CS664 Computer Vision

1. Introduction

Dan Huttenlocher

Computer Vision

- Machines that "see"
  - Broad field, any course will cover a subset of problems and techniques
- Closely related fields of study
  - Artificial intelligence and machine learning (CS)
  - Spatial statistics (Math/Stats)
  - Image processing (EE/CE)
  - Algorithms and optimization (CS/OR)
  - Optics and reflectance models (Physics)
  - Human visual perception and cognition (Psych)
  - Animal vision (Neuroscience)

Preparation

- Background in
  - Algorithms and data structures
  - Writing code with attention to efficiency
  - Basic probability
  - Linear algebra
- Some degree of mathematical sophistication
  - Ability to learn new math along the way without too much effort

Course Requirements

- Readings for class
- Short in-class quizzes
- Two programming assignments
  - Probably easiest in C/C++ because of libraries such as OpenCV
- Open ended final project on topic of your choice

Applications of Computer Vision

- Computer vision increasingly useful as digital images become ubiquitous
  - But still many simple-seeming problems still beyond state of art
- Wide range of areas
  - Image and video enhancement for consumer and entertainment applications
  - Automatic detection of faces and license plates for privacy, recognition for security
  - Automated inspection for industrial applications
  - Robotic automation and user assistance

Applications

- Industrial inspection
  - Wide range of industries from electronics to product labels to food
  - Match images to ideal prototypes
    - Highly controlled imaging conditions

Applications

- License plate reading
  - Roads and parking
    - E.g., garages, central London
  - License plate finding and character recognition
  - Structured location of plate

Applications

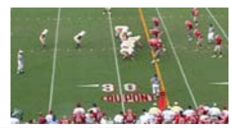
- Face detection and recognition
  - Finding faces in cluttered images
  - Recognizing faces (more controlled conditions)
  - Obscuring for privacy
    - E.g., Google's Streetview

Applications

- Image cleanup (inpainting)
  - Replace pixels by filling in surrounding image
  - Model as diffusion or spatial statistical process
  - Texture challenging

Applications

- Image compositing and synthesis
  - Image/video insertion
  - Panoramic images by stitching together multiple photos
  - Merging best of several photos together

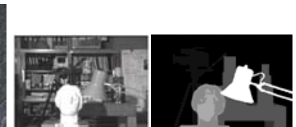


Applications

- Robotics
  - Ego-based and environment-based cameras
  - Integration of different sensing modalities
    - Lidar, radar, ir

Active Sensing Helps But…

- Lidar data provides cloud of points
  - Still “image-like” but with distances instead of intensities (or both)
- Still a vision problem
  - Analogous to working with stereo data



Applications

- Driving assistance (limited)
  - Monitor freeway lane change and forward vehicles

Visual Road Following

- Mobileye lane departure warning product

Computer Vision Algorithms

- Making things run fast an important part of practical computer vision techniques
  - Both algorithms and attention to coding details
- Dynamic programming (DP) common
  - Methods that cache solutions to sub-problems rather than re-computing them
- Applies to problems that can be decomposed into sequence of stages
  - Each stage expressed in terms of results of fixed number of previous stages

Basic Example

- Consider following problem
  - For every pixel in an mxn image, sum all the pixel values in a wxh window around the pixel

2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3

3x3 sum ⇒

4	7	9	8
5	9	12	11
5	9	12	11
4	7	9	8

  - Naïve method takes O(mnwh) time
    - 4 nested loops
  - Solve in O((m+w)(n+h)) time
    - Low constants, faster even for small windows

Simplify: What About 1D?

- For every element in an n-vector sum all elements in a width w interval

1	0	1	2	3	2	0	1	1
1	2	3	6	7	5	3	2	2
+0 -0	+1 -0	+2 -1	+3 -0	+2 -1	+0 -2	+1 -3	+1 -2	+0 -0

w=3

- Running sum – slide window of width w
  - Add entering element, subtract exiting one
  - Time independent of width w

2D Sums from 1D Sums

- Compute horizontal sums using sliding window
- On result compute vertical sums
  - This gives overall sum

2	1	2	3
1	0	1	2
1	0	1	2
2	1	2	3

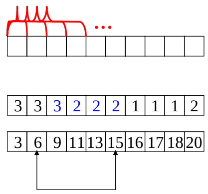
3	5	6	5
1	2	3	3
1	2	3	3
3	5	6	5

4	7	9	8
5	9	12	11
5	9	12	11
4	7	9	8

- Running time independent of w,h
  - Just add 2 and subtract 2 elements per pixel
- Extension to variable size windows

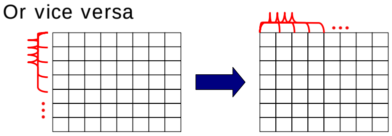
Using Integral Images

- Fast summations over arbitrary sized rectangles (intervals) – consider 1D
- Cumulative sum
  - S[x] = f[0] + … + f[x]
- DP recurrence O(n) time
  - S[x] = S[x-1] + f[x]
- Sum over region of f independent of size k
  - F[x]+…+F[x+k-1] = S[x+k-1]-S[x-1]



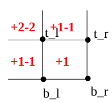
N-Dimensional Integral Images

- Analogous for higher dimensions, 2D:
  - S[x,y] = f[0,0] + … + f[0,y] + … f[x,0] + … + f[x,y]
- Separate recurrence per dimension
  - C[x,y] = C[x,y-1] + f[x,y] (column sum)
  - S[x,y] = S[x-1,y] + C[x,y] (total sum)
  - Or vice versa



Fast Sums Using Integral Images

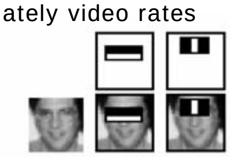
- Sum over arbitrary rectangle in constant time – with integral image preprocessing
  - S[b_r] + S[t_l-(1,1)] – S[b_l-(1,0)] – S[t_r-(0,1)]



- Also sum over arbitrary region, linear time
  - Running time proportional to length of boundary not area

Fast Detection With II

- Features formed from combinations of sums over rectangles
  - For example positive and negative regions
  - Running time independent of rectangle size
- Viola and Jones use for face detection at approximately video rates



Fast Object Detection

- Classifier (set of rectangles) learned from examples

Course Outline

- Filtering
- Edge detection
- Corner Detection
- Interest points – SIFT features
- 2D Geometry/Transforms
- Matching, Chamfer and Hausdorff
- Distance Transforms
- 3D camera geometry
- Multiview geometry
- Image Panoramas
- Image stitching/mosaicing

Course Outline

- Visual motion/optical flow
- Parametric motion
- Structure from motion
- Stereo
- Markov Random Fields for stereo
- MRF Inference
- Image segmentation
- Face Recognition, Subspace Methods
- Object Category Recognition
- Flexible template models
- Tracking by Matching