ATHENA

Remote Sensing Science Center for Cultural Heritage

H2020-TWINN-2015. Grant Agreement no 691936	
Project full title:	Remote Sensing Science Center for Cultural Heritage
Project acronym:	ATHENA
Work Package	WP4
Deliverable	D4.2 Report of the 2nd Summer School

	H2020-TWINN-2015 **Grant Agreement no 691936** This project is funded under the **EUROPEAN COMMISSION** in the Framework Programme for Research and Innovation (2014-2020).	
Call:	Work programme **H2020** under **"Spreading Excellence and Widening Participation", call: H2020-TWINN-2015: Twinning** (Coordination and Support Action).	
Project full title:	**Remote Sensing Science Center for Cultural Heritage**	
Project acronym:	**ATHENA**	
Work Package (WP):	**WP4**	
Deliverable (D):	**D4.2 (Report of the 2nd summer school)**	
Due date of deliverable:	**June 2017** **(Month 19 of the project)**	**Version: 1**
Author(s):	**Daniele Cerra, Gunter Schreier**	
Contributor(s):	**Vasiliki Lysandrou, Athos Agapiou, Diofantos G. Hadjimitsis**	
Start date of project:	**1/12/2015**	Duration: **36 months**

Dissemination Level		
PU	Public	√
CO	Confidential, only for members of the consortium (including the Agency Services)	□

Document Sign-off				
Nature	**Name**	**Role**	**Partner**	**Date**
DRAFT	Daniele Cerra Gunter Schreier	Work Package Leader / Partner 2	DLR	19/06/2017
REVIEWED	Diofantos G. Hadjimitsis Athos Agapiou Vasiliki Lysandrou	Project leader	CUT	26/06/2017
APPROVED	Diofantos G. Hadjimitsis Gunter Schreier	Project coordinator Partner 2	CUT/DLR	30/06/2017

Work Package: 4 – Training and knowledge transfer Deliverable: D4.2 – Report on the 2nd Summer School				
Sections to be protected	Description	Owner	Access Rights	
			Period	Type*
none				

Table of Contents

Summary

The specific deliverable summarizes the material related to the 2nd Summer School of the project entitled "Special issues of Optical Remote Sensing". The deliverable contains actions completed prior the accomplishment of the Summer School, such as the agenda, while it also includes all the material delivered during the Summer School (e.g. presentations, supportive documents etc.), the list of participants and pictures from the event.

1. Introduction

The 2nd Summer School of ATHENA project has been successfully accomplished in line to the timeline of the project. The ATHENA Summer School took place in Cyprus University of Technology premises in Limassol, Cyprus between the 12th and the 15th of June 2017.

Visiting scientist from the Remote Sensing Technology Institute of DLR (Dr Daniele Cerra) met with members of the Remote Sensing and Geo-Environment Research Lab of the Department of Civil Engineering and Geomatics to introduce them to the typical processing chain for applications using satellite images, with a special focus on hyperspectral image processing and archaeological applications.

On Monday a reminder on image characteristics has been given, along with the properties of image filters carried out in time domain. The attendees have programmed and applied sample filters during the hands-on sessions.

On Tuesday, the concept of Fourier transform has been used to introduce filtering operations in the frequency domain. An overview on contextual analysis of image elements (edge extraction, texture estimation, invariant features) has been given, along with practical exercises building up from previous topics.

Wednesday and Thursday have been completely allocated to hyperspectral image processing. On Thursday, the related basic concepts have been introduced along with an overview on the applications, and a tutorial on dimensionality reduction has been given and tested by the attendees in the Matlab environment.

Following, spectral unmixing techniques have been used in the frame of a longer exercise aimed at performing supervised classification of different kinds of crops in a hyperspectral image. To close, an interactive tutorial on band selection has been given.

2. Agenda of the summer school

ATHENA 2nd Summer School Agenda

ATHENA

Remote Sensing Science Center for Cultural Heritage

2nd Summer School Agenda

Topic: Special issues of Optical Remote Sensing

Date: 12-15 June, 2017

Hosted by: Cyprus University of Technology

Venue: Dorothea 2nd floor
Cyprus University of Technology, Limassol, Cyprus

Trainer: Dr. Daniele Cerra (DLR)

Project Coordination Team

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 691936. Work programme H2020 under **"Spreading Excellence and Widening Participation"**, call: **H2020-TWINN-2015: Twinning** (Coordination and Support Action).

ATHENA Annual Meeting Agenda

Page | 2

Monday 12th June

09:30 – 11:30	**Introduction to Matlab**
	Basic Operations with Images in Grayscale Values
11:30 - 12:00	Coffee break
12:00 – 14:00	**Filtering of Digital Images**
	The Frequency Domain

Tuesday 13th June

09:30 – 11:30	**Edge Extraction**
	Introduction to Hyperspectal Image Processing
11:30 - 12:00	Coffee break
12:00 – 14:00	**Spectral Indices**

Wednesday 14th June

09:30 – 11:30	**Dimensionality Reduction**
	Spectral Unmixing
11:30 - 12:00	Coffee break
12:00 – 14:00	**Classification based on Spectral Unmixing Results**

ATHENA Annual Meeting Agenda

Thursday 15th June

09:30 – 11:30	**Band Selection**
11:30 - 12:00	Coffee break
12:00 – 14:00	**Clustering, Morphological Postprocessing**

Page | 3

3. List of Participants

Contracted Researchers as well as graduate and Master students of the Cyprus University of Technology attended the Summer School. The list of participants for each day is given below.

Monday, 12th June 2017

ATHENA Remote Sensing Science Center for Cultural Heritage

Cyprus University of Technology | Consiglio Nazionale delle Ricerche | DLR

H2020-TWINN-2015 - Remote Sensing Science Center for Cultural Heritage - A T H E N A
Topic: Special issues of Optical Remote Sensing
Trainer: Dr. Daniele Cerra (DLR)
Date: Monday, 12th June, 2017
Venue: CUT - Limassol, Cyprus

List of participants

A/A	NAME	INSTITUTION	CONTACT DETAILS	SIGNATURE
1	ANASTASIA YFANTIDOU	Cyprus University of Technology	ai.yfantidou@edu.cut.ac.cy 99739713	[signature]
2	Marina Pekri	Cyprus University of Technology	marinapekri@yahoo.com 99264228	[signature]
3	Maria Prodromou	Cyprus University of Technology	mapro93@gmail.com 97646746	[signature]
4	Georgia Matromalli	Cyprus University of Technology	georgiamatromalli@gmail.com	[signature]
5	Ioannis Georgiades	Cyprus University of Technology	yiannis_georgiades@hotmail.com 99036055	[signature]

1

ATHENA
Remote Sensing Science Center for Cultural Heritage

European Commission | HORIZON 2020

6	ZORPAS Eleftherios	CUT	elzorpas@gmail.com 99-037819	
7	Kyriaki Louca	CUT	Kiriaki-95@gmail.com 99295007	
8	Eirini Pavlou	CUT	ep.pavlou@edu.cut.ac.cy 99264965	
9	Athos Agapiou	CUT	athos.agapiou@cut.ac.cy	
10	Argyro Nisantzi	CUT	argyro.nisantzi@cut.ac.cy	
11	Vasiliki Lysandrou	CUT	vasiliki.lysandrou@cut.ac.cy	
12	Kyriakos Themistocleous	CUT	k.themistocleous@cut.ac.cy	
13	DANIELE CERRA	DLR	DANIELE.CERRA@DLR.DE	
14				
15				
16				

2

Tuesday, 13th June 2017

ATHENA
Remote Sensing Science Center for Cultural Heritage

European Commission | HORIZON 2020

Cyprus University of Technology | Consiglio Nazionale delle Ricerche | DLR

H2020-TWINN-2015 - Remote Sensing Science Center for Cultural Heritage - A T H E N A
Topic: Special issues of Optical Remote Sensing
Trainer: Dr. Daniele Cerra (DLR)
Date: Tuesday, 13th June, 2017
Venue: CUT - Limassol, Cyprus

List of participants

A/A	NAME	INSTITUTION	CONTACT DETAILS	SIGNATURE
1	ANASTASIA YFANTIDOU	CUT	ai.yfantidou@edu.cut.ac.cy 99739713	
2	Maria Prodromou	CUT	mapro93@gmail.com	
3	Marina Petri	CUT	marinapetri@yahoo.com 99264220	
4	DIMITRIS KOUHARTSIOUK	CUT	d.kouhartsiouk@cut.ac.cy	
5	Georgia Mattouralli	CUT	georgiamattouralli@gmail.com	

1

ATHENA Remote Sensing Science Center for Cultural Heritage

6	Ioannis Georgiades	CUT	gioannis_georgiades@hotmail.com 99036355	
7	Athos Agapiou	—	athos.agapiou@cut.ac.cy	
8	Andreas Christofe	CUT	andreas.christofe@cut.ac.cy	
9	Argyro Nisantzi	CUT	argyro.nisantzi@cut.ac.cy	
10	Christiana Papoutsa	CUT	christiana.papoutsa@cut.ac.cy	
11	KYRIACOS THEMISTOCLEOUS	CUT	k.themistocleous@cytanet.com.cy	
12	Vasiliki Lysandrou	CUT	vasiliki.lysandrou@cut.ac.cy	
13	MARIOS TZOUVARAS	CUT	marios.tzouvaras@cut.ac.cy	
14	ZORPAS Elefterios	CUT	elzorpas@gmail.com	
15	Kyriaki Louca	CUT	kiriaki_95@hotmail.com	
16	Eirini Pavlou	CUT	ep_pavlou@edu.cut.ac.cy	

2

ATHENA Remote Sensing Science Center for Cultural Heritage

17	DANIELE CERRA	DLR	DANIELE.CERRA@DLR.DE	
18				
19				
20				

3

Wednesday, 14th June 2017

ATHENA Remote Sensing Science Center for Cultural Heritage — European Commission — HORIZON 2020

Cyprus University of Technology — Consiglio Nazionale delle Ricerche — DLR

H2020-TWINN-2015 - Remote Sensing Science Center for Cultural Heritage - A T H E N A
Topic: Special issues of Optical Remote Sensing
Trainer: Dr. Daniele Cerra (DLR)
Date: Wednesday, 14th June, 2017
Venue: CUT - Limassol, Cyprus

List of participants

A/A	NAME	INSTITUTION	CONTACT DETAILS	SIGNATURE
1	Christiana Papoutsa	CUT	christiana.papoutsa @cut.ac.cy	[signature]
2	Argyro Nisantzi	CUT	argyro.nisantzi@ cut.ac.cy	[signature]
3	Vasiliki Lysandrou	CUT	vasiliki.lysandrou@ cut.ac.cy	[signature]
4	Athina Agapiou	—	athina.agapiou @cut.ac.cy	[signature]
5	Ioannis Georgiades	CUT	gioanni_georgiades@ hotmail.com	[signature]

1

ATHENA Remote Sensing Science Center for Cultural Heritage — European Commission — HORIZON 2020

A/A	NAME	INSTITUTION	CONTACT DETAILS	SIGNATURE
6	Andreas Christofe	CUT	andreas.christofe@ cut.ac.cy	[signature]
7	Zorpas Eleftherios	CUT	elzorpas@gmail.com	[signature]
8	Maria Prodromou	CUT	mapro93@gmail.com	[signature]
9	KYRIACOS THEMISTOCLEOUS	CUT	kt53@cytanet.com.cy	[signature]
10	Marina Pekri	CUT	marinapekri@ yahoo.com	[signature]
11	Anastasia Yfantidou	CUT	ai.yfantidou@edu.cut.ac.cy	[signature]
12	Rodanthi Mamouri	CUT	rodanthi.mamouri@ cut.ac.cy	[signature]
13	MARIOS TZOUVARAS	CUT	marios.tzouvaras @ cut.ac.cy	[signature]
14	DIMITRIS KOUHARTSIOUK	CUT	d.kouhartsiouk@cut.ac.cy	[signature]
15	EIRINI PAVLOU	CUT	ep.pavlou@edu. cut.ac.cy	[signature]
16	Kyriaki Louca	CUT	kyriaki-95@hotmail.com	[signature]

2

ATHENA Remote Sensing Science Center for Cultural Heritage

European Commission HORIZON 2020

17	DANIELE CERRA	DLR	DANIELE.CERRA@DLR.DE	[signature]
18	Georgia Matrovalli	CUT	georgiamatrovalli@gmail.com	[signature]
19				
20				

3

Thursday, 15th June 2017

ATHENA Remote Sensing Science Center for Cultural Heritage

European Commission HORIZON 2020

Cyprus University of Technology | Consiglio Nazionale delle Ricerche | DLR

H2020-TWINN-2015 - Remote Sensing Science Center for Cultural Heritage - A T H E N A
Topic: Special issues of Optical Remote Sensing
Trainer: Dr. Daniele Cerra (DLR)
Date: Thursday, 15th June, 2017
Venue: CUT - Limassol, Cyprus

List of participants

A/A	NAME	INSTITUTION	CONTACT DETAILS	SIGNATURE
1	Argyro Nisanzi	CUT	argyro.nisanzi@cut.ac.cy	[signature]
2	KYRIACOS THEMISTOCLEOUS	CUT	k733@cytanet.com.cy	[signature]
3	Maria Prodromou	CUT	mapro93@gmail.com	[signature]
4	Georgia Matrovalli	CUT	georgiamatrovalli@gmail.com	[signature]
5	Ioannis Georgiades	CUT	gioannis.georgiades@hotmail.com	[signature]

1

ATHENA Remote Sensing Science Center for Cultural Heritage

6	Andreas Christofe	CUT	andreas.christofe@cut.ac.cy	
7	Zovpas Eleftherios	CUT	elzovpas@gmail.com	
8	Dimitris Kouhartsiouk	CUT	d.kouhartsiouk@cut.ac.cy	
9	Kyriaki Louca	CUT	Kiriaki_95@hotmail.com	
10	Vasiliki Lysandrou	CUT	vasiliki.lysandrou@cut.ac.cy	
11	Christiana Papoutsa	CUT	christiana.papoutsa@cut.ac.cy	
12	Anastasia Yfantidou	CUT	ai.yfantidou@edu.cut.ac.cy	
13	Daniele Cerra	DLR	DANIELE.CERRA@DLR.DE	
14				
15				
16				

2

4. Presentations during the summer school

All presentations of the Summer School are given in the Annex of the present Deliverable, in the following order:

4.1 Image Characteristics
4.2 Filtering in Time Domain
4.3 Filtering in Frequency Domain
4.4 Contextual Analysis – Texture, Edges, Invarant Features
4.5 Clustering and Classification
4.6 Introduction to Hyperspectal Image Processing
4.7 Hyperspectral Data - Applications
4.8 Dimensionality Reduction – PCA Tutorial
4.9 Spectral Unmixing
4.10 Tutorial on Band Selection

Additionally, a manual prepared by the trainer relative to Remote Sensing Exercises with Matlab with a special focus on Hyperspectral Image Processing has been disseminated to the trainees and is attached at the end of the Annex.

5.Photographs taken during the 2nd Summer School


ATHENA


Revealing hidden information
ATHENA
REMOTE SENSING
SCIENCE CENTER
FOR CULTURAL HERITAGE

ANNEX

PRESENTATIONS OF THE SUMMER SCHOOL (4.1-4.10) and Manual on Remote Sensing Exercises with Matlab with a special focus on Hyperspectral Image Processing

Remote Sensing Image Processing with Matlab

with a special focus on hyperspectral data analysis

Limassol, Cyprus University of Technology

12-15.06.2017

Daniele Cerra, German Aerospace Center (DLR)

DLR

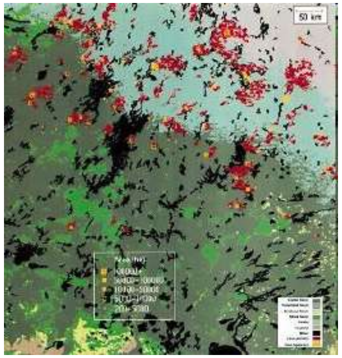
ATHENA
Remote Sensing Science
Center for Cultural Heritage

Image Processing Workflow

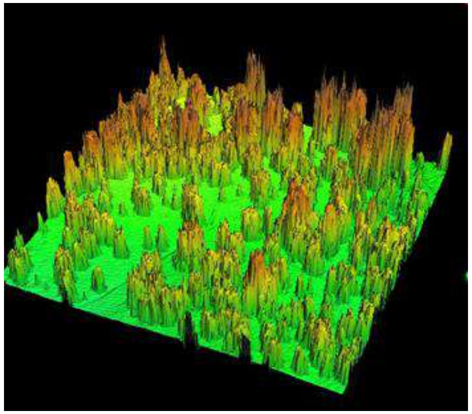
Daniele Cerra, German Aerospace Center (DLR)

Processing of Remotely Sensed Data:
from a Bunch of Numbers to…

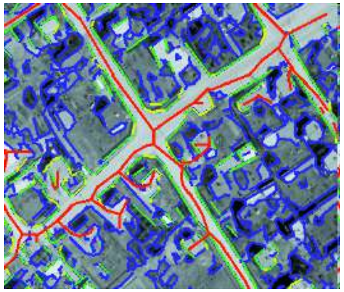
Land Cover Classification



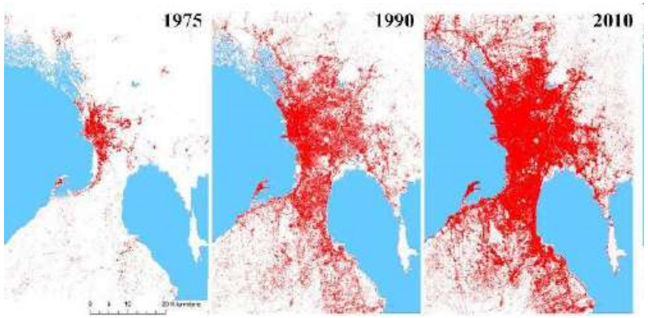
Target Detection
(here: Forest Fires)



3D Surface Modelling



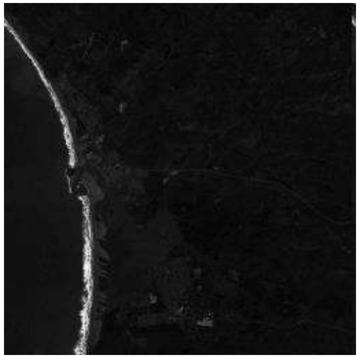
Feature Extraction



Multitemporal Analysis
(here: Urban Sprawl Monitoring)

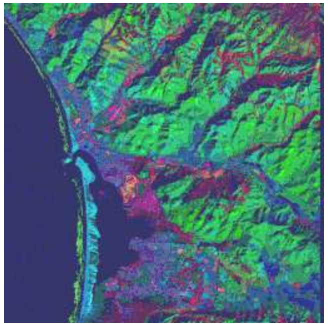
- # Image Acquisition & Correction
  - Raw Data → Raw Image → Image

03	29	38	48
59	96	94	04
05	06	96	97
87	76	75	45



- # Low-level Analysis
  - Image → Image
    - Time domain
    - Frequency domain

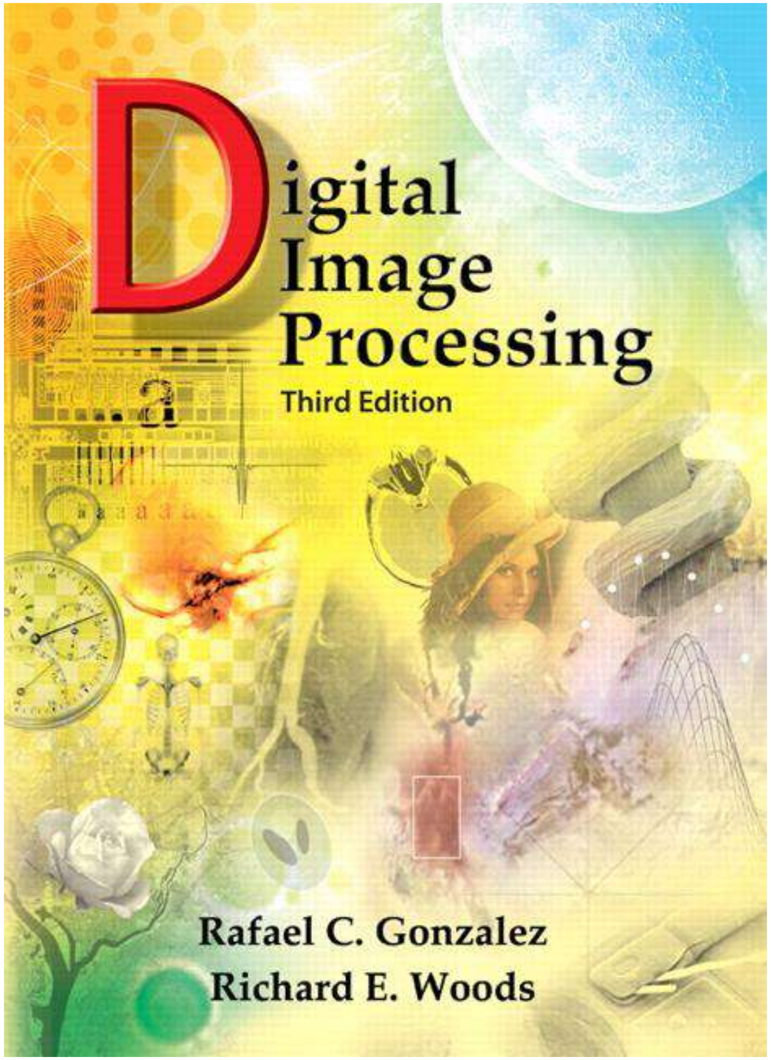
- # Mid-level Analysis
  - Image → Features / Attributes
    - Feature Extraction
    - Clustering / Segmentation



- Beach Bar
- Wave Breakers
- Vegetation1
- Vegetation2
- Golf Course
- Urban Area
- Shadows
- Sea
- Mountains (bright slopes)

- # High-level Analysis
  - Features → Recognition
  - Classification

A nice book!


Digital
Image
Processing
Third Edition
Rafael C. Gonzalez
Richard E. Woods

Summary

- Image Acquisition & Characteristics
  - Spatial, radiometric & spectral resolution
  - Image Correction
- Image enhancement
  - Time Domain
  - Frequency Domain
- Sampling & Aliasing
- Image Features
- Image Clustering
- Image Classification

Summary

- Image Acquisition & Characteristics
  - Spatial, radiometric & spectral resolution
  - Image Correction
- Image enhancement
  - Time Domain
  - Frequency Domain
- Sampling & Aliasing
- Image Features
- Image Clustering
- Image Classification

Image Acquisition

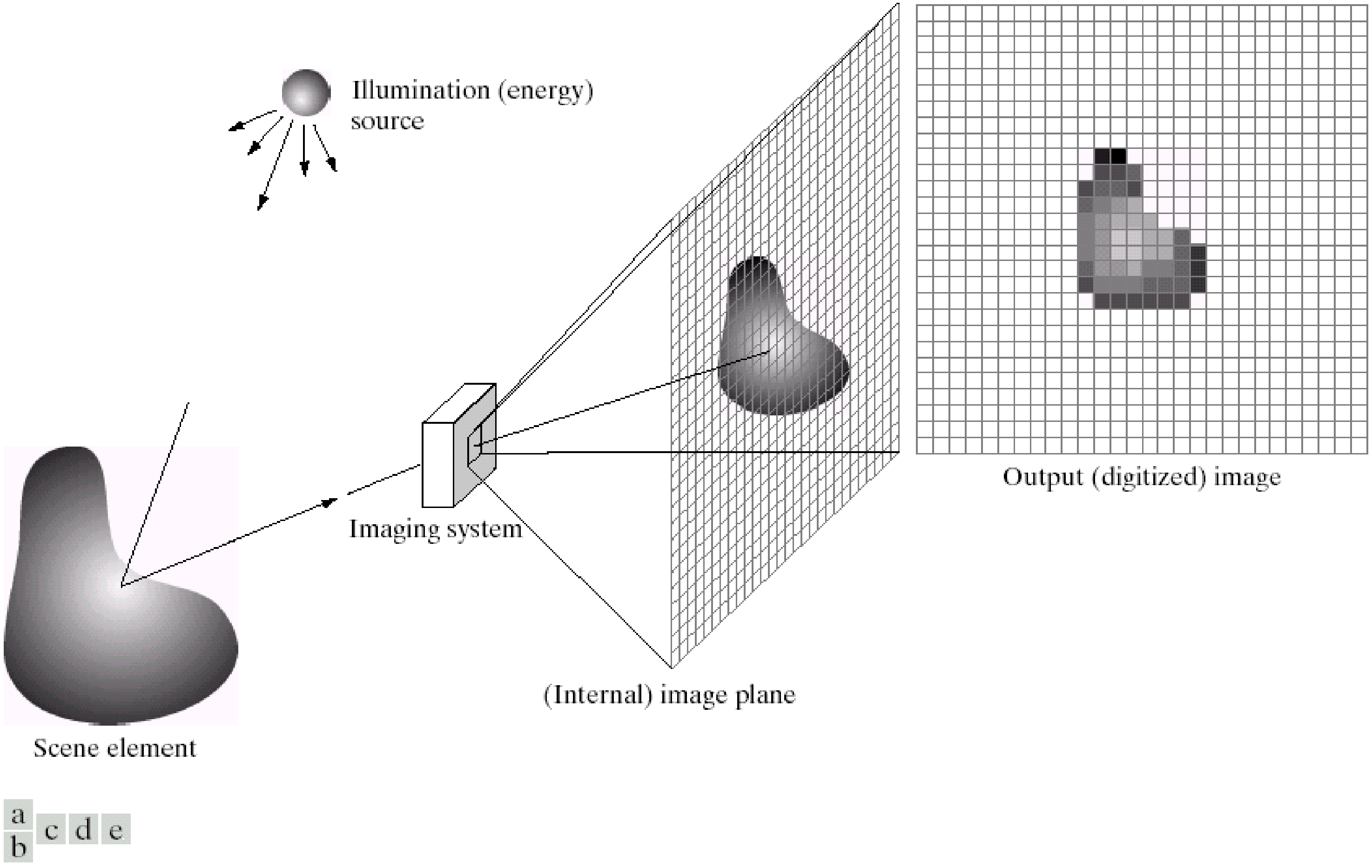
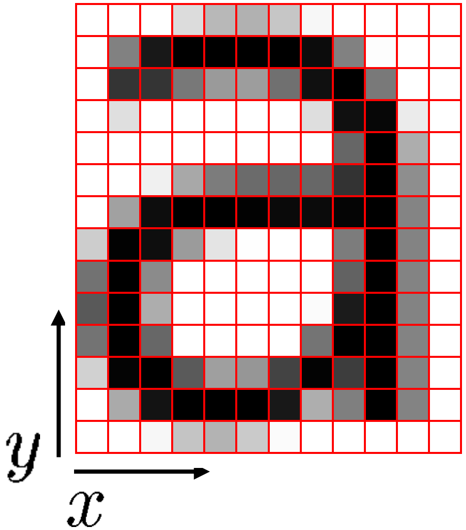


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image representation

- In a digital image, both the coordinates and the image value become discrete quantities
- Images can be represented as 2D arrays (matrices) of integer values: I[i,j] (or I[r,c])
- The term gray level is used to describe monochromatic intensity

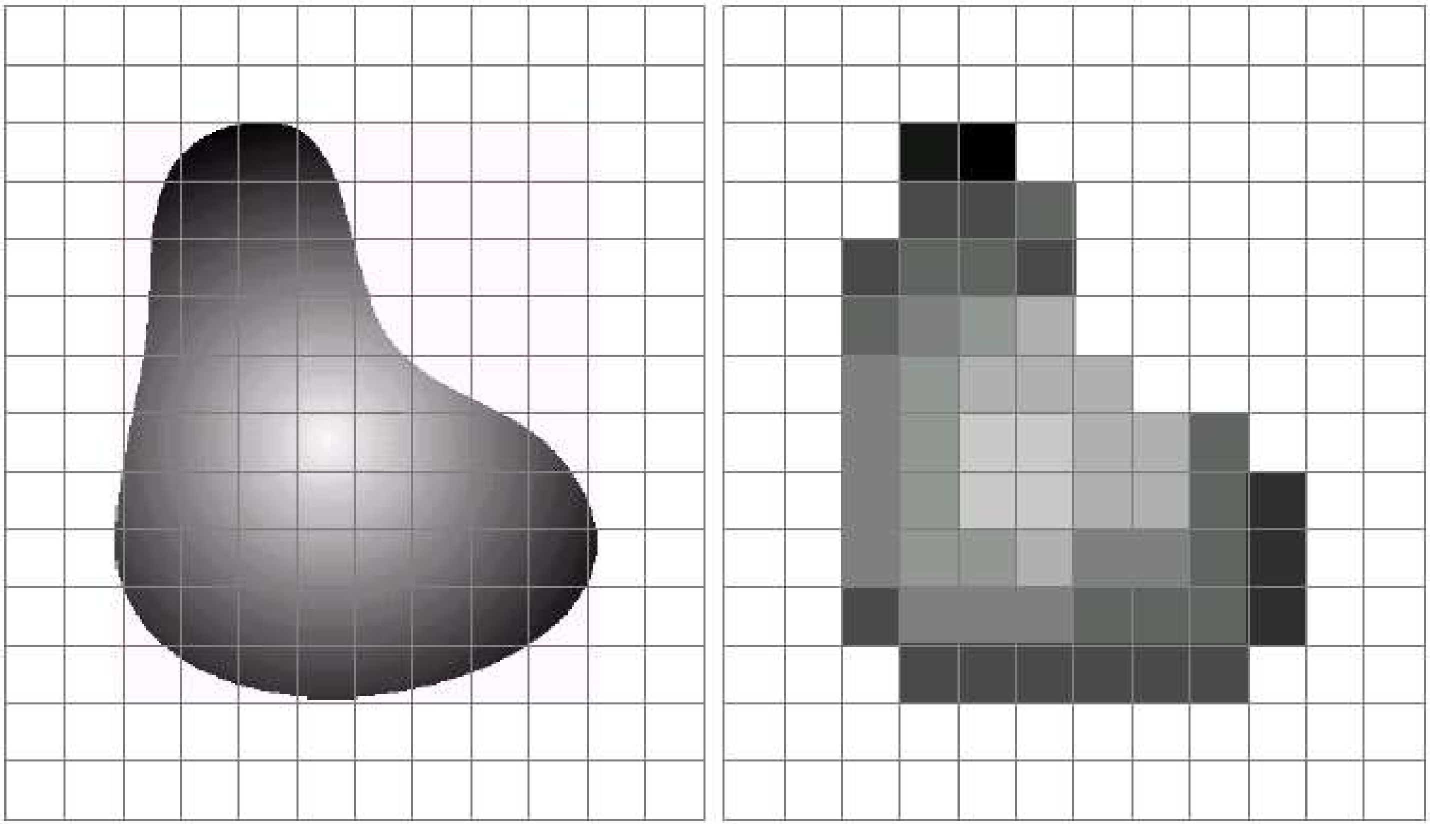
A rasterized form of the letter 'a' magnified 16 times



j →, i ↓

1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Sampling and quantization



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Spatial resolution ← Sampling

- Spatial resolution is the smallest discernible detail in an image
- Ground Sampling Distance is the principal factor determining spatial resolution

1024 x 1024 samples → 30m

512 pixels → 60m

256 pix → 120 m

128 → 240m

64 → 480

32 → 1 km

Ground Sampling Distance (GSD) is not directly proportional to the number of pixels!

