Computer Vision and Image Processing

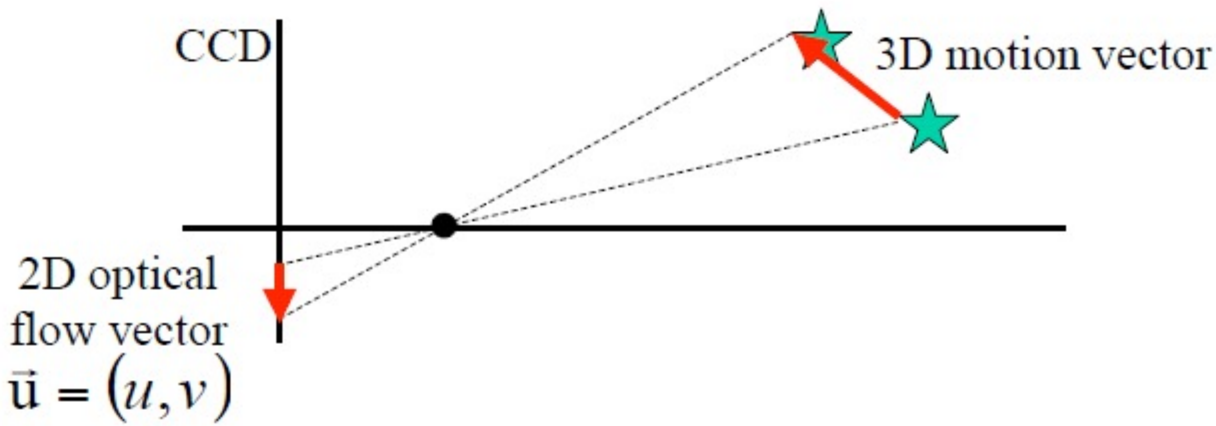
Gil Ben-Artzi

4

Optical Flow

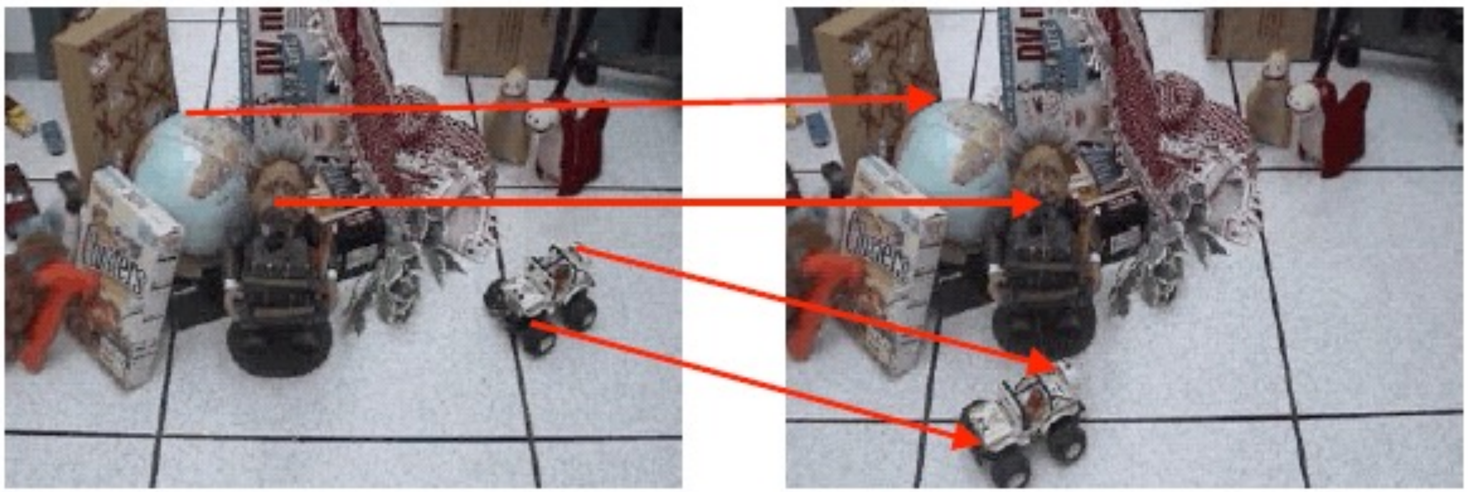
Motion Field and Optical Flow

- Motion field - 3D motion in the real world
- Optical flow - Pixel movement in the image, which is the result of motion field projection



