Lecture-8

Feature-based Registration

Steps in Feature-based Registration

- Find features
- Establish correspondences between features in two images (correlation, point correspondence)
- Fit transformation
- Apply transformation (warp)

Features

- All pixels (spatiotemporal approach)
- Corner points
- Interest points
- Straight lines
- Line intersections
- Features obtained using Gabor/Wavelet filters
- …

Transformations

- Affine
- Projective
- Psuedo-perspective
- Rational polynomial

Good Features to Track

- Corner like features
- Moravec's Interest Operator

Corner like features

$$C = \begin{bmatrix} \sum_Q f_x^2 & \sum_Q f_x f_y \\ \sum_Q f_x f_y & \sum_Q f_y^2 \end{bmatrix}$$

Q is an image patch

Eigen Values

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Corners

- For perfectly uniform region $\lambda_1 = \lambda_2 = 0$
- If Q contains an ideal step edge, then

$$\lambda_2 = 0, \lambda_1 > 0$$

- if Q contains a corner of black square on white background

$$\lambda_1 \geq \lambda_2 \succ 0$$

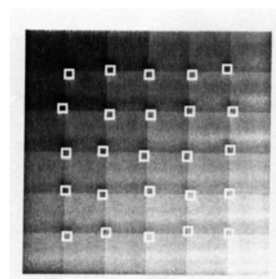
Algorithm Corners

- Compute the image gradient (f_x, f_y) over entire image f.
- For each image point p:
  - form the matrix C over (2N+1)X(2N+1) neighborhood Q of p;
  - compute the smallest eigenvalue of C;
  - if eigenvalue is above some threshold, save the coordinates of p into a list L.

Algorithm Corners

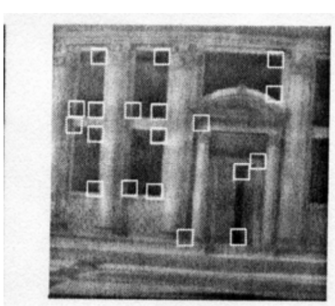
- Sort L in decreasing order of eigenvalues.
- Select the top candidate corner, and perform Non-maxima suppression
  - Scanning the sorted list top to bottom: for each current point, p, delete all other points on the list which belong to the neighborhood of p.

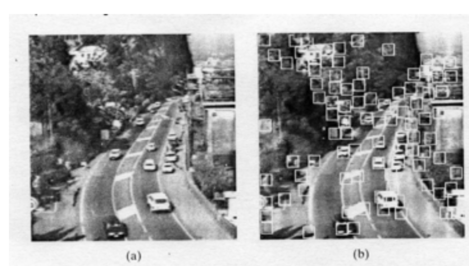
Results



(a)

Results

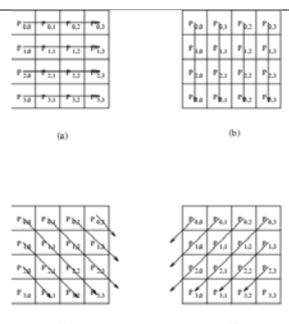


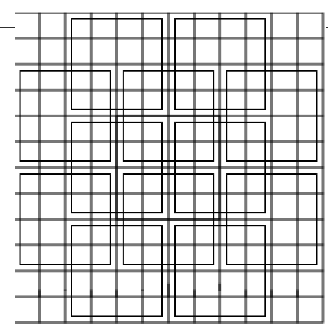


Moravec’s Interest Operator

Algorithm

- Compute four directional variances in horizontal, vertical, diagonal and anti-diagonal directions for each 4 by 4 window.
- If the minimum of four directional variances is a local maximum in a 12 by 12 overlapping neighborhood, then that widow (point) is interesting.





