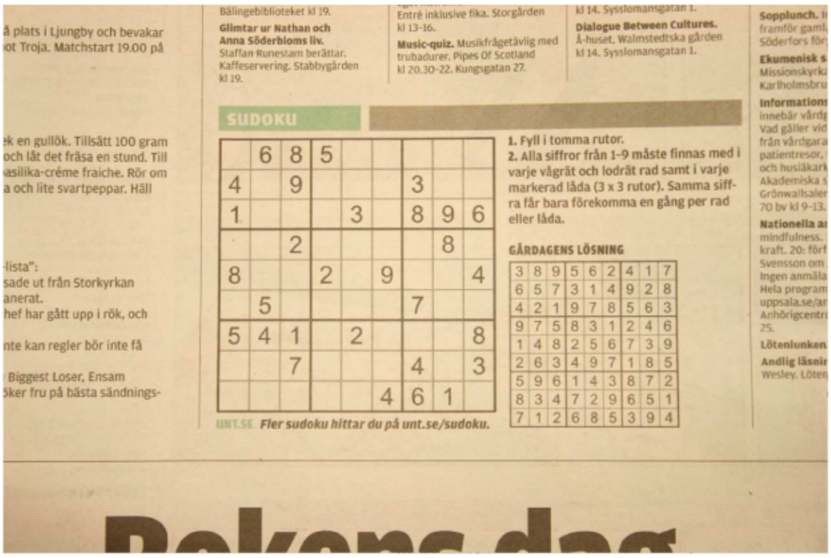
Project 1: Sudoku solver

Write a program that finds the sudoku square in the image, detects the 81 fields, and identifies the number in the fields that have a number. The output should be a 9x9 matrix with the given numbers and 0 values in empty fields. Optionally, your program can then solve the puzzle.

The file http://www.cb.uu.se/~cris/bild2projects/sudoku.zip contains 5 photographs of puzzles, taken from the newspaper. Some of them have numbers written in them with ballpoint. By distinguishing black from blue or red you can filter the image to ignore the ballpoint marks.

To recognize the numbers there are several approaches. Because these puzzles are all from the same newspaper, and photographed from the same distance, a simple correlation should work. Feel free to try other approaches.

The program should work, without modification, on all images given. If you prefer, you can scan a collection of puzzles yourself, using a flat-bed scanner.



å plats i Ljungby och bevakar
ot Troja. Matchstart 19.00 på

ek en gullök. Tillsätt 100 gram
och låt det fräsa en stund. Till
asilika-créme fraiche. Rör om
a och lite svartpeppar. Häll

-lista":
sade ut från Storkyrkan
anerat.
hef har gått upp i rök, och

nte kan regler bör inte få

Biggest Loser, Ensam
öker fru på bästa sändnings-

Bälingebiblioteket kl 19.

Glimtar ur Nathan och Anna Söderbloms liv. Staffan Runestam berättar. Kaffeservering. Stabbygården kl 19.

Entré inklusive fika. Storgården kl 13-16.

Music-quiz. Musikfrågetävlig med trubadurer. Pipes Of Scotland kl 20.30-22. Kungsgatan 27.

kl 14. Sysslomansgatan 1.

Dialogue Between Cultures. Å-huset. Walmstedtska gården kl 14. Sysslomansgatan 1.

SUDOKU

	6	8	5					
4		9				3		
1				3		8	9	6
		2					8	
8			2		9			4
	5					7		
5	4	1		2				8
		7				4		3
					4	6	1	

UNT.SE *Fler sudoku hittar du på unt.se/sudoku.*

1. Fyll i tomma rutor.
2. Alla siffror från 1-9 måste finnas med i varje vågrät och lodrät rad samt i varje markerad låda (3 x 3 rutor). Samma siffra får bara förekomma en gång per rad eller låda.

