IN55200 – Digital Image Analysis

INTRODUCTION

- Practical information
- What will you learn in this course?
- Examples of applications of digital image analysis
- Repetition of key material from INF2310

Fritz Albregtsen & Anne H.S. Solberg 21.08.2018

Practical information - Lecturers

- **Fritz Albregtsen**
  - IFI/UiO (Fourth floor, room 4459, OJD building)
  - Telephone: *22852463 / 911 63 005*
  - Email: *fritz@ifi.uio.no*

- **Anne Schistad Solberg**
  - IFI/UiO (Fourth floor, room 4458, OJD building)
  - Telephone: *22852435*
  - Email: *anne@ifi.uio.no*

- **Kristine**
  - Email: *krisbhei@student.matnat.uio.no*

Practical information - Schedule

- Lectures
  - **Fritz Albregtsen and Anne Schistad Solberg**
  - When: Tuesday 10:15-12:00.
  - Where: “Logo” (2438), OJD (IFI2)
- Exercises
  - **Kristine Baluka Hein**
  - Group 1:
    - When: ~~Wednesday 08:15-10:00~~. Tuesday 12:15-14:00
    - Where: ~~“Modula” (2443)~~, “Assembler” (3417)
      First time 28.08.2018
- IFI2 Coordinates:

X _ _ _ [0,...,10]: Floor

_ X _ _ [1,..., 4]: Proximity to Metro line

_ _ X X [1, ...,72]: Distance from Research Park

Web page

- http://www.uio.no/studier/emner/matnat/ifi/IN5520/
  - Information about the course
  - Lecture plan
  - Lecture notes
  - Exercise material
  - Course requisite description
  - Exam information
  - Messages

Course material

- All slides will be made available on the course web site.
- The slides define the course requisites.
- Exercises will be introduced as we go along.

- No books defining all course requisites
  - Gonzalez & Woods: Digital Image Processing, 3rd ed., 2008.
  + additional material

Exercises

- The ordinary weekly exercises are NOT obligatory.
  - But definitely a good idea to do them anyway ☺
  - The ordinary exercises can be solved in any programming language, solutions will be provided in Matlab.

- Mandatory exercises (“term project”)
  - Two parts (October & November)
  - Individual work

  - A little extra work for PhD-students taking this course as IN9520

Exam

- Written exam (4 hours), date is not determined yet ...
- No written sources of information allowed at exam
- A little extra work for PhD-students taking this course as IN9520
- Follow the web page for updates on the exam.

Term project

- Sadly, we see plagiarism and cheating on term papers, but the reaction may be severe.
- Therefore you should read the following document:
  www.uio.no/studier/admin/obligatoriske-aktiviteter/mn-ifi-oblig.html
  (in Norwegian)

  Please notice routines on cheating and plagiarism!

- Using available source code and applications is **perfectly OK** and will be **credited** as long as the origin is cited

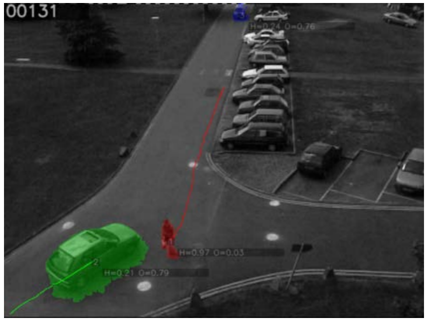
- The term project is **individual** work, and the handed in result should clearly be your own work

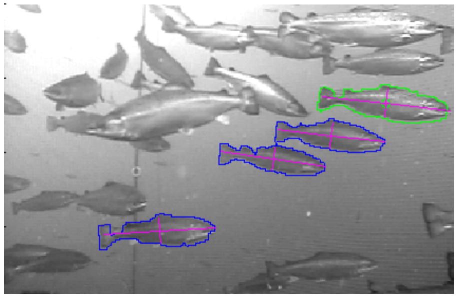
Lecture plan (left margin of web page)

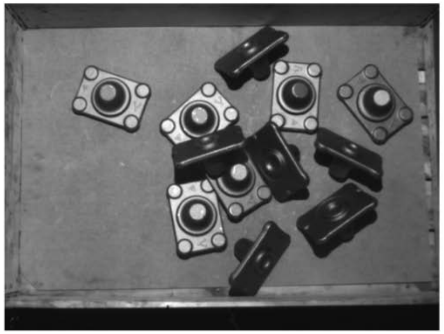
August	20	21	22	23	24	25	26	Introduction and preliminaries	Anne / Fritz
	27	28	29	30	31	01	02	Features from images, Texture	Fritz
September	03	04	05	06	07	08	09	Local to global: Hough Transform	Fritz
	10	11	12	13	14	15	16	Object representation	Fritz
	17	18	19	20	21	22	23	Object description	Fritz
	24	25	26	27	28	29	30	Feature extraction	Anne
October	01	02	03	04	05	06	07		
	08	09	10	11	12	13	14	Introduction to Classification	Anne
	15	16	17	18	19	20	21	Support Vector Machines	Anne
	22	23	24	25	26	27	28	Classification in practice, Unsupervised classification	Anne
November	29	30	31	01	02	03	04	Feature selection & feature transforms	Anne
	05	06	07	08	09	10	11	Edge based segmentation	Fritz
	12	13	14	15	16	17	18	Mathematical morphology	Fritz
	19	20	21	22	23	24	25		
	26	27	28	29	30	01	02	Previous exam exercises	Anne / Fritz
December	03	04	05	06	07	08	09	Course summary	Fritz / Anne
	10	11	12	13	14	15	16		
	17	18	19	20				**EXAM ???**	

What is image analysis ?

- **Image analysis** is the art and science whose ultimate goal is to give computers "*vision*"
  - Read handwritten documents
  - Recognize people
  - Find objects
  - Measure the world in 3D
  - Guide robots
  - Decision support (e.g. medical)
- **Image processing** is often used in the more limited sense of *simple image manipulations*:
  - Removing noise
  - Changing contrast
  - Improving edges
  - Coding and compression

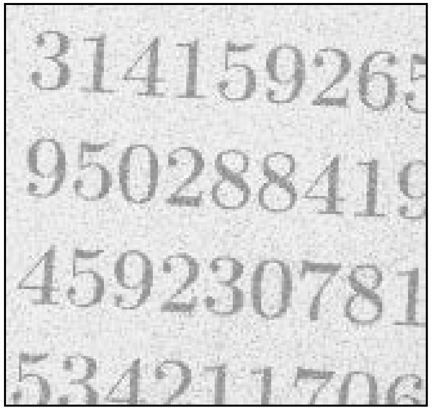




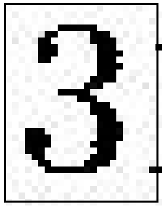


From pixels to features to class

- Objects often correspond to regions. We need the spatial relationship between the pixels.
- For text recognition: the information is in the shape, not in the gray levels.
- Classification: learn features that are common for one type of objects.



Segmentation →



Region feature extraction →

Region features:
- Area
- Perimeter
- Curvature
- Moment of inertia
- Topology
- ....

Classification → 3

Object classification - introduction

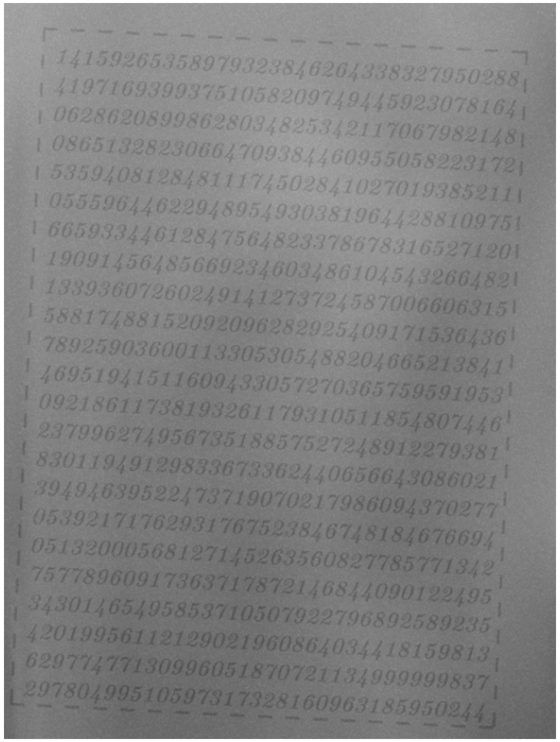
→ DOG

Task: use the entire image to classify the image into one of a set of known classes

Which object does the image contain ?