$$V_h = \sum_{j=0}^{3}\sum_{i=0}^{2}(P(x+i, y+j) - P(x+i+1, y+j))^2$$

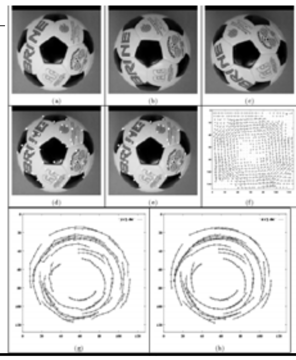
$$V_v = \sum_{j=0}^{2}\sum_{i=0}^{3}(P(x+i, y+j) - P(x+i, y+j+1))^2$$

$$V_d = \sum_{j=0}^{2}\sum_{i=0}^{2}(P(x+i, y+j) - P(x+i+1, y+j+1))^2$$

$$V_a = \sum_{j=0}^{2}\sum_{i=1}^{3}(P(x+i, y+j) - P(x+i-1, y+j+1))^2$$

$$V(x,y) = \min(V_h(x,y), V_v(x,y), V_d(x,y), V_a(x,y))$$

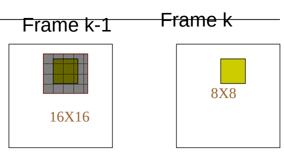
$$I(x,y) = \begin{cases} 1 & if V(x,y) local \max \\ 0 & 0therwise \end{cases}$$



Correlation

- Similarity/Dissimilarity Measures
  - Sum of Squares Difference (SSD)
  - Normalized Correlation
  - Mutual Correlation
  - Mutual information *H(X;Y)=H(X)-H(X|Y)*
- Use
  - Gray levels
  - Laplacian of Gaussian
  - Gradient magnitude

Block Matching



Block Matching

- For each 8X8 block, centered around pixel (x,y) in frame k, B_k
  - Obtain 16X16 block in frame k-1, centered around (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)

$$(u(x,y),v(x,y)) = \arg\min_{u,v=-3\ldots3} \sum_{i=0}^{-7}\sum_{j=0}^{-7}\left(f_k(x+i,y+j) - f_{k-1}(x+i+u,y+j+v)\right)^2$$

Minimum Absolute Difference (MAD)

$$(u(x,y),v(x,y)) = \arg\min_{u,v=-3\ldots3} \sum_{i=0}^{-7}\sum_{j=0}^{-7}\left|\left(f_k(x+i,y+j) - f_{k-1}(x+i+u,y+j+v)\right)\right|$$

Maximum Matching Pixel Count (MPC)

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

$$(u(x,y),v(x,y))=\arg\max_{u,v=-3\ldots3}\sum_{i=0}^{-7}\sum_{j=0}^{-7}T(x+i,y+j;u,v)$$

Cross Correlation

$$(u,v)=\arg\max_{u,v=-3\ldots3}\sum_{i=0}^{-7}\sum_{j=0}^{-7}\left(f_k(x+i,y+j)).(f_{k-1}(x+i+u,y+j+v)\right)$$

Normalized Correlation

$$(u,v)=\arg\max_{u,v=-3..3}\frac{\sum_{i=0}^{-7}\sum_{j=0}^{-7}\left((f_k(x+i,y+j)-\mu_1).(f_{k-1}(x+i+u,y+j+v)-\mu_2)\right)}{\sqrt{\left(\sum_{i=0}^{-7}\sum_{j=0}^{-7}(f_k(x+i,y+j)-\mu_1)^2\right)\left(\sum_{i=0}^{-7}\sum_{j=0}^{-7}(f_{k-1}(x+i+u,y+j+v)-\mu_2)^2\right)}}$$

μ_1 and μ_2 are the means of patch-1 and patch-2 respectively.

Mutual Correlation

$$(u,v)=\arg\max_{u,v=-3\ldots3}\frac{1}{64\sigma_1\sigma_2}\sum_{i=0}^{-7}\sum_{j=0}^{-7}\left(f_k(x+i,y+j)-\mu_1).(f_{k-1}(x+i+u,y+j+v)-\mu_2\right)$$

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

Issues with Correlation

- Patch Size
- Search Area
- How many peaks

Spatiotemporal Models

- First order Taylor series

$$f(x,y,t)=f(x+dx,y+dy,t+dt)$$

$$f(x,y,t)=f(x,y,t)+\frac{\partial f}{\partial x}dx+\frac{\partial f}{\partial y}dy+\frac{\partial f}{\partial t}dt$$

$$f_x u+f_y v+f_t=0$$

$$\left[\sum X^T \mathbf{f_X}\mathbf{f_X}^T X\right]\delta a=-\sum X^T \mathbf{f_X} f_t$$

Correlation

Bilinear and Pseudo-Perspective

$$(\sum \Phi\Phi^T)\mathbf{q} = -\sum f_t \Phi$$

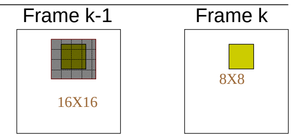
$$\Phi^T = \left[f_x(xy, x, y, 1), \quad f_y(xy, x, y, 1) \right]$$ **bilinear**

$$\Phi^T = \left[f_x(x,y,1) \quad f_y(x,y,1) \quad c_1 \quad c_2 \right]$$

$$c_1 = x^2 f_x + xyf_x$$ **Pseudo perspective**

$$c_2 = xyf_x + y^2 f_y$$

Correlation Vs Spatiotemporal



Correlation Complexity

- m*m multiplications and additions
- 2*m*m additions and 2 divisions for two means
- 2*m*m multiplications and additions for variances

$$(u,v) = \arg\max_{u,v=-3\ldots3} \frac{1}{64\sigma_1\sigma_2} \sum_{i=0}^{-7}\sum_{j=0}^{-7} (f_k(x+i, y+j) - \mu_1).(f_{k-1}(x+i+u, y+j+v) - \mu_2)$$

Spatiotemporal Complexity

- 3* m*m subtractions for sptiotemporal derivatives
- (36+6)*m*m additions for generating linear system
- 6*6*6 multiplications and additions for solving 6 by 6 linear system

$$\left[\sum X^T \mathbf{f_x} \mathbf{f_x^T} X\right] \delta a = -\sum X^T \mathbf{f_x} f_t$$

Feature-based Matching

Feature-based Matching

- The input is formed by f1 and f2, two frames of an image sequence.
- Let Q1, Q2 and Q' be three NXN image regions.
- Let "d" be the unknown displacement vector between f1 and f2 of a feature point "p", on which Q1 is centered.

Algorithm

- Set d=0, center Q1 on p1.

1. Estimate the displacement "d0" of "p", center of "Q1", using Lucas and Kanade method. Let d=d+d0.
2. Let Q' bet the patch obtained by warping Q1 according to "d0". Compute Sum of Square (SSD) difference between new patch Q' and corresponding patch Q2 in frame f2.
3. If SSD is more than a threshold, set Q1=Q' and go to step 1, otherwise exit.

Lucas & Kanade (Least Squares)

- Optical flow eq

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

- Consider n by n window

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

$$\vdots$$

$$f_{xn^2} u + f_{yn^2} v = -f_{tn^2}$$

$$\begin{bmatrix} f_{x1} & f_{y1} \\ \vdots & \vdots \\ f_{xn^2} & f_{yn^2} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t1} \\ \vdots \\ -f_{tn^2} \end{bmatrix}$$

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

Shi-Tomasi-Kanade(STK) Tracker

$$I(\delta(\mathrm{x}), t+\tau) = I(\mathrm{x}, t)$$

$$\delta(\mathrm{x}) = A\mathrm{x} + d$$

$$\varepsilon = \sum_W \left[I(A\mathrm{x} + d, t+\tau) - I(\mathrm{x}, t) \right]^2$$

After the first order Taylor expansion at A=I and d=0, we can get a linear 6×6 system:

$$Tz = f$$

$$z = \begin{bmatrix} A_{11} & A_{12} & A_{21} & A_{22} & d_1 & d_2 \end{bmatrix}^T$$

$$f = I_t \sum_W \begin{bmatrix} xI_x & yI_x & xI_y & yI_y & I_x & I_y \end{bmatrix}^T$$

$$T = \sum_W \left(\begin{bmatrix} xI_x & yI_x & xI_y & yI_y & I_x & I_y \end{bmatrix}^T \begin{bmatrix} xI_x & yI_x & xI_y & yI_y & I_x & I_y \end{bmatrix} \right)$$

Shi-Tomasi-Kanade(STK) Tracker

- Advantages:
  - Easy to implement .
  - Works very well with a small transformation.
- Drawbacks:
  - Fail to track in presence of a large rotation.
- Reason:
  - The rotation component implied in affine model is non-linear.

Useful Links

http://twtelecom.dl.sourceforge.net/sourceforge/opencvlibrary/OpenCVReferenceManual.pdf

http://vision.stanford.edu/~birch/klt/.