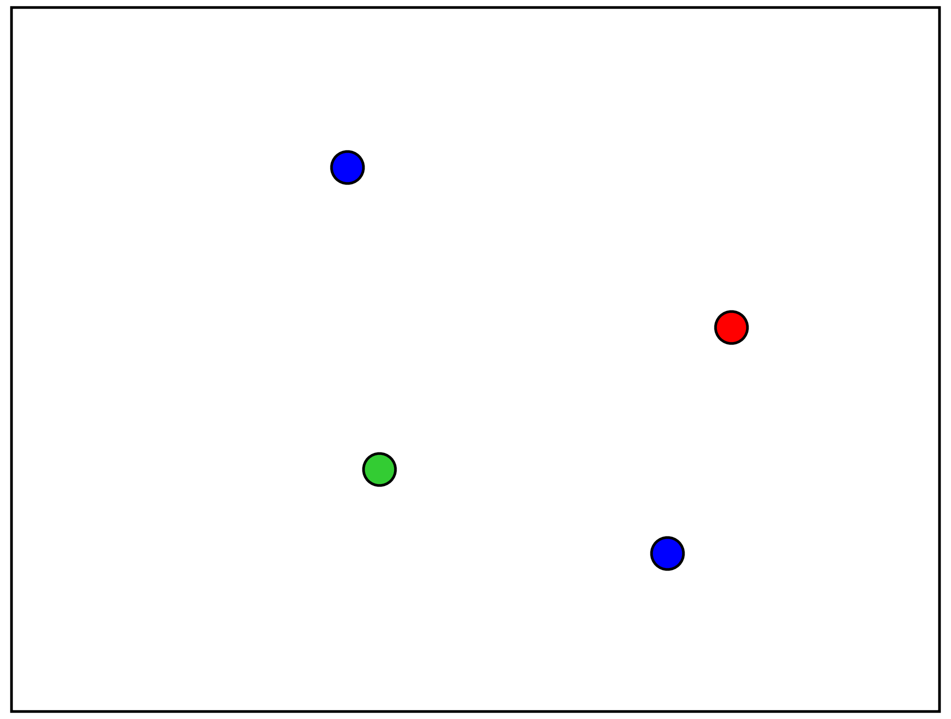
Image source

Source: S. Lazebnik

Estimating optical flow



$I(x,y,t-1)$

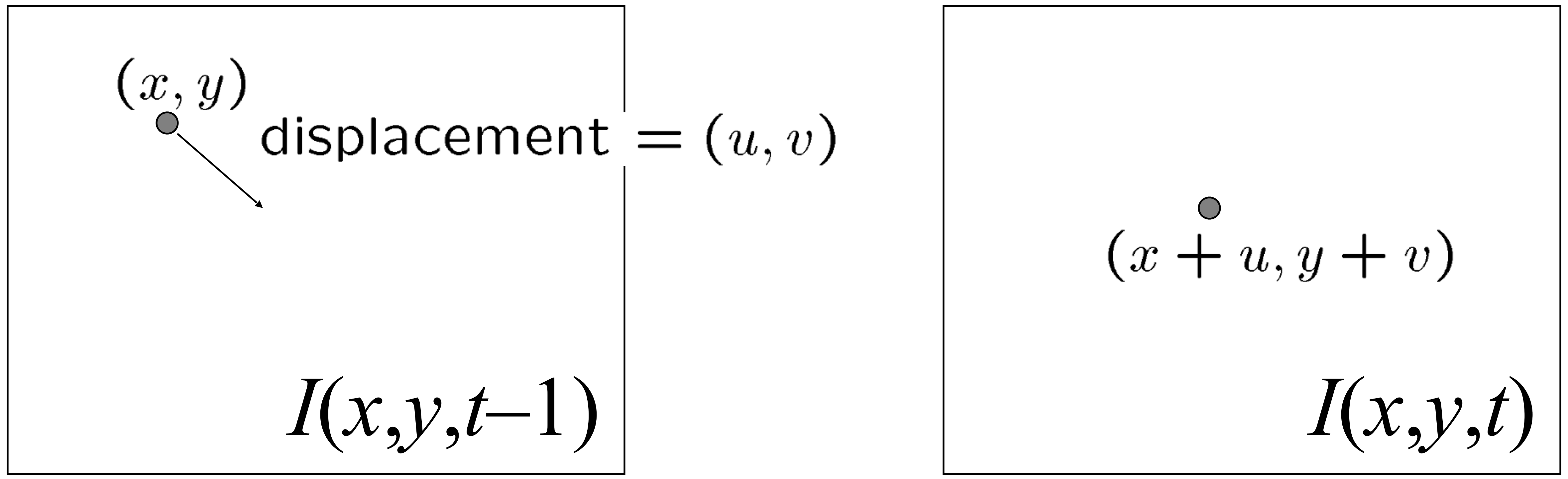


$I(x,y,t)$

- Given two consecutive frames, estimate the motion field u(x,y) and v(x,y) between them
- How can we estimate it? Some assumptions:
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Small motion:** points do not move very far
  - **Spatial coherence:** points move like their neighbors

Source: S. Lazebnik

The brightness constancy constraint

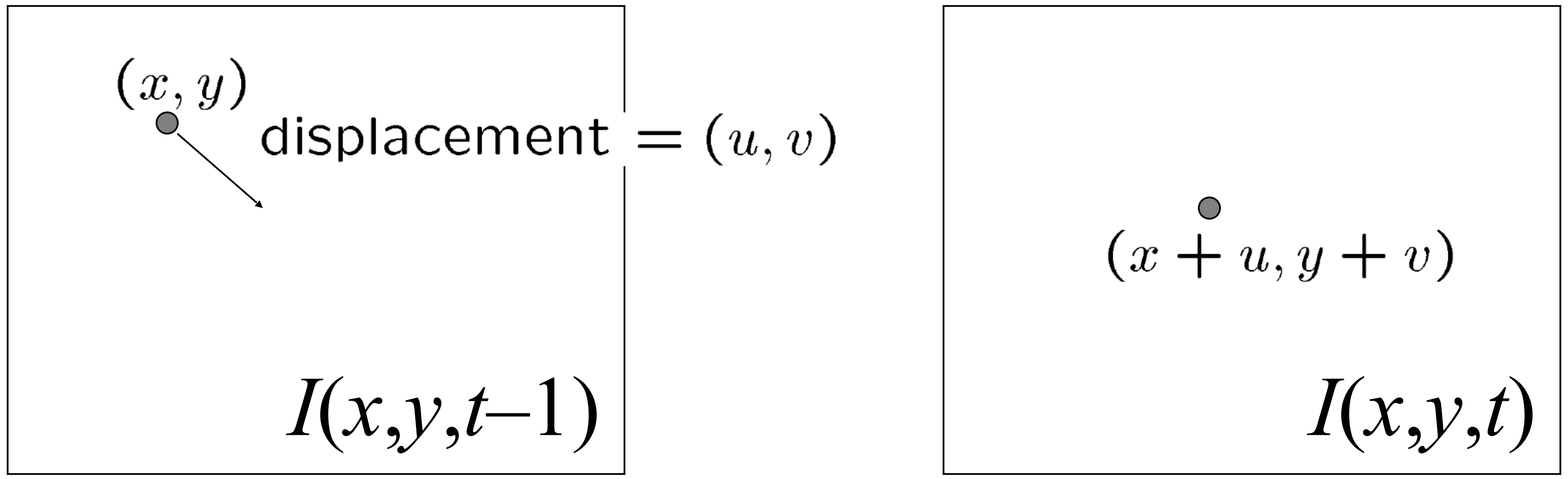


Simple loss function [Lucas & Kanade 1981]. Find flow that minimizes:

$$\mathcal{L}(u, v) = \sum_{x,y} [I(x, y, t-1) - I(x + u(x, y), y + v(x, y), t)]^2$$

Source: S. Lazebnik

The brightness constancy constraint



Brightness Constancy Equation:

$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

Linearizing the right side using Taylor expansion:

Derivative in y direction

$$I(x, y, t-1) \approx I(x, y, t) + I_x\, u(x, y) + I_y\, v(x, y)$$

Derivative in time: $I(x, y, t-1) - I(x, y, t)$

Therefore: $I_x\, u + I_y\, v + I_t \approx 0$

Source: S. Lazebnik

The brightness constancy constraint

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

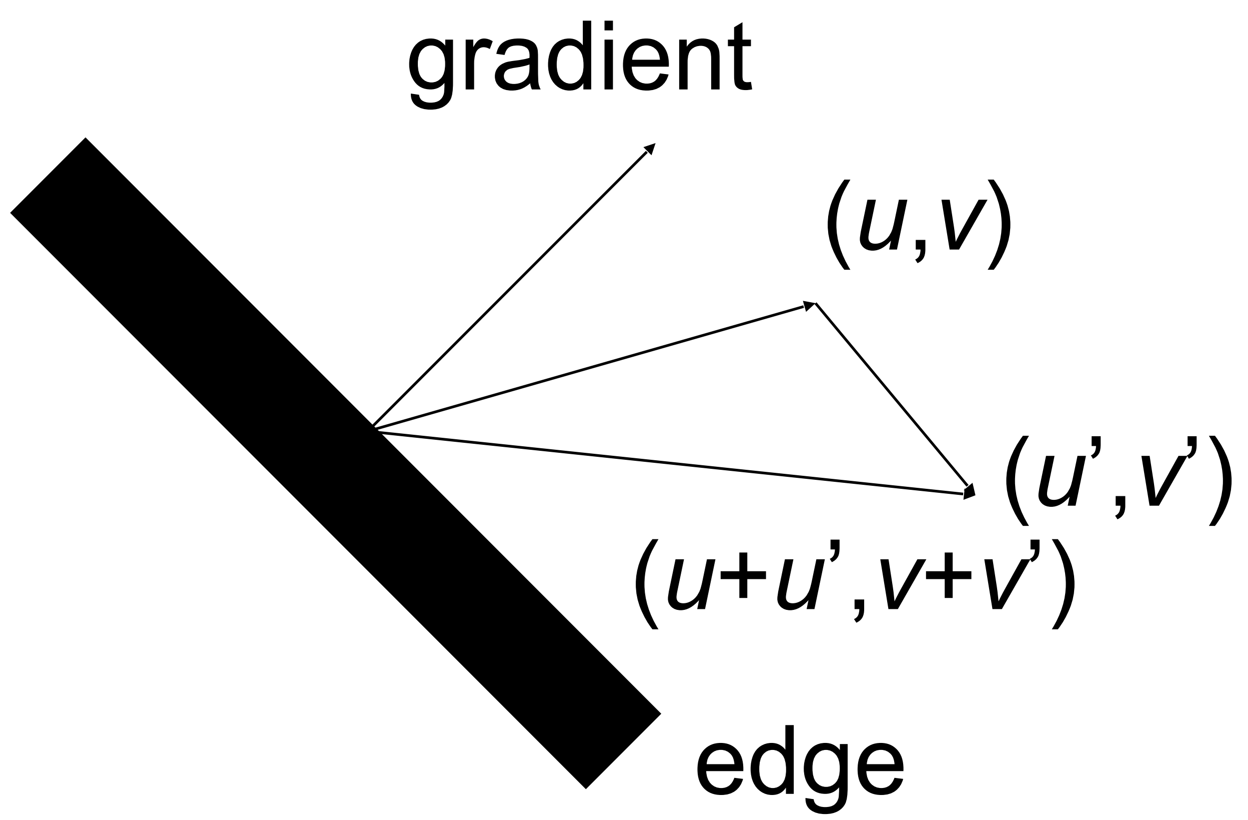
- How many equations and unknowns per pixel?
  - One equation, two unknowns

- Under-constrained. Let's rewrite it:

$$\nabla I \cdot (u, v) + I_t = 0$$

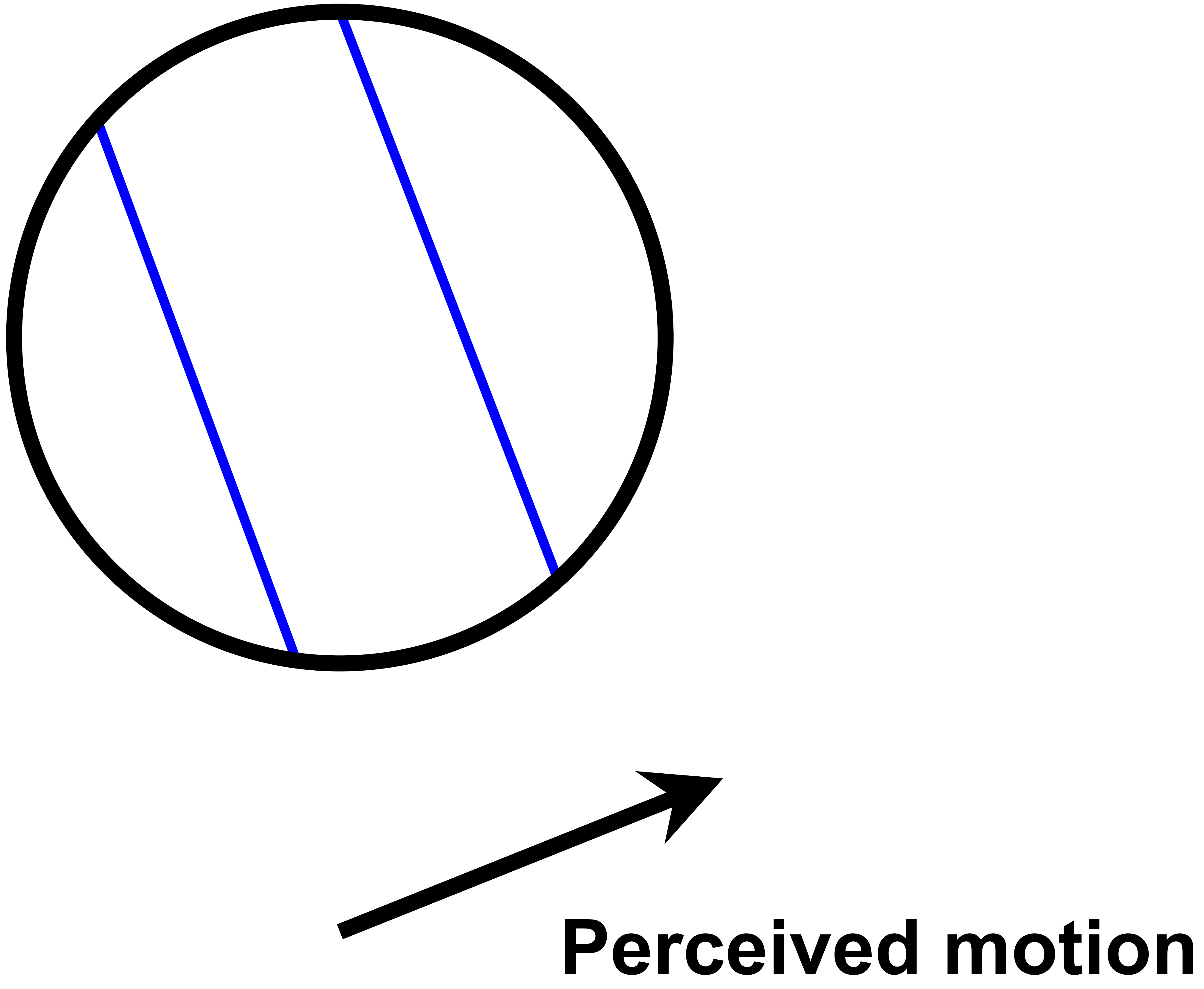
- The component of the flow perpendicular to the image gradient (i.e., parallel to the edge) is unknown!

