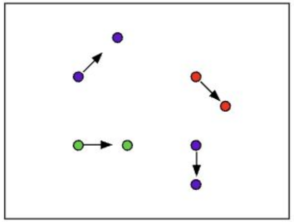
Lucas Kanade, Horn Schnuk, Seam Carving

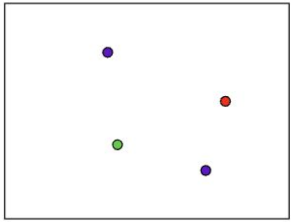
Simran Bagaria

Optical Flow Problem (Review)

- Given two subsequent frames, estimate the apparent motion field u(x,y), v(x,y) between them
- u(x, y) measuring the horizontal movement of the pixel at location (x, y), v(x, y) measures the vertical movement.
- Together, the pixel at (x, y, t-1) goes to (x+u, y+v, t)



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



$I(x,y,t)$

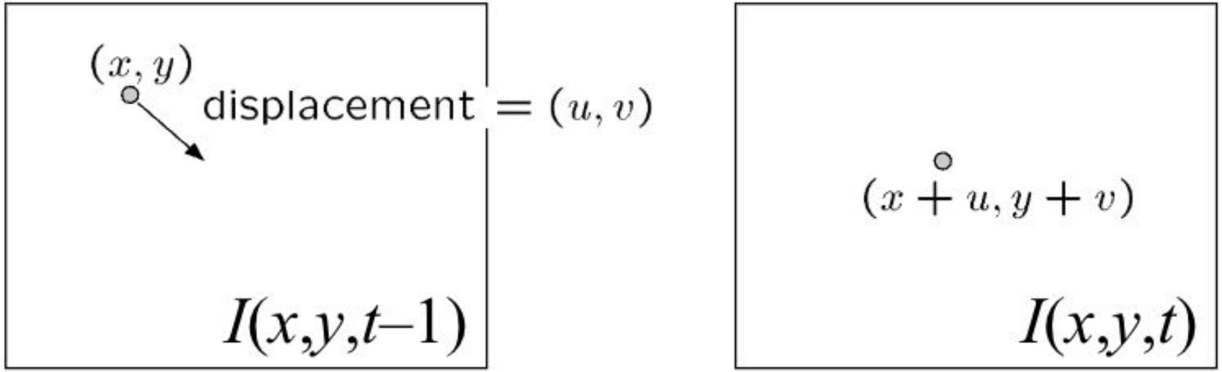
Lucas-Kanade

- Method for recovering image motion at pixels from optical flow
- 3 key assumptions:
  1. **small motions:** points do not move very far
  2. **spatial coherence:** points move like their neighbors
  3. **brightness constancy:** the brightness of a pixel remains constant between consecutive frames

Lucas-Kanade: Brightness Constancy Equation

Brightness Constancy: the brightness of a pixel remains constant between consecutive frames

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



First-Order Taylor Expansion

- The first-order taylor expansion of a function $f(x + \Delta x)$ around x is:

$$f(x + \Delta x) \approx f(x) + \nabla f \cdot \Delta x$$

- Now, we apply this to the RHS of the brightness constancy equation

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

Brightness Constancy Equation

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

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

(taylor expansion)

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

(subtract from both sides)

$$I_x \cdot u + I_y \cdot v + I_t \approx 0 \rightarrow \nabla I \cdot [u \;\; v]^T + I_t = 0$$

(brightness constancy assumption)

One equation, two unknowns!

Spatial Coherence Constraint

- Problem: 1 equation, 2 unknowns
- **spatial coherence:** points move like their neighbors
- Assume the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

Spatial Coherence Constraint

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix} \qquad \underset{25\text{x}2}{A} \; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Multiplying by A^T to solve for *d* gives us: $(A^T A)\ d = A^T b$

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

The summations are over all pixels in the 5 x 5 window

Conditions for solving this Lucas-Kanade equation

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

$$A^T A \qquad\qquad\qquad A^T b$$

When is This Solvable?

- **$A^T A$** should be invertible
- **$A^T A$** should not be too small, otherwise it is close to being non-invertible
  - 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)

Q. Does this remind anything to you?

M = A^TA is the Harris corner detector!

