Introduction to Machine Learning

Author: Victor Carbune

Day 3 - July 18, 2014
ROSEdu Summer Workshops

Outline

Introduction to Computer Vision
- Common topics
- Mathematics: what and where?

Feature Extraction from Images
- Harris Corners
- SIFT Features

ML Models in Computer Vision
- What and where?

OpenCV
- Matching Feature Descriptors

Computer Vision intersects with:

- Mathematics
- Machine Learning
- Artificial Intelligence
- Algorithms
- Signal Processing
- Computer Graphics
- Robotics

Computer Vision vs. Computer Graphics

- **Graphics**
  Generate an image from scratch, using a known model

- **Vision**
  Understand contents of the image, build a scene model

Problems and Breakthroughs

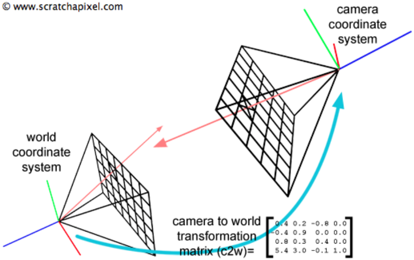
- Computer Vision is a huge field and our talk is short and introductory
- Let's go through the common topics and types of problems you might encounter, while studying the field

Image Formation and Camera Models

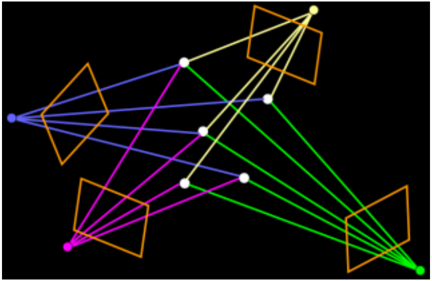
- First step: understand devices acquiring your data and how the image that you are processing is formed
- How? Compute the transformation matrices such that you can project the 3D points in world coordinates into the 2D image coordinates that you have

Image Formation and Camera Models

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$



- What can we do with multiple cameras?

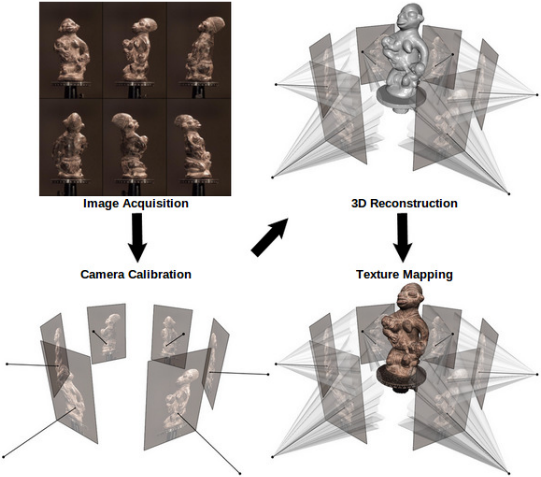

[1]

---

[1] http://vision.ucsd.edu/~manu/research.html

Multiview Geometry: 3D Reconstruction

- What can we do with multiple cameras?



[2]

2 http://carlos-hernandez.org/research.html

Feature Extraction

- Prerequisite: finding common points in series of images
- Solution: feature extraction and feature matching

- Breakthrough by David Lowe in ’99: SIFT features

- Developed in '88, by C. Harris and M. Stephens [3]

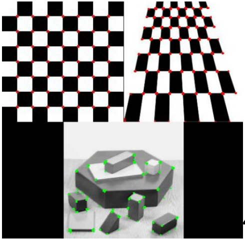
- Find difference in intensity, in all directions, for point (u, v)

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \, [\underbrace{I(x+u, y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}}]^2$$

[3] http://www.bmva.org/bmvc/1988/avc-88-023.pdf

Features: Harris Corners

- OpenCV already has it (see practice section)
- Some sample results you can get:

[4]

[4]image from OpenCV tutorial

Features: Scale Invariant Feature Transformation

- SIFT features are probably one of the most important breakthroughs in the field

- For any object, there are definitely many interesting points that one can consider unique. So which ones should we consider?
- SIFT features identifies in an image those feature vectors that are invariant to scaling, rotation or translation

Features: Scale Invariant Feature Transformation

- High-level overview of the stages [5]

[5] http://www.cs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf

Features: Scale Invariant Feature Transformation

- High-level overview of the stages [5]

- **Scale-space extrema detection**: searches over all scales and image locations and keep potential interest points, invariant to scale and orientation

[5] http://www.cs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf

- High-level overview of the stages [5]

- **Scale-space extrema detection**: searches over all scales and image locations and keep potential interest points, invariant to scale and orientation
- **Keypoint localization**: a model is fit to determine location and scale and stable keypoints are kept

[5] http://www.cs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf

- High-level overview of the stages [5]

- **Scale-space extrema detection**: searches over all scales and image locations and keep potential interest points, invariant to scale and orientation
- **Keypoint localization**: a model is fit to determine location and scale and stable keypoints are kept
- **Orientation assignment**: orientation is assigned to each keypoint, based on local image gradient direction