Optical Flow

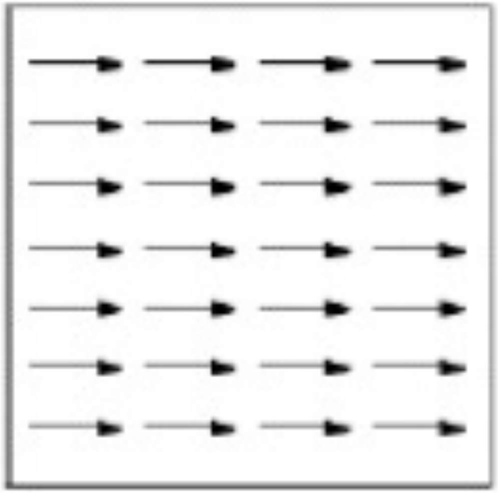
- The movement of **<u>pixels</u>**



Optical Flow

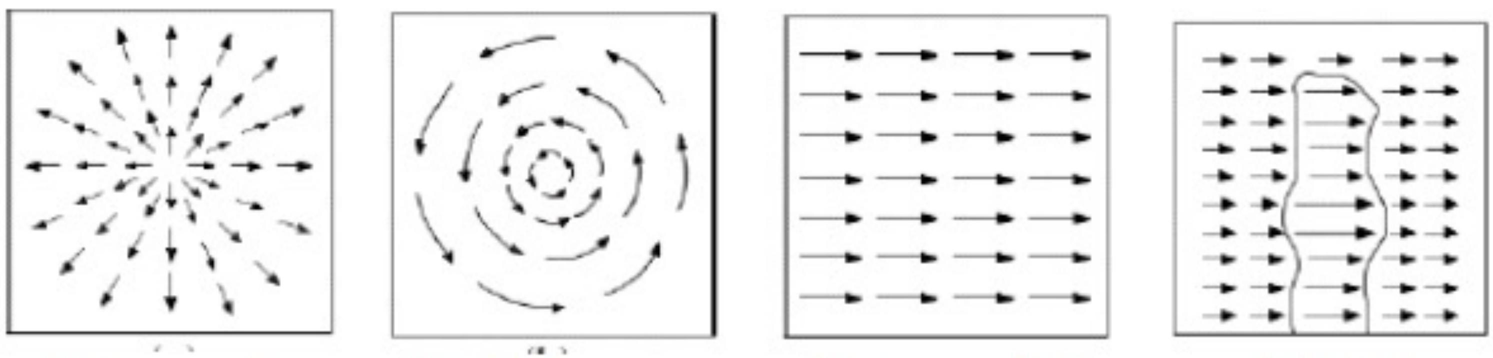
- For each pixel we have a vector describing:
  - The direction of its motion
  - The velocity of it’s motion across the image

Optical Flow



Horizontal
translation

Optical Flow



Forward motion

Rotation

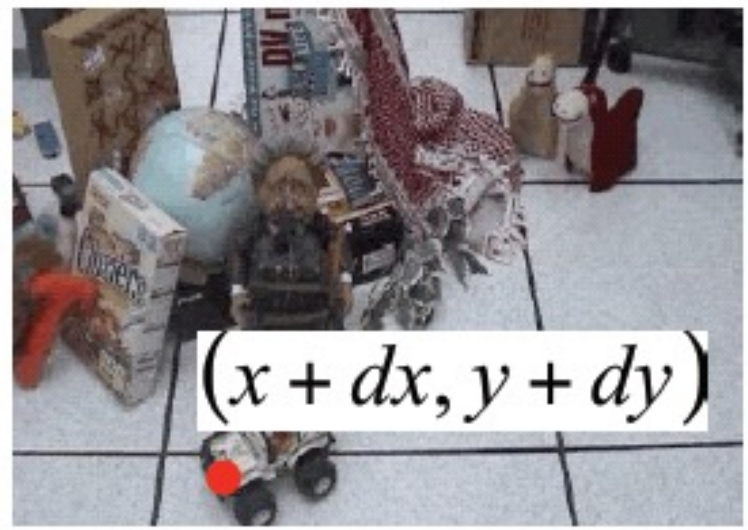
Horizontal translation

Closer objects appear to move faster!!

Optical Flow: Brightness Constancy

Time = t

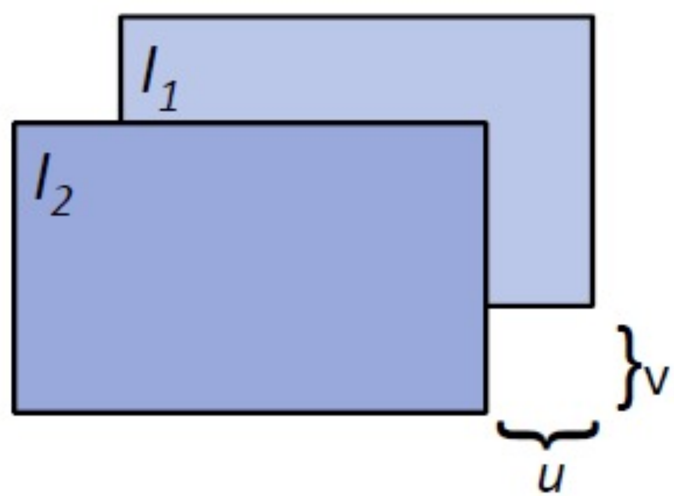
Time = $t+dt$



$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

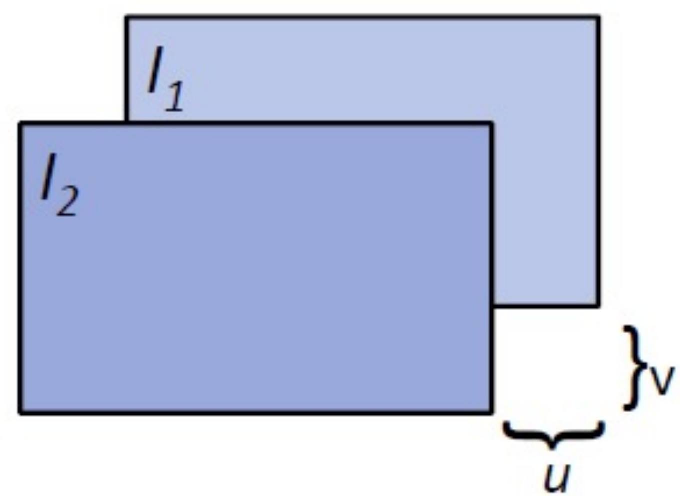
Optical flow calculation

- Assumptions
  - The intensity of a given object point does not change between frames. This is called brightness consistency.
- Let's assume <u>large </u>regions share the same intensity