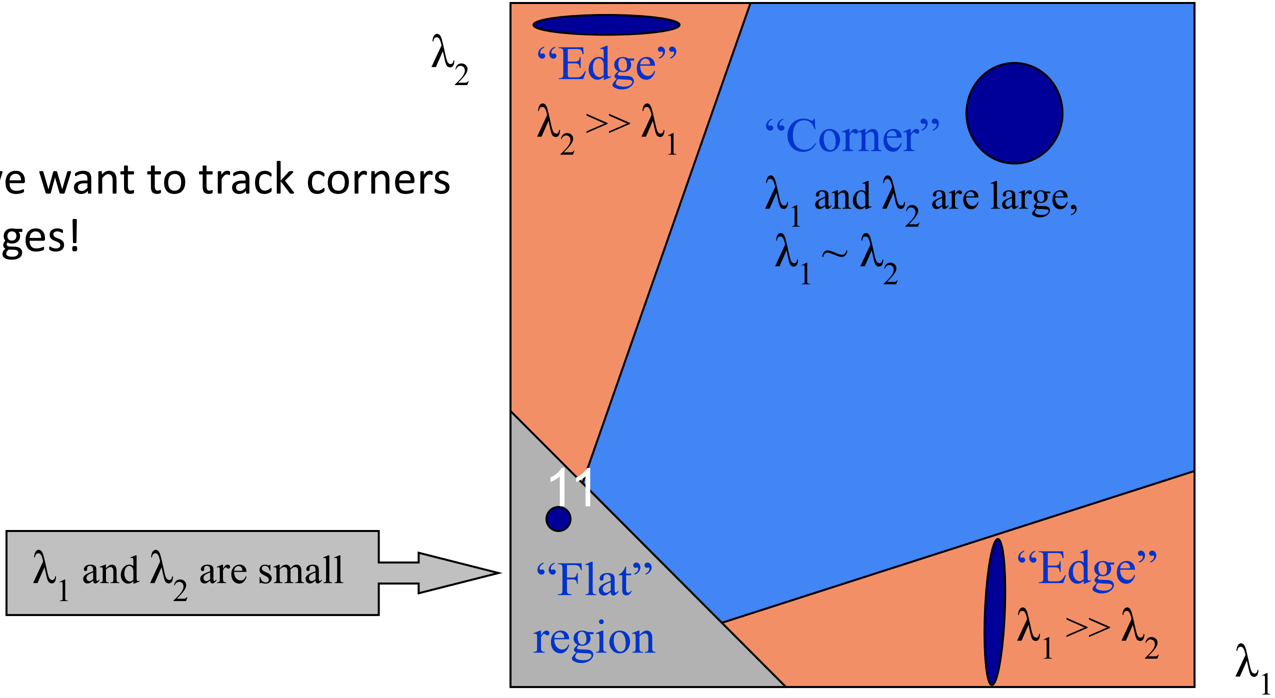
$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

- Eigenvectors and eigenvalues of A^TA relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change
  - The other eigenvector is orthogonal to it

Interpreting the eigenvalues

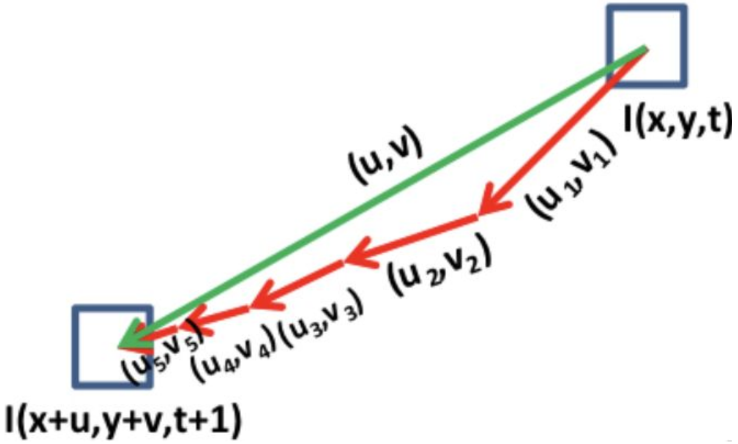
Classification of image points using eigenvalues of the second moment matrix:

This is why we want to track corners instead of edges!



