Local Features

Readings: Shi and Tomasi
Lowe

Due: Problem Set #2

March 4, 2008

Today - Local Features

Matching image features across images is important for recognition (indexing) and tracking

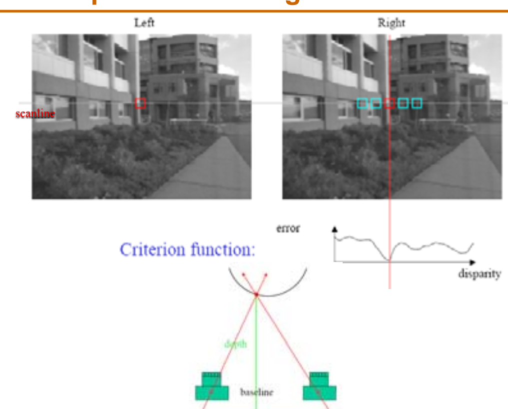
- Interest Features
- Correspondences
- Affine Patch Tracking
- Descriptors – Scale and Rotation Invariant Descriptors (Lowe)

Correspondence using window matching

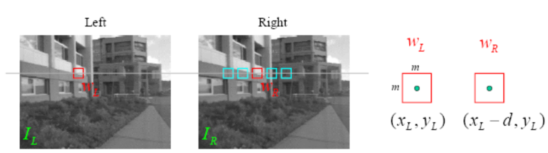
Points are highly individually ambiguous…

More unique matches are possible with small regions of image.

Correspondence using window matching



Sum of Squared (Pixel) Differences



w_L and w_R are corresponding m by m windows of pixels.

We define the window function :

$$W_m(x,y) = \{u,v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x,y,d) = \sum_{(u,v)\in W_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2$$

Image Normalization

- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- The cameras do not see exactly the same surfaces, so their overall light levels can differ.
- For these reasons and more, it is a good idea to normalize the pixels in each window:

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v)$$ Average pixel

$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v)\in W_m(x,y)} [I(u,v)]^2}$$ Window magnitude

$$\hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}}$$ Normalized pixel

Images as Vectors

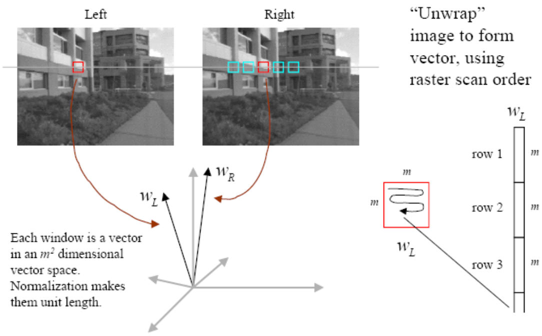
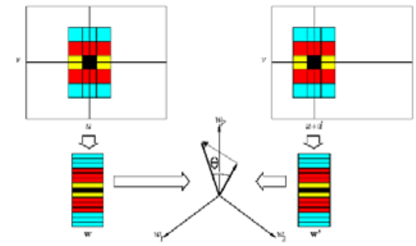


Image Windows as Vectors



Possible Metrics

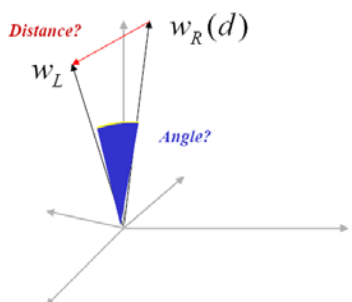
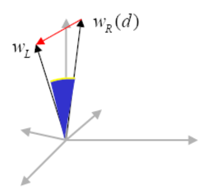


