Lucas-Kanade Optical Flow

Computer Vision 16-385

Carnegie Mellon University (Kris Kitani)

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

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

spatial derivative

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

optical flow

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

temporal derivative

How can we use the brightness constancy equation to estimate the optical flow?

unknown

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

known

We need at least ____ equations to solve for 2 unknowns.

unknown

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

known

Where do we get more equations (constraints)?

Where do we get more equations (constraints)?

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

Assume that the surrounding patch (say 5x5) has **constant flow**

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us $\boxed{}$ equations

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$
\begin{gathered}
I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v = -I_t(\boldsymbol{p}_1) \\
I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v = -I_t(\boldsymbol{p}_2) \\
\vdots \\
I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v = -I_t(\boldsymbol{p}_{25})
\end{gathered}
$$

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\begin{bmatrix} I_x(\boldsymbol{p}_1) & I_y(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2) & I_y(\boldsymbol{p}_2) \\ \vdots & \vdots \\ I_x(\boldsymbol{p}_{25}) & I_y(\boldsymbol{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\boldsymbol{p}_1) \\ I_t(\boldsymbol{p}_2) \\ \vdots \\ I_t(\boldsymbol{p}_{25}) \end{bmatrix}$$

Matrix form

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\underbrace{\begin{bmatrix} I_x(\boldsymbol{p}_1) & I_y(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2) & I_y(\boldsymbol{p}_2) \\ \vdots & \vdots \\ I_x(\boldsymbol{p}_{25}) & I_y(\boldsymbol{p}_{25}) \end{bmatrix}}_{\substack{A \\ 25 \times 2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{x \\ 2 \times 1}} = - \underbrace{\begin{bmatrix} I_t(\boldsymbol{p}_1) \\ I_t(\boldsymbol{p}_2) \\ \vdots \\ I_t(\boldsymbol{p}_{25}) \end{bmatrix}}_{\substack{b \\ 25 \times 1}}$$

How many equations? How many unknowns? How do we solve this?

Least squares approximation

$$\hat{x} = \arg\min_{x} ||Ax - b||^2$$ is equivalent to solving $$A^\top A\hat{x} = A^\top b$$

Least squares approximation

$$\hat{x} = \arg\min_x ||Ax - b||^2$$ is equivalent to solving $$A^\top A\hat{x} = A^\top b$$

To obtain the least squares solution solve:

$$\underbrace{\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}}_{A^\top A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\hat{x}} = \underbrace{- \begin{bmatrix} \sum_{p \in P} I_x I_t \\ \sum_{p \in P} I_y I_t \end{bmatrix}}_{A^\top b}$$

where the summation is over each pixel ***p*** in patch ***P***

$$x = (A^\top A)^{-1} A^\top b$$

Least squares approximation

$$\hat{x} = \arg\min_x ||Ax - b||^2 \text{ is equivalent to solving } A^\top A\hat{x} = A^\top b$$

To obtain the least squares solution solve:

$$\begin{array}{ccc} A^\top A & \hat{x} & A^\top b \end{array}$$

$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{p \in P} I_x I_t \\ \sum_{p \in P} I_y I_t \end{bmatrix}$$

where the summation is over each pixel $\boldsymbol{p}$ in patch $\boldsymbol{P}$

Sometimes called '**Lucas-Kanade Optical Flow'**
(special case of the LK method with a translational warp model)

When is this solvable?

$$A^\top A\hat{x} = A^\top b$$

$A^\top A$ should be invertible

$A^\top A$ should not be too small

λ_1 and λ_2 should not be too small

$A^\top A$ should be well conditioned

λ_1/λ_2 should not be too large (λ_1=larger eigenvalue)

Where have you seen this before?

$$A^{\top}A = \begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix}$$

Where have you seen this before?

$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

Harris Corner Detector!

Implications

- Corners are when $\lambda 1$, $\lambda 2$ are big; this is also when Lucas-Kanade optical flow works best

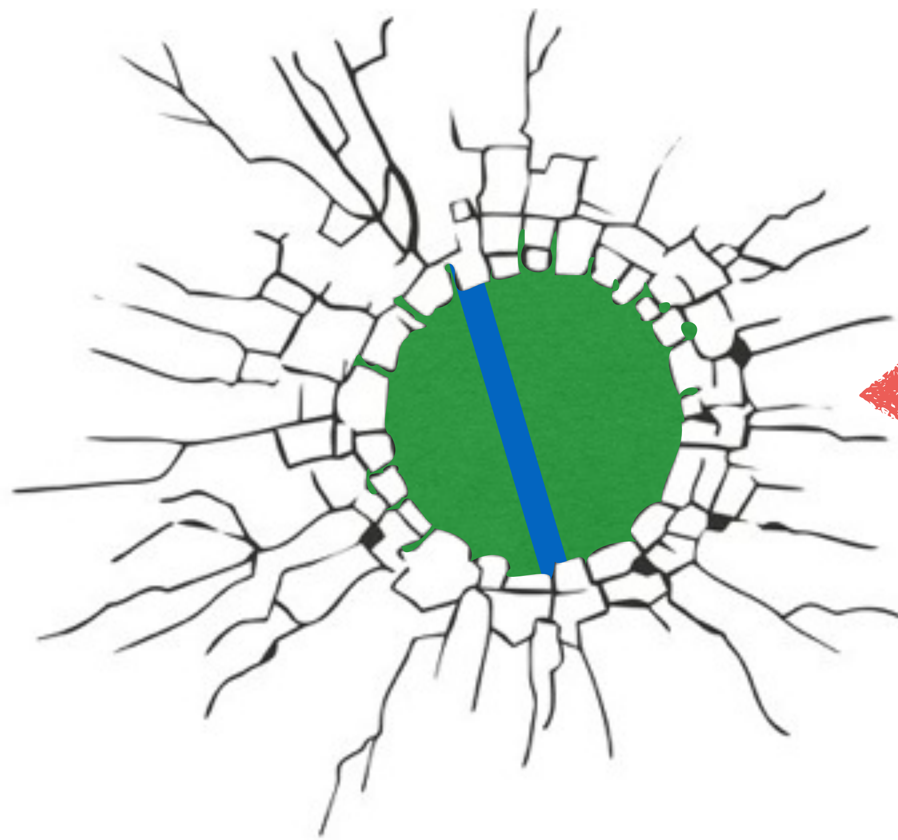
- Corners are regions with two different directions of gradient (at least)

- Corners are good places to compute flow!

What happens when you have no 'corners'?

You want to compute optical flow.
What happens if the image patch contains only a line?