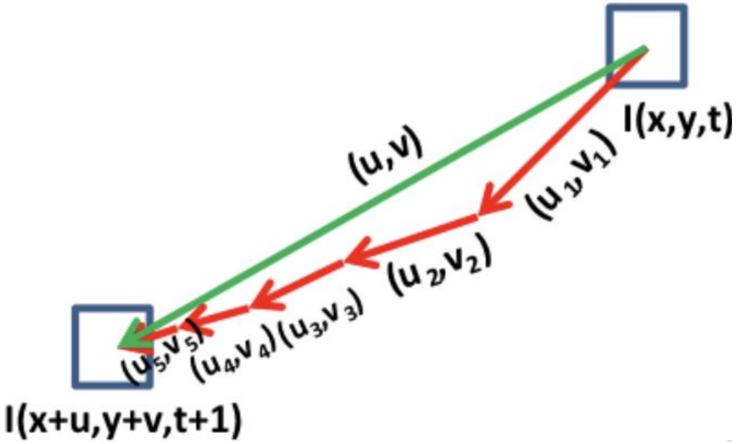
Iterative Lucas Kanade

- Problem: motion usually isn't very small, so regular Lucas-Kanade doesn't work
- Solution: we just repeatedly do this method!



Iterative Lucas-Kanade Algorithm

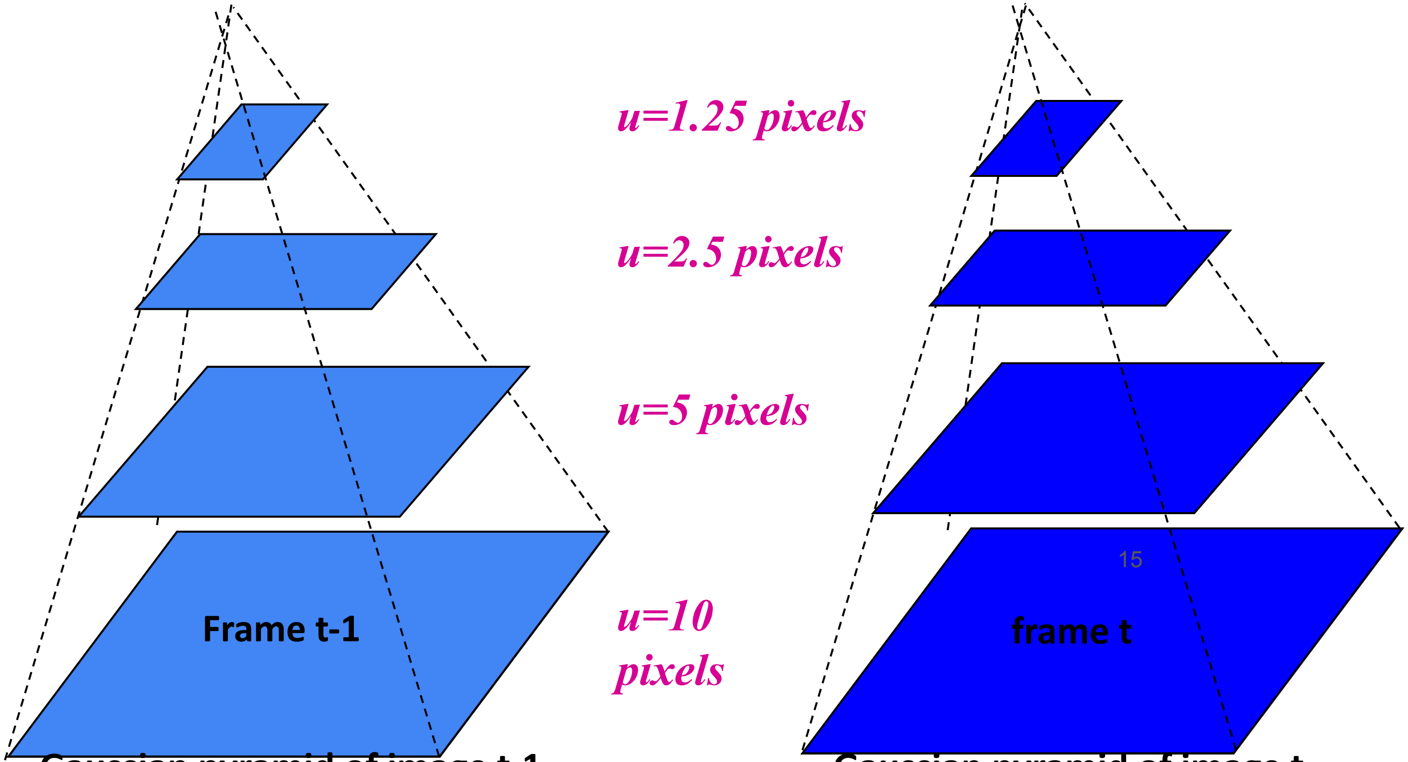
1. Estimate velocity at each pixel by solving Lucas-Kanade equations
2. Warp I(t-1) towards I(t) using the estimated flow field
3. Repeat until convergence