If (u, v) satisfies the equation,
so does (u+u', v+v') if $\nabla I \cdot (u', v') = 0$



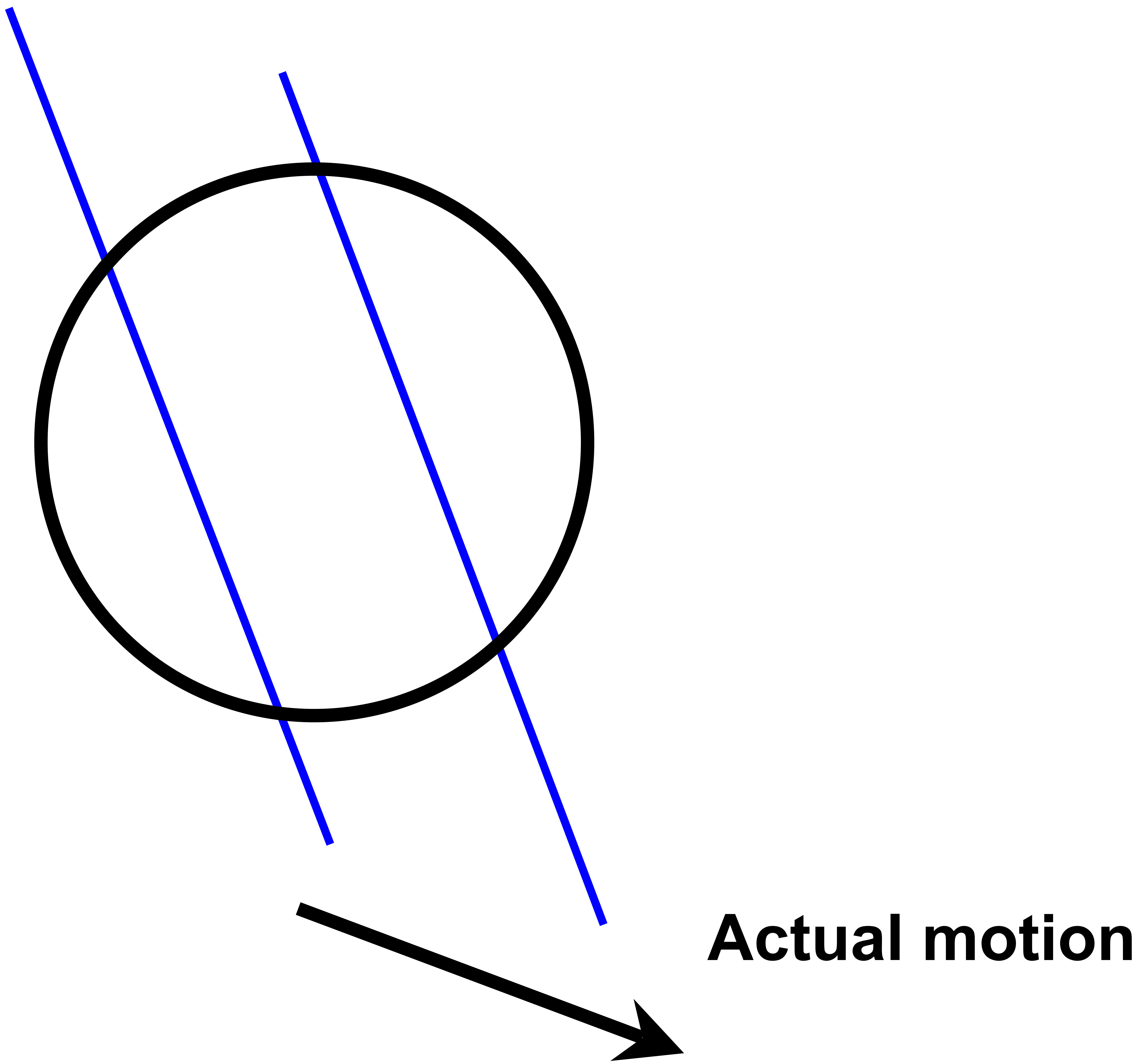
Source: S. Lazebnik

The aperture problem



Source: S. Lazebnik

The aperture problem



Source: S. Lazebnik

The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion

Source: S. Lazebnik

The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion

Source: S. Lazebnik

Solving the aperture problem

- How to get more equations for a pixel?
- **Spatial coherence constraint:** assume the pixel's neighbors have the same (u,v)
  - E.g., if we use a 5x5 window, that gives us 25 equations per pixel

$$\nabla I(\mathbf{x}_i) \cdot [u, v] + I_t(\mathbf{x}_i) = 0$$

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

[Lucas & Kanade 1981]

Source: S. Lazebnik

Lucas-Kanade flow

Least squares problem:

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

When is this system solvable?

[Lucas & Kanade 1981]

Source: S. Lazebnik

Lucas-Kanade optical flow

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

$$\underset{n\times 2}{\mathbf{A}} \; \underset{2\times 1}{\mathbf{d}} = \underset{n\times 1}{\mathbf{b}}$$

- Solution given by $(\mathbf{A}^T\mathbf{A})\mathbf{d} = \mathbf{A}^T\mathbf{b}$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

(summations are over all pixels in the window)

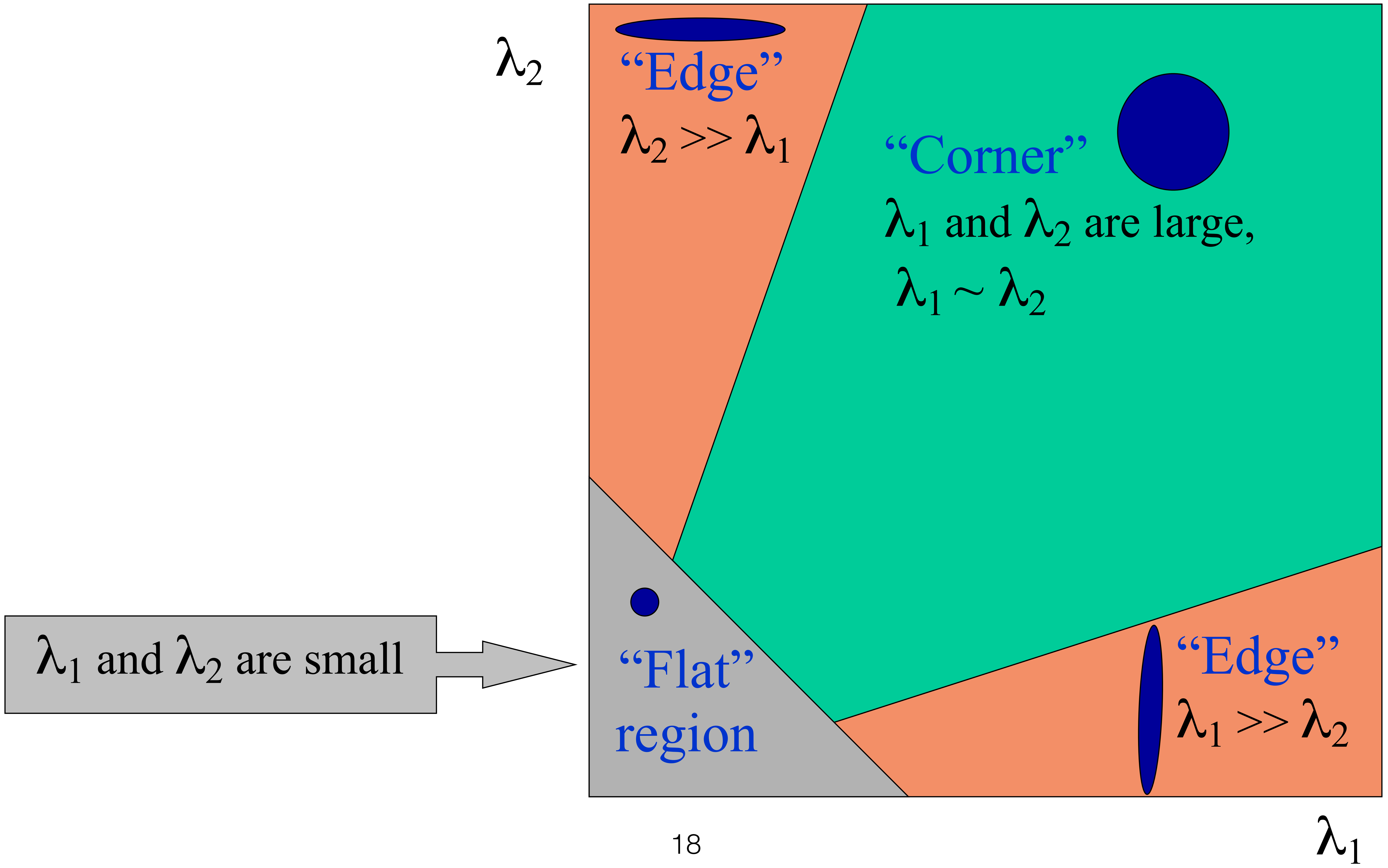
$\mathbf{M} = \mathbf{A}^T\mathbf{A}$ is the "second moment" matrix (also Gauss-Newton approximation to Hessian)

[Lucas & Kanade 1981]

Source: S. Lazebnik

Analyzing the second moment matrix

- Estimation of optical flow is well-conditioned precisely for regions with high “cornerness”:

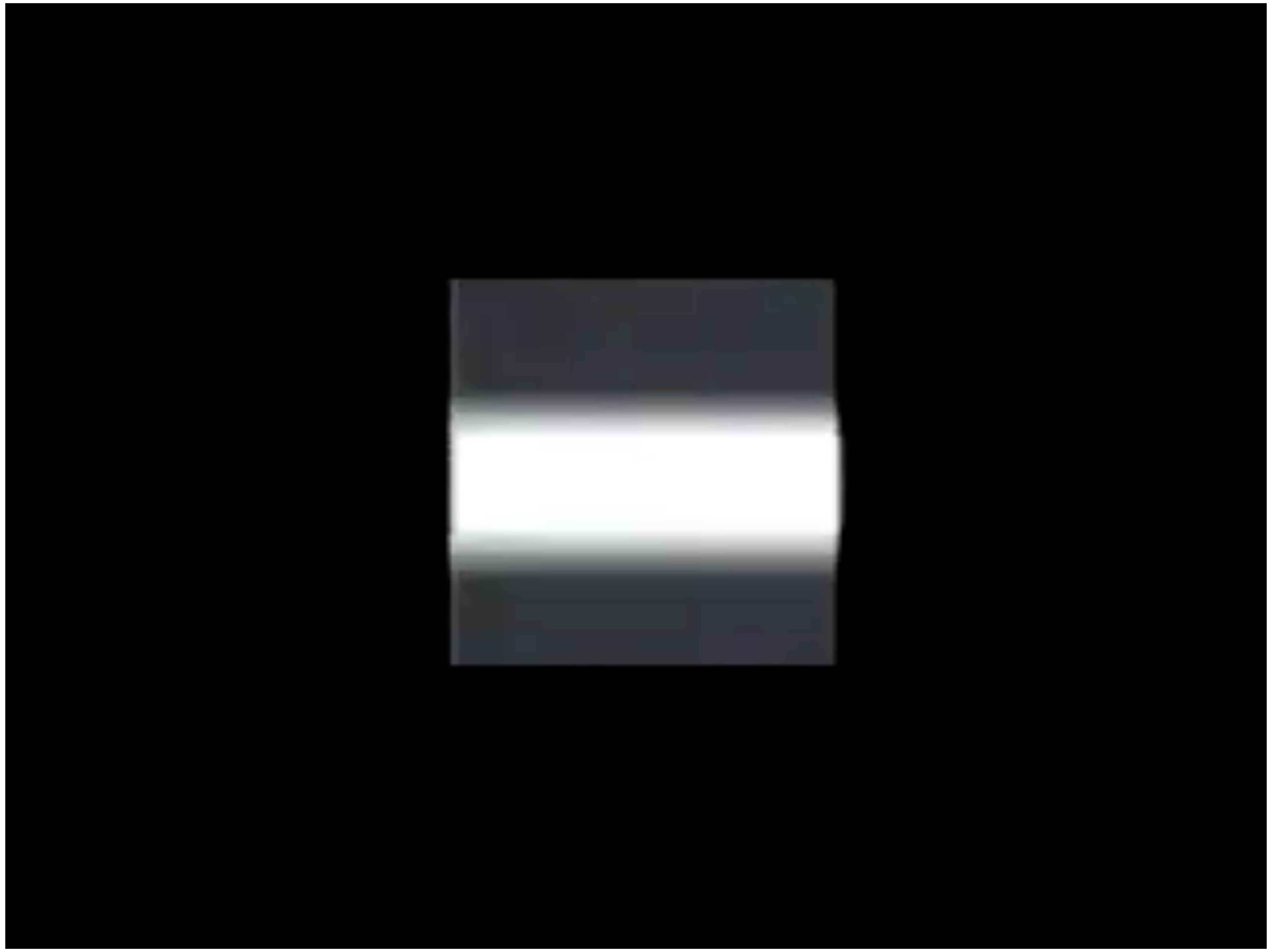


Eigenvalues of **M**

Source: S. Lazebnik

Conditions for solvability

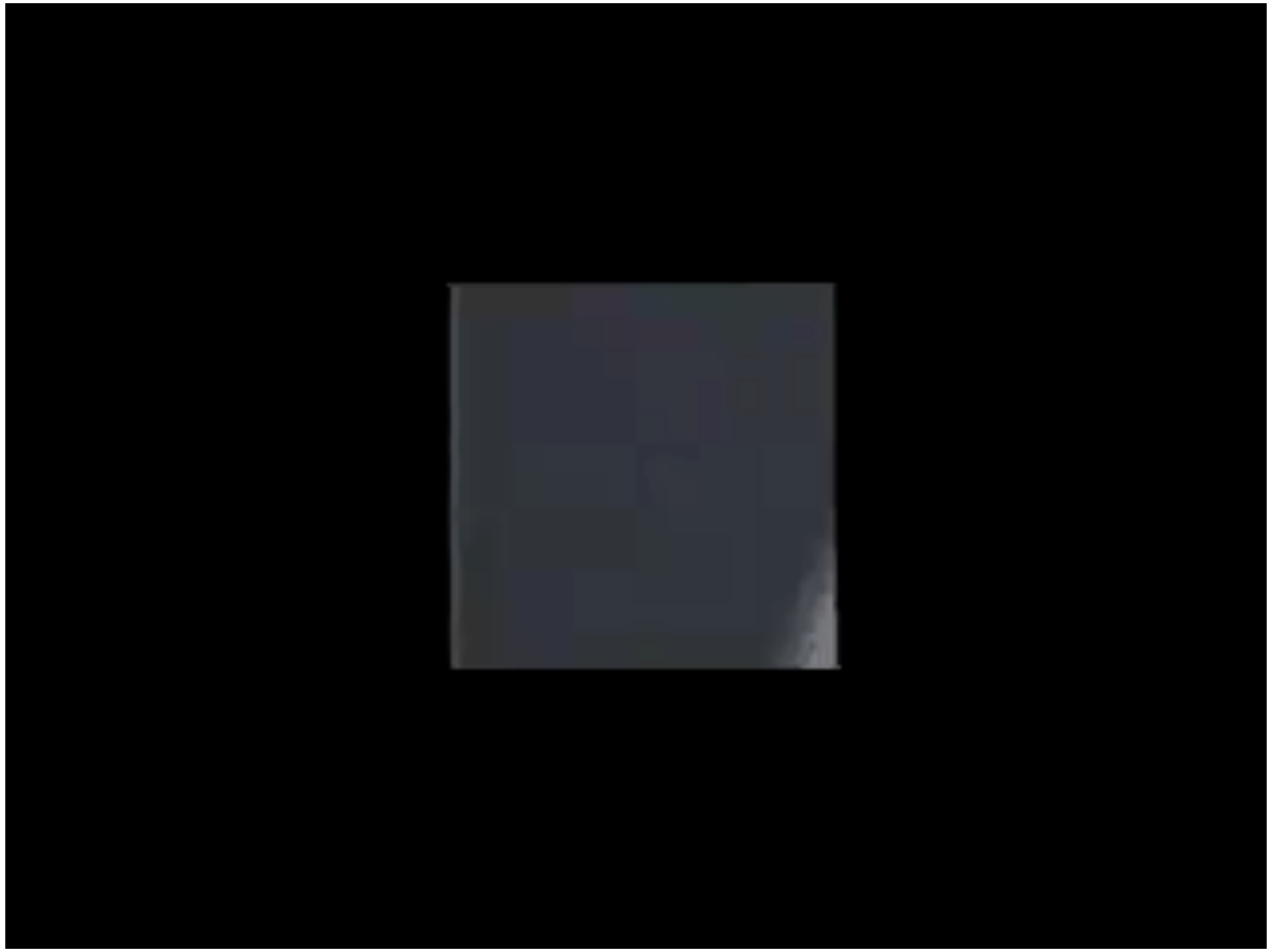
Bad case: single, straight edge



Source: S. Lazebnik

Conditions for solvability

Good case

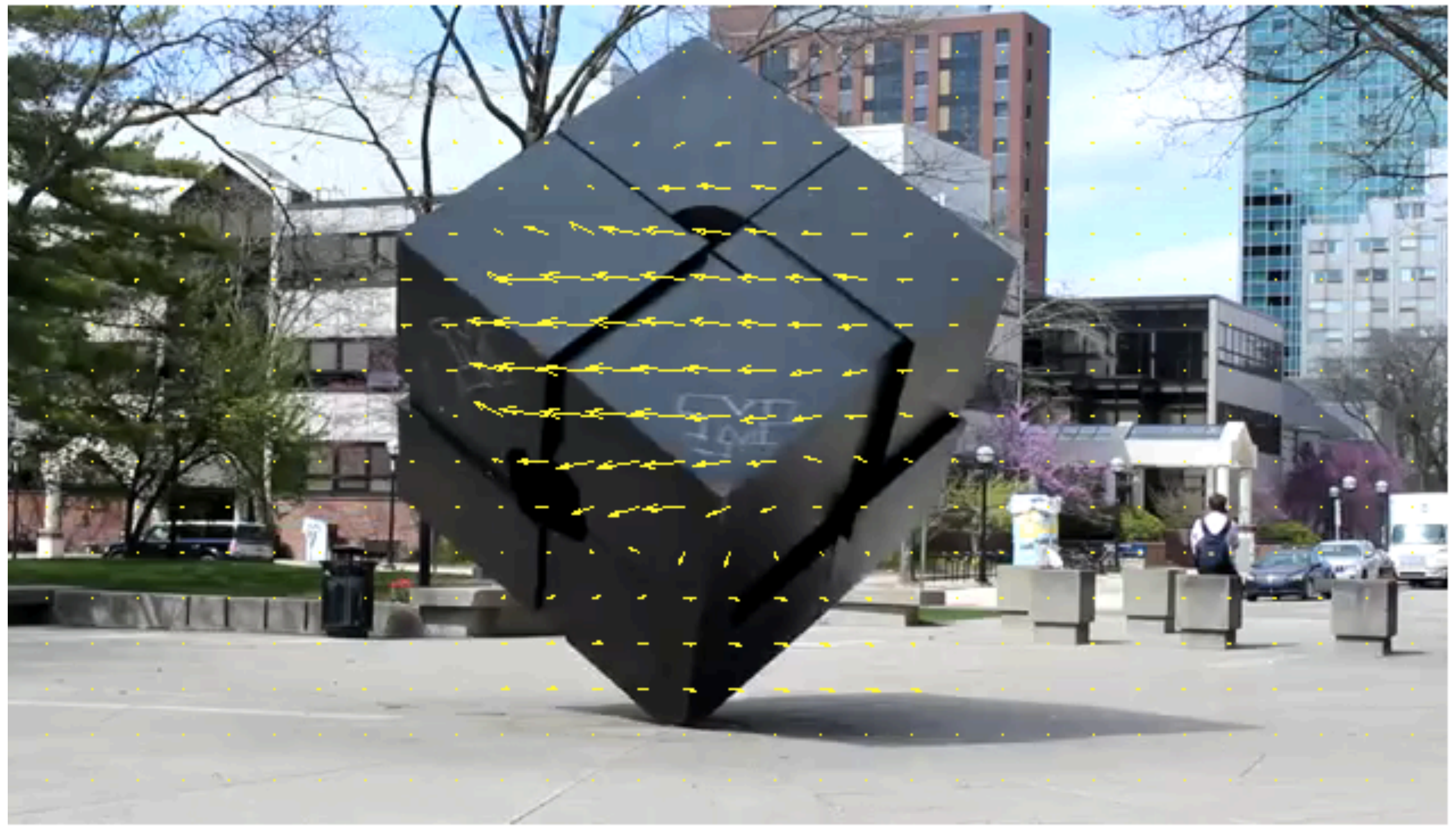


Source: S. Lazebnik

Lucas-Kanade flow example

Input frames

Output



Source: MATLAB Central File Exchange

Source: S. Lazebnik

Fixing the errors in Lucas-Kanade

- The motion is large (larger than a pixel)
  - Iterative refinement
  - Multi-resolution (coarse-to-fine) estimation

- Local ambiguity
  - Smooth using graphical model refinement

Source: S. Lazebnik

Large motions

Source: Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Idea #1: iterative estimation

Goal: minimize matching error

$$\mathcal{L}(u,v) = \sum_{x,y} \sum_{x'=x-N}^{x+N} \sum_{y'=y-N}^{y+N} [I(x', y', t-1) - I(x' + u(x,y), y' + v(x,y), t)]^2$$

where the window width/height is 2N+1

Iterative algorithm:

Initialize flow: $u_0(x, y) = v_0(x, y) = 0$

For each iteration *i*:

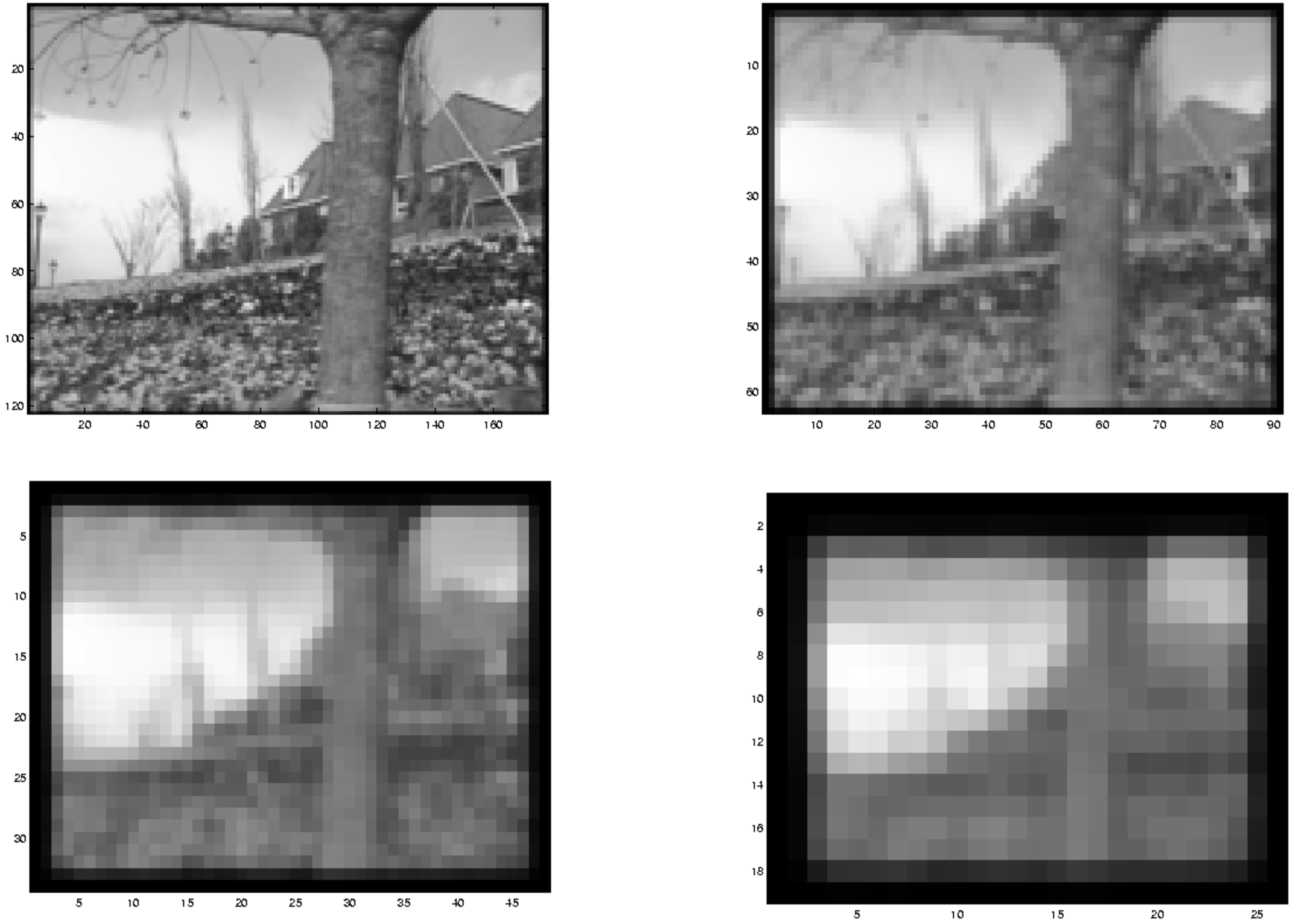
1. Linearize around current solution. Find an update for problem:

$$\underset{\Delta u, \Delta v}{\text{argmin}} \; \mathcal{L}(u_i + \Delta u, v_i + \Delta v)$$

by solving linear least squares problem for each pixel: $Ad = b$

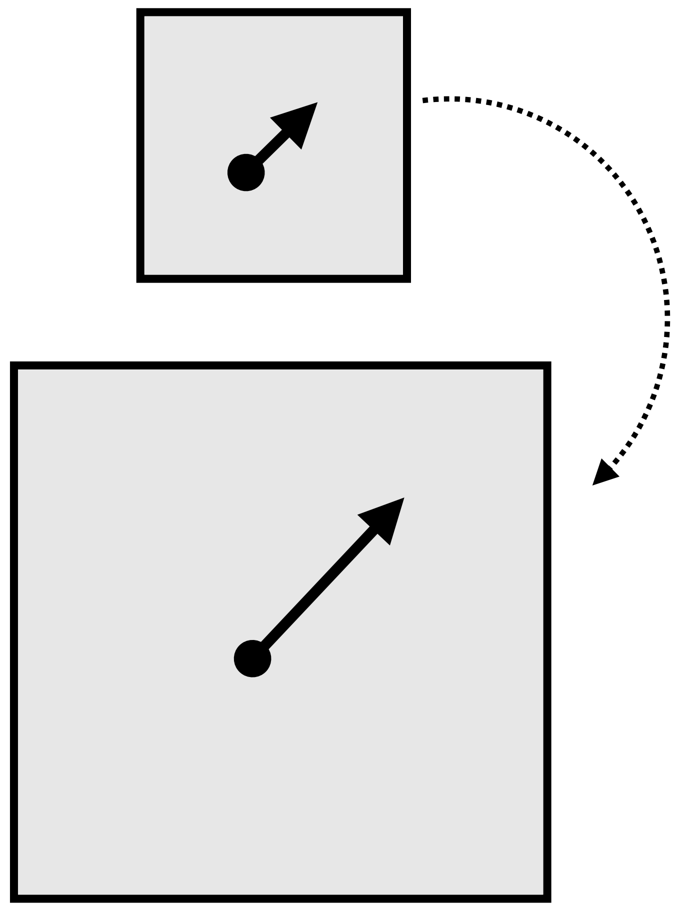
2. Apply updates: $u_{i+1} = u_i + \Delta u$ and $v_{i+1} = v_i + \Delta v$

