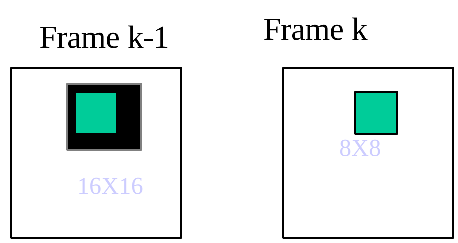
Lecture-18

Block-based & Token-based Optical Flow

Block Matching



Origin is at bottom right corner

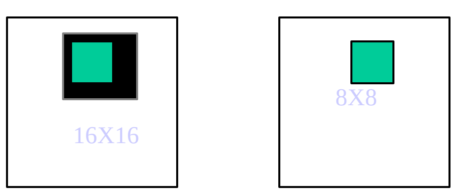
Block Matching

- For each 8X8 block, centered around pixel (x,y) in frame k, B_k
  - Obtain 16X16 block in frame k-1, enclosing (x,y), B_{k-1}
  - Compute Sum of Squares Differences (SSD) between 8X8 block, $B_{k,}$ and all possible 8X8 blocks in B_{k-1}
  - The 8X8 block in B_{k-1} centered around (x’,y’), which gives the least SSD is the match
  - The displacement vector (optical flow) is given by u=x-x’; v=y-y’

Sum of Squares Differences (SSD)

$$(u(x,y), v(x,y)) = \arg\min_{\substack{u=0\ldots-8 \\ v=0\ldots8}} \sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k(x+i, y+j) - f_{k-1}(x+i+u, y+j+v)\right)^2$$

Frame k-1 Frame k



Origin is at bottom right corner

Minimum Absolute Difference (MAD)

$$(u(x,y),v(x,y)) = \arg\min_{\substack{u=0\ldots-8\\ v=0\ldots 8}} \sum_{i=0}^{-7}\sum_{j=0}^{7} \left| \left(f_k(x+i,y+j) - f_{k-1}(x+i+u,y+j+v) \right) \right|$$

Maximum Matching Pixel Count (MPC)

$$T(x,y;u,v) = \begin{cases} 1 & \text{if } |f_k(x,y) - f_{k-1}(x+u,y+v)| \le t \\ 0 & \text{Otherwise} \end{cases}$$

$$(u(x,y),v(x,y)) = \arg\max_{\substack{u=0\ldots-8\\ v=0\ldots 8}} \sum_{i=0}^{-7}\sum_{j=0}^{7} T(x+i,y+j;u,v)$$

Cross Correlation

$$(u,v) = \arg\max_{\substack{u=0\ldots-8\\ v=0\ldots8}} \sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k(x+i, y+j)\right).\left(f_{k-1}(x+i+u, y+j+v)\right)$$

Potential problem

$$(u(x,y), v(x,y)) = \arg\min_{\substack{u=0\ldots-8\\ v=0\ldots8}} \sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k(x+i, y+j) - f_{k-1}(x+i+u, y+j+v)\right)^2$$

$$(u(x,y), v(x,y)) = \arg\min_{\substack{u=0\ldots-8\\ v=0\ldots8}} \sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k{}^2 - 2 f_k f_{k-1} + f_{k-1}{}^2\right)$$

Normalized Correlation

$$(u,v) = \arg\max_{\substack{u=0\ldots-8\\ v=0\ldots8}} \frac{\sum_{i=0}^{j=-7}\sum_{j=0}^{7}\left(f_k(x+i, y+j)\right).\left(f_{k-1}(x+i+u, y+j+v)\right)}{\sqrt{\sum_{i=0}^{-7}\sum_{j=0}^{7} f_{k-1}(x+i+u, y+j+v).\, f_{k-1}(x+i+u, y+j+v)}}$$

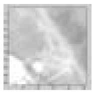
Mutual Correlation

$$(u,v) = \arg\max_{\substack{u=0\ldots-8\\ v=0\ldots8}} \frac{1}{64\,\sigma_1\sigma_2}\sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k(x+i,y+j)-\mu_1\right).\left(f_{k-1}(x+i+u,\,y+j+v)-\mu_2\right)$$

