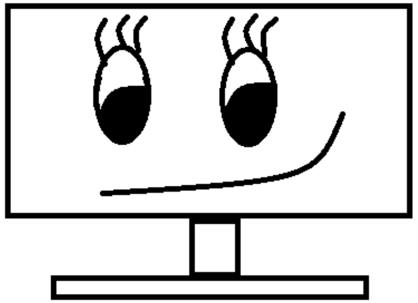
Seneca



CVI620/ DPS920
Introduction to Computer Vision

Introduction

Seneca College

Vida Movahedi

Overview

- Computer Vision
  - Definition
  - History
- A Computer Imaging System
  - An example
- Computer Vision problems and applications
- Available software and libraries
  - OpenCV

What is Computer Vision?

- “The science of creating a similar capability [as human vision] in computers and, if possible, to improve upon it” [2]
- “Computing properties of the 3-D world from one or more digital images” [3]
- “The transformation of data from a still or video camera into either a decision or a new representation” [1]
- See the Introduction slides from https://courses.cs.washington.edu/courses/cse576/20sp/calendar/

Related Fields

- Image Processing
  - Image properties
  - Image-to-image transformations, such as enhancement, compression, restoration
  - Usually needed as a pre-processing step of computer vision
- Pattern Recognition
  - Finding patterns, learning properties of objects, learn to detect or recognize
- Photogrammetry
  - Obtain reliable and accurate measurements from imaging
  - More precise in measurements than computer vision

Brief history [4]

- (1966) Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”!
- There are thousands of researchers working on this problem!

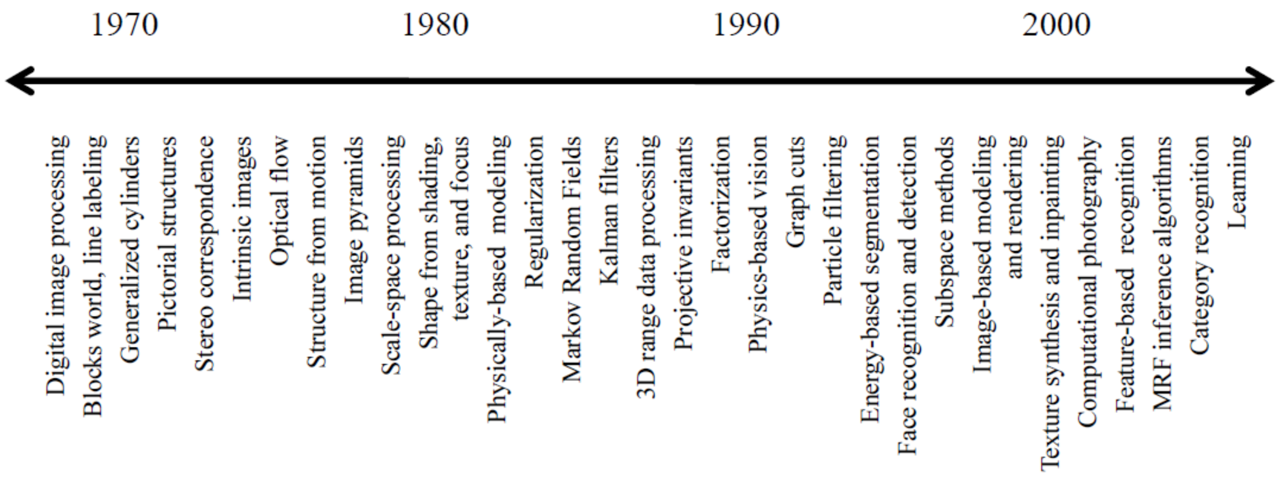
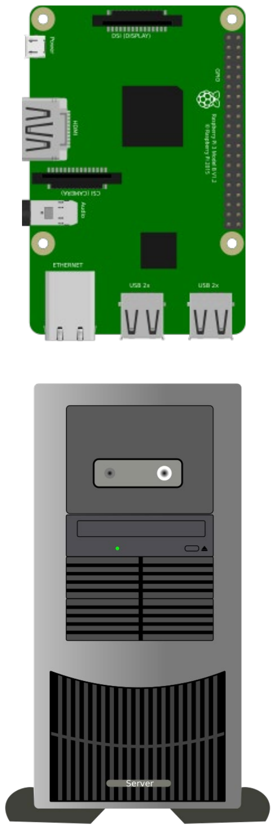