Conventional approach to image classification

- Goal: get the series of digits,
  e.g. 14159265358979323846......

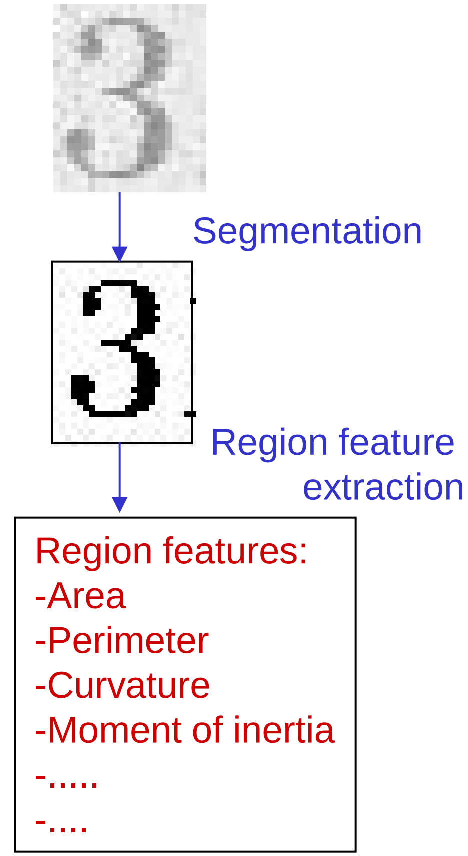


Steps in the program:

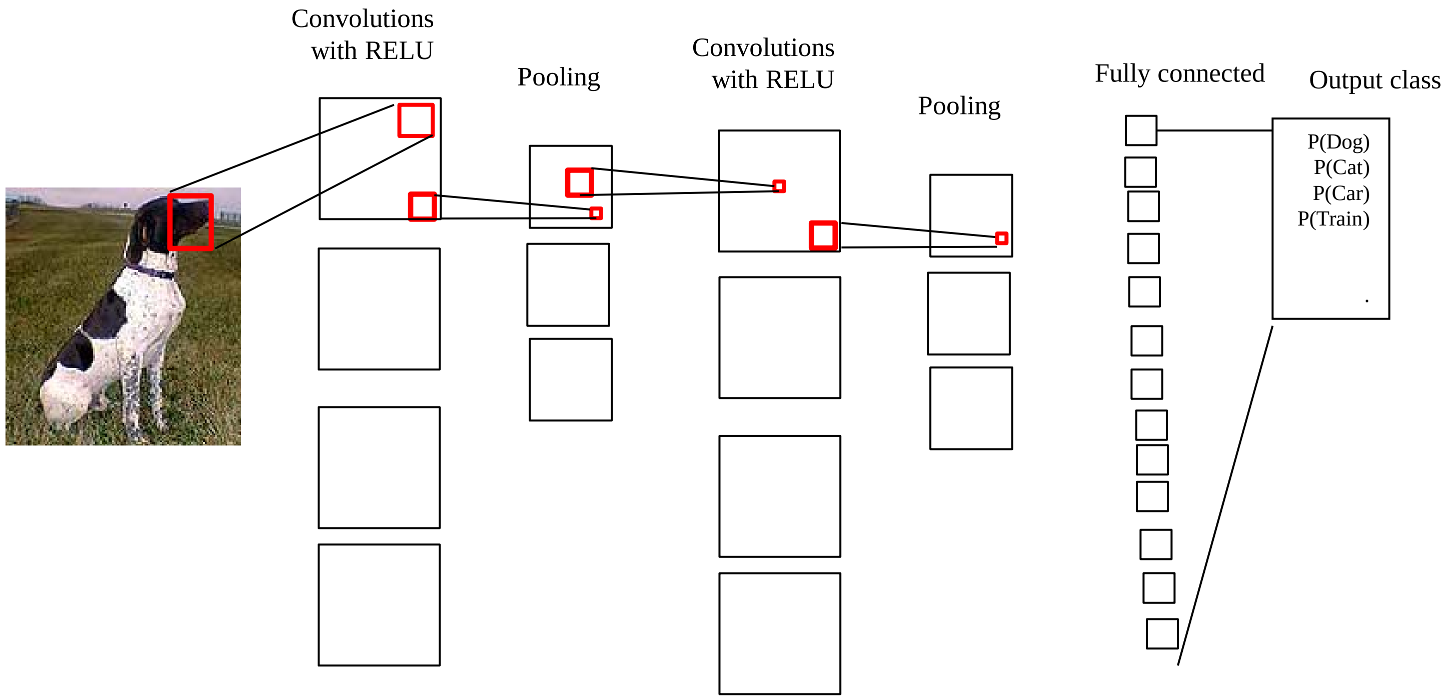
1. **Segment** the image to find digit pixels.
2. Find angle of rotation and rotate back.
3. Create region objects – **one object pr. digit** or connected component.
4. **Compute features** describing shape of objects
5. Train a classifier on many objects of each digit.
6. Assign a class label to each new object, i.e., the class with the highest probability.

Feature-based classification

- Input to the classifier is a set of features derived from the image data, not the image data itself.
- We would either:
  - Segment the image to identify each object, then extract features from the objects pixels, and classify the object..
  - OR: Compute feature for each pixel in a sliding window around each pixel (e.g. texture features)
  - Classification would be done pixel-by-pixel.



Classification using a convolutional neural net



State-of-the art image classification

- Deep learning using convolutional nets (INF 5860) can solve many image classification problems
  - In particular: classify ONE object in an image
  - Steady progress each year

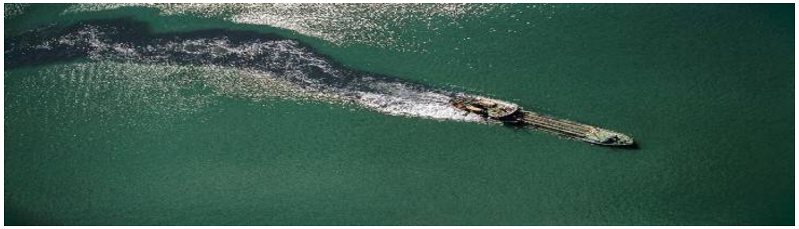
- Many problems cannot be solved using convolutional nets, so traditional image analysis methods are highly needed.

- A good basis for image analysis and machine learning is:
  - IN 55200 (Fall)
  - INF 5860 Machine learning for image analysis (Spring)

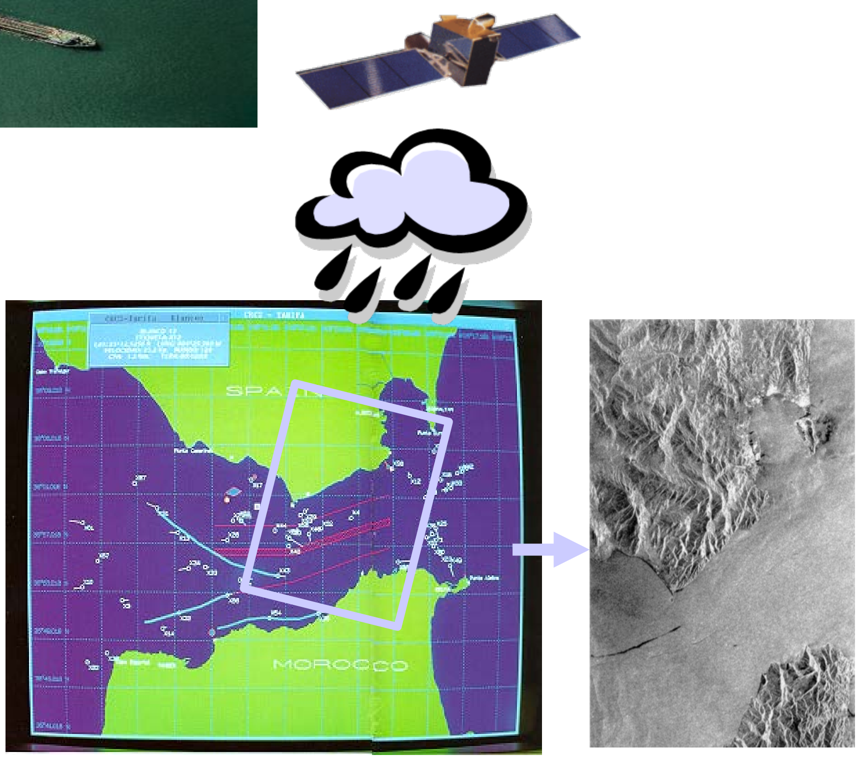
Applications of image analysis ...