Idea #2: Multi-scale estimation



Source: Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Idea #2: Multi-scale estimation



Initialize flow: $u_0(x, y) = v_0(x, y) = 0$

For each scale *s* of Gaussian pyramid:

Initialize flow from previous (coarser) scale

For each iteration *i*:

1. Linearize around current solution. Find an update for problem:

$$\underset{\Delta u, \Delta v}{\operatorname{argmin}} \; \mathcal{L}(u_i + \Delta u, v_i + \Delta v)$$

by solving linear least squares problem for each pixel: $Ad = b$

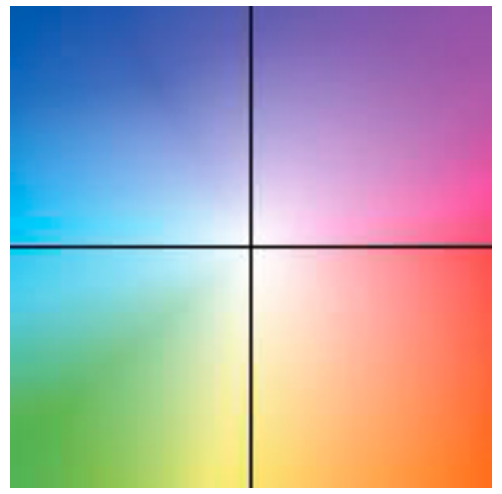
2. Apply updates: $u_{i+1} = u_i + \Delta u$ and $v_{i+1} = v_i + \Delta v$

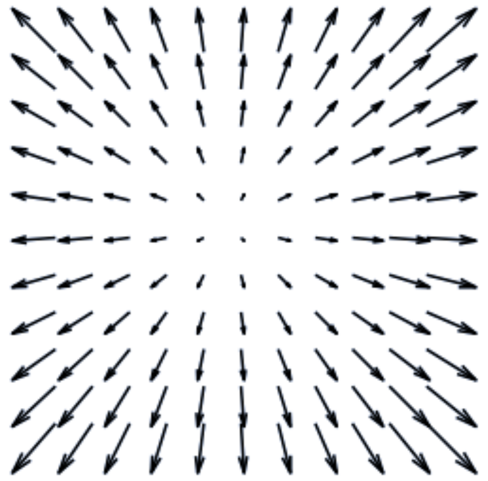
3. Extra trick for smoother flow: apply median filter to u_{i+1} and v_{i+1}

Example

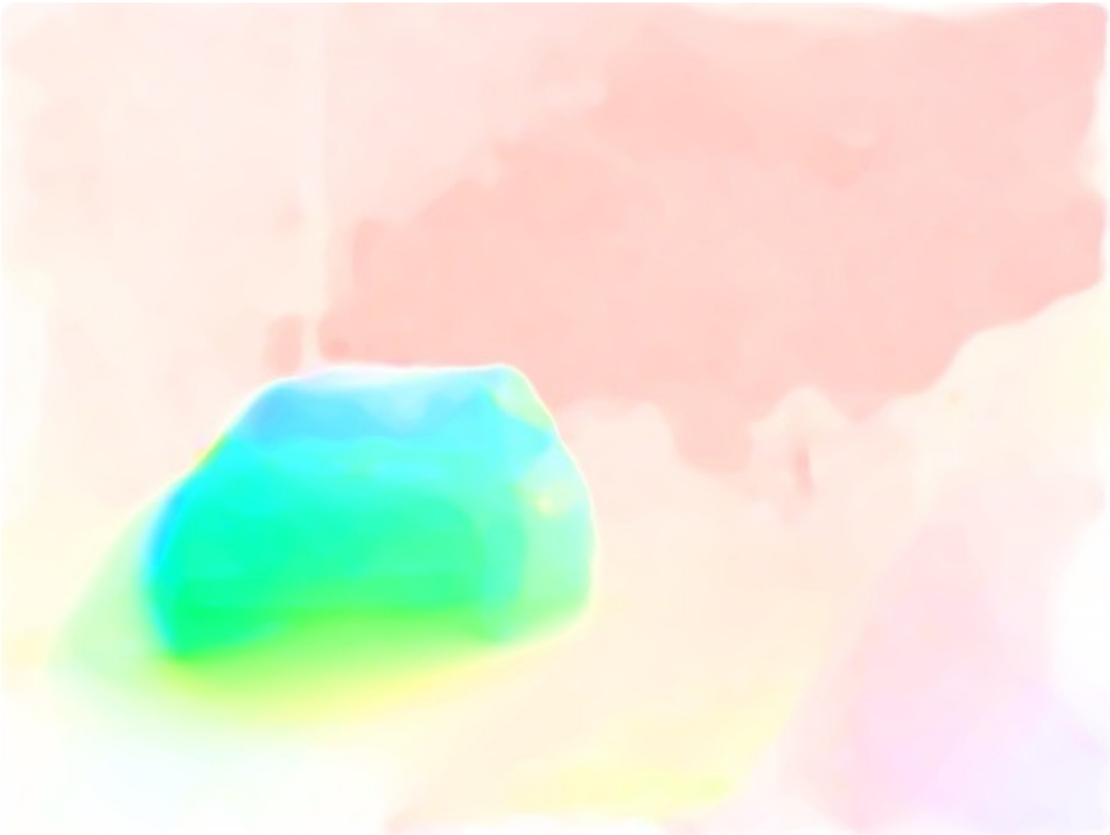
Input two frames

Coarse-to-fine LK





Flow visualization



Coarse-to-fine LK with median filtering

Source: Ce Liu

Smoothness assumption

Goal: minimize matching error + smoothness [Horn and Schunck 1981]

$$\sum_{x,y}[I(x,y,t-1) - I(u(x),v(y),t)]^2 + \sum_{p}\sum_{p' \in \mathcal{N}}(u(p)-u(p'))^2 + (v(p)-v(p'))^2$$

$E_d(u,v)$ match cost

$E_s(u,v)$ smoothness

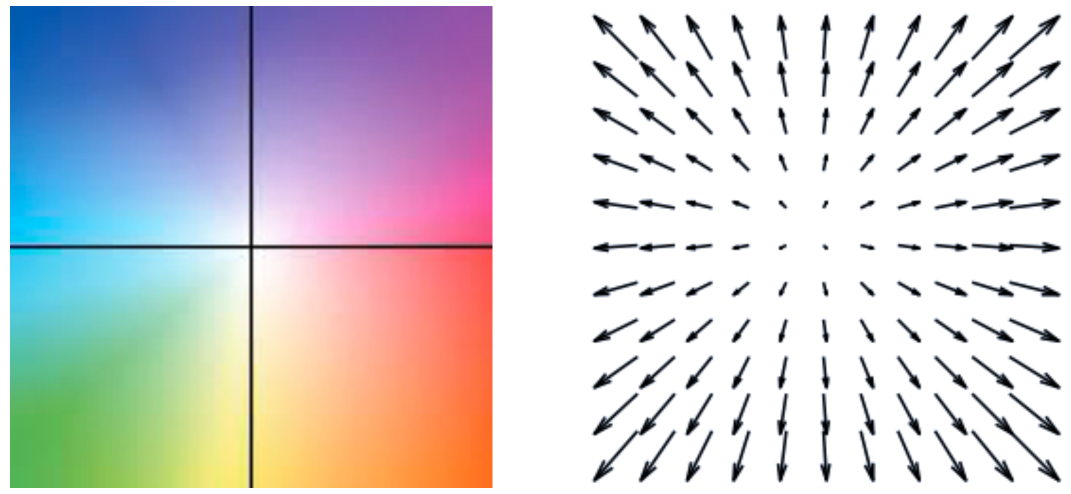
where p and p’ are neighboring pixels

- Can solve using gradient descent or nonlinear least squares

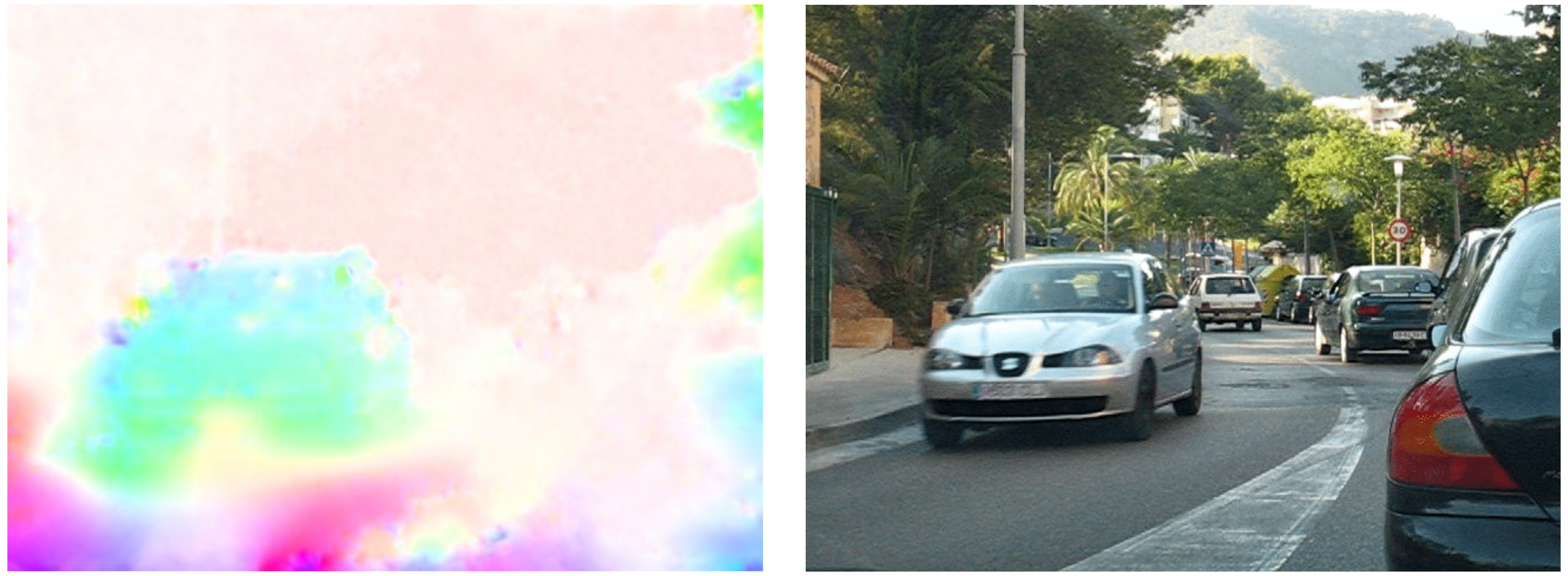
Smoothness assumption

Input two frames

Horn-Schunck



Flow visualization

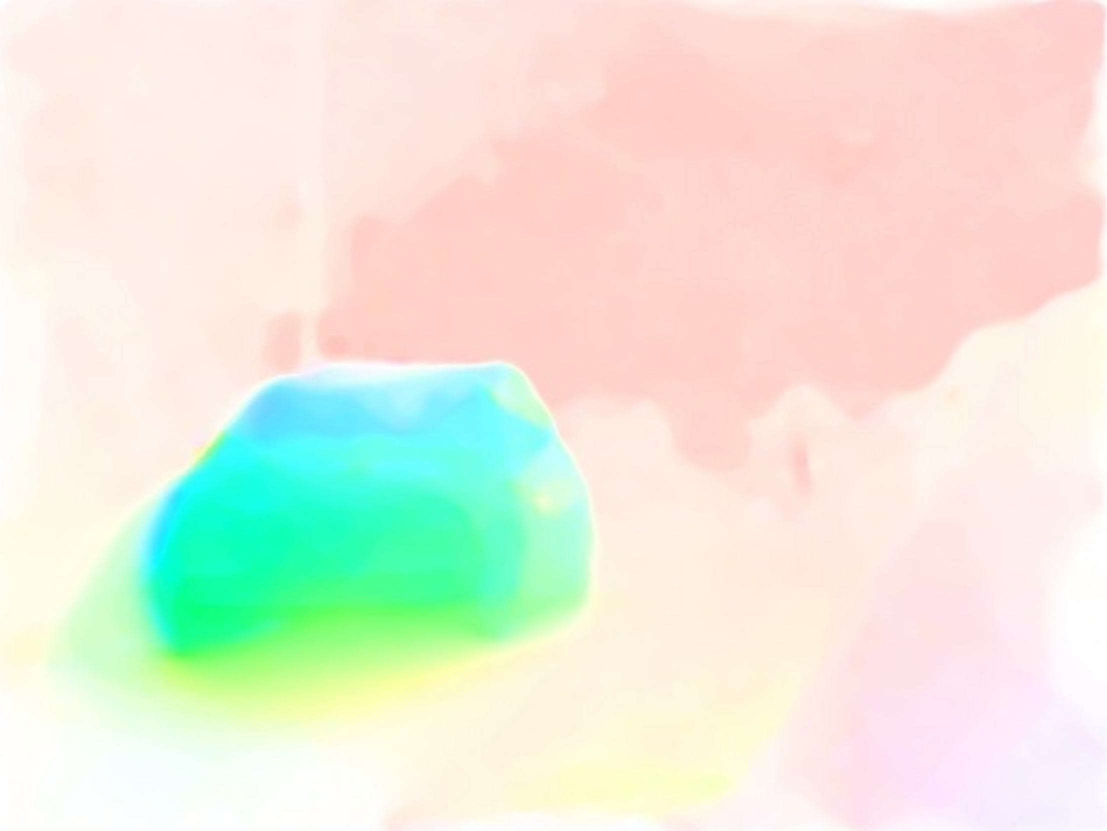


Coarse-to-fine LK

Source: Ce Liu

Warping

I_1



Flow: u, v

→

$\hat{I}_2 = \text{warp}(I_1; u, v)$

- I_1 should be similar to I_2 after **warping** flow,
  i.e. mapping (x, y) → (x + u(x, y), y + v(x, y))
- As we estimate flow, the warped I_1 becomes closer and closer to I_2