Pyramid Lucas-Kanade

- Problem: motion usually isn’t very small, so regular Lucas-Kanade doesn’t work
- Solution: reduce resolution of images until the motion is small

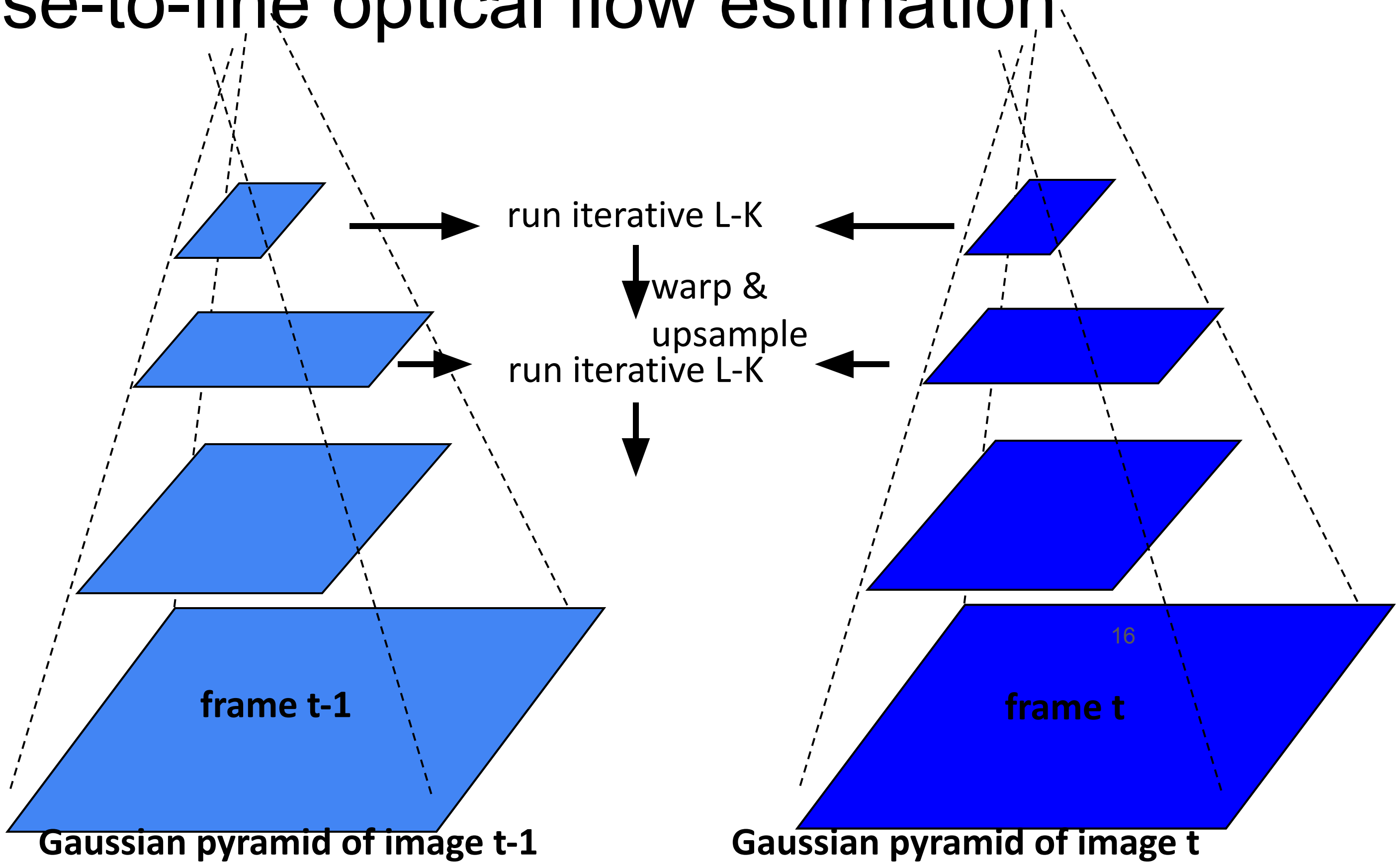
Coarse-to-fine optical flow estimation



Gaussian pyramid of image t-1

Gaussian pyramid of image t

Coarse-to-fine optical flow estimation



Warping and Upsampling

- At each pyramid level, we repeatedly calculate flow and warp the image based on that flow
- Once a pyramid level has converged, we upsample the flow to the next finer level
- To upsample:
  - Multiply flow vectors to match the finer image scale
- This upsampled flow is used as the starting point for iterative Lucas-Kanade at the finer level.