Optical Flow - Brightness constancy

- Given images I_1 and I_2, we can find the translation *(u,v)* that will minimize the squared error $$E(u,v) = \sum_x \sum_y \left(I_1(x,y) - I_2(x+u, y+v)\right)^2$$
- Average over area of overlap
- Can also search for rotations: *(u,v,α)*



Cross Correlation

- Starting from the *SSD*

$$E(u,v)=\sum_x\sum_y\left(I_1(x,y)-I_2(x+u,y+v)\right)^2$$

- Since $(a-b)^2=a^2-2ab+b^2$
- We can write

$$E(u,v)=\sum_x\sum_y I_1{}^2-2\sum_x\sum_y I_1(x,y)\cdot I_2(x+u,y+v)+\sum_x\sum_y I_2{}^2$$

- Since $\Sigma I_1{}^2$ and $\Sigma I_2{}^2$ are almost constant, minimizing the *SSD* maximizes the cross-correlation $\Sigma I_1 I_2$

$$C(u,v)=\sum_x\sum_y I_1(x,y)\cdot I_2(x+u,y+v)$$

Normalized Cross Correlation

- Given two images I_1 and I_2 , search for the translation *(x, y)* maximizing the cross-correlation

$$C(u,v) = \sum_x \sum_y I_1(x,y) \cdot I_2(x+u, y+v)$$

- Normalized Cross Correlation eliminates additions and multiplications effects

$$NC(u,v) = \frac{\sum (I_1(x,y) - \hat{I}_1) \cdot (I_2(x+u, y+v) - \hat{I}_2)}{\sqrt{\sum (I_1(x,y) - \hat{I}_1)^2} \sqrt{\sum (I_2(x,y) - \hat{I}_2)^2}}$$

Limitation (Search Based)

- Discrete accuracy: checking every possible translation
- Complexity increases exponentially with numbers of parameters
  - Translation: (u,v) Complexity is N^2
  - Rotations: (u,v,α) Complexity is N^3
  - Zoom: (u,v,α,s) Complexity is N^4

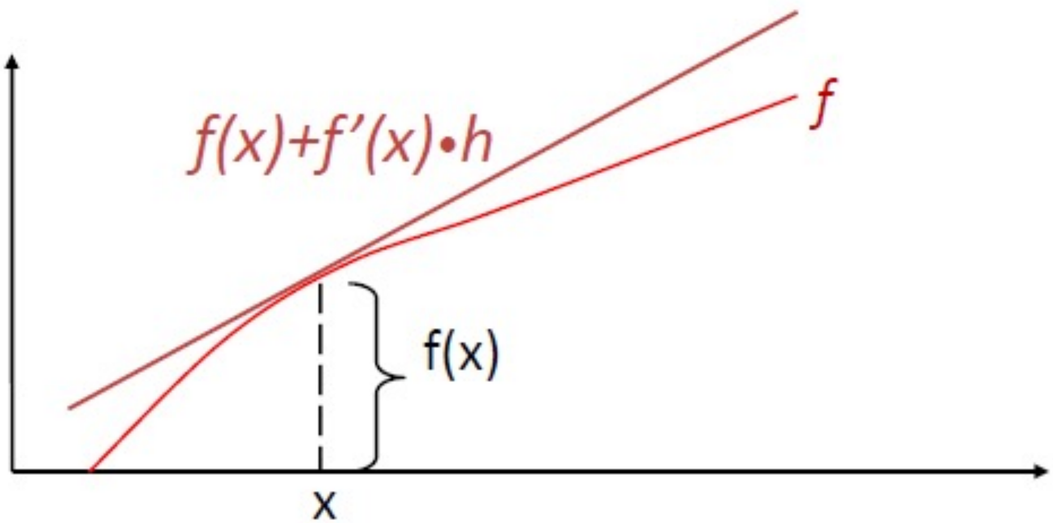
How Can We Improve?

- Add Assumption: Small motion
- Small means approximately less than 1 pixel
  - We can use Taylor approximation for the differences between the image

Lucas-Kanade: Taylor Approximation

- Local Taylor approximation in 1D:

$$f(x+h) \approx f(x) + f'(x) \cdot h$$



- Local Taylor approximation in 2D for images:

$$f(x+u, y+v) \approx f(x, y) + \frac{\partial f}{\partial x} \cdot u + \frac{\partial f}{\partial y} \cdot v$$

Optical Flow – LK

- MSE when shifting I_2 relative to I_1 by *(u,v):*

$$E(u,v)=\sum_x\sum_y\left[I_2(x+u,y+v)-I_1(x,y)\right]^2$$

- To simplify, we look at a single pixel (No ∑ ∑) *and use Taylor approximation*

$$E(u,v)=\left[I_2(x+u,y+v)-I_1(x,y)\right]^2\approx$$

$$\left[I_2(x,y)+\frac{\partial I_2}{\partial x}\cdot u+\frac{\partial I_2}{\partial y}\cdot v-I_1(x,y)\right]^2=$$

$$(I_x\cdot u+I_y\cdot v+I_t)^2$$

$$where\quad I_x=\frac{\partial I_2}{\partial x};\quad I_y=\frac{\partial I_2}{\partial y};\quad I_t=I_2-I_1;$$

