Correspondence
Matching feature points

We know how to detect good points
Next question: **How to match them?**

Two interrelated questions:
1. How do we *describe* each feature point?
2. How do we *match* descriptions?
Feature descriptor

$x_1$, $x_2$, $y_1$, $y_2$
Feature matching

- Measure the distance between (or similarity between) every pair of descriptors

<table>
<thead>
<tr>
<th></th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$d(x_1, y_1)$</td>
<td>$d(x_1, y_2)$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$d(x_2, y_1)$</td>
<td>$d(x_2, y_2)$</td>
</tr>
</tbody>
</table>
Better representation than color: Edges
Towards a better feature descriptor

• Match *pattern of edges*
  • Edge orientation – clue to shape

• Be resilient to *small deformations*
  • Deformations might move pixels around, but slightly
  • Deformations might change edge orientations, but slightly
Invariance to deformation by quantization

Between 30 and 45
Invariance to deformation by quantization

\[ g(\theta) = \begin{cases} 
0 & \text{if } 0 < \theta < \frac{2\pi}{N} \\
1 & \text{if } \frac{2\pi}{N} < \theta < \frac{4\pi}{N} \\
2 & \text{if } \frac{4\pi}{N} < \theta < \frac{6\pi}{N} \\
N - 1 & \text{if } 2(N - 1)\frac{\pi}{N} \end{cases} \]
Spatial invariance by histograms

2 blue balls, one red box

![Diagram showing 2 blue balls and 1 red box with corresponding bar graph showing 2 for balls and 1 for boxes.](image-url)
Rotation Invariance by Orientation Normalization

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

[Lowe, SIFT, 1999]
The SIFT descriptor

SIFT – Lowe IJCV 2004
Feature matching

Given a feature in $I_1$, how to find the best match in $I_2$?

1. Define distance function that compares two descriptors
2. Test all the features in $I_2$, find the one with min distance
Feature distance

How to define the difference between two features $f_1, f_2$?

- Simple approach: $L_2$ distance, $||f_1 - f_2||$
- can give good scores to ambiguous (incorrect) matches
Feature distance

How to define the difference between two features $f_1, f_2$?

- Better approach: ratio distance = $||f_1 - f_2|| / ||f_1 - f'_2||$
  - $f_2$ is best SSD match to $f_1$ in $I_2$
  - $f'_2$ is 2nd best SSD match to $f_1$ in $I_2$
  - gives large values for ambiguous matches
Structure from motion

• Given a bunch of images
• Get correspondences
  • Run interest point detector
  • Get SIFT descriptors
  • Match to get correspondences
• Use correspondences for
  • Estimating F/E, R, t
  • Estimating 3D structure of the world
Dense correspondence

• What if we need depth for every pixel?
• Setup: assume rectified images
  • Correspondence only along scan-lines
  • Can be represented using disparity
Dense correspondence
Dense correspondence

• Goal: Assign disparity value to each pixel
• Problem: most pixels will be ambiguous
• Solution: propagate from unambiguous to ambiguous pixels
• Basic idea: nearby pixels likely to have same disparity (*smoothness*)
Dense correspondence

• Goal:
  • Assign disparity value to each pixel

• Basic idea:
  • Disparity image should be *smooth*

• Energy minimization
  • $\min E(d)$, where $d$ is disparity image
  • $E(d) = E_{data}(d) + E_{smoothness}(d)$

• $E_{data}(d)$: scores based on NCC (for example)
• $E_{smoothness}(d) = \sum_{i,j} \rho(d(i, j) - d(i, j + 1)) + \rho(d(i, j) - d(i + 1, j))$
Markov Random Fields

- Probabilistic model
- Undirected graphical model

\[ P(d) \propto e^{-E(d)} \]

- Undirected graph with nodes and edges
- Unary potential on nodes = data term
- Binary potential on edges = smoothness term

\[ E(d) = \sum_{(i,j) \in V} \phi_u(d(i,j)) + \sum_{((i,j),(k,l)) \in E} \phi_b(d(i,j), d(k,l)) \]
Optimizing MRFs

• NP-Hard
• Approximate solutions
  • Message passing
  • Graph cut-based solutions
Dense correspondence with MRFs
Dense correspondence

• Goal: Assign disparity value to each pixel
• Problem: most pixels will be ambiguous
• Solution: propagate from unambiguous to ambiguous pixels
• Basic idea: nearby pixels likely to have same disparity (*smoothness*)
Dense correspondence

