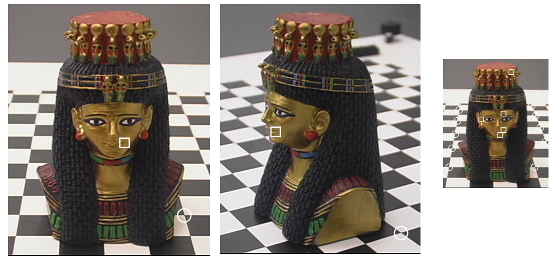
Image Primitives and Correspondence

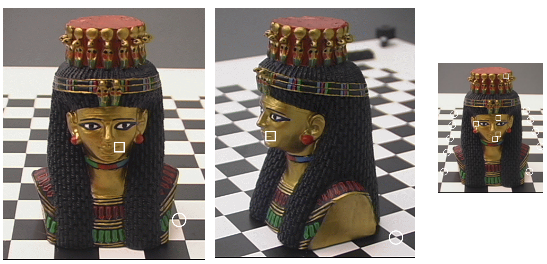
Jana Kosecka
George Mason University

Image Primitives and Correspondence



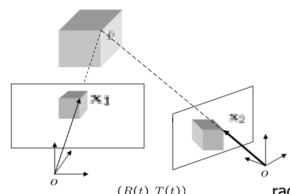
Given an image point in left image, what is the (corresponding) point in the right image, which is the projection of the same 3-D point

Image Primitives and Correspondence



Difficulties – ambiguities, large changes of appearance, due to change Of viewpoint, non-uniquess

Matching- Correspondence



radiance

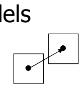
Lambertian assumption $I_1(\mathrm{x}_1) = \mathcal{R}(p) = I_2(\mathrm{x}_2)$

Rigid body motion $\mathrm{x}_2 = h(\mathrm{x}_1) = \frac{1}{\lambda_2(\mathrm{X})}(R\lambda_1(\mathrm{X})\mathrm{x}_1 + T)$

Correspondence $I_1(\mathrm{x}_1) = I_2(h(\mathrm{x}_1))$

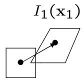
Local Deformation Models

- Translational model



$h(\mathrm{x}) = \mathrm{x} + d$ $I_1(\mathrm{x}_1) = I_2(h(\mathrm{x}_1))$

- Affine model



$h(\mathrm{x}) = A\mathrm{x} + d$ $I_1(\mathrm{x}_1) = I_2(h(\mathrm{x}_1))$

- Transformation of the intensity values taking into account occlusions and noise

$$I_1(\mathrm{x}_1) = f_o(\mathrm{X}, g) I_2(h(\mathrm{x}_1) + n(h(\mathrm{x}_1)))$$

Feature Tracking and Optical Flow

- Translational model

$$I_1(\mathrm{x}_1) = I_2(\mathrm{x}_1 + \Delta\mathrm{x})$$

- Small baseline

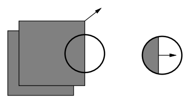
$$I(\mathrm{x}(t), t) = I(\mathrm{x}(t) + \mathbf{u}dt, t + dt)$$

- RHS approximation by the first two terms of Taylor series

$$\nabla I(\mathrm{x}(t), t)^T \mathbf{u} + I_t(\mathrm{x}(t), t) = 0$$

- Brightness constancy constraint

Aperture Problem



• Normal flow

$$\mathbf{u}_n \doteq \frac{\nabla I^T \mathbf{u}}{\|\nabla I\|} \cdot \frac{\nabla I}{\|\nabla I\|} = -\frac{I_t}{\|\nabla I\|} \cdot \frac{\nabla I}{\|\nabla I\|}$$

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

Optical Flow

• Integrate around over image patch

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

• Solve

$$\nabla E_b(\mathbf{u}) = 2 \sum_{W(x,y)} \nabla I(\nabla I^T \mathbf{u} + I_t)$$
$$= 2 \sum_{W(x,y)} \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \right)$$

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} = 0$$

$$G\mathbf{u} + \mathbf{b} = 0$$

Optical Flow, Feature Tracking

$$\mathbf{u} = -G^{-1}\mathbf{b}$$

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

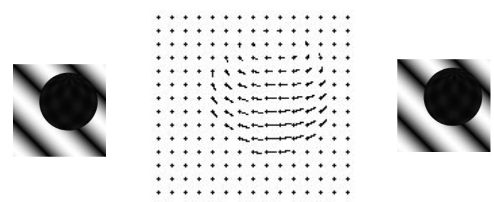
Conceptually:

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

Optical Flow

• Previous method - assumption locally constant flow



• Alternative regularization techniques (locally smooth flow fields, integration along contours)
• Qualitative properties of the motion fields

Point Feature Extraction

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

• Compute eigenvalues of G
• If smalest eigenvalue σ of G is bigger than τ - mark pixel as candidate feature point

• Alternatively feature quality function (Harris Corner Detector)