These images have been resampled to 1024 x 1024 pixels

1024 x 1024

512 x 512

256 x 256

128 x 128

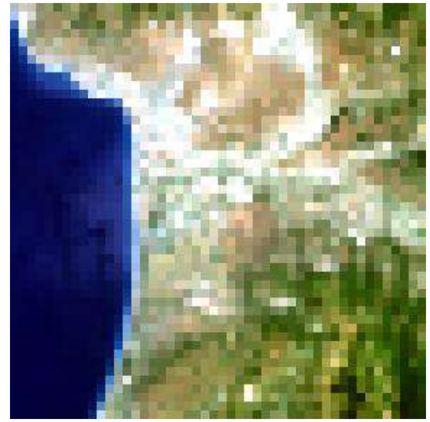
64 x 64

32 x 32

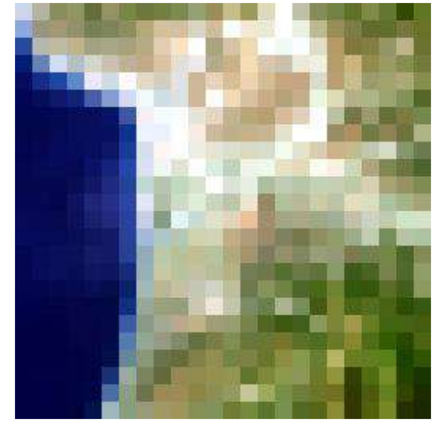
Spatial resolution: Resampling

Resampling without interpolation (nearest-neighbour resampling)

128 x 128



64 x 64



32 x 32

Resampling with interpolation (each pixel is a combination of neighbouring pixels)

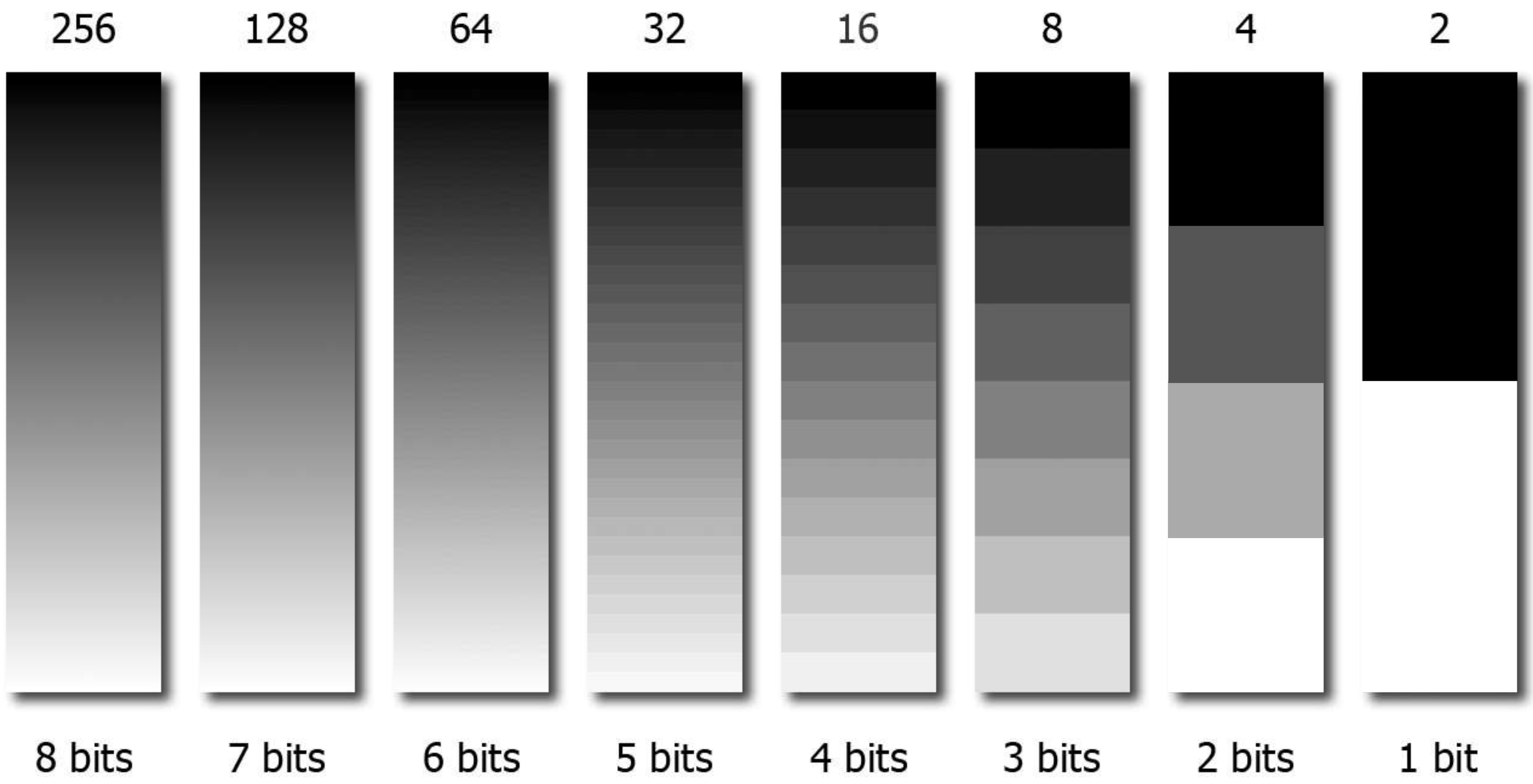
128 x 128

64 x 64

32 x 32

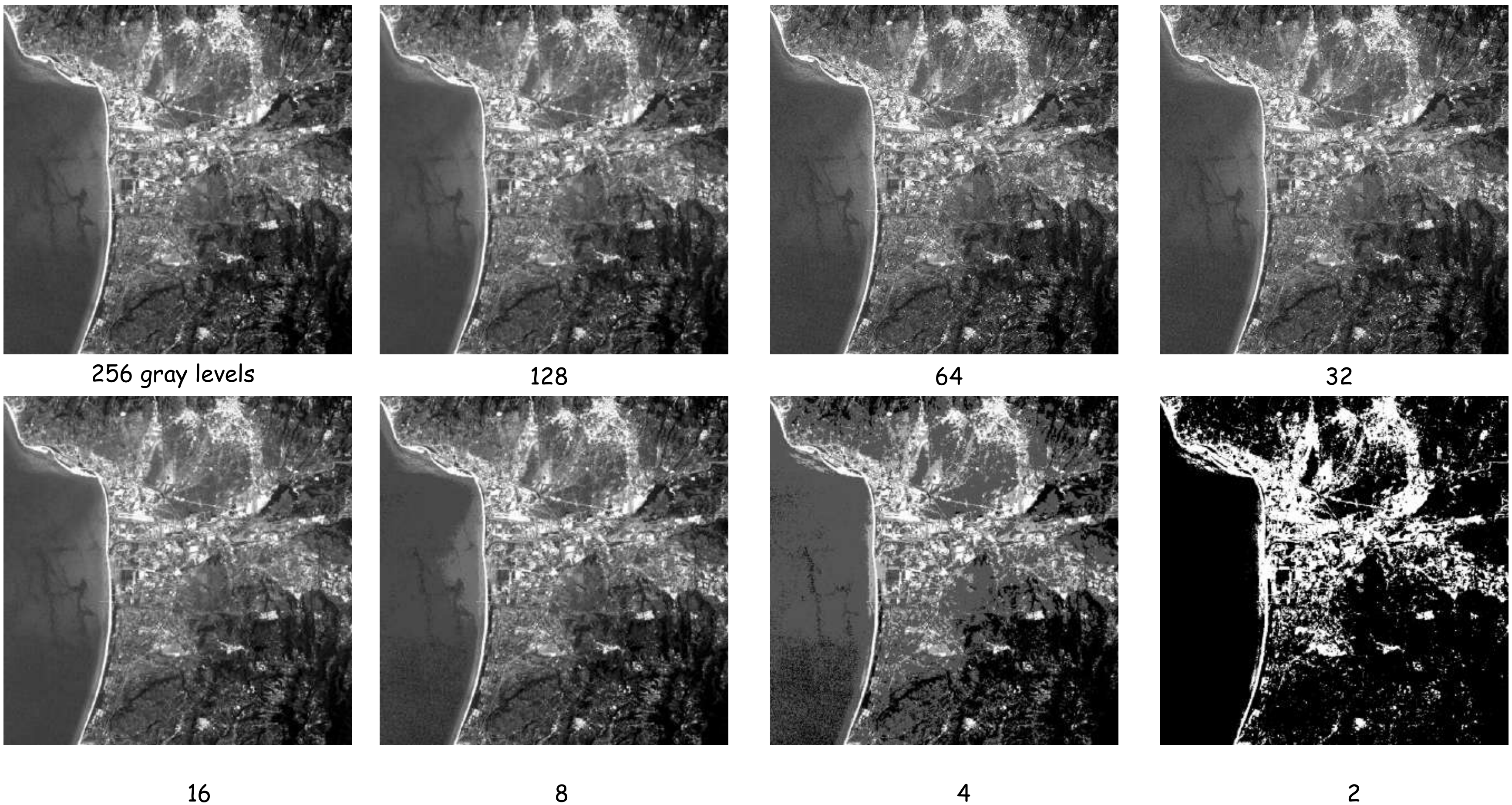
DLR

Radiometric Resolution

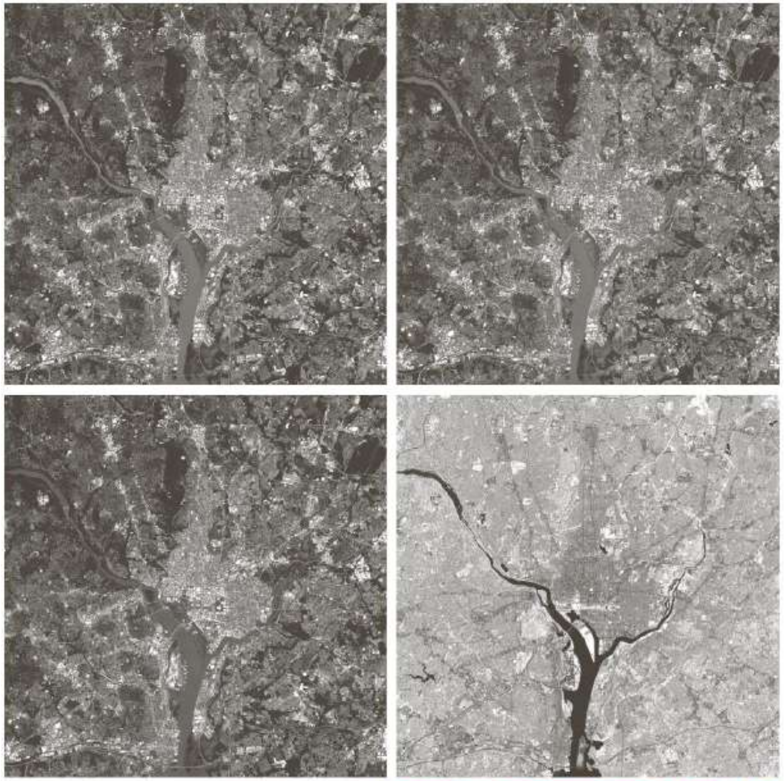


Radiometric resolution ← Quantization

- Radiometric resolution refers to the smallest discernible change in gray level (often power of 2)
- The human eye is inefficient at distinguishing differences in gray levels much beyond the limit of 16 (but to the machine it may make a big difference)

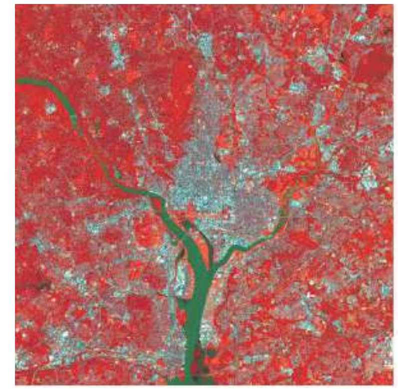


Spectral Resolution & Color Display



R G B

3 2 1



NIR R G

4 3 2

Reflected energy for each pixel in the frequencies Blue, Green, Red & Near Infrared

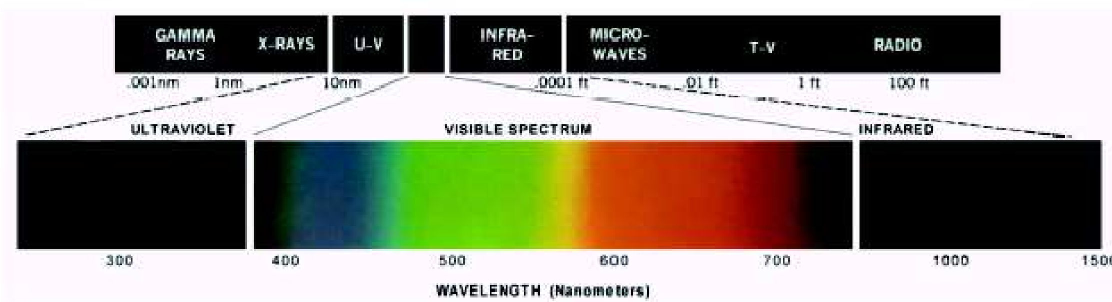


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

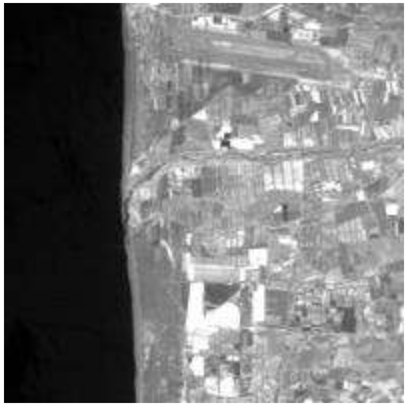
DLR

Remote Sensing: multiband Images (Landsat)

Band 1

Band 2

Band 3



Band 4

Band 5

Band 7

Remote Sensing: multiband Images

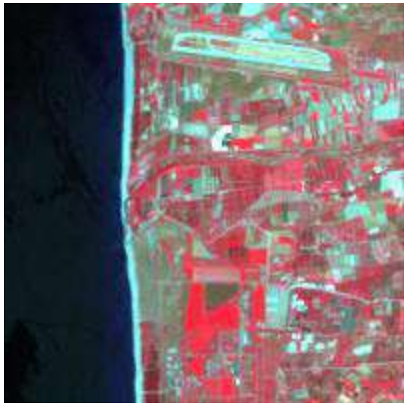
- We can visualize 3 bands at a time: pseudocolor

True Color

R G B

3 2 1

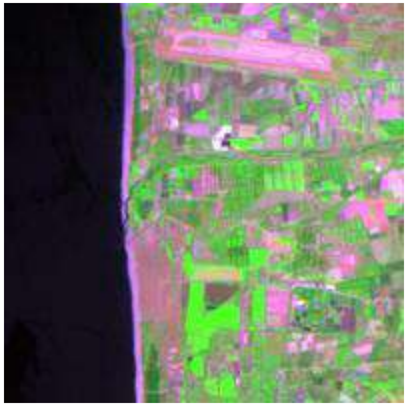
False Color Composite



NIR R G

4 3 2

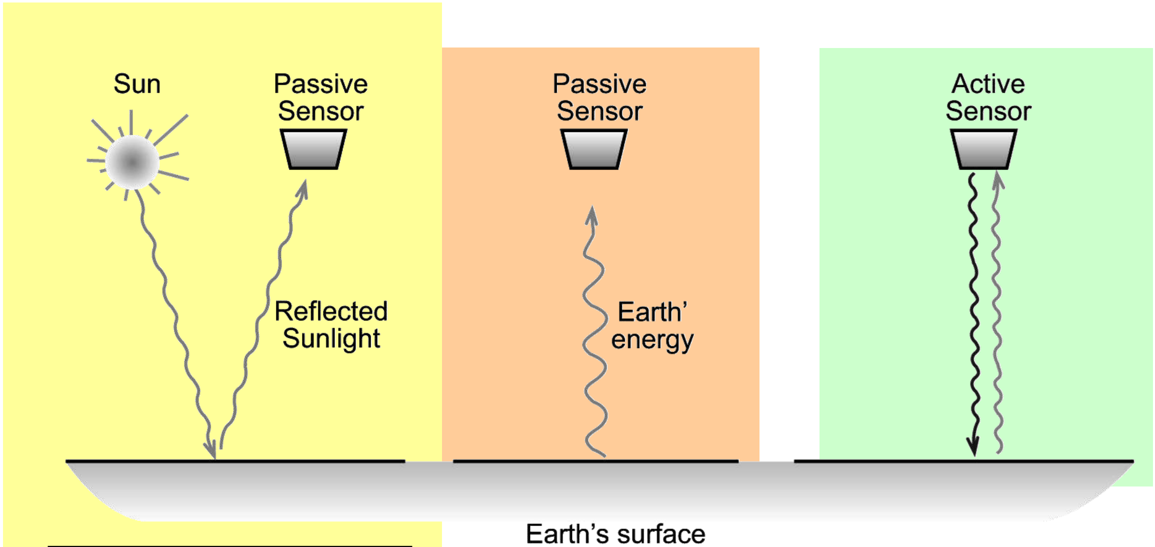
False Color Composite



SWIR NIR R

7 4 3

How many Sensors / kinds of images / datasets in RS?



Optical Passive

- **Panchromatic**
- **Multispectral**
- **Hyperspectral**

Thermal

Radar (SAR)

Lidar

Sonar

DLR CCG

Optical Passive Sensors in Remote Sensing

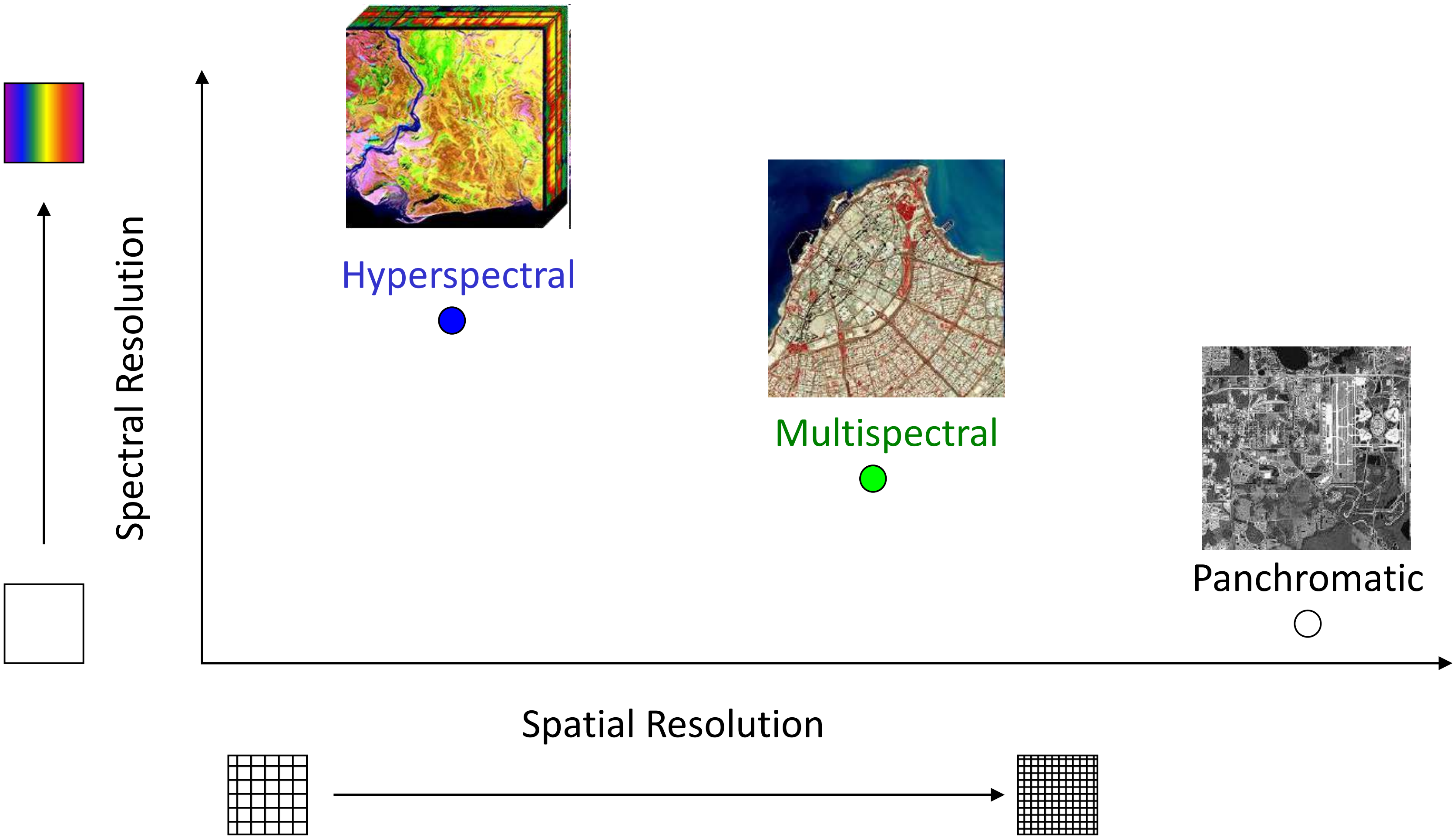
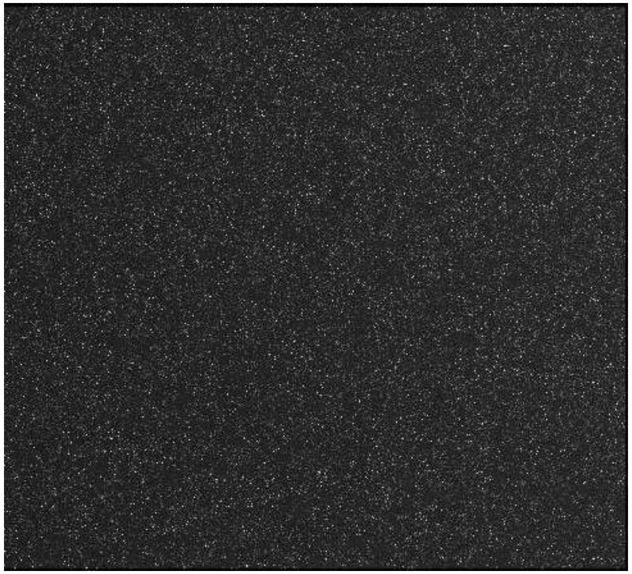


Image Correction

- Raw images are minimally processed images coming directly from the image sensors
- They usually go through several correction steps
- Some important ones are:
  - Dark Signal Correction
  - Non-linearity Correction
  - Odd-even Effect Correction
  - Dead Pixels Flagging

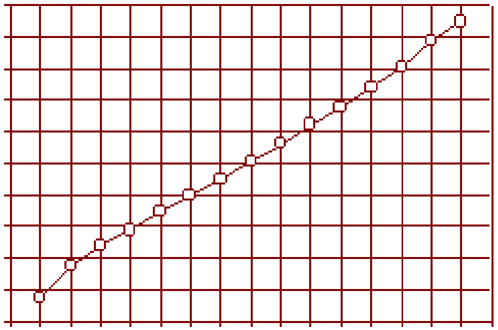
Dark Signal Correction

- Electronic interferences make the recorded signal (a bit) different than it really is
- Dark Signal Measurements
  - Shutter Method
    - Before every take an acquisition is made with the shutter closed. The resulting signal is the „dark current“
  - Deep Space Looking
    - Measures thermal radiation that can affect Dark Signal measurements
  - Dark pixels of the SWIR detector
    - Dark signal depends on the stability of the supplied voltage
    - During image special pixels which stay dark are used as reference



Dark Signal Measurement

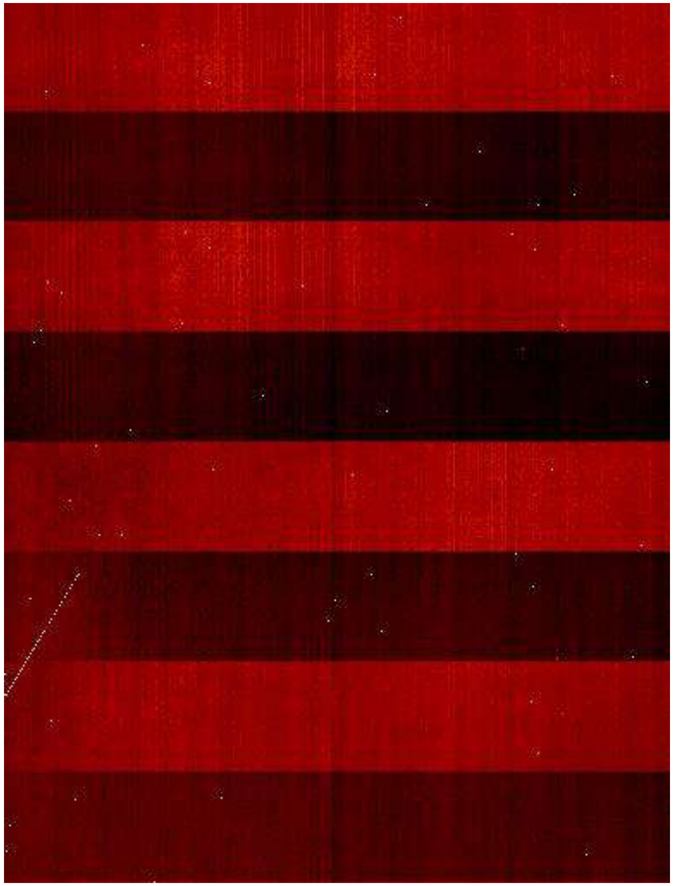
Non-linearity Correction



- The response of a detector as a function of integration time is not linear
- The (non-linear) pixel response is measured at different exposition times and a correction is estimated as a linear function
- During the process dark signal has to be taken into account

Odd-Even Effect

- The odd-even column effect consists of variations of the signal between the columns of the array
  - It is due to differences between sensor arrays
- It is easy to correct
  - Check the difference between the average values of a given column and its neighbouring columns



Raw Image with Odd-even Effect

Dead Pixel Map

- A list of pixels which readings do not have any meaning
- They are declared as „dead“ and ignored (set to 0)
- Different kinds of dead pixels:
  - No response
  - Very large output (hot pixel), saturates easily
  - Flickering pixel (constantly changing output)
  - Constant output

Image Correction

- Once our raw data are corrected, the image is formed and usually undergoes other correction steps….
  - Atmospheric Correction (more about it later - Hyperspectral)
  - Geometric Correction / Orthorectification (more about it later- SIFT)

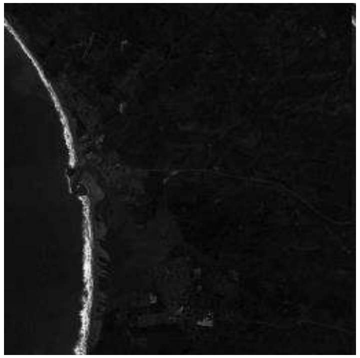
Figure1: Forbidden City (Origin)
Data Source: Space Imaging Co.

Figure 2: Forbidden City
(Relative Geometrically Corrected)

Image by C. Wu

- Image Acquisition & Correction
  - Raw Data→ Raw Image → Image
- Low-level Analysis
  - **Image → Image**
    - **Time domain**
    - Frequency domain
- Mid-level Analysis
  - Image → Features / Attributes
    - Feature Extraction
    - Clustering / Segmentation
- High-level Analysis
  - Features → Recognition

03	29	38	48
59	96	94	04
05	06	96	97
87	76	75	45

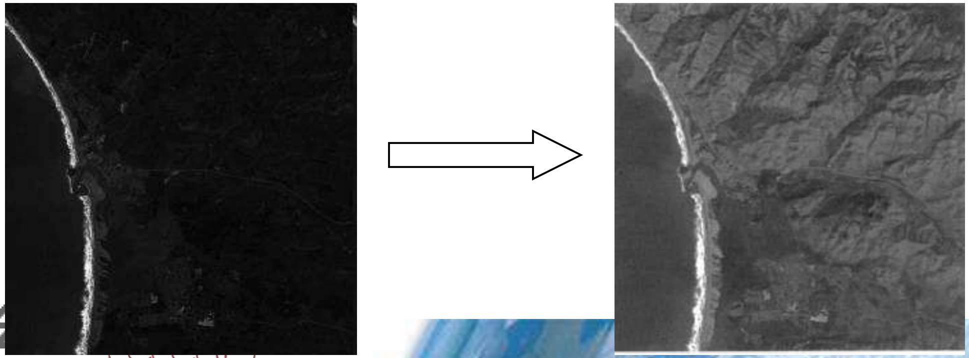


Summary

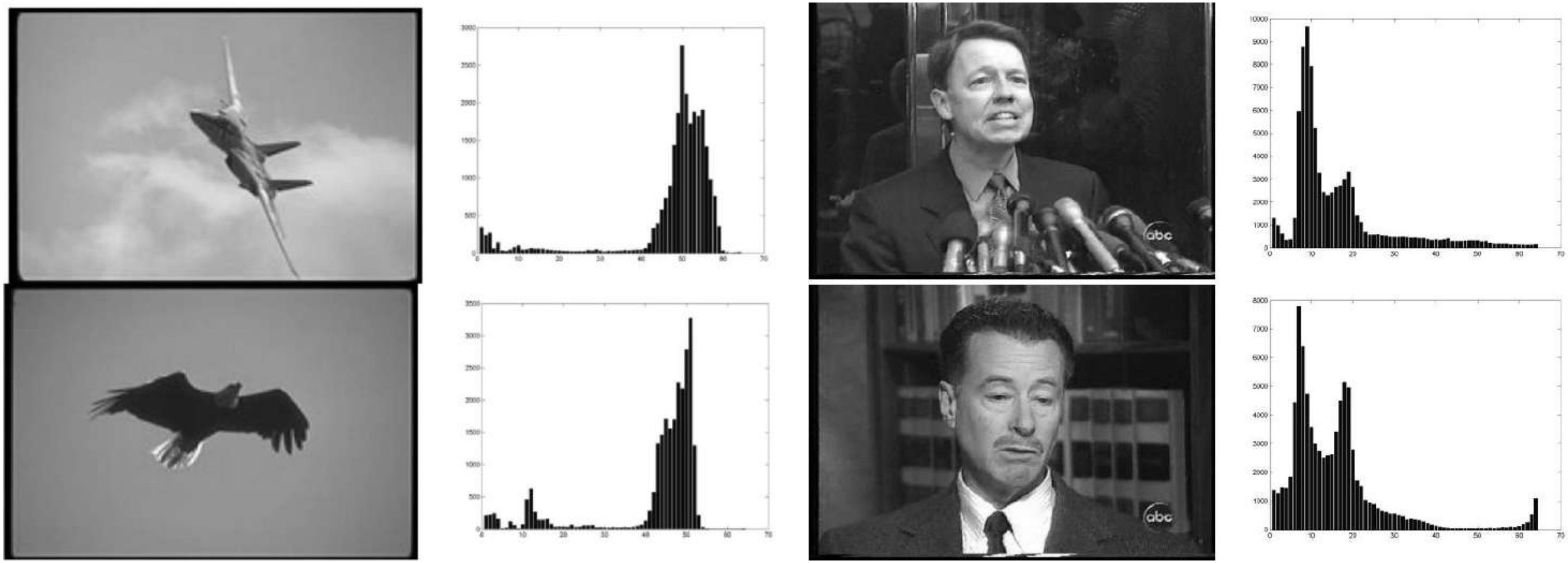
- Image Acquisition & Characteristics
  - Spatial, radiometric & spectral resolution
  - Image Correction
- Image enhancement
  - Time Domain
    - Global Techniques: Histogram Stretch
    - Local Techniques: Moving Window Transform
  - Frequency Domain
- Sampling & Aliasing
- Image Features
- Image Clustering
- Image Classification