Optical Flow - Lucas Kanade(LK)

- LK presented a simple way to solve the brightness constancy (BC) equation
- But it only works for small motion
  - The key idea – for small motion we can linearize the equation and solve them easily
- We are given two images I_1, I_2 and our goal is to find (u,v) the displacement of each pixel

$$\text{Our cost function: } E(u,v) = \sum_{x,y} (I_x u + I_y v + I_t)^2$$

$$\text{Our goal: } \min_{u,v} E(u,v)$$

Optical Flow - LK

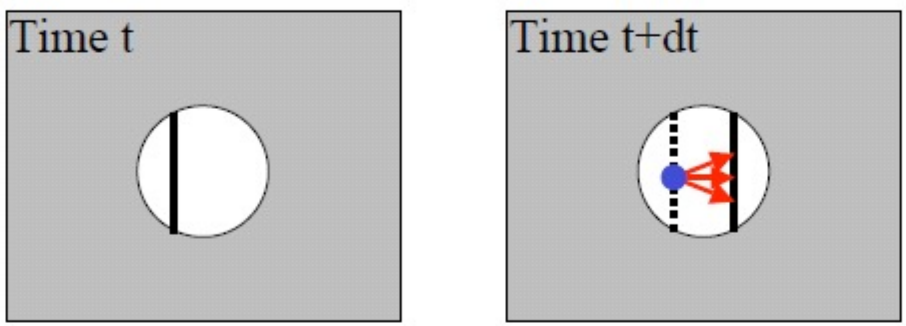
- The LK optical flow equation: $I_x u + I_y v = -I_t$
  - I_x : The *x* derivative of image I_2
  - I_y : The *y* derivative of image I_2
  - I_t : The image difference $I_2 - I_1$

- How to solve?

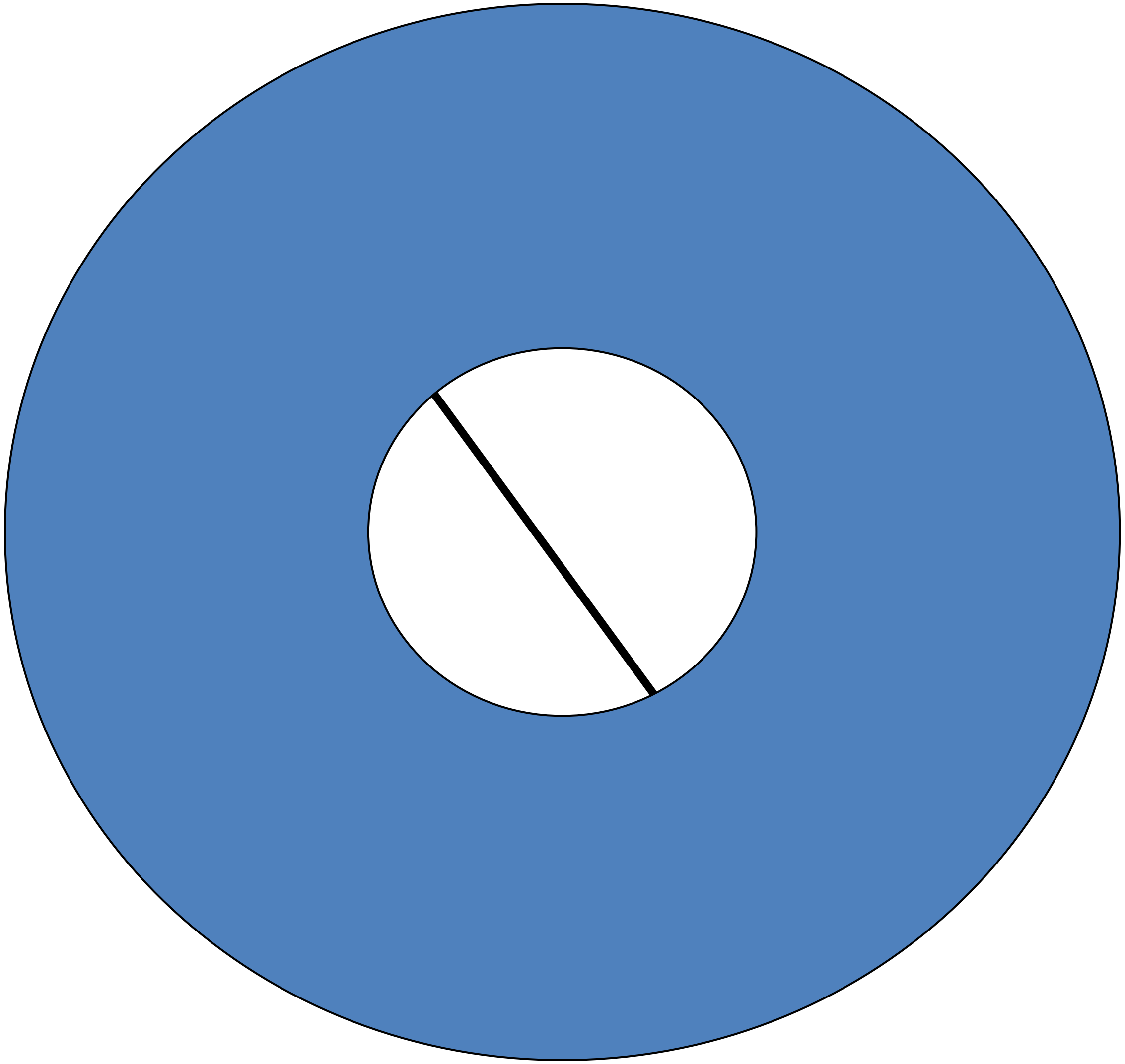
$$I_x u + I_y v = -I_t \Longrightarrow \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