Flow with warping

Initialize flow: $u_0(x, y) = v_0(x, y) = 0$

For each scale *s*:

  Initialize flow from previous (coarser) scale

  For each iteration *i*:

1. Warp I_2 to be be closer to I_1 using

2. Match I_1 to **warped** I_2. Each pixel searches in local neighborhood.

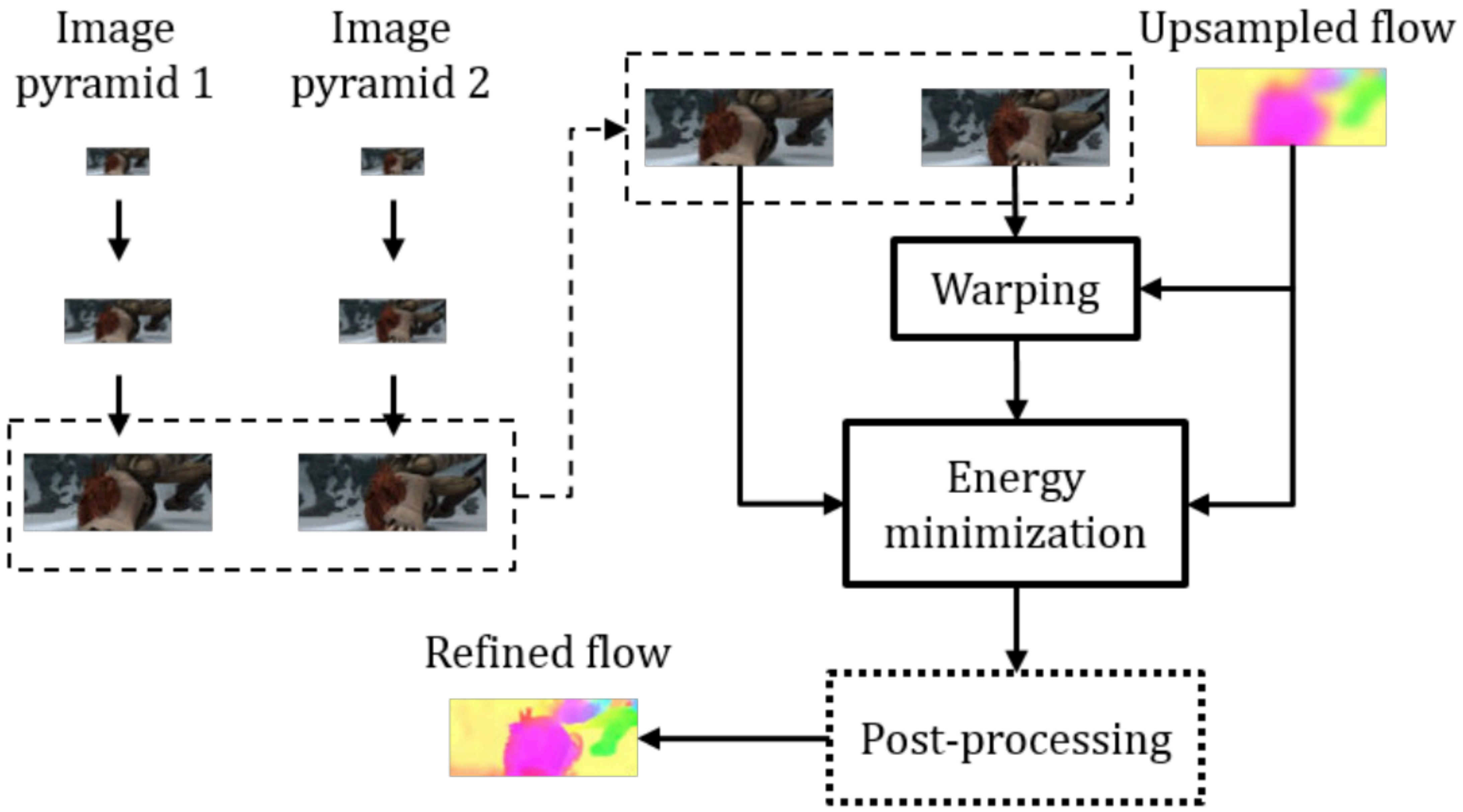
$$\operatorname*{argmin}_{\Delta u, \Delta v} \mathcal{L}(\Delta u, \Delta v)$$

by solving linear least squares problem for each pixel: $Ad = b$

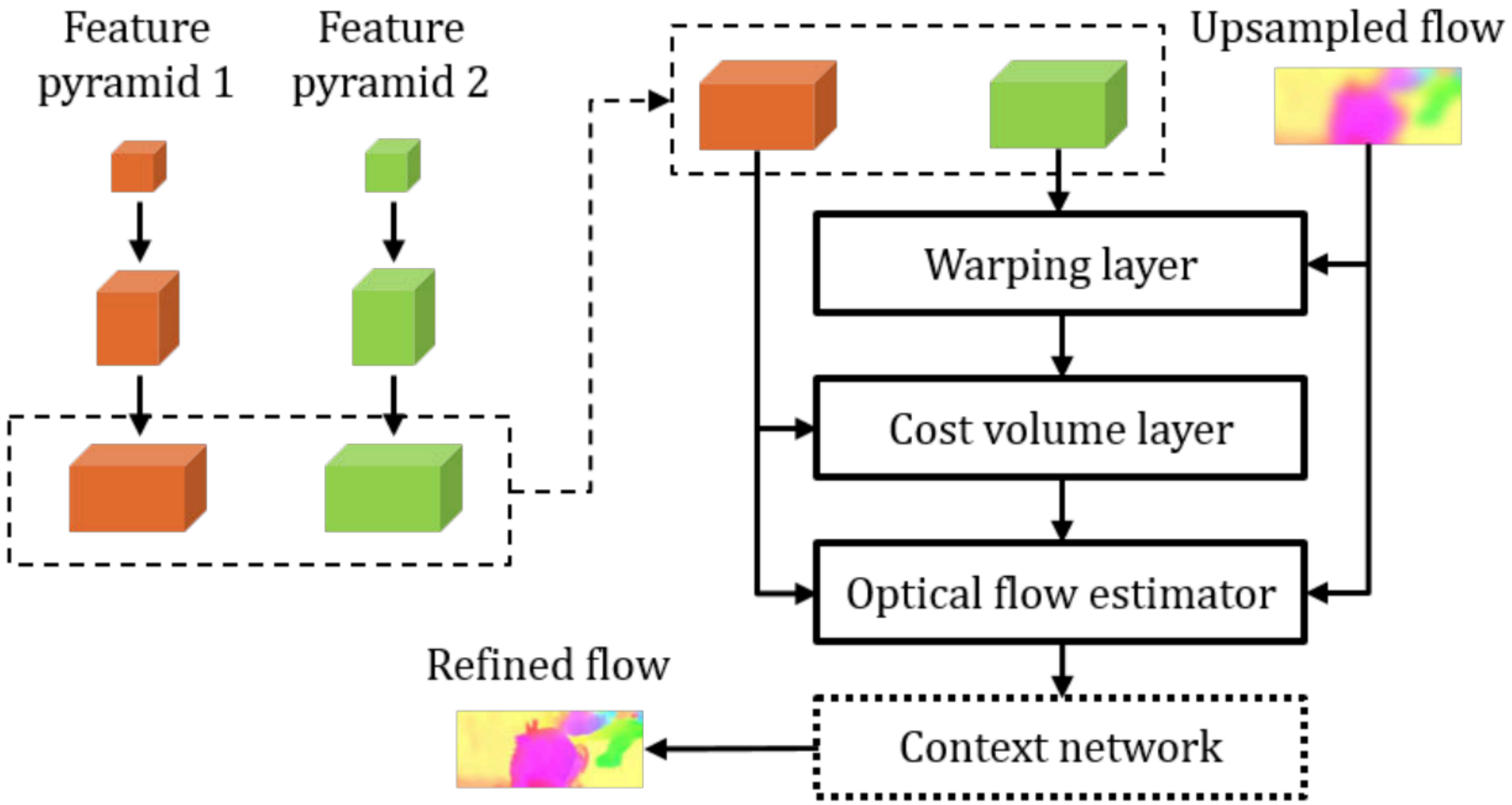
3. Apply updates: $u_{i+1} = u_i + \Delta u$ and $v_{i+1} = v_i + \Delta v$

Flow CNNs

Match CNN features instead of pixels!



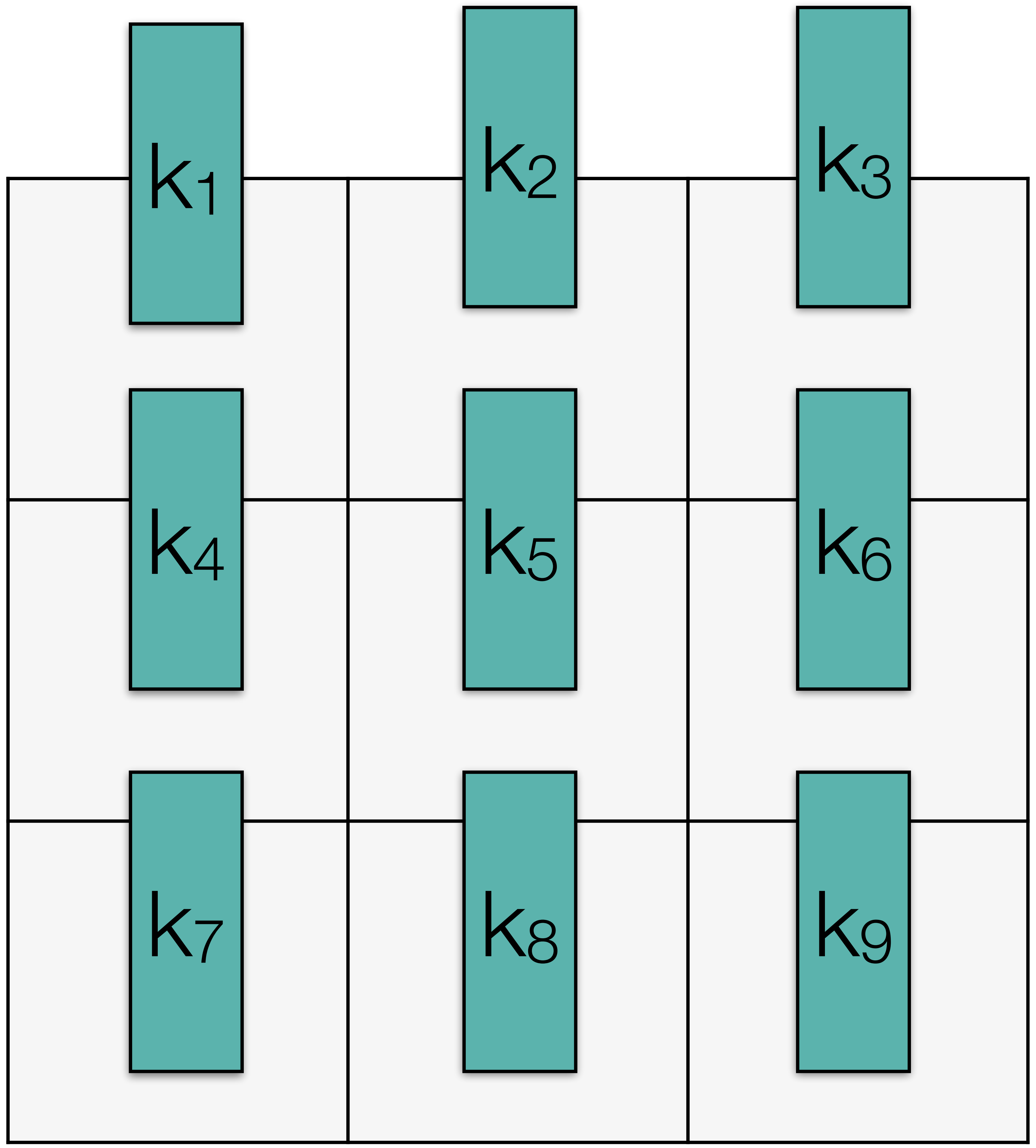
Traditional coarse-to-fine flow



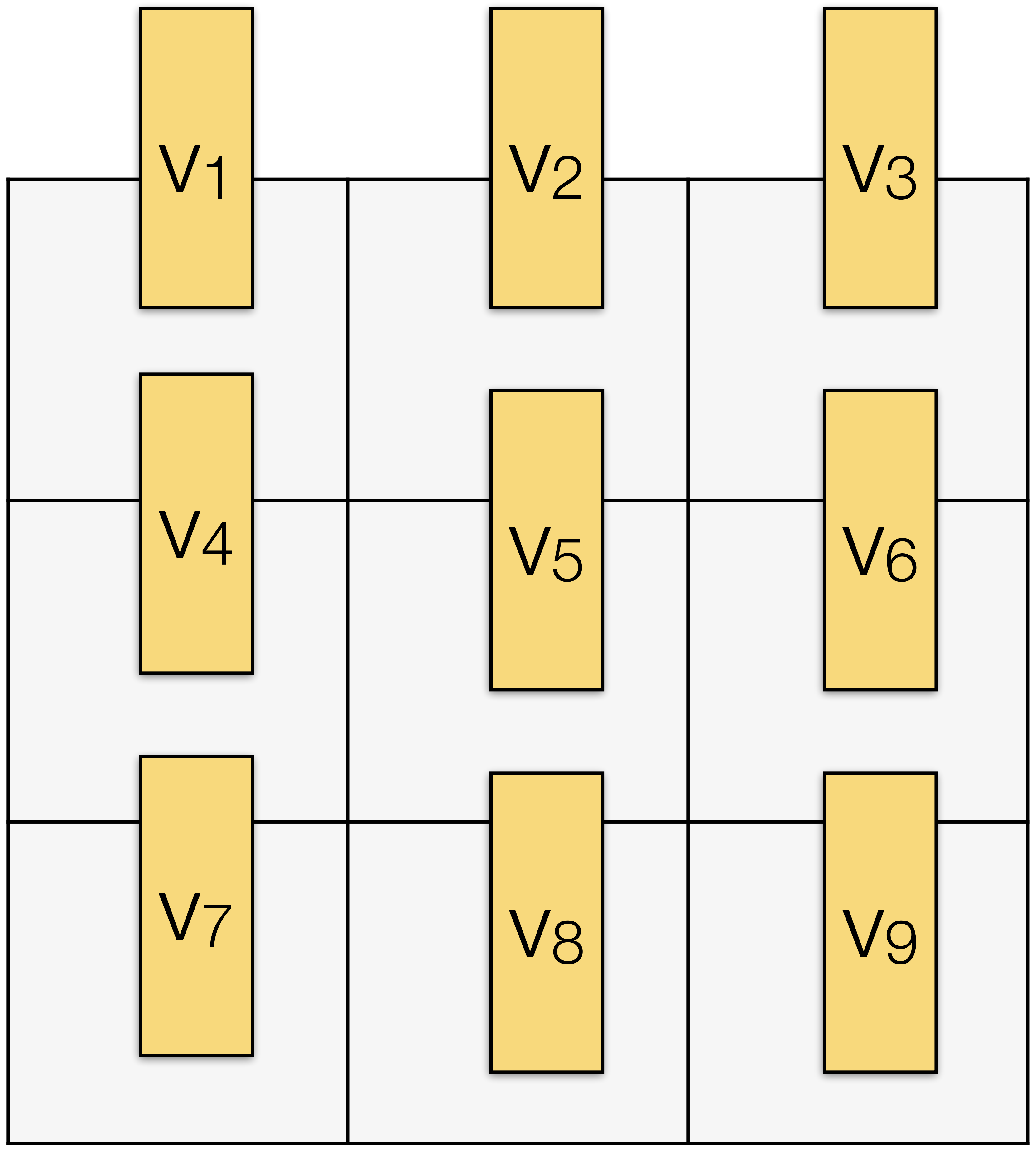
PWC-net

[Sun et al., “PWC-Net”, 2018]