Image enhancement

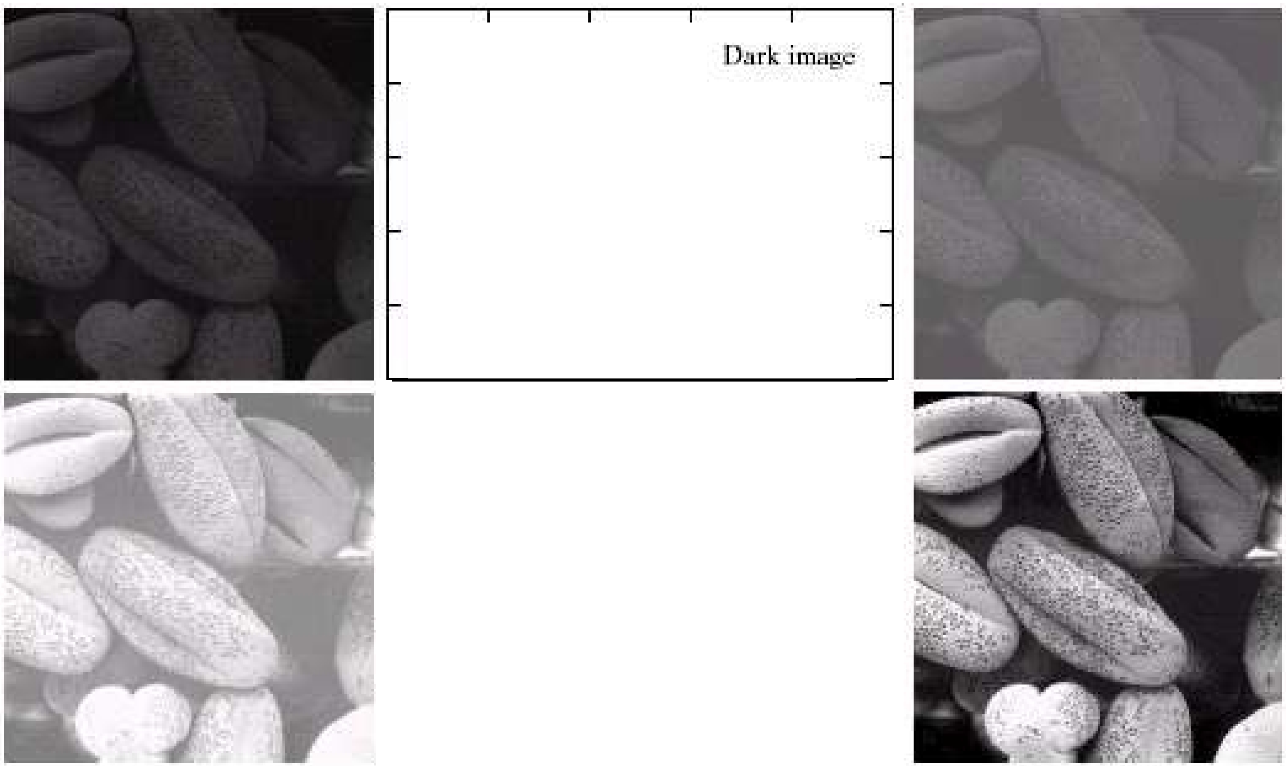
- *Enhance*: to make greater (as in value, desirability, or attractiveness)
- The principal objective of enhancement is to process an image so that the result is more suitable than the original for a *specific* application
- Enhancement is subjective!
    - A good technique for a given application is not valid for another one



Sample Histograms, Natural Images

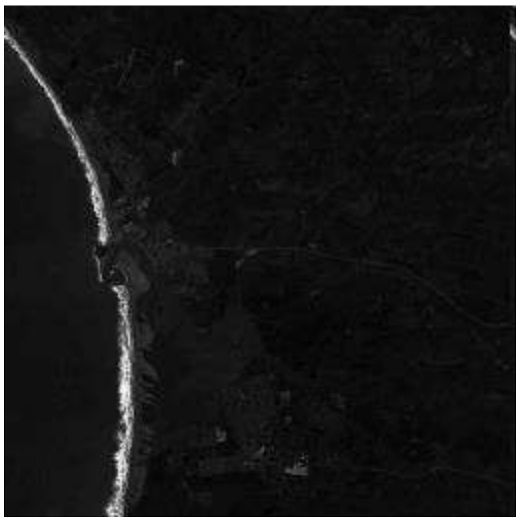


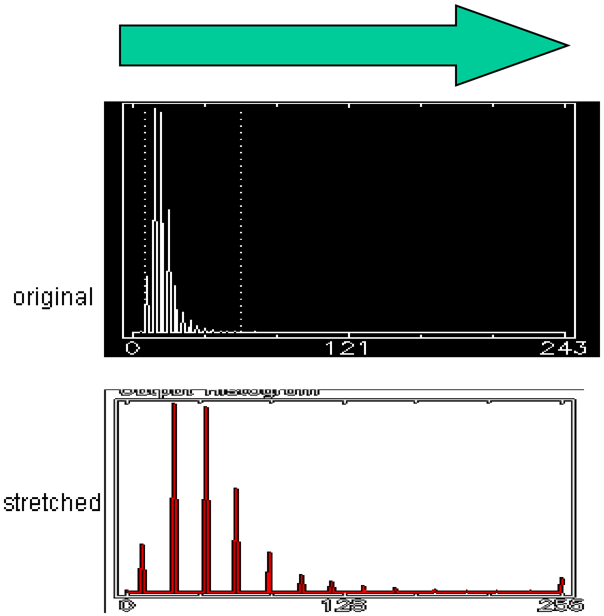
Histogram processing

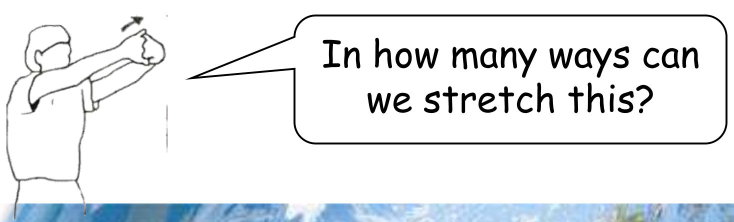


How do you expect the histograms for these pictures?

Histogram Stretching



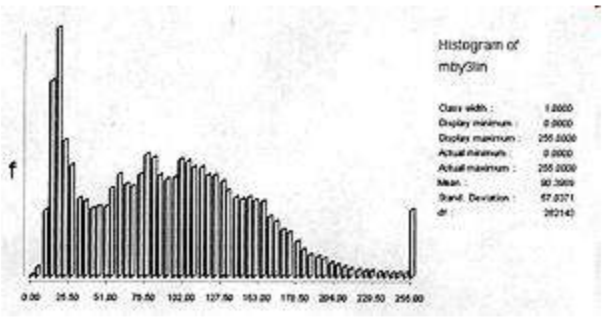




Histogram Stretching

Selective Linear Stretch

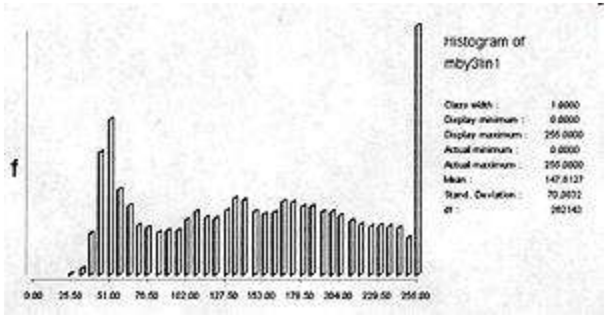
- We take Digital Numbers between 5 and 65
- We expand these from 0 to 255
- All values < 5 are set to 0
- All values > 65 are set to 255
- All values in between are stretched proportionally

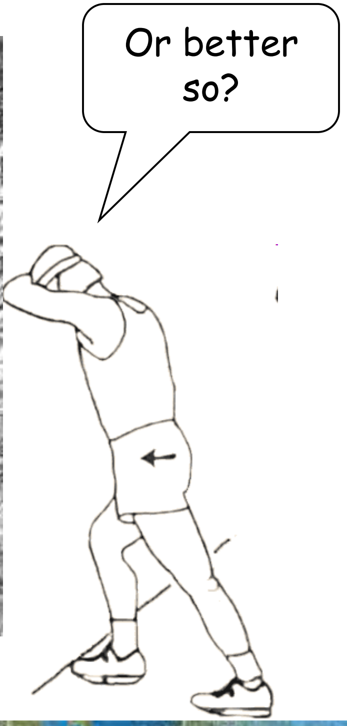


Histogram Stretching

Selective Linear Stretch, let us try to get rid of these dark areas!

- We take now the DNs between 0 and 45
- We expand these from 0 to 255

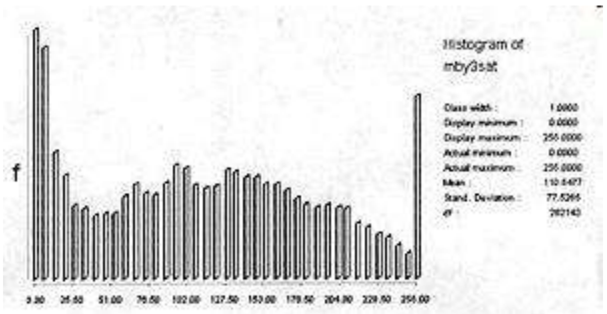




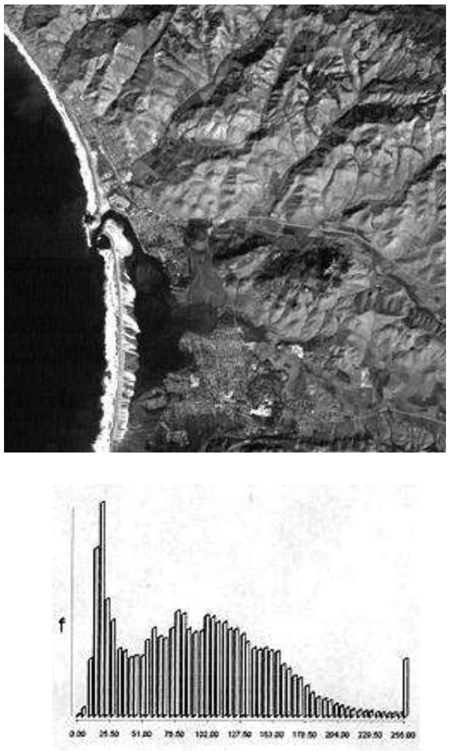
Histogram Stretching

Linear-with-Saturation stretch

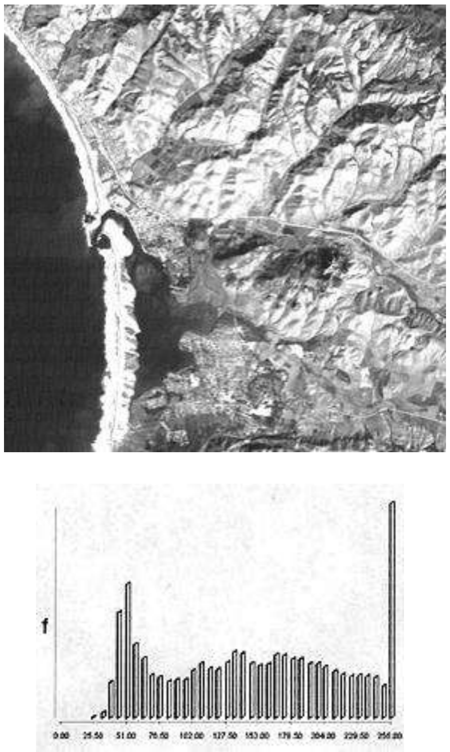
We assign 5% of pixels at each end (tail) of the histogram to single values, and stretch the values in between



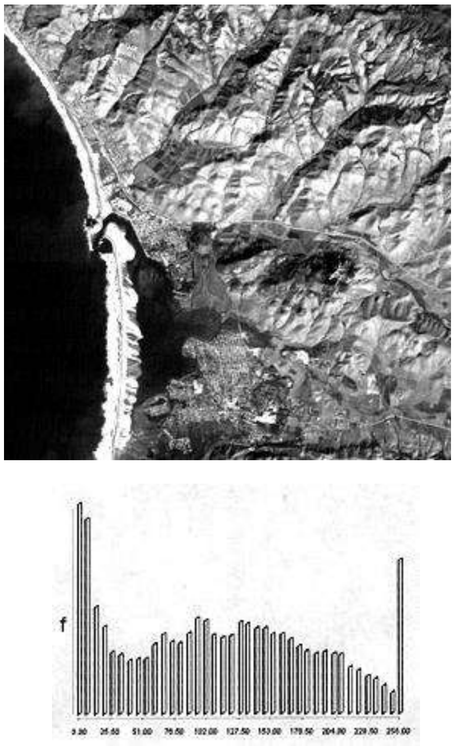
Histogram Stretching: Comparison



Selective
Stretch I



Selective
Stretch II

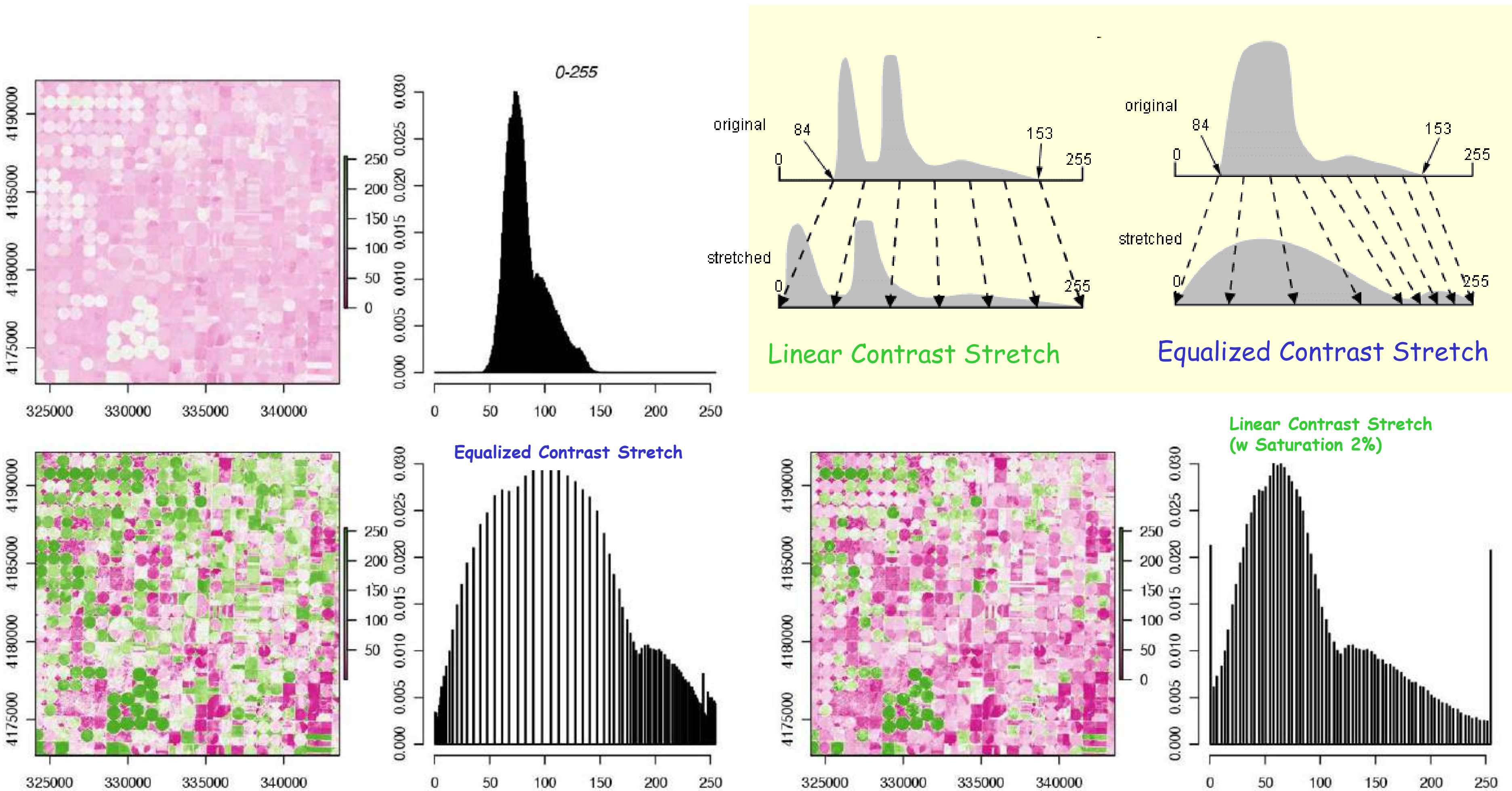


Automatic
Linear-with-
Saturation

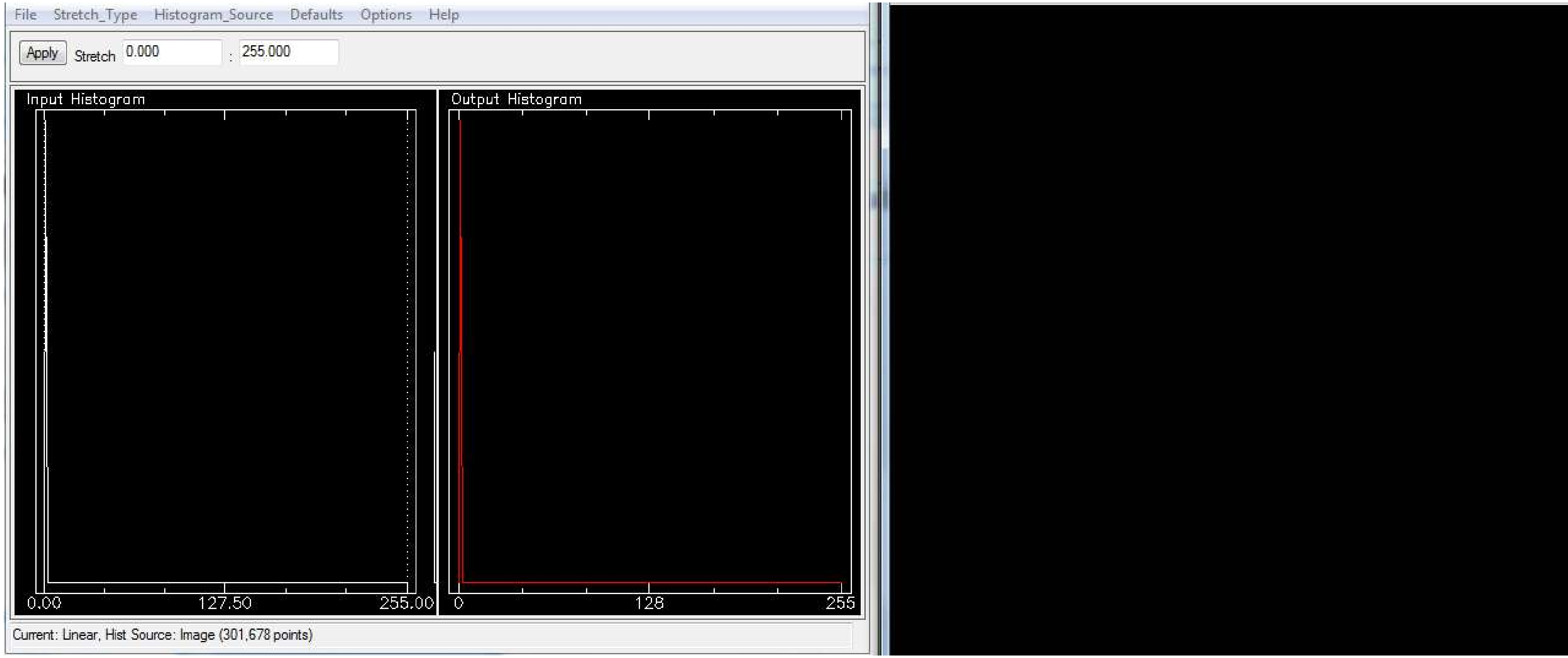
Linear Histogram Stretch

- All 4 images are mapped to a similar output image by applying the same histogram stretch function

Histogram Equalization vs. Linear Stretch



Histogram Equalization vs. Linear Stretch

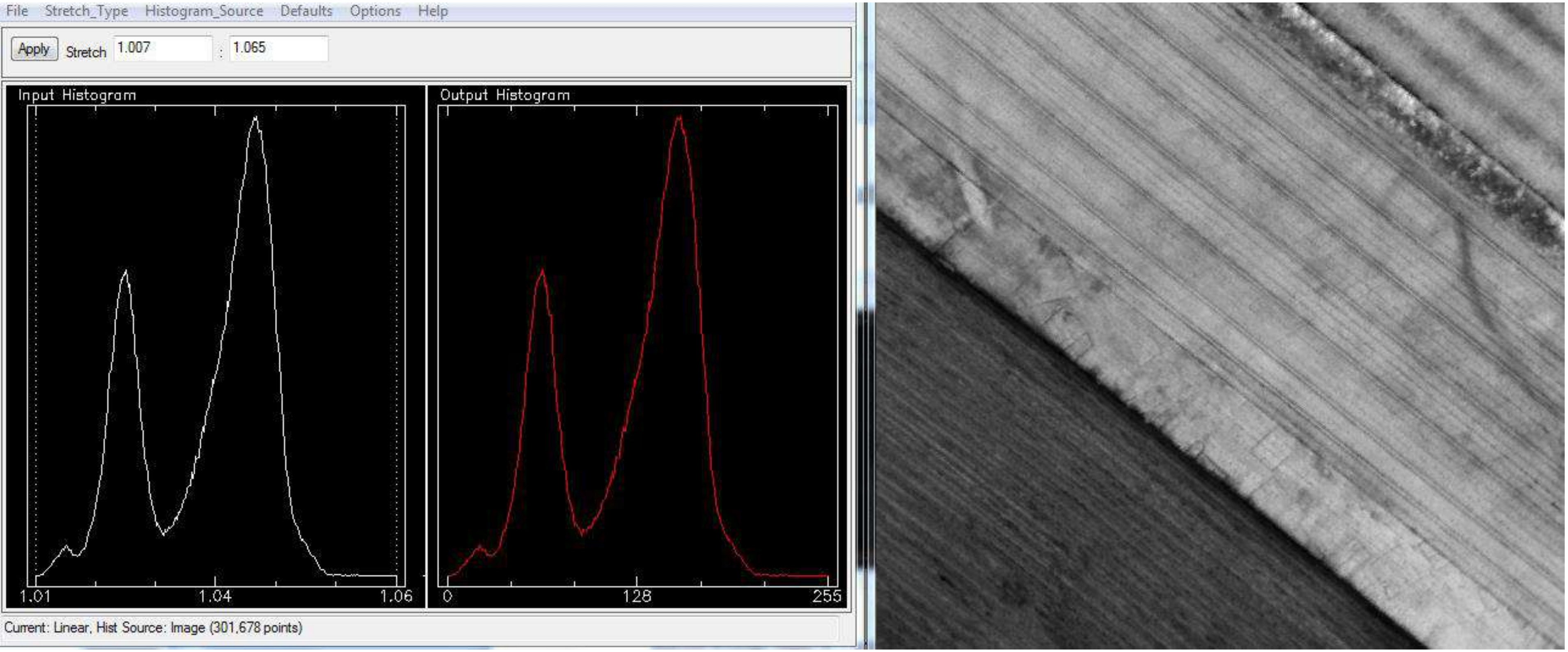


Original histogram Stretched hist. Image

Buried Roman ruins in Carnuntum, Austria, stretch 0-255

Dataset courtesy of prof. M. Doneus, University of Wien

Histogram Equalization vs. Linear Stretch

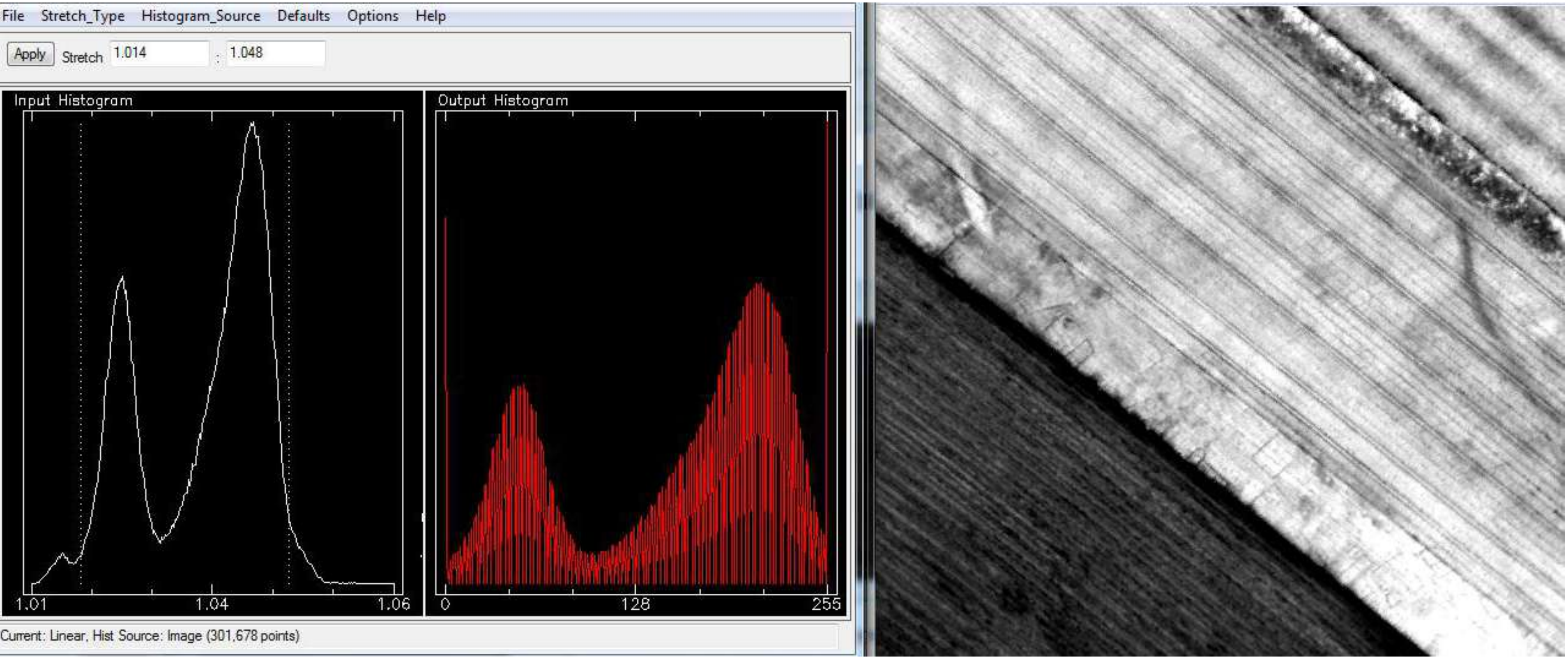


Original histogram Stretched hist. Image

Buried Roman ruins in Carnuntum, Austria, linear stretch

DLR CCG

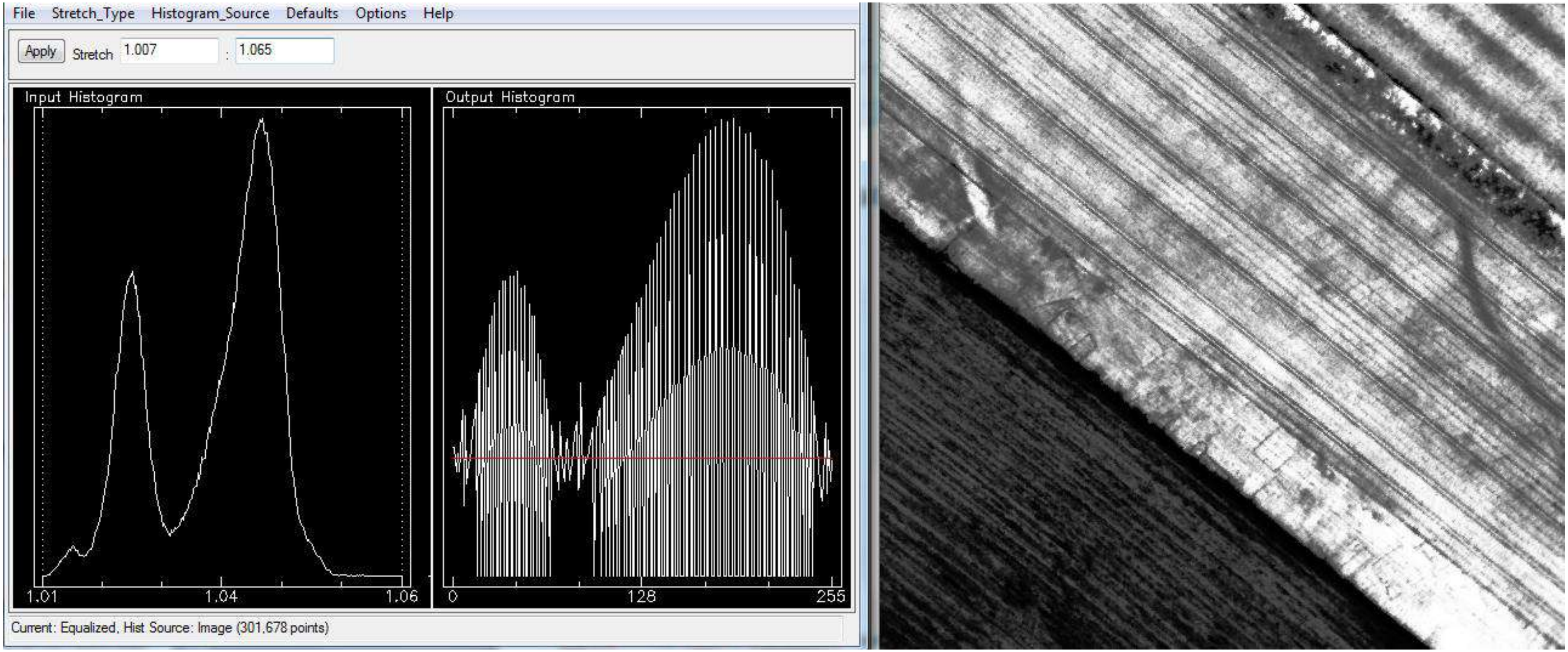
Histogram Equalization vs. Linear Stretch



Original histogram Stretched hist. Image

Buried Roman ruins in Carnuntum, Austria, linear stretch with 2% saturation

Histogram Equalization vs. Linear Stretch



Original histogram Stretched hist. Image

Buried Roman ruins in Carnuntum, Austria, histogram equalization

Summary

- Image Acquisition
  - Spatial, radiometric & spectral resolution
  - Image Correction
- Image enhancement
  - Time Domain
    - Global Techniques: Histogram Stretch
    - Local Techniques: Moving Window Transform
  - Frequency Domain
- Sampling & Aliasing
- Image Features
- Image Clustering
- Image Classification

Image Filtering in Time Domain

Daniele Cerra, German Aerospace Center (DLR)