The problems

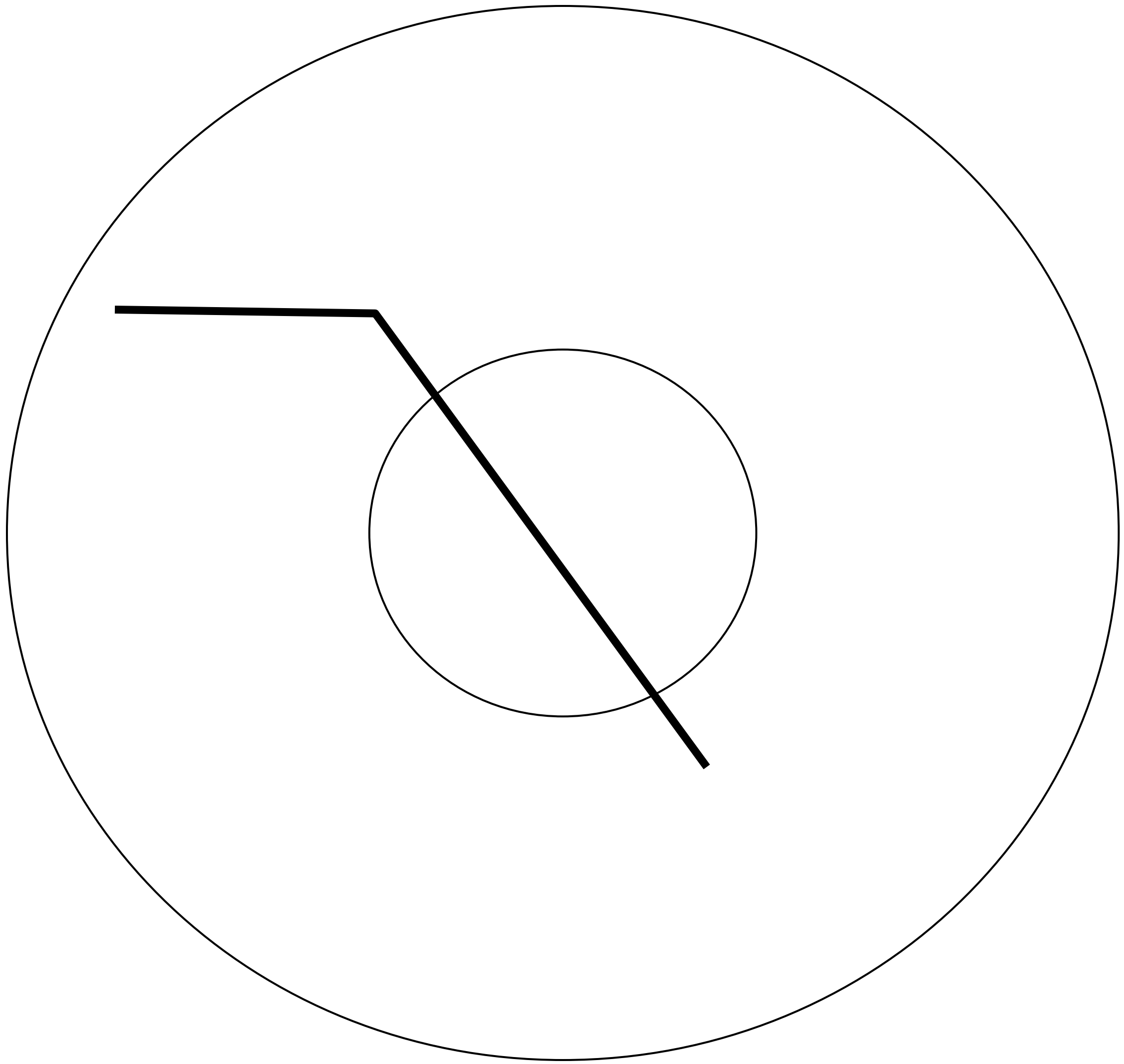
- We have one equations, but two unknowns
- What it means:



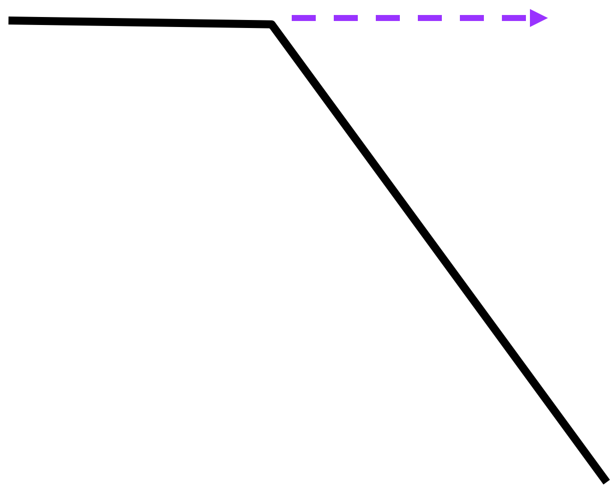
The Aperture Problem



The Aperture Problem

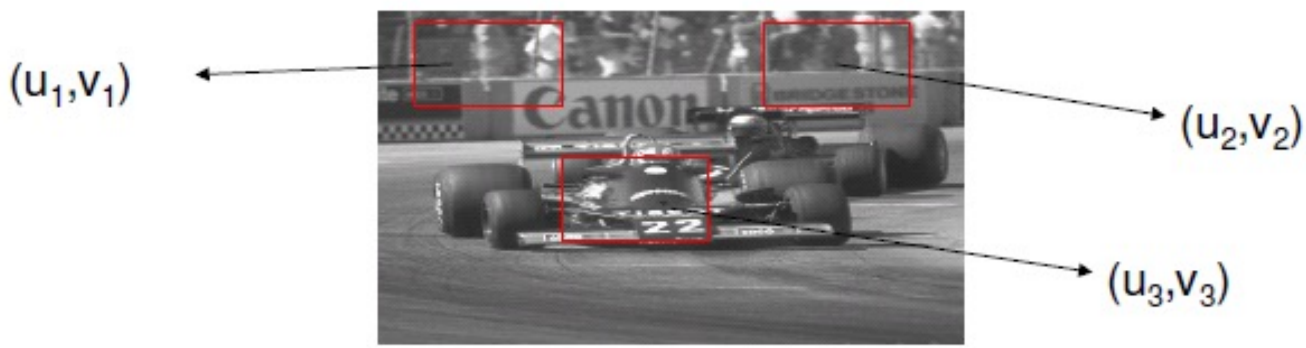


The Aperture Problem



Add additional Constraint

- Add local smoothness assumption

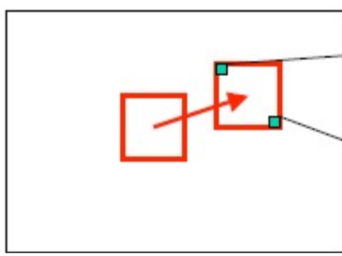


- Same optical flow for the entire block

Lucas Kanade (LK)

$$I_x u + I_y v = -I_t \quad \Longrightarrow \quad \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

Assume constant (u,v) in small neighborhood



$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \end{bmatrix}$$

LK - 5x5 Window

$$\underset{\substack{A \\ 25\times2}}{\begin{bmatrix} I_x(\mathrm{p}_1) & I_y(\mathrm{p}_1) \\ I_x(\mathrm{p}_2) & I_y(\mathrm{p}_2) \\ \vdots & \vdots \\ I_x(\mathrm{p}_{25}) & I_y(\mathrm{p}_{25}) \end{bmatrix}} \underset{\substack{d \\ 2\times1}}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{\substack{b \\ 25\times1}}{\begin{bmatrix} I_t(\mathrm{p}_1) \\ I_t(\mathrm{p}_2) \\ \vdots \\ I_t(\mathrm{p}_{25}) \end{bmatrix}}$$

How to solve: generic approach

$$E(u,v) = \sum_{x,y} (I_x \cdot u + I_y \cdot v + I_t)^2$$

- Finding *(u,v)* by setting derivatives to zero:

$$\begin{cases} \dfrac{\partial E}{\partial u} = \sum_{x,y} I_x \cdot (I_x \cdot u + I_y \cdot v + I_t) \quad = 0 \\ \dfrac{\partial E}{\partial v} = \sum_{x,y} I_y \cdot (I_x \cdot u + I_y \cdot v + I_t) \quad = 0 \end{cases}$$

$$\begin{cases} \begin{bmatrix} \sum_{x,y} I_x \cdot I_x & \sum_{x,y} I_x \cdot I_y \\ \sum_{x,y} I_y \cdot I_x & \sum_{x,y} I_y \cdot I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_{x,y} I_x \cdot I_t \\ \sum_{x,y} I_y \cdot I_t \end{bmatrix} \end{cases}$$

Lukas-Kanade optical flow

We have over constrained equation set:

$$\underset{25\times2}{A}\ \underset{2\times1}{d} = \underset{25\times1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

Solution: solve least squares problem

Python: np.dot(numpy.linalg.pinv(A),b)

Matlab: pinv(A)*b

Least Square

Given: $\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$

minimum least squares solution given by solution (in d) of:

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

- The summations are over all pixels in the K x K window

The solution matrix is $(A^\top A)^{-1} A^\top b$

LK Optical flow

- T solution involves the inverse of:

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

When is this solvable?

- **A^{T}A** should be invertible
- **A^{T}A** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **A^{T}A** should not be too small
- **A^{T}A** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Solutions

- Homogenous area = singular matrix, rank 0

Solutions

- Edge = singular, rank 1

Solutions

- Textured area = invertible matrix!

Putting it all together

- Select the target window to track
  - Can be every pixel or a specific window
- Use LK/Corr to look for in the next image
- The (u,v) are the requested offsets

Small Motion is a realistic assumption?

Is this motion small enough?

- Probably not—it's much larger than one pixel
- How might we solve this problem?