GÅRDAGENS LÖSNING

3	8	9	5	6	2	4	1	7
6	5	7	3	1	4	9	2	8
4	2	1	9	7	8	5	6	3
9	7	5	8	3	1	2	4	6
1	4	8	2	5	6	7	3	9
2	6	3	4	9	7	1	8	5
5	9	6	1	4	3	8	7	2
8	3	4	7	2	9	6	5	1
7	1	2	6	8	5	3	9	4

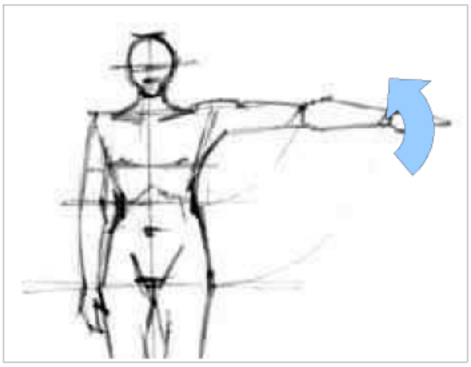
Sopplunch. I
framför gaml
Söderfors för
Ekumenisk s
Missionskyrk
Karlholmsbru
Information
innebär vård
Vad gäller vid
från vårdgara
patientresor,
och husläkark
Akademiska s
Grönwallsaler
70 bv kl 9-13.
Nationella a
mindfulness.
kraft. 20: förf
Svensson om
Ingen anmäla
Hela program
uppsala.se/ar
Anhörigcentr
25.
Lötenlunken
Andlig läsni
Wesley. Löten

Project 2: Hand gesture recognition

Write a program that recognizes 8 different hand gestures. You will need a web cam for this project.

Set up the web cam so that it sees your upper body. Your program should obtain a series of images from the web cam, and recognize the motion of one or both of your hands. This should be relatively easy if the only moving things the camera sees are your hands. Your program should not depend on a uniform background, but rather look for motion by comparing subsequent frames. Additional complexity can be added by allowing other moving things. For example, try to distinguish whether the moving object is a hand or a head.

Your program should be able to recognize 8 different commands. You can choose yourself what different motions can be considered commands. For example: left hand up, left hand down, right hand up, one hand up while the other goes down, moving hand left to right, etc. etc.



Project 3: There’s not yet a project 3

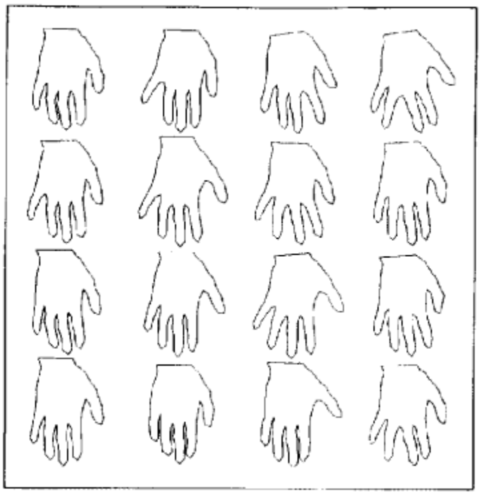
Project 4: Identifying hand pose using active shape models

In this project you will be experimenting with active shape models.

Download the active shape model toolbox at the File Exchange: http://www.mathworks.com/matlabcentral/fileexchange/26706 . Disregard the example, it does not correspond with this project.

First you will be taking a sequence of photographs of the same hand, open and palm down on the table. Use a uniform, dark background. Explore the range of motion of the fingers, such that each photograph captures the hand in a different pose. You are aiming at something similar to the drawing below, taken from the Cootes et al. paper that first introduced active shape models (Computer Vision and Image Understanding 61(1):38-59, 1995). Now train the model with this data. Examine the various parameters of the model: How does each affect the shape? Are they independent? Does the average shape represent a possible shape?

Next, take a new image, one that wasn't used in creating the model, and fit the model to it. Is the fit dependent on initialisation? Does the model generalize well? If the hand in the photograph is in a pose not earlier seen by the model, is the model able to adapt? What happens if you change the background to a different colour? And a less uniform background?