DLR

Histogram Stretching : Multiband - Recap

Original RGB image

Histogram equalization of each individual band/channel

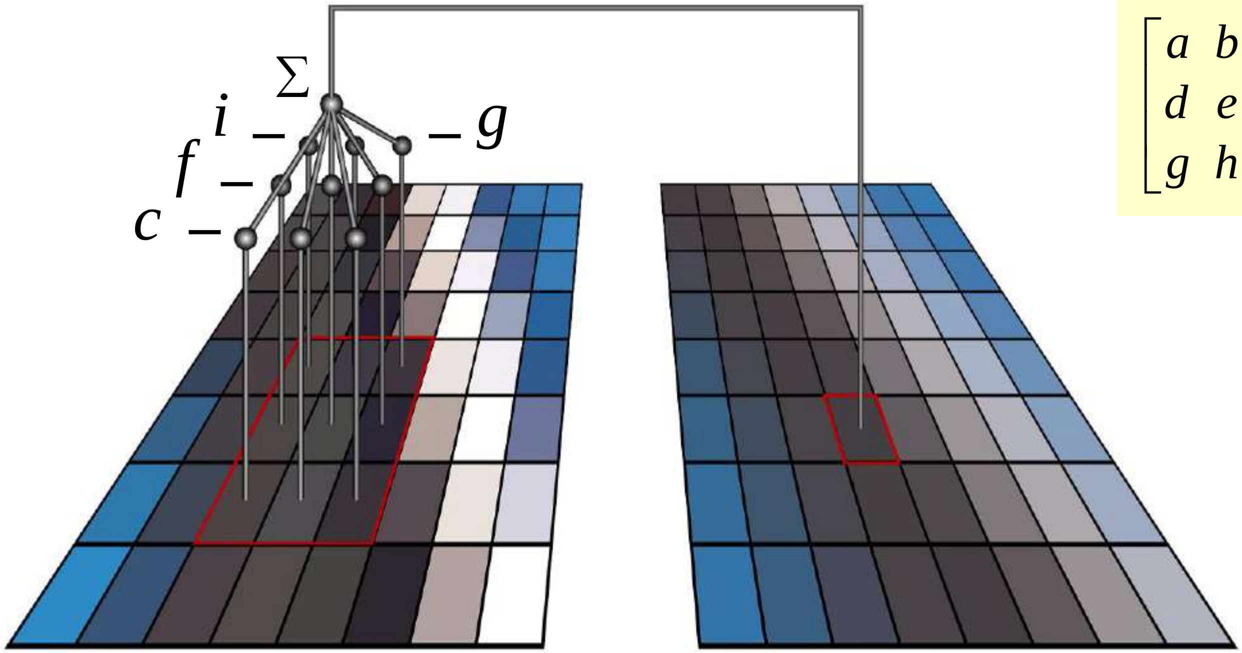
Histogram linear with saturation stretch at 2% (from each individual band/channel)

Summary

- Image Acquisition & Characteristics
    - Spatial, radiometric & spectral resolution
    - Image Correction
- Image enhancement
    - Time Domain
        - Global Techniques: Histogram Stretch
        - Local Techniques: Moving Window Transform
    - Frequency Domain
- Image Features
- Image Clustering
- Image Classification

Local Techniques:

Convolution by Moving Window

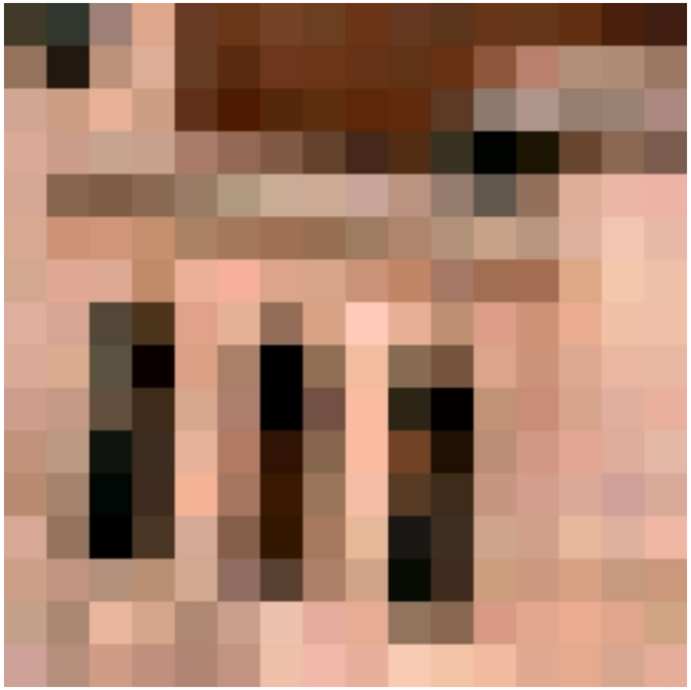


$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

R. A. Peters

Low-pass Filter

Moving Window Transform: Example

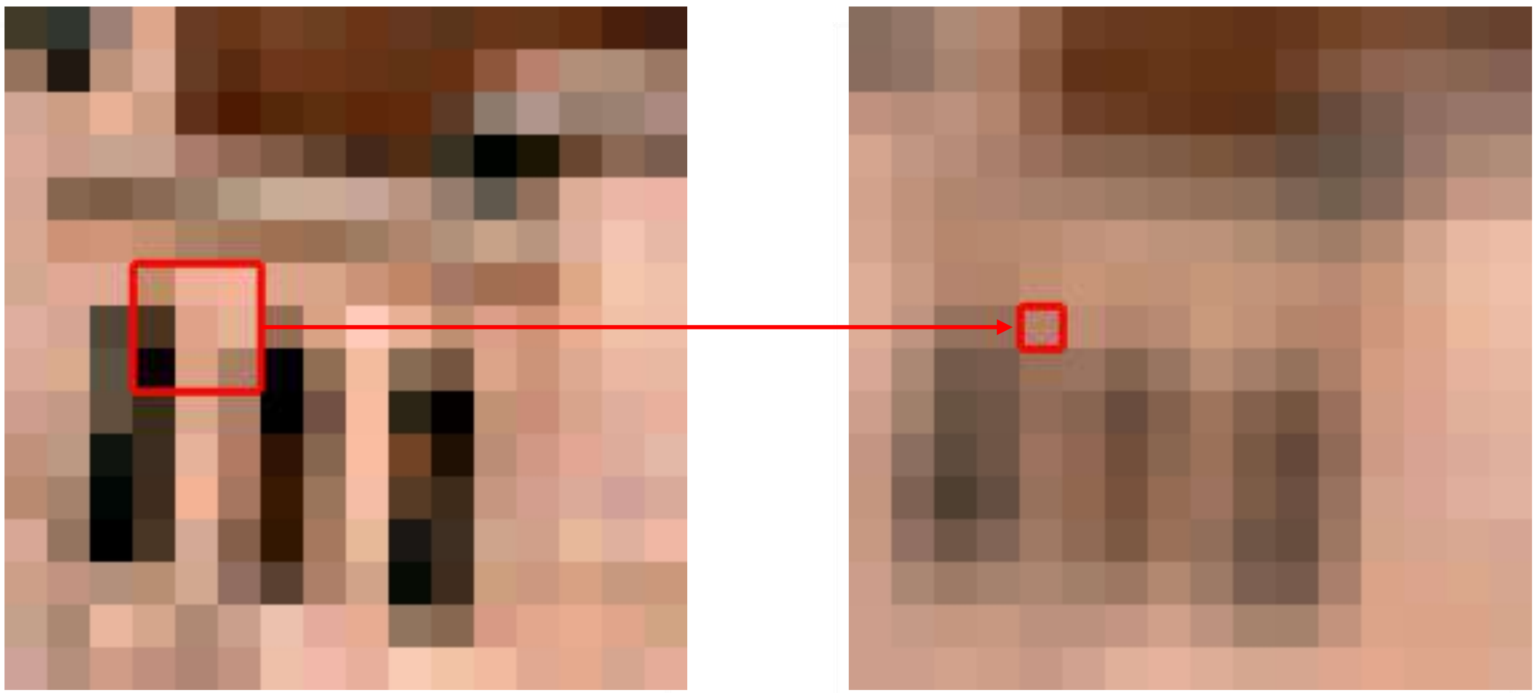


original

3x3 average

Low-pass Filter

Moving Window Transform: Example

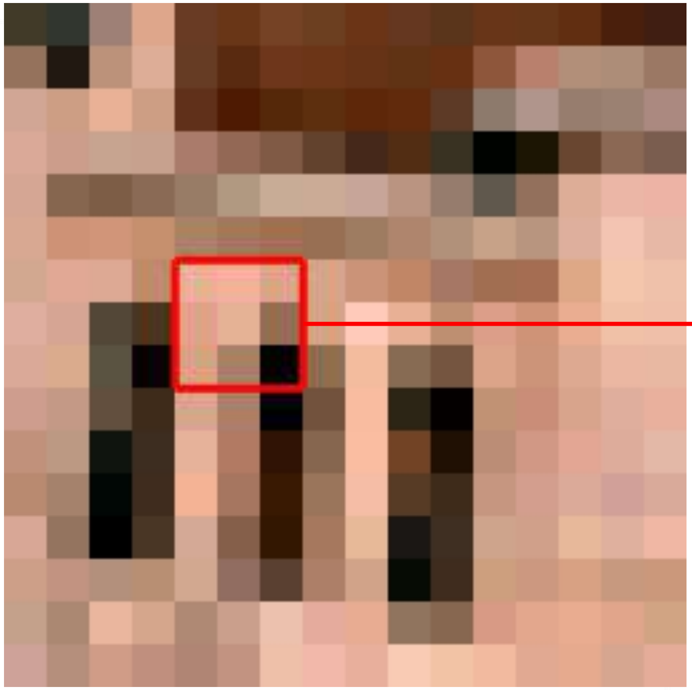


original

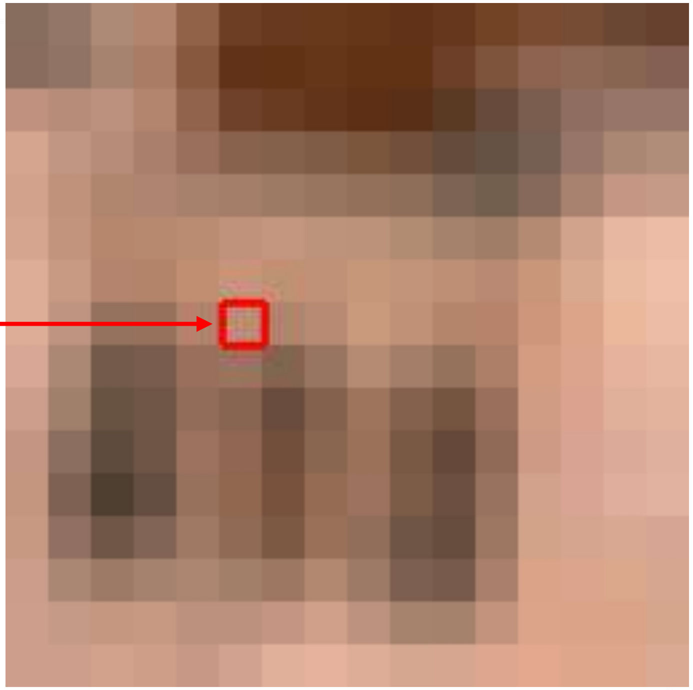
3x3 average

Low-pass Filter

Moving Window Transform: Example



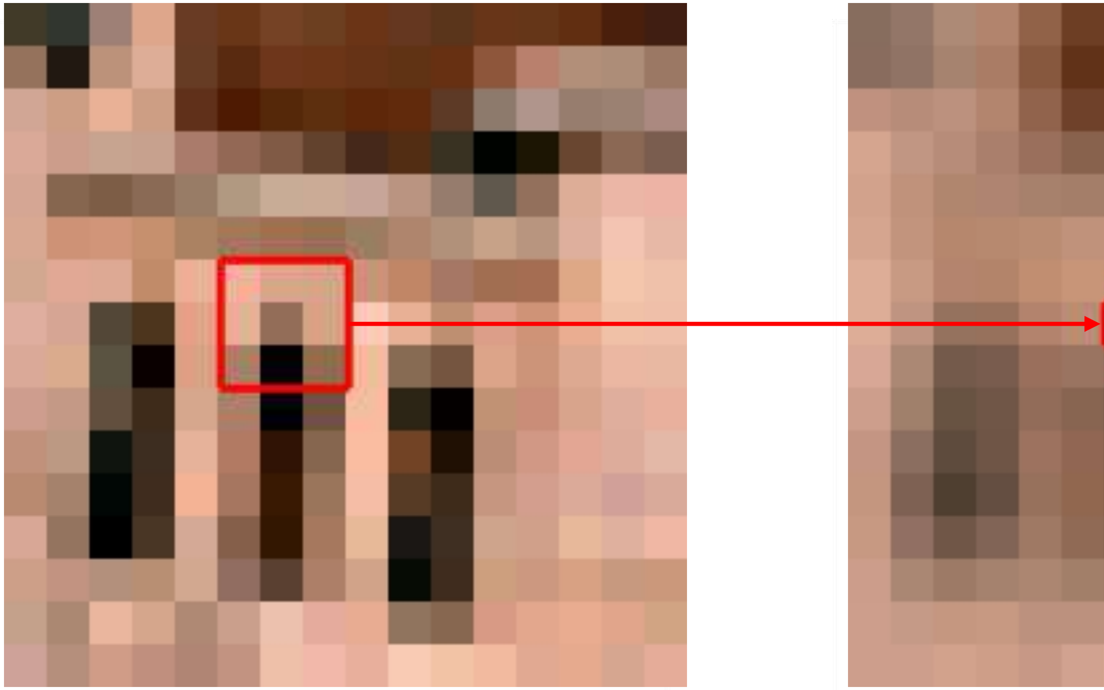
original



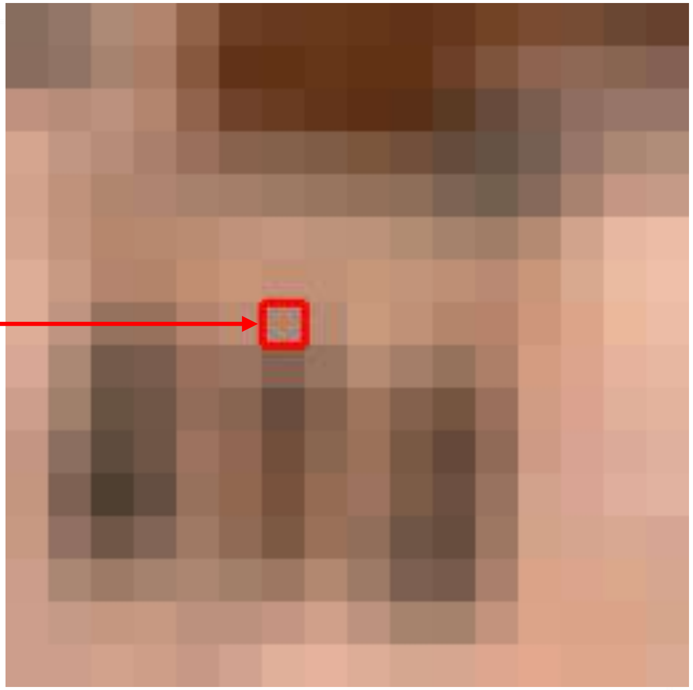
3x3 average

Low-pass Filter

Moving Window Transform: Example



original



3x3 average

Blurring = Arithmetic Mean Filter

$$f(x,y)=\frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t)$$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Mean of a window Sxy of size m x n centered in (x, y) ↔ Output of a m x n lowpass (blurring) filter, $m = n = 3$

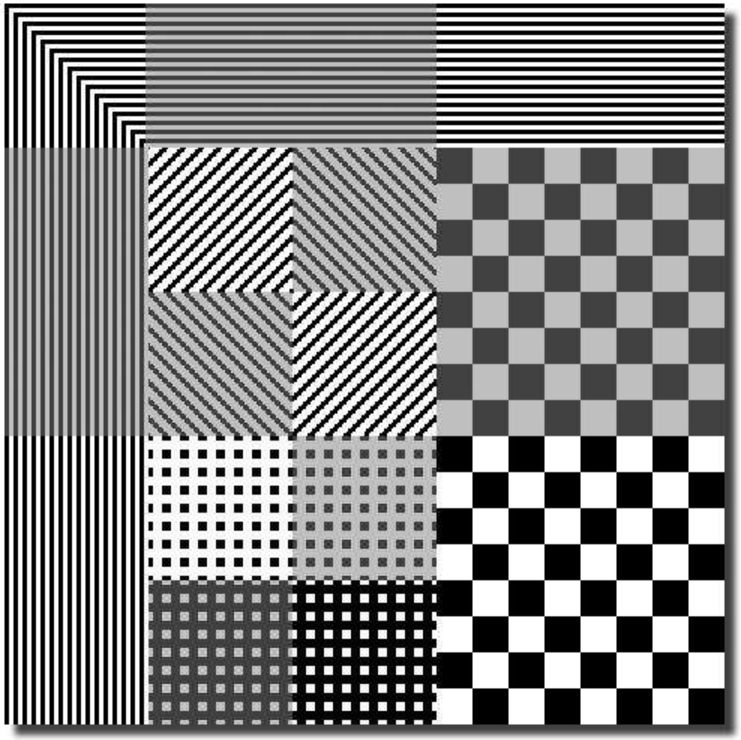
1	3	2
2	4	2
1	4	2

→ 2.333

1. Removes noise to some extent
2. Makes image smoother

1. Destroys details
2. Affects edges
3. Blurring

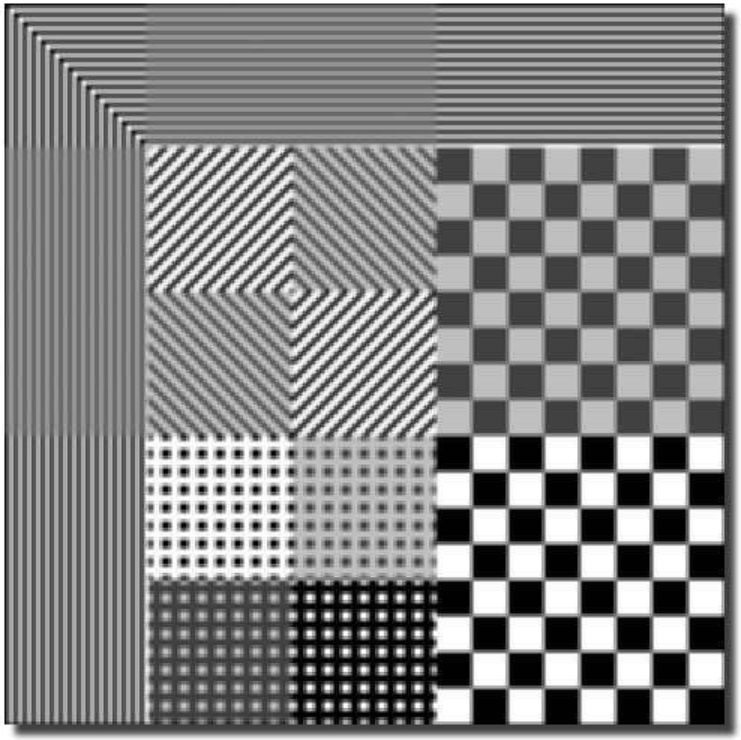
Convolution Examples: Original Images



$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

Low-pass Filter

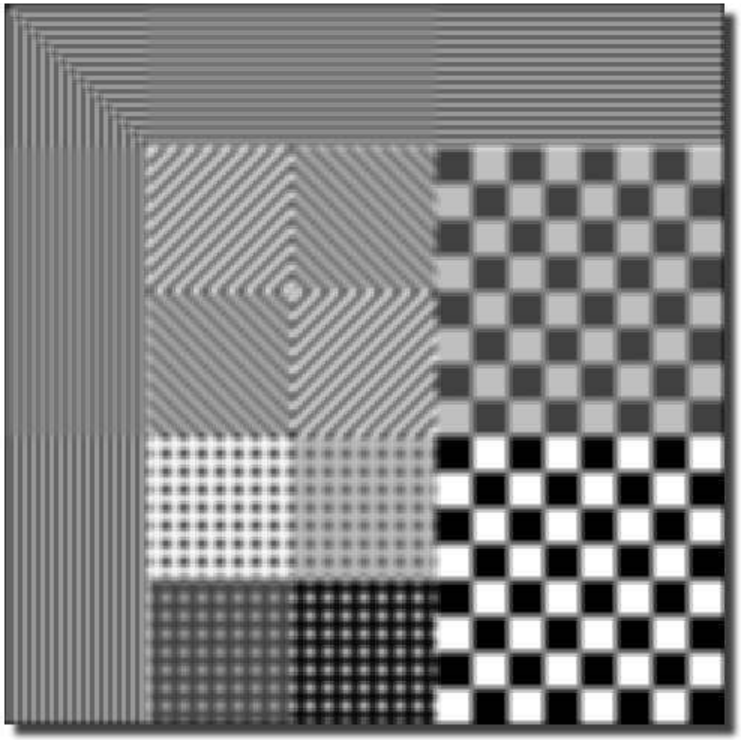
Convolution Examples: 3×3 Blur



$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

Low-pass Filter

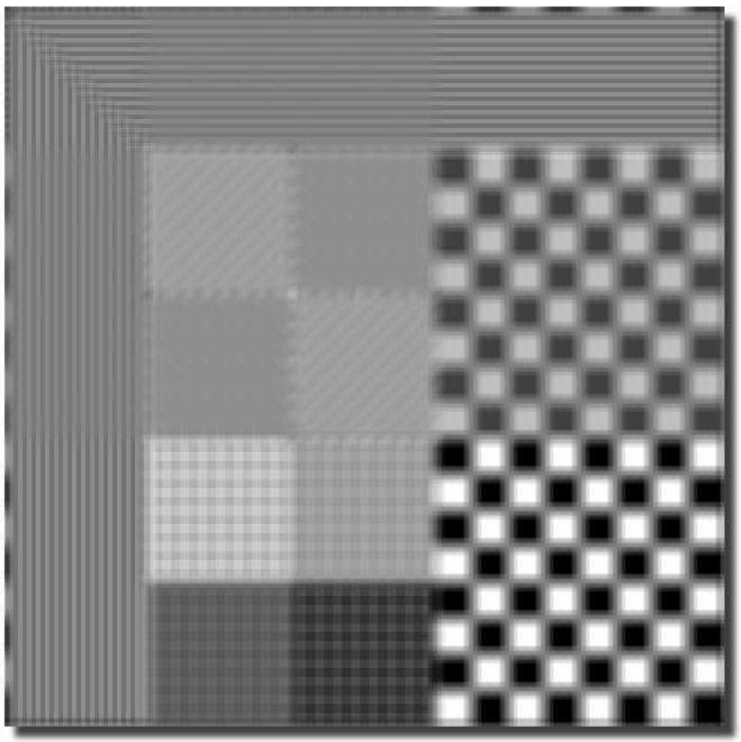
Convolution Examples: 5×5 Blur

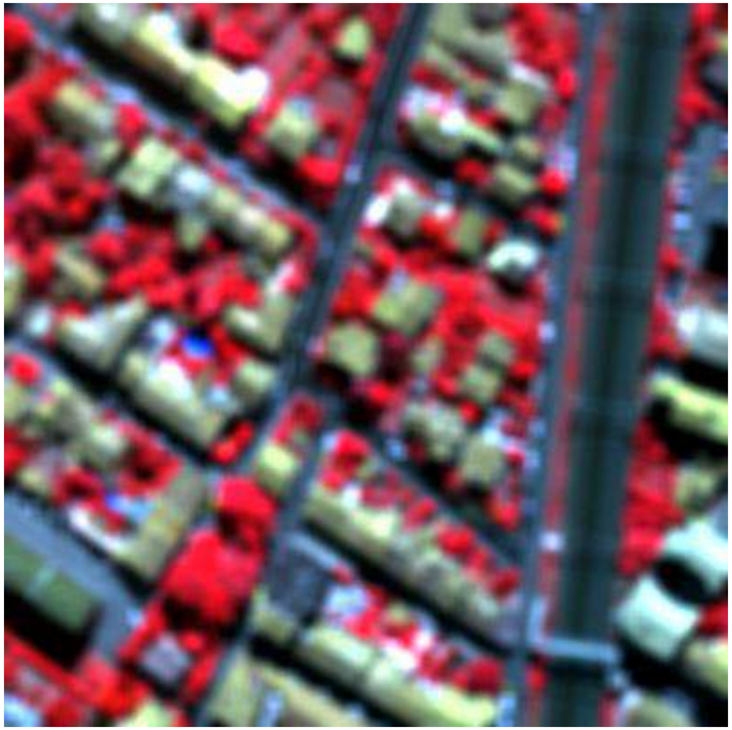


$$\frac{1}{81}\begin{bmatrix}1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\end{bmatrix}$$

Low-pass Filter

Convolution Examples: 9×9 Blur

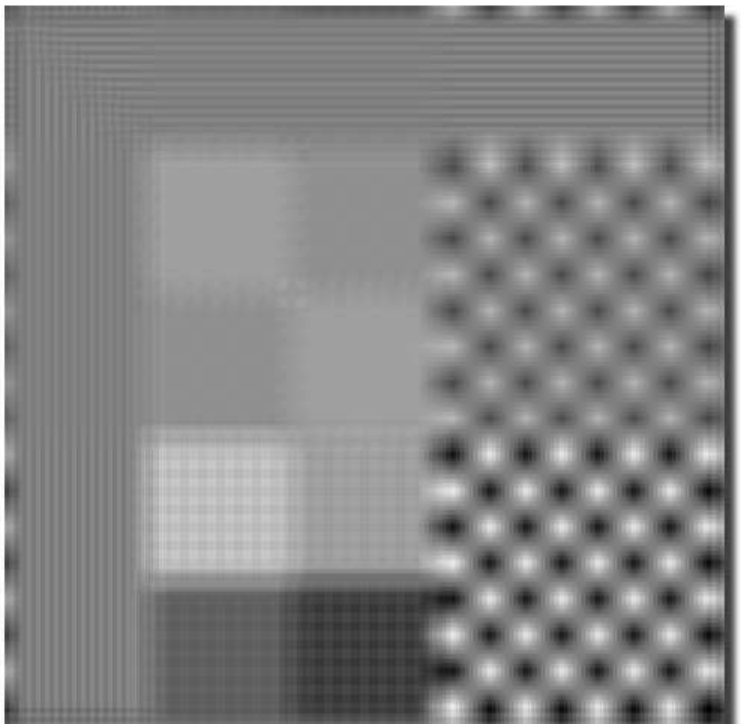


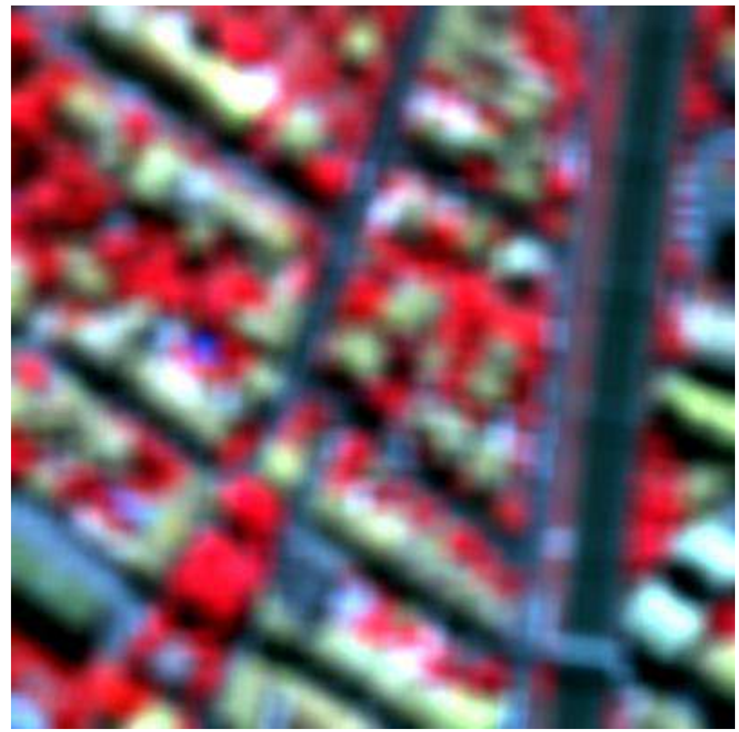


$$\frac{1}{289}\begin{bmatrix}1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\end{bmatrix}$$