• Obtain disparity through optimization

\[ d^* = \arg \min_d \sum_{(i,j)} E_{data}(d(i, j)) + \sum_{(i,j),(k,l) \in \mathcal{N}} E_{smooth}(d(i, j), d(k, l)) \]

Based on e.g. NCC distance

\[ (d(i,j) - d(k,l))^2 \]
Detour: Graphical models

• Probabilistic models with graphs
• Nodes are variables
• Edges determine dependency structure
• independent of given all

- I wake up
- Alarm rings
- It is 8 am
- Naughty roommate
- Roommate sets up prank
- I wake up
Markov Random Fields (MRFs)

- Probabilistic model
- Represented by graph
- Each node is random variable
- Edges represent dependence structure
  - independent of given all
Markov Random Fields (MRFs)

- Hammersley-clifford theorem

\[ P(X = x) \propto e^{-E(x)} \]

\[ E(x) = \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \psi_{ij}(x_i, x_j) \]

\[ x^* = \arg \max_x P(X = x) = \arg \min_x E(x) \]
Dense correspondence as MRFs

• Obtain disparity through optimization
• Random variable: disparity
• Find most likely disparity

\[ d^* = \arg \min_d \sum_{(i,j)} E_{data}(d(i, j)) + \sum_{(i,j),(k,l) \in \mathcal{N}} E_{smooth}(d(i, j), d(k, l)) \]

Based on e.g. NCC distance

\[ (d(i,j) - d(k,l))^2 \]
Aligning depth boundaries to image boundaries

• Some pairs more likely to have same disparity

• $w(i,j) \ (d(i,j) - d(k,l))^2$

• $w(i,j) = 0$ for edges

• *Conditional* Random Field (CRF)
Other applications of MRFs / CRFs

Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.
Liang-Chieh Chen*, George Papandreou*, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. In *ICLR*, 2015
Optimizing MRFs

• NP-Hard
• Approximate solutions
  • Message passing
  • Mean field-based inference
  • Graph cut-based solutions

Dense correspondence with MRFs
Optical flow
Optical flow due to camera motion

• Consider camera translating and rotating

\[ \mathbf{P} = (X, Y, Z)^T \]

\[ x = \frac{X}{Z} \quad y = \frac{Y}{Z} \]

\[ \dot{\mathbf{P}} = -\mathbf{t} - \omega \times \mathbf{P} \]
Optical flow due to camera motion

\[
\begin{bmatrix}
u \\
v
\end{bmatrix} = \begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \frac{1}{Z} \begin{bmatrix}
-1 & 0 & x \\
0 & -1 & y
\end{bmatrix} \begin{bmatrix}
t_x \\
t_y \\
t_z
\end{bmatrix} + \begin{bmatrix}
xy & -(1 + x^2) & y \\
1 + y^2 & -xy & -x
\end{bmatrix} \begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\]
Optical flow for moving scenes
Optical flow for moving scenes
Optical flow for moving scenes

• Optical flow helps *grouping*

• *Gestalt principle of common fate*
  • *Things that move together belong together*
Motion segmentation in humans

Motion segmentation in humans

Motion segmentation in humans

1. Collect videos
2. Segment using motion
3. Train ConvNet

Optical flow for moving scenes

• Motion is cue for recognition
  • Gestures, actions, ...

Optical flow for moving scenes

- Motion is cue for recognition
  - Gestures, actions, ...

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without optical flow</td>
<td>73.0%</td>
</tr>
<tr>
<td>With optical flow</td>
<td>88.0%</td>
</tr>
</tbody>
</table>

Estimating optical flow

• Yet another correspondence problem!
• But:
  • Bad: scene can move
  • Good: changes are usually small (classic optical flow problem: <1 pixel)
Optical flow constraint equation

• Image intensity *continuous* function of \( x, y, t \)
• In time \( dt \), pixel \((x,y,t)\) moves to \((x + u \, dt, y + v \, dt, t + dt)\)

\[
\min_{u,v}(I(x + u\Delta t, y + v\Delta t, t + \Delta t) - I(x, y, t))^2
\]

\[\equiv \min_{u,v}(I(x, y, t) + I_xu\Delta t + I_yv\Delta t + I_t\Delta t - I(x, y, t))^2\]

\[\equiv \min_{u,v}(I_xu\Delta t + I_yv\Delta t + I_t\Delta t)^2\]

\[
I_x u + I_y v + I_t = 0
\]