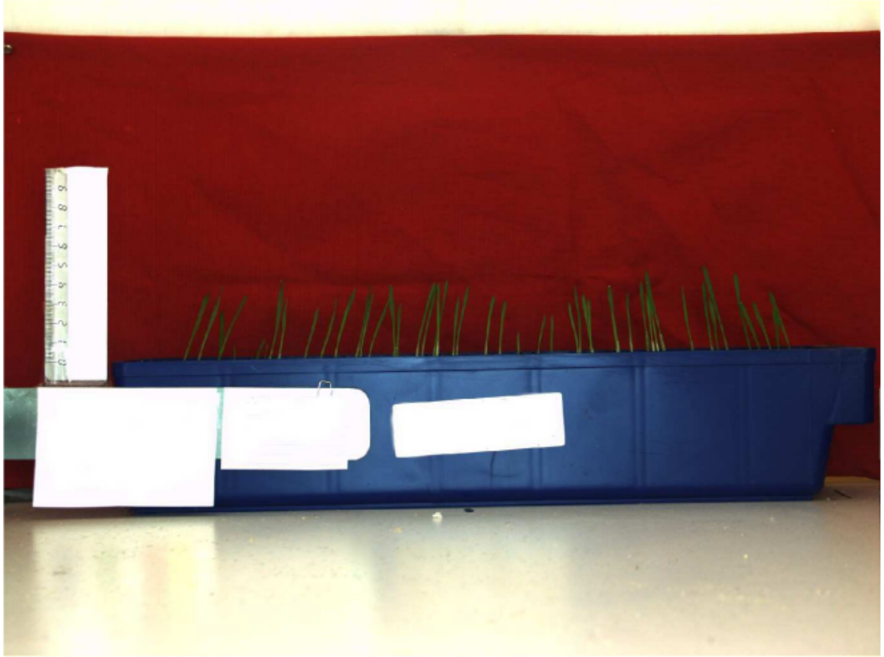


Project 5: Monitoring leave length

Write a program that detects the leaves and measures their length in the images given.

The file http://www.cb.uu.se/~cris/bild2projects/plants.zip contains 8 photographs of young plants, taken from the side. Each plant is composed of a single leaf. Note that the red background gives excellent contrast with the green leafs if you use the proper colour space.

You program should detect all the leafs in one image, measure their length, and return the data in a form you find suitable. Note that leafs can cross each other in the image. Try to design your program such that these can be separated. The program should work without modification on all the images provided.



Project 6: Detecting and measuring cells in a Pap smear

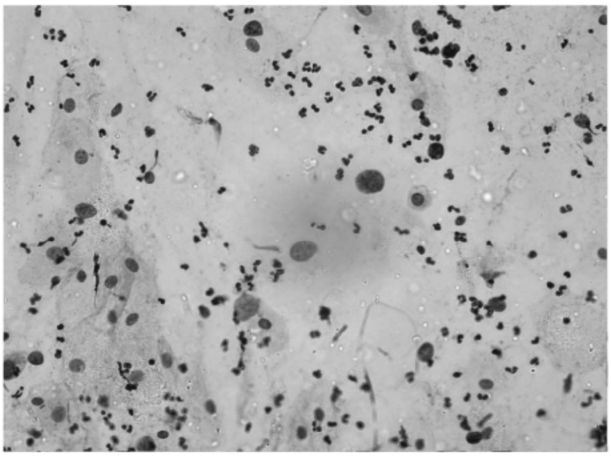
Automating the assessment of Pap smears has been attempted since the 1950’s. It is still not a fully solved problem. But in this project we won’t be as demanding as the real-world medical application.

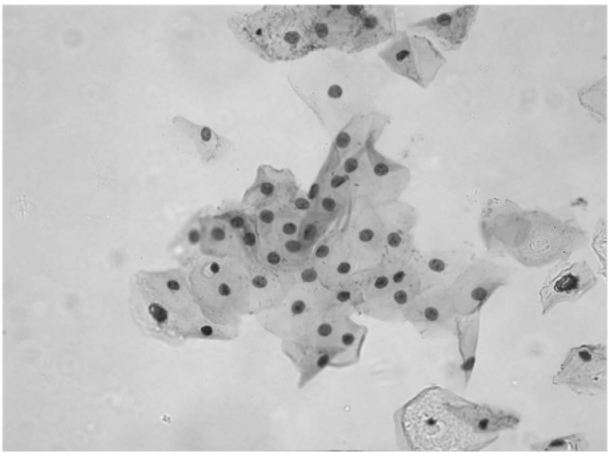
A Pap smear is a collection of cells from the cervix, and can be used for early detection of cervical cancer. The problem is that manual evaluation of a smear can take up to 20 minutes. A person needs to look at every single cell on the whole slide, looking for the few odd cells among 100 000 cells. No wonder there’s such a strong interest in automation!

