Introduction to Artificial Intelligence

Computer Vision

Janyl Jumadinova
October 5, 2016
Computer Vision

Make computers understand images and video
Computer Vision

Make computers understand images and video
Computer Vision

Make computers understand images and video
Computer Vision

Make computers understand images and video
What kind of scene?
Where are the cars?
How far is the building?
Why computer vision matters?

- Safety
- Health
- Security

- Comfort
- Fun
- Access
Applications of Computer Vision

"Face Recognition"   "Pose Estimation"   "Body Tracking"

"Speech Reading"   "Palm Recognition"   "Car Tracking"
Segmentation

- Compact representation for image data in terms of a set of components
Segmentation

- Compact representation for image data in terms of a set of components
- Components share “common” visual properties
Segmentation

- Compact representation for image data in terms of a set of components
- Components share “common” visual properties
- Properties can be defined at different level of abstractions
Segmentation

- **Tokens**
  - whatever we need to group (pixels, points, surface elements, etc.)
Segmentation

- **Tokens**
  - whatever we need to group (pixels, points, surface elements, etc.)

- **Bottom up segmentation**
  - tokens belong together because they are locally coherent
Segmentation

- **Tokens**
  - whatever we need to group (pixels, points, surface elements, etc.)

- **Bottom up segmentation**
  - tokens belong together because they are locally coherent

- **Top down segmentation**
  - tokens belong together because they lie on the same object
What is Segmentation?

Clustering image elements that “belong together”

- **Partitioning**
  - Divide into regions/sequences with coherent internal properties

- **Grouping**
  - Identify sets of coherent tokens in image
OpenCV

► An open source BSD licensed computer vision library
  - Patent-encumbered code isolated into “non-free” module
    (SIFT, SURF, some of the Face Detectors, etc.)
► Available on all major platforms
  - Android, iOS, Linux, Mac OS X, Windows
► Written primarily in C++
  - Bindings available for Python, Java, even MATLAB (in 3.0).
► Well documented at http://docs.opencv.org
► Source available at https://github.com/Itseez/opencv
OpenCV

Image Processing

Filters
Transformations
Edges, contours
Robust features
Segmentation

Video, Stereo, 3D

Calibration
Pose estimation
Optical Flow
Detection and recognition
Depth
OpenCV

1. Load an image from the disk, display it on our screen, and write it to file in a different format.
OpenCV

1. Load an image from the disk, display it on our screen, and write it to file in a different format
2. Access and manipulate pixels
OpenCV: Pixel

- **Grayscale**: each pixel has a value between 0 (black) and 255 (white)
  - values between 0 and 255 are varying shades of gray
OpenCV: Pixel

- **Grayscale:** each pixel has a value between 0 (black) and 255 (white)
  - values between 0 and 255 are varying shades of gray

- **Color:** pixels are normally represented in the RGB color space
  - one value for the Red component, one for Green, and one for Blue
  - each of the three colors is represented by an integer in the range 0 to 255
  - how “much” of the color there is
OpenCV: Coordinate System

- The point (0, 0) corresponds to the upper left corner of the image
- x value increases as we move to the right
- y value increases as we move down
OpenCV represents images as NumPy arrays (matrices).

- NumPy is a library for the Python programming language that provides support for large, multi-dimensional arrays.
- To access a pixel value, we need to supply the x and y coordinates of the pixel.
- OpenCV actually stores RGB values in the order of Blue, Green, and Red.