Attendance Form

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:
- The first part of the function is the brightness constancy.

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x\mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:
- The second part is the smoothness constraint. It’s trying to make sure that the changes between pixels are small.

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:
- ***α*** is a regularization constant. Larger values of ***α*** lead to smoother flows across time.

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- Recall that: $|\nabla u|^2 + |\nabla v|^2 = u_x^2 + u_y^2 + v_x^2 + v_y^2$
- Substituting this into our original energy equation, we get:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(|\nabla u|^2 + |\nabla v|^2 \right) \right] dx\, dy$$

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(u_x^2 + u_y^2 + v_x^2 + v_y^2 \right) \right] dx\, dy$$

Horn-Schunk method for optical flow

- Now, to minimize this, first we need to find the gradient with respect to u and v
- For this, we use the Euler-Lagrange Equation

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0$$

$$\frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial v_y} = 0$$

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right] dx\, dy$$

Euler Lagrange Equation

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0$$

- Focusing just on u and computing each term individually:

$$\frac{\partial \mathcal{L}}{\partial u} = 2I_x(I_x u + I_y v + I_t)$$

$$\frac{\partial \mathcal{L}}{\partial u_x} = 2\alpha^2 u_x$$

$$\frac{\partial \mathcal{L}}{\partial u_y} = 2\alpha^2 u_y$$

$\Longrightarrow$

$$\frac{d}{dx}\left(\frac{\partial \mathcal{L}}{\partial u_x}\right) = \frac{d}{dx}(2\alpha^2 u_x) = 2\alpha^2 u_{xx}$$

$$\frac{d}{dy}\left(\frac{\partial \mathcal{L}}{\partial u_y}\right) = \frac{d}{dy}(2\alpha^2 u_y) = 2\alpha^2 u_{yy}$$

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right] dx\, dy$$

Euler Lagrange Equation

- Plugging everything in for u:

$$2I_x(I_xu + I_yv + I_t) - 2\alpha^2 u_{xx} - 2\alpha^2 u_{yy} = 0$$
$$I_x(I_xu + I_yv + I_t) - \alpha^2(u_{xx} + u_{yy}) = 0$$

- Similarly, for v we have:

$$2I_y(I_xu + I_yv + I_t) - 2\alpha^2 v_{xx} - 2\alpha^2 v_{yy} = 0$$
$$I_y(I_xu + I_yv + I_t) - \alpha^2(v_{xx} + v_{yy}) = 0$$

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0$$

$$\frac{\partial \mathcal{L}}{\partial u} = 2I_x(I_xu + I_yv + I_t)$$

$$\frac{d}{dx}\left(\frac{\partial \mathcal{L}}{\partial u_x}\right) = 2\alpha^2 u_{xx}$$

$$\frac{d}{dy}\left(\frac{\partial \mathcal{L}}{\partial u_y}\right) = 2\alpha^2 u_{yy}$$

$$E = \iint \left[(I_xu + I_yv + I_t)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right] dx\, dy$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

- This minimization can be solved by taking the derivative with respect to u and v, we get the following 2 equations:

$$I_x(I_x u + I_y v + I_t) - \alpha^2 (u_{xx} + u_{yy}) = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2 (v_{xx} + v_{yy}) = 0$$

Horn-Schunk method for optical flow

- By taking the derivative with respect to u and v, we get the following 2 equations:

$$I_x(I_x u + I_y v + I_t) - \alpha^2(u_{xx} + u_{yy}) = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2(v_{xx} + v_{yy}) = 0$$

- Focusing on $u_{xx} + u_{yy}$: this essentially represents the 2nd derivative, so we estimate it with $\overline{u}(x,y) - u(x,y)$.
- $\overline{u}(x,y)$ is the weighted average of u measured at (x,y) over its neighborhood of 5 x 5 pixels
- This makes sense because the estimation measures the deviation from the average change.