- Medical applications, e.g., ultrasound, MR, cell images
- Industrial inspection
- Traffic surveillance
- Text recognition, document handling
- Coding and compression
- Biometry
  - identification by face recognition, fingerprint or iris
- Earth resource mapping by satellite images
- Sea-bed mapping (sonar)
- Mapping of oil reservoirs (seismic)

Example: Oil-spill detection

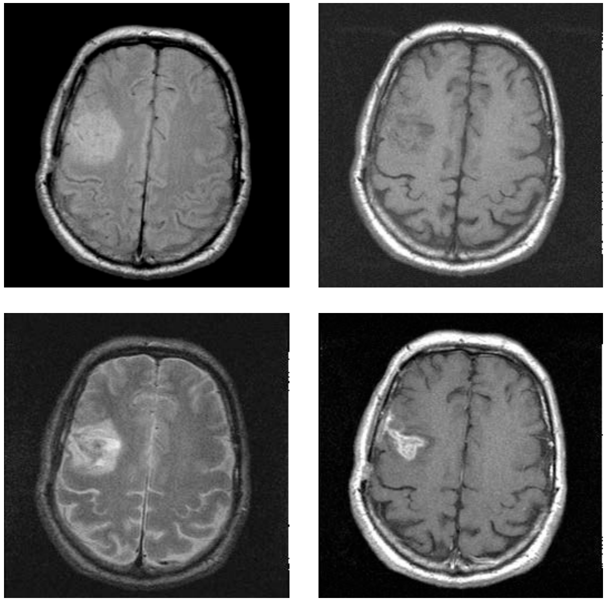


Tanker spilling oil

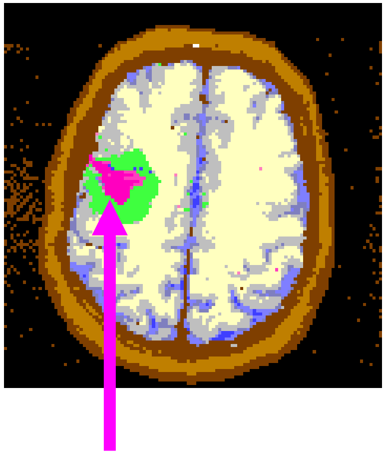


Radar image of oil-spill

Example: Tissue classification in MR images



MR images of brain



Classification into tissue types. Tumor marked in red.

Weed recognition in precision farming

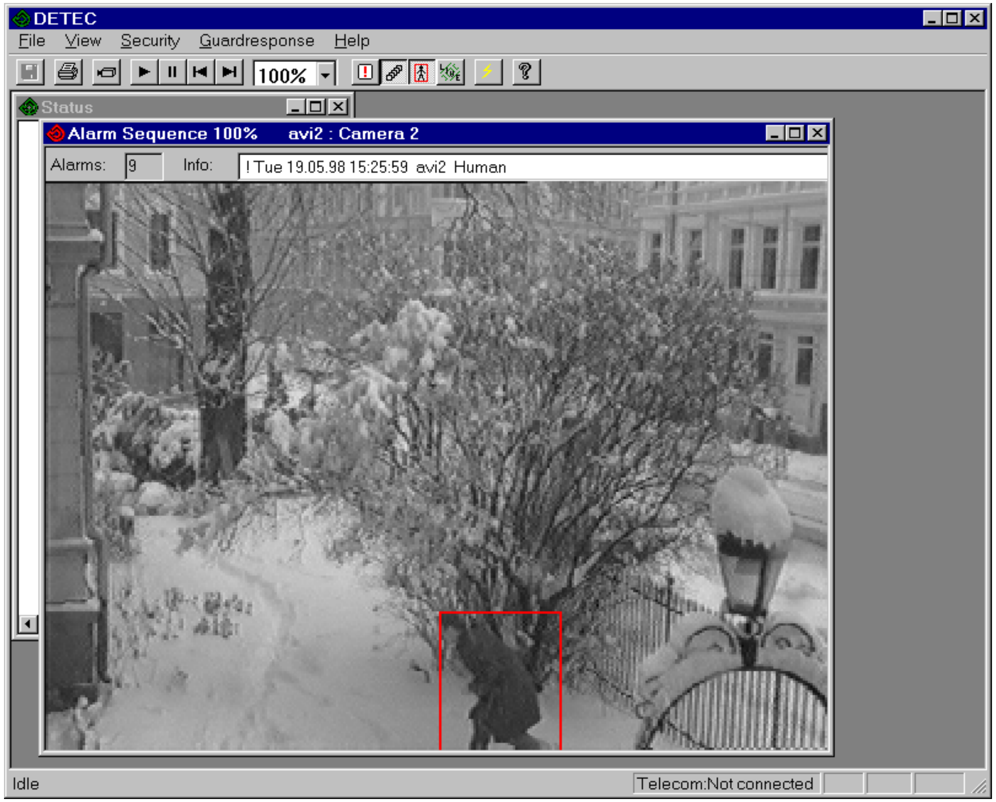
- Detect and recognize invasive weed species in cereal fields

- Classify weeds in real time to enable on-line control of herbicide spray

- Largely unsolved problem, potential huge savings in weed control costs (commercial potential!)

Smart video surveillance

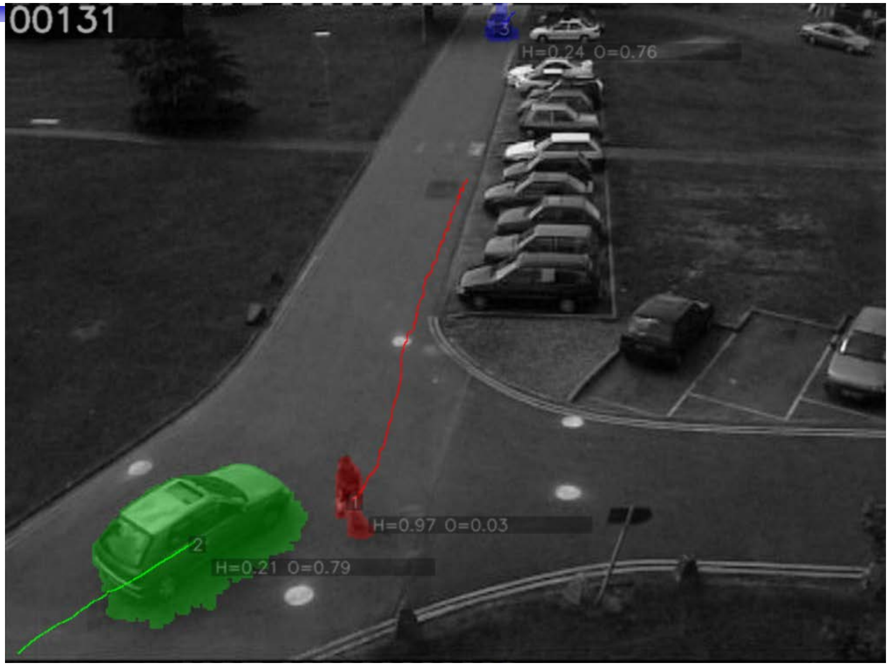
- Detect and classify events in real-time in surveillance video
- Track objects and alert if humans enter no-go-zones
- Outdoor imagery is challenging, wind, weather and sun causes large changes in scene

