Lecture-18

Block-based & Token-based Optical Flow

Block Matching

Frame k-1

16X16

Frame k

5X5

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)

\[(u(x, y), v(x, y)) = \arg \min_{i=0, \ldots, 7} \sum_{i=0}^{7} \sum_{j=0}^{7} \left( f_i(x+i, y+j) - f_{k-1}(x+i+u, y+j+v) \right)^2 \]

Origin is at bottom right corner
Minimum Absolute Difference (MAD)

\[ (u(x, y), v(x, y)) = \arg \min_{u=0, \ldots, 8} \sum_{i=0}^{7} \sum_{j=0}^{7} |f_i(x+i, y+j) - f_{i-1}(x+i+u, y+j+v)| \]

Maximum Matching Pixel Count (MPC)

\[ T(x, y; u, v) = \begin{cases} 1 & \text{if } |f_i(x, y) - f_{i-1}(x+u, y+v)| \leq t \\ 0 & \text{Otherwise} \end{cases} \]

\[ (u(x, y), v(x, y)) = \arg \max_{u=0, \ldots, 8} \sum_{i=0}^{7} \sum_{j=0}^{7} T(x+i, y+j; u, v) \]
Cross Correlation

\[ (u, v) = \arg \max_{u=0...8} \sum_{v=0...8} \sum_{i=0}^{7} \sum_{j=0}^{7} (f_k(x+i, y+j))(f_{k-1}(x+i+u, y+j+v)) \]

Potential problem

\[ (u(x, y), v(x, y)) = \arg \min_{u=0...8} \sum_{v=0...8} \sum_{i=0}^{7} \sum_{j=0}^{7} (f_k(x+i, y+j) - f_{k-1}(x+i+u, y+j+v))^2 \]

\[ (u(x, y), v(x, y)) = \arg \min_{u=0...8} \sum_{v=0...8} \sum_{i=0}^{7} \sum_{j=0}^{7} (f_k^2 - 2f_kf_{k-1} + f_{k-1}^2) \]

Normalized Correlation

\[ (u, v) = \arg \max_{u=0...8} \sum_{v=0...8} \frac{\sum_{i=0}^{7} \sum_{j=0}^{7} (f_k(x+i, y+j))(f_{k-1}(x+i+u, y+j+v))}{\sqrt{\sum_{i=0}^{7} \sum_{j=0}^{7} f_{k-1}(x+i+u, y+j+v) \cdot f_{k-1}(x+i+u, y+j+v)}} \]
Mutual Correlation

\[ (u, v) = \arg \max_{u=0 \ldots 8, \, \, v=0 \ldots 8} \frac{1}{\sigma_1 \sigma_2} \sum_{i=0}^{7} \sum_{j=0}^{7} \left( f_i (x + i, y + j) - \mu_i \right) \left( f_{i-1} (x + i + u, y + j + v) - \mu_i \right) \]

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

Correlation Surface

\[ C(u, y) = \sum_{i=0}^{7} \sum_{j=0}^{7} \left( f_i (x + i, y + j) \right) \left( f_{i-1} (x + i + u, y + j + v) \right) \]

Mission, reference, Correlation surface
Correlation Using FFT

\[ C(u, y) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (f_k(x+i, y+j) \ast f_{k-1}(x+i, y+j)) \]

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

\[ C(w_1, w_2) = F(w_1, w_2) \ast G(w_1, w_2) \]

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

\[ c(x, y) = f(x, y) \otimes g(x, y) \]

\[ C(w_1, w_2) = F(w_1, w_2) \ast G(w_1, w_2) \]

\[ \tilde{C}(w_1, w_2) = \frac{F(w_1, w_2) \ast G(w_1, w_2)}{|F(w_1, w_2) \ast G(w_1, w_2)|} \]

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
\[ \tilde{C}(w_1, w_2) = \frac{F(w_1, w_2) \ast G(w_1, w_2)}{|F(w_1, w_2) \ast G(w_1, w_2)|} = \frac{G(w_1, w_2) e^{-i(w_1 u+w_2 v)} \ast G(w_1, w_2)}{|G(w_1, w_2) e^{-i(w_1 u+w_2 v)} \ast G(w_1, w_2)|} = e^{-i(w_1 u+w_2 v)} \]

\[ c(x, y) = \delta(x-u, y-v) \]

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

- Maximum Velocity
- Consistent Match
- Common Motion
- Minimum Velocity

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

Establish correspondence by minimizing:

$$\delta(X_{p}^{k-1}, X_{q}^{k}, X_{r}^{k+1}) = \sum_{x=1}^{m} \sum_{z=1}^{m} \frac{\| X_{p}^{k-1} X_{q}^{k} - X_{q}^{k} X_{r}^{k+1} \|}{\| X_{x}^{k-1} X_{y}^{k} - X_{y}^{k} X_{z}^{k+1} \|} + \frac{\| X_{q}^{k} X_{r}^{k+1} \|}{\| X_{y}^{k} X_{z}^{k+1} \|}$$

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
  \[
  M[i, j] = \delta(X_{p}^{k-1}, X_{q}^{k}, X_{r}^{k+1})
  \]

• 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
    \[
    B[i,j] = \sum_{i' \neq i} M[i', j] + \sum_{j' \neq j} M[i, j']
    \]
    for each \( i \).
  - Select \([i,l_i]\) pair with highest priority value and make \( \Phi^k(i) = l_i \)
  - Mask row \( i \) and column \( l_i \) from \( M \).


\[
M = \begin{bmatrix}
0.6 & 0.3 \\
0.7 & 0.2 
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0.8 \\
1 
\end{bmatrix}
\]
Tracking Finger Tips