Correlation between CNN features

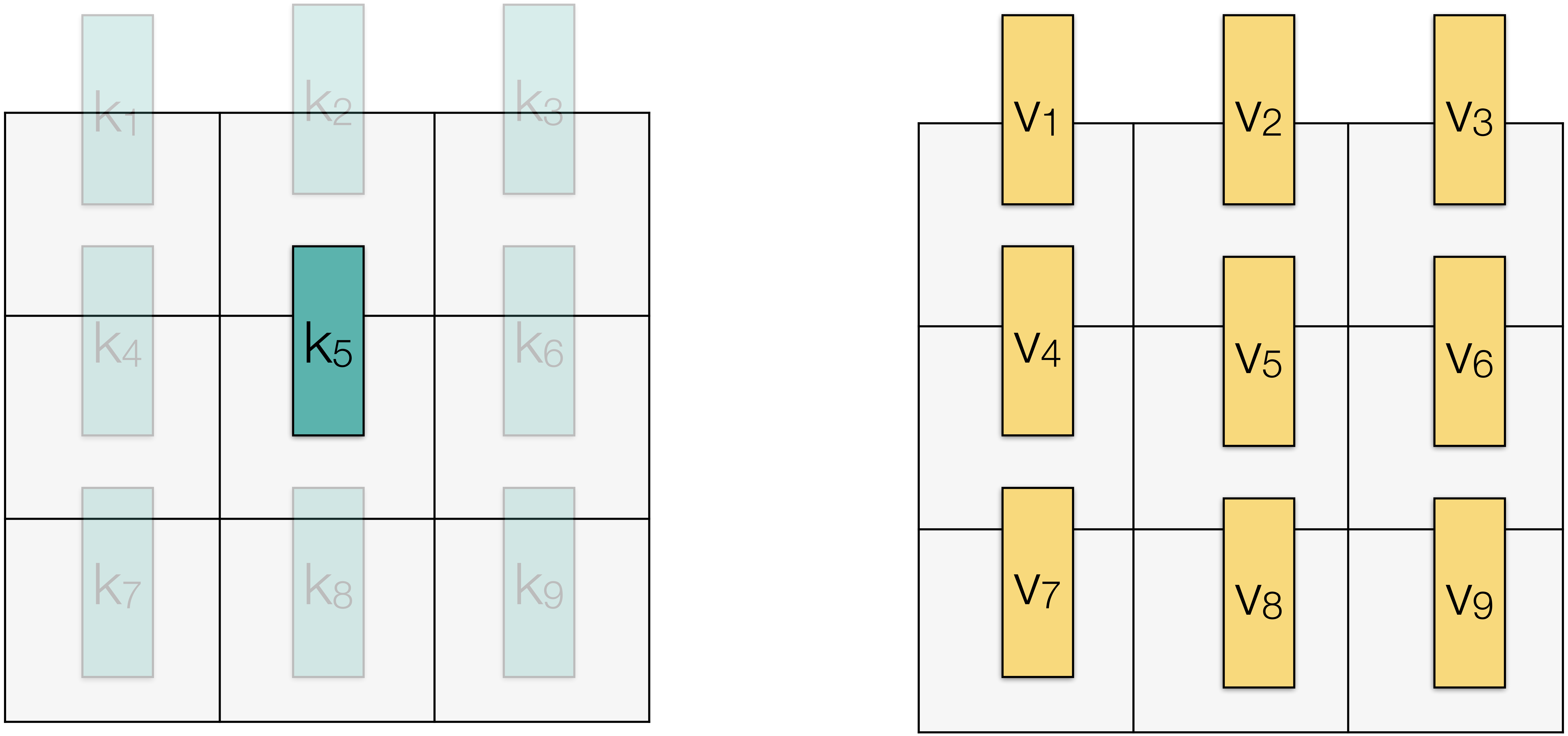


CNN feature map for I_1



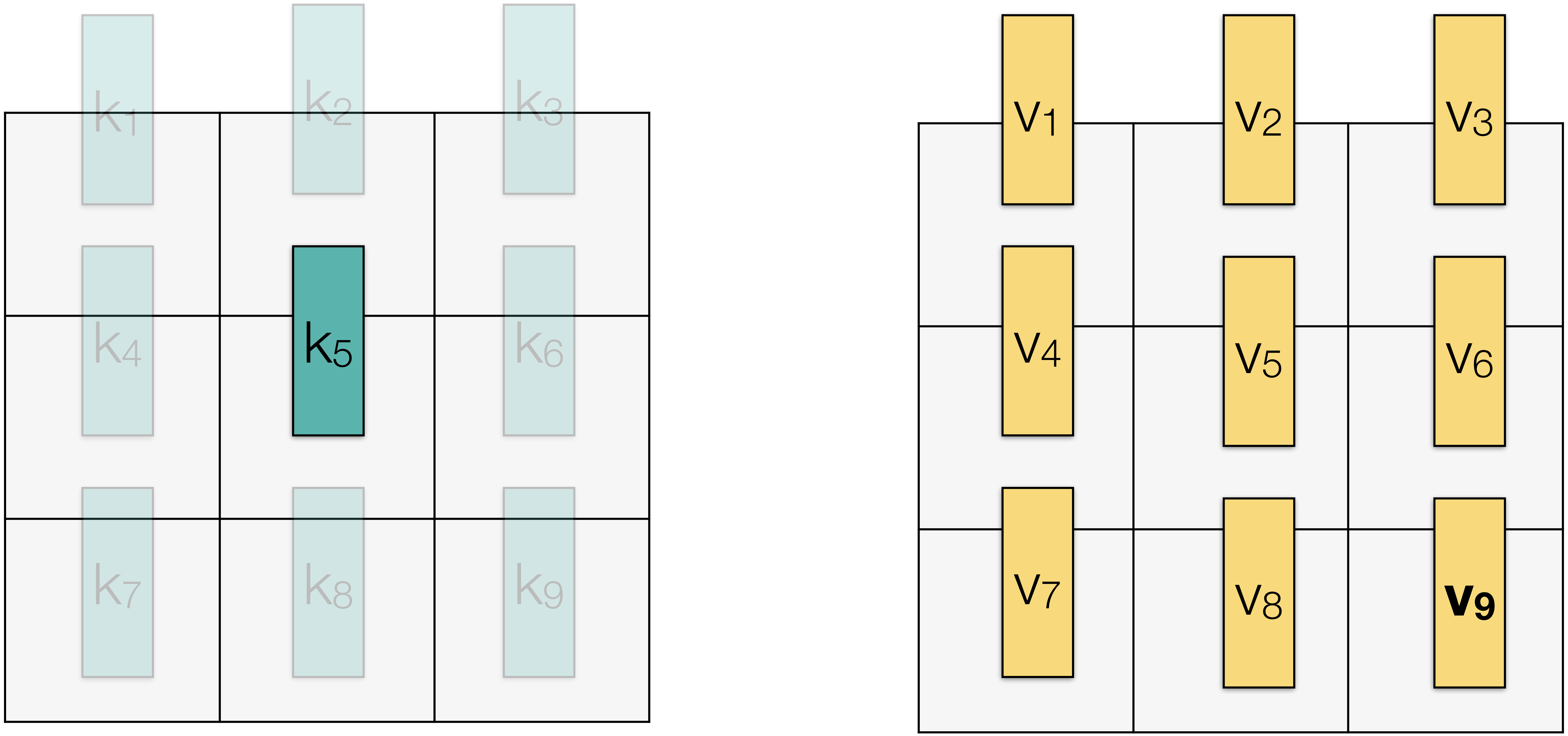
CNN feature map for I_2

Correlation between CNN features



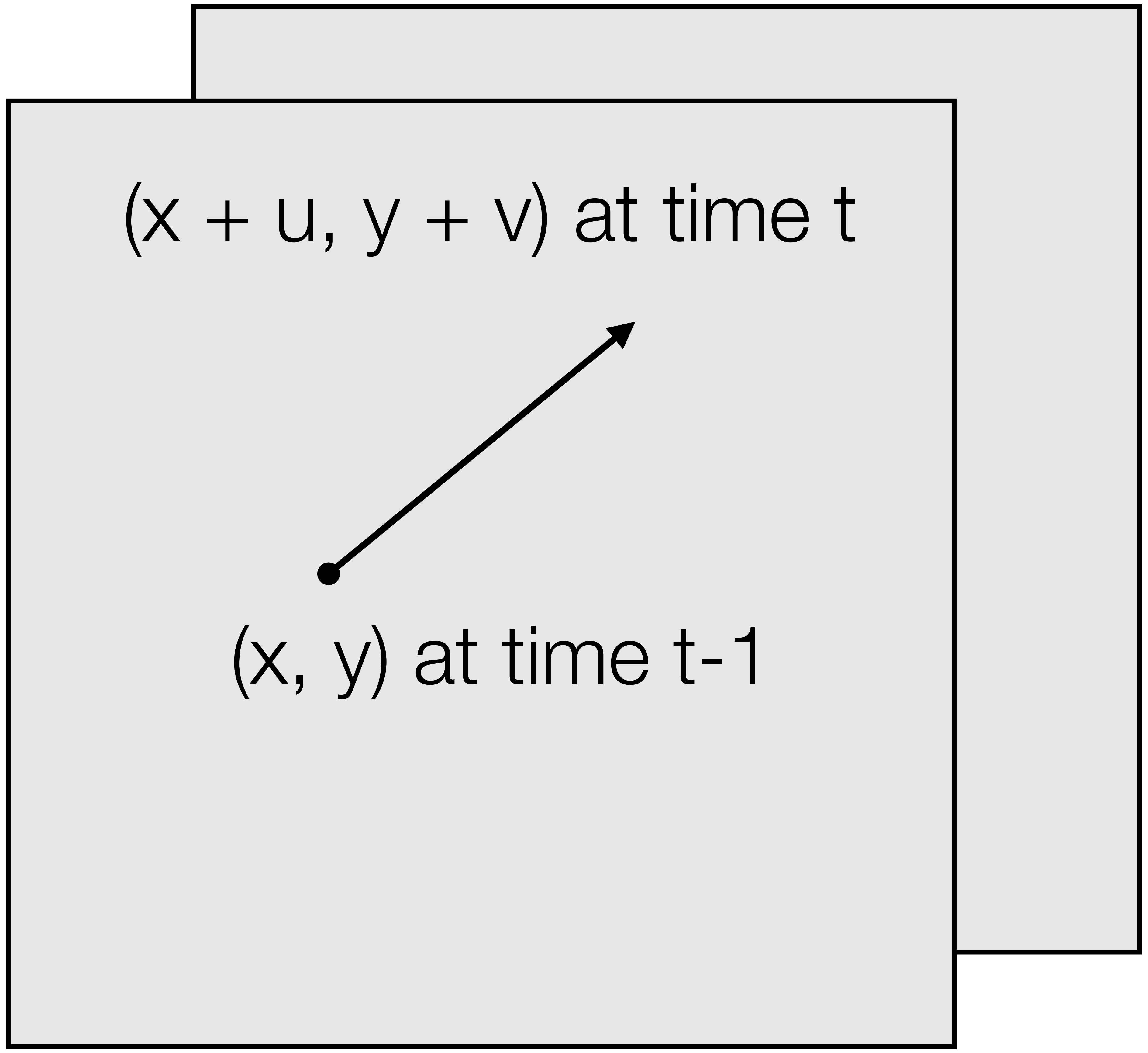
Take dot product between features and choose largest one.