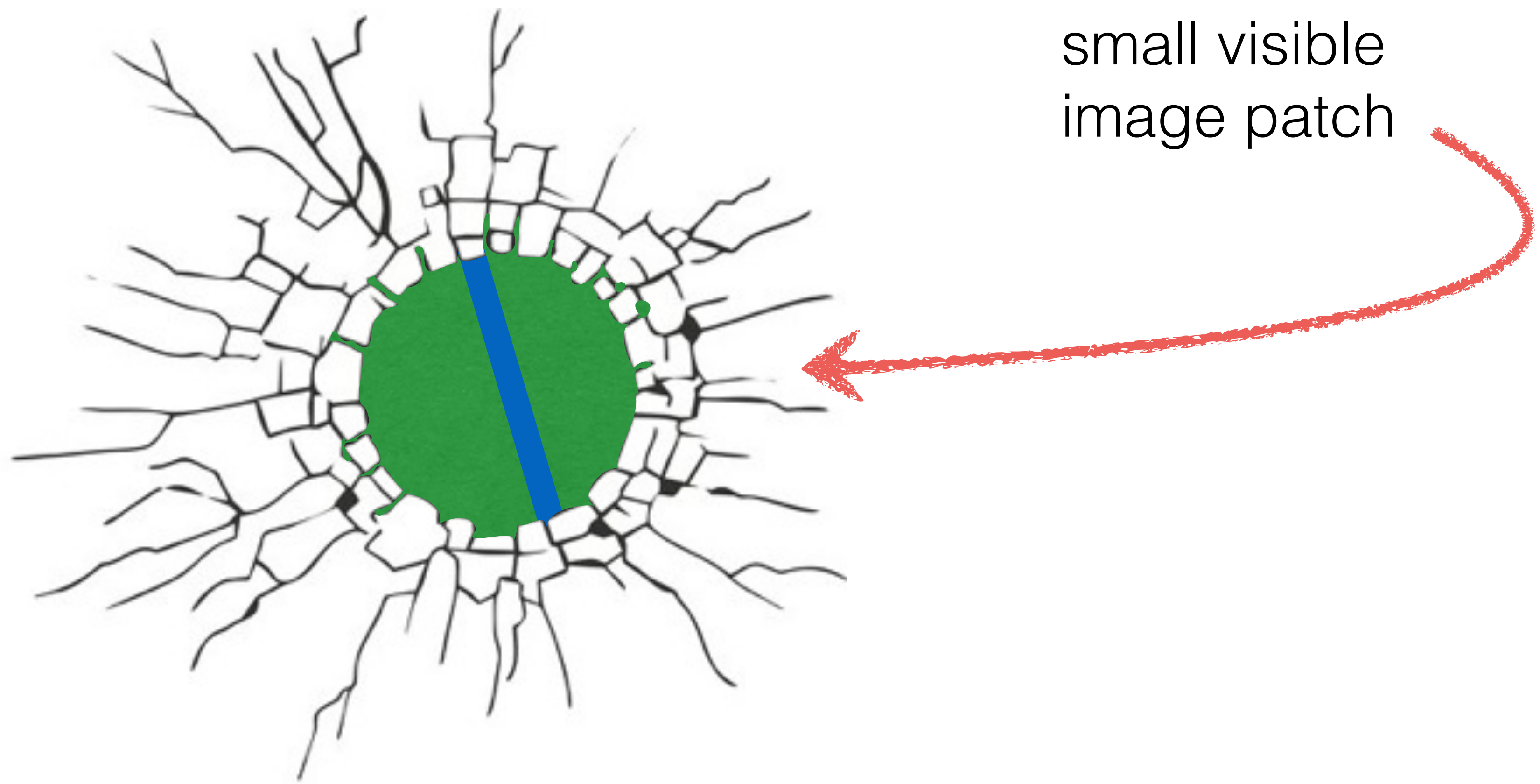
Aperture Problem



small visible
image patch

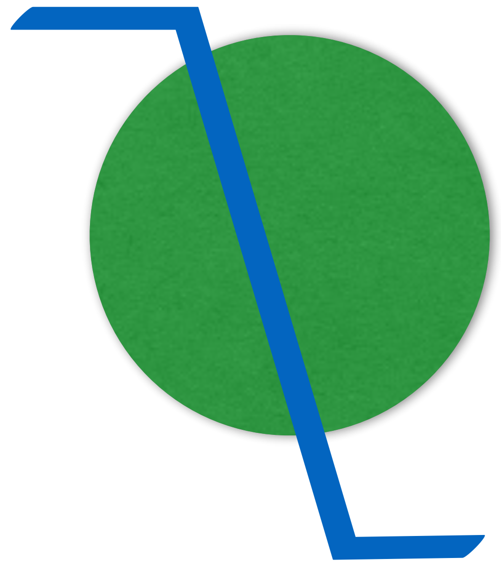
In which direction is the line moving?

Aperture Problem

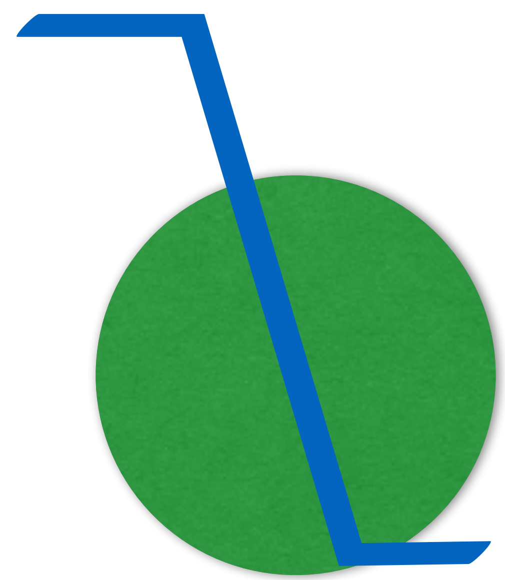


In which direction is the line moving?