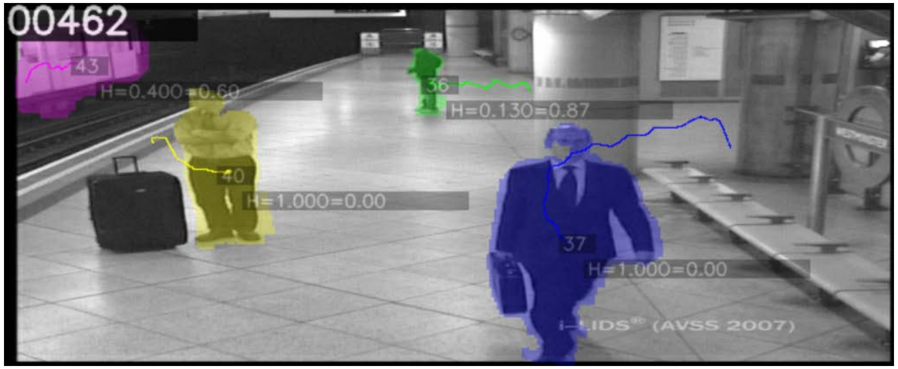


DETEC

Tracking and classification of objects

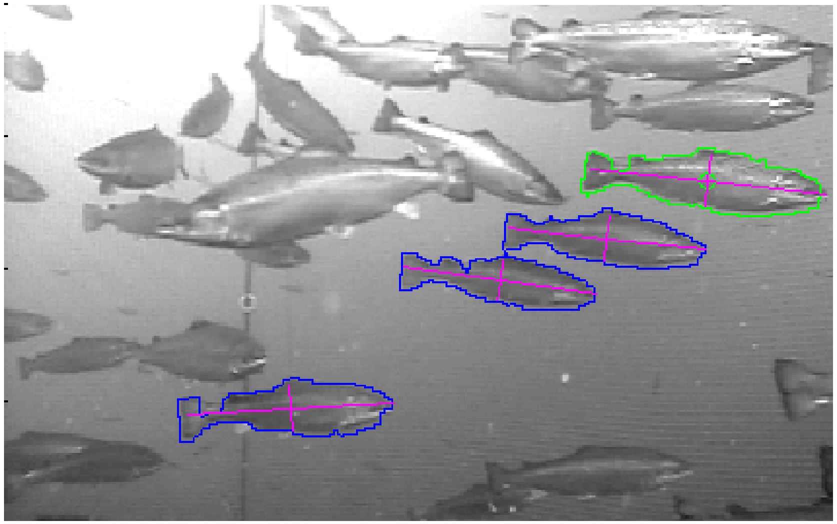
- Challenges:
  - Objects may be poorly segmented or occluded, so shape or appearance models may be useless
  - One blob may contain several objects
- Solutions:
  - Analyze motion patterns within blobs (decide object class)
  - Detect heads, arms and other human parts (decide number of objects within blob)





Automatic fish segmentation

- Pick single fish from underwater video of a fish farm
- Estimation of fish statistics
  - Size (for weight estimates)
  - Motion
- Challenges:
  - Illumination varies
  - Seawater murky, food / particles
  - No contrast
  - Fish overlap
  - Fish may swim in any direction
- Solution:
  - Active contours, initialized with a fish-shape
  - Use information from two cameras



INF2310 – a brief repetition

- See https://www.uio.no/studier/emner/matnat/ifi/INF2310/v18/undervisningsplan/

- Topics covered in the course:
  - Image representation, sampling and quantization.
  - Compression and coding
  - Color imaging
  - Grey-level mapping
  - Geometrical operations

Assumed known

  - Filtering and convolution in the image domain
  - Fourier transform
  - Segmentation by thresholding
  - Edge detection

Good understanding needed

2-D convolution

- Input image f(x,y)
  The resulting image *g(x,y)* is given by

$$g(x,y) = \sum_{j=-w_1}^{w_1} \sum_{k=-w_2}^{w_2} h(j,k) f(x-j, y-k)$$

$$= \sum_{j=x-w_1}^{x+w_1} \sum_{k=y-w_2}^{y+w_2} h(x-j, y-k) f(j,k)$$

- h is a $m \times n$ filter with size $m=2w_1+1$, $n=2w_2+1$
- The result is a weighed sum of the input pixels surrounding pixel *(x,y).* The weights are given by *h(j,k).*
- The pixel value of the next pixel in the out image is found by moving the filter one position and computing again.

Separable filters

- Geometrical shapes: rectanglar and square
- Rectangular mean filters are separable.

$$h(i,j)=\frac{1}{25}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}=\frac{1}{25}\begin{bmatrix}1 & 1 & 1 & 1 & 1\end{bmatrix}*\begin{bmatrix}1\\ 1\\ 1\\ 1\\ 1\end{bmatrix}$$

- Advantage: fast filtering

Non-uniform low pass filters

– 2D Gauss-filter:

$$h(x,y) = \exp\left(-\frac{\left(x^2+y^2\right)}{2\sigma^2}\right)$$

– Parameter σ is standard deviation (width)

– Filter size must be set relative to σ

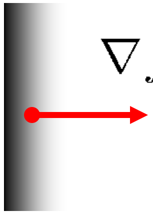
Digital gradient operators

- The gradient of *f(x)* is $\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

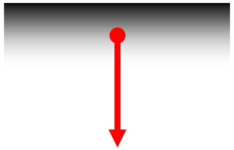
- The (intensity) gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

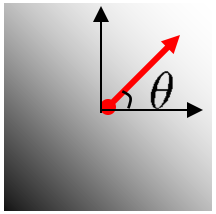
- The gradient points in the direction of most rapid (intensity) change



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y} \right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

Gradient operators

- Prewitt-operator

$$H_x(i,j)=\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix},\; H_y(i,j)=\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix}$$

- Sobel-operator

$$H_x(i,j)=\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix},\; H_y(i,j)=\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$$

- Frei-Chen-operator

$$H_x(i,j)=\begin{bmatrix}1 & 0 & -1\\ \sqrt{2} & 0 & -\sqrt{2}\\ 1 & 0 & -1\end{bmatrix},\; H_y(i,j)=\begin{bmatrix}-1 & -\sqrt{2} & -1\\ 0 & 0 & 0\\ 1 & \sqrt{2} & 1\end{bmatrix}$$

Gradient direction and magnitude

- Horisontal edge component:
  - Compute $g_x(x,y)=H_x*f(x,y)$

  => Convolve with the horisontal filter kernel H_x
- Vertical edge component:
  - Compute $g_y(x,y)=H_y*f(x,y)$

  => Convolve with the vertical filter kernel H_y

The *gradient direction* is given by:

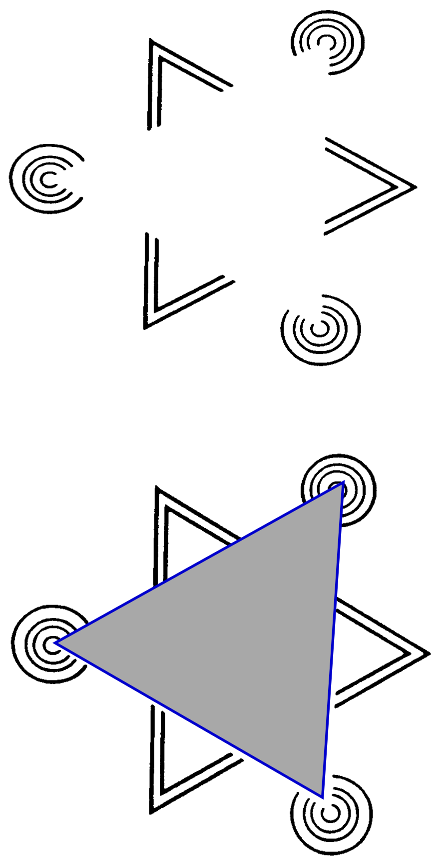
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

The *edge strength* is given by the gradient magnitude

$$||\nabla f|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Edge extraction

- Several basic edge extraction techniques were taught in INF2310
- In this context edges are both edges in intensity, color and texture
- Edges are important for many reasons:
  - Much of the information in an image is contained in the edges. In many cases semantic objects are delineated by edges
  - We know that biological visual systems are highly dependent on edges



Edge extraction

- The standard operator is the so called Sobel operator.
- In order to apply Sobel on an image you convolve the two x- and y-direction masks with the image:

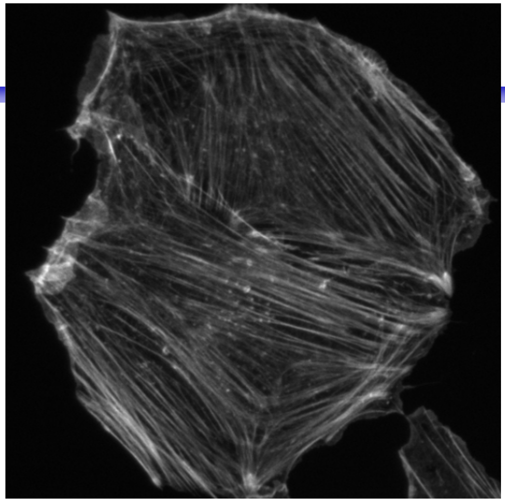
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

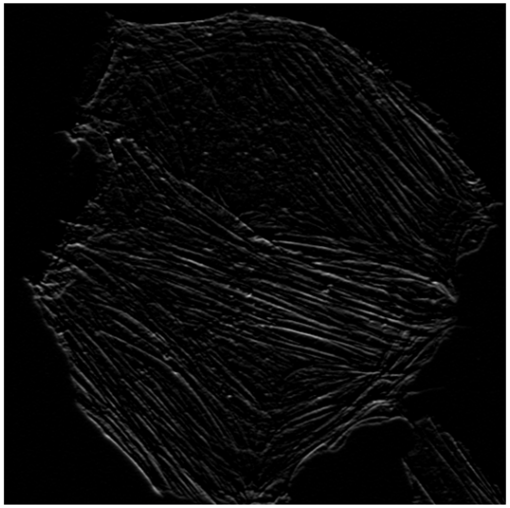
Edge extraction - Sobel

- This will give you two images, one representing the horizontal components of the gradient, one representing the vertical component of the gradient.