Your task is to write a program that detects and delineates the cell nuclei. These are the larger, dark, round shapes seen in both images. Note that the smaller, darker shapes, which are very common in the left image, are blood cells and should be ignored.

The areas with intermediate grey value around the nuclei are the cell cytoplasms. Try to distinguish these from the unoccupied background also.

The images are here: http://www.cb.uu.se/~cris/bild2projects/papsmear.zip





Project 7: Non-local means

All the noise reduction filters we've seen in the course do something with the local neighbourhood of a pixel, to determine what the noise-free value of that pixel is supposed to be. Buades et al. described an alternative in Multiscale Modeling and Simulation 4(2):490-530 (2005). They propose to compare the local neighbourhood of a pixel with all possible patches in the image, pick similar patches, and average the value of the centre pixel for all those patches. This means, the output is not a weighted average of pixels in the neighbourhood, but it is a weighted average of similar pixels all over the image.

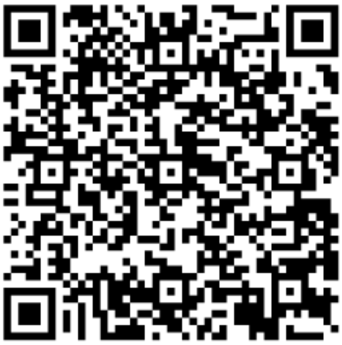
Get the paper here: http://www.cb.uu.se/~cris/bild2projects/NLMeans.pdf , and study section 5. Then implement the denoising algorithm described.

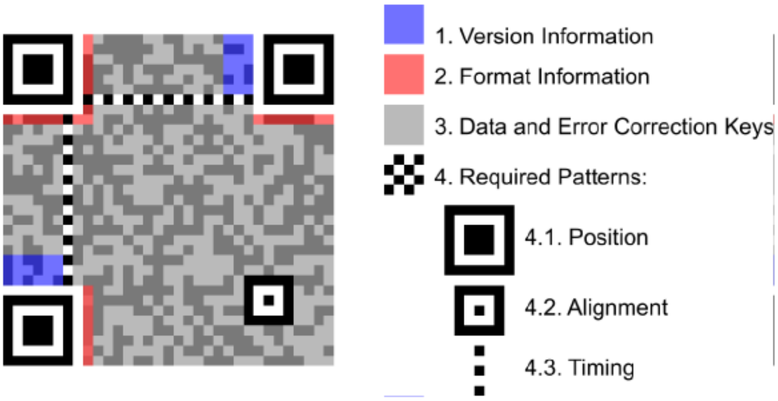
Project 8: 2D bar code reading

QR codes are increasingly common 2D bar codes. Most modern cell phones can take a photograph of such a bar code and give the user the information encoded in it, for example a web URL, an email address, a vcard, etc.

Your task is to write a program that detects the QR code in an image, corrects it for rotation and perspective, finds the top-left corner and the grid spacing, and reads the code as a matrix of ones and zeros. Decoding the QR code is optional.

