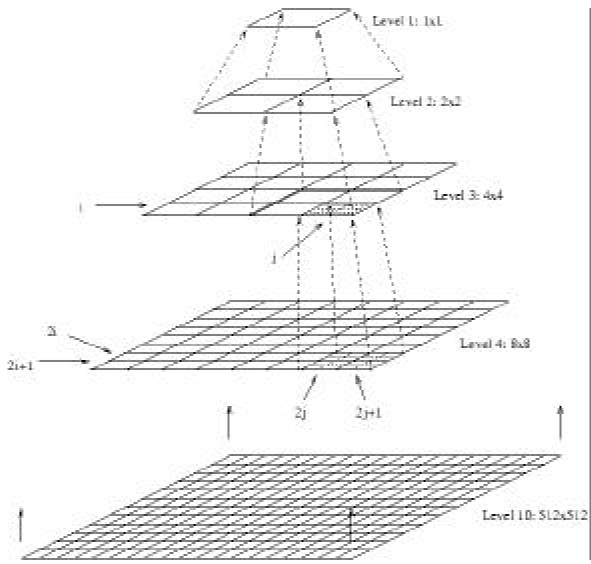
Comments

- Algorithm-1 works only for small motion.
- If object moves faster, the brightness changes rapidly, 2x2 or 3x3 masks fail to estimate spatiotemporal derivatives.
- Pyramids can be used to compute large optical flow vectors.

Pyramids

- Very useful for representing images.
- Pyramid is built by using multiple copies of image.
- Each level in the pyramid is 1/4 of the size of previous level.
- The lowest level is of the highest resolution.
- The highest level is of the lowest resolution.

Pyramid

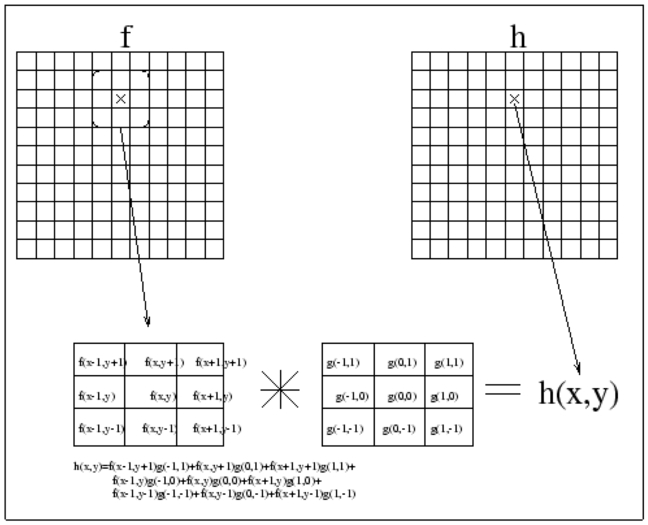


Gaussian Pyramids

$$g_l(i,j) = \sum_{m=-2}^{2}\sum_{n=-2}^{2} w(m,n)\, g_{l-1}(2i+m, 2j+n)$$

$$g_l = REDUCE[g_{l-1}]$$

Convolution



Gaussian Pyramids

$$g_{l,n}(i,j) = \sum_{p=-2}^{2}\sum_{q=-2}^{2} w(p,q)\, g_{l,n-1}\left(\frac{i-p}{2}, \frac{j-q}{2}\right)$$

$$g_{l,n} = EXPAND[g_{l,n-1}]$$

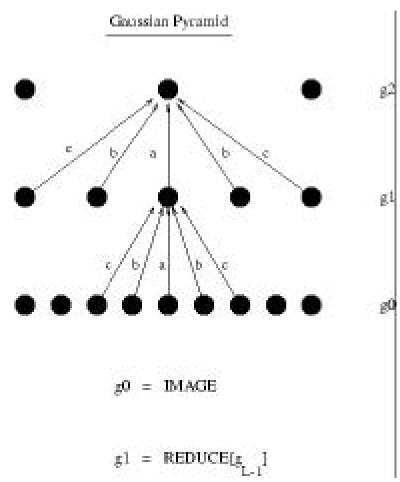
Reduce (1D)

$$g_l(i) = \sum_{m=-2}^{2} \hat{w}(m) g_{l-1}(2i+m)$$

$$g_l(2) = \hat{w}(-2) g_{l-1}(4-2) + \hat{w}(-1) g_{l-1} \hat{w}(4-1) +$$
$$\hat{w}(0) g_{l-1}(4) + \hat{w}(1) g_{l-1}(4+1) + \hat{w}(2) g_{l-1}(4+2)$$

$$g_l(2) = \hat{w}(-2) g_{l-1}(2) + \hat{w}(-1) g_{l-1} \hat{w}(3) +$$
$$\hat{w}(0) g_{l-1}(4) + \hat{w}(1) g_{l-1}(5) + \hat{w}(2) g_{l-1}(6)$$