Correlation between CNN features



$u = 1, v = 1$

Simple application: slow motion



- Flow used in lots of familiar places!
  - E.g., video compression, denoising, action recognition, …
- One application: use flow to estimate where pixel will be *between* frames
- Synthesize intermediate frames

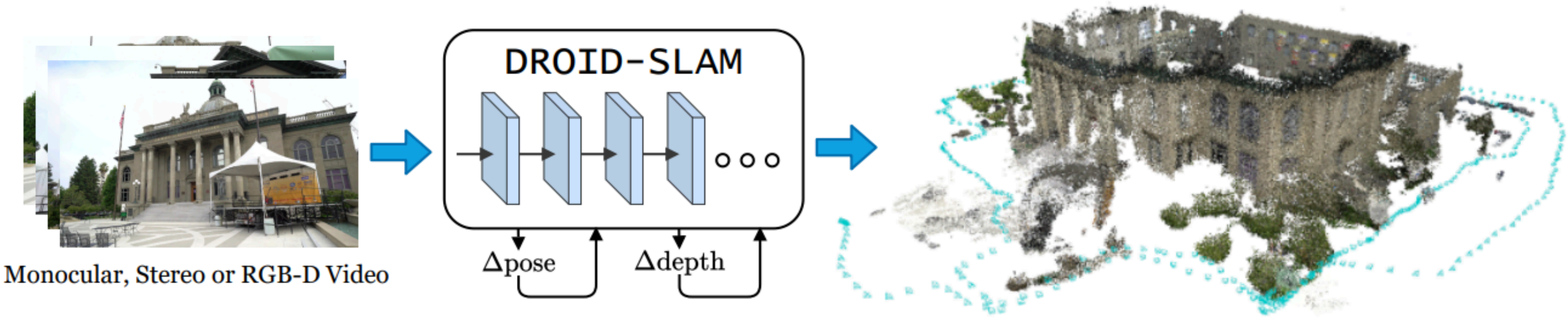
Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation

Huaizu Jiang[1], Deqing Sun[2], Varun Jampani[2]
Ming-Hsuan Yang[3,2], Erik Learned-Miller[1], Jan Kautz[2]
[1]UMass Amherst [2]NVIDIA [3]UC Merced

(No audio commentary)

https://people.cs.umass.edu/~hzjiang/projects/superslomo/superslomo_public.mp4

Structure from motion with flow

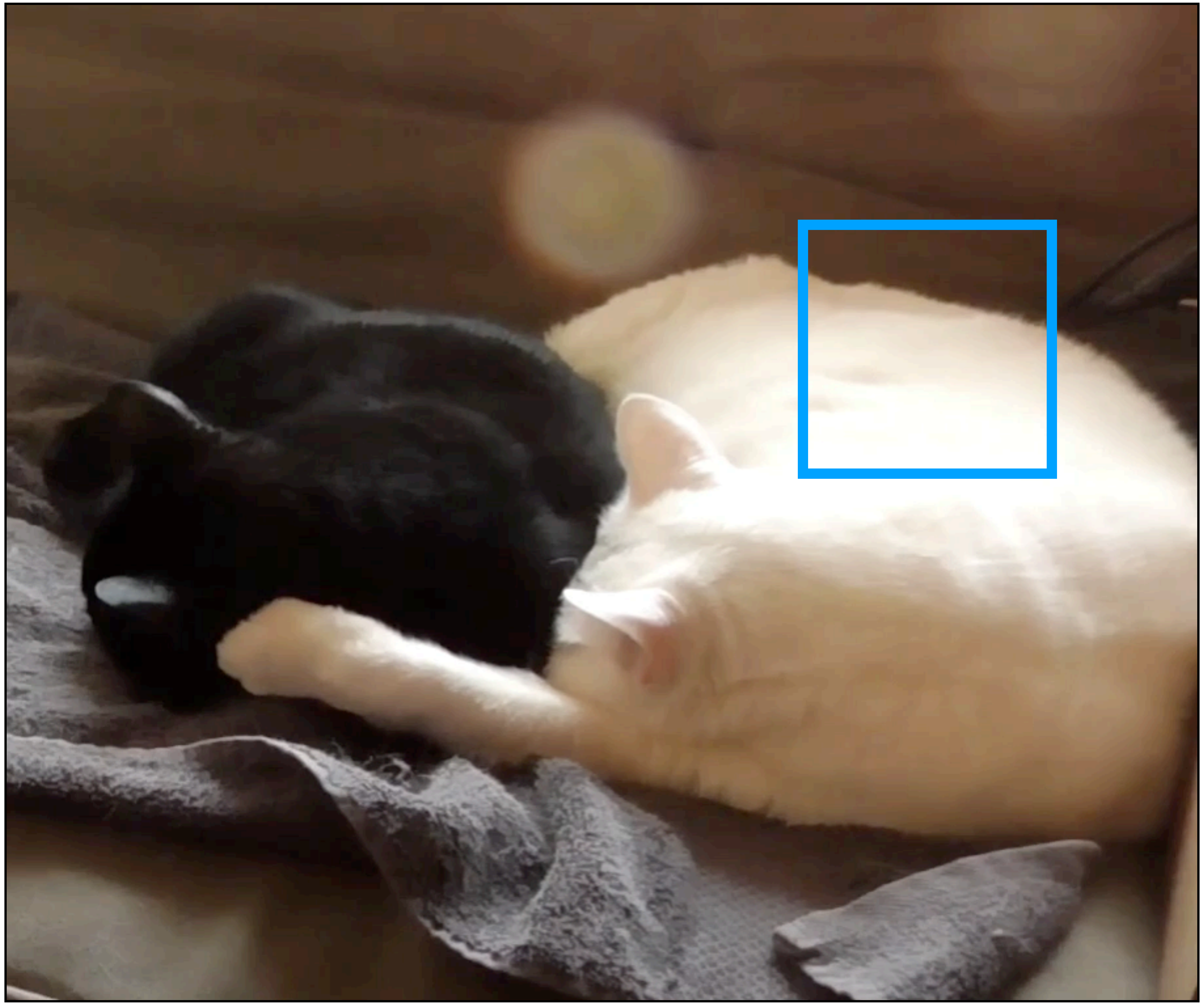


Use optical flow to find correspondences between frames.

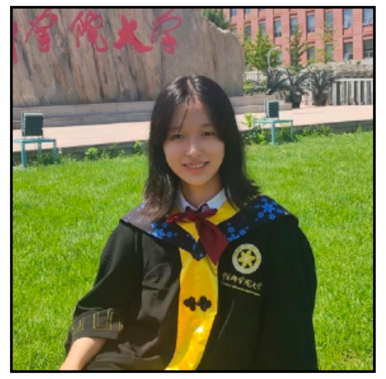
[Teed and Deng, 2021]

Structure from motion with flow

[Teed and Deng, 2021]



Application: motion magnification.
enlarge tiny motions, making them easier to perceive [Liu et al., 2005].

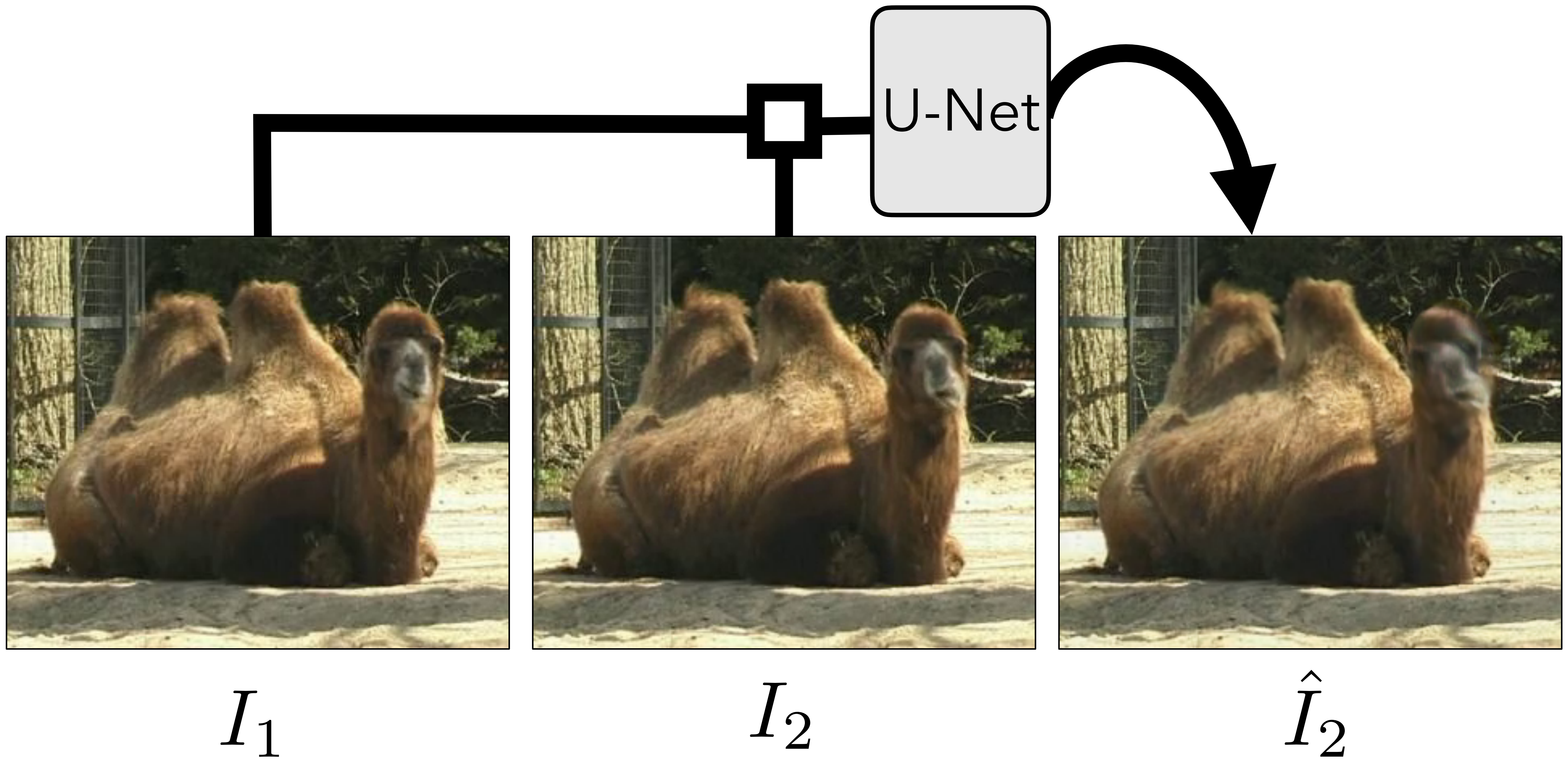


Zhaoying Pan

Daniel Geng

[Pan*, Geng*, Owens, "Self-Supervised Motion Magnification by Backpropagating Through Optical Flow", NeurIPS 2023]

