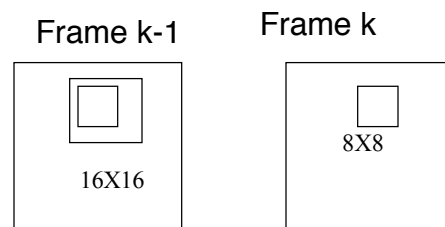


Lecture-18

Block-based & Token-based Optical Flow

Block Matching



Origin is at bottom right corner

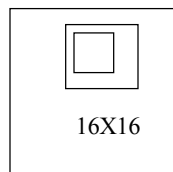
Block Matching

- 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 $u=x-x'$; $v=y-y'$

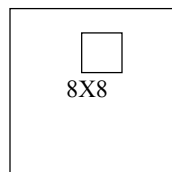
Sum of Squares Differences (SSD)

$$(U(x,y), V(x,y)) = \underset{\substack{u=0 \dots 8 \\ v=0 \dots 8}}{\operatorname{argmin}} \sum_{i=0}^7 \sum_{j=0}^7 (f_k(x+i, y+j) - f_{k-1}(x+i+u, y+j+v))^2$$

Frame k-1



Frame k



Origin is at bottom right corner

Minimum Absolute Difference (MAD)

$$(U(x, y), V(x, y)) = \underset{\substack{u=0 \dots 8 \\ v=0 \dots 8}}{\operatorname{argmin}} \sum_{i=0}^7 \sum_{j=0}^7 |f_k(x+i, y+j) - f_{k'}(x+i+u, y+j+v)|$$

Maximum Matching Pixel Count (MPC)

$$T(x, y; u, v) = \begin{cases} 1 & \text{if } |f_k(x, y) - f_{k'}(x+u, y+v)| \leq t \\ 0 & \text{Otherwise} \end{cases}$$

$$(U(x, y), V(x, y)) = \underset{\substack{u=0 \dots 8 \\ v=0 \dots 8}}{\operatorname{argmax}} \sum_{i=0}^7 \sum_{j=0}^7 T(x+i, y+j; u, v)$$

Cross Correlation

$$(U, V) = \arg \max_{\substack{u=0 \dots 8 \\ v=0 \dots 8}} \prod_{i=0}^7 \prod_{j=0}^7 (f_k(x+i, y+j)) \cdot (f_{k'}(x+i+u, y+j+v))$$

Potential problem

$$(U(x, y), V(x, y)) = \arg \min_{\substack{u=0 \dots 8 \\ v=0 \dots 8}} \prod_{i=0}^7 \prod_{j=0}^7 (f_k(x+i, y+j) - f_{k'}(x+i+u, y+j+v))^2$$

$$(U(x, y), V(x, y)) = \arg \min_{\substack{u=0 \dots 8 \\ v=0 \dots 8}} \prod_{i=0}^7 \prod_{j=0}^7 (f_k^2 - 2f_k f_{k'} + f_{k'}^2)$$

Normalized Correlation

$$(U, V) = \arg \max_{\substack{u=0 \dots 8 \\ v=0 \dots 8}} \frac{\prod_{i=0}^7 \prod_{j=0}^7 (f_k(x+i, y+j)) \cdot (f_{k'}(x+i+u, y+j+v))}{\sqrt{\prod_{i=0}^7 \prod_{j=0}^7 f_k(x+i, y+j) \cdot f_{k'}(x+i+u, y+j+v)}}$$

Mutual Correlation

$$(U, V) = \arg \max_{\substack{u=0 \dots 8 \\ v=0 \dots 8}} \frac{1}{64} \sum_{i=0}^7 \sum_{j=0}^7 (f_k(x+i, y+j) - \mu_1) \cdot (f_k(x+i+u, y+j+v) - \mu_2)$$

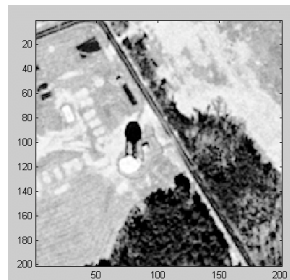
Sigma and mu are standard deviation and mean of patch-1 and patch-2 respectively