Reduce



Expand (1D)

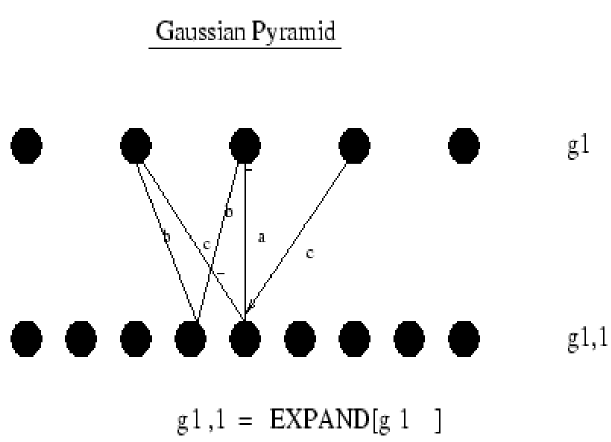
$$g_{l,n}(i) = \sum_{p=-2}^{2} \hat{w}(p) g_{l,n-1}(\frac{i-p}{2})$$

$$g_{l,n}(4) = \hat{w}(-2) g_{l,n-1}(\frac{4-2}{2}) + \hat{w}(-1) g_{l,n-1}(\frac{4-1}{2}) +$$

$$\hat{w}(0) g_{l,n-1}(\frac{4}{2}) + \hat{w}(1) g_{l,n-1}(\frac{4+1}{1}) + \hat{w}(2) g_{l,n-1}(\frac{4+2}{2})$$

$$g_{l,n}(4) = \hat{w}(-2) g_{l,n-1}(1) + \hat{w}(0) g_{l,n-1}(2) + \hat{w}(2) g_{l,n-1}(3)$$

Expand



Convolution Mask

$$[w(-2), w(-1), w(0), w(1), w(2)]$$

Convolution Mask

- Separable

$$w(m,n) = \hat{w}(m)\hat{w}(n)$$

- Symmetric

$$\hat{w}(i) = \hat{w}(-i)$$

$$[c, b, a, b, c]$$

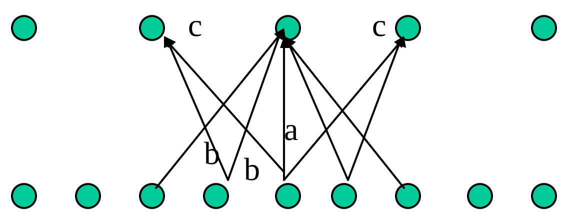
Convolution Mask

- The sum of mask should be 1.

$$a + 2b + 2c = 1$$

•All nodes at a given level must contribute the same total weight to the nodes at the next higher level.

$$a + 2c = 2b$$



Convolution Mask

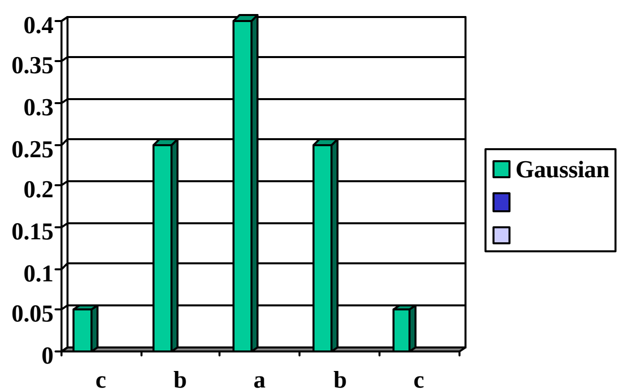
$$\hat{w}(0) = a$$

$$\hat{w}(-1) = \hat{w}(1) = \frac{1}{4}$$

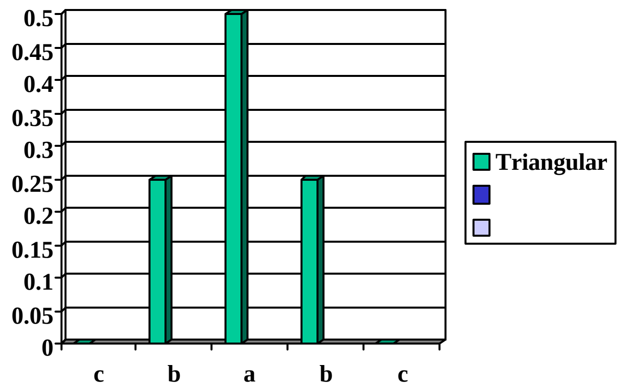
$$\hat{w}(-2) = \hat{w}(2) = \frac{1}{4} - \frac{a}{2}$$