Image Metrics



(Normalized) Sum of Squared Differences

$$C_{SSD}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

$$= \|w_L - w_R(d)\|^2$$

Normalized Correlation

$$C_{NC}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

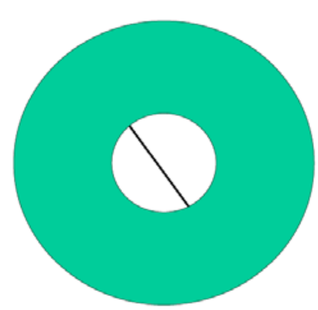
$$= w_L \cdot w_R(d) = \cos\theta$$

$$d^* = \arg\min_d \|w_L - w_R(d)\|^2 = \arg\max_d w_L \cdot w_R(d)$$

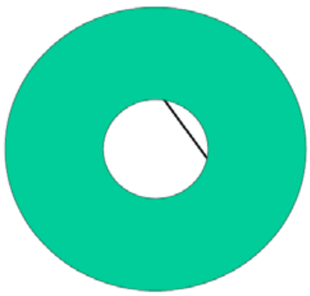
Local Features

Not all windows are good for matching

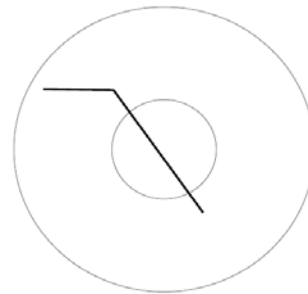
Aperture Problem and Normal Flow



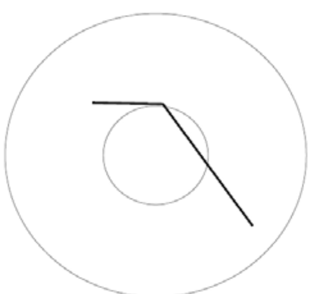
Aperture Problem and Normal Flow



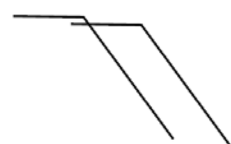
Aperture Problem and Normal Flow



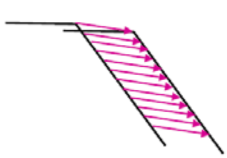
Aperture Problem and Normal Flow



Aperture Problem and Normal Flow



Aperture Problem and Normal Flow



Optical flow constraint equation

Brightness should stay constant as you track motion

$$I(x+u\delta t, y+v\delta t, t+\delta t) = I(x,y,t)$$

1st order Taylor series, valid for small δt

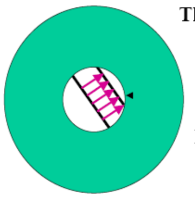
$$I(x,y,t) + u\delta t I_x + v\delta t I_y + \delta t I_t = I(x,y,t)$$

Constraint equation

$$u I_x + v I_y + I_t = 0$$

"BCCE" - Brightness Change Constraint Equation

Aperture Problem and Normal Flow



The gradient constraint:

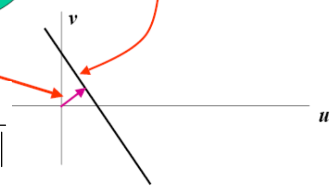
$$\mathbf{I}_x u + \mathbf{I}_y v + \mathbf{I}_t = 0$$

$$\nabla \mathbf{I} \cdot \mathbf{u} = -\mathbf{I}_t$$

Defines a line in the *(u,v)* space

Normal Flow:

$$\mathbf{u}_\perp = -\frac{\mathbf{I}_t}{|\nabla \mathbf{I}|} \frac{\nabla \mathbf{I}}{|\nabla \mathbf{I}|}$$



Lucas-Kanade: Integrade Gradients Over Patch

Assume a single velocity for all pixels within an image patch

$$E(u,v) = \sum_{x,y \in \Omega} \left(I_x(x,y)u + I_y(x,y)v + I_t \right)^2$$

Solve with:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

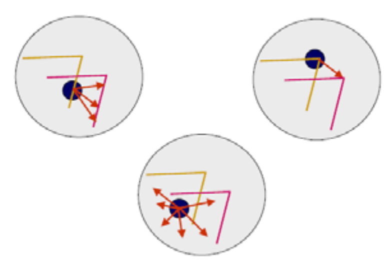
On the LHS: sum of the 2x2 outer product tensor of the gradient vector

$$\left(\sum \nabla I \nabla I^T \right) \vec{U} = -\sum \nabla I I_t$$

Selecting Good Features

- What's a "good feature"?
  - Satisfies brightness constancy
  - Has sufficient texture variation
  - Does not have too much texture variation
  - Corresponds to a "real" surface patch
  - Does not deform too much over time