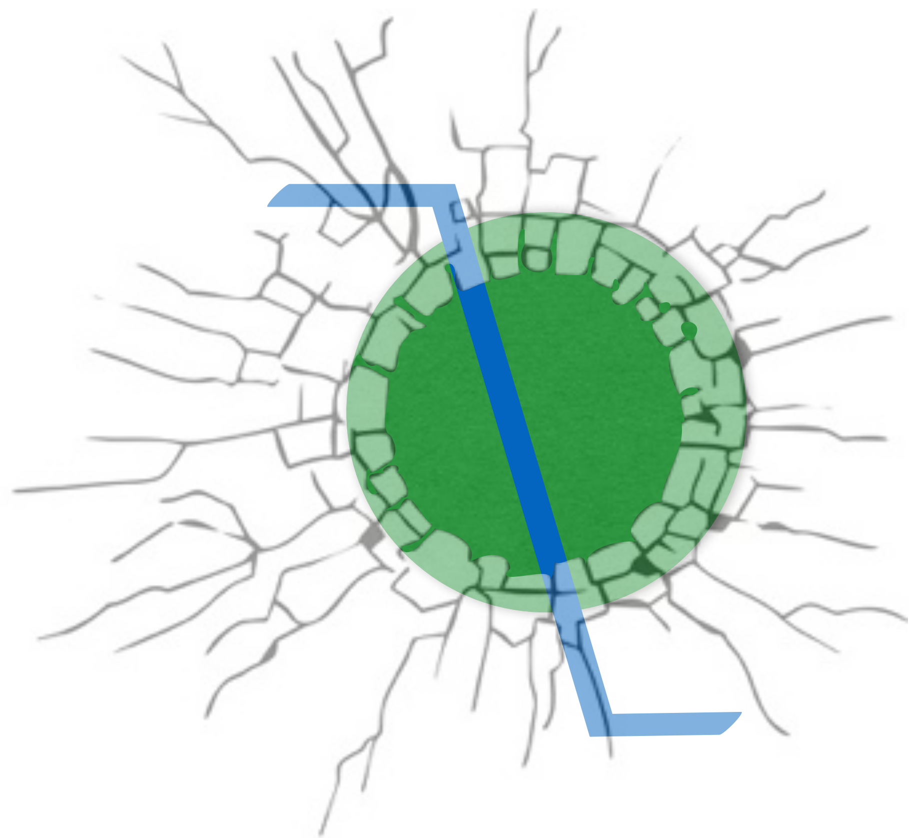
Aperture Problem



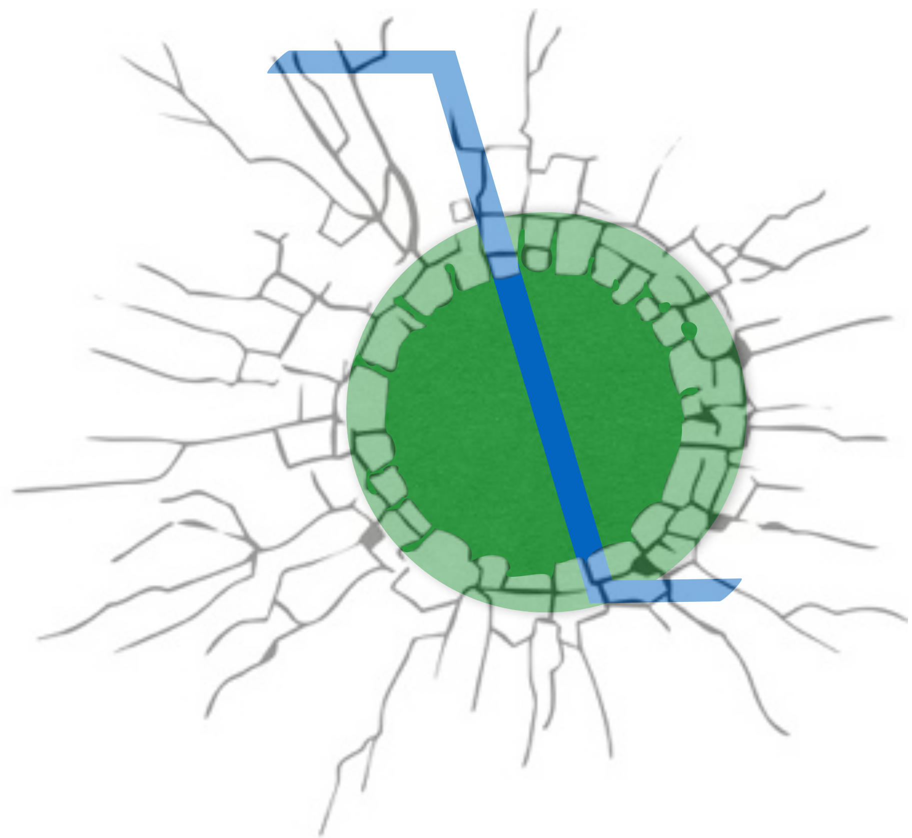
Aperture Problem

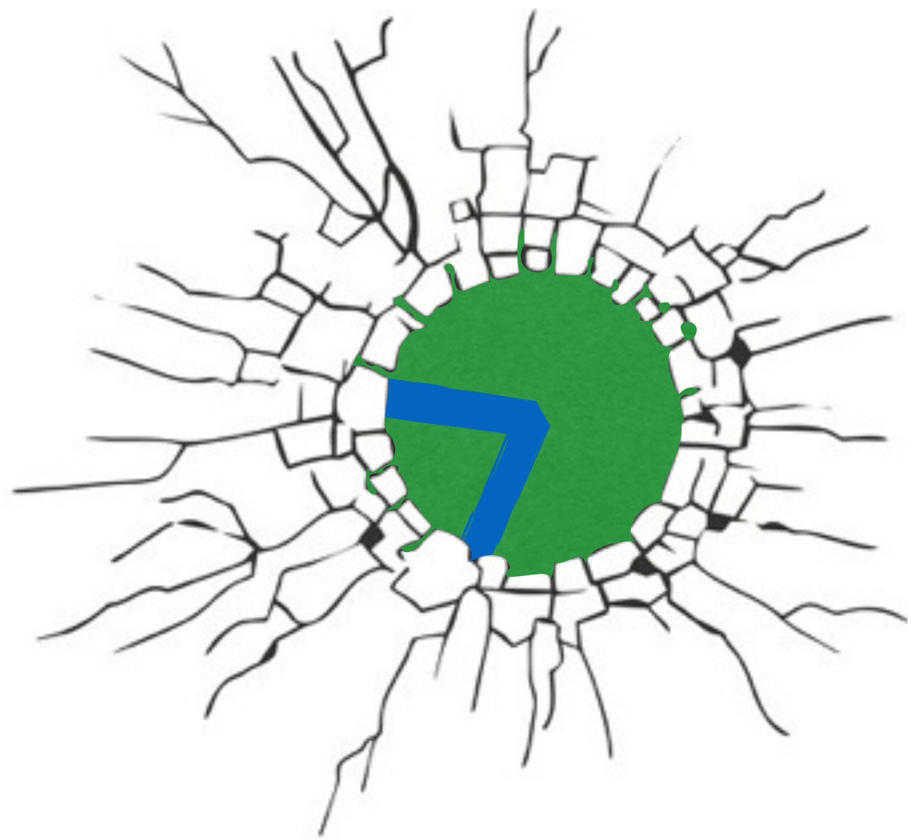


Aperture Problem

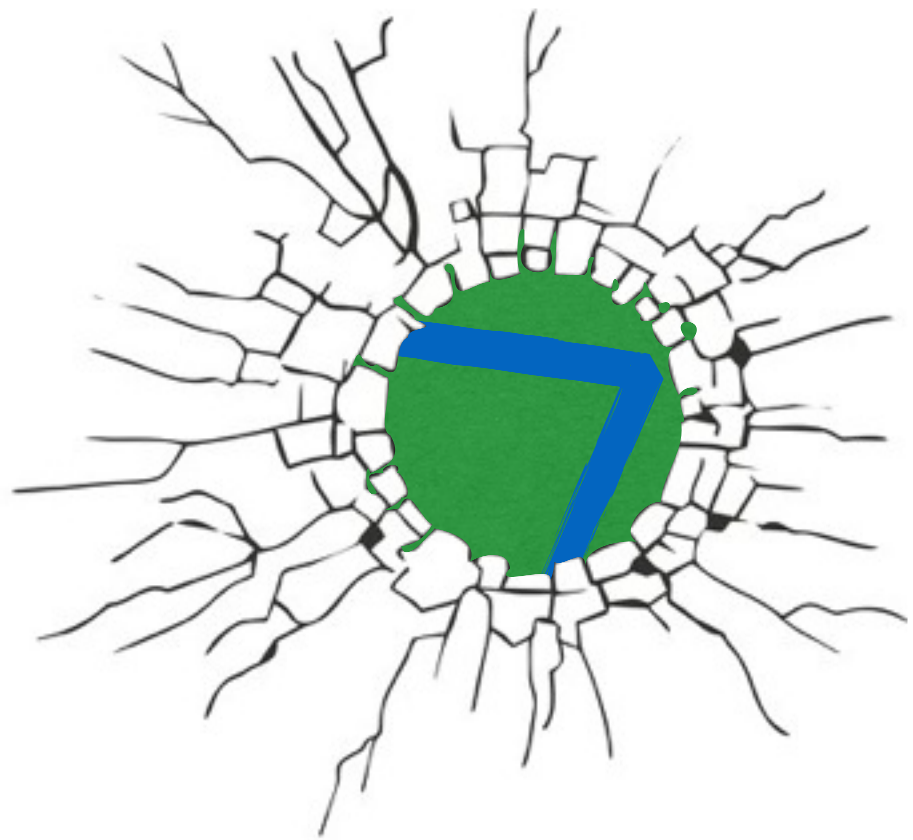


Aperture Problem





Want patches with different gradients to the avoid aperture problem



Want patches with different gradients to the avoid aperture problem

H(x,y) = y

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5

optical flow: (1,1)

I(x,y)

-	-	-	-
-	1	1	1
-	2	2	2
-	3	3	3
-	4	4	4

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

Compute gradients

$$I_x(3,3) = 0$$
$$I_y(3,3) = 1$$
$$I_t(3,3) = I(3,3) - H(3,3) = -1$$

Solution:

$$v = 1$$

We recover the v of the optical flow but not the u.

This is the aperture problem.

Note:
The Lucas-Kanade method is a general framework for image matching (not only optical flow)

Applied to optical flow:
Assume that the flow over a small image patch is constant

Horn-Shunck Optical Flow (1981)

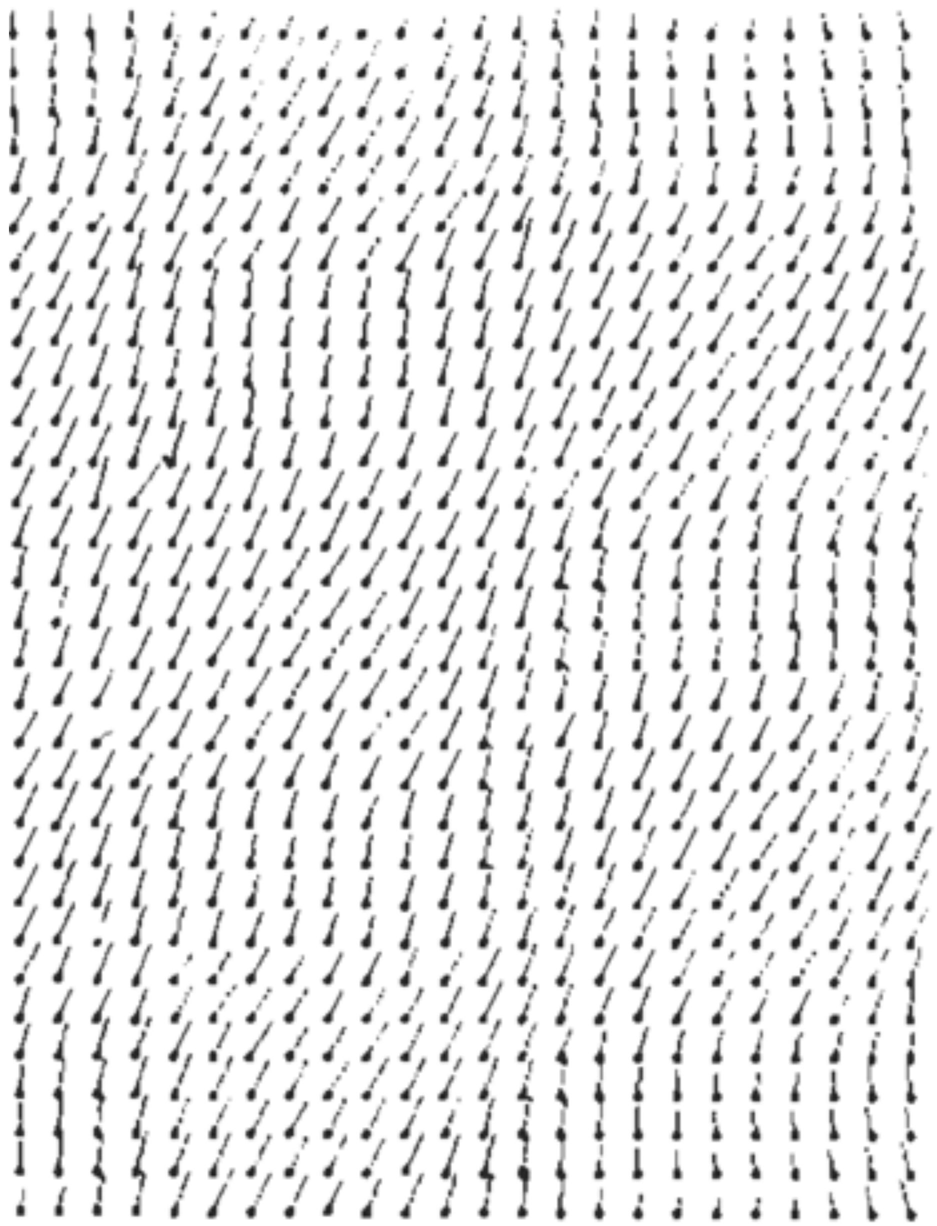
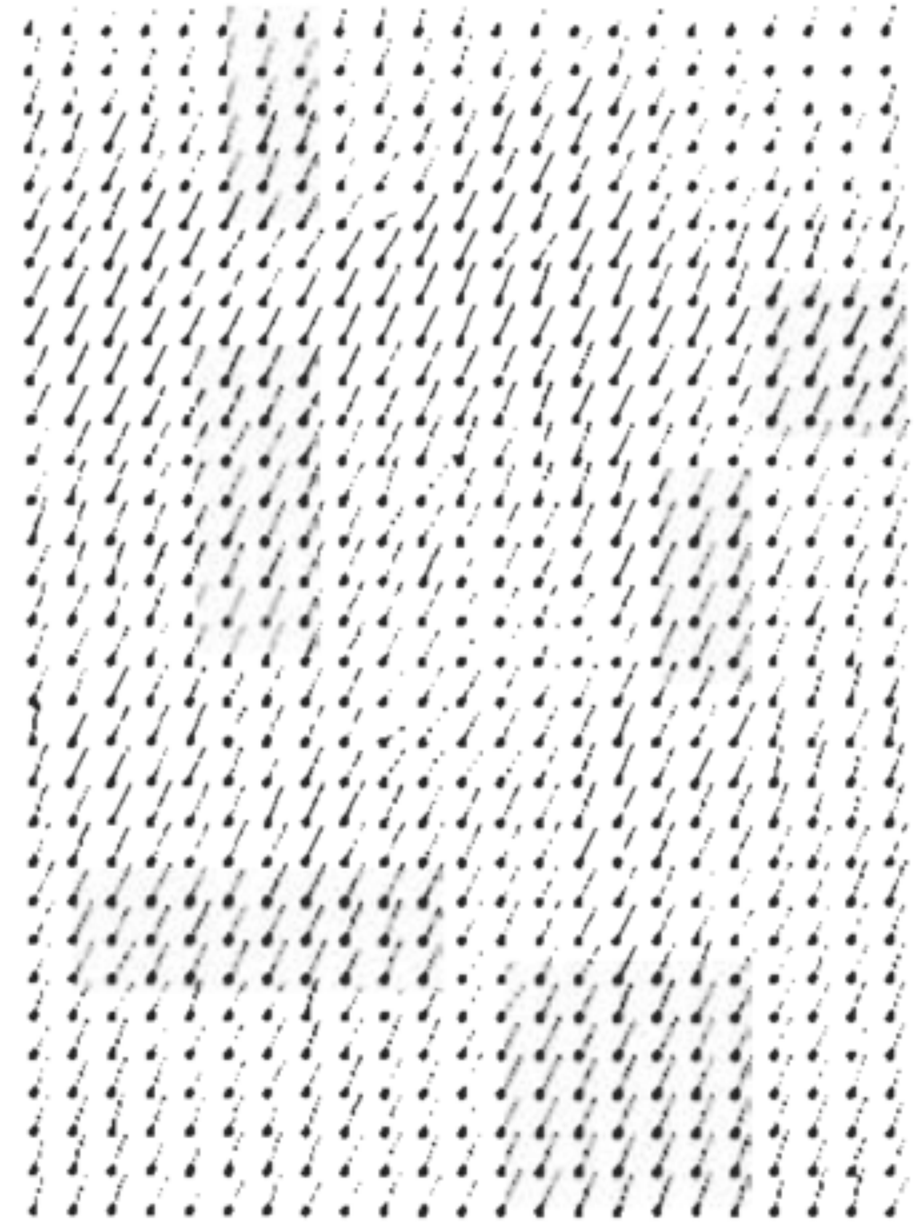
'smooth' flow