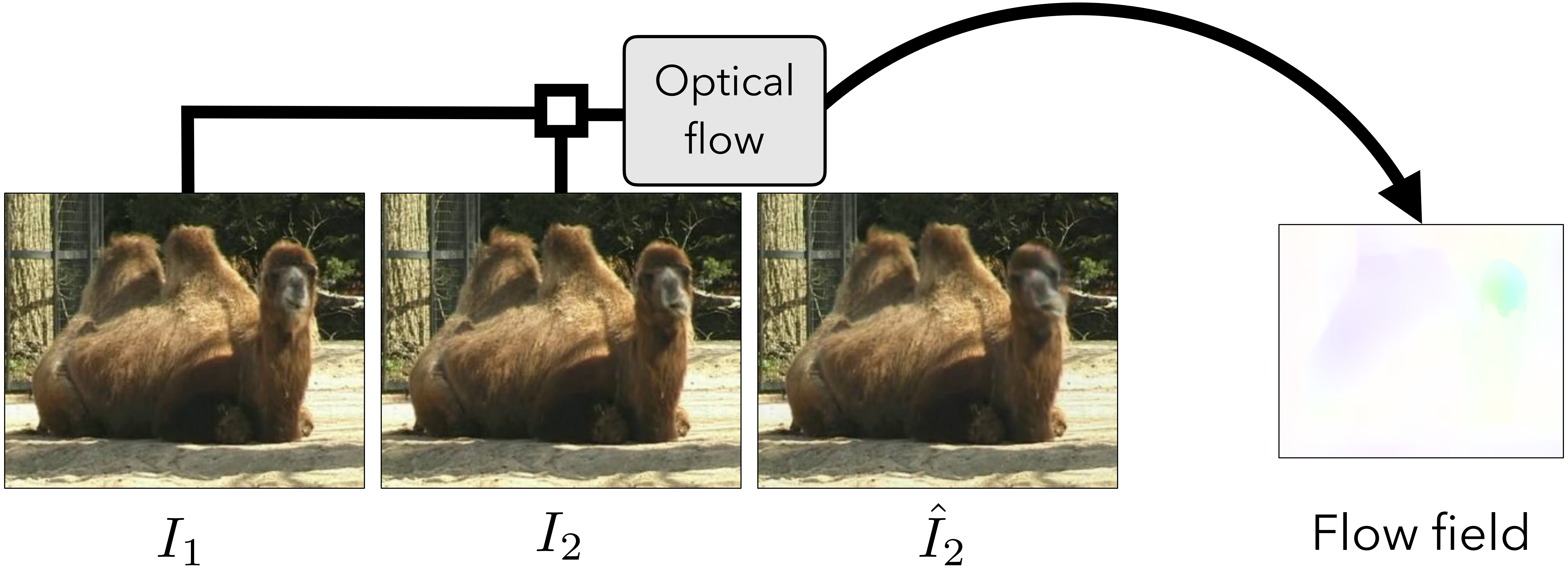
Motion magnification model



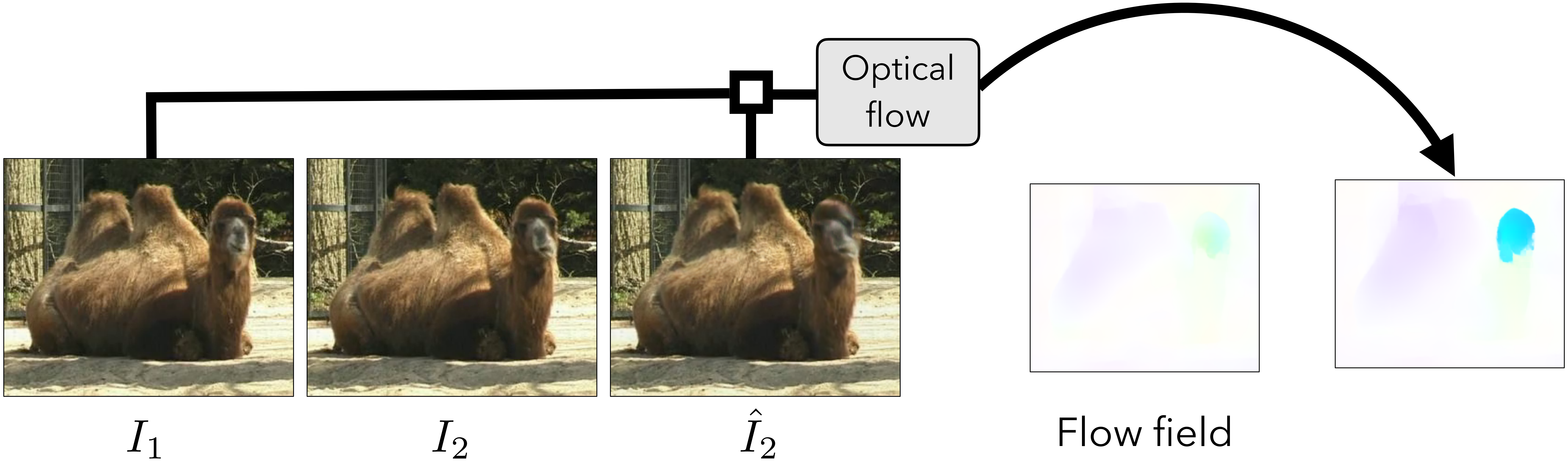
Input frames

Magnified 2nd frame

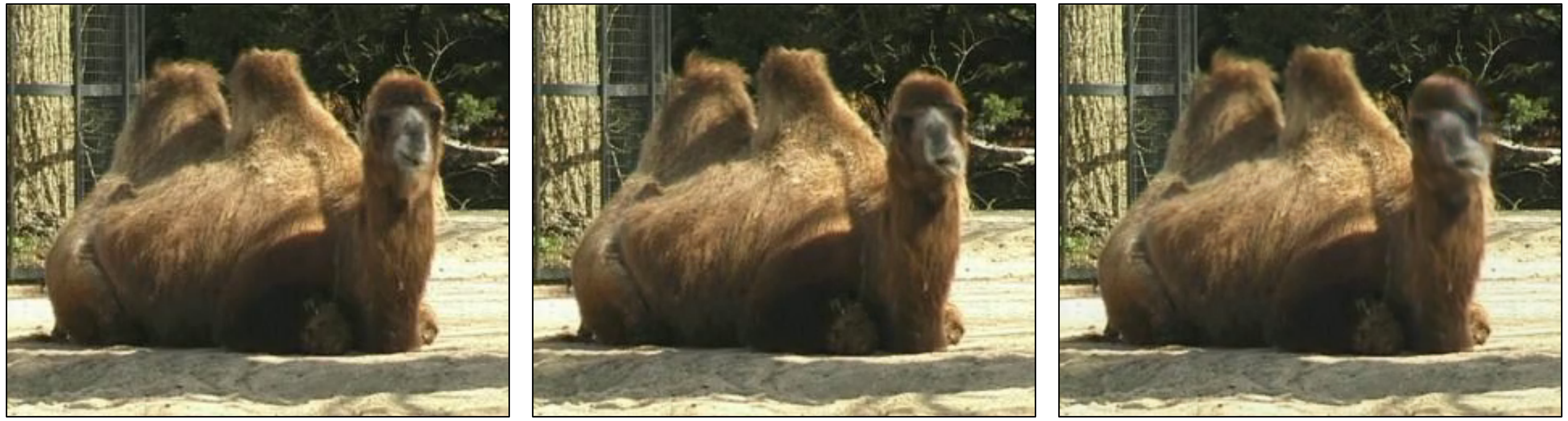
Magnification loss



Magnification loss



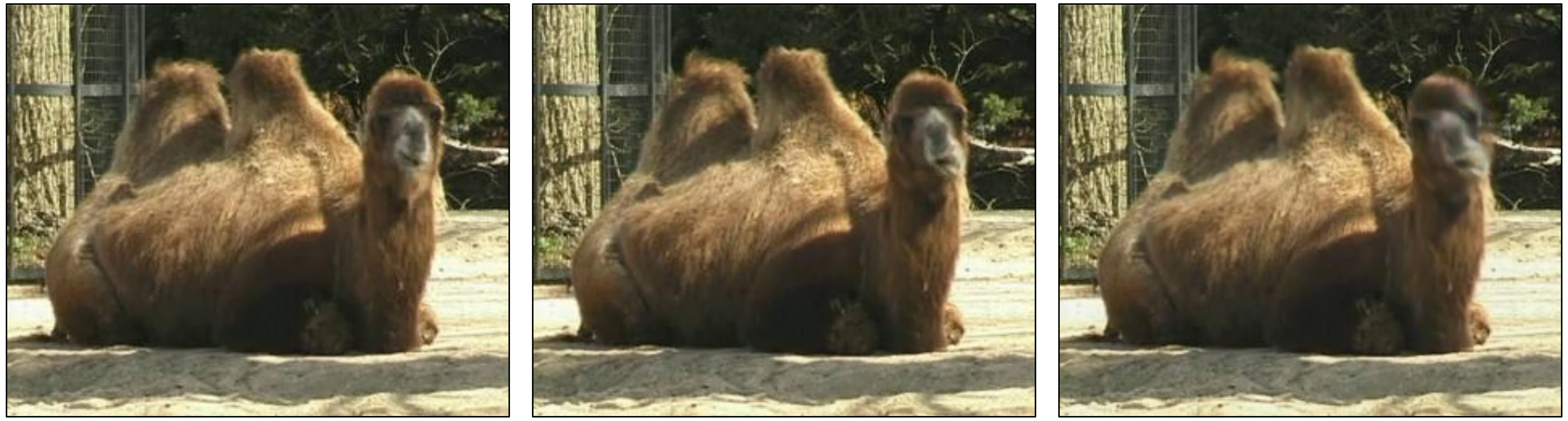
Motion magnification loss



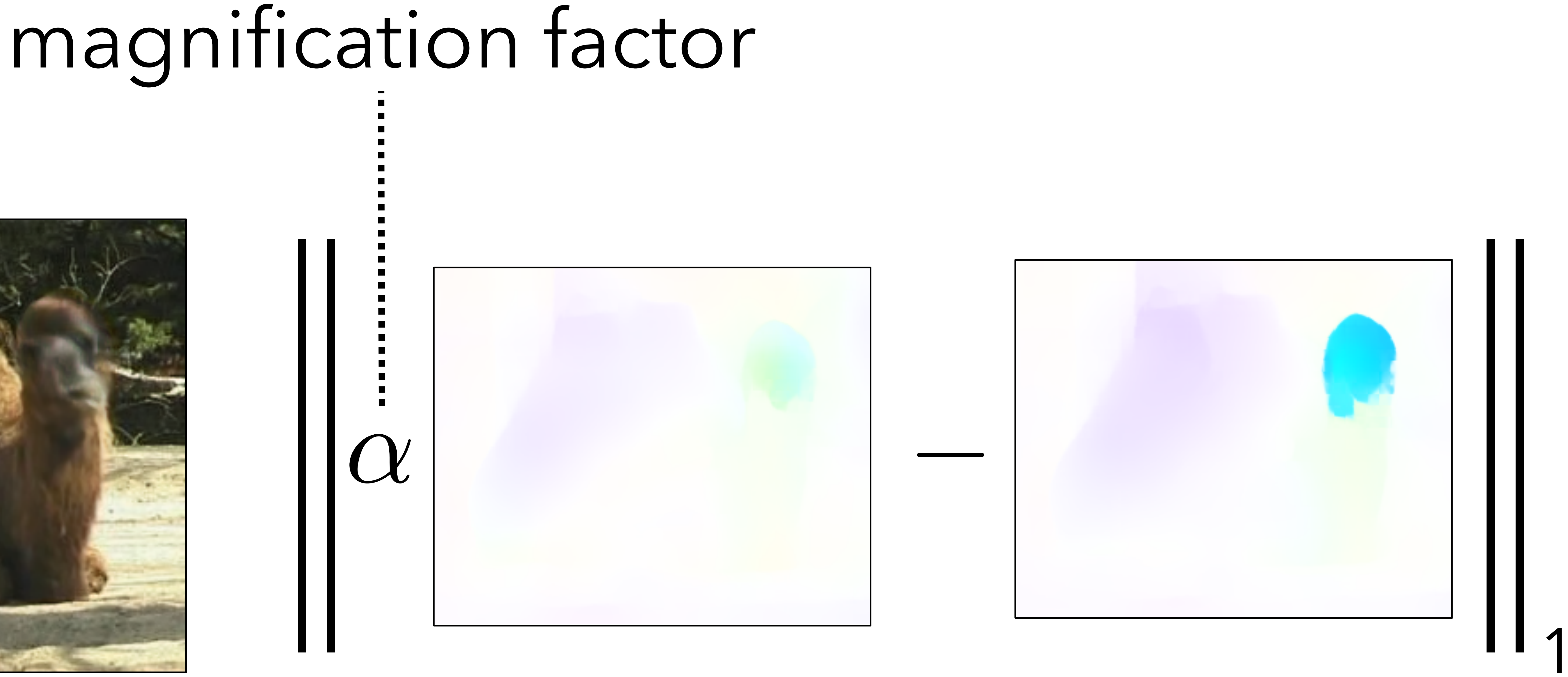
I_1 I_2 $\hat{I}_2$

$\Big\| \alpha$  $-$  $\Big\|_1$

Motion magnification loss



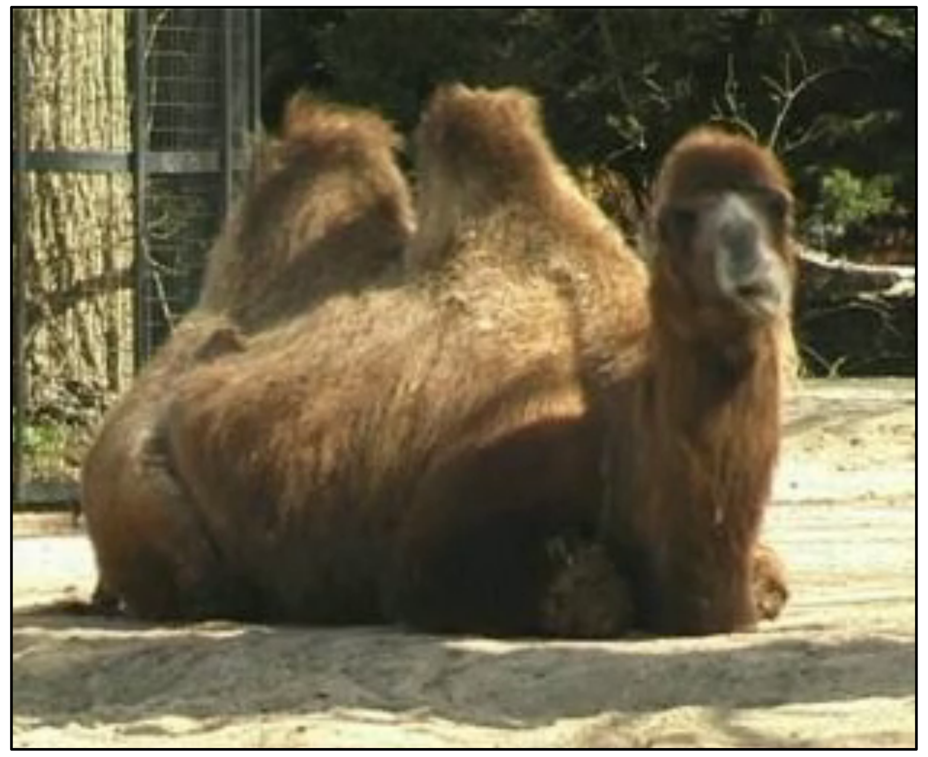
I_1 I_2 $\hat{I}_2$



Backprop through the optical flow during training!

Motion magnification loss

I_1

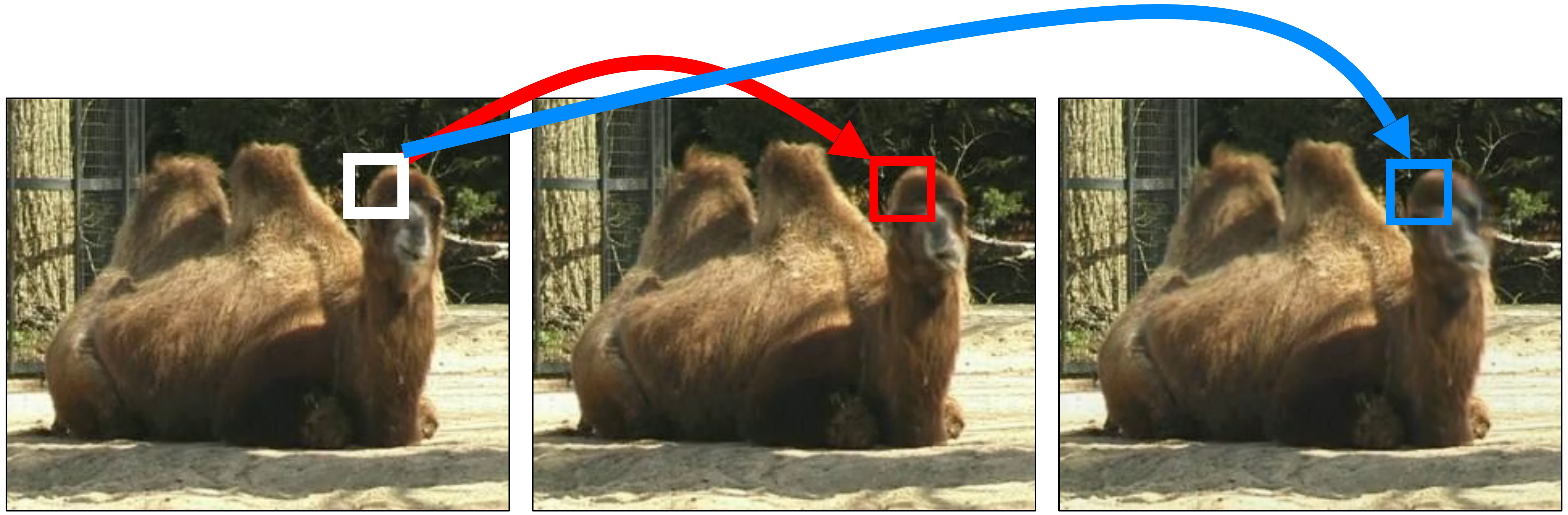


I_2

$\hat{I}_2$

$\mathcal{L}_{mag} = \Big\| \alpha$  $-$  $\Big\|_1$

Motion magnification loss



I_1 I_2 $\hat{I}_2$

$\mathcal{L}_{mag} = \Big\| \alpha$  $-$  $\Big\|_1$

Motion magnification loss

Original

Ours

Targeting White Cat (α=20)

Targeting Black Cat (α=20)

No Targeting (α=15)

Ignoring Arm (α=15)


Original
5x Magnified


Original
5x Magnified


Original
30x Magnified

Next lecture: light and color