Low-pass Filter

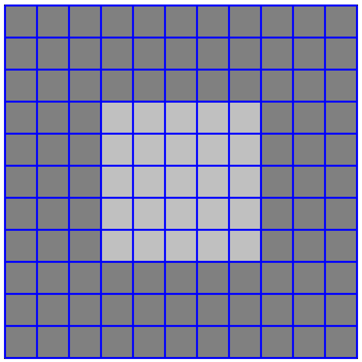
Convolution Examples: 17×17 Blur

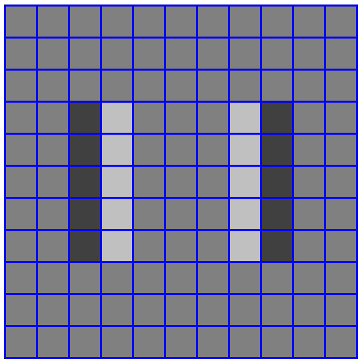


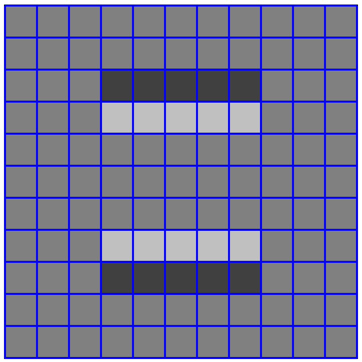


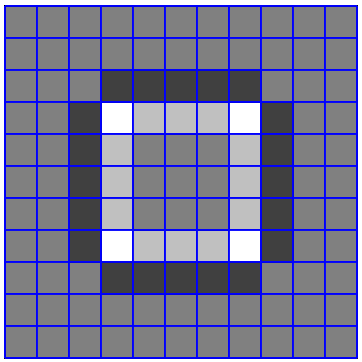
High-pass Filter

Moving Windows for Edge Detection









$$I(r,c)$$

$$2I(r,c) - I(r,c-1) - I(r,c+1)$$

$$2I(r,c) - I(r-1,c) - I(r+1,c)$$

$$4I(r,c) - I(r-1,c) - I(r+1,c) - I(r,c-1) - I(r,c+1)$$

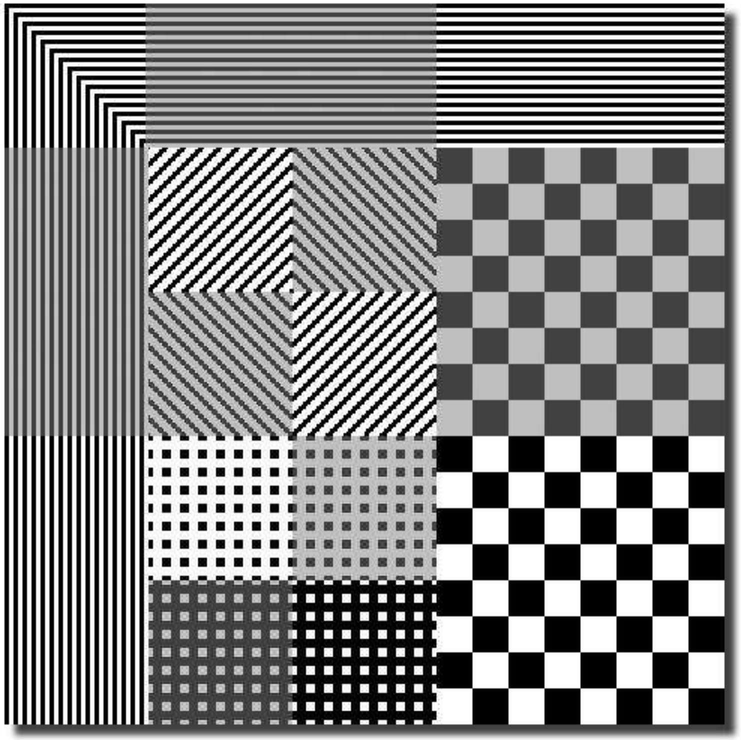
510
255
0
-255

-1	2	-1

	-1	
	2	
	-1	

	-1	
-1	4	-1
	-1	

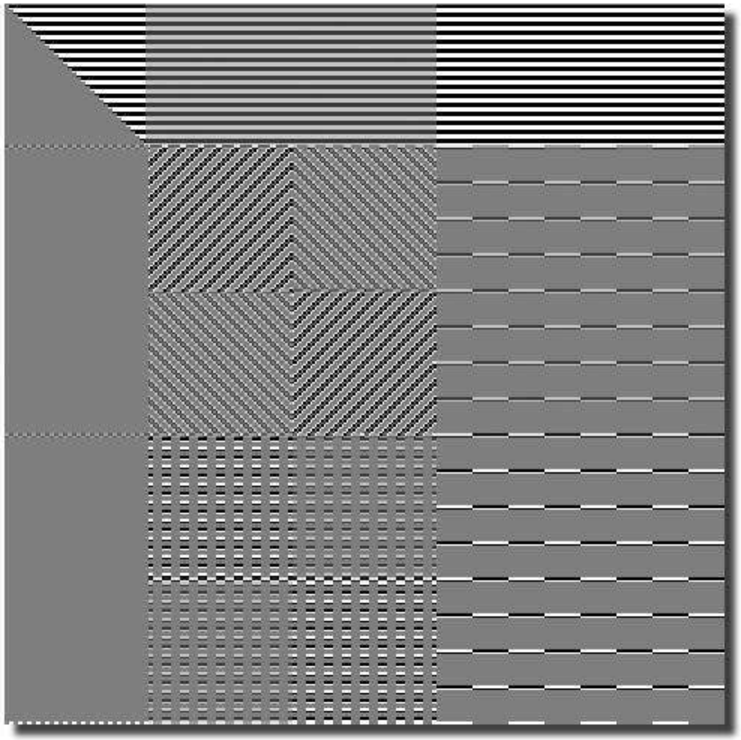
Convolution Examples: Original Images

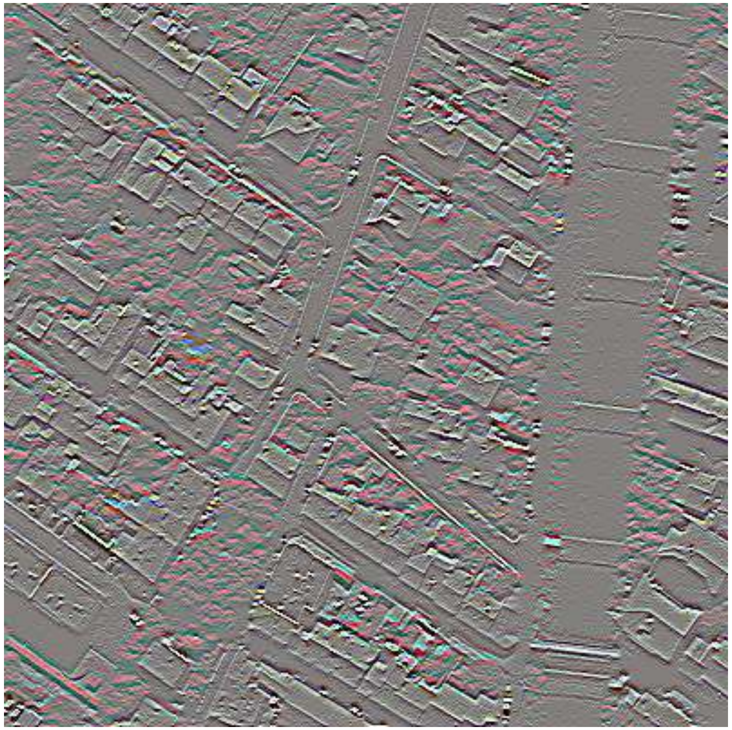


$$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$$

High-pass Filter

Convolution Examples: Vertical Difference



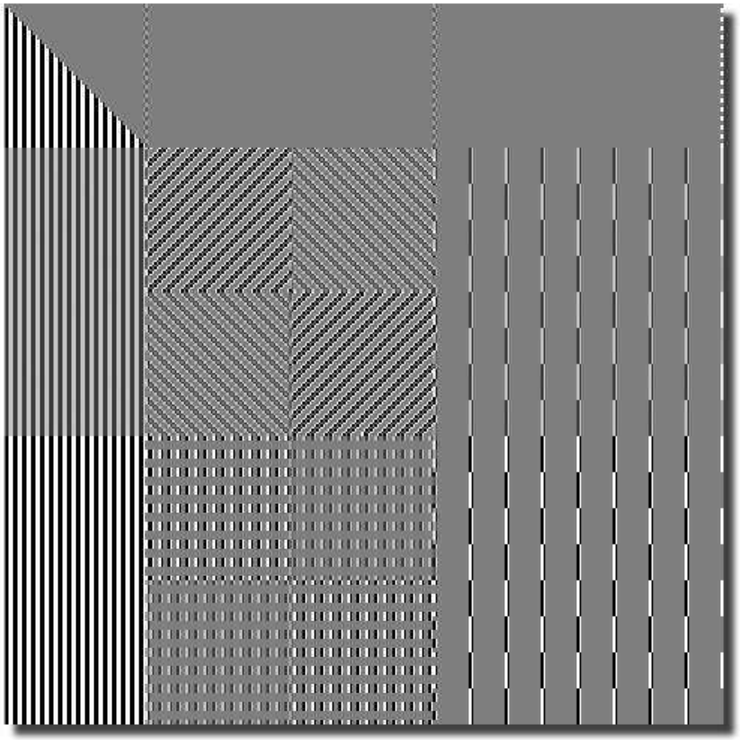


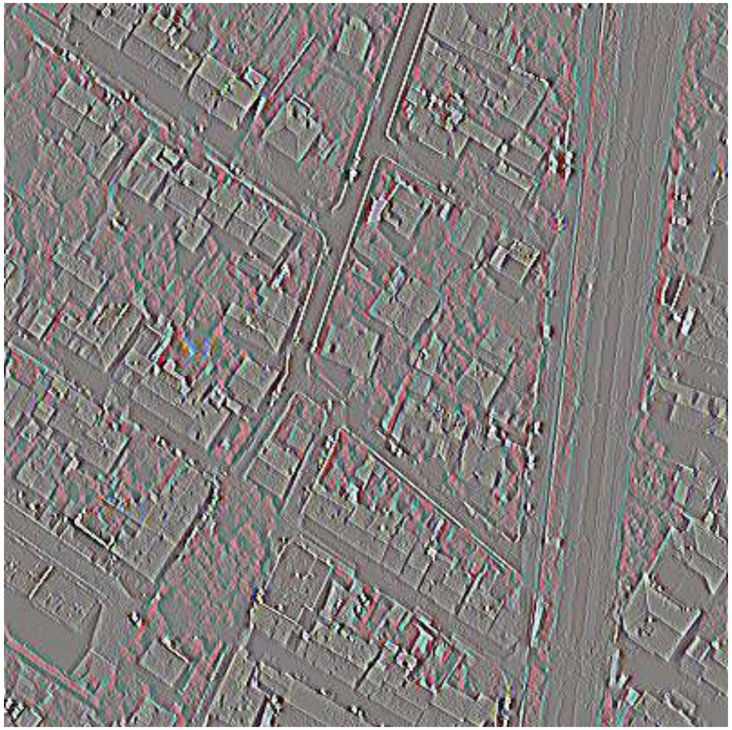
DLR

$$\begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$$

High-pass Filter

Convolution Examples: Horizontal Difference

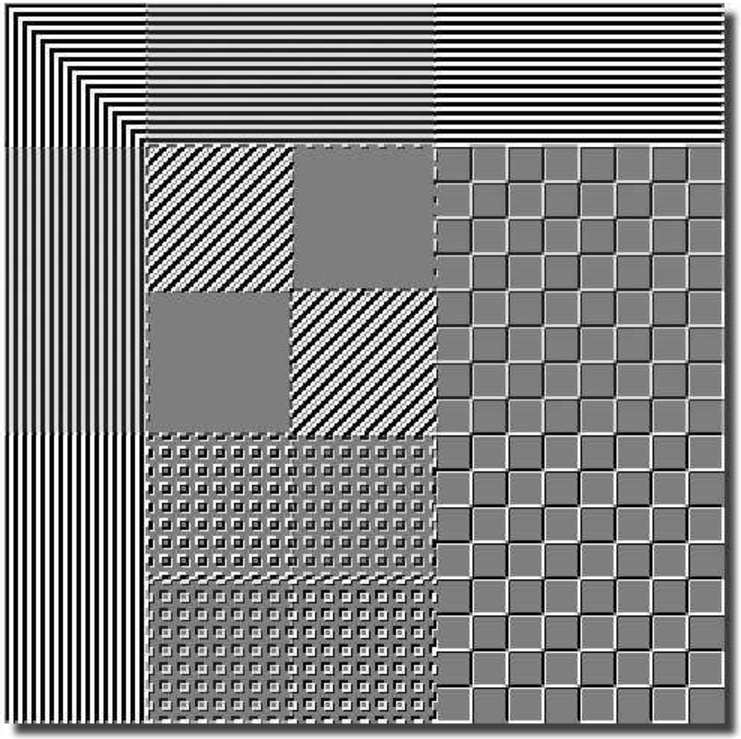


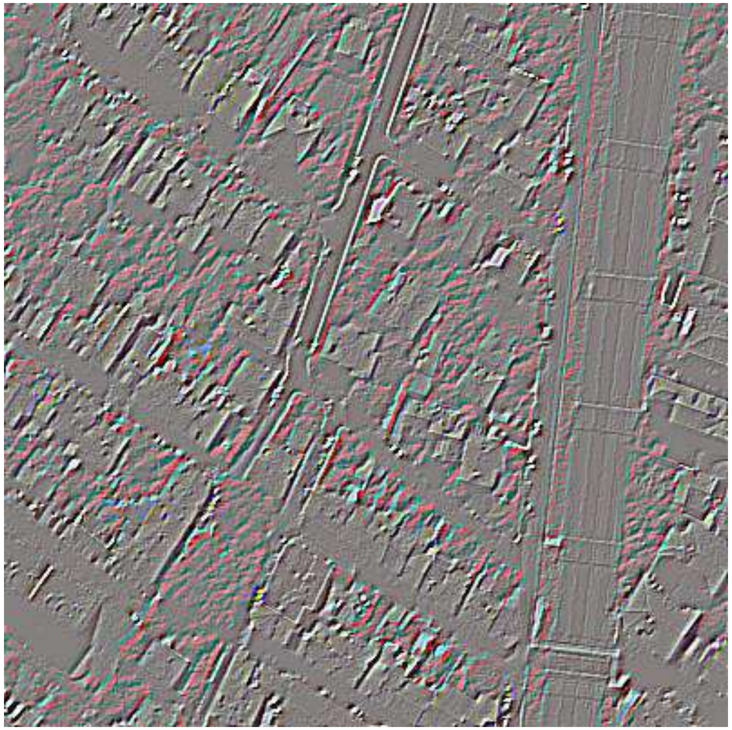


$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

High-pass Filter

Convolution Examples: Diagonal Difference



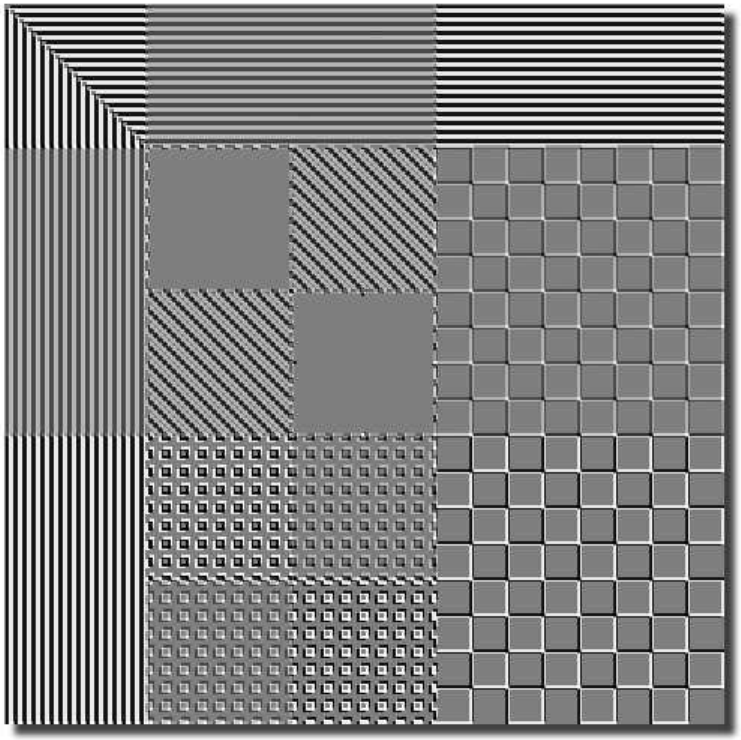


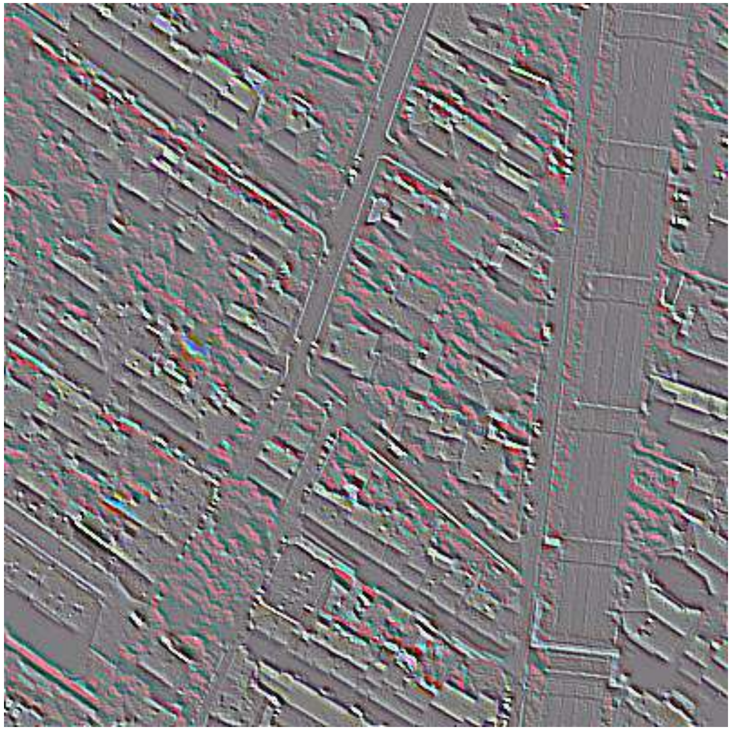
DLR

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

High-pass Filter

Convolution Examples: Diagonal Difference

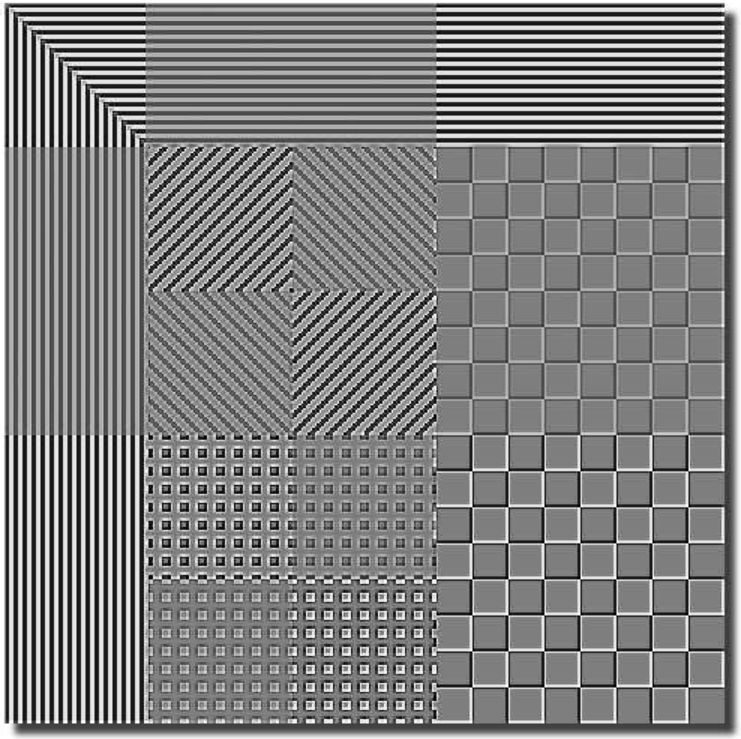




$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

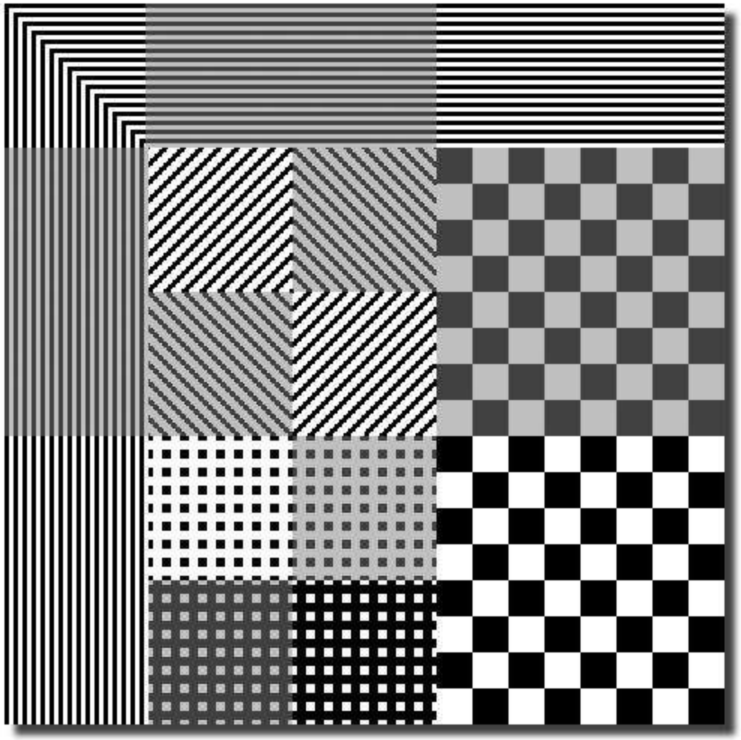
High-pass Filter

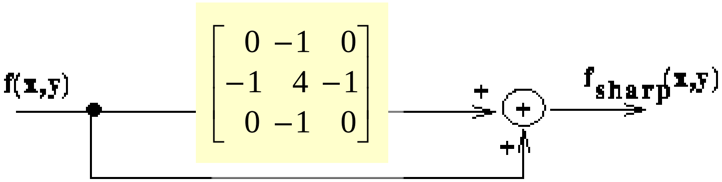
Convolution Examples: H + V + D Diff.



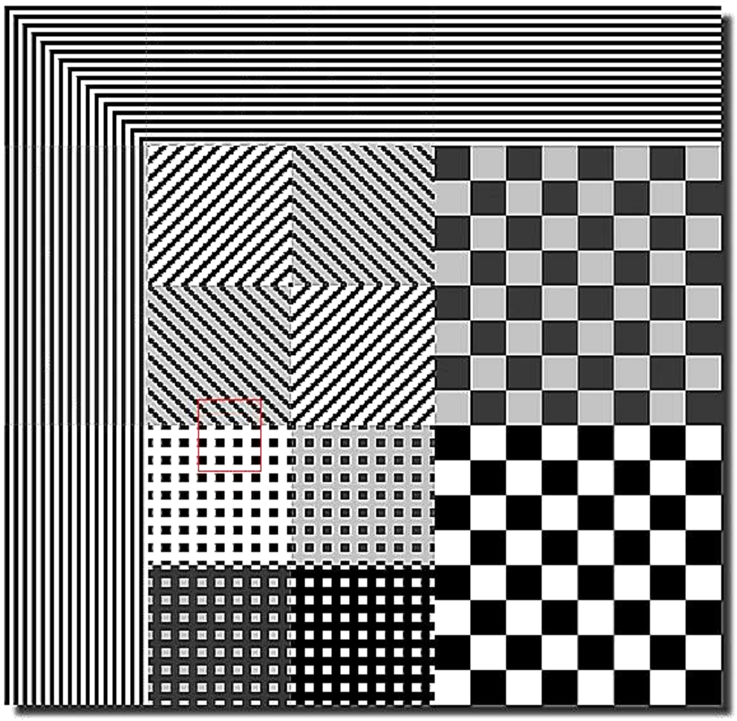
DLR

Convolution Examples: Original Images

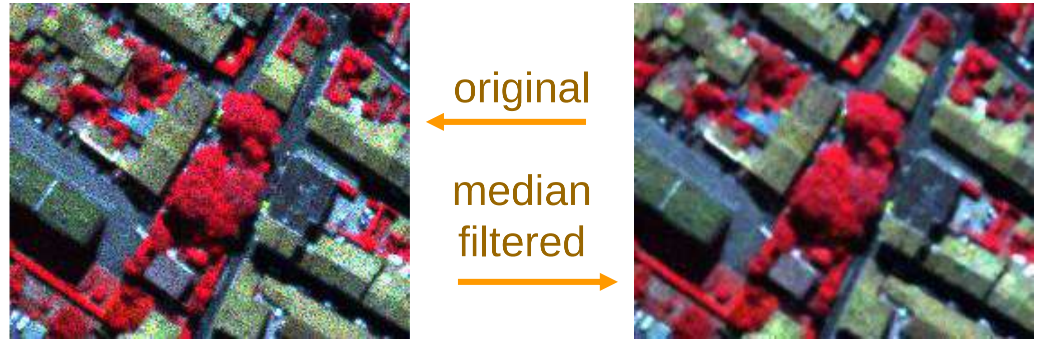




Convolution Examples: Sharpening



The Median Filter



– Returns the median value of the pixels in a neighborhood

– Morphological filter

– It does not "create" new pixel values but only rearranges values already present in the image, therefore..

– It preserves edges

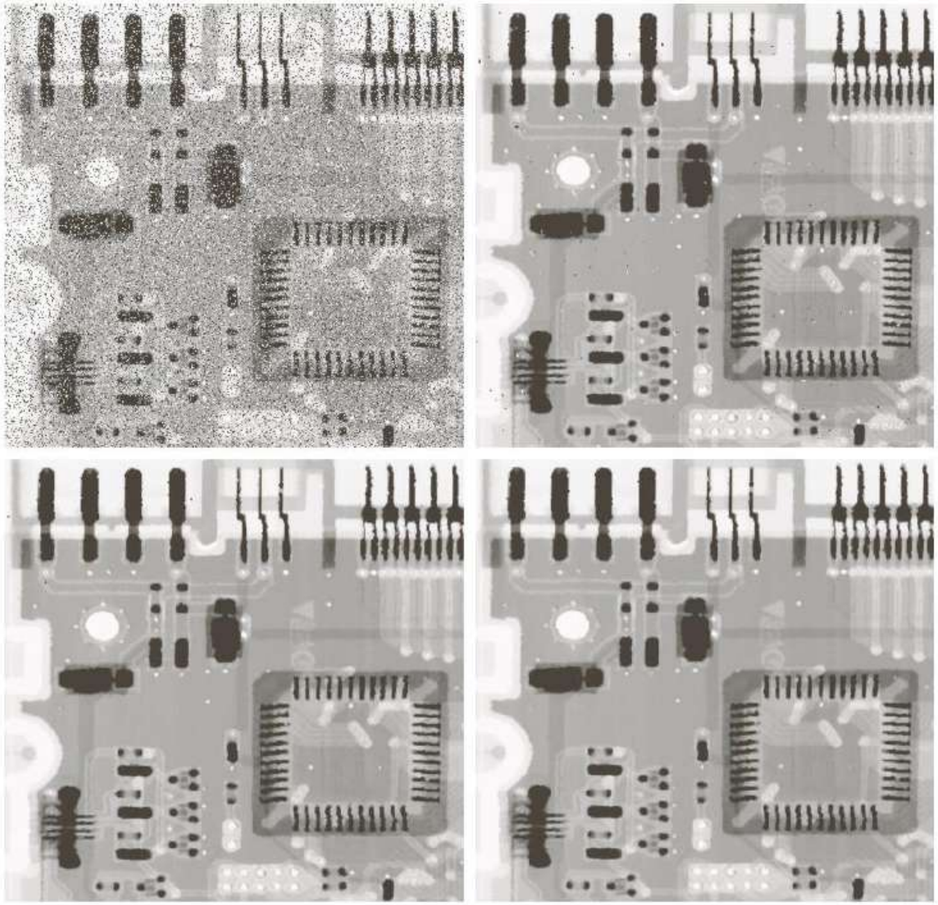
Median Filter

1. Good choice to remove impulse noise
2. Preserves edges
3. Avoids blurring

1	1	2	2	2	2	3	4	4

1	3	2
2	4	2
1	4	2

→ 2

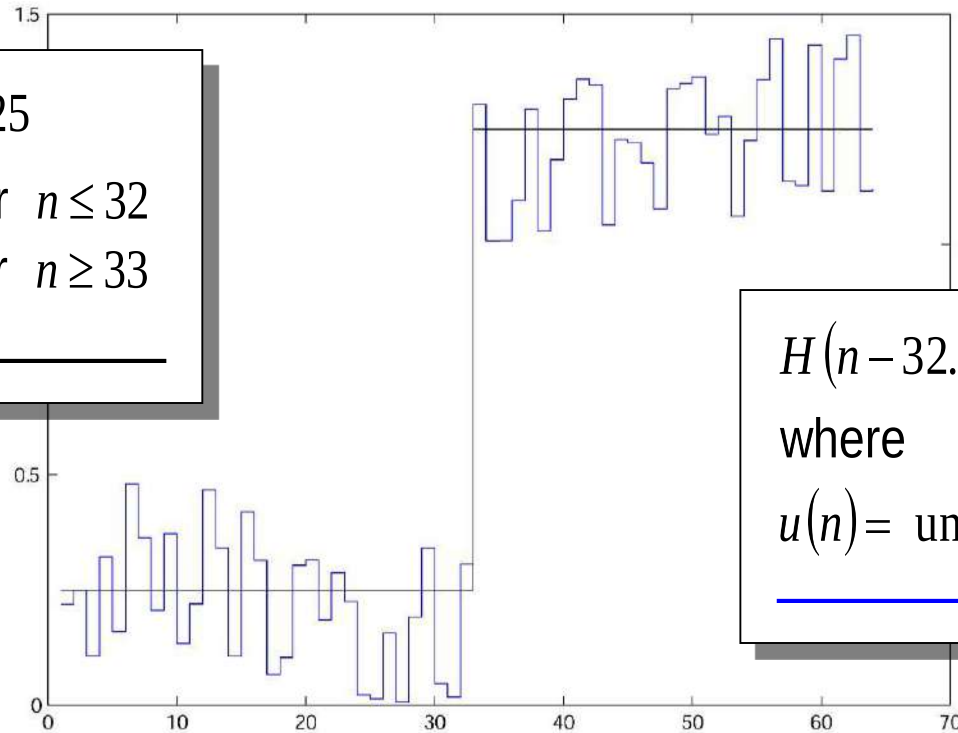


a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Example: A Noisy Step Edge

$$H(n-32.5)+0.25 = \begin{cases} 0.25 & \text{for } n \le 32 \\ 1.25 & \text{for } n \ge 33 \end{cases}$$



