Outline

- Main Goal
- The Basics
- Preprocessing
- Bag of Features
- Features
- Resources
Main Goal

- Object detection / recognition
- Other relevant areas:
  - 3D reconstruction from stereo / motion
  - Object category recognition
  - Image processing
  - ...

What a Computer Sees

What you see:  What a computer sees:
The Basics

- Basic image operations
  - Blur
  - Gradient / Laplacian
  - Edge Detection
  - Scale Space
  - Blob Detection
The Basics – Blurring

- Gaussian
  
  \[ G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \]

- Convolve the Image with a Gaussian Kernel
  
  \[ I_{blurred} = G \ast I \]

- Convolution: For each pixel x,y:

  \[ \sum_{x, y} I_{\text{window}(x, y)} \cdot \text{Kernel} \]
Side Note - Anisotropic Diffusion

- Main idea: Don’t blur equally everywhere:
The Basics – Gradient / Laplacian

- Gradient = First derivative of an image
  \[ \Delta I = \sqrt{I_x^2 + I_y^2} \]

- Laplacian = Second derivative of an image
  \[ \nabla I = (I_{xx} + I_{yy}) \]
Derivatives of a Symmetric 2D Gaussian
Edge Detection

- Two main ideas:
  - Find places where the gradient is high:
    - Canny, others
  - Find places where the laplacian is 0:
    - Lindeberg, others
The Basics – Scale Space

- Scale is important
- Often blur the images and use all of the resulting images

[Lindeberg, 94]
The Basics – Blob Detection

- Look for places where the first and second derivatives are 0

*Figure 1.12: Blob detection by detection of scale-space maxima of the normalized Laplacian operator: (a) Original image. (b) Circles representing the 250 scale-space maxima of $(\nabla_{\text{norm}} L)^2$ having the strongest normalized response. (c) Circles overlayed on image.*

[Lindeberg, 98]
Preprocessing

- Normalizing Contrast
Preprocessing

- Decorrelation Stretch
Bag of Features

- Popular technique for framing object detection / recognition problems

- Main idea:

  1. Compute Features
  2. Find Matching Features in Bag
  3. Classify
Bag of Features

Main simplification:
- Spatial relations between features are discarded

Questions:
- What are the features?
- How do we “match” features?
- How do we decide what we should do with these matches?
Features

- Still an open area of research
- Can be anything you can imagine:
  - Simplest – The pixel values
  - Simple – image patches, edges, corners
  - More complicated – SIFT, MSER, ...
- Invariance / Discriminative Power
Features - SIFT

- SIFT = Scale Invariant Feature Transform
- Detect “keypoints”
Features - SIFT

- Compute the descriptor:

Image gradients  ➔  Keypoint descriptor

[Lowe, 04]
Features - SIFT

- What we get:

[Image: Lowe, 04]
Features - MSER

- **MSER = Maximally Stable Extremal Regions**
- Pick regions which are stable when we threshold the image with different threshold values

[Matas, 02]
What now?

- Find “nearest” feature in the image?
- Compute distances to a set of features?
- Ultimately, we want reduced data so that we can pass it to a machine learning algorithm
  - Support Vector Machine (SVM)
  - Neural Network
  - Boosted Classifier
Capturing Natural Light

Debevec (USC)
What is this?
Can you tell now?
Problems / Rule of Thumb

- Occlusions
- Lighting variability
- Noise
- Limited computing power
- If it would be hard for a human, it will usually be hard for computer vision.
Resources

- Matlab Image Processing Toolbox
- Peter’s Functions for Computer Vision:
  - If Matlab doesn’t have it already, check here first.
    - www.csse.uwa.edu.au/~pk/research/matlabfns/
- Andrea Vedaldi’s Website:
  - vision.ucla.edu/~vedaldi/
- OpenCV
  - C / C++ / bindings to python