• Optical flow constraint equation: One equation, two variables
Lucas-Kanade

• Assume all pixels in patch have the same flow
• When will this produce a unique solution?

\[
\begin{pmatrix}
\nabla I(x_1, y_1)^T \\
\vdots \\
\nabla I(x_n, y_n)^T
\end{pmatrix}
\begin{pmatrix}
u \\
v
\end{pmatrix}
= -
\begin{pmatrix}
I_t(x_1, y_1) \\
\vdots \\
I_t(x_n, y_n)
\end{pmatrix}
\]
Aperture problem
Aperture problem
Lucas-Kanade

\[
\begin{pmatrix}
\nabla I(x_1, y_1)^T \\
\vdots \\
\nabla I(x_n, y_n)^T
\end{pmatrix}
\begin{pmatrix}
u \\
v
\end{pmatrix}
=
\begin{pmatrix}
I_t(x_1, y_1) \\
\vdots \\
I_t(x_n, y_n)
\end{pmatrix}
\]

• Equation of the form \(Ax = b\)
• Solve using Normal equations: \(x = (A^TA)^{-1}A^Tb\)
• Need \(A^TA\) to be invertible - corners!
Lucas-Kanade

• What if we consider the whole image as one patch?
  • Constant optical flow for the entire image?
• Better: what if we consider flow as a *parametric function* of pixel location?
  • e.g. affine
    \[
    \begin{bmatrix}
    u \\
    v
    \end{bmatrix} = Ax + b
    \]
  • More generally:
    \[
    \begin{bmatrix}
    u \\
    v
    \end{bmatrix} = f(x, \theta)
    \]
• “Motion models”
Lucas-Kanade

\[
\min_{\theta} \sum_x (I(x + f(x, \theta) dt, t + dt) - I(x, t))^2
\]

• Solve by iterating on \( \theta \)
• Newton iteration
• Can we remove the parametric assumption?

Horn-Schunk

\[ E(u, v) = E_{data}(u, v) + E_{smoothness}(u, v) \]

\[ E(u, v) = \int \int (I(x + u(x, y)\Delta t, y + v(x, y)\Delta t, t + \Delta t) - I(x, y, t))^2 \]
\[ + \alpha(\|\nabla u\|^2 + \|\nabla v\|^2) \, dx \, dy \]
Horn-Schunk

\[ E(u, v) = E_{\text{data}}(u, v) + E_{\text{smoothness}}(u, v) \]

\[ E(u, v) = \int \int (I_x u + I_y v + I_t)^2 + \alpha(\|\nabla u\|^2 + \|\nabla v\|^2) \, dx \, dy \]
Variational minimization

- u and v are functions
- Euler-lagrange equations
  - Similar to “gradient=0”

\[
\min_q \int L(t, q(t), \dot{q}(dt)) \, dt
\]

\[
\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = 0
\]
Variational minimization

\[
\min_q \int L(t, q(t), \dot{q}(t)) dt \\
\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = 0
\]

\[
\min_{u,v} \int \int f(x,y,u,v,u_x,u_y,v_x,v_y) dx dy \\
\frac{\partial f}{\partial u} - \frac{d}{dx} \frac{\partial f}{\partial u_x} - \frac{d}{dy} \frac{\partial f}{\partial u_y} = 0 \\
\frac{\partial f}{\partial v} - \frac{d}{dx} \frac{\partial f}{\partial v_x} - \frac{d}{dy} \frac{\partial f}{\partial v_y} = 0
\]
MPI-Sintel

• Open-source animated movie “Sintel”
• “Naturalistic” video
• Ground truth optical flow
• Large motions
• Complex scenes
MPI-Sintel results
Optical flow with large displacements

• Optical flow constraint equation assumes differential optical flow
• “Large displacement”?
• Key idea: reducing resolution reduces displacement
• Reduce resolution, then upsample?
  • will lose fine details
Optical flow with large displacements

• Key idea 2: Use upsampled flow as *initialization*
• *Changes to initialization will be infinitesimal*

Optical flow for large displacements

• Horn-schunk variants match using color - Bad!
• Use descriptor matching (e.g. SIFT) on sparse points
• Use smoothness to propagate

Large displacement optical flow (LDOF)
Coarse-to-fine processing

• A specific instance of a general idea

• Coarse scales:
  • Global / large structures
  • Long-range relationships
  • But: imprecise localization

• Fine scales:
  • Precise localization
  • But: aperture problem

• Idea: start from coarse scales, add fine scale detail
Coarse-to-fine processing