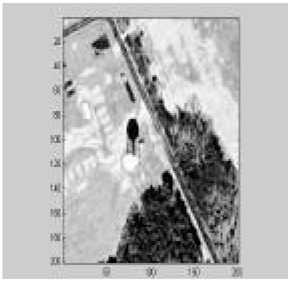
Sigma and mu are standard deviation and mean of patch-1 and patch-2 respectively

Correlation Surface

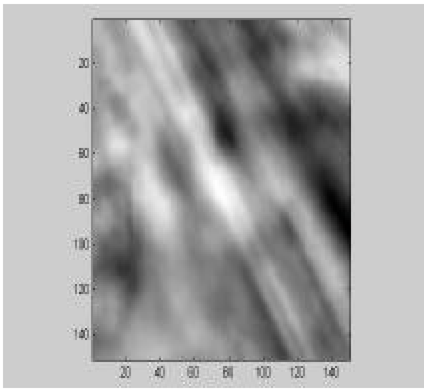
$$C(u,y) = \sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k(x+i,y+j)\right).\left(f_{k-1}(x+i+u,y+j+v)\right)$$



mission



reference



Correlation surface

Correlation Using FFT

$$C(u,y)=\sum_{i=0}^{-7}\sum_{j=0}^{7}\left(f_k(x+i,y+j)).(f_{k-1}(x+i+u,y+j+v)\right)$$

$$c(u,v)=f(x,y)\otimes g(x,y)$$

Fourier transform

convolution

$$C(w_1,w_2)=F(w_1,w_2).G(w_1,w_2)$$

Append smaller patch with zeros to make it equal to bigger patch
Find FFT of reference (*f(x,y)* and mission (*g(x,y)*) patches
multiply two FFTs
Find inverse FFT of the product, this will give you a correlation surface
Find the peak in the correlation surface
This method is faster for a large patch size

Phase Correlation

$$c(x,y)=f(x,y)\otimes g(x,y)$$

$$C(w_1,w_2)=F(w_1,w_2).G(w_1,w_2)$$

$$\tilde{C}(w_1,w_2)=\frac{F(w_1,w_2).G(w_1,w_2)}{|F(w_1,w_2).G(w_1,w_2)|}$$

Assume

$$f(u,v)=g(x+u,y+v)$$

Then

$$F(w_1,w_2)=G(w_1,w_2)e^{-(i(w_1u+w_2v))}$$

Now

$$\tilde{C}(w_1,w_2)=\frac{F(w_1,w_2).G(w_1,w_2)}{|F(w_1,w_2).G(w_1,w_2)|}=\frac{G(w_1,w_2)e^{-(i(w_1u+w_2v))}.G(w_1,w_2)}{|G(w_1,w_2)e^{-(i(w_1u+w_2v))}.G(w_1,w_2)|}=e^{-(i(w_1u+w_2v))}$$

$$c(x,y)=\boldsymbol{d}(x-u,y-v)$$

So it is very easy to find the peak in the correlation surface

Issue with Correlation

- Patch Size
- Search Area
- How many peaks

- Should use pyramids here too for large displacements

Token-based Optical Flow

- Find tokens
  - Moravec's interest operator
  - Corners
  - Edges
- Solve point correspondence

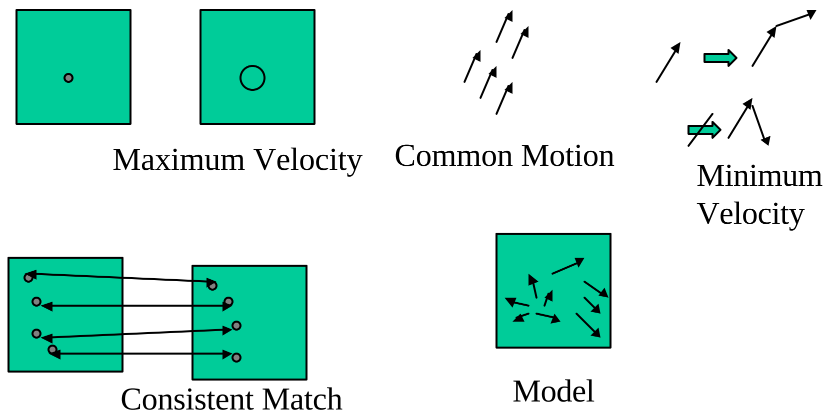
Point Correspondence

- Given *n* video frames taken at different time instants and *m* points in each frame, the motion correspondence problem deals with a mapping of a point in one frame to another point in the second frame such that no two points map onto the same point.