- Thus using Sobel you can derive both the local gradient magnitude and the gradient direction.



Grayscale image



«Horizontal» edges

Edge extraction - Laplace

- Another frequently used technique for edge detection is based on the use of discrete approximations to the *second derivative.*
- The *Laplace operator* is given by

$$\nabla^2(f(x,y)) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- This operator changes sign where f(x,y) has an inflection point, it is equal to zero at the exact edge position

Edge extraction - Laplace

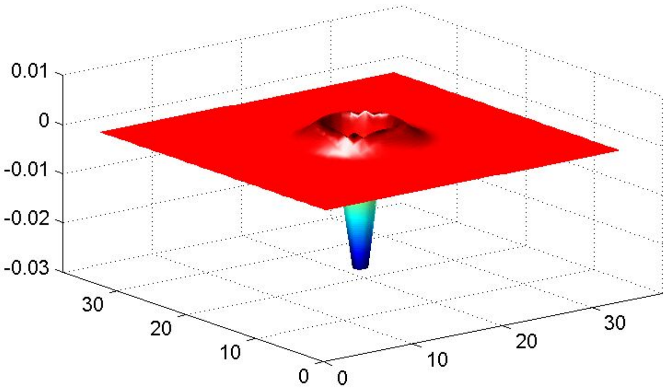
- Approximating second derivatives on images as finite differences gives the following mask

$$\nabla^2(f(x,y)) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$
$$\approx -f(x-1,y) + 2f(x,y) - f(x+1-y)$$
$$-f(x,y-1) + 2f(x,y) - f(x,y+1)$$

0	-1	0
-1	4	-1
0	-1	0

Edge extraction - LoG

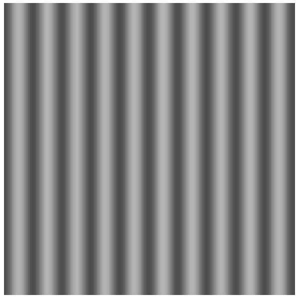
- Since the Laplace operator is based on second derivatives it is extremely sensitive to noise.
- To counter this it is often combined with Gaussian pre-filtering in order to reduce noise.
- This gives rise to the so called Laplacian-of-Gaussian (LoG) operator.



Sinusoids in images

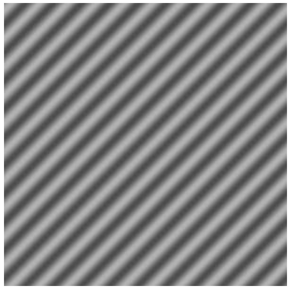
$$f(x,y) = 128 + A\cos\left(\frac{2\pi(ux+vy)}{N} + \phi\right)$$

A - amplitude
u - horisontal frequency
v - vertical frequency
ϕ - phase

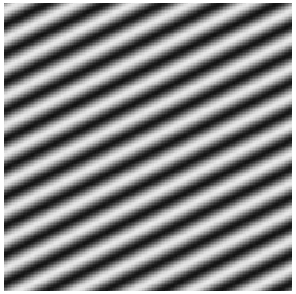


A=50, u=10, v=0

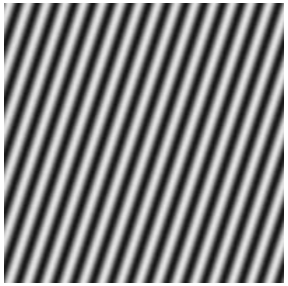
A=20, u=0, v=10



A=50, u=10, v=10



A=100, u=5, v=10



A=100, u=15, v=5

Note: u and v are the number of cycles (horisontally and vertically) in the image

2-D Discrete Fourier transform (DFT)

f(x,y) is a pixel in a $N \times M$ image

Definition:

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

$$\boxed{e^{j\theta}=\cos\theta+j\sin\theta}$$

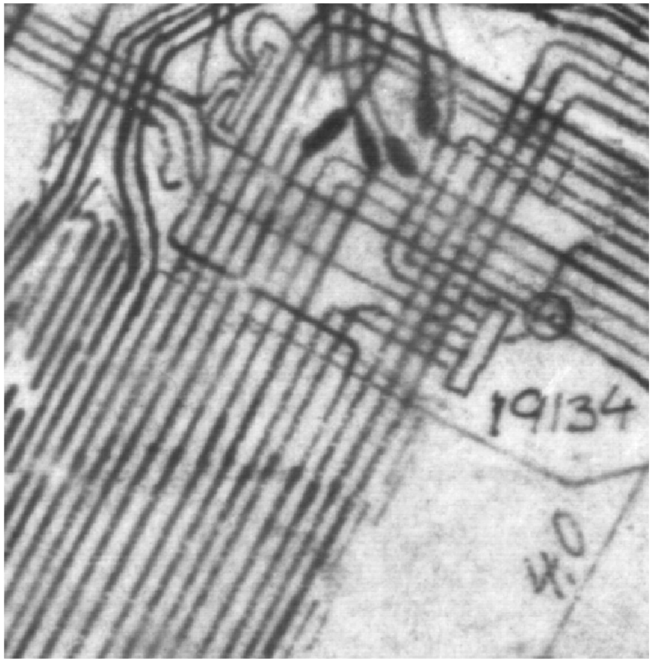
This can also be written:

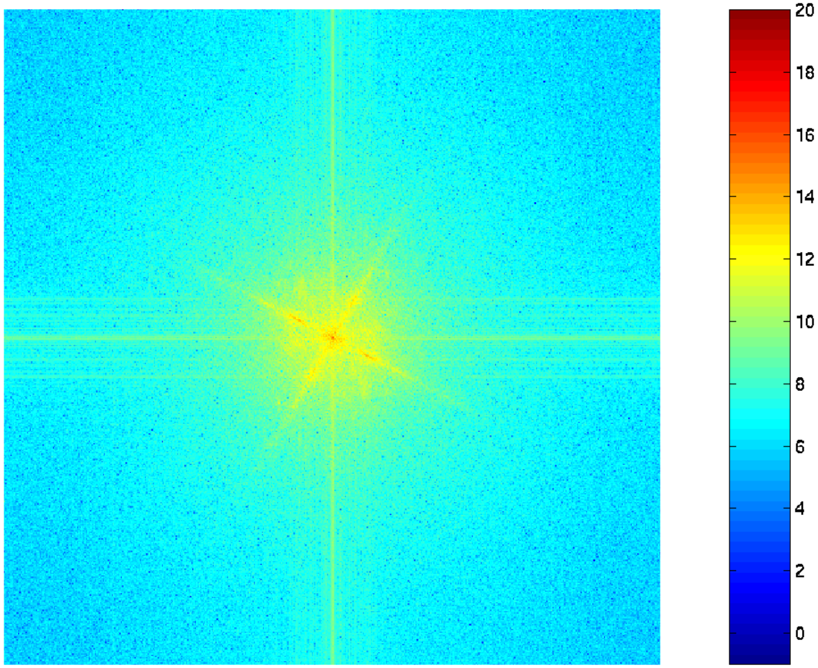
$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\left[\cos(2\pi(ux/M+vy/N))-j\sin\left(2\pi(ux/M+vy/N)\right)\right]$$

Inverse transform:

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$

Example – oriented structure





The convolution theorem

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v) \cdot H(u,v)$$

Convolution in the image domain

$\Leftrightarrow$

Multiplication in the frequency domain

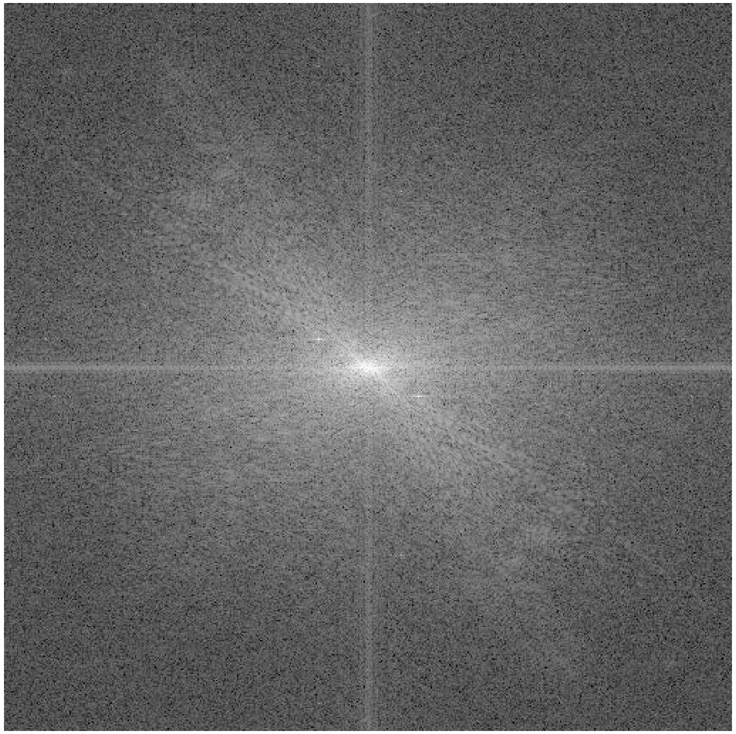
$$f(x,y) \cdot h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