The problem: Small motion assumption

- We need a way to measure by small movement even if the movement is larger
  - Use iterative approach
  - Use multi scale estimation

Iterative LK approach

- Compute image derivatives I_x, I_y. Set *u,v* to 0.
- Compute once $$A = \begin{bmatrix} \sum I_x \cdot I_x & \sum I_x \cdot I_y \\ \sum I_y \cdot I_x & \sum I_y \cdot I_y \end{bmatrix}$$
- <u>Iterate</u> until convergence *(I_t ≈0):*
  – compute $$b = \begin{bmatrix} \sum I_x \cdot I_t \\ \sum I_y \cdot I_t \end{bmatrix}, I_t(x, y) = I_2(x, y) - I_1(x+u, y+v)$$
  – Solve equations to compute residual motion
  $$A \cdot \begin{bmatrix} du \\ dv \end{bmatrix} = -b$$
  – Update total motion with residual motion: *u+=du, v+=dv*
  – <u>**Warp**</u> I_2 towards I_1 with total motion *(u,v)*.

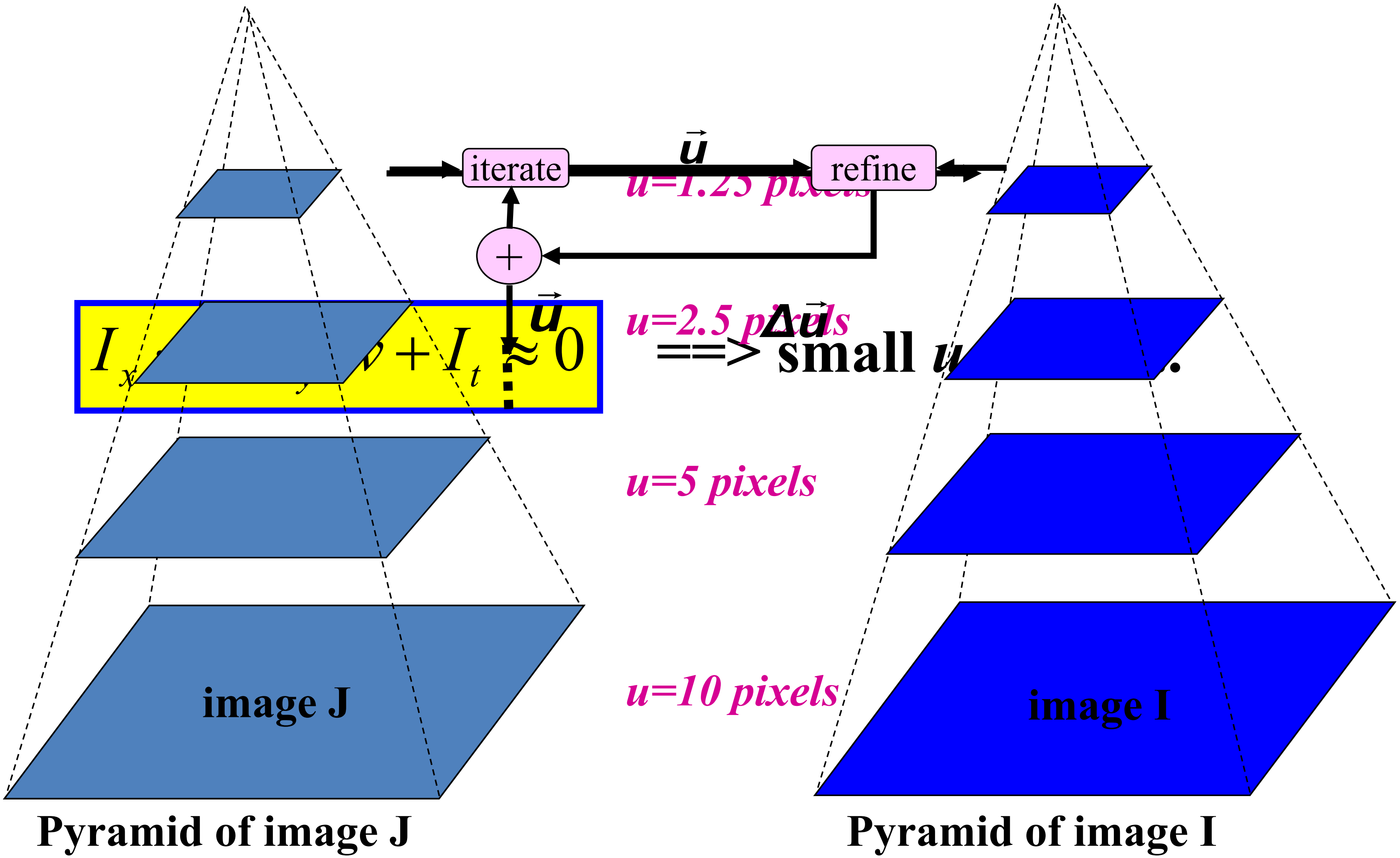
Why Iterative Approach?

- Compute the image derivatives only once
- Has two stages in each iteration:
  - Motion Estimation
  - Warping
- Works even with poor motion estimation, as long as it reduces the residual error
- Warping of one image towards the other is done from original image using total motion, and not from previous image using residual motion. (Repetitive warping blurs!)