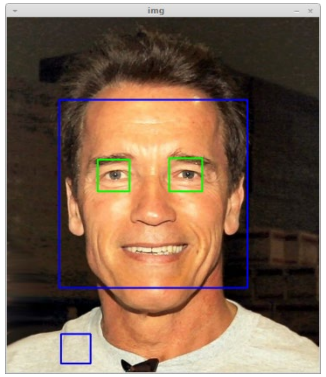
Figure 1.6 A rough timeline of some of the most active topics of research in computer vision.

A Computer Imaging System

An imaging or video recording
device or source

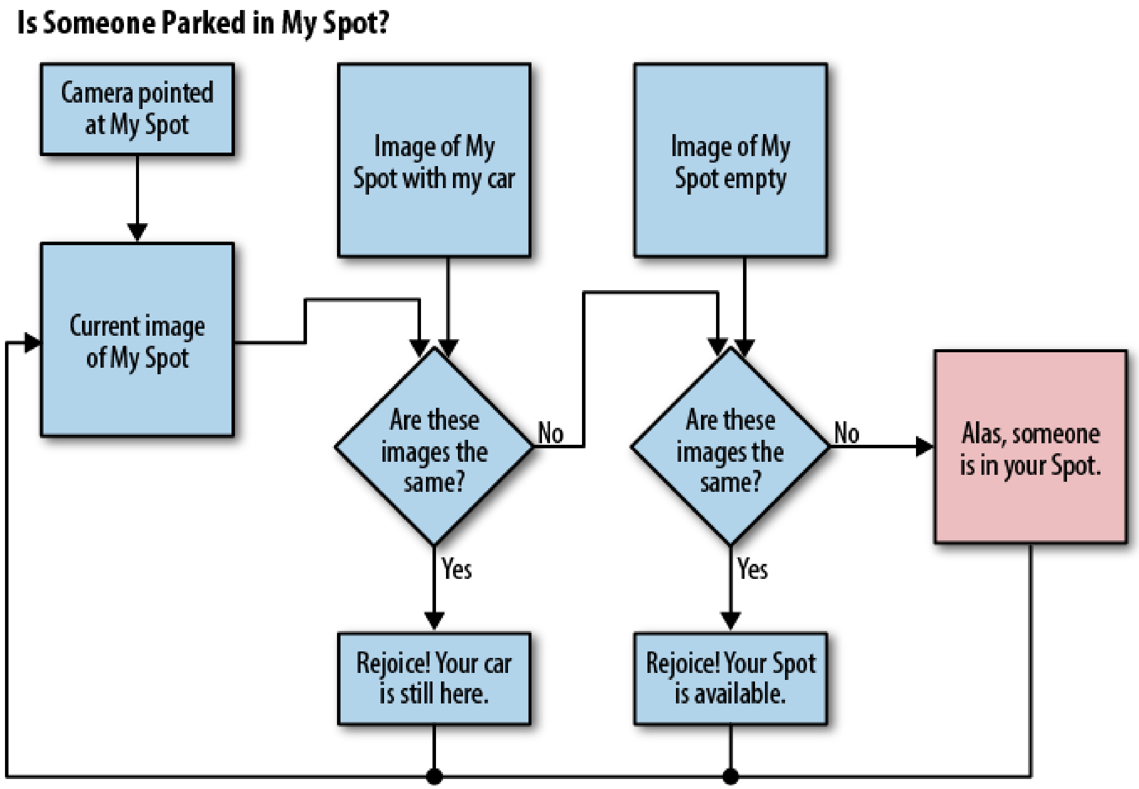


Computer processing or
algorithms



A new representation
or a decision

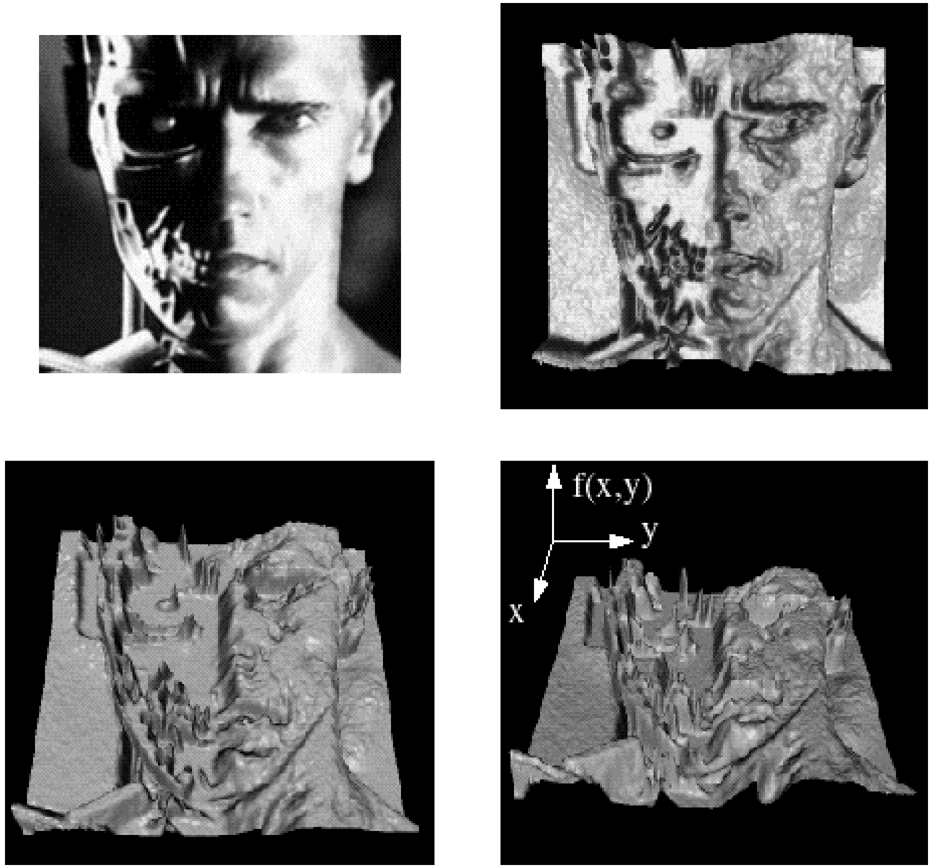
Example: Is someone parked in my spot? [2]


Is Someone Parked in My Spot?
Camera pointed at My Spot
Image of My Spot with my car
Image of My Spot empty
Current image of My Spot
Are these images the same?
No
Yes
Are these images the same?
No
Yes
Alas, someone is in your Spot.
Rejoice! Your car is still here.
Rejoice! Your Spot is available.

- Does this work? Is it easy to implement?

Why is it hard? [2]

- 3D to 2D
- **Sampled** on to a rectangular grid
- **Quantized** values for intensity (round to nearest integer)
- Hardware is not as sophisticated as a biological brain
- Not enough training
    - (a baby starts seeing the world at very early age)
- Algorithms?



Reference: http://courses.cs.washington.edu/courses/cse576

Easy vs. Hard Problems [2]

???	???
How wide is this plate? Is it dirty?	Look at a picture of a random kitchen, and find all the dirty plates.
Did something change between these two images?	Track an object or person moving through a crowded room of other people.
Measure the diameter of a wheel. Check to see if it is bent.	Identify arbitrary parts on pictures of bicycles.
What color is this leaf?	What kind of leaf is this?

Application Areas

- Industrial inspection and quality control
- Surveillance and security
- Face recognition
- Gesture recognition, fingerprint recognition
- Optical Character Recognition (OCR)
- Road monitoring, Driver monitoring, automotive safety
- Autonomous Vehicles (land, underwater, space)
- Space applications
- Military applications
- Retail (automated checkout)
- Medical imaging (MRI, CT, Ultrasound, etc.)
- Image databases, morphing, stitching
- Virtual reality, telepresence, telerobotic
- And many more!