Constraint:
Flow of neighboring pixels should be smooth

Lucas-Kanade Optical Flow (1981)

'constant' flow

Contraint:
Flow over a small patch should be the same

Horn-Schunck

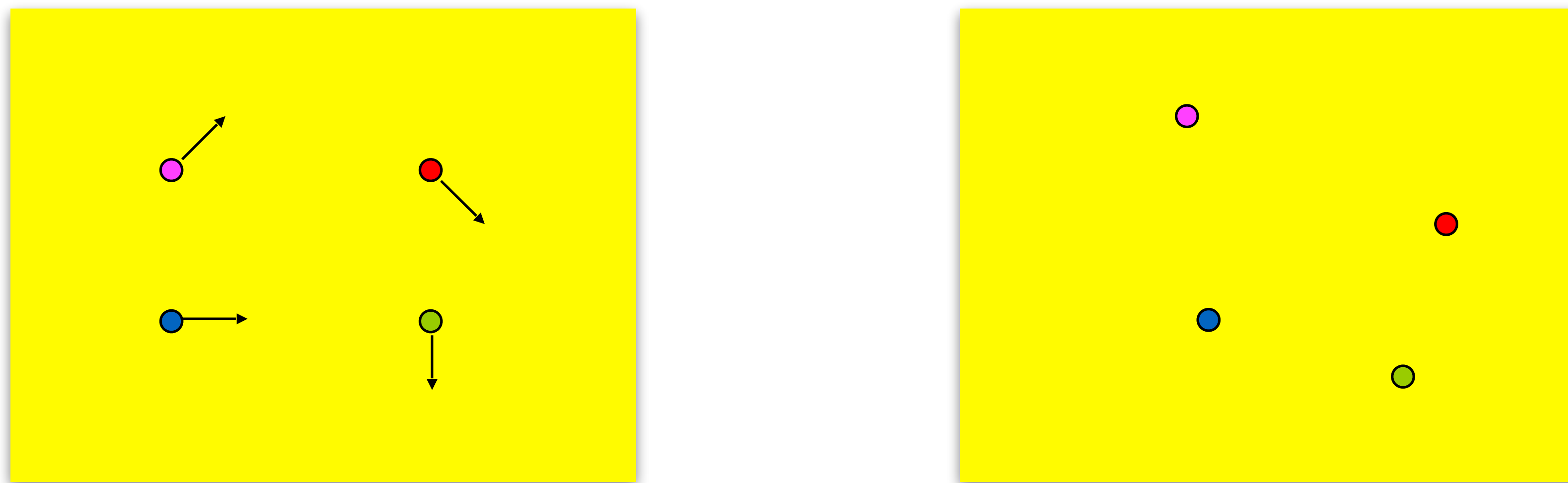
16-385 Computer Vision

Carnegie Mellon University (Kris Kitani)

Horn-Schunck Optical Flow (1981)	**Lucas-Kanade Optical Flow (1981)**
brightness constancy small motion	method of differences
'smooth' flow (flow can vary from pixel to pixel)	**'constant' flow** (flow is constant for all pixels)
global method (dense)	local method (sparse)

Optical Flow

(Problem definition)



$I(x, y, t)$ $\quad$ $I(x, y, t')$

Estimate the motion
(flow) between these
two consecutive images

How is this different from estimating a 2D transform?

Key Assumptions

(unique to optical flow)

Color Constancy

(Brightness constancy for intensity images)

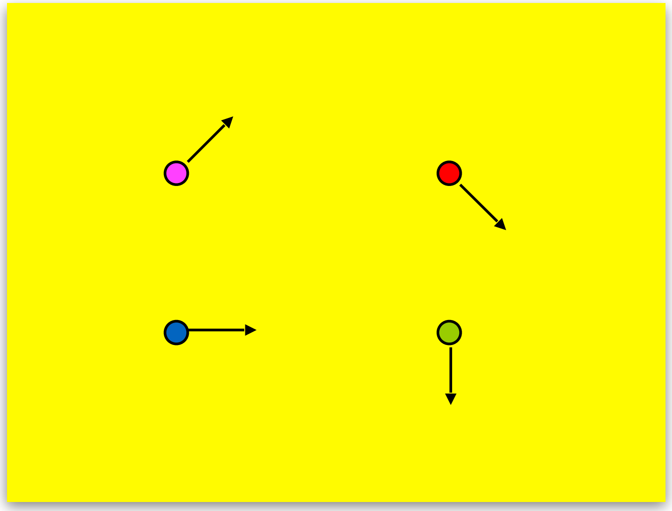
Implication: allows for pixel to pixel comparison
(not image features)

Small Motion

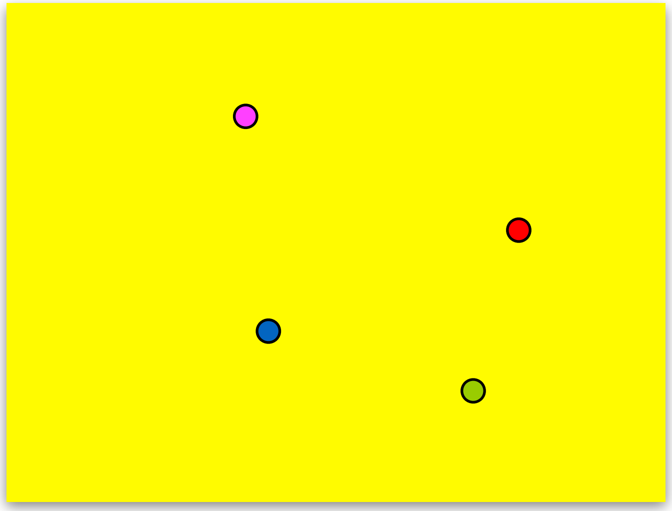
(pixels only move a little bit)

