Given an image point in left image, what is the (corresponding) point in the right image, which is the projection of the same 3-D point.

Difficulties – ambiguities, large changes of appearance, due to change of viewpoint, non-uniqueness.

Lambertian assumption

Rigid body motion

Correspondence

Matching - Correspondence

Local Deformation Models

- Translational model
  \[ h(x) = x + a \]

- Affine model
  \[ h(x) = Ax + a \]

- Transformation of the intensity values taking into account occlusions and noise
  \[ I_1(x_1) = f (X, g) I_2(h(x_1)) + n(h(x_1)) \]

Feature Tracking and Optical Flow

- Translational model
  \[ I_1(x_1) = I_2(x_1 + \Delta x) \]

- Small baseline
  \[ I(x(t), t) = I(x(t) + ut, t + dt) \]

- RHS approximation by the first two terms of Taylor series
  \[ \nabla I(x(t), t) T u + I_t(x(t), t) = \mathbb{C} \]
Aperture Problem

• Normal flow

\[ u_n = \frac{\nabla I_x u}{\|\nabla I\|} = -I_t / \|\nabla I\| \cdot \nabla I \]

Given brightness constancy constraint at single point – all we can recover is normal flow

Optical Flow

• Integrate around over image patch

\[ E_0(u) = \sum_{W(x,y)} [\nabla I^T(x, y, t) u(x, y) + I_t(x, y, t)]^2 \]

• Solve

\[ \nabla E_0(u) = 2 \sum_{W(x,y)} \nabla I \nabla I^T u + I_t \]

Optical Flow, Feature Tracking

\[ u = -G^{-1} h \]

\[ G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \]

Conceptually:
- rank(G) = 0 blank wall problem
- rank(G) = 1 aperture problem
- rank(G) = 2 enough texture – good feature candidates

In reality: choice of threshold is involved

Point Feature Extraction

\[ G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \cdot \]

• Compute eigenvalues of G
• If smallest eigenvalue \( \sigma \) of G is bigger than \( \tau \) - mark pixel as candidate feature point
• Alternatively feature quality function (Harris Corner Detector)

\[ C(G) = \det(G) + k \cdot \text{trace}^2(G) \]

Harris Corner Detector - Example
Feature Selection

- Compute Image Gradient \( \nabla I = [I_x, I_y] \)
- Compute Feature Quality \( Q(x,y) \)
- Search for local maxima

\[ Q(x,y) = (I_x^2 + I_y^2) + k \sum |I_{xy}| \]

**Feature Tracking**

- Translational motion model
  \[ F(d) = \min_{\lambda} \sum_{x,y} [I(x+d) - I(x)]^2 \]
- Closed form solution
  \[ d = \frac{-\partial F}{\partial d} = \frac{\sum I_x I_{y,d} + \sum I_y I_{x,d}}{\sum I_x I_{x,d} + \sum I_y I_{y,d}} \]
  1. Build an image pyramid
  2. Start from coarsest level
  3. Estimate the displacement at the coarsest level
  4. Iterate until finest level

Coarse to fine feature tracking

1. Compute \( d_h = -C_h \)
2. Warp the window \( W_b(x) \) in the second image by \( x + d_h \)
3. Update the displacement \( d \leftarrow d + 2d_h \)
4. Go to finer level \( h \leftarrow h - 1 \)
5. At the finest level repeat for several iterations

Affine feature tracking

\[ E(A, d, \lambda, \theta, \delta g) = \sum_{x,y} \frac{(I(x+y) - (\lambda g(x+y) + \delta g))^2}{\sum I_x I_{x,d} + \sum I_y I_{y,d}} \]

Contrast change

\[ S = \begin{bmatrix} x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \\ x^2 & y^2 & x^2 & y^2 & 2xy & x & y & 1 \end{bmatrix} \]

\[ x = [a_{11}, a_{12}, a_{21}, a_{22}, d_x, d_y, \delta g] \]

\[ x = S^{-1} \xi \]

Structure and Motion Recovery from Video

1. Use multiple image stream to compute the information about camera motion and 3D structure of the scene
2. Tracking image features over time

Original sequence

Tracked Features
Structure and Motion Recovery from Video

Computed model
3D coordinates of the feature points

Original picture

Wide baseline matching

Point features detected by Harris Corner detector

Region based Similarity Metric

- Sum of squared differences
  \[ SSD(h) = \sum_{\bar{x} \in W(x)} \| I_1(\bar{x}) - I_2(h(\bar{x})) \|^2 \]
- Normalize cross-correlation
  \[ NCC(h) = \frac{\sum_{\bar{x} \in W(x)} (I_1(\bar{x}) - I_1) (I_2(h(\bar{x})) - I_2)}{\sqrt{\sum_{\bar{x} \in W(x)} (I_1(\bar{x}) - I_1)^2 \sum_{\bar{x} \in W(x)} (I_2(h(\bar{x})) - I_2)^2}} \]
- Sum of absolute differences
  \[ SAD(h) = \sum_{\bar{x} \in W(x)} | I_1(\bar{x}) - I_2(h(\bar{x})) | \]

NCC score for two widely separated views

NCC score