Motion
Introduction to Computer Vision
CSE 152
Lecture 18a

Continuous Motion
• Consider a video camera moving continuously along a trajectory (rotating & translating).
• How do points in the image move
• What does that tell us about the 3-D motion & scene structure?

Rigid Motion: General Case
Position and Orientation
Rotation Matrix & Translation vector
Rigid Motion:
Velocity Vector: T
Angular Velocity Vector: \( \omega \) (or \( \Omega \))

\[
\dot{p} = T + \omega \times p
\]

Motion Field Equation
\[
\begin{align*}
\dot{u} &= \frac{T_y - T_z f}{Z} + \frac{\partial y}{f} \frac{\partial v}{f} - \frac{\partial y^2}{f} \\
\dot{v} &= \frac{T_y - T_z f}{Z} + \frac{\partial y}{f} - \frac{\partial y^2}{f} \\
\end{align*}
\]
• T: Components of 3-D linear motion
• \( \omega \): Angular velocity vector
• (u,v): Image point coordinates
• Z: depth
• f: focal length

Forward Translation & Focus of Expansion
[Gibson, 1950]
Rotational MOTION FIELD

The “instantaneous” velocity of points in an image

Pure Rotation

\[ \omega = (0, 0, 1)^T \]

Optical Flow

Estimating the motion field from images

1. Feature-based (Sect. 8.4.2 of Trucco & Verri)
   1. Detect Features (corners) in an image
   2. Search for the same features nearby (Feature tracking).

2. Differential techniques (Sect. 8.4.1)

Definition of optical flow

OPTICAL FLOW = apparent motion of brightness patterns

Ideally, the optical flow is the projection of the three-dimensional velocity vectors on the image

Mathematical formulation

[Note change of notation: image coordinates now (x,y), not (u,v)]

\[ \text{where} \quad I(x, y, t) = \text{brightness at image point } (x, y) \text{ at time } t \]

Consider scene (or camera) to be moving, so x(t), y(t)

Brightness constancy assumption:

\[ \frac{dI}{dt} = 0 \]

Optical flow constraint equation:

\[ \frac{dI}{dt} = \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \]
Solving for flow

Optical flow constraint equation:
\[
\frac{dI}{dt} = \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0
\]

- We can measure \( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t} \)
- We want to solve for \( \frac{dx}{dt}, \frac{dy}{dt} \)
- One equation, two unknowns

What is the correspondence of P & P’

Contour plots of image intensity in two images

Normal Flow

Illusion Works Barber Pole Illusion

Two ways to get flow

1. Think globally, and regularize over image
2. Look over window and assume constant motion in the window
Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

\[ E(u, v) = \sum_{x,y} \left( I_u(x,y)u + I_v(x,y)v + I_{uv}(x,y)uv + I_x^2 + I_y^2 \right) \]

\[ \frac{dE(u, v)}{du} = \sum_{x,y} 2I_u(x,y) \left( u + I_v(x,y)v + I_{uv}(x,y)uv + I_x^2 + I_y^2 \right) = 0 \]

\[ \frac{dE(u, v)}{dv} = \sum_{x,y} 2I_v(x,y) \left( u + I_v(x,y)v + I_{uv}(x,y)uv + I_x^2 + I_y^2 \right) = 0 \]

Solve with:

\[ \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum I_x I_u \\ \sum I_y I_u \end{bmatrix} \]

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

\[ \left( \sum \nabla I \nabla I^T \right) \hat{U} = -\sum \nabla I \]

Lucas-Kanade: Singularities and the Aperture Problem

Let \( M = \sum \left( \nabla I \nabla I^T \right) \) and \( \hat{b} = \left[ \sum I_x I_x, \sum I_x I_y \right]^T \)

- Algorithm: At each pixel compute \( \hat{U} \) by solving \( MU = \hat{b} \)
  - \( M \) is singular if all gradient vectors point in the same direction
    - e.g., along an edge
    - of course, trivially singular if the summation is over a single pixel
    - i.e., only normal flow is available (aperture problem)
- Corners and textured areas are OK

Optical flow result

Recognition I

Introduction to Computer Vision
CSE 152
Lecture 18-b

Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.
Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

Object Recognition: The Problem

Given: A database $D$ of “known” objects and an image $I$:

1. Determine which (if any) objects in $D$ appear in $I$
2. Determine the pose (rotation and translation) of the object

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background – false positives

Object Recognition Issues:

- How general is the problem?
  - 2D vs. 3D
  - Range of viewing conditions
  - Available context
  - Segmentation cues
- What sort of data is best suited to the problem?
  - Whole images
  - Local 2D features (color, texture, 3D (range) data)
- What information do we have in the database?
  - Collection of images?
  - 3-D models?
  - Learned representation?
  - Learned classifiers?
- How many objects are involved?
  - Small: brute force search
  - Large: ??

A Rough Recognition Spectrum

Appearance-Based Vision: A Pattern Classification Viewpoint

1. Feature Space + Nearest Neighbor
2. Dimensionality Reduction
3. Bayesian Classification
4. Appearance Manifolds
Sketch of a Pattern Recognition Architecture

Example: Face Detection

Pattern Classification Summary

Image as a Feature Vector

Nearest Neighbor Classifier

Comments

Supervised vs. Unsupervised: Do we have labels?

- Supervised:
  - Nearest Neighbor
  - Bayesian
    - Plug in classifier
    - Distribution-based
    - Projection Methods (Fisher’s, LDA)
  - Neural Network
  - Support Vector Machine
  - Kernel methods

- Unsupervised:
  - Clustering
  - Reinforcement learning

Supervised

- Nearest Neighbor
- Bayesian

Plug in classifier
Distribution-based
Projection Methods (Fisher’s, LDA)

Neural Network
Support Vector Machine
Kernel methods

Unsupervised

- Clustering
- Reinforcement learning

Supervised

Nearest Neighbor
Bayesian

Plug in classifier
Distribution-based
Projection Methods (Fisher’s, LDA)

Neural Network
Support Vector Machine
Kernel methods

Unsupervised

- Clustering
- Reinforcement learning

Supervised

Nearest Neighbor
Bayesian

Plug in classifier
Distribution-based
Projection Methods (Fisher’s, LDA)

Neural Network
Support Vector Machine
Kernel methods

Unsupervised

- Clustering
- Reinforcement learning

Supervised

Nearest Neighbor
Bayesian

Plug in classifier
Distribution-based
Projection Methods (Fisher’s, LDA)

Neural Network
Support Vector Machine
Kernel methods

Unsupervised

- Clustering
- Reinforcement learning

Supervised

Nearest Neighbor
Bayesian

Plug in classifier
Distribution-based
Projection Methods (Fisher’s, LDA)

Neural Network
Support Vector Machine
Kernel methods

Unsupervised

- Clustering
- Reinforcement learning