Multiplication in the image domain

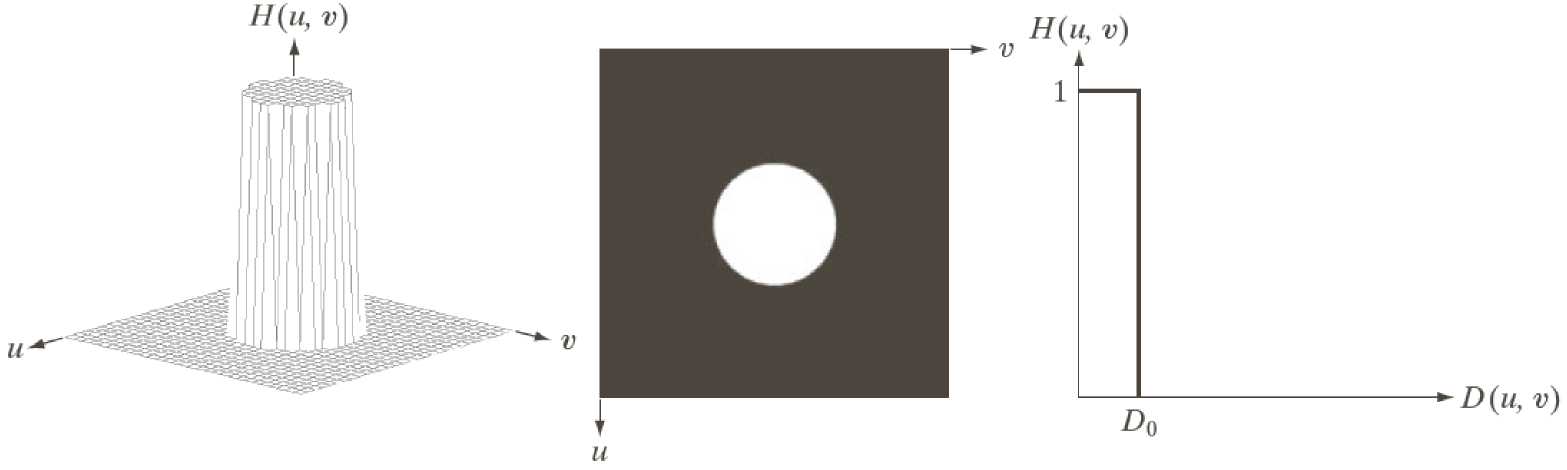
$\Leftrightarrow$

Convolution in the frequency domain

How do we filter out this effect?



The ”ideal” low pass filter



a b c

FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Example - ideal low pass

Original

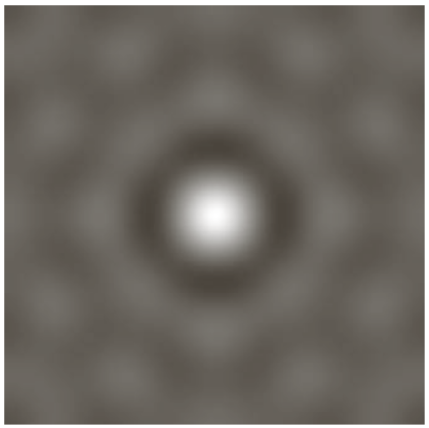
D_0=0.2

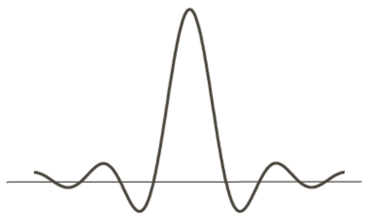
D_0=0.3

Look at these image in high resolution.
You should see ringing effects in the two rightmost images.

What causes the ringing effect?

Ideal lowpass in the image domain





fft of H(u,v)
for ideal lowpass

1D profile
for ideal lowpass

- Note that the filter profile has negative coefficients
- It has similar profile to a Mexican-hat filter (Laplace-of-Gaussian)
- The radius of the circle and the number of circles per unit is inversely proportional to the cutoff frequency
  - Low cutoff gives large radius in image domain

Butterworth low pass filter

- Window-functions are used to reduce the ringing effect.
- Butterworth low pass filter of order *n* :

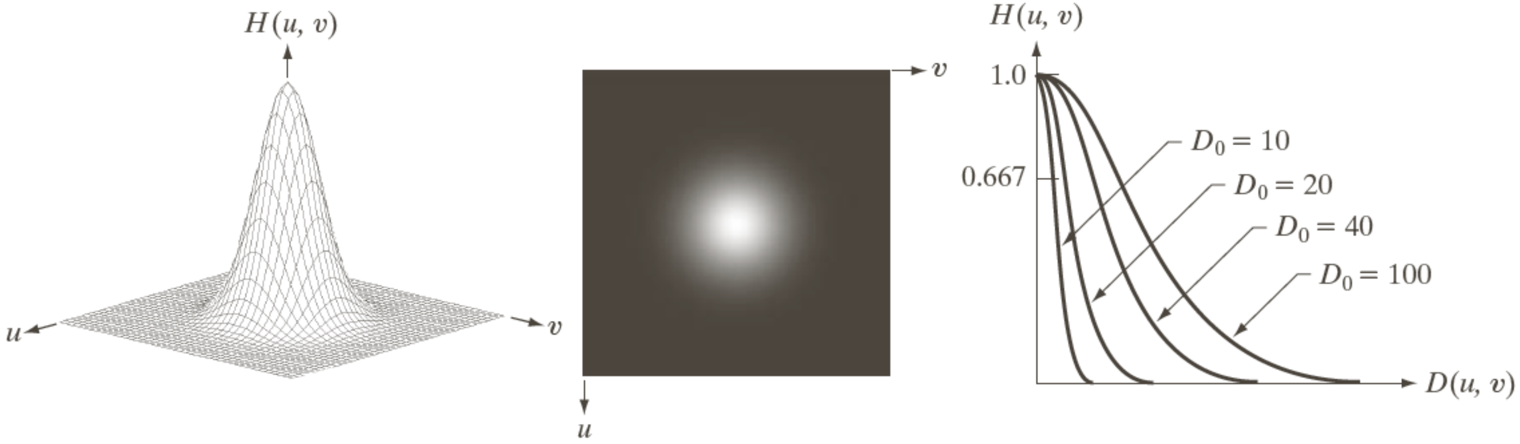
$$H(u,v) = \frac{1}{1 + \left[D(u,v) / D_0 \right]^{2n}}$$

- D_0 describes the point where *H(u,v)* has decreased to half of its maximum
  - Low filter order (*n* small): *H(u,v)* decreases slowly: Little ringing
  - High filter order (*n* large): *H(u,v)* decreases fast: More ringing

- Other filters can also be used,
  e.g.: Gaussian, Bartlett, Blackman, Hamming, Hanning

Gaussian lowpass filter

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$



a b c

FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

High pass filtering

- Simple (”Ideal”) high pass filter:

$$H_{hp}(u,v) = \begin{cases} 0, D(u,v) \le D_0, \\ 1, D(u,v) > D_0. \end{cases}$$

or

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

- Butterworth high pass filter:

$$H_{hpB}(u,v) = \frac{1}{1 + \left[D_0 / D(u,v)\right]^{2n}}$$

- Gaussian high pass filter:

$$H_{hpG}(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Ideal, Butterworth and Gaussian highpass