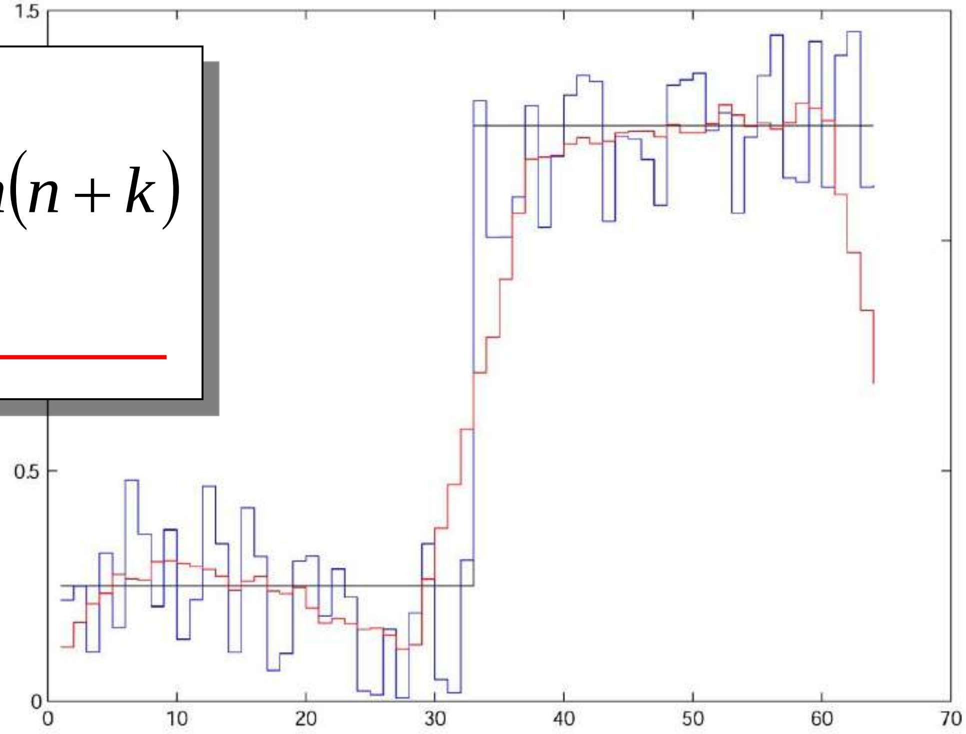
$$H(n-32.5)+u(n)$$

where

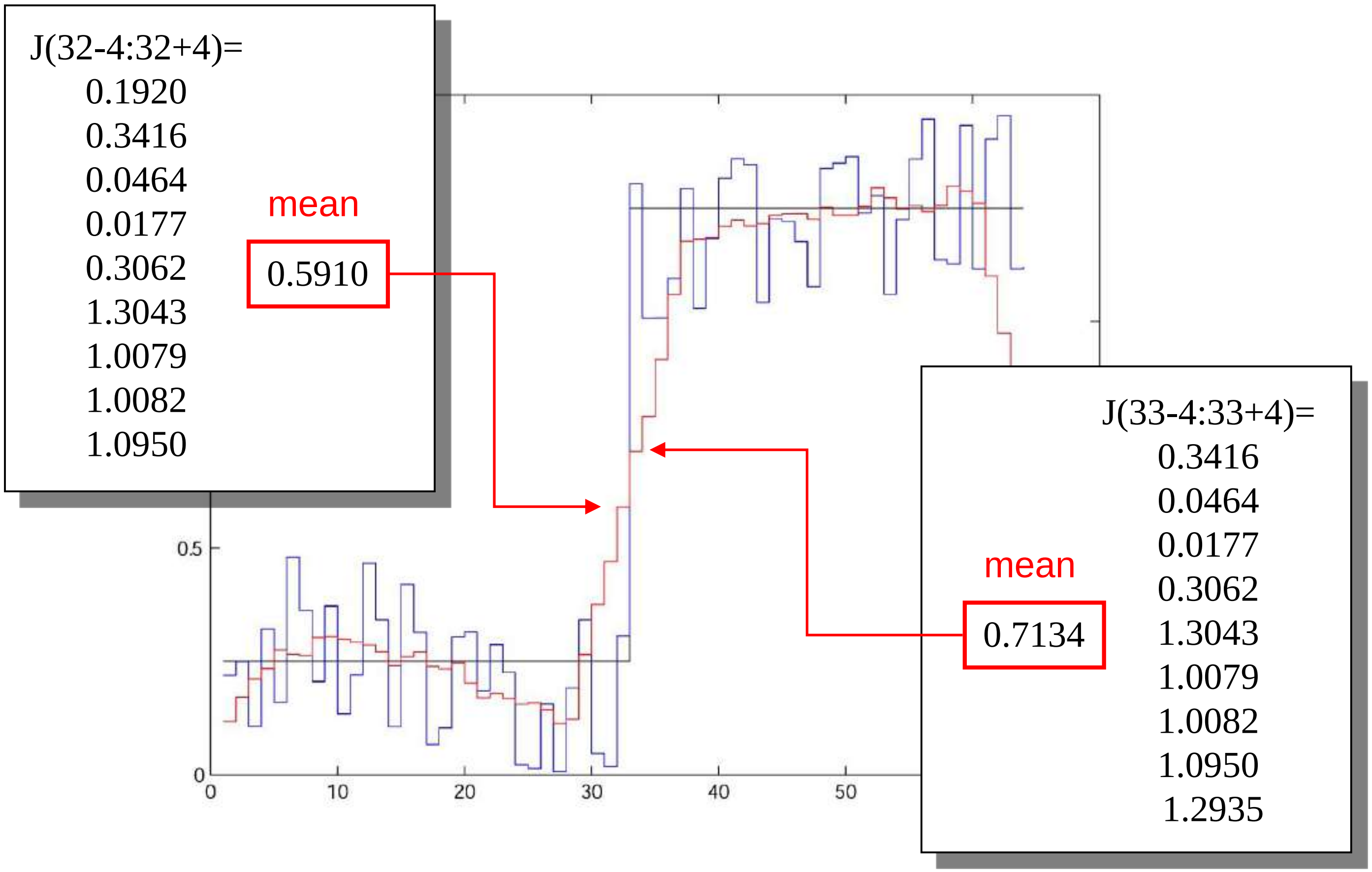
$$u(n) = \text{ unif } (-0.25, 0.25)$$

Blurred Noisy 1D Step Edge

$$h(n) = \frac{1}{9}\sum_{k=-4}^{4} h(n+k)$$

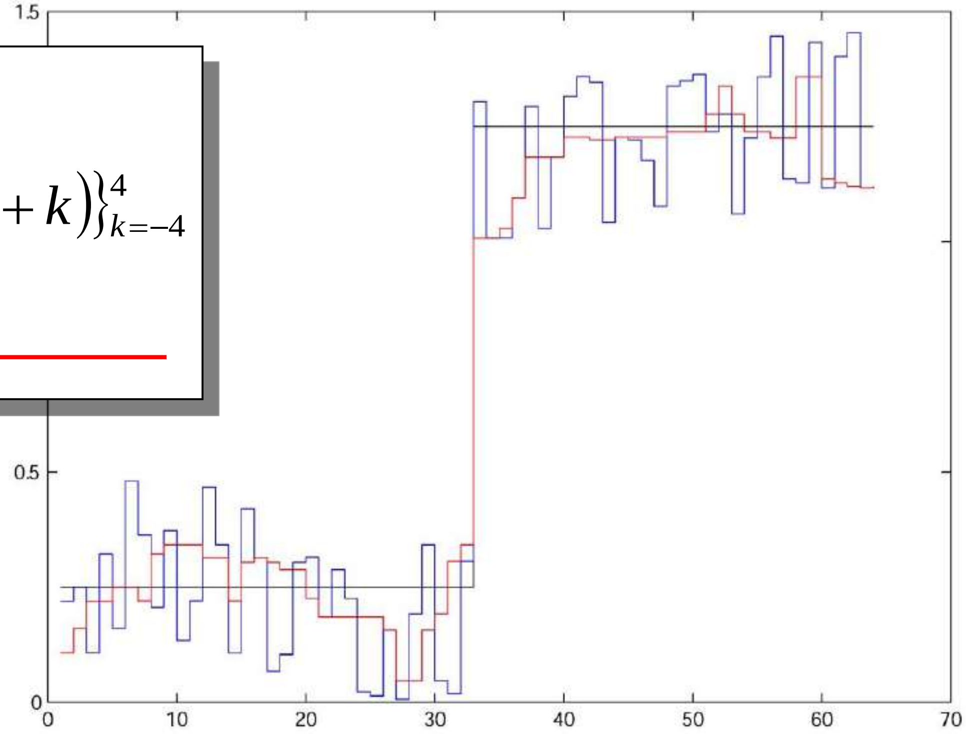


Blurred Noisy 1D Step Edge



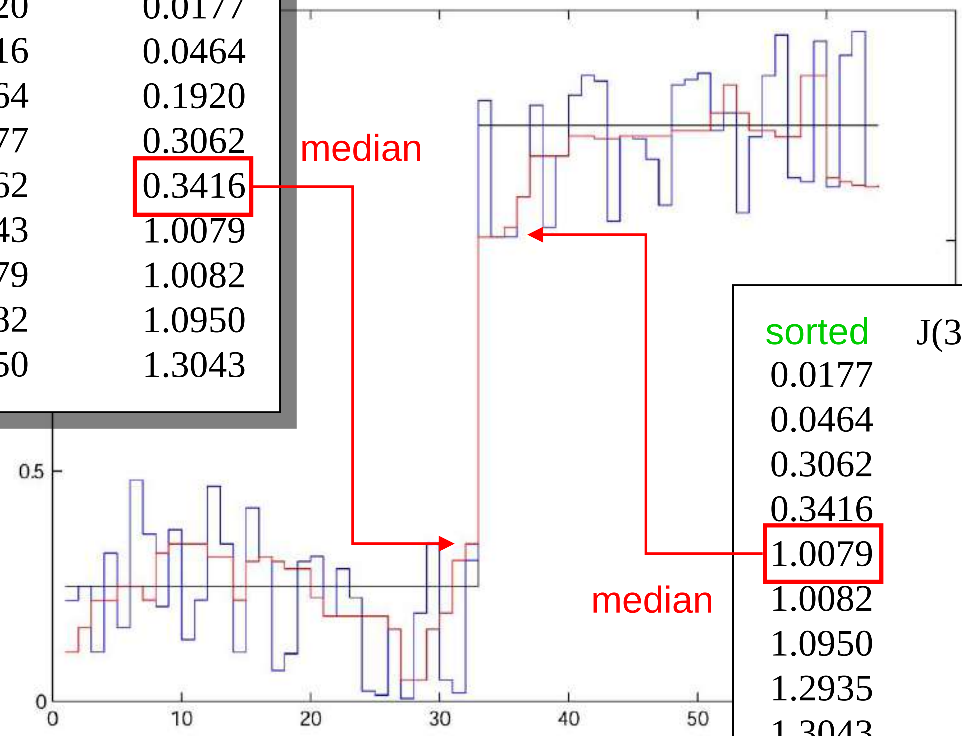
Median Filtered Noisy 1D Step Edge

$$h(n) = \text{med}\{h(n+k)\}_{k=-4}^{4}$$



Median Filtered Noisy 1D Step Edge

J(32-4:32+4)=	sorted
0.1920	0.0177
0.3416	0.0464
0.0464	0.1920
0.0177	0.3062
0.3062	**0.3416** (median)
1.3043	1.0079
1.0079	1.0082
1.0082	1.0950
1.0950	1.3043

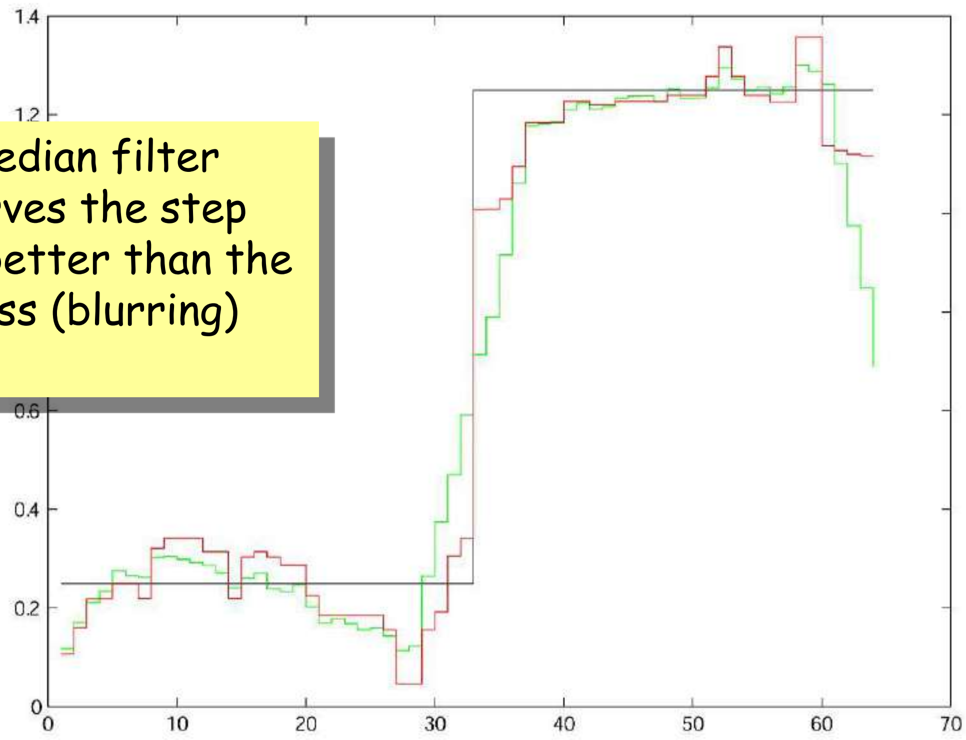


sorted	J(33-4:33+4)=
0.0177	0.3416
0.0464	0.0464
0.3062	0.0177
0.3416	0.3062
1.0079 (median)	1.3043
1.0082	1.0079
1.0950	1.0082
1.2935	1.0950
1.3043	1.2935

DLR

A Noisy Step Edge

Median vs. Blurred (Low-pass)



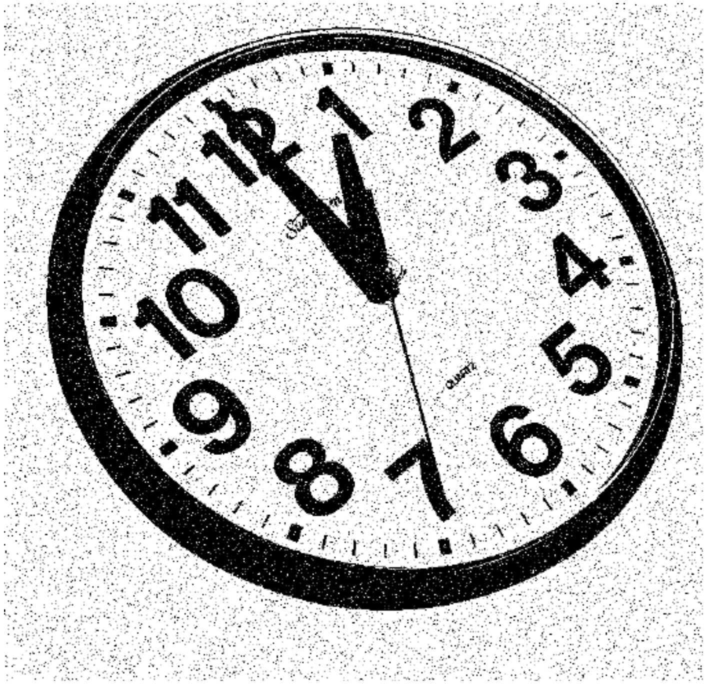
The median filter preserves the step edge better than the low-pass (blurring) filter.

median

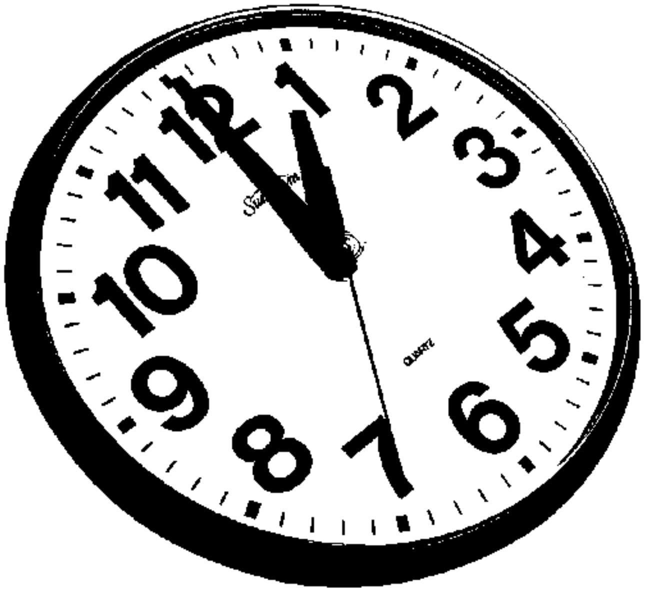
blurred

step

Median Filtering of Binary Images



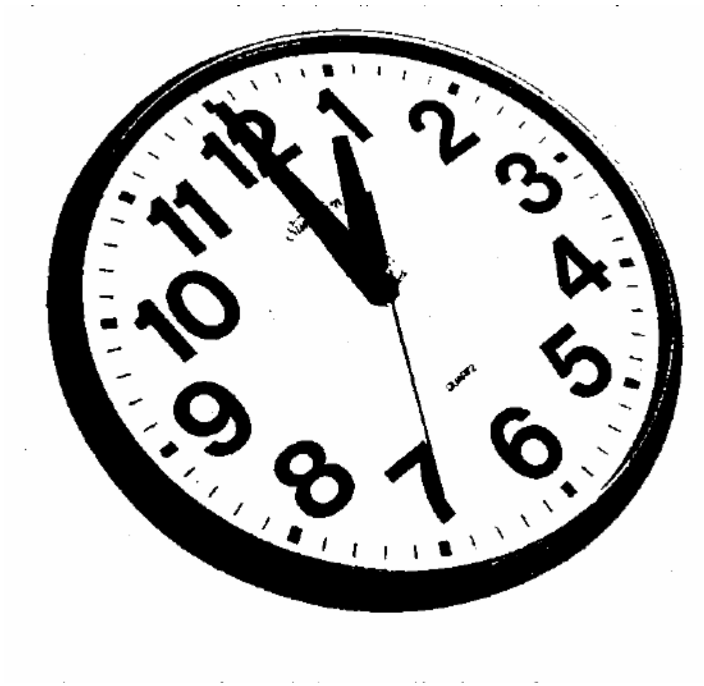
Noisy



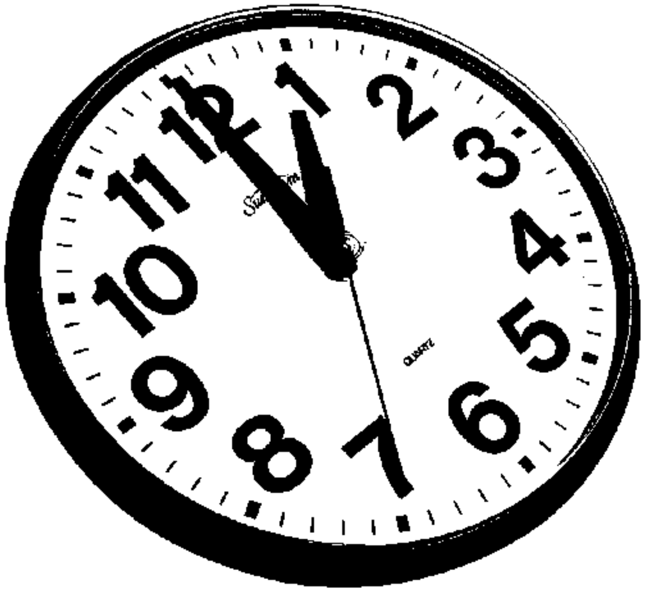
Original

DLR

Median Filtering of Binary Images

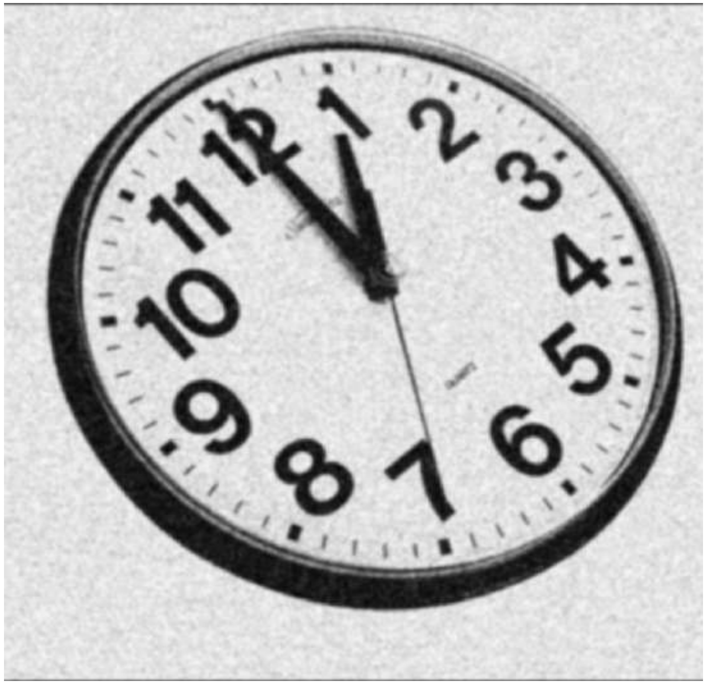


Median Filtered Noisy

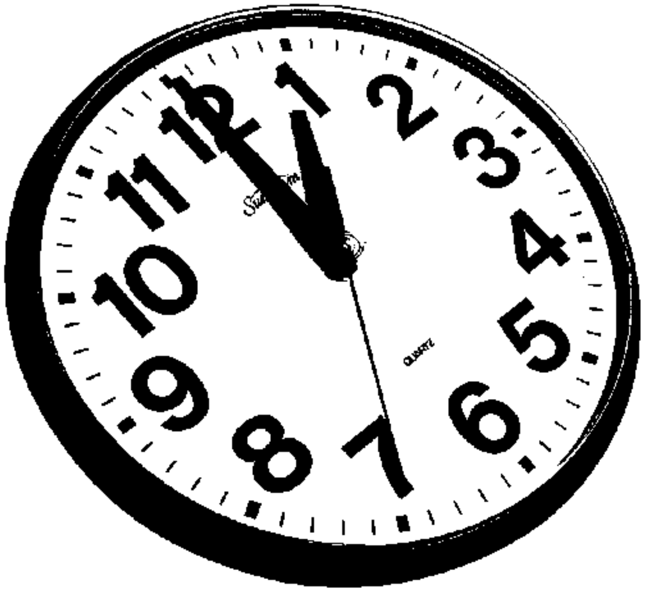


Original

Filtering of Grayscale Images



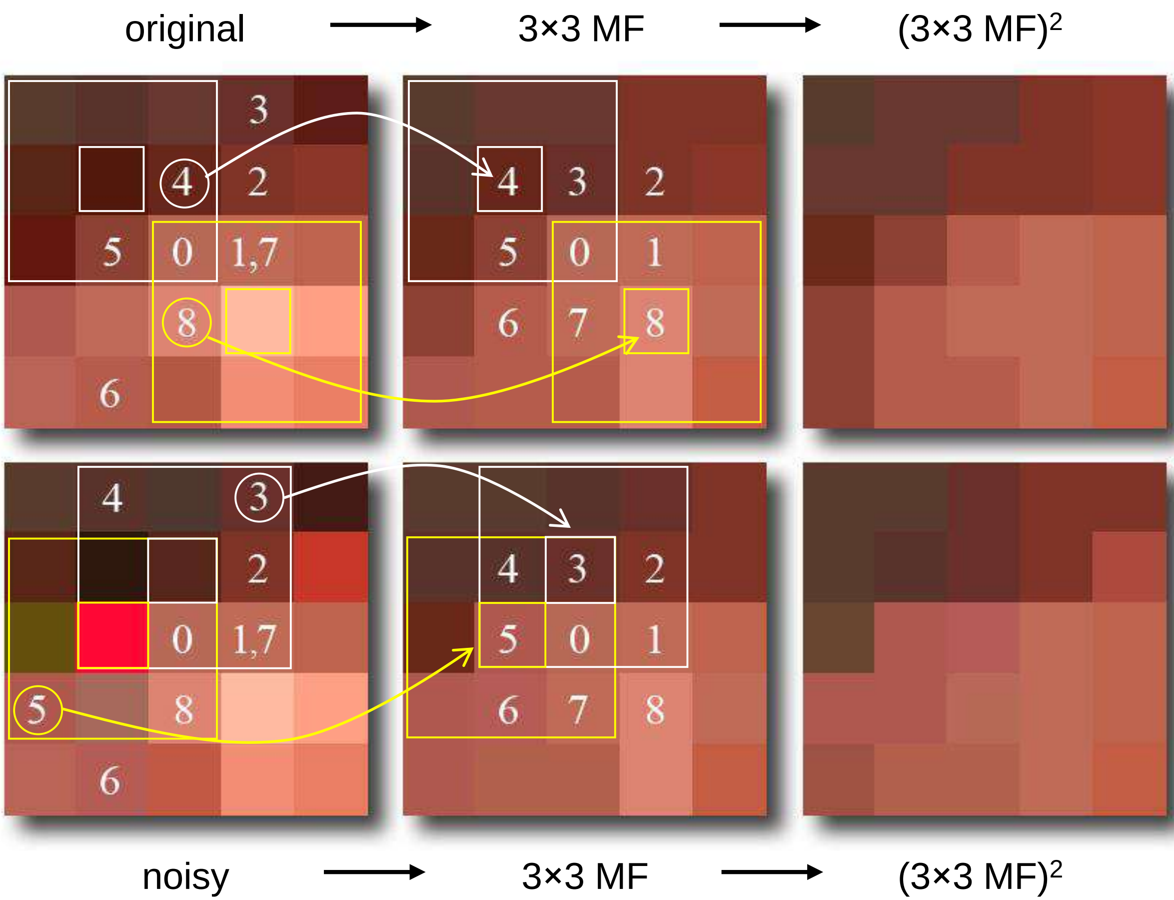
Blurred Noisy



Original

Color Median Filter

The output color at (r,c) is always selected from a nbhd of (r,c) in the input image.



R. A. Peters

Filtering of Multiband Images

Noisy

Noisy

Filtering of Multiband Images

3x3 Blur

3x3 Median

Filtering of Multiband Images

3x3 Blur X 2

3x3 Median X 2

Filtering of Multiband Images

3x3 Blur X 5

3x3 Median X 5

Filtering of Multiband Images

3x3 Blur X 10

3x3 Median X 10

Limit and Root Images

3x3-blur x n→∞

3x3-median root

Image Filtering in Frequency Domain

Daniele Cerra, German Aerospace Center (DLR)

DLR

- Image Acquisition & Correction
  - Raw Data→ Raw Image → Image
- Low-level Analysis
  - **Image → Image**
    - Time domain
    - **Frequency domain**
- Mid-level Analysis
  - Image → Features / Attributes
    - Feature Extraction
    - Clustering / Segmentation
- High-level Analysis
  - Features → Recognition

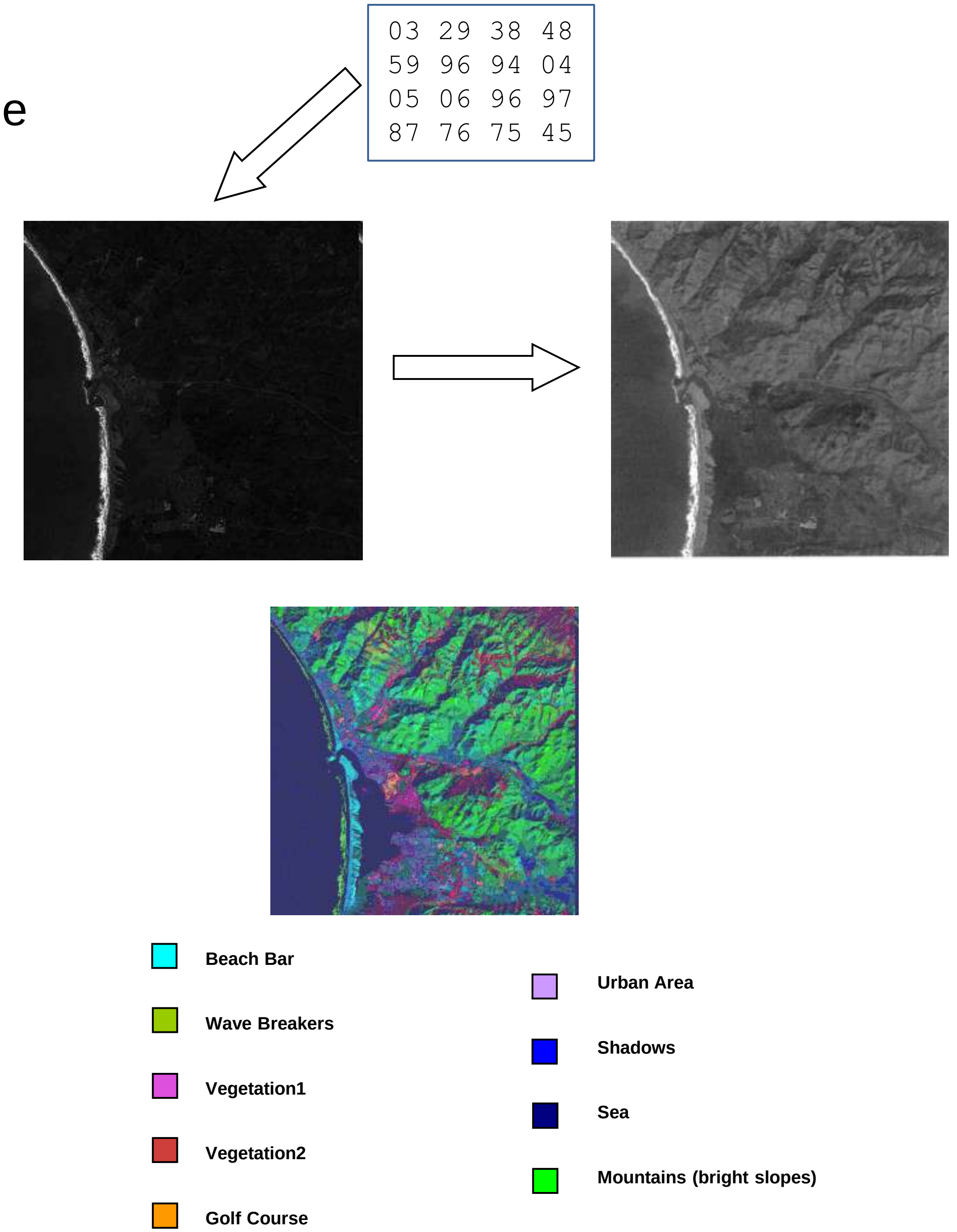
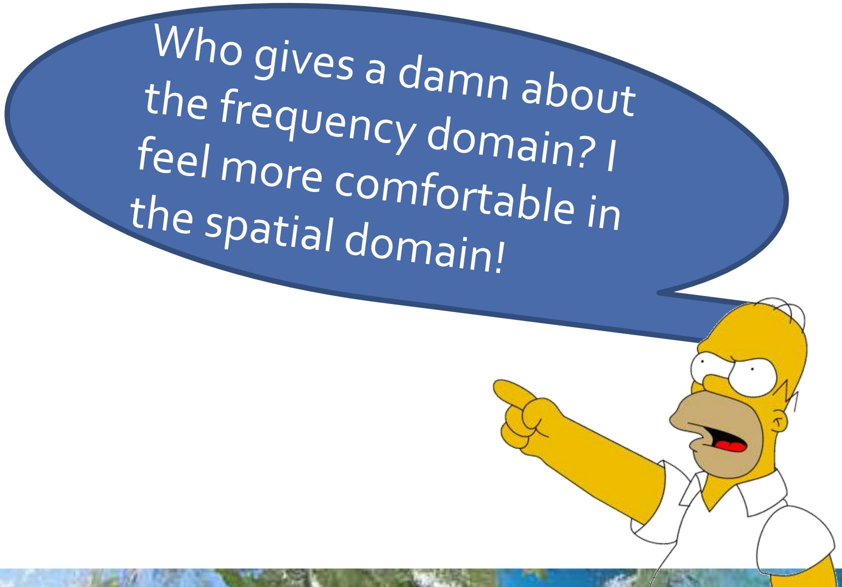


Image enhancement

– Images can be represented (and enhancement can be done in):

- Time domain
  - Measurements with respect to a point in time and/or positions in space
  - Direct manipulation of pixels
- Frequency domain
  - Representation of a signal in terms of its ondulations
  - Main concept: Fourier Transform



Physical reason: the universe works with „waves“… is just that us humans see things differently

Practical reason: operating in the frequency domain is often computationally convenient, and enables operations which are very difficult to conduct in the spatial domain

The Fourier Transform

Jean Baptiste Joseph Fourier

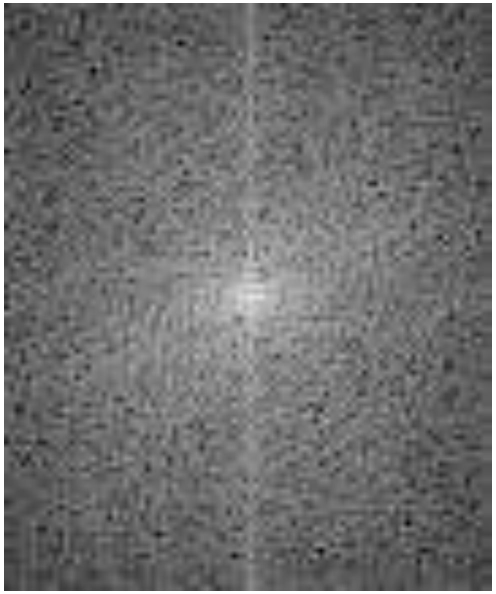
- Fourier had a crazy idea: Any periodic function can be written as a weighted sum of sines and cosines of different frequencies (1807)

  → Fourier series

- Even functions that are not periodic can be expressed as the integral of sines and cosines multiplied by a weighting function.

  → Fourier transform

The Fourier Transform



Jean Baptiste Joseph Fourier

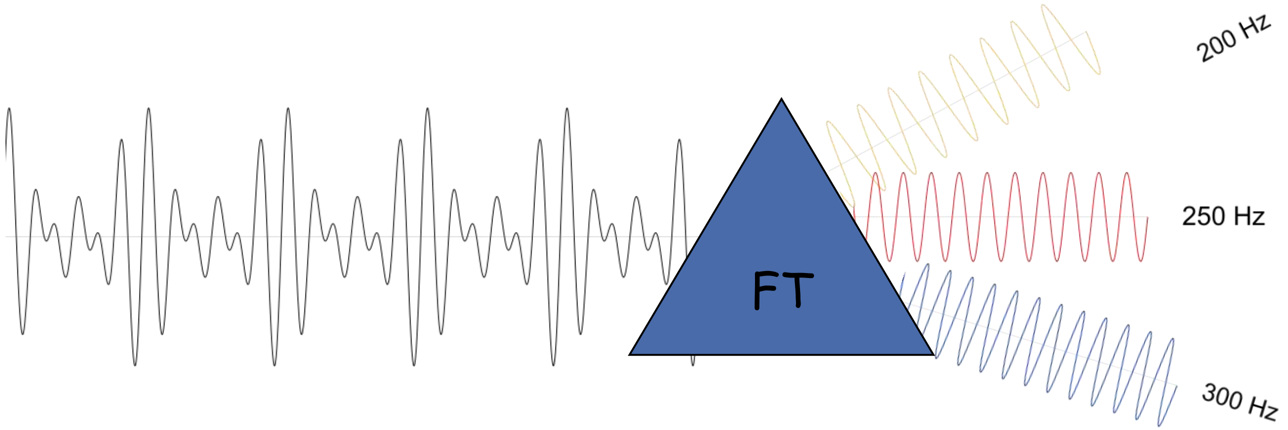
- (Part of) the result of such transform can be represented like this…
- Don’t panic! ☺

What is the idea of the Fourier Transform?