Horn-Schunk method for optical flow

- Substituting into the original equations:

$$I_x(I_x u + I_y v + I_t) - \alpha^2(u_{xx} + u_{yy}) = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2(v_{xx} + v_{yy}) = 0$$

$$\Downarrow$$

$$I_x(I_x u + I_y v + I_t) - \alpha^2(\bar{u}(x,y) - u(x,y)) = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2(\bar{v}(x,y) - v(x,y)) = 0$$

Horn-Schunk method for optical flow

- Rearranging, we get:

$$I_x(I_x u + I_y v + I_t) - \alpha^2(\bar{u}(x,y) - u(x,y)) = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2(\bar{v}(x,y) - v(x,y)) = 0$$

$$\Downarrow$$

$$(I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \overline{u} - I_x I_t$$
$$I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \overline{v} - I_y I_t$$

- This is linear in u and v, which means there's an analytical solution for each pixel!

Horn-Schunk method for optical flow

- Analytical solution for:

$$(I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \overline{u} - I_x I_t$$
$$I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \overline{v} - I_y I_t$$

- is:

$$u = \bar{u} - \frac{I_x(I_x\bar{u} + I_y\bar{v} + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

$$v = \bar{v} - \frac{I_y(I_x\bar{u} + I_y\bar{v} + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

Seam Carving

Seam Carving

- Assume input I is size m x n
- Output I is m x n',
  - where n'<n

- Basic Idea: remove unimportant pixels from the image
  - Unimportant = pixels with less "energy"

$$E(I) = |\frac{\partial I}{\partial x}| + |\frac{\partial I}{\partial y}|$$

$$E(I) = \sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}$$

- Intuition for gradient-based energy:
  - Preserve edges
  - Human vision more sensitive to edges – so try remove content from smoother areas
  - Simple enough for producing some nice results

Dynamic Programming

Input: Given an energy E(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

- Create a cost matrix M with the following property:
  - M(i, j) = minimal cost of a seam going through pixel (i, j)
  - starting from j=0

M(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

M(i, 0) = E(i, 0) of a seam going through pixel (i, j)

M(i, j)

5	8	12	3

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

Q. What do you think should be this value?

5	8	12	3
	?		

M(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

M(i, j) = total energy of seam going through pixel (i, j) from j=0

5	8	12	3
	2+5		

M(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

The recurrence formula

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

5	8	12	3
	2+5		

M(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

M(i, j)

5	8	12	3
	7		

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

5	8	12	3
	7	?	

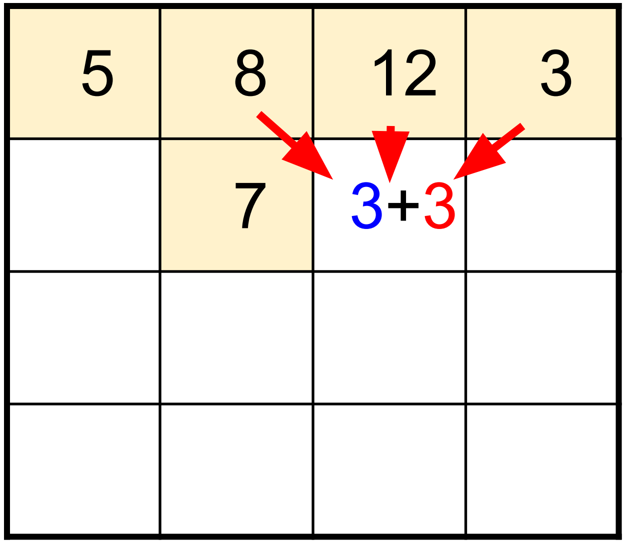
M(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$



M(i, j)

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Dynamic Programming

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i, j)

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	8+8

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(i, j)

Searching for minimum seam

Backtrack: Find the minimum M(i, j=m)

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

M(i, j)

This is the minimum in the last row

Backtrack

After finding minimum M(i, j) at row j,
find minimum M(i, j-1) but only be looking at neighboring locations: i-1, i, i+1

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

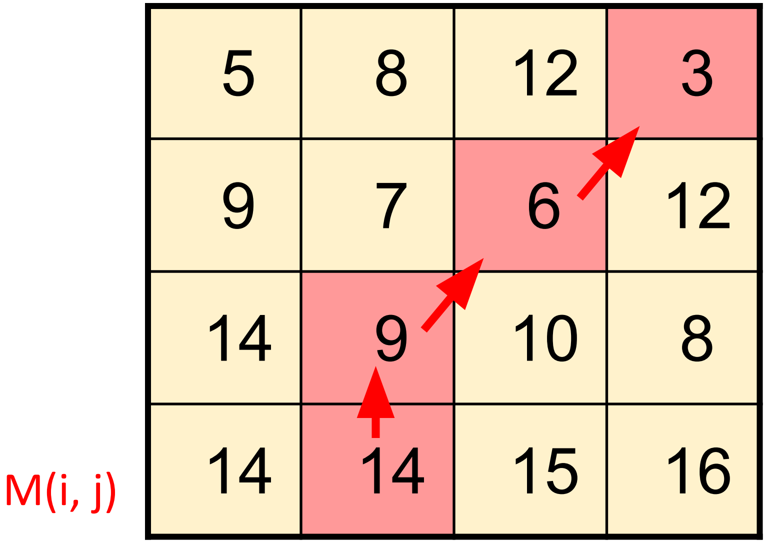
M(i, j)