a=.4 GAUSSIAN, a=.5 TRINGULAR

Approximate Gaussian



Triangular

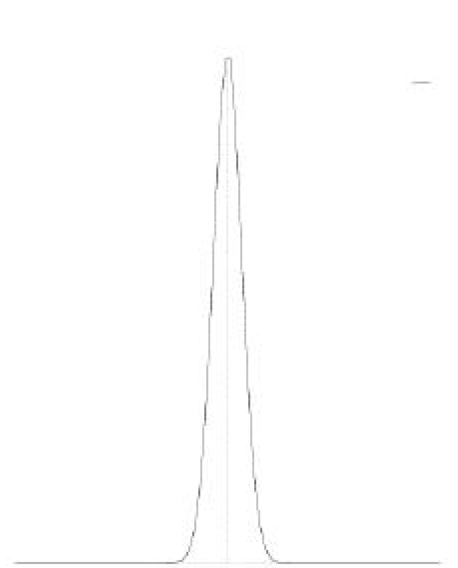


Gaussian

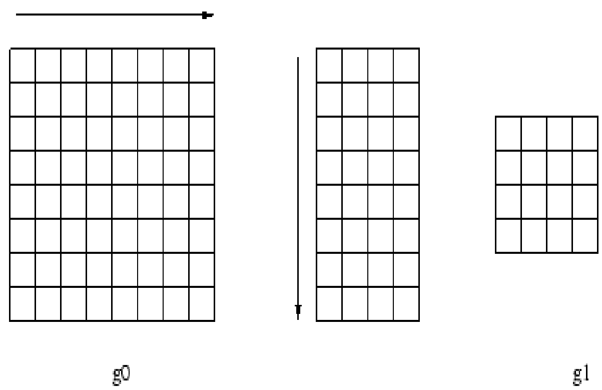
x	-3	-2	-1	0	1	2	3
g(x)	.011	.13	.6	1	.6	.13	.011

Gaussian

$$g(x) = e^{\frac{-x^2}{2\sigma^2}}$$



Separability



Algorithm

- Apply 1-D mask to alternate pixels along each row of image.
- Apply 1-D mask to each pixel along alternate columns of resultant image from previous step.

Gaussian Pyramid

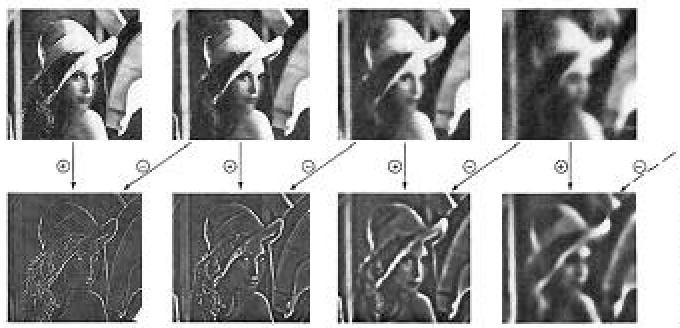
Laplacian Pyramids

- Similar to edge detected images.
- Most pixels are zero.
- Can be used for image compression.

$$L_1 = g_1 - EXPAND[g_2]$$

$$L_2 = g_2 - EXPAND[g_3]$$

$$L_3 = g_3 - EXPAND[g_4]$$



Coding using Laplacian Pyramid

- Compute Gaussian pyramid

$$g_1, g_2, g_3, g_4$$

- Compute Laplacian pyramid

$$L_1 = g_1 - EXPAND[g_2]$$
$$L_2 = g_2 - EXPAND[g_3]$$
$$L_3 = g_3 - EXPAND[g_4]$$
$$L_4 = g_4$$

- Code Laplacian pyramid

Decoding using Laplacian pyramid

• Decode Laplacian pyramid.
• Compute Gaussian pyramid from Laplacian pyramid.

$$g_3 = EXPAND[g_4] + L_3$$

$$g_2 = EXPAND[g_3] + L_2$$

$$g_1 = EXPAND[g_2] + L_1$$

- is reconstructed image.