Correlation Surface

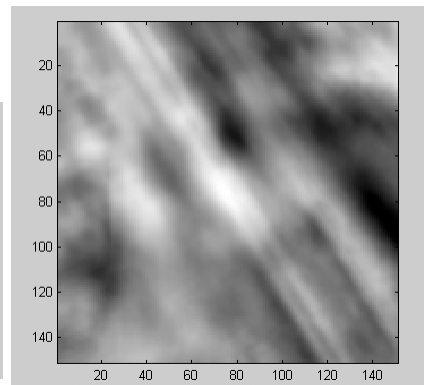
$$C(u, v) = \sum_{i=0}^7 \sum_{j=0}^7 (f_k(x+i, y+j) - \mu_1) \cdot (f_k(x+i+u, y+j+v) - \mu_2)$$



mission



reference



Correlation surface

Correlation Using FFT

$$C(u, v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (f_k(x+i, y+j)) \cdot (f_k(x+i+u, y+j+v))$$

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



Fourier transform

convolution

$$C(w_1, w_2) = F(w_1, w_2) \cdot G(w_1, w_2)$$

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

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

$$C(w_1, w_2) = F(w_1, w_2) \cdot G^*(w_1, w_2)$$

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

Assume

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

Then

$$F(w_1, w_2) = G(w_1, w_2) e^{i(w_1 u + w_2 v)}$$

Now

$$\tilde{C}(w_1, w_2) = \frac{F(w_1, w_2) \cdot G^*(w_1, w_2)}{|F(w_1, w_2) \cdot G^*(w_1, w_2)|} = \frac{G(w_1, w_2) e^{i(w_1 u + w_2 v)} \cdot G^*(w_1, w_2)}{|G(w_1, w_2) e^{i(w_1 u + w_2 v)} \cdot G^*(w_1, w_2)|} = e^{i(w_1 u + w_2 v)}$$

$$c(x, y) = \int \int x \int u, y \int v$$

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

Issues 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

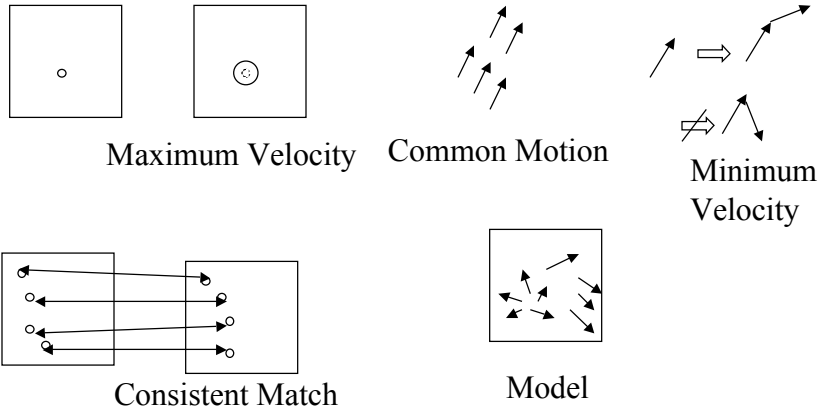
Point Correspondence

- 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 \rightarrow Cost Function
- Algorithm \rightarrow Minimize the cost function

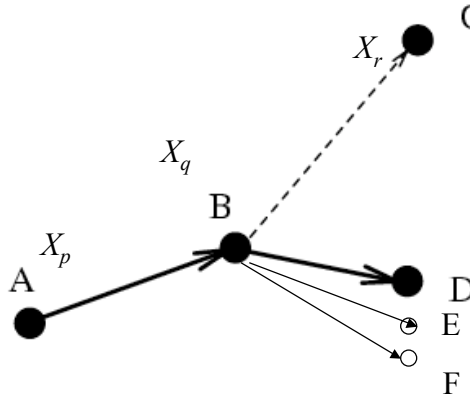
Constraints



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 frame 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:

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

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

Greedy Algorithm

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

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

- 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 $\varphi^k(i) = l_i$
 - Mask row i and column l_i from M .

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

$M[i, j]$

j

			*				

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

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

i

Tracking Finger Tips