Implication: linearization of the brightness
constancy constraint

Approach



$I(x, y, t)$

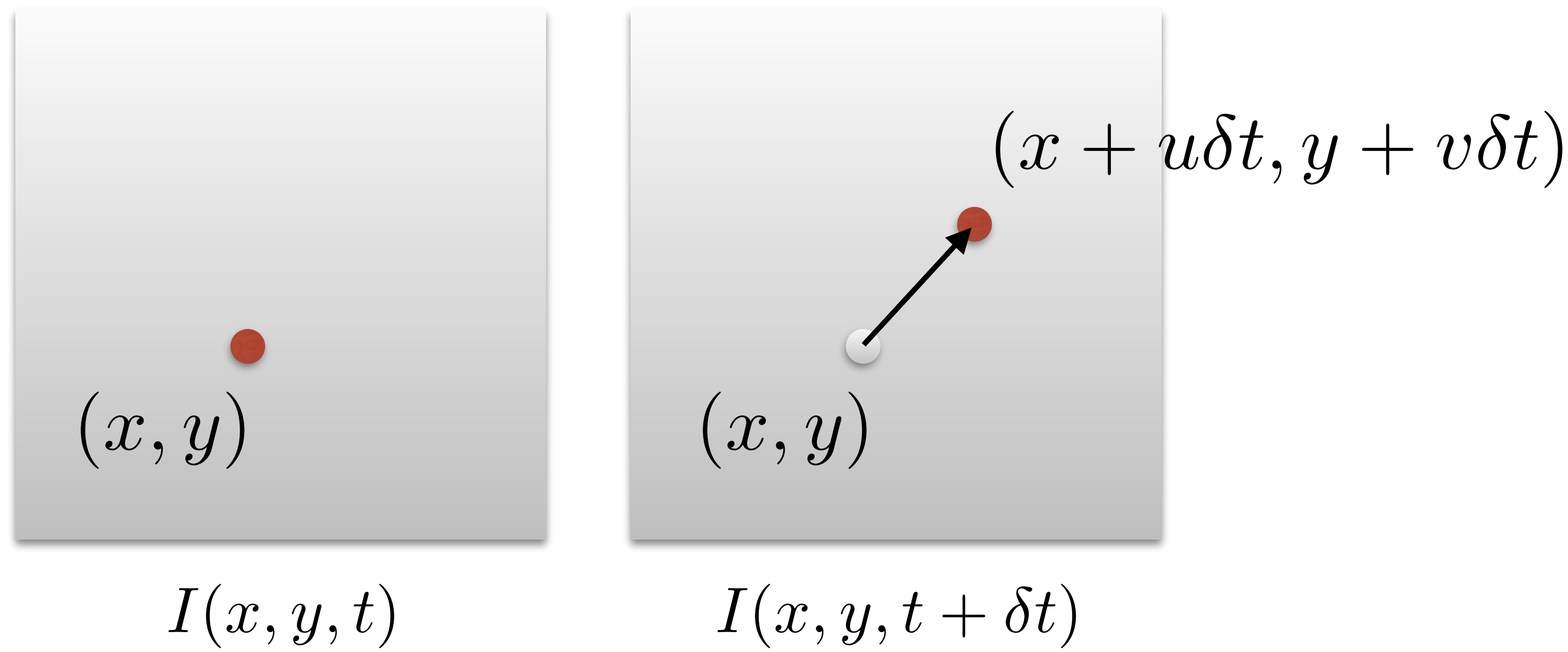


$I(x, y, t')$

Look for nearby pixels with the same color

(small motion) (color constancy)

Brightness constancy



Optical flow (velocities): (u, v) Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

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

For a really small time step…

Optical Flow Constraint equation

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

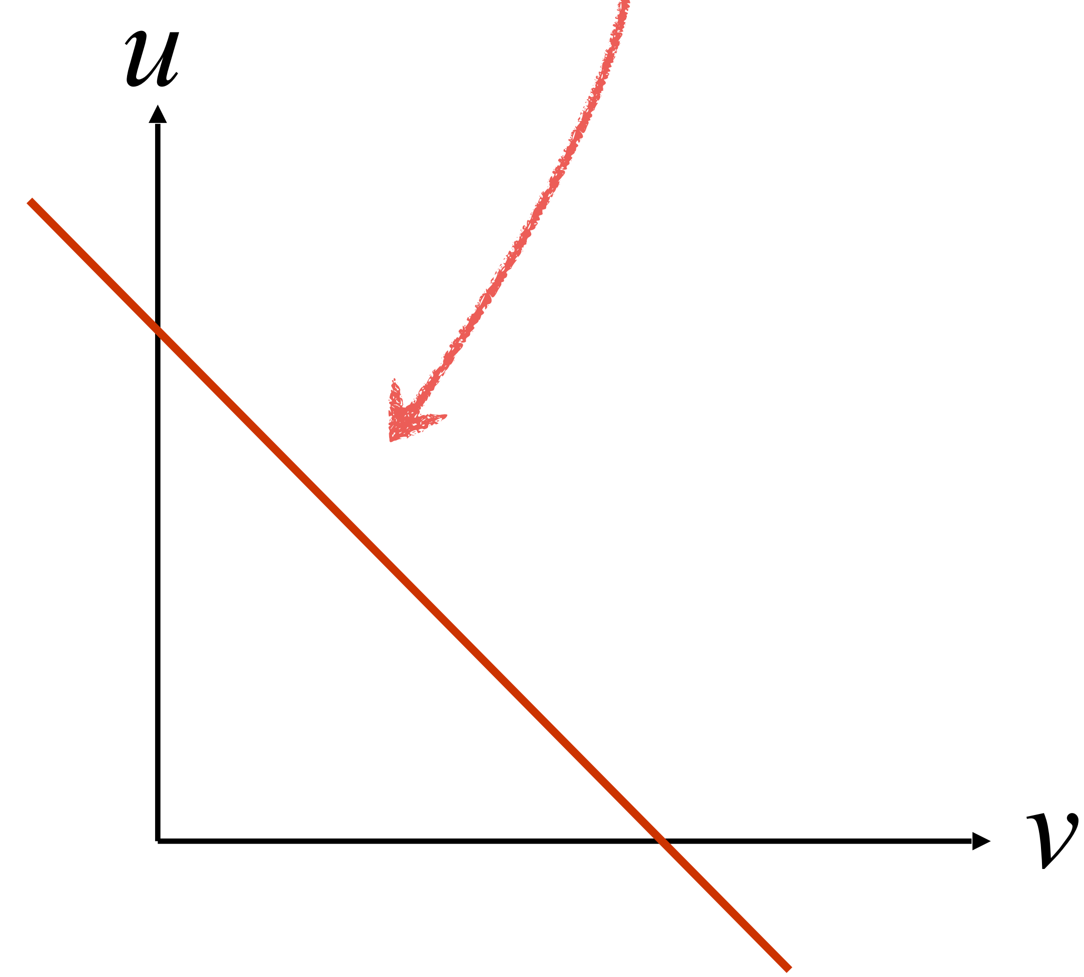
$$\cancel{I(x,y,t)} + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = \cancel{I(x,y,t)}$$

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

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

Solution lies on a straight line

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

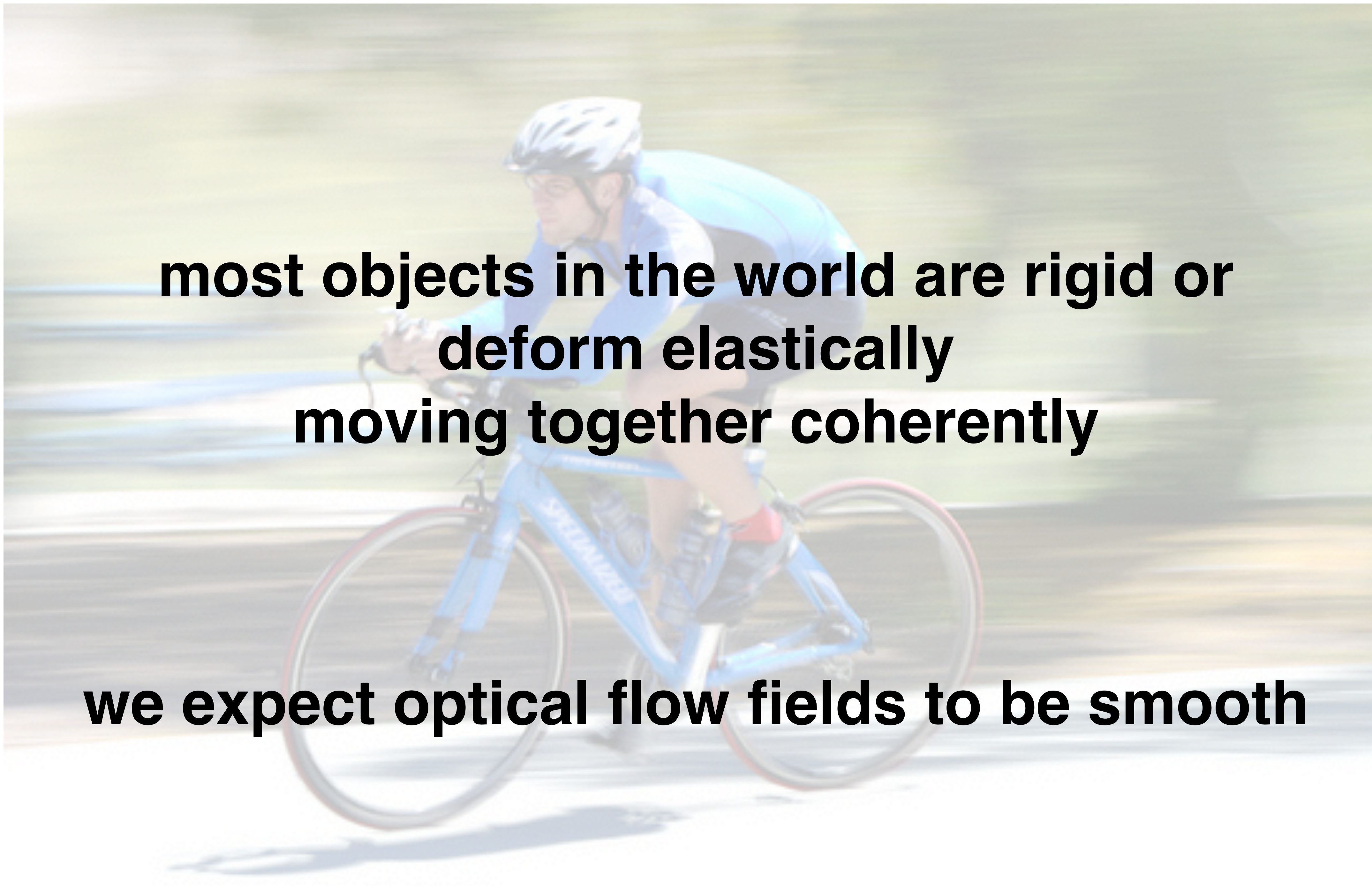


The solution cannot be determined uniquely with a single constraint (a single pixel)

Where can we get an additional constraint?