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

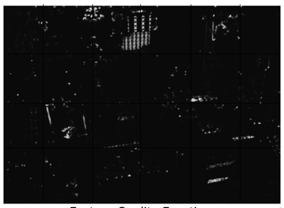
Harris Corner Detector- Example

Feature Selection

- Compute Image Gradient $\nabla I^T = [I_x, I_y]$
- Compute Feature Quality $C(\mathbf{x})$ measure for each pixel

$$C(\mathbf{x}) = det(G) + k.\text{trace}^2(G) \qquad G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- Search for local maxima



Feature Quality Function

Local maxima of feature quality function

Feature Tracking

- Translational motion model

$$E(\mathbf{d}) = \min_{\mathbf{d}} \sum_{W(\mathbf{x})} [I_2(\tilde{\mathbf{x}} + \mathbf{d}) - I_1(\tilde{\mathbf{x}})]^2$$

- Closed form solution

$$\mathbf{d} = -G^{-1}\mathbf{b}$$

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

$$\mathbf{b} \doteq \begin{bmatrix} \sum_{W(\mathbf{x})} I_x I_t \\ \sum_{W(\mathbf{x})} I_y I_t \end{bmatrix}$$

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

2 1 0

1. compute $\mathbf{d}_k = -G\mathbf{b}$
2. warp the window $W(\mathbf{x})$ in the second image by $2\mathbf{d}_k$
3. update the displacement $\mathbf{d} \leftarrow \mathbf{d} + 2\mathbf{d}_k$
4. go to finer level $k \leftarrow k - 1$
5. At the finest level repeat for several iterations

Affine feature tracking

$$E(A, \mathbf{d}, \lambda_E, \delta_E) = \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} w(\tilde{\mathbf{x}})[I(\tilde{\mathbf{x}}, 0) - (\lambda_E I(\boxed{A\tilde{\mathbf{x}} + \mathbf{d}}, t) + \delta_E)]^2$$

Contrast change

Intensity offset

$$S = \begin{bmatrix} x^2 I_x^2 & xy I_x^2 & x^2 I_x I_y & xy I_x I_y & x I_x^2 & x I_x I_y & x I_x I & x I_x \\ xy I_x^2 & y^2 I_x^2 & xy I_x I_y & y^2 I_x I_y & y I_x^2 & y I_x I_y & y I_x I & y I_x \\ x^2 I_x I_y & xy I_x I_y & x^2 I_y^2 & xy I_y^2 & x I_x I_y & x I_y^2 & x I_y I & x I_y \\ xy I_x I_y & y^2 I_x I_y & xy I_y^2 & y^2 I_y^2 & y I_x I_y & y I_y^2 & y I_y I & y I_y \\ x I_x^2 & y I_x I_y & x I_x I_y & y I_x I_y & I_x^2 & I_x I_y & I_x I & I_x \\ x I_x I_y & y I_y^2 & x I_y^2 & y I_y^2 & I_x I_y & I_y^2 & I_y I & I_y \\ x I_x I & y I_x I & x I_y I & y I_y I & I_x I & I_y I & I^2 & I \\ x I_x & y I_x & x I_y & y I_y & I_x & I_y & I & 1 \end{bmatrix}$$

$$\mathbf{z} \doteq [a_{11}\ a_{12}\ a_{21}\ a_{22}\ d_x\ d_y\ \lambda\ \delta_E]^T$$

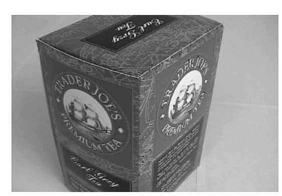
$$\mathbf{z} = S^{-1}\mathbf{c}$$

Tracked Features

Structure and Motion Recovery from Video

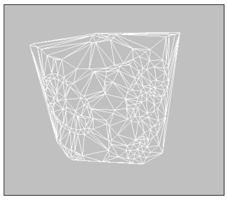
1. Use multiple image stream to compute the information about camera motion and 3D structure of the scene
2. Tracking image features over time

Original sequence

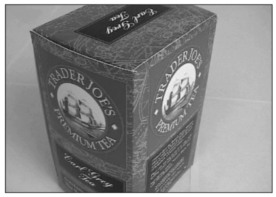


Tracked Features

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

$$SSD(h) = \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} \|I_1(\tilde{\mathbf{x}}) - I_2(h(\tilde{\mathbf{x}})\|^2$$

- Normalize cross-correlation

$$NCC(h) = \frac{\sum_{W(\mathbf{x})}(I_1(\tilde{\mathbf{x}}) - \bar{I}_1)(I_2(h(\tilde{\mathbf{x}})) - \bar{I}_2))}{\sqrt{\sum_{W(\mathbf{x})}(I_1(\tilde{\mathbf{x}}) - \bar{I}_1)^2 \sum_{W(\mathbf{x})}(I_2(h(\tilde{\mathbf{x}})) - \bar{I}_2)^2)}}$$

- Sum of absolute differences

$$SAD(h) = \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} |I_1(\tilde{\mathbf{x}}) - I_2(h(\tilde{\mathbf{x}}))|$$

NCC score for two widely separated views

NCC score