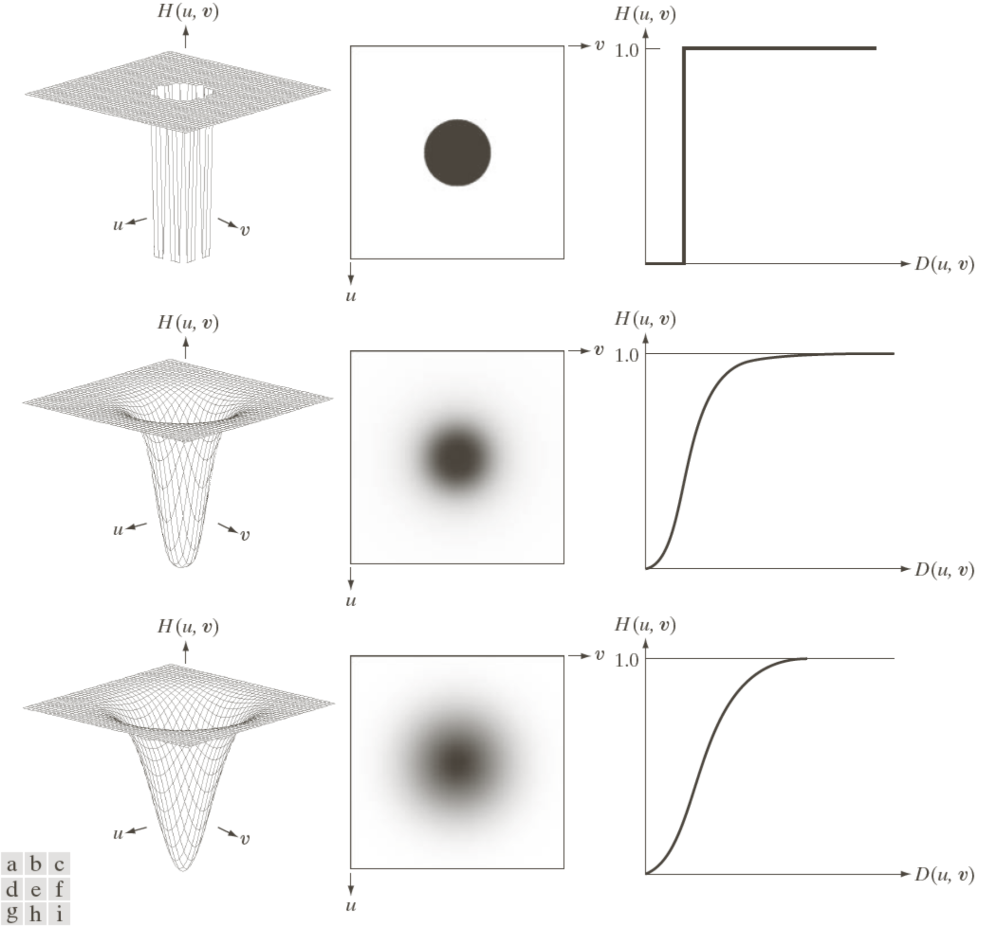


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Example – Butterworth highpass

a b c

FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

Bandpass and bandstop filters

- Bandpass filter: Keeps only the energy in a given frequency band $<D_{low},D_{high}>$ (or $<D_0-W/2,D_0+ W/2>$)

- W is the width of the band
- D_0 is its radial center.

- Bandstop filter: Removes all energy in a given frequency band $<D_{low},D_{high}>$

Bandstop/bandreject filters

- Ideal

$$H_{bs}(u,v)=\begin{cases}1 & \text{if } D(u,v)<D_0-\frac{W}{2}\\ 0 & \text{if } \mathrm{D}_0-\frac{W}{2}\le D(u,v)\le D_0+\frac{W}{2}\\ 1 & \text{if } D(u,v)>D_0+\frac{W}{2}\end{cases}$$

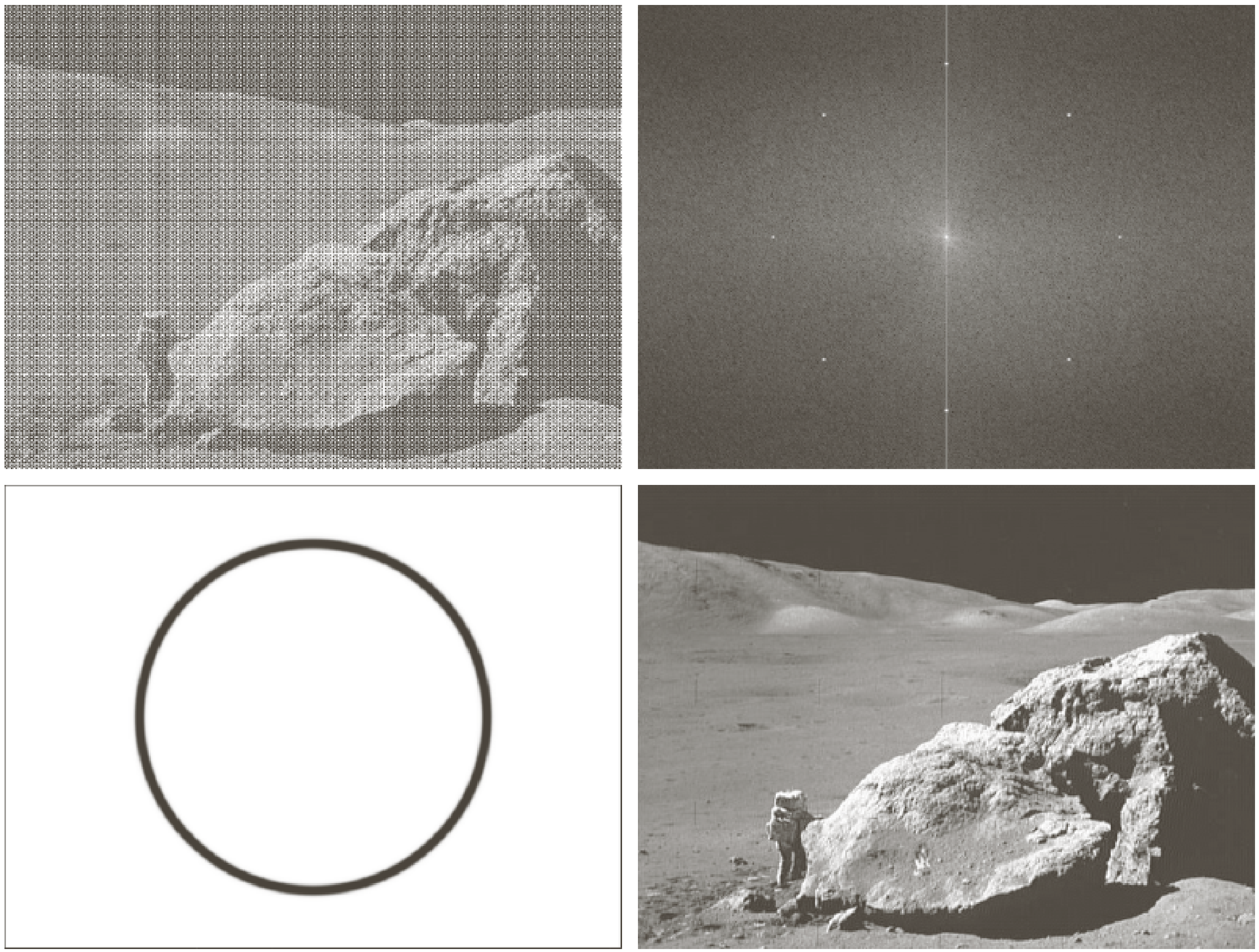
- Butterworth

$$H_{bs\mathbf{B}}(u,v)=\frac{1}{1+\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}}$$