Horn-Schunck Optical Flow (1981)	**Lucas-Kanade Optical Flow (1981)**
brightness constancy	method of differences
small motion	
'smooth' flow (flow can vary from pixel to pixel)	**'constant' flow** (flow is constant for all pixels)
global method	local method

Smoothness



most objects in the world are rigid or deform elastically
moving together coherently

we expect optical flow fields to be smooth

Key idea
(of Horn-Schunck optical flow)

Enforce
brightness constancy

Enforce
smooth flow field

to compute optical flow

Enforce brightness constancy

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

For every pixel,

$$\min_{u,v} \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

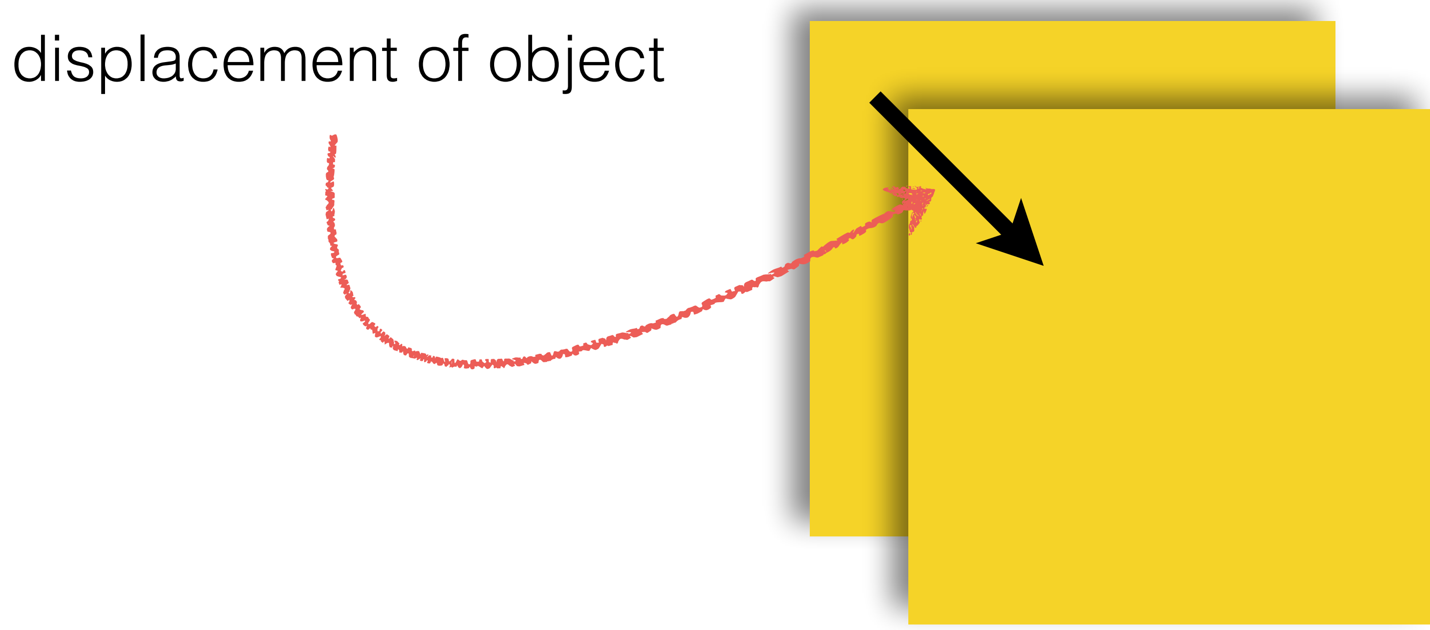
Enforce brightness constancy

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

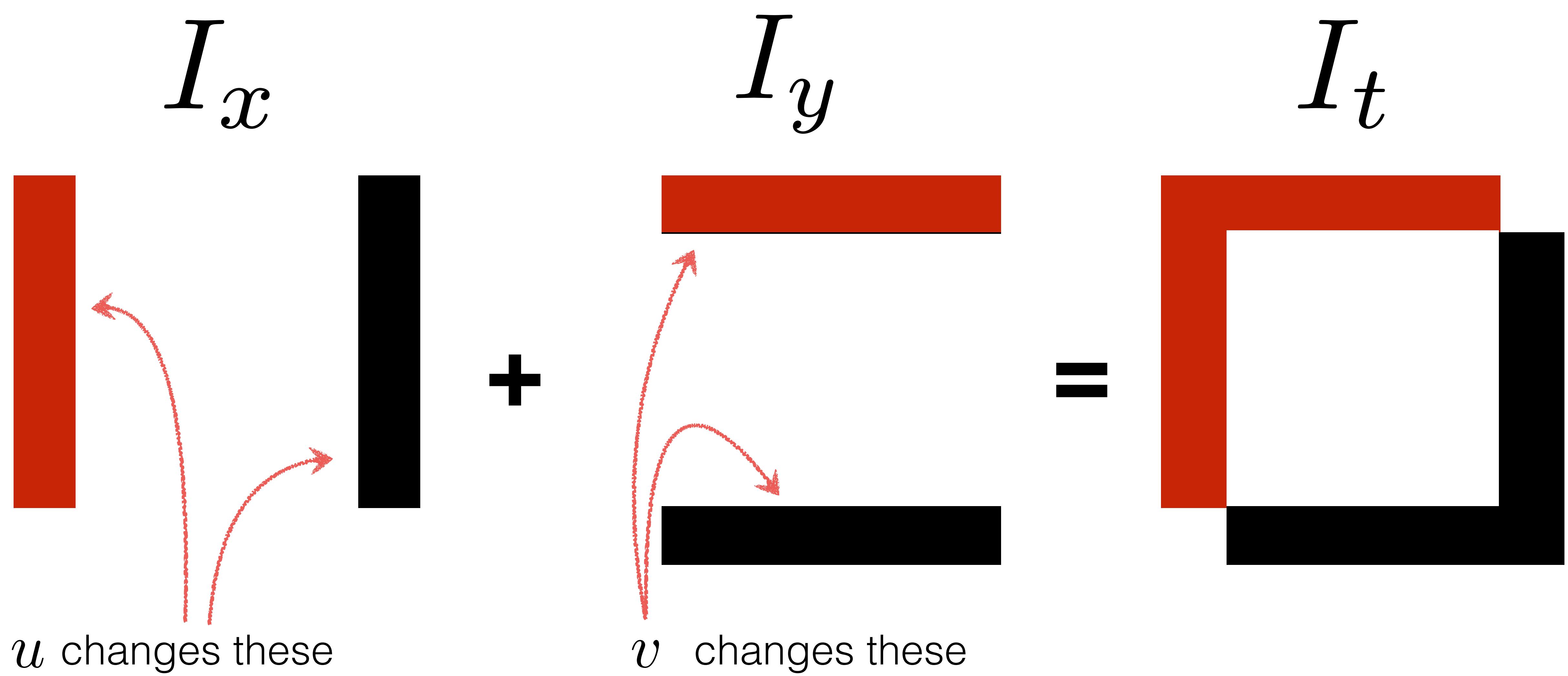
For every pixel,

$$\min_{u,v} \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

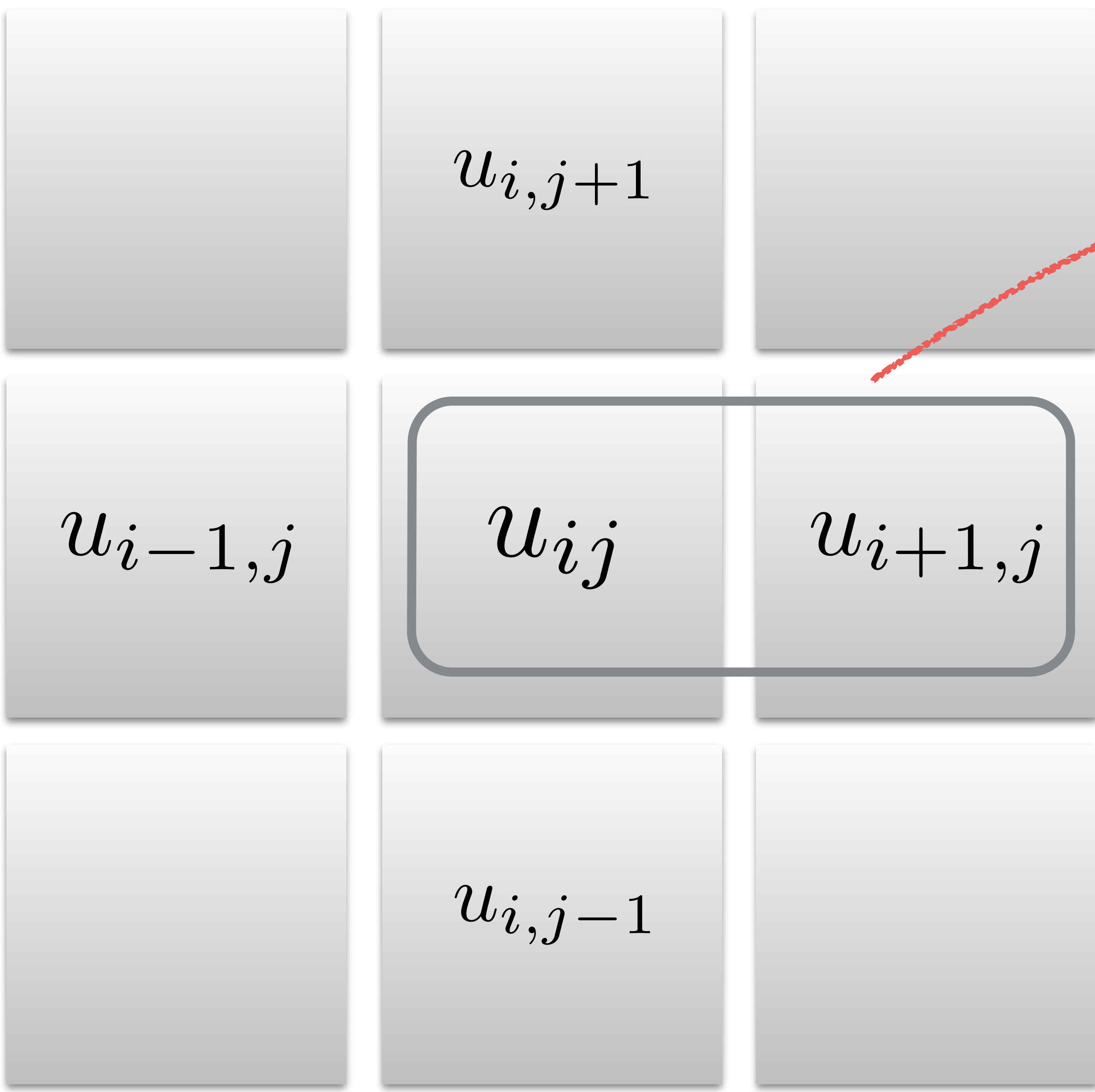
lazy notation for $I_x(i,j)$



Find the optical flow such that it satisfies:



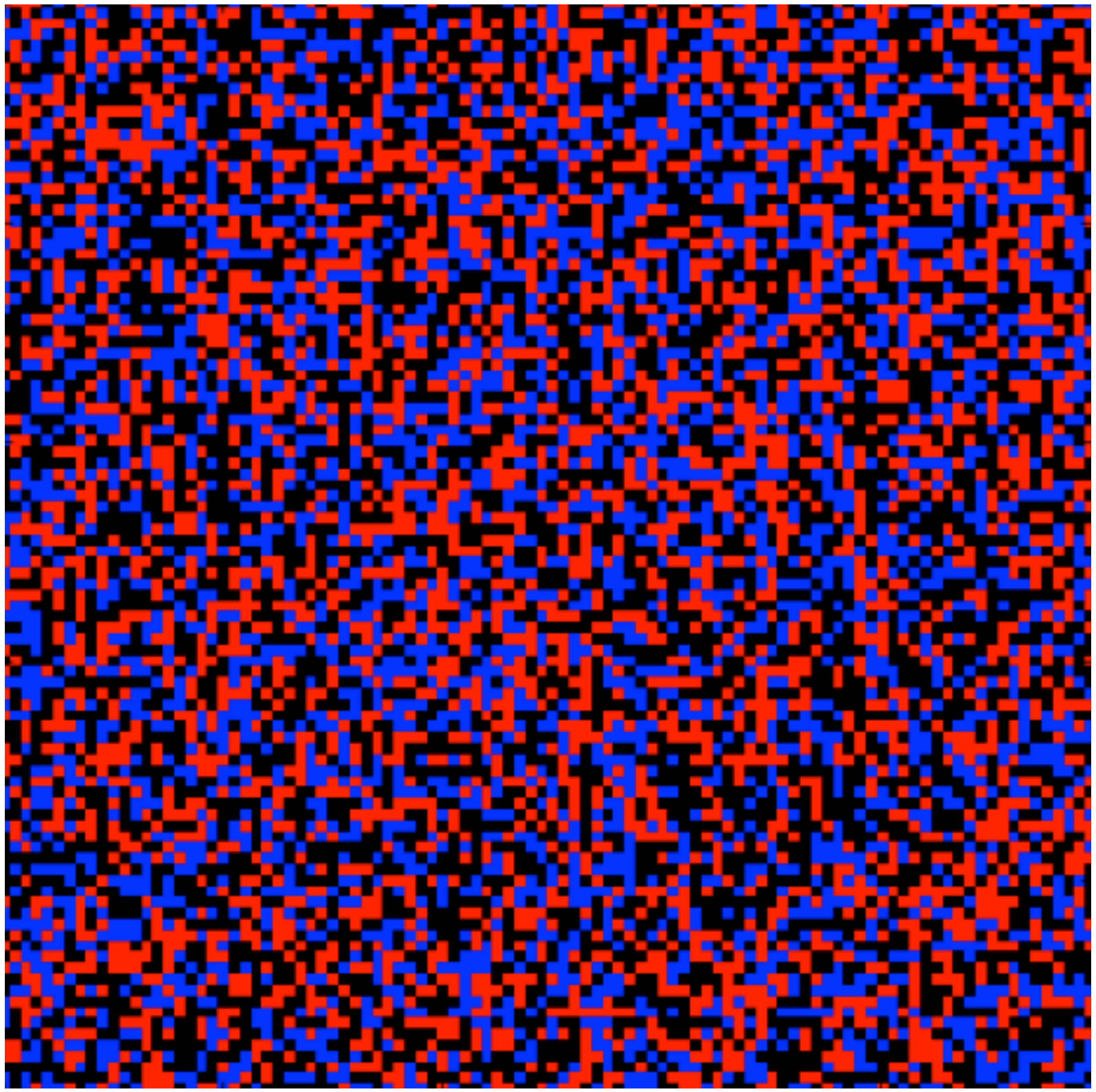
Enforce smooth flow field



u-component of flow

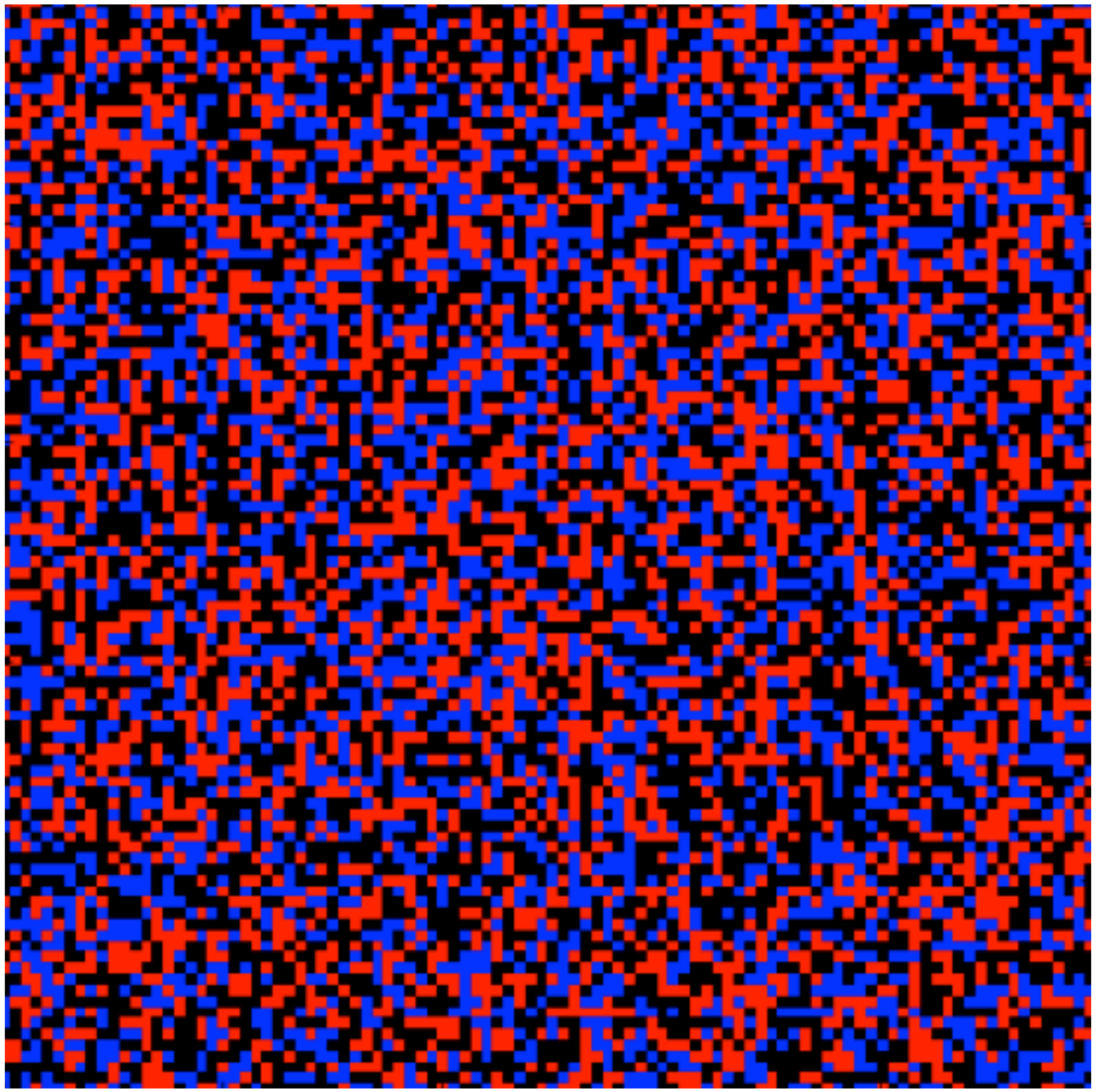
$$\min_{\boldsymbol{u}} (u_{i,j} - u_{i+1,j})^2$$