Gaussian

- Most natural phenomenon can be modeled by Gaussian.
- Take a bunch of random variables of any distribution, find the mean, the mean will approach to Gaussian distribution.
- Gaussian is very smooth function, it has infinite no of derivatives.

Gaussian

- Fourier Transform of Gaussian is Gaussian.
- If you convolve Gaussian with itself, it is again Gaussian.
- There are cells in human brain which perform Gaussian filtering.
  - Laplacian of Gaussian edge detector

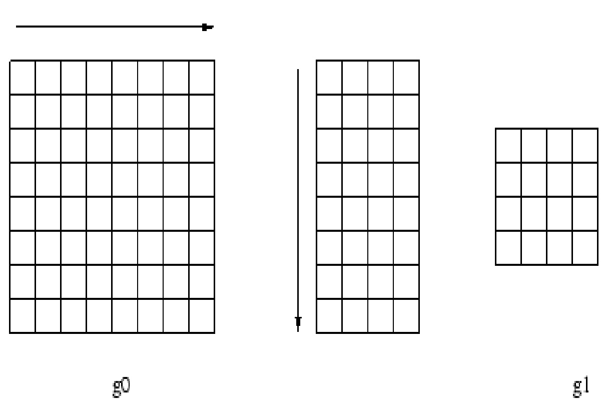
Carl F. Gauss

- Born to a peasant family in a small town in Germany.
- Learned counting before he could talk.
- Contributed to Physics, Mathematics, Astronomy,…
- Discovered most methods in modern mathematics, when he was a teenager.

Carl F. Gauss

- Some contributions
  - Gaussian elimination for solving linear systems
  - Gauss-Seidel method for solving sparse systems
  - Gaussian curvature
  - Gaussian qudrature

Separability



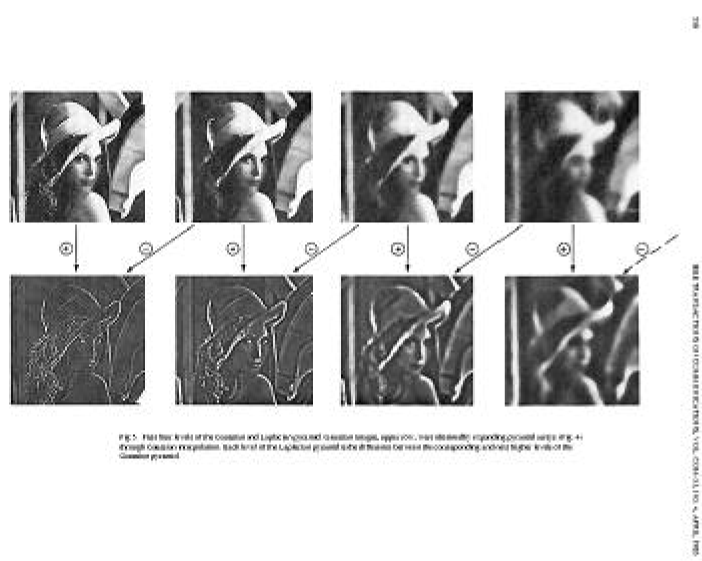
Algorithm

- Apply 1-D mask to alternate pixels along each row of image.
- Apply 1-D mask to each pixel along alternate columns of resultant image from previous step.

Gaussian Pyramid

Laplacian Pyramids

- Similar to edge detected images.
- Most pixels are zero.
- Can be used for image compression.



Coding using Laplacian Pyramid

- Compute Gaussian pyramid

$$g_1, g_2, g_3, g_4$$

- Compute Laplacian pyramid

$$L_1 = g_1 - EXPAND[g_2]$$
$$L_2 = g_2 - EXPAND[g_3]$$
$$L_3 = g_3 - EXPAND[g_4]$$
$$L_4 = g_4$$