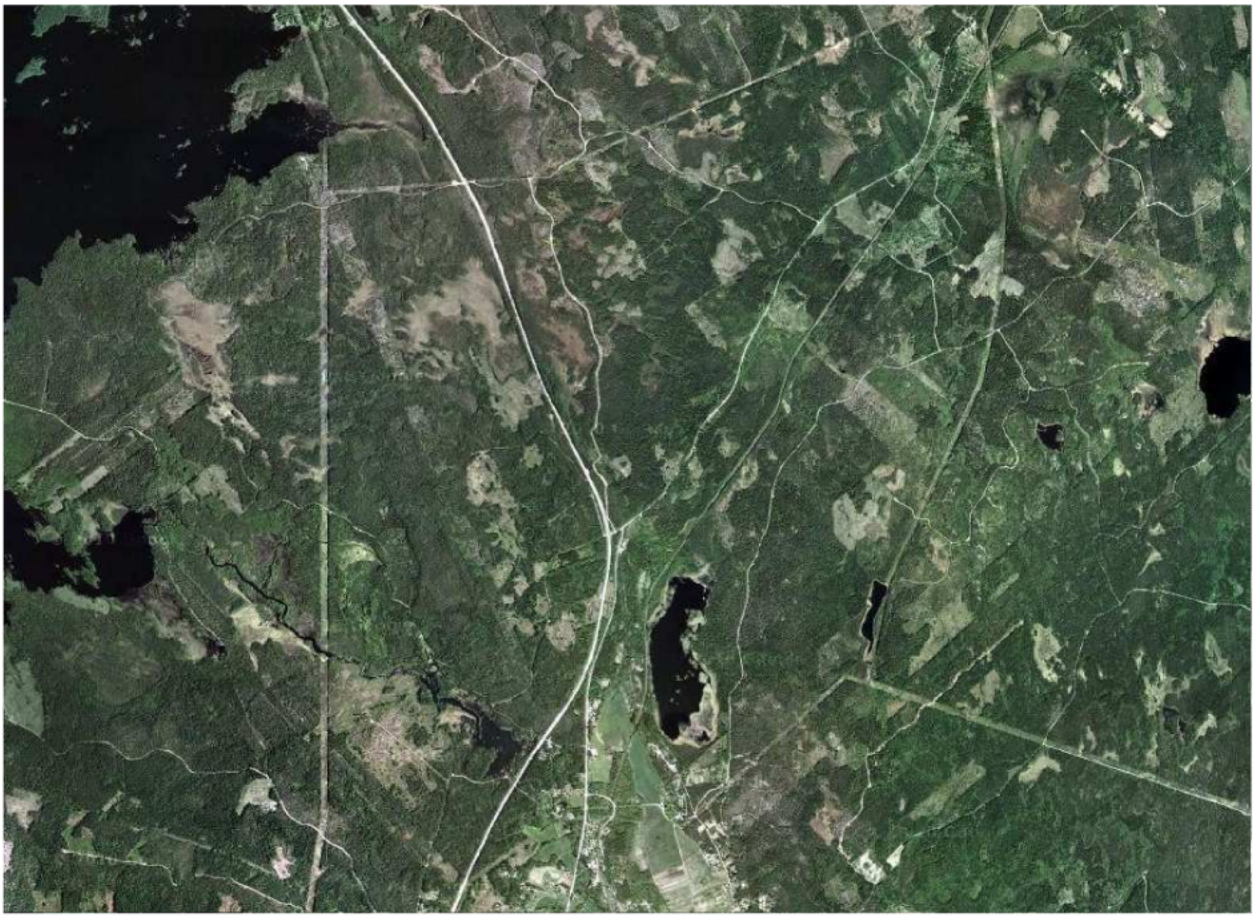
See http://en.wikipedia.org/wiki/QR_Code and http://www.swetake.com/qr/qr1_en.html .