Searching for Minimum

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

M(i, j)

Searching for Minimum



Preserved Energy

Average Pixel Energy

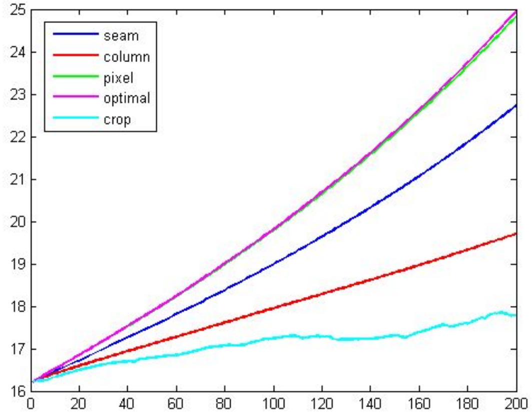
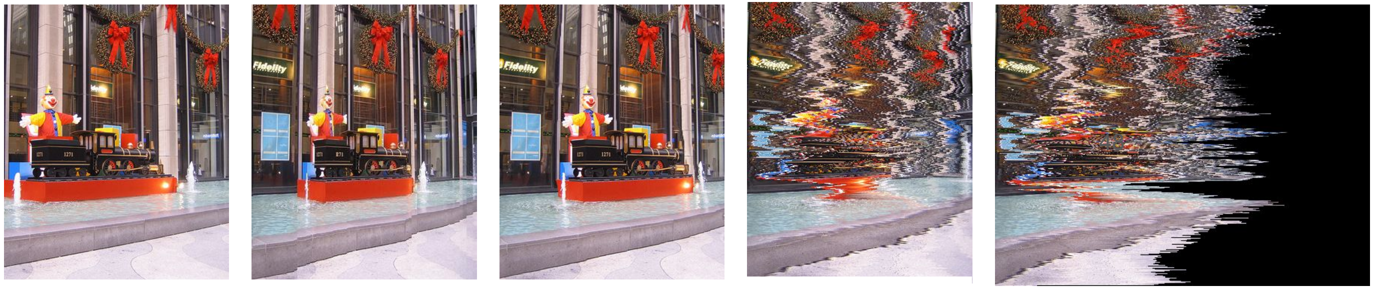


Image Reduction

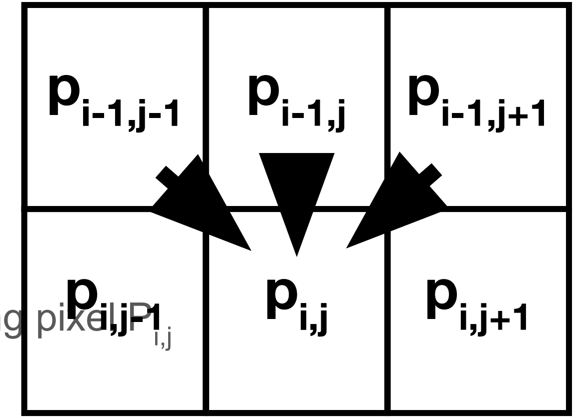


crop column seam pixel optimal

Minimize Inserted Energy

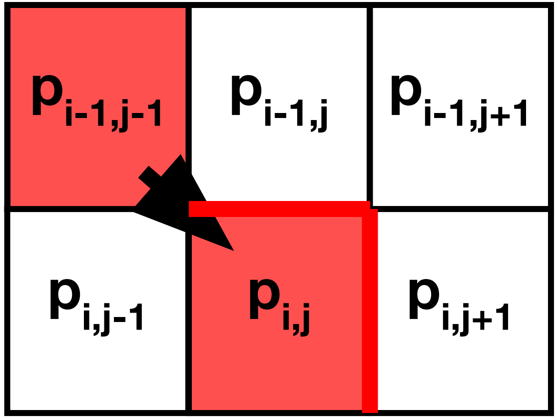
- Instead of removing the seam of least energy, remove the seam that *inserts the least energy* to the image !

