Local Features

Matching points across images important for:
- object identification (instance recognition)
- object (class) recognition
- pose estimation
- stereo (3-d shape)
- motion estimate
- stitching together photographs into a mosaic etc

Today

Interesting points, correspondence.

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

Criterion function:
### Sum of Squared (Pixel) Differences

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

We define the window function:

$$W(u,v) = \{u,v | x - \frac{m}{2} \leq x \leq x + \frac{m}{2}, \ y - \frac{m}{2} \leq y \leq y + \frac{m}{2} \}$$

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

$$C(u,v,d) = \sum_{(x,y) \in W(u,v)} (I_L(x,y) - I_R(x,y-d))^2$$

### 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:

  $$I = \frac{1}{W(u,v)} \sum_{(x,y) \in W(u,v)} I(u,v)$$  
  
  $$|I|_{W(u,v)} = \sqrt{\sum_{(x,y) \in W(u,v)} I(u,v)^2}$$  
  
  $$\hat{I}(x,y) = I(x,y) - \bar{I}$$  
  
### Images as Vectors

Each window is a vector in an $m^2$ dimensional vector space. Normalization makes them unit length.

### Image windows as vectors

“Unwrap” image to form vector, using raster scan order.

### Possible metrics

**Distance?**

$$w_K(d)$$

**Angle?**

$$w_L$$

$$w_R$$

### Image Metrics

- (Normalized) Sum of Squared Differences

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

  $$= \|w_L - w_K(d)\|$$

- Normalized Correlation

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

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

  $$d' = \arg \min \|w_L - w_K(d)\| = \arg \max \ w_L \cdot w_K(d)$$
Local Features

Not all points are equally good for matching…
(Review) Differential approach: Optical flow constraint equation

Brightness should stay constant as you track motion

1st order Taylor series, valid for small \( \delta \)

\[
I(x, y, t) + u \delta t_x + v \delta t_y + \delta t_z = I(x, y, t)
\]

Constraint equation

\[
ul_x + vl_y + l_z = 0
\]

“BCCE” - Brightness Change Constraint Equation

Combining Local Constraints

\[
\nabla I \cdot U = -l_z
\]

\[
\nabla I^2 \cdot U = -l_z
\]

\[
\nabla I^3 \cdot U = -l_z
\]

etc.
Lucas-Kanade: Integrate gradients over a 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:

\[
\begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2 \\
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
\end{bmatrix} = \begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t \\
\sum I_{xy} \\
\sum I_{xy} \\
\end{bmatrix}
\]

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

\[ \sum \nabla^2 I = -\sum \nabla I_t \]

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

Good Features to Track

\[
\begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2 \\
\sum I_{xy} & \sum I_{xy} \\
\end{bmatrix}
\begin{bmatrix}
u \\
u \\
\end{bmatrix} = \begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t \\
\sum I_{xy} \\
\sum I_{xy} \\
\end{bmatrix}
\]

When is This Solvable?

- A should be invertible
- A should not be too small due to noise
  - eigenvalues \( \lambda_1 \) and \( \lambda_2 \) of A should not be too small
- A should be well-conditioned
  - \( \lambda_1/\lambda_2 \) should not be too large (\( \lambda_1 = \) larger eigenvalue)

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

Harris detector

Auto-correlation matrix

\[
\begin{bmatrix}
\sum (I(x_1, y_1))^2 & \sum I(x_1, y_1) I(x_2, y_2) \\
\sum I(x_2, y_2) I(x_1, y_1) & \sum I(x_2, y_2)^2 \\
\end{bmatrix}
\]

- 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

large \( \lambda_1 \), small \( \lambda_2 \)

small \( \lambda_1 \), small \( \lambda_2 \)

Today

Interesting points, correspondence.