- Code Laplacian pyramid

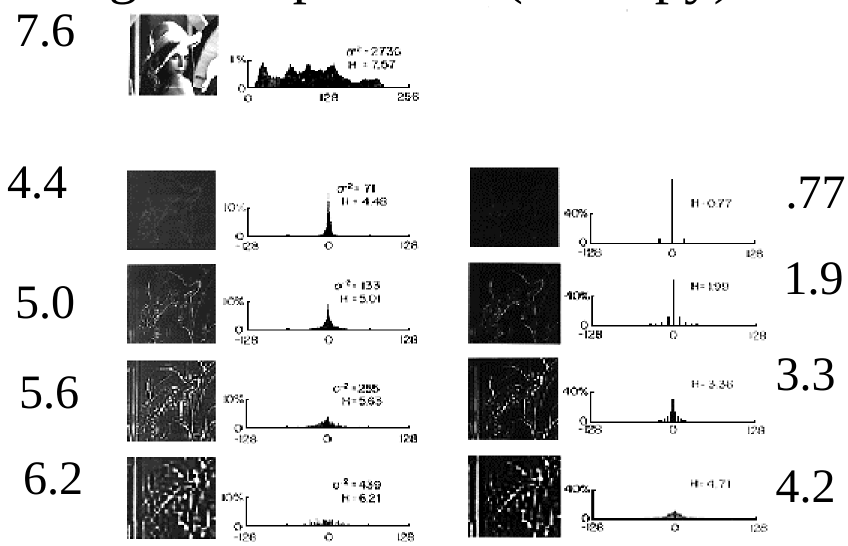
Decoding using Laplacian pyramid

- Decode Laplacian pyramid.
- Compute Gaussian pyramid from Laplacian pyramid.

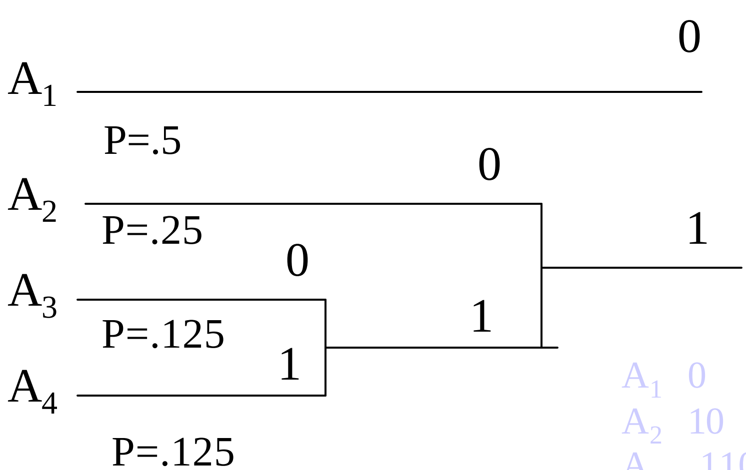
$$g_3 = EXPAND[g_4] + L_3$$
$$g_2 = EXPAND[g_3] + L_2$$
$$g_1 = EXPAND[g_2] + L_1$$

- is reconstructed image.

Image Compression (Entropy)



Huffman Coding (Example-1)



A_1 0
A_2 10
A_3 110
A_4 111

Huffman Coding

Entropy $H = -\sum_{i=0}^{255} p(i)\log_2 p(i)$

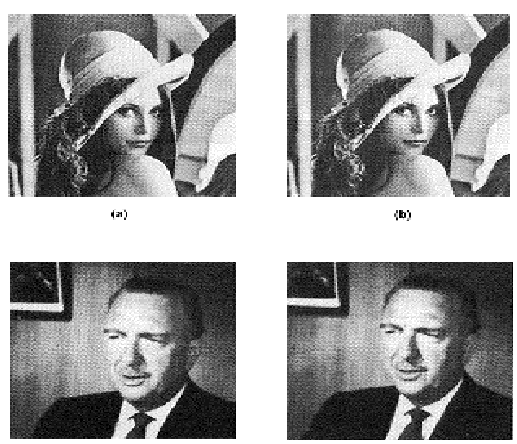
$$H = -.5\log .5 - .25\log .25 - .125\log .125 -$$
$$.125\log .125 = 1.75$$