Which flow field is more likely?



$$\sum_{ij}(u_{ij} - u_{i+1,j})^2 \quad ? \quad \sum_{ij}(u_{ij} - u_{i+1,j})^2$$

Which flow field is more likely?



less likely

more likely

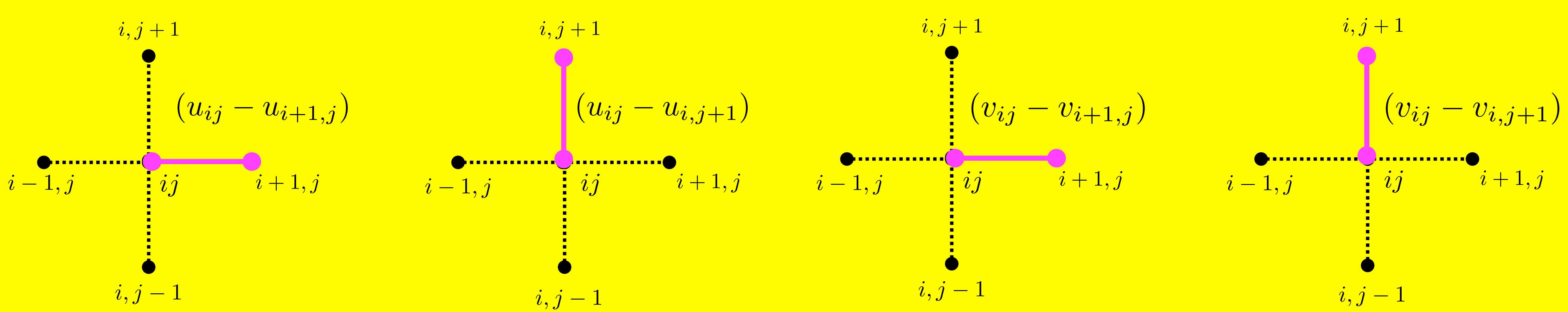
optical flow objective function

Optical flow constraint

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

Smoothness constraint

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



Horn-Schunck optical flow

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{ij} \left\{ E_d(i,j) + \lambda E_s(i,j) \right\}$$

where

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

and

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

How do we solve this minimization problem?

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{ij} \left\{ E_d(i,j) + \lambda E_s(i,j) \right\}$$

How do we solve this minimization problem?

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{ij} \left\{ E_d(i,j) + \lambda E_s(i,j) \right\}$$

Compute partial derivative, derive update equations (gradient decent)

Compute the partial derivatives of this huge sum!

$$\sum_{ij} \left\{ \frac{1}{4} \left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right] + \lambda \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2 \right\}$$

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

it's not so bad…

$$\frac{\partial E}{\partial u_{kl}} =$$

how many terms depend on k and l?

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

it's not so bad…

$$\frac{\partial E}{\partial u_{kl}}=2(u_{kl}-\bar{u}_{kl})+2\lambda(I_x u_{kl}+I_y v_{kl}+I_t)I_x$$

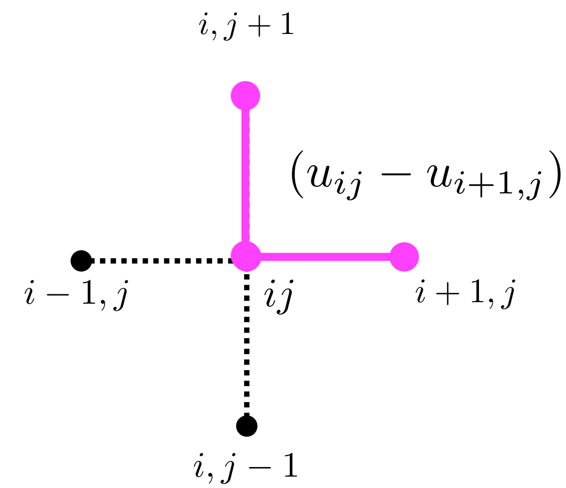
how many terms depend on k and l?

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

$$(u_{ij}^2-2u_{ij}u_{i+1,j}+u_{i+1,j}^2) \qquad (u_{ij}^2-2u_{ij}u_{i,j+1}+u_{i,j+1}^2)$$

(variable will appear four times in sum)



$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl}-\bar{u}_{kl}) + 2\lambda(I_x u_{kl}+I_y v_{kl}+I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl}-\bar{v}_{kl}) + 2\lambda(I_x u_{kl}+I_y v_{kl}+I_t)I_y$$

local average $\quad \bar{u}_{ij} = \frac{1}{4}\left\{u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}\right\}$

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

(set derivatives to zero and solve for unknowns u and v)

$$(1 + \lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

(set derivatives to zero and solve for unknowns u and v)

$$(1 + \lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

this is a linear system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ *how do you solve this?*

ok, take a step back, why are we doing all this math?

We are solving for the optical flow (u,v) given two constraints

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

smoothness

brightness constancy

We need the math to minimize this
(back to the math)

Partial derivatives of Horn-Schunck objective function E:

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

(set derivatives to zero and solve for unknowns u and v)

$$(1 + \lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$$ *how do you solve this?*

$$(1 + \lambda I_x^2) u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2) v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

Recall $\boldsymbol{x} = \mathbf{A}^{-1} \boldsymbol{b} = \frac{\mathrm{adj}\mathbf{A}}{\det \mathbf{A}} \boldsymbol{b}$

$$(1+\lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1+\lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

Recall $\boldsymbol{x} = \mathbf{A}^{-1}\boldsymbol{b} = \dfrac{\mathrm{adj}\mathbf{A}}{\det \mathbf{A}}\boldsymbol{b}$

Same as the linear system:

$$\{1+\lambda(I_x^2+I_y^2)\}u_{kl} = (1+\lambda I_x^2)\bar{u}_{kl} - \lambda I_x I_y \bar{v}_{kl} - \lambda I_x I_t$$

(determinant of A)

$$\{1+\lambda(I_x^2+I_y^2)\}v_{kl} = (1+\lambda I_y^2)\bar{v}_{kl} - \lambda I_x I_y \bar{u}_{kl} - \lambda I_y I_t$$

(determinant of A)

$$\{1+\lambda(I_x^2+I_y^2)\}u_{kl} = (1+\lambda I_x^2)\bar{u}_{kl} - \lambda I_x I_y \bar{v}_{kl} - \lambda I_x I_t$$

(determinant of A)

$$\{1+\lambda(I_x^2+I_y^2)\}v_{kl} = (1+\lambda I_y^2)\bar{v}_{kl} - \lambda I_x I_y \bar{u}_{kl} - \lambda I_y I_t$$

(determinant of A)

Rearrange to get update equations:

$$\underbrace{\hat{u}_{kl}}_{\text{new value}} = \underbrace{\bar{u}_{kl}}_{\text{old average}} - \frac{I_x\bar{u}_{kl} + I_y\bar{v}_{kl} + I_t}{1+\lambda(I_x^2+I_y^2)} I_x$$

$$\hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x\bar{u}_{kl} + I_y\bar{v}_{kl} + I_t}{1+\lambda(I_x^2+I_y^2)} I_y$$

check for missing lambda

ok, take a step back, why did we do all this math?

We are solving for the optical flow (u,v) given two constraints

$$\sum_{ij} \left\{ \frac{1}{4} \left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right] + \lambda \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2 \right\}$$

smoothness　　　　brightness constancy

We needed the math to minimize this
(now to the algorithm)

Horn-Schunck Optical Flow Algorithm

1. `Precompute image gradients` $I_y \quad I_x$

2. `Precompute temporal gradients` I_t

3. `Initialize flow field` $\boldsymbol{u} = \mathbf{0}$

   $\boldsymbol{v} = \mathbf{0}$

4. `While not converged`

   `Compute flow field updates for each pixel:`

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{1 + \lambda(I_x^2 + I_y^2)} I_x \qquad \hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{1 + \lambda(I_x^2 + I_y^2)} I_y$$