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

Invariant Local Features

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

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:
  \[ D(x) = D + \frac{\partial^2 D}{\partial x^2} \Delta x + \frac{\partial^2 D}{\partial y^2} \Delta y + \frac{\partial^2 D}{\partial x \partial y} \Delta x \Delta y \]
- Offset of extremum (use finite differences for derivatives):
  \[ \Delta x = -\frac{\partial^2 D}{\partial y^2} \Delta y \]
  \[ \Delta y = -\frac{\partial^2 D}{\partial x^2} \Delta x \]

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)

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
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

![Graph showing feature stability to noise](image)

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

![Graph showing feature stability to affine change](image)

Distinctiveness of features

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

![Graph showing distinctiveness of features](image)

A good SIFT features tutorial

By Estrada, Jepson, and Fleet.
An application of SIFT features in my own research…

The couch potato project: Learning from looking at images.

Bill Freeman, MIT
Joint work with: Josef Sivic, Andrew Zisserman (Oxford);
Bryan Russell (MIT), Alyosha Efros (CMU).
December 18, 2004

What can you learn about object categories by simply looking at images?

Labelled training databases
Labelling object classes in images is tedious, and can introduce biases.

Overview of our Method

Extracting Words

- Find interest points using shape adapted (white) and maximally stable (yellow) regions
- Map ellipses to a circle
- Compute SIFT descriptor over circle
SIFT (scale invariant feature transforms)

David Lowe, IJCV 2004

Visual words

- Vector quantize SIFT descriptors to a vocabulary of 2237 “visual words”.
- Heuristic design of descriptors makes these words somewhat invariant to:
  - Lighting
  - 2-d Orientation
  - 3-d Viewpoint

Examples of visual words

More visual words

Polysemy—the same word with different meanings

Experiment E

Figure 3: Polysemy, Example of a single visual word corresponding to two different (but locally similar) parts on two different object categories. (a) Top row shows occurrences of this visual word on the motorbike category, bottom row on the airplane category. The parts tend to occur consistently on different categories, i.e. this visual word lies mostly on the motorbike saddle and the airplane wing. (b) Corresponding normalized frames. Note the similarity of the normalized frames.
Observation matrix – experiment E

<table>
<thead>
<tr>
<th>Visual word #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame #</td>
</tr>
</tbody>
</table>

13.8 % non-zero entries

Binarized observation matrix – experiment E

<table>
<thead>
<tr>
<th>Visual word #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame #</td>
</tr>
</tbody>
</table>

13.8 % non-zero entries

Example segmentations

<table>
<thead>
<tr>
<th>Categories</th>
<th>pLSA</th>
<th>LDA</th>
<th>Texture</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.2 ub</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1-3 ub</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>1-3</td>
<td>97</td>
<td>56</td>
</tr>
<tr>
<td>D</td>
<td>1-4</td>
<td>98</td>
<td>70</td>
</tr>
<tr>
<td>E</td>
<td>1-4 bg</td>
<td>78</td>
<td>931</td>
</tr>
<tr>
<td>F</td>
<td>1-5,7-8 bg</td>
<td>59</td>
<td>1515</td>
</tr>
</tbody>
</table>

Results: All Experiments

Example images

Original images

Segmentations

All detected visual words

000117 000306 001448 001567 010748 010758

Background I  Background II  Background III

Faces  Motorbikes  Airplanes  Cars
Faces
Motorbikes
Airplanes
Cars

Background I
Background II
Background III
Figure 11: Multiple objects in an image. (a) pLSA example: Two objects are present in this image: a motorbike (topic 1 - green) and a car (topic 6 - red). The learned mixture coefficients $P(v|d)$ are 0.41 (motorbike - green), 0.02 (bag 1 - magenta), 0.16 (face - yellow), 0.19 (bag 2 - cyan), 0.04 (bag 3 - blue), 0.14 (car - red), 0.02 (airplane - black). In total there are 74 elliptical regions in this image of which 95 (72 unique visual words) are shown (have $P(v|d) > 0.5$). (b) LDA example: Two objects are present in this image, a face (yellow) and a car (red). The learned mixture weights $\phi$ are 0.19 (car (red)), 0.07 motorbike (green), 0.16 airplane (black), 0.14 background (blue), 0.44 face (yellow).