[5] http://www.cs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf

Features: Scale Invariant Feature Transformation

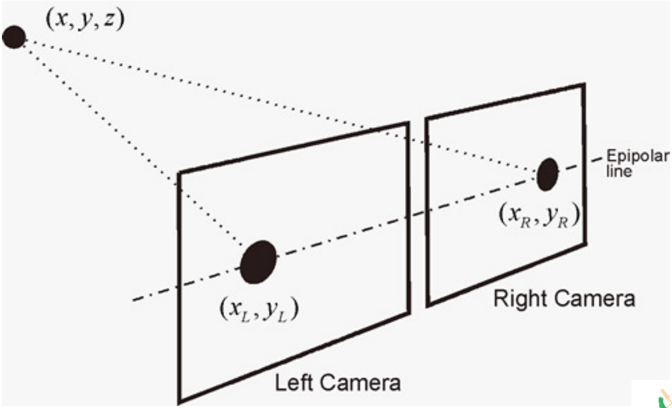
- High-level overview of the stages [5]

- **Scale-space extrema detection**: searches over all scales and image locations and keep potential interest points, invariant to scale and orientation
- **Keypoint localization**: a model is fit to determine location and scale and stable keypoints are kept
- **Orientation assignment**: orientation is assigned to each keypoint, based on local image gradient direction
- **Keypoint descriptor**: around each keypoint, local information is transformed into a representation tolerant to shape distortion or illumination change

[5] `http://www.cs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf`

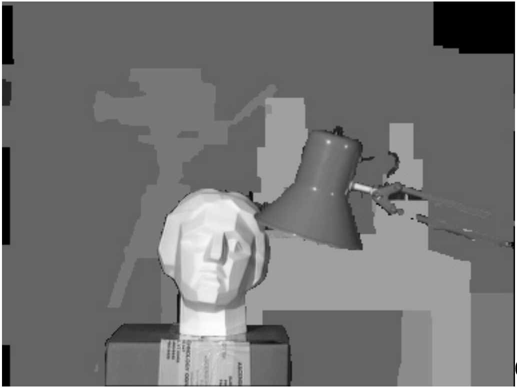
Features: Scale Invariant Feature Transformation

- What can we compute using this setup?



Stereo Matching: Disparity Map

- Disparity map: sensing depth



6

---

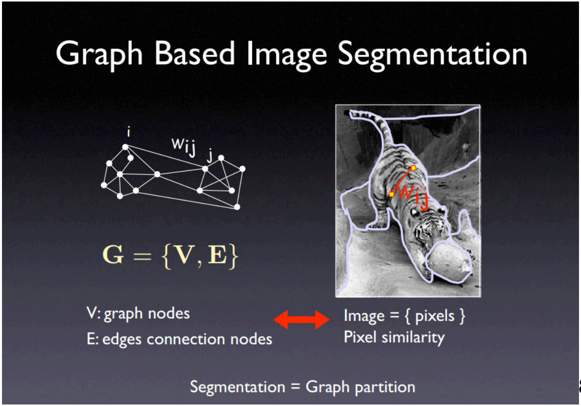
6 Seelinktotutorialinpracticesession

- Partition an image into multiple segments. Simplify the image and make it easier to analyze

[7]

[7] Jianbo Shi, `http://cis.upenn.edu/~jshi`

- Graph Cuts!



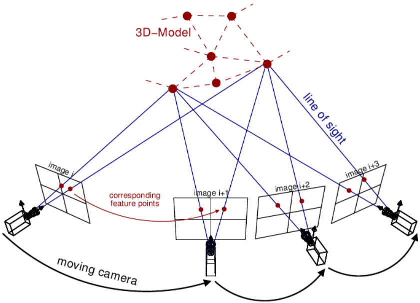
[8]

[8] Jianbo Shi, `http://cis.upenn.edu/~jshi`

Image Segmentation: Algorithms

- Mean Shift Filtering
- Clustering using Gaussian Mixture Models

- Estimates 3D structure from sequence of 2D images

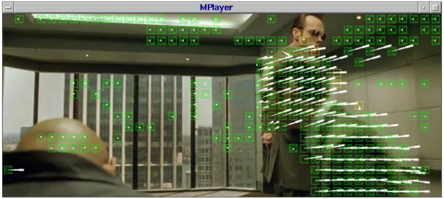


9

9 http://www.tnt.uni-hannover.de/

Reverse Part: Motion Estimation

- The reverse problem is an ill-posed problem, as it requires estimating 3D motion from 2D images



[10]

- You're using the technique every time you watch a movie, as video compression determines motion vectors for motion compensation

[10] http://xiph.org/~xiphmont/demo/theora/demo2.html

Computer Vision: Machine Learning?

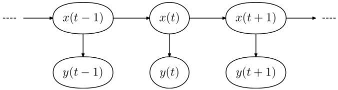
- Maybe not as much ML so far as you might have expected - you need some domain specific knowledge at first

Computer Vision: Machine Learning?