Image Compression

1.58

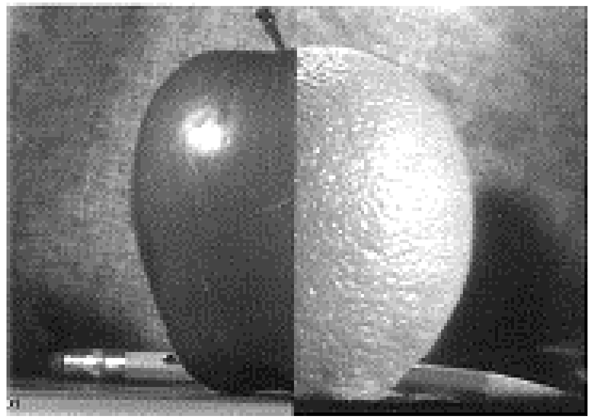
1

.73

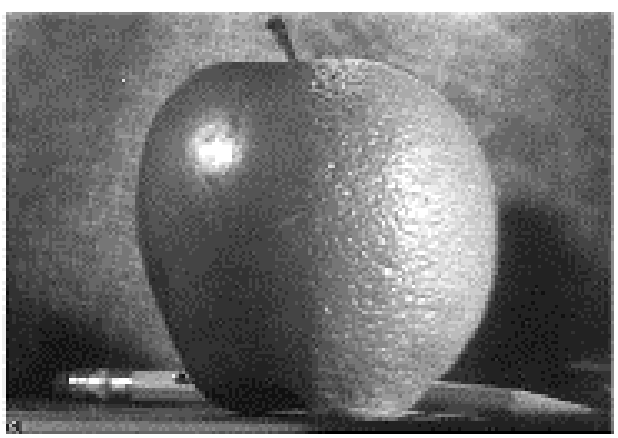


(a) (b) (c) (d)

Combining Apple & Orange



Combining Apple & Orange



Algorithm

- Generate Laplacian pyramid Lo of orange image.
- Generate Laplacian pyramid La of apple image.
- Generate Laplacian pyramid Lc by copying left half of nodes at each level from apple and right half of nodes from orange pyramids.
- Reconstruct combined image from Lc.

- http://ww-bcs.mit.edu/people/adelson/papers.html
  - The Laplacian Pyramid as a compact code, Burt and Adelson, IEEE Trans on Communication, 1983.

- http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-source.html
- ftp://csd.uwo.ca/pub/vision

  Performance of optical flow techniques,

Barron, Fleet and Beauchermin

Algorithm-2 (Optical Flow)

- Create Gaussian pyramid of both frames.
- Repeat
  - apply algorithm-1 at the current level of pyramid.
  - propagate flow by using bilinear interpolation to the next level, where it is used as an initial estimate.
  - Go back to step 2

Horn&Schunck Method

- Good only for translation model.
- Oversmoothing of boundaries.
- Does not work well for real sequences.

Other Optical Flow Methods

Important Issues

- What motion model?
- What function to be minimized?
- What minimization method?

Minimization Methods

- Least Squares fit
- Weighted Least Squares fit
- Newton-Raphson
- Gradient Descent
- Levenberg-Marquadet

Lucas & Kanade (Least Squares)

- Optical flow eq

$$f_x u + f_y v = -f_t$$

- Consider 3 by 3 window

$$f_{x1} u + f_{y1} v = -f_{t1}$$

$$\vdots$$

$$f_{x9} u + f_{y9} v = -f_{t9}$$

$$\begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ \\ f_{x9} & f_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ \\ -f_{t9} \end{bmatrix}$$

Lucas & Kanade

$$\mathbf{Au} = \mathbf{f_t}$$

$$\mathbf{A^T A u} = \mathbf{A^T f_t}$$

$$\mathbf{u} = (\mathbf{A^T A})^{-1} \mathbf{A^T f_t}$$

$$\Uparrow$$

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} (f_{xi} u + f_{yi} v + f_{ti})^2$$

Lucas & Kanade

$$\min \sum_{i=-2}^{2}\sum_{j=-2}^{2}(f_{xi}u+f_{yi}v+f_{ti})^2$$

$$\Downarrow$$

$$\sum(f_{xi}u+f_{yi}v+f_{ti})f_{xi}=0$$

$$\sum(f_{xi}u+f_{yi}v+f_{ti})f_{yi}=0$$

Lucas & Kanade

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi}f_{yi} \\ \sum f_{xi}f_{yi} & \sum f_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum f_{xi}f_{ti} \\ -\sum f_{yi}f_{ti} \end{bmatrix}$$

Lucas & Kanade

$$\min \sum_{i=-2}^{2} \sum_{j=-2}^{2} w_i (f_{xi} u + f_{yi} v + f_{ti})^2$$

$$\Downarrow$$

$$\mathbf{WAu} = \mathbf{Wf}_{\mathbf{t}}$$

$$\mathbf{A}^{\mathbf{T}}\mathbf{WAu} = \mathbf{A}^{\mathbf{T}}\mathbf{Wf}_{\mathbf{t}}$$

$$\mathbf{u} = (\mathbf{A}^{\mathbf{T}}\mathbf{WA})^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{Wf}_{\mathbf{t}}$$