Project 9: Road detection in satellite images

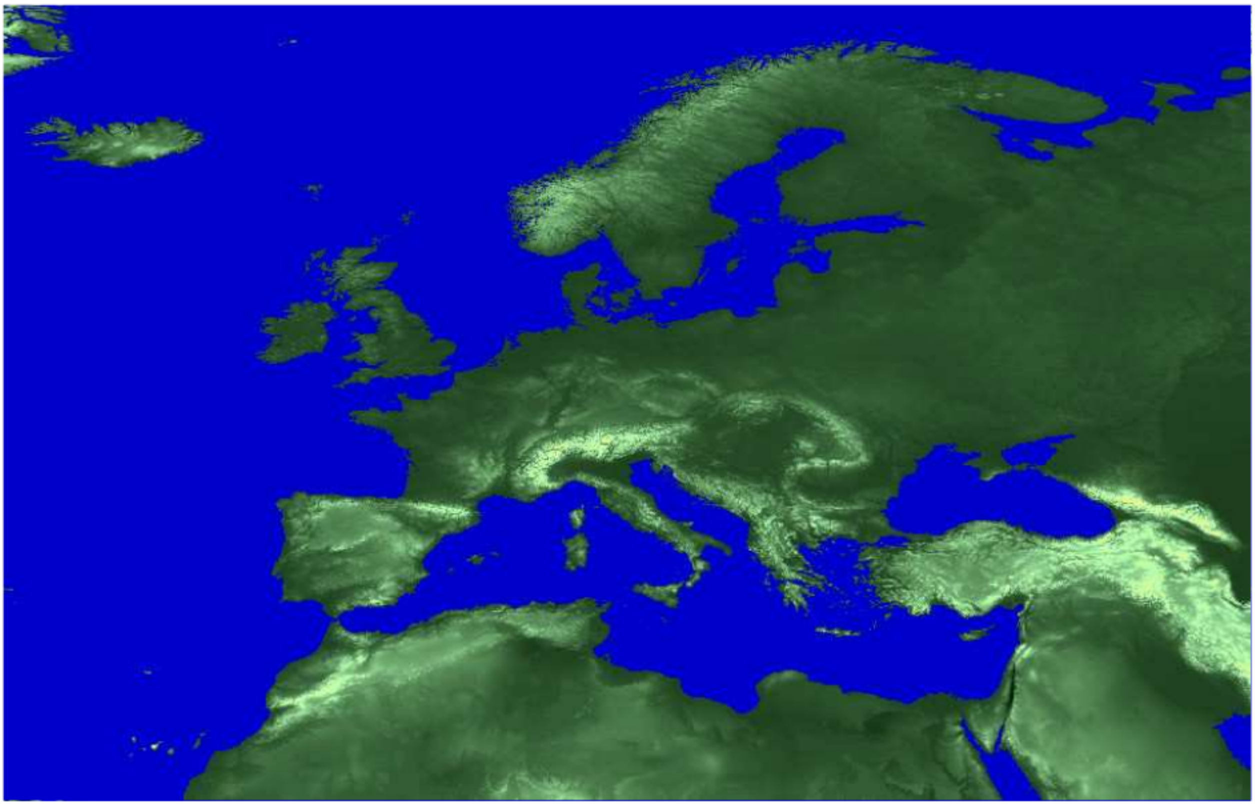
A common problem is to detect roads in satellite images. The image below is taken from http://maps.google.com/ . Get some screen shots from that site and write a program that detects the roads. Warning: this problem is not as easy as it looks!



Project 10: Where does that river end?

Image analysis routines are also useful for very different data analysis tasks. For example, the file http://www.cb.uu.se/~cris/bild2projects/europe.mat contains a matrix with elevation data for Europe. The value at each point is the distance in meters above sea level, and is averaged over an area of one square kilometre. The seas and oceans have been marked with the value -500. The task for this project is to use image analysis routines to identify the portions of land mass that evacuate their rivers in each of the seas.

Note that the matrix is very large, and processing might therefore be quite slow. Try down sampling the data first, while developing the algorithms. Once the algorithm is working, run it on the full data set to get a high-resolution result.



The data comes from the Global Land One-kilometer Base Elevation (GLOBE) data set, made available by the National Geophysical Data Center (http://www.ngdc.noaa.gov/mgg/topo/report/).

Project 11: Registration

I found a bunch of images of the main building of MIC online. The first two images are from Google Street View, and are exactly the same photograph, but with different distortions to make it look like a different point of view. The other two images are taken from different perspectives, and under very different conditions. Write a program that matches these images, and warps them all to a common geometry. The difference in colours, shadows, background, and plants make this task complicated. There are two different approaches to this task:

- Compare the pixel values directly, using a suitable error measure. Find a geometric transformation that minimizes the error measure.
- Find relevant landmarks in one image, then search for those same landmarks in the second one. Find a geometric transformation that matches these landmarks as well as possible.

Evaluate both approaches.

You can fetch the images here: http://www.cb.uu.se/~cris/bild2projects/MIC.zip, or you can use images of a different object to work on.