- Maybe not as much ML so far as you might have expected - you need some domain specific knowledge at first
- But there's plenty of it: object tracking, object recognition, hollistic scene understanding, image reconstruction and many others

Object Tracking: Particle Filters

- Hidden Markov Model

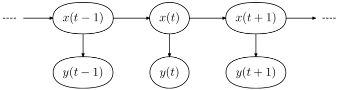


[11]

[11] http://en.wikipedia.org/wiki/Particle_filter

Object Tracking: Particle Filters

- Hidden Markov Model



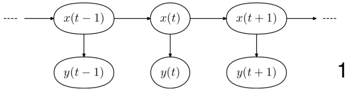
[11]

- State is described through a set of particles
- Alternating steps: prediction, update (measurement)

[11] http://en.wikipedia.org/wiki/Particle_filter

Object Tracking: Particle Filters

- Hidden Markov Model



[11]

- State is described through a set of particles
- Alternating steps: prediction, update (measurement)

- Several lines of code at most!
- Motivating example, for studying them in-depth: `https://www.youtube.com/watch?v=B4ianyQTnCE`

[11] `http://en.wikipedia.org/wiki/Particle_filter`

Object Recognition

- Step 1: is there a car in this image?

Object Recognition

- Step 1: is there a car in this image?
- Step 2: what kind of cars are in this image?

Object Recognition

- Step 1: is there a car in this image?
- Step 2: what kind of cars are in this image?
- Step 3: where are the cars in this image?

Object Recognition: Visual Words

- Idea: describe images through visual "words". How?

- Idea: describe images through visual "words". How?
- Step 1. Get a large "positive" training set of images containing the object you want to recognize

Object Recognition: Visual Words

- Idea: describe images through visual "words". How?
- Step 1. Get a large "positive" training set of images containing the object you want to recognize
- Step 2. Extract local features from each of the images (8 x 8 pixel patches).

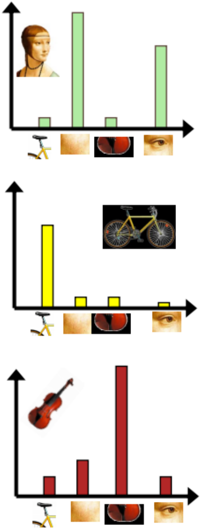
Object Recognition: Visual Words

- Idea: describe images through visual "words". How?
- Step 1. Get a large "positive" training set of images containing the object you want to recognize
- Step 2. Extract local features from each of the images (8 x 8 pixel patches).
- Step 3. Use a clustering technique (e.g. k-means) to identify the top-K most representative features. This is your "visual vocabulary". What distance can you use?

Object Recognition: Visual Words

- Idea: describe images through visual "words". How?
- Step 1. Get a large "positive" training set of images containing the object you want to recognize
- Step 2. Extract local features from each of the images (8 x 8 pixel patches).
- Step 3. Use a clustering technique (e.g. k-means) to identify the top-K most representative features. This is your "visual vocabulary". What distance can you use?
- Do the same for a "negative" training set

Object Recognition: Histograms by Visual Words



Object Recognition: Naive Bayes

- Now an image is represented by a set of words $(w_1, ..., w_n)$

Object Recognition: Naive Bayes

- Now an image is represented by a set of words $(w_1, ..., w_n)$
- Why not use Naive Bayes we discussed about yesterday?

$$P(y = (\text{has car})|w_1, ..., w_n) \propto$$
$$P(y = (\text{has car})) \prod_i P(w_i|y = (\text{has car}))$$

Object Recognition: Naive Bayes

- Now an image is represented by a set of words $(w_1, ..., w_n)$
- Why not use Naive Bayes we discussed about yesterday?

$$P(y = \text{(has car)}|w_1, ..., w_n) \propto$$
$$P(y = \text{(has car)}) \prod_i P(w_i|y = \text{(has car)})$$

- Nice read about visual words and their applications

http://luthuli.cs.uiuc.edu/~daf/courses/CS-498-DAF-PS/IRBG.pdf

Object Recognition: Naive Bayes

- Now an image is represented by a set of words $(w_1, ..., w_n)$
- Why not use Naive Bayes we discussed about yesterday?

$$P(y = \text{(has car)} | w_1, ..., w_n) \propto$$
$$P(y = \text{(has car)}) \prod_i P(w_i | y = \text{(has car)})$$

- Nice read about visual words and their applications

  http://luthuli.cs.uiuc.edu/~daf/courses/CS-498-DAF-PS/IRBG.pdf

- Solving remaining steps, is a fairly tough task

Object Recognition: State of the Art

- ImageNet Competition (yearly) [12]

[12] http://www.image-net.org/

Object Recognition: State of the Art

- ImageNet Competition (yearly) [12]

- 72%, 2010
- 74%, 2011
- **85%, 2012**

[12] http://www.image-net.org/

Object Recognition: State of the Art

- ImageNet Competition (yearly) [12]

- 72%, 2010
- 74%, 2011
- **85%, 2012**

- Guess the technology used in 2012!

[12] http://www.image-net.org/

Questions?

- Questions?

Questions?

- Questions?

- Next: let's play a bit with OpenCV in python

End of workshop!

- Hope you had fun and learned more about ML!
- Feel free to keep in touch, come back with questions, feedback, ideas or anything else. Thanks for attending!