- Gaussian

$$H_{bsG}(u,v)=1-e^{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2}$$

An example of bandstop filtering



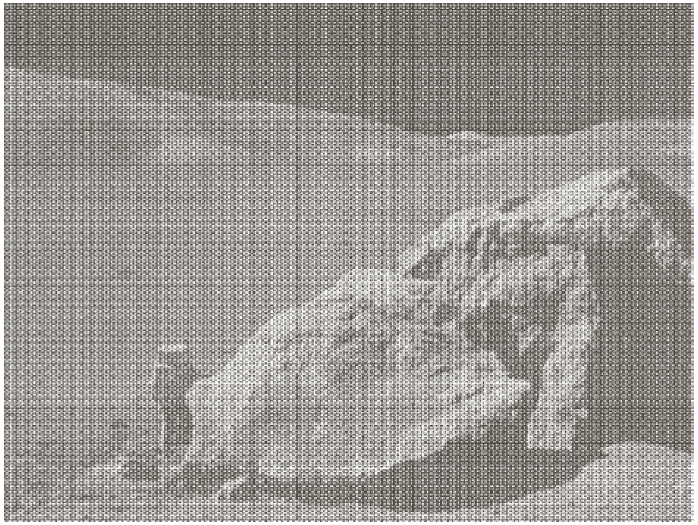
a b
c d

FIGURE 5.16
(a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

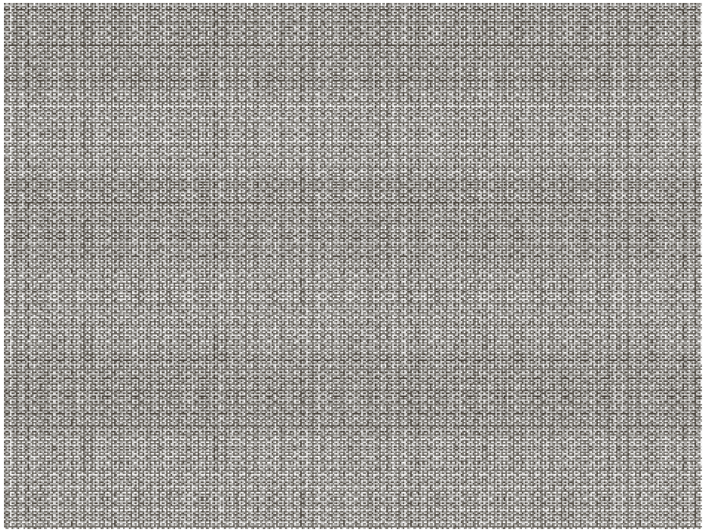
Bandpass filters

- Are defined by

$$H_{bp}(u,v) = 1 - H_{bs}(u,v)$$



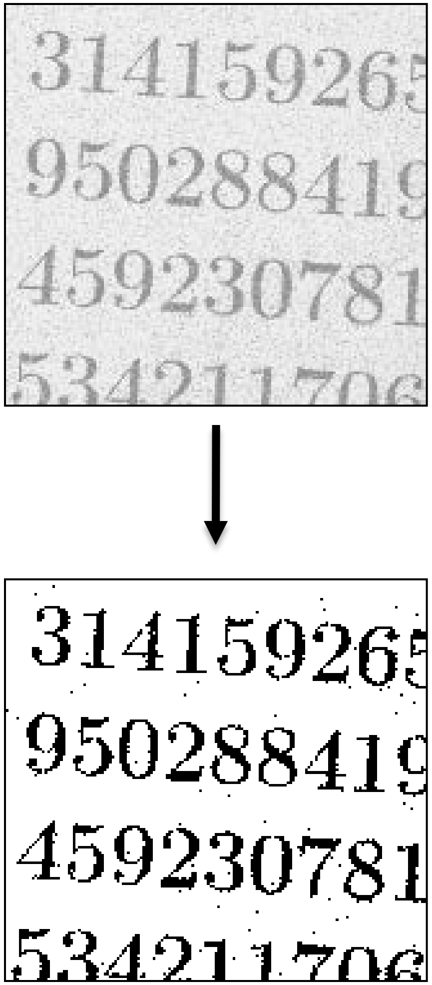
Original



Result after bandpass filtering

Segmentation and thresholding

- Segmentation
  - Function that labels each pixel in input image with a group label
  - Usually “foreground” and “background”
  - Each group shares some common properties
    - Similar color
    - Similar texture
    - Surrounded by the same edge
- Thresholding
  - One way of segmentation is by defining a threshold on pixel intensity



Segmentation and thresholding

Remember, regions that have semantic importance
do not always have any particular local visual distinction.

Segmentation and thresholding

- The only segmentation method taught in INF2310 was thresholding.
- Thresholding is a transformation of the input image *f* to an output (segmented) image *g* as follows:

$$g(i,j) = \begin{cases} 1, & f(i,j) \geq T \\ 0, & f(i,j) < T \end{cases}$$

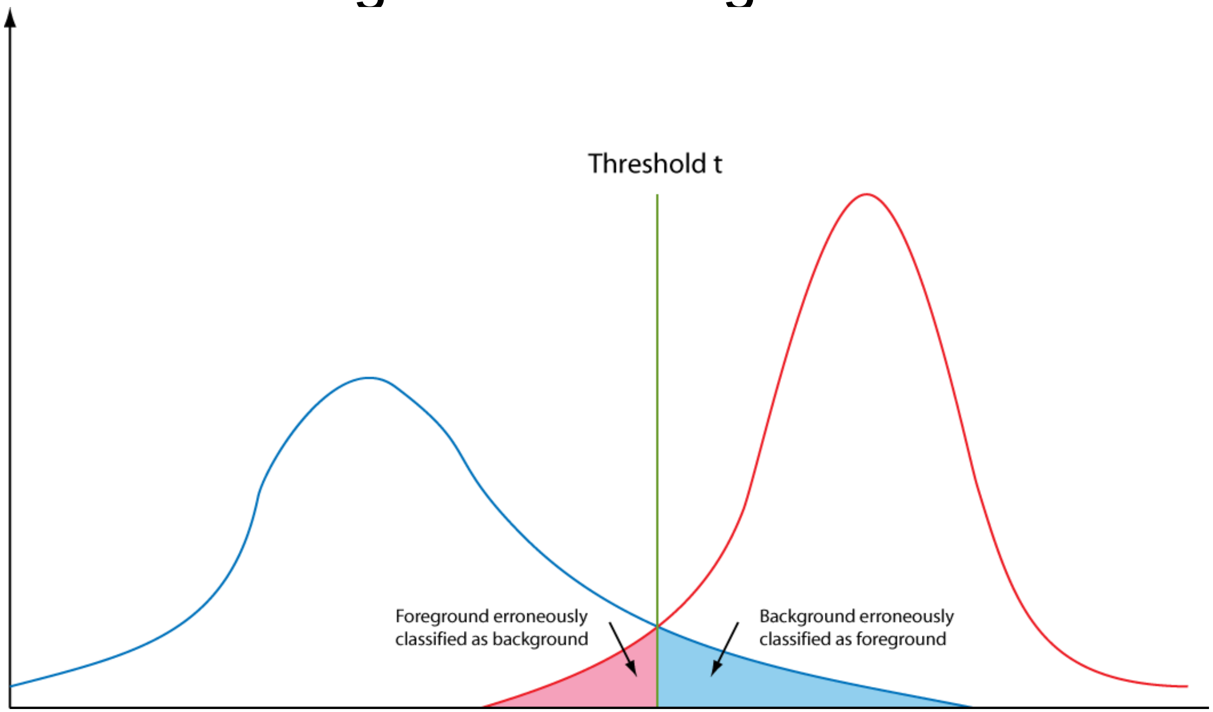
- Many variants of the basic definition ...

Segmentation and thresholding

- This seemingly simple method must be used with care:
  - How do you select the threshold, manually or automatically?
  - Do you set a threshold that is global or local (on a sliding window or blockwise)?
  - Purely local method, no contextual considerations are taken
- Automatic threshold selection methods:
  - Otsu's method
  - Ridler-Calvard's method
- Local thresholding methods:
  - Local applications of Otsu and Ridler-Calvard
  - Niblack's method

Segmentation and thresholding

- Remember that you normally make an error performing a segmentation using thresholding:



Segmentation and thresholding

- Assume that the histogram is the sum of two distributions $b(z)$ and $f(z)$, b and f are the normalized background and foreground distributions respectively, and z is the gray level.
- Let B and F be the prior probabilities for the background and foreground ($B+F=1$).
- In this case the histogram can be written

$$p(z)=Bb(z)+Ff(z).$$

Segmentation and thresholding

- The total thresholding error will be:

$$E(t) = F\int_{-\infty}^{t} f(z)dz + B\int_{t}^{\infty} b(z)dz$$

- Using Leibnitz's rule for derivation of integrals and by setting the derivative equal to zero you can find the optimal value for *t* :

$$\frac{E(t)}{dt} = 0 \Rightarrow Ff(T) = Bb(T)$$

Segmentation and thresholding

$$\frac{E(t)}{dt} = 0 \Rightarrow Ff(T) = Bb(T)$$

- This is a general solution.
- Does not depend on the type of distribution.

- In the case of *f* and *b* being Gaussian distributions, it is possible to solve the above equation explicitly.

Segmentation and thresholding

- In INF2310 we introduced two methods
  (Ridler-Calvard and Otsu)
  for determining segmentation thresholds automatically.

- Region- and edge-based methods will be covered
  in detail in the INF5520 lectures.

Exercise & next lecture

- Exercise: Practical use of Matlab, see web page.
- Next lecture: Features from images – Texture analysis.