Key Points

- Constraints → Cost Function
- Algorithm → Minimize the cost function

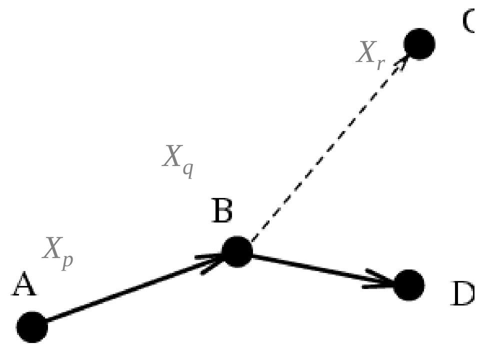
Constraints



Proximal Uniformity Constraint

- Most objects in the real world follow smooth paths and cover small distance in a small time.
  - Given a location of point in a frame, its location in the next fame lies in the proximity of its previous location.
  - The resulting trajectories are smooth and uniform.

Proximal Uniformity Constraint



Proximal Uniformity Constraint

Establish correspondence by minimizing:

$$d(X_p^{k-1}, X_q^k, X_r^{k+1}) = \frac{\| \overline{X_p^{k-1} X_q^k} - \overline{X_q^k X_r^{k+1}} \|}{\sum_{x=1}^{m} \sum_{z=1}^{m} | \overline{X_x^{k-1} X_y^k} - \overline{X_y^k X_z^{k+1}} \|} + \frac{\| \overline{X_q^k X_r^{k+1}} \|}{\sum_{x=1}^{m} \sum_{z=1}^{m} \| \overline{X_y^k X_z^{k+1}} \|}$$

Initial correspondence is known, for each x in the denominator of the first term y is known.

Greedy Algorithm

- For k=2 to n-1 do
- Construct M, an mxm matrix, with the points from k-th frame along the rows and points from (k+1)-th frame along the columns. Let

$$M[i, j] = d(X_p^{k-1}, X_q^k, X_r^{k+1})$$

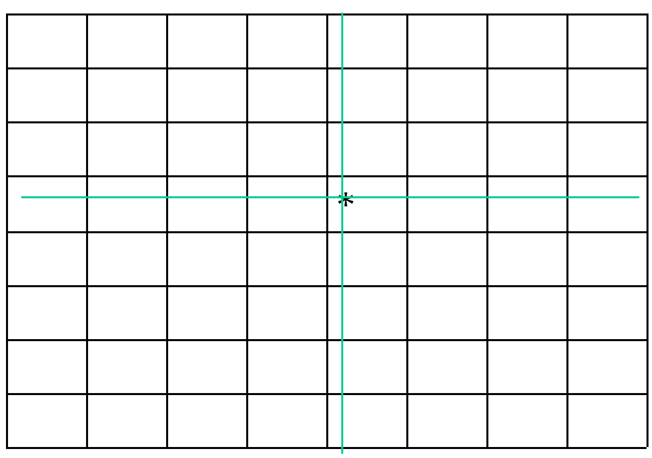
- for a=1 to m do
  - Identify the minimum element $[i, l_i]$ in each row i of M
  - Compute *priority matrix*, B, such that $B[i, l_i] = \sum_{j=1, j \neq l_i}^{m} M[i, j] + \sum_{k=1, k \neq i}^{m} M[k, l_i]$ for each *i*.
  - Select $[i, l_i]$ pair with highest *priority* value and make $f^k(i) = l_i$
  - Mask row *i* and column l_i from M.

http://www.cs.ucf.edu/~vision/papers/PAPER3.PDF

M[i,j]

j

i



$$M = \begin{bmatrix} .6 & .3 \\ .7 & .2 \end{bmatrix}$$

$$B = \begin{bmatrix} & .8 \\ & 1 \end{bmatrix}$$

Tracking Finger Tips