– Newton's prism separates a stream of white light into different colors

– These colors are components of the light at different frequencies

What is the idea of the Fourier Transform?



– The FT decomposes any periodic (or time-limited) signal in terms of its frequency components

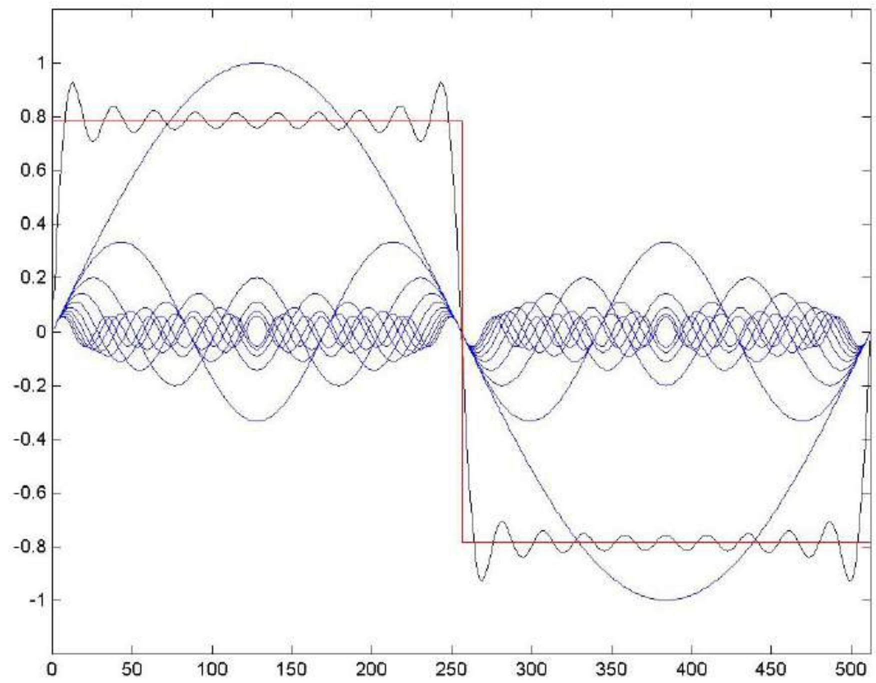
Frequency-Domain Representation

Surprise: **any** signal can be described by a sum of sinusoids!

The sinusoids (in blue) are called “basis functions”.

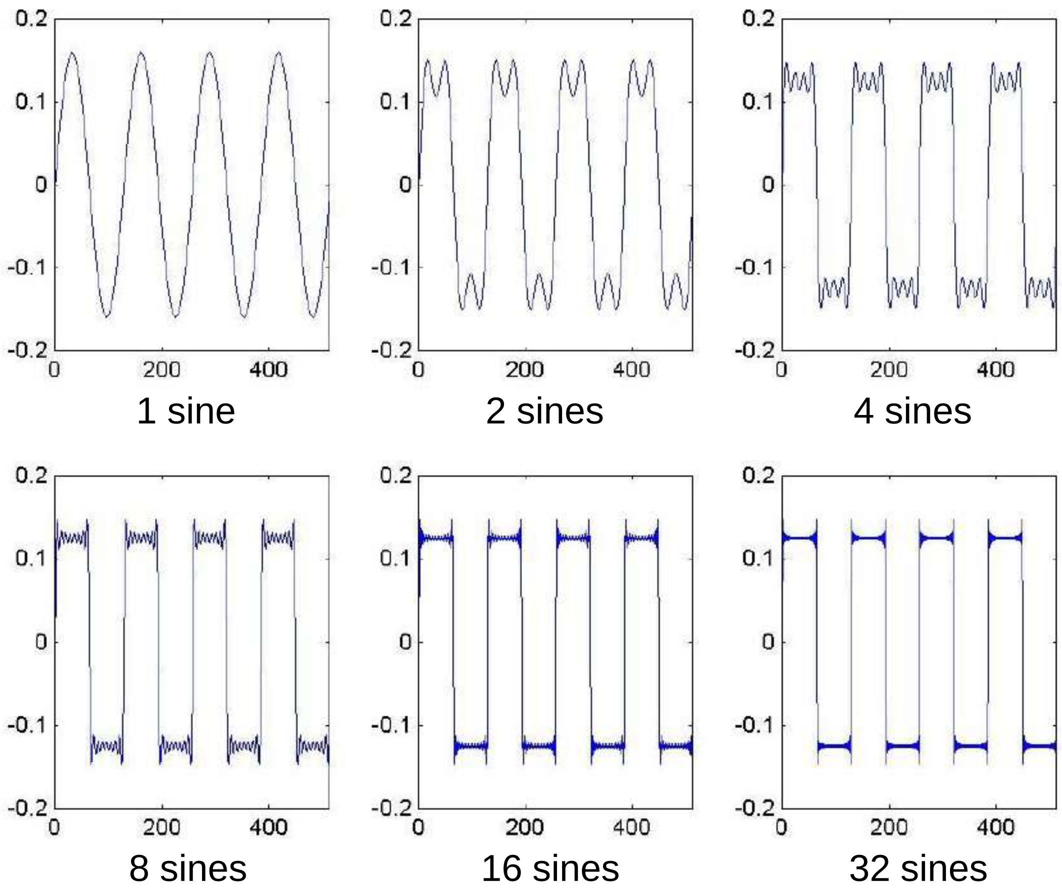
The coefficients which make them “larger” or “smaller” are the “Fourier coefficients”.

Their sum is the **black** function approximating the square wave in red.



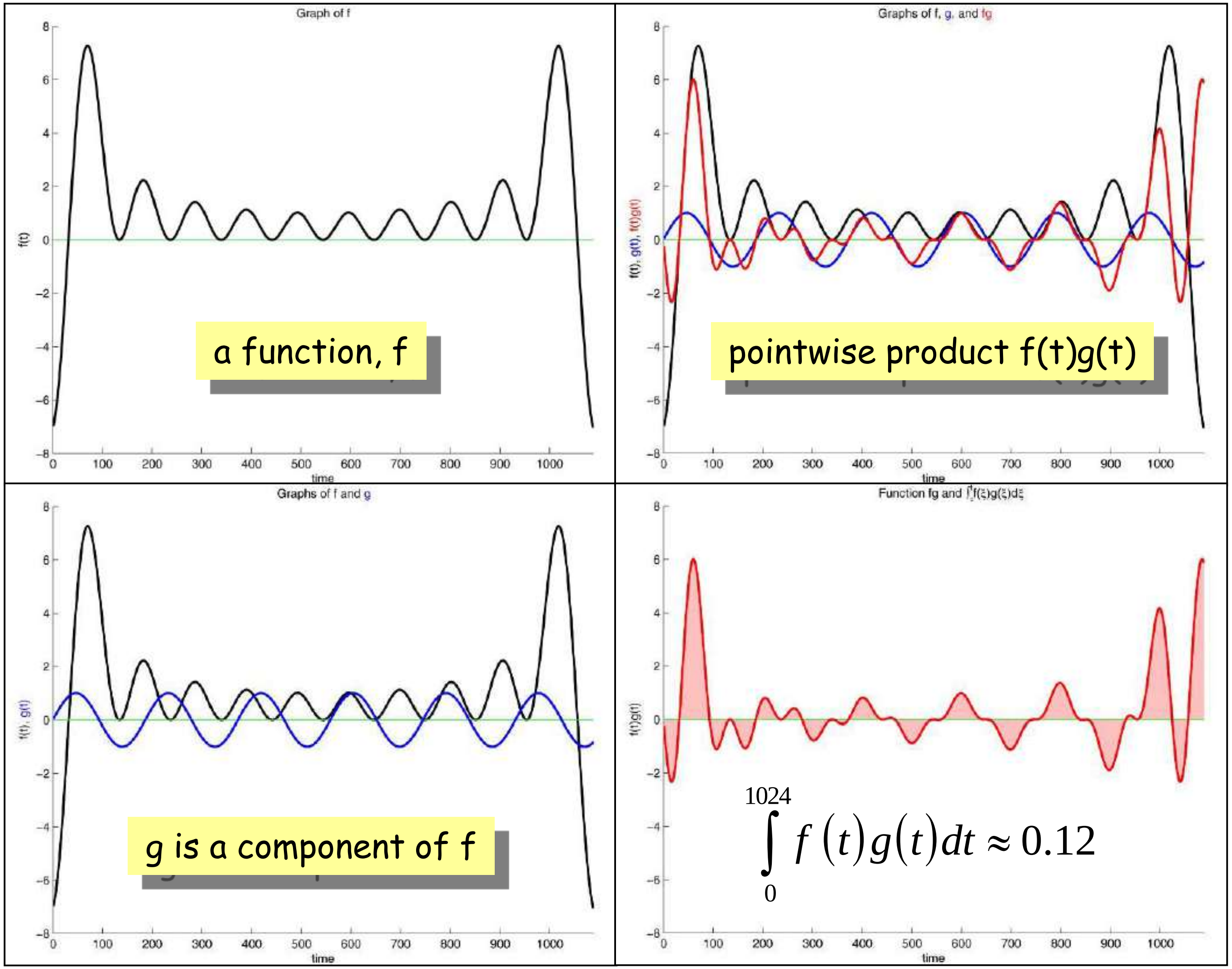
Example: Partial Sums of a Square Wave

The limit of the given sequence of partial sums[1] is exactly a square wave

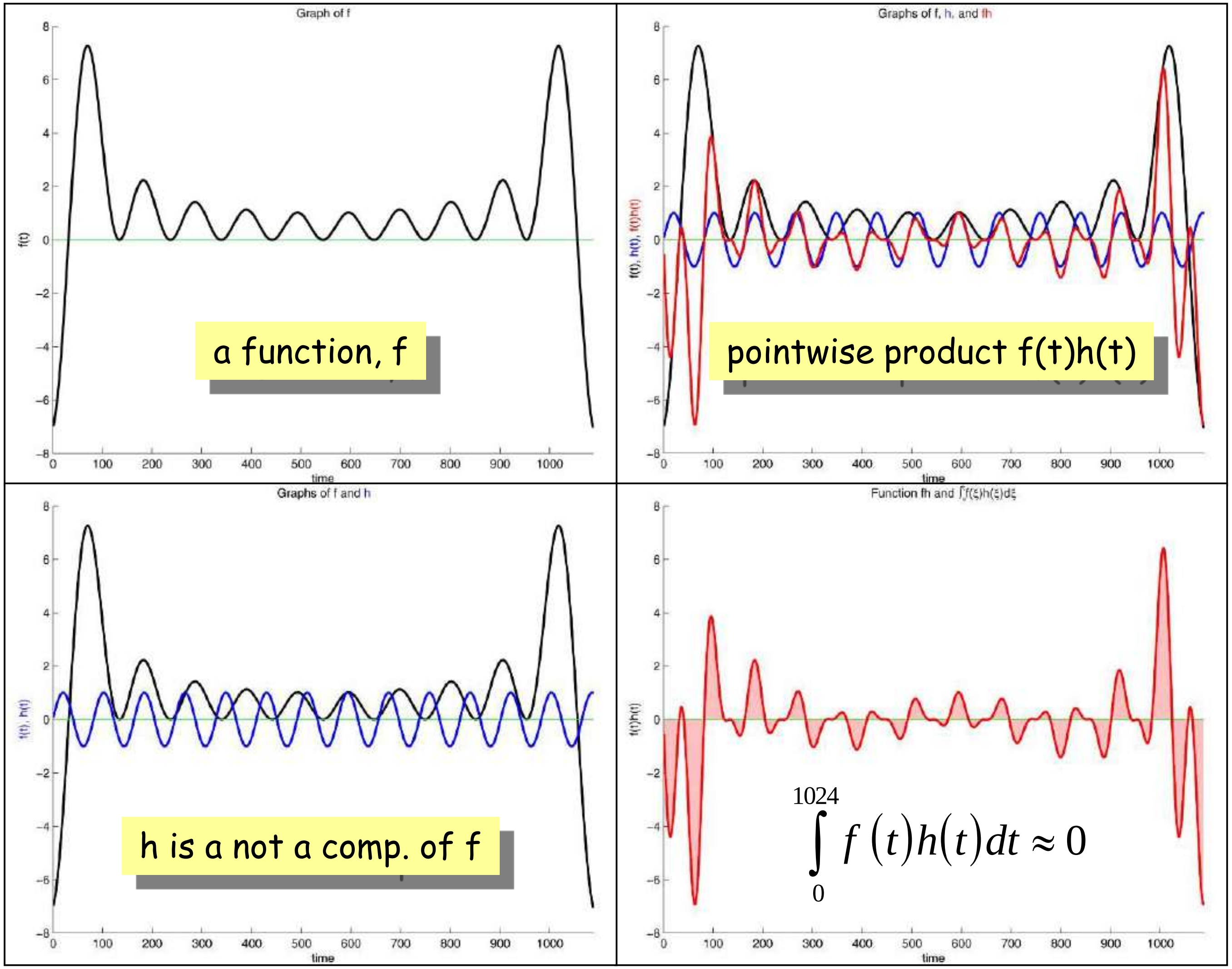


[1] the limit as n approaches infinity of the sum of n sines.

Inner Products: a measure of similarity



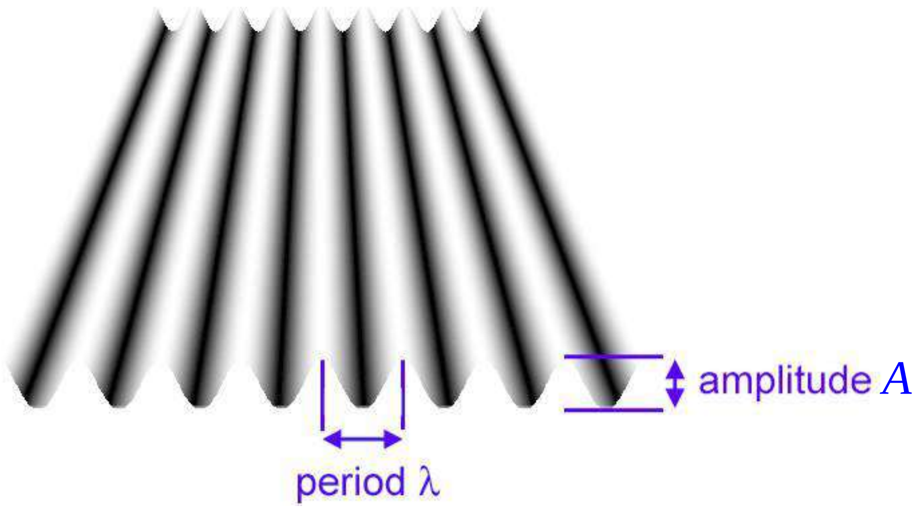
Inner Products: a measure of similarity

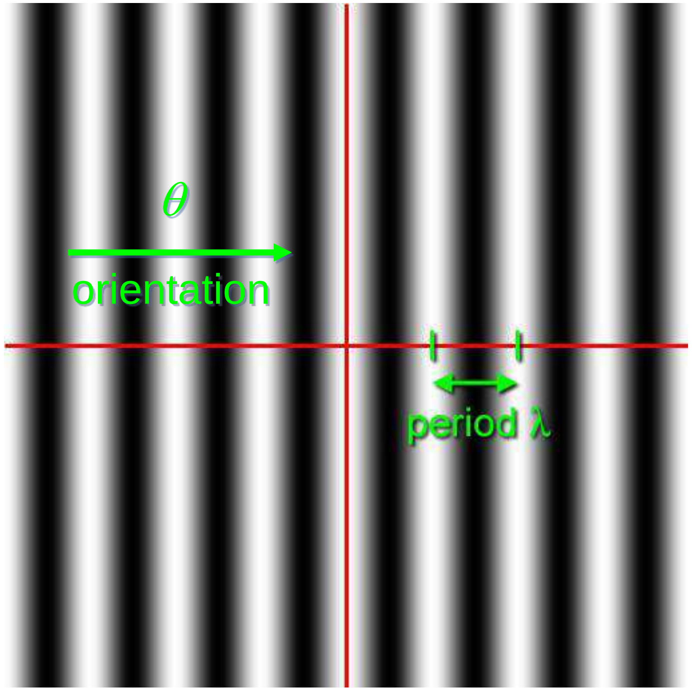


$$\int_0^{1024} f(t)h(t)dt \approx 0$$

2D Sinusoids:

... are plane waves with grayscale amplitudes, periods in terms of lengths, ...





ϕ = phase shift

We can represent 2D Signals as a sum of these….

I

DLR

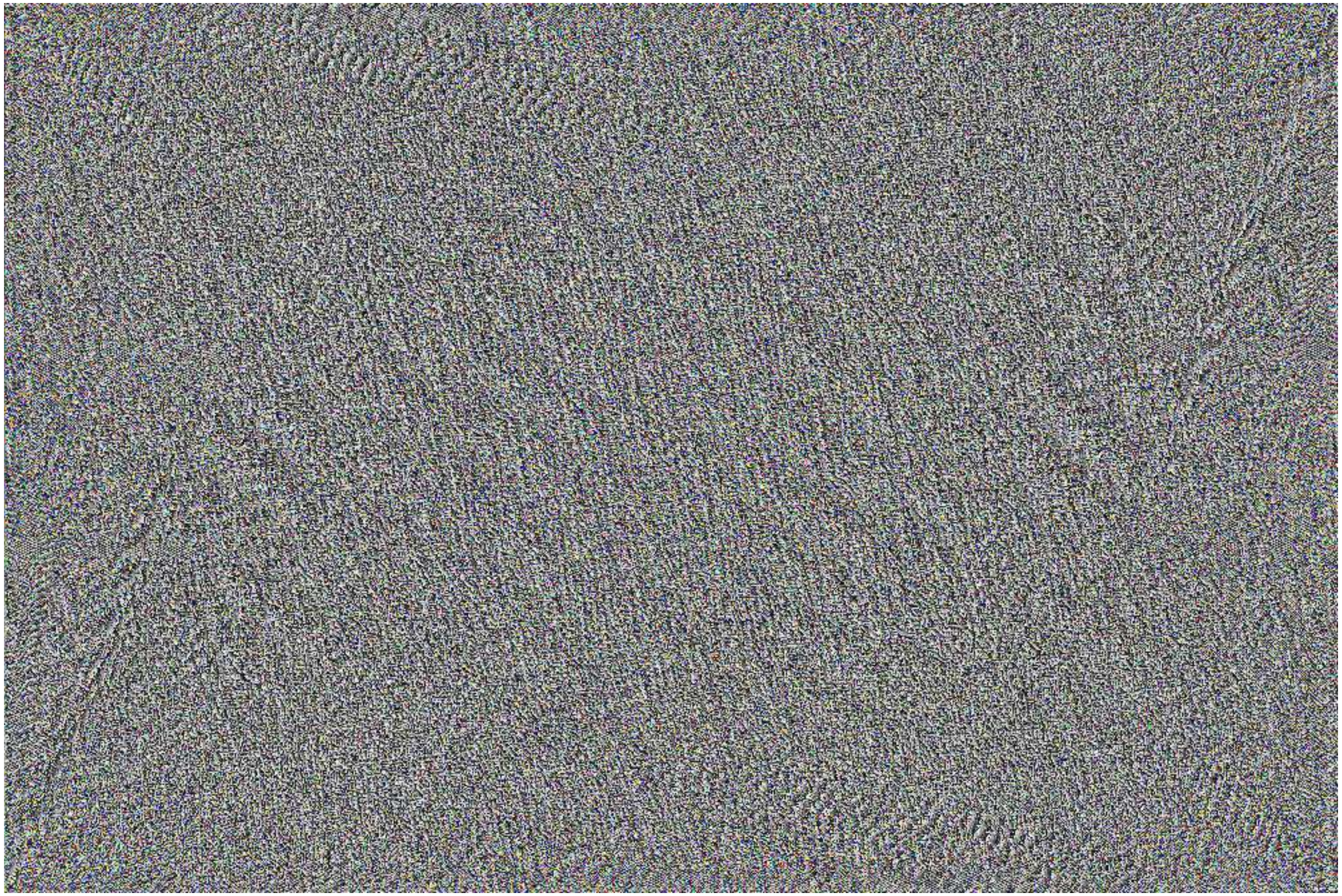
R. A. Peters

Fourier Spectrum
(related to the **amplitude** of the sinusoids)

$$\log\left|\mathcal{F}\{I\}\right|$$

Fourier Phase
(related to the “location” –shifting- of the sinusoids)

$$\angle \mathfrak{F}\{I\}$$

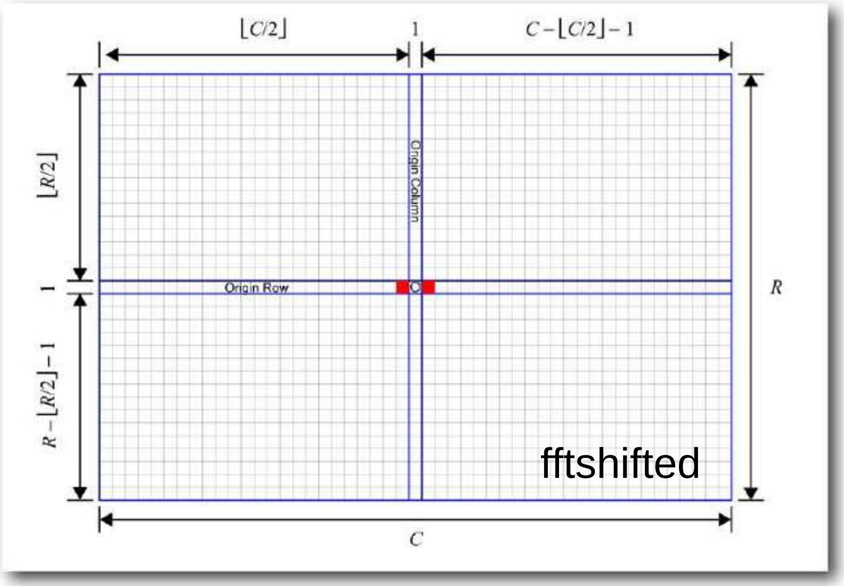


DLR

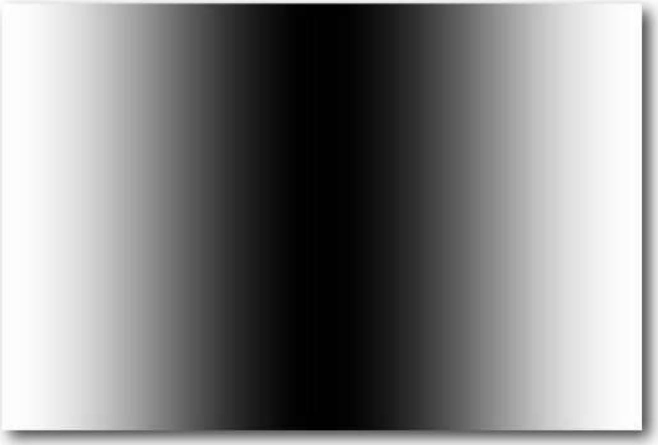
How to map frequencies onto an image?

"horizontal" is the wavefront direction.

Frequency Domain



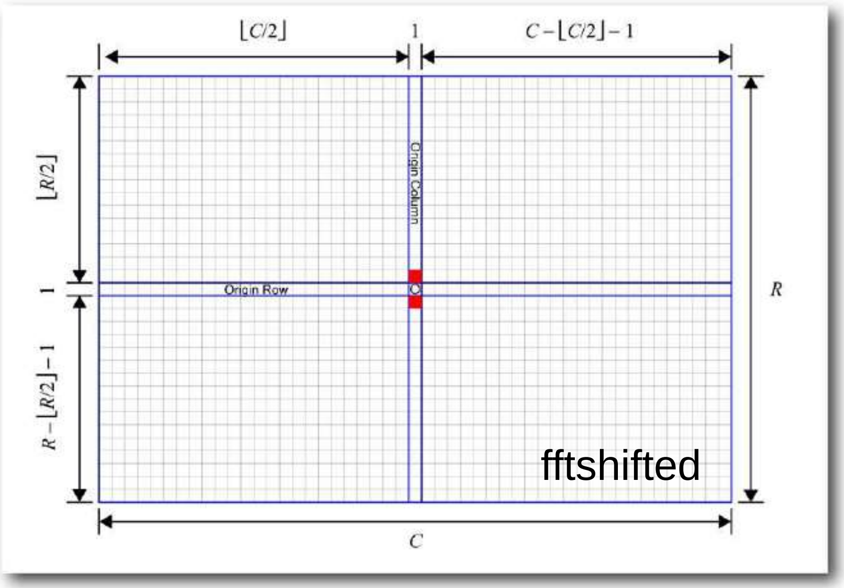
Spatial Domain

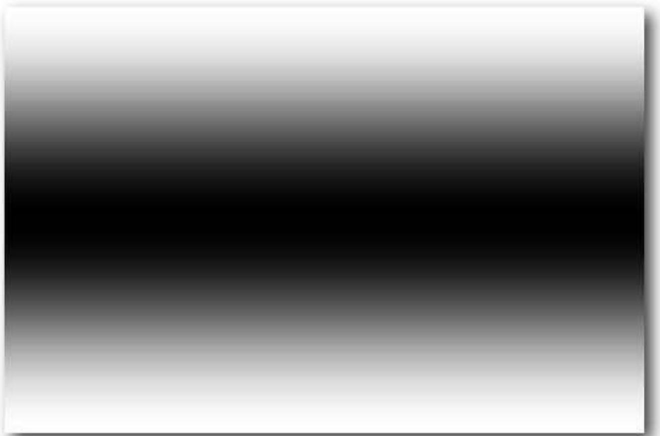


lowest-possible-frequency horizontal sinusoid

Inverse FFTs of Impulses

"vertical" is the wavefront direction.

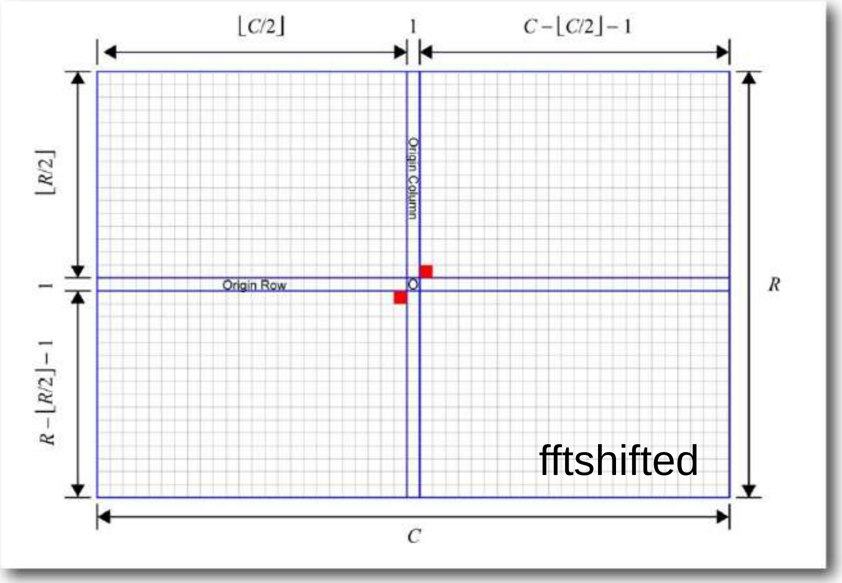


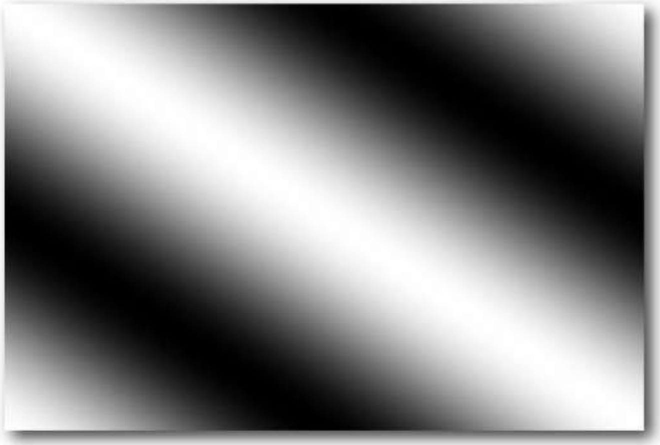


lowest-possible-frequency vertical sinusoid

Inverse FFTs of Impulses

"negative diagonal" is the wavefront direction.

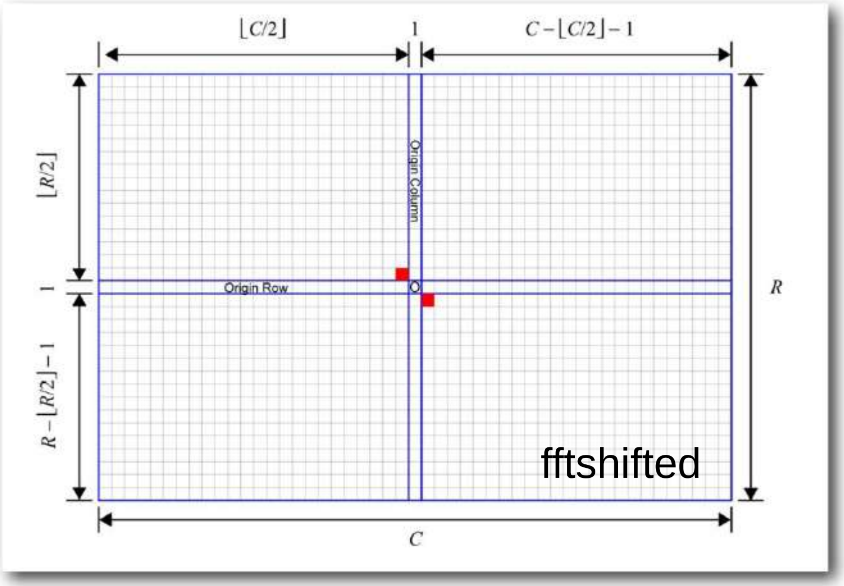


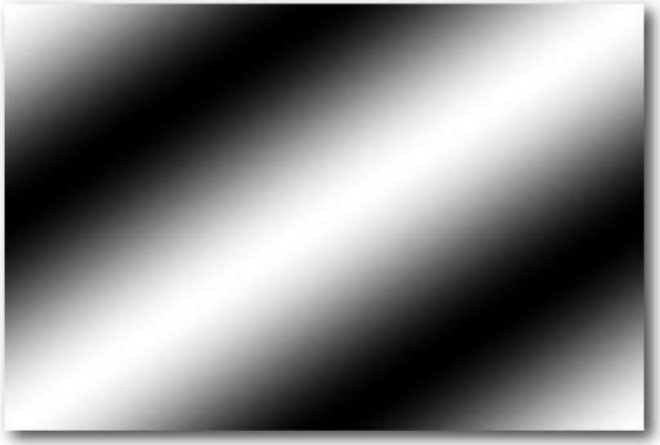


lowest-possible-frequency negative diagonal sinusoid

Inverse FFTs of Impulses

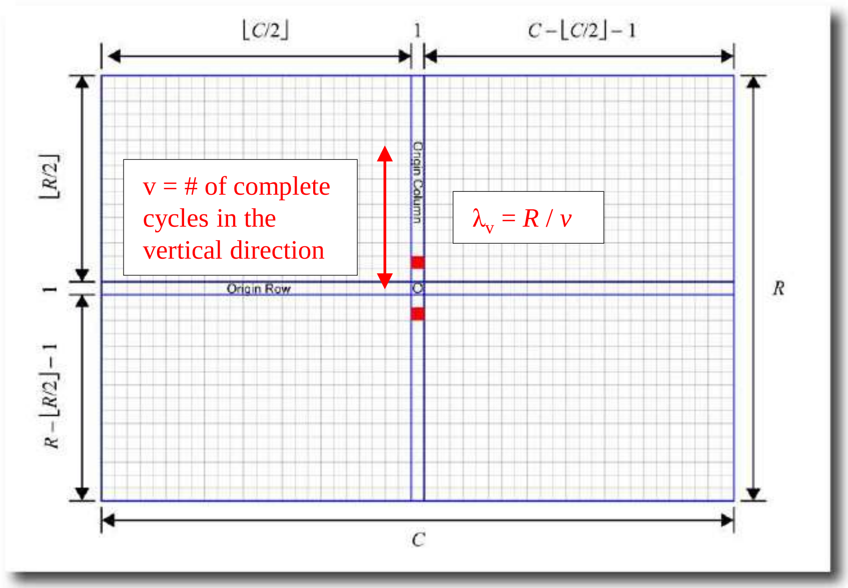
"positive diagonal" is the wavefront direction.