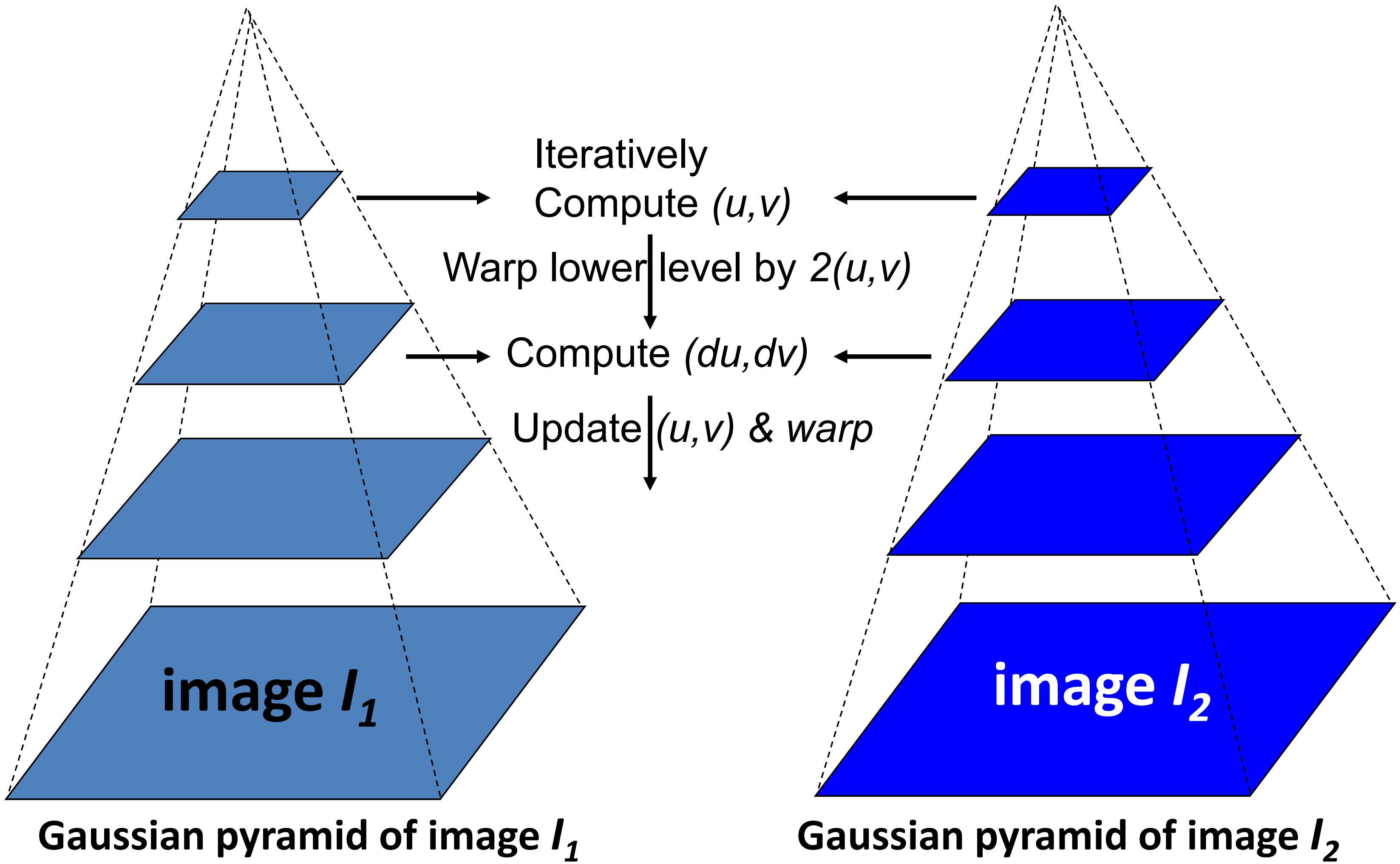
Limitations

- Using iterative LK assumes we are in proximity of the solution
- If the motion is too large it might not converge
- How can we improve it?

Multiscale (Coarse-to-fine) Estimation



Iterative & Multiscale approach



Horn-Schuck (HS) Optical Flow

- Lucas-Kanade method is based on local regions
- What if we would like to solve for the (u,v) displacement of all the pixels at once?
  - So now we have for each pixel: $u_{(x,y)}, v_{(x,y)}$
- HS presented such a cost function with a solution

Horn-Schunck (HS) Optical Flow

- They add an assumption
  - Neighboring pixels move together= same $u_{(x,y)}, v_{(x,y)}$ for nearby pixels
  - This is called **smoothness** assumption or **regularization**
- Note that we also assumed such an assumption in LK but we use it per block

HS Optical Flow

- They defined a new error (cost) function
- The regular one is now called the data term

$$\sum_{\Omega} [\, u \cdot I_x + v \cdot I_y + I_t \,]^2$$

- But they added the smoothness term/constraint

$$\sum_{\Omega} u_x^2 + u_y^2 + v_x^2 + v_y^2$$

- Data term vs. Smoothness term

HS Optical Flow

- The final error function to minimize is

$$\sum_{\Omega} [\ u \cdot I_x + v \cdot I_y + I_t\ ]^2 \quad + \lambda \sum_{\Omega}\ u_x^2 + u_y^2 + v_x^2 + v_y^2$$

where $\lambda > 0$ is a balance between how important is the data term relative to smoothness term

HS vs. LK

HS

LK

http://eric-yuan.me/coarse-to-fine-optical-flow/

HS vs. LK vs. Pyramids LK