Perception

July 21, 2005
CS 486/686
University of Waterloo

Outline

- Perception
  - Computational vision
- Reading: R&N Sect. 24.1-24.3

Perception

- Perception provides agents with information about the world
- Perception is initiated by sensors:
  - Microphones
  - Laser range finders
  - Sonars
  - Movement detectors
  - Video cameras (vision)
  - Etc.

Computational Vision

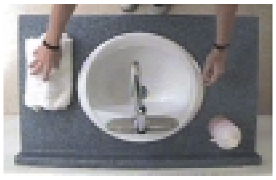
- Vision: very rich perception mode
- Computational vision:
  - Set of algorithmic/computational approaches to perceive the world from images
  - Focus on image analysis (more than image capture)
  - Inspired by the human vision system

Vision vs Graphics

- Image formation:
  - f(world) → image
  - Field of graphics
- Image analysis:
  - f^{-1}(image) → world
  - Field of computational vision

Image analysis

- Analysis of the information in a scene
- For instance, what are the
  - objects?
  - object properties?
  - object relations?



Video sequence analysis

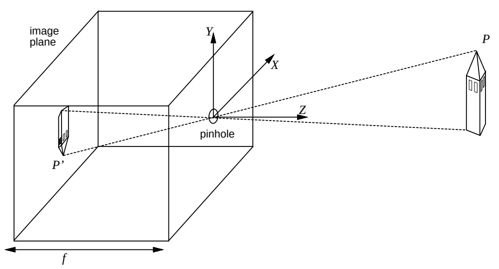
- Info analysis in a frame sequence

- For instance, what are the
  - events?
  - object movements?

Image Formation

- Vision gathers light scattered from objects in a scene and creates a two-dimensional image
- Image plane:
  - Coated with photosensitive material
- Digital image:
  - Rectangular grid of a few million pixels

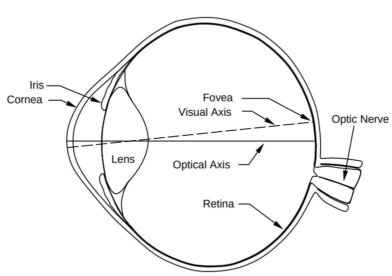
Pin-hole camera



Pin-hole camera

- $P=(X,Y,Z)$ in scene $\rightarrow$ $P'=(x',y',f)$ in image
- Distance equations:
  - $-x/f = X/Z \rightarrow x = -fX/Z$
  - $-y/f = Y/Z \rightarrow y = -fY/Z$
- Proportions are preserved
- Minus sign means that image is inverted

Human eye



Lens systems

- Pin-hole system is a rough approximation of lens system

- With lens
  - let in more light than a pin-hole
  - But image may be out of focus

Photometry

- Photometry is the study of light
- Light is crucial for vision
- Pixel brightness measures light intensity I(x,y)
- But light is emitted/reflected by many objects
  - This complicates image analysis!

Spectrophotometry

- Light comes in different colors
- Colors correspond to different wave lengths
- All colors perceived by human eyes correspond to linear combinations of
  - Red (700 nm)
  - Green (546 nm)
  - Blue (436 nm)
- Pixel often an RGB measurement

Model-based vision

- Since image = f(world),
  - Understand f
    - Photometry, spectrophotometry, physics, camera engineering, etc.
  - Compute world = f^{-1}(image)
    - But f is not completely understood
    - f is often noisy
    - So how can we compute f^{-1}?

Statistical vision

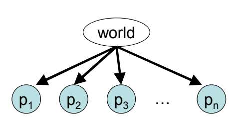
- To model uncertainty, let
  - f:world→image becomes P(image|world)
  - f^{-1}:image→world becomes P(world|image)
- Image analysis: compute most likely world for a given image
  - world* = argmax_{world} P(world|image)
    but where do we get P(world|image)?

Statistical Vision

- Could use Bayes Theorem:
  - world* = argmax_{world} P(world|image)
    = argmax_{world} P(image|world)P(world)
  - Called the generative approach
- Could use machine learning
  - Learn f^{-1} or P(world|image) from data
  - E.g. neural networks

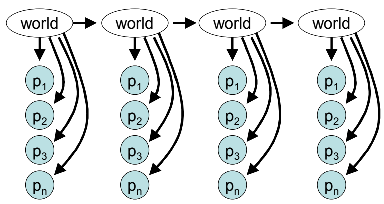
Generative Approach

- One random variable per pixel
- Assume pixels generated independently
  - P(image|world) = Π_i P(pixel_i|world)
  - Naïve Bayes model



Generative Approach

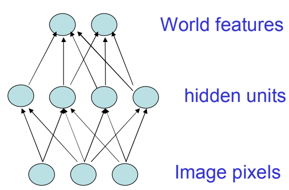
- For video sequences:
  - Hidden Markov model



Generative Approach

- It is a principled approach
- But often intractable…
  - "world" is too complex and must be decomposed
  - Too many pixels

Neural Networks



- Problems:
  - Too many pixels
  - Difficult to capture invariance

Image Processing

- Images contain millions of pixels
  - But many pixels may be uninformative
  - Extract relevant features
- Image processing
  - Extract all kinds of low level features
    - E.g., edges, corners
  - Then extract higher level information

Smoothing

- To handle pixel noise, smooth the image:
  - Pixel intensity ← (weighted) average of neighbors' intensity
- Gaussian filter
  - Assign weights proportional to Gaussian distribution
  - E.g. $I(x_0,y_0) = \Sigma_{x,y} I(x,y) G_\sigma(d)$
    - Where d is the distance between (x_0,y_0) and (x,y)
    - And $G_\sigma(d) = e^{-d^2/2s^2}$ / [sqrt(2π)σ²]

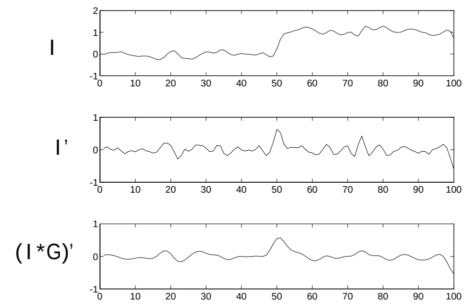
Convolution

- Convolution h = f*g
  - $h(x) = \Sigma_u f(u) g(x-u)$
- Hence, smoothing is a convolution of I with G_σ
  - i.e., $I*G_\sigma$

Edge Detection

- Edges:
  - Sharp change in intensity I
  - Idea: compute derivative of intensity I'
- Noise:
  - But noise could cause sharp intensity changes
  - Solution: smooth before edge detection
  - Hence compute $(I*G)'$

1D edge detection



Optimized edge detection

- Smoothed edge detection: $(I*G)'$
- Theorem: $(f*g)' = f*g'$
- Proof:
  - $(f*g)' = \partial[\Sigma_u f(u)\, g(x-u)] / \partial x$
    $= \Sigma_u f(u)\, \partial g(x-u)/\partial x$
    $= \Sigma_u f(u)\, g'(x-u)$
    $= f*g'$
- Hence $(I*G)' = I*G'$

2D Edge Detection

- Edges can have any orientation
- Can we avoid differentiating in all directions?
  - Yes: differentiate w.r.t x and y separately
- Compute:
  - $R_V(x,y) = I(x,y)*[G'_\sigma(x)G_\sigma(y)]$
  - $R_H(x,y) = I(x,y)*[G'_\sigma(y)G_\sigma(x)]$
  - $R(x,y) = R_H(x,y)^2 + R_V(x,y)^2$

Next Class

- Next Class:
  - Robotics
  - Russell and Norvig Ch. 25