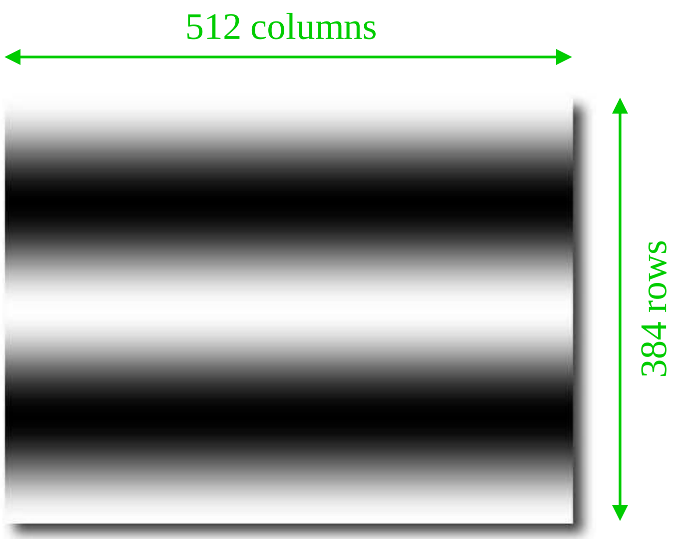




lowest-possible-frequency positive diagonal sinusoid

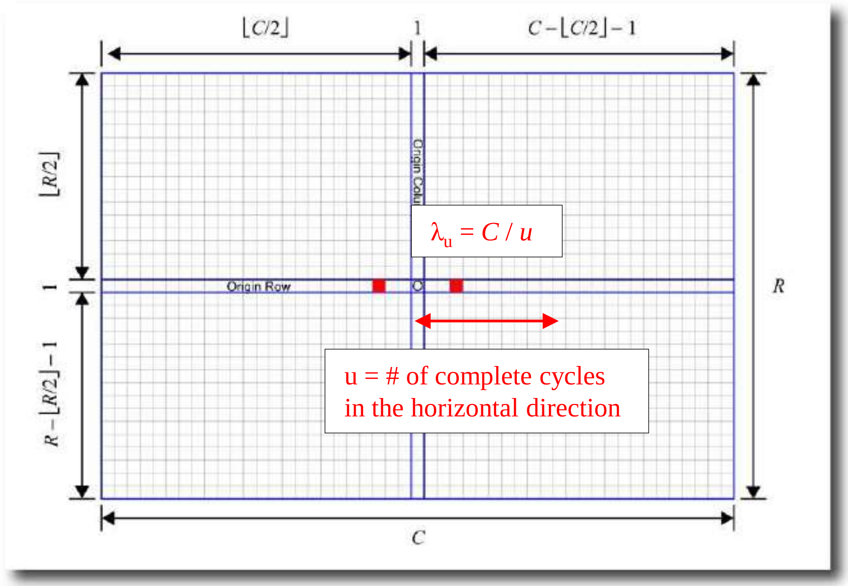
Frequencies and Wavelengths in the Fourier Plane

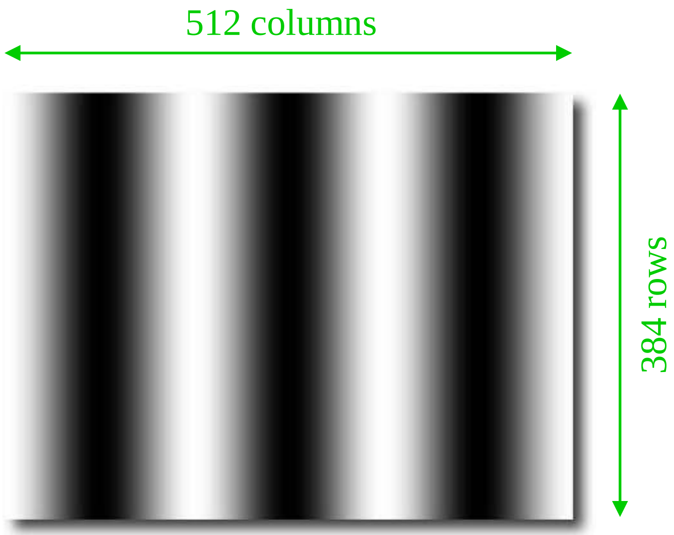




frequencies: $(u,v) = (0,2)$; wavelength: $\lambda_v = 192$

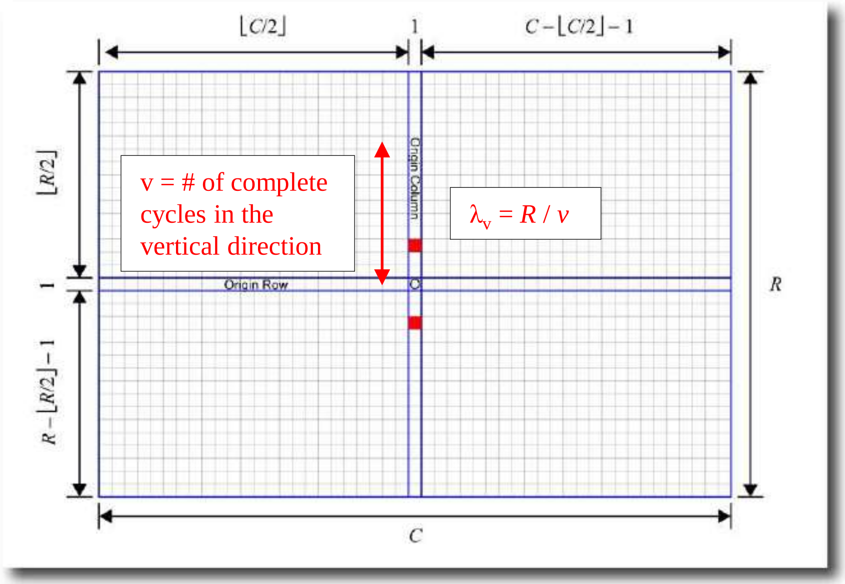
Frequencies and Wavelengths in the Fourier Plane

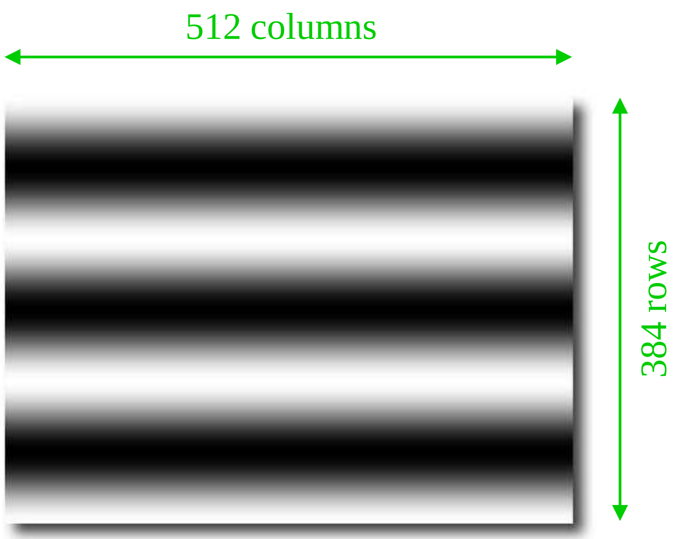




frequencies: $(u,v) = (3,0)$; wavelength: $\lambda_u = 170\frac{2}{3}$

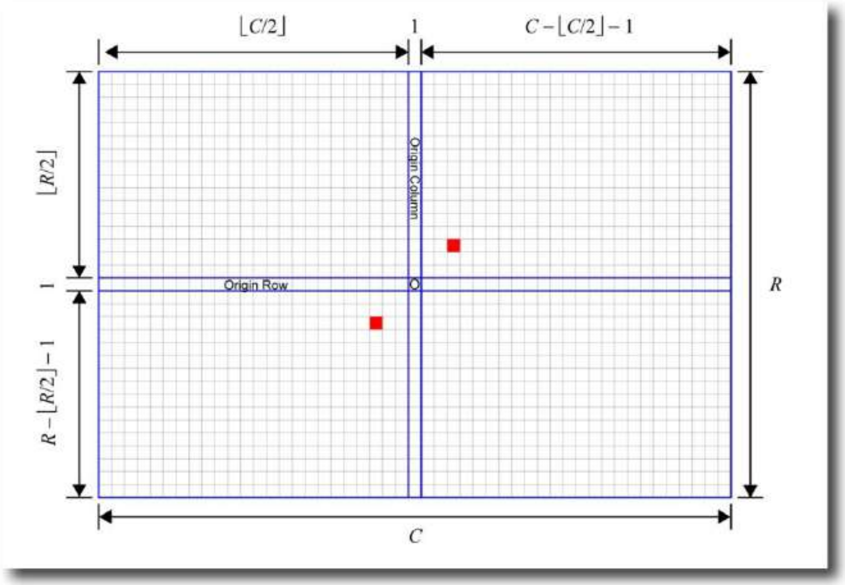
Frequencies and Wavelengths in the Fourier Plane

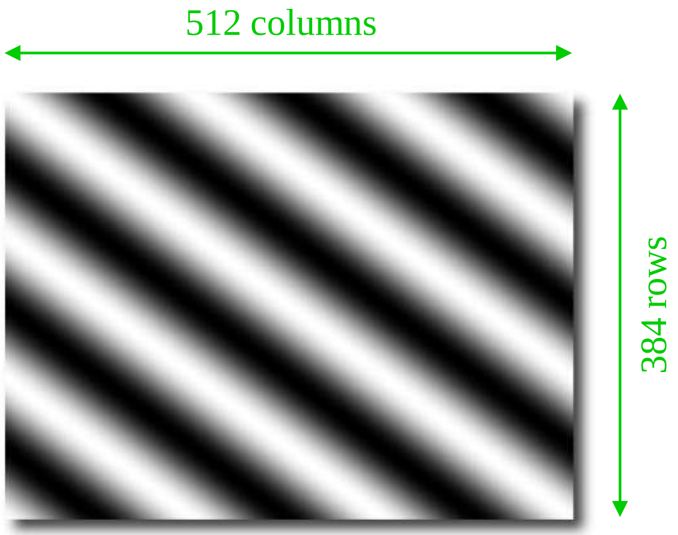




frequencies: $(u,v) = (0,3)$; wavelength: $\lambda_v = 128$

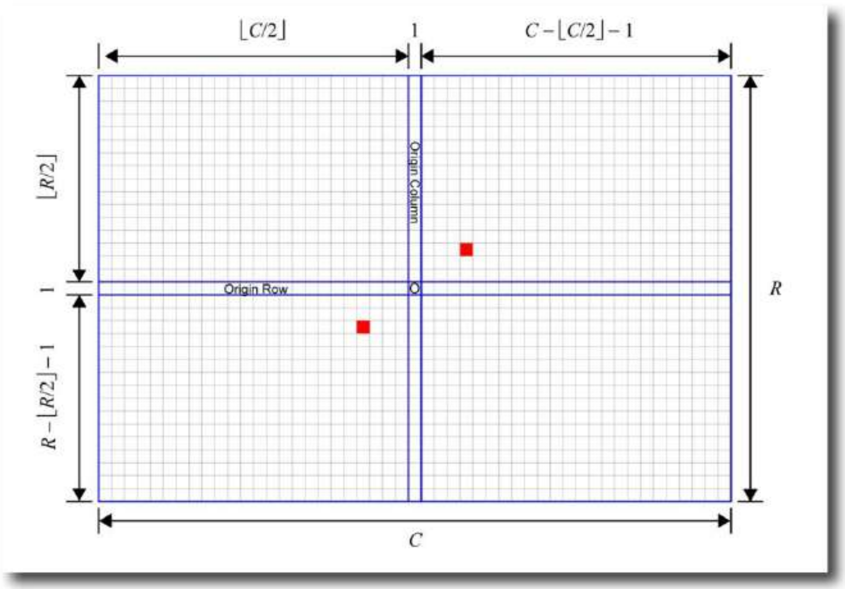
Frequencies and Wavelengths in the Fourier Plane

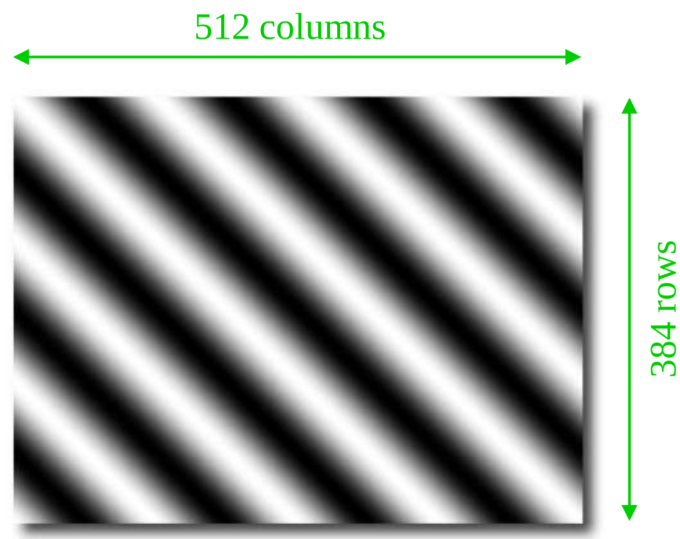




frequencies: $(u,v) = (3,3)$; wavelengths: $(\lambda_u, \lambda_v) = (170\frac{2}{3},128)$

Frequencies and Wavelengths in the Fourier Plane

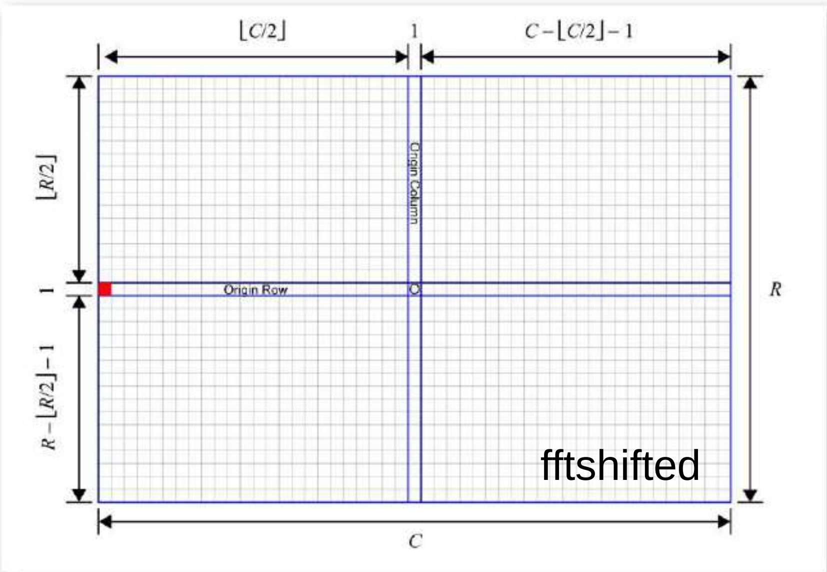


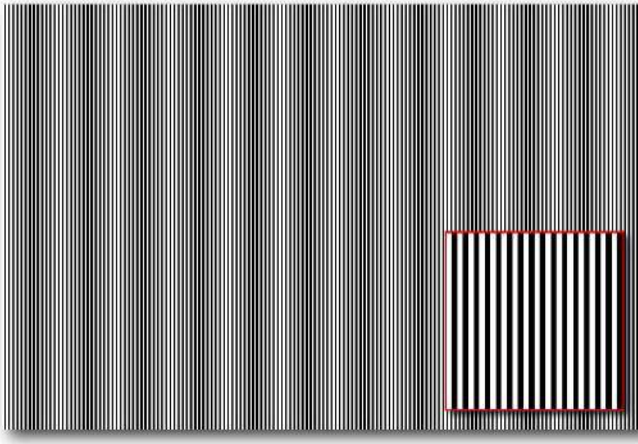


frequencies: $(u,v) = (4,3)$; wavelengths: $(\lambda_u, \lambda_v) = (128,128)$

Inverse FFTs of Impulses

"horizontal" is the wavefront direction.

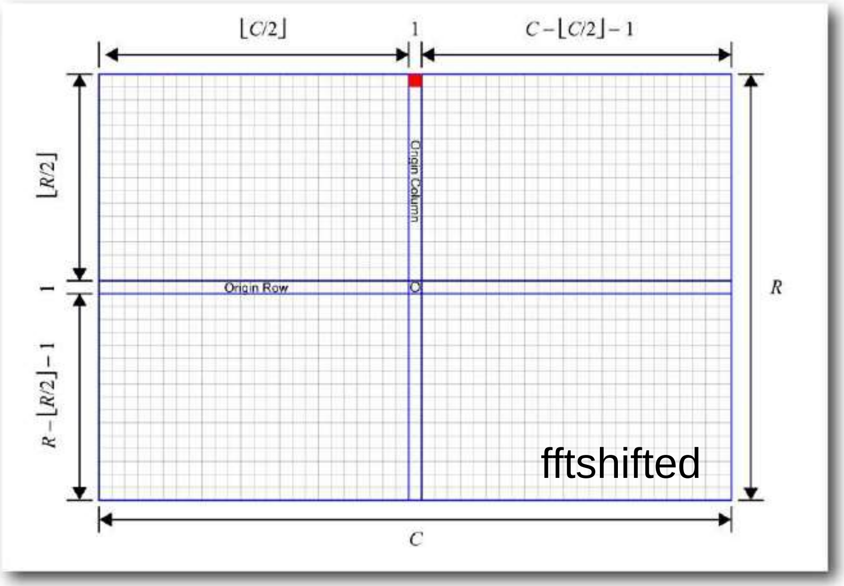


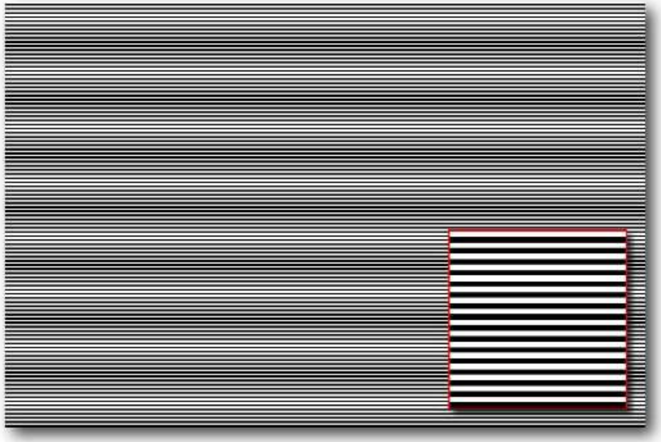


highest-possible-frequency horizontal sinusoid (C is even)

Inverse FFTs of Impulses

"vertical" is the wavefront direction.

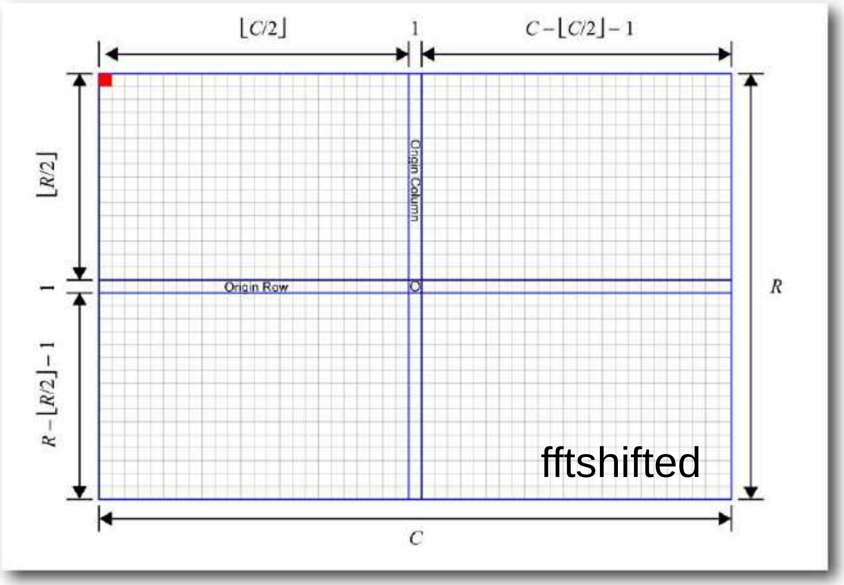


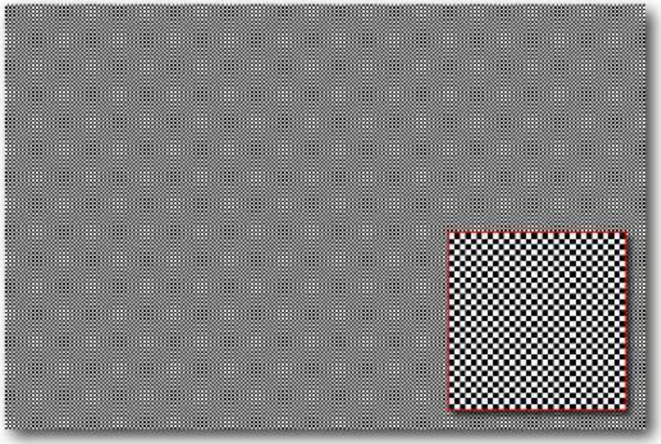


highest-possible-frequency vertical sinusoid (*R* is even)

Inverse FFTs of Impulses

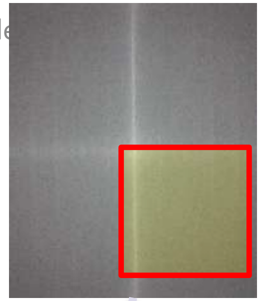
a checker-board pattern.

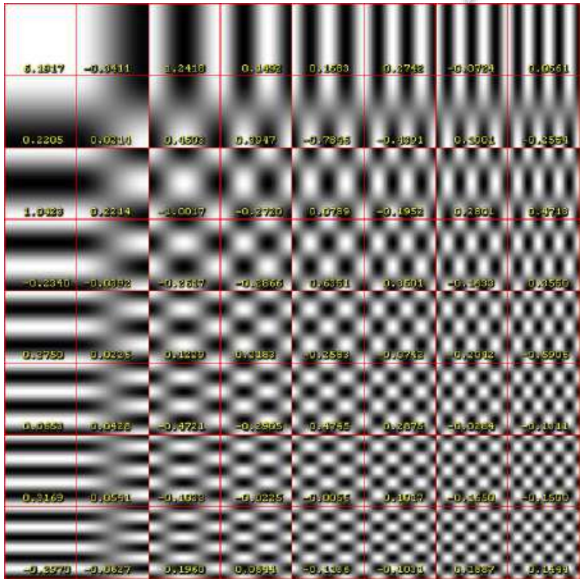




highest-possible-freq horizontal+vertical sinusoid (R & C even)

Example: build the image of an ‚A'



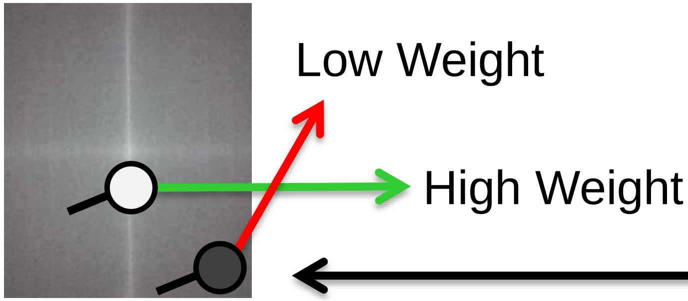


\+ 6.192 ×

Final Image

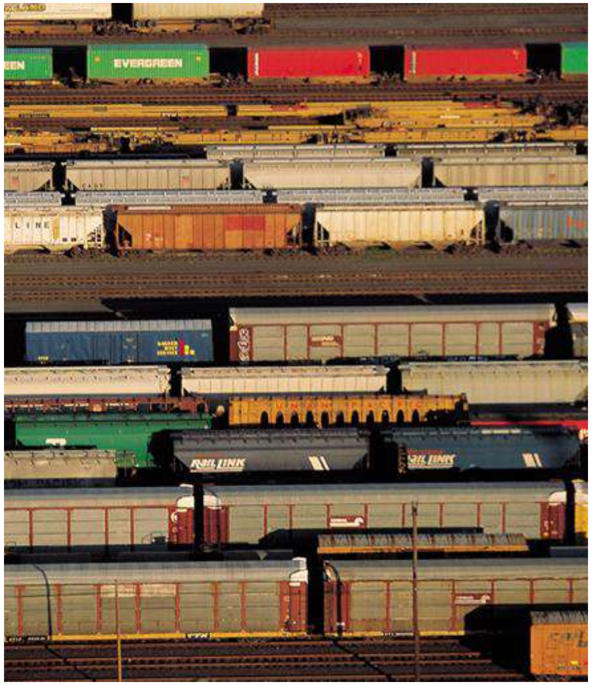
Weighted 2D functions

Original 2D functions

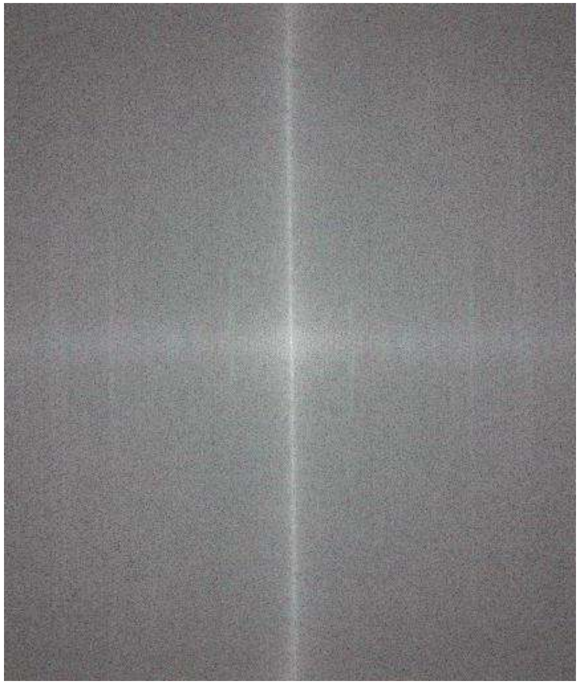


2D functions
(similar to Fourier impulses)
(each may correspond to a given pixel in the spectrum)

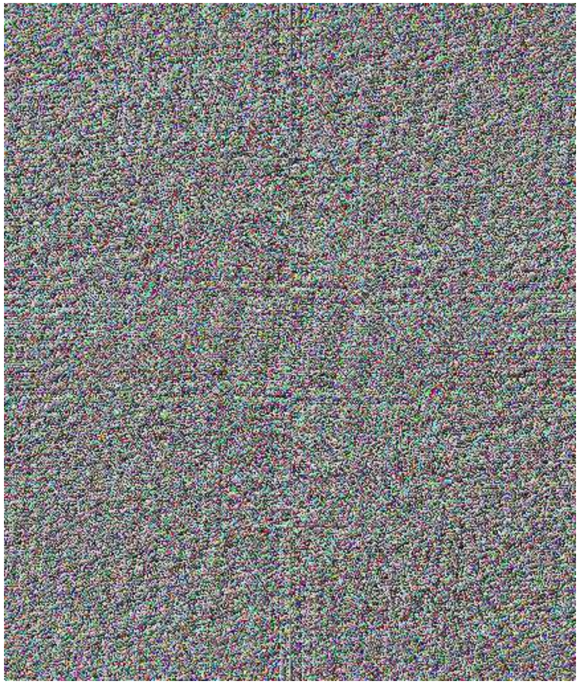
FT of an Image (Spectrum + Phase)



Image



Fourier spectrum



Phase ('Panning' of each sinusoid)

Examples

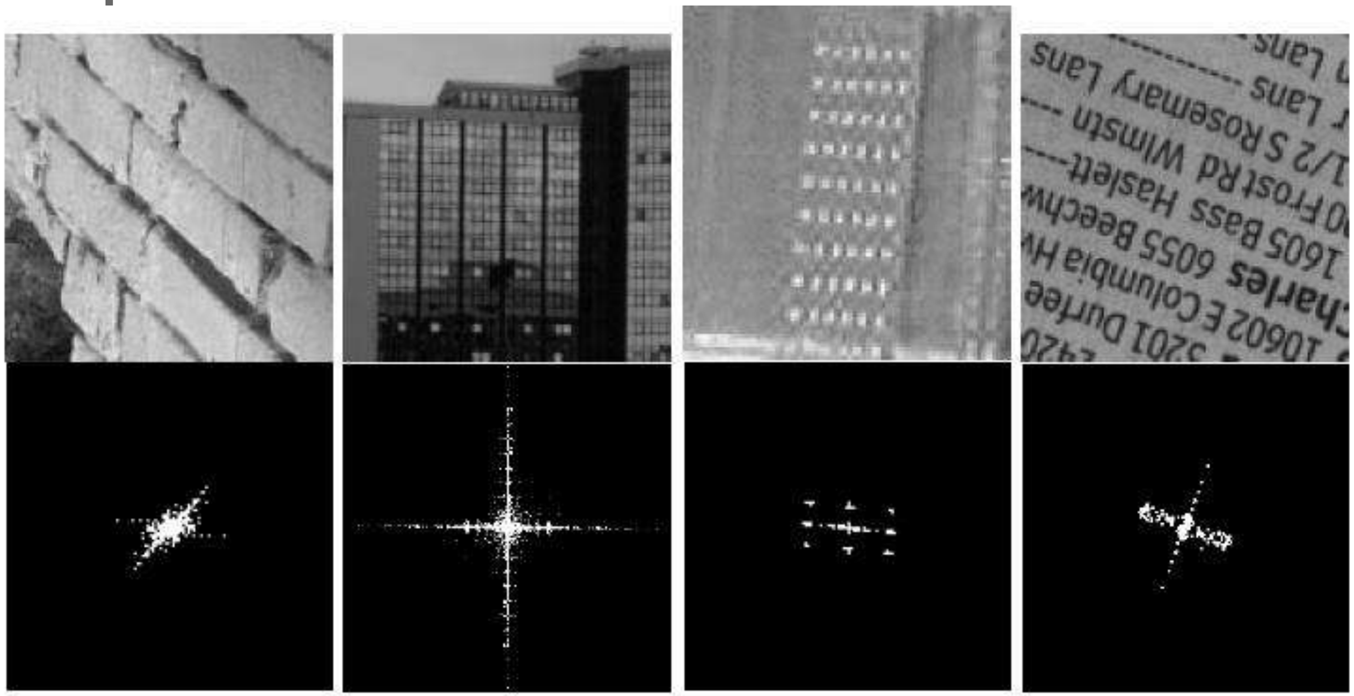
