Applications
2013-03-04

Anders Brun
Centre for Image Analysis
Swedish University of Agricultural Sciences
Uppsala University
Today's lecture

- Applications
  - Ida-Maria Sintorn (Vironova & SSBA)
  - Cris Luengo (Research Projects & the Image Analysis 2 Course)
  - [Anders Brun (Handwritten Text Recognition & PhD positions)]

- Opportunity to ask questions and put methods into context.
OCR – Optical Character Recognition

• First patents on OCR in 1929 and 1933
• First commercial computer 1951
• Template matching, Fourier descriptors, central moments, dynamic programming, thinning, contour following, ...
• Any method that is feasible for the hardware at the time being has been tested.
OCR, a solved problem?

Handwritten script

No

Typewriter

Yes

Printed latin script

Yes

Printed "fraktur"

Well…
Our HTR Project

• We will create a “Google” for large collections of handwritten text

• Human experts can only (accurately) transcribe and code a few pages of text per day!

• It is time for researchers in the humanities to use super computers too!
Preprocessing using Color

Original

Computed Mask

Final

Red

Green

Blue
Problems in C61
Revelations of Saint Bridget of Sweden

Damaged pages
Densely written text with slight overlaps
Slanting lines

Image from: Fredrik Wahlberg
Problem description: C64

Image from: Fredrik Wahlberg
Overview


Image from: Fredrik Wahlberg
Features

- Template matching in 2D is expensive

- 1D-fetures
  - Upper/lower contour
  - Projection

- Matching using “Dynamic Time Warping”
Experiments

Figure 7: Examples of hits for the word χρί (Christi). The template has been fitted to the text line by dynamic time warping. The last hit is semi-correct by having found the nominative Christus. (Section 3.3.4.)
Experiments

Figure 8: Example of occurrences of the word *och*. The template has been fitted to the text line by dynamic time warping. (Section 3.3.4.)
Experiments

Figure 9: Example of occurrences of the word corpus. The template has been fitted to the text line by dynamic time warping. (Section 3.3.4).
Programmig Languages

- Matlab + Mex
- Octave?
- C / C++
- Java
- OpenCL, CUDA
- Python + C-modules
Toolkits

- OpenCV (Open Source Computer Vision)
- ITK (Insight Segmentation and Registration Toolkit)
- ImageJ
- Intel Performance Primitives (IPP, IPPI)
- FFTW (“Fastest Fourier Transforms in the West.”)
- Camera Calibration Toolkit (Matlab)
- DIPLib (Matlab)
Lessons Learned

• Insist to be involved when the images are taken to avoid spending half the project repairing mistakes from the image acquisition. However, this is usually the situation when people call in image analysis experts – usually, it is a little bit too late.

• Add visual markers, prepare your samples with special dyes, add special lighting and use color or multispectral cameras to make the image analysis part as simple as possible. It is not cheating. Be smart and help your algorithms as much as possible.

• If a human expert cannot see the correct answer, it is generally a bad sign – you will have a tough time designing an algorithm to do the job.
Lessons Learned

• Representing the information in a suitable form is often crucial to solve a problem efficiently
  – encoding color using RGB vs HSV
  – contours as chain codes vs fourier coefficients
  – vector graphics vs bitmap graphics

• Given the right representation of data, your problem is typically *easy* to solve.
Lessons Learned

• Computerized Image Analysis is efficient when e.g.
  – You need to repeat the process many times
  – You need an answer fast (real-time)
  – When objectivity/accuracy is important
  – When the task is enormous (many images)
  – When a human is not there (a Mars rover)
  – Your problem is not 2-D (3D, 4D, 5D, ...)

• If you can completely solve a task using Photoshop (GIMP) and a few hours of manual work, then consider to “just do it”.

Things we have not talked about

- Learning (Neural nets, SVMs, Boosting)
- Statistical techniques (Bayesian networks, Markov random fields)
- Computer vision, stereo vision, 2.5D (Kinect)
- Higher-dimensional images (3D, 4D, 5D, …)
- Movies (2D + time, 3D + time)

- Some of this pops up in the Image Analysis 2 Course later this fall!
Suggested Exercises

- Check out old written exams
- Finish your labs
- Write your handwritten double sided A4 sheet for the exam

- And remember, you can always try things out in Matlab