Tracking Inserted Energy



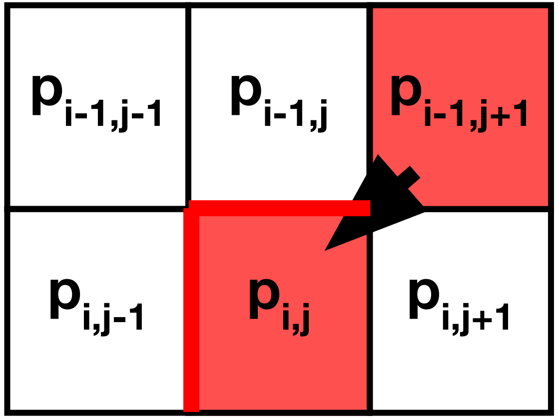
- Three possibilities when removing pixel $P_{i,j}$

Pixel $P_{i,j}$: Left Seam



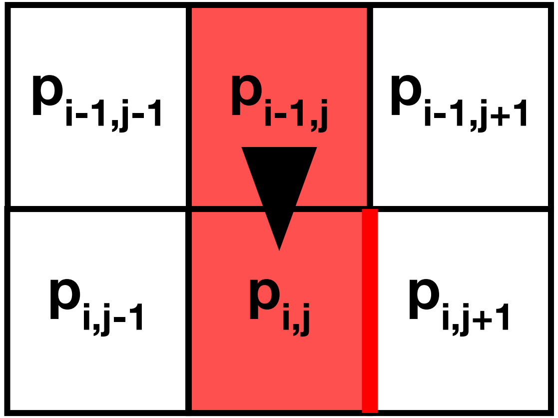
$$C_L(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)|$$

Pixel $P_{i,j}$: Right Seam



$$C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)|$$

Pixel $P_{i,j}$: Vertical Seam



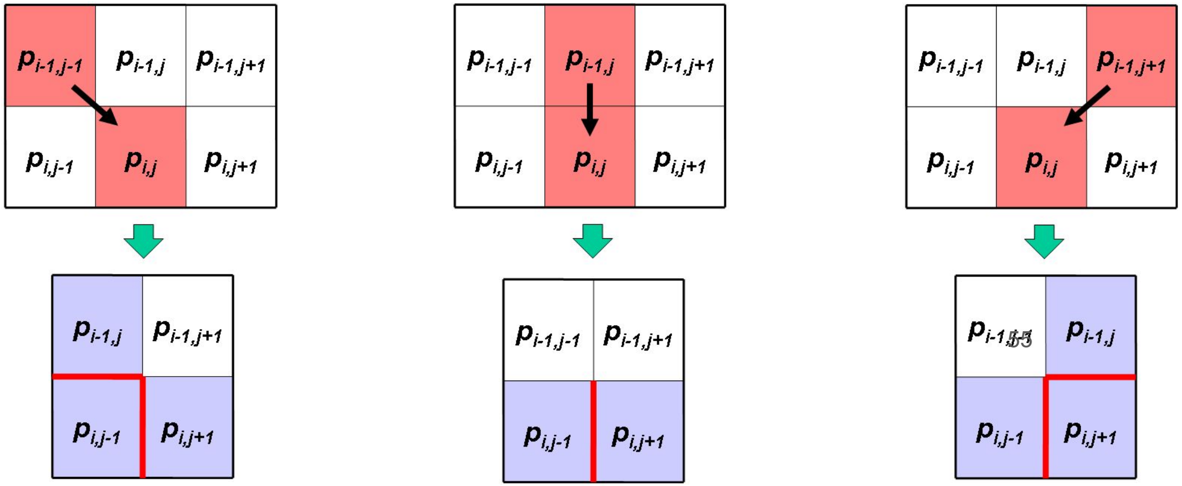
$$C_V(i,j) = |I(i,j+1) - I(i,j-1)|$$

Old Backward Cost Matrix

$$M(i,j) = E \quad j) + \min \begin{cases} M(i-1, j-1) &) \\ M(i-1, j) &), \\ M(i-1, j+1) &) \end{cases}$$

New Forward Looking Cost Matrix

$$M(i,j) = E(i,j) + min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_V(i,j) \\ M(i-1,j+1) + C_R(i,j) \end{cases}$$



Backward vs. Forward

Backward

Forwar
d