Companies on Computer Vision

- David Lowe maintains an excellent overview of vision companies:
    - http://www.cs.ubc.ca/spider/lowe/vision.html

Software and libraries

- OpenCV → used in this course
- Matlab
  - Computer Vision Toolbox
  - Image Processing Toolbox
- SimpleCV (with Python)
- A list available here: https://www.cs.cmu.edu/~cil/v-source.html

OpenCV (http://opencv.org)

- Open source computer vision library
- Free to use personally or commercially
- Written in C and C++
- Runs under Linux, Windows, Mac OS X
- Active development on interfaces for Python, Java, MATLAB, Android, iOS for mobile applications
- Originated from Intel
- Widely used [1]

Sample Code

Read & display an image

```
# Import OpenCV
import cv2 as cv

# Read an image
img = cv.imread("corners.png")

# Create a window
cv.namedWindow("Image Window", cv.WINDOW_AUTOSIZE)

# Show the image in the above window
cv.imshow("Image Window", img)

# Wait for the user to press a key
cv.waitKey(0)

# Close the window
cv.destroyWindow("Image Window")
```

Using C

```
//Include file for every supported OpenCV function
#include <opencv2/opencv.hpp>
using namespace cv;

int main( int argc, char** argv ) {

   // read an image into an array of type cv::Mat
  Mat img = cv::imread( argv[1], -1 );

  if( img.empty() ) return -1;

  // create a window
  namedWindow( "Example 2-1", cv::WINDOW_AUTOSIZE );

  // show the image in the above window
  imshow( "Example 2-1", img );

  // wait for the user to press a key
  waitKey( 0 );

  // close the window
  destroyWindow( "Example 2-1" );

  return 0;
}
```

Sample Code

Connect a camera and display feed

```
import cv2 as cv

# Start a video capture, using device's camera
cap = cv.VideoCapture(0)

# Check if video file opened successfully
if (cap.isOpened() == False):
    print("Error opening video stream or file")

# Get and print out frame size
frame_width = int(cap.get(
                  cv.CAP_PROP_FRAME_WIDTH))
frame_height = int(cap.get(
                  cv.CAP_PROP_FRAME_HEIGHT))

print("Frame width: " , frame_width)
print("Frame height: " , frame.height)
```

```
# Read until video is completed
while(cap.isOpened()):
    # Capture frame-by-frame
    ret, frame = cap.read()
    if ret == False:
        break

    # Display the frame
    cv.imshow('frame',frame)
    key = cv.waitKey(33)

    # Press Q on keyboard to exit
    if key & 0xFF == ord('q'):
        break

# Release the video capture
cap.release()

# Close all the frames
cv.destroyAllWindows()
```

Summary

- Computer Vision is about computers seeing as we do!
- A computer Imaging system contains 3 components: An input (an image or video); a processing algorithm; and an output (a decision or presentation)
- Tasks that are easy for us may be very hard for a computer, and vice versa.
- Computer vision problems and applications are very broad.
- Opening an image or a video is easy using the OpenCV library.

References

[1] Learning OpenCV 3, A. Kaehler & G. Bradski

- Available online through Safari Books, Seneca libraries
- https://senecacollege-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01SENC_ALMA5153244920003226&context=L&vid=01SENC&search_scope=default_scope&tab=default_tab&lang=en_US

[2] Practical Computer Vision with SimpleCV, K. Demaagd et al.

- Available online through Safari Books, Seneca libraries
- https://senecacollege-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01SENC_ALMA5153198780003226&context=L&vid=01SENC&search_scope=default_scope&tab=default_tab&lang=en_US

[3] Introductory Techniques for 3-D Computer Vision, E. Trucco & A. Verri

[4] Computer Vision: Algorithms and Applications, R. Szeliski (http://szeliski.org/Book)