Local Patch Analysis



23

Good Features to Track

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

$$\mathbf{A} \quad \mathbf{u} = \mathbf{b}$$

When is This Solvable?

- **A** should be invertible
- **A** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **A** should not be too small
- **A** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Both conditions satisfied when $min(\lambda_1, \lambda_2) > c$

Harris Corner Detector

Same idea, based on the idea of auto-correlation

Important difference in all directions => interest point

Harris Corner Detector

Auto-correlation function for a point (x, y) and a shift $(\Delta x, \Delta y)$

$$f(x,y) = \sum_{(x_k,y_k)\in W} \left(I(x_k,y_k) - I(x_k+\Delta x, y_k+\Delta y)\right)^2$$

Discret shifts can be avoided with the auto-correlation matrix

with $I(x_k+\Delta x, y_k+\Delta y) = I(x_k,y_k) + \begin{pmatrix} I_x(x_k,y_k) & I_y(x_k,y_k) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$

$$f(x,y) = \sum_{(x_k,y_k)\in W} \left(\begin{pmatrix} I_x(x_k,y_k) & I_y(x_k,y_k) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right)^2$$

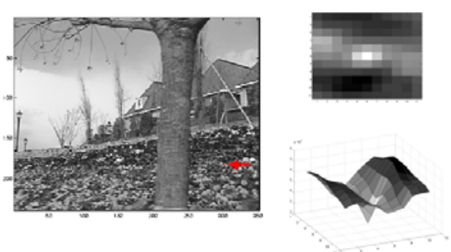
Harris Corner Detector

Auto-correlation matrix

$$= \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{bmatrix} \sum_{(x_k,y_k)\in W} (I_x(x_k,y_k))^2 & \sum_{(x_k,y_k)\in W} I_x(x_k,y_k) I_y(x_k,y_k) \\ \sum_{(x_k,y_k)\in W} I_x(x_k,y_k) I_y(x_k,y_k) & \sum_{(x_k,y_k)\in W} (I_y(x_k,y_k))^2 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

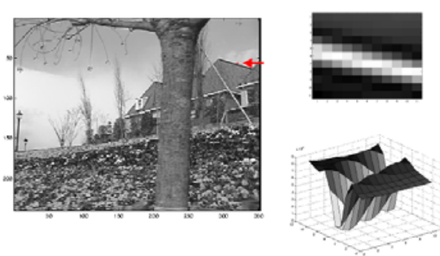
- Auto-correlation matrix
  - captures the structure of the local neighborhood
  - measure based on eigenvalues of this matrix
    - 2 strong eigenvalues => interest point
    - 1 strong eigenvalue => contour
    - 0 eigenvalue => uniform region
- Interest point detection
  - threshold on the eigenvalues
  - local maximum for localization

Selecting Good Features



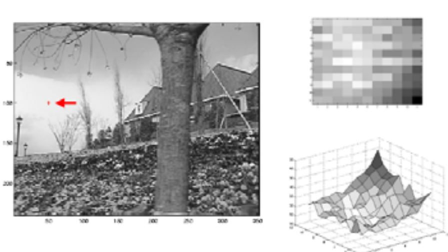
λ_1 and λ_2 are large

Selecting Good Features



large λ_1, small λ_2

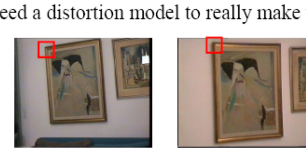
Selecting Good Features



small λ_1, small λ_2

Feature Distortion

- Feature may change shape over time
  - Need a distortion model to really make this work



Find displacement (u,v) that minimizes SSD error over feature region

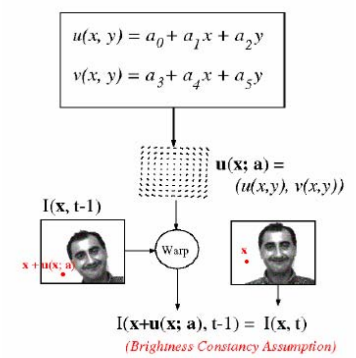
$$\sum_{(x,y)\in F\subset J} [I(W_x(x,y), W_y(x,y)) - J(x,y)]^2$$

(minimize with respect to W_x and W_y)

Shi and Tomasi: use affine model for verification

$$W_x(x,y) = ax + by + c$$
$$W_y(x,y) = ex + fy + g$$

Affine Motion



Affine Motion

$$u(x,y) = a_1 + a_2 x + a_3 y$$
$$v(x,y) = a_4 + a_5 x + a_6 y$$

Substituting into the B.C.C.E.:

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

$$I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0$$

Each pixel provides 1 linear constraint in 6 *global* unknowns
(minimum 6 pixels necessary)

Least Square Minimization (over all pixels):

$$Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2$$

Today - Local Features

Matching image features across images is important for recognition (indexing) and tracking

- Interest Features
- Correspondences
- Affine Patch Tracking
- Descriptors – Scale and Rotation Invariant Descriptors
  (Lowe)

CVPR 2003 Tutorial

Recognition and Matching Based on Local Invariant Features

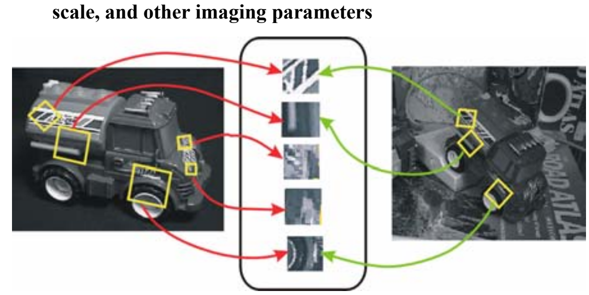
David Lowe
Computer Science Department
University of British Columbia

Object Recognition

- Definition: **Identify an object and determine its pose and model parameters**
- Commercial object recognition
  - Currently a $4 billion/year industry for inspection and assembly
  - Almost entirely based on template matching
- Upcoming applications
  - Mobile robots, toys, user interfaces
  - Location recognition
  - Digital camera panoramas, 3D scene modeling

Invariant Local Features

- **Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters**



SIFT Features

Advantages of invariant local features

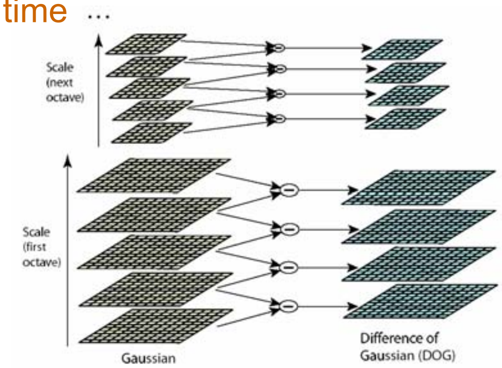
- Locality: **features are local, so robust to occlusion and clutter (no prior segmentation)**
- Distinctiveness: **individual features can be matched to a large database of objects**
- Quantity: **many features can be generated for even small objects**
- Efficiency: **close to real-time performance**
- Extensibility: **can easily be extended to wide range of differing feature types, with each adding robustness**

Scale invariance

Requires a method to repeatably select points in location and scale:

- The only reasonable scale-space kernel is a Gaussian (Koenderink, 1984; Lindeberg, 1994)
- An efficient choice is to detect peaks in the difference of Gaussian pyramid (Burt & Adelson, 1983; Crowley & Parker, 1984 – but examining more scales)
- Difference-of-Gaussian with constant ratio of scales is a close approximation to Lindeberg's scale-normalized Laplacian (can be shown from the heat diffusion equation)

Scale space processed one octave at a time …



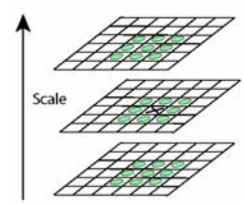
Key point localization

- **Detect maxima and minima of difference-of-Gaussian in scale space**
- **Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)**
- **Taylor expansion around point:**

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

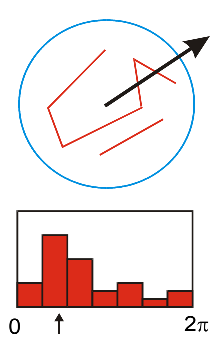
- **Offset of extremum (use finite differences for derivatives):**

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$$



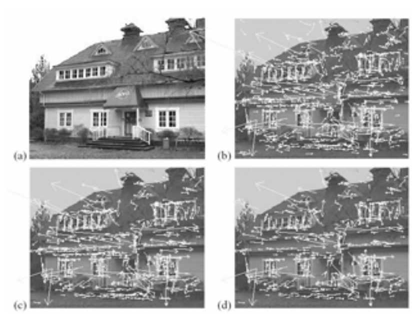
Select canonical orientation

- **Create histogram of local gradient directions computed at selected scale**
- **Assign canonical orientation at peak of smoothed histogram**
- **Each key specifies stable 2D coordinates (x, y, scale, orientation)**



Example of keypoint detection

Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)

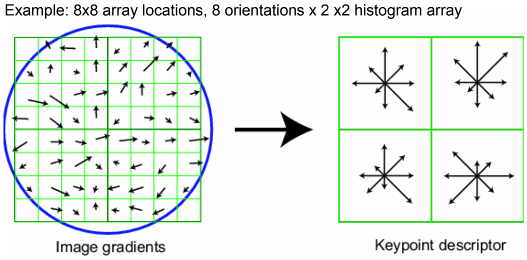


(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

SIFT vector formation

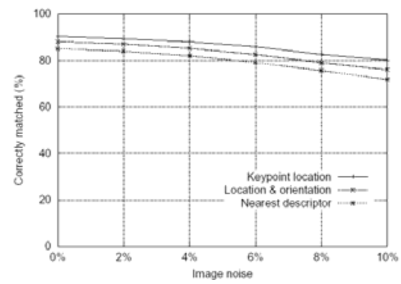
- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions

Example: 8x8 array locations, 8 orientations x 2 x2 histogram array



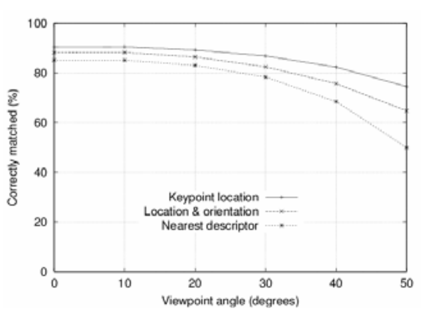
Feature stability to noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



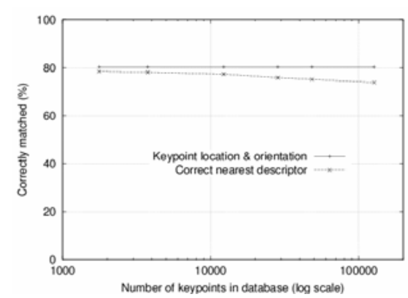
Feature stability to affine change

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



Distinctiveness of features

- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match



A good SIFT features tutorial

http://www.cs.toronto.edu/~jepson/csc2503/tutSIFT04.pdf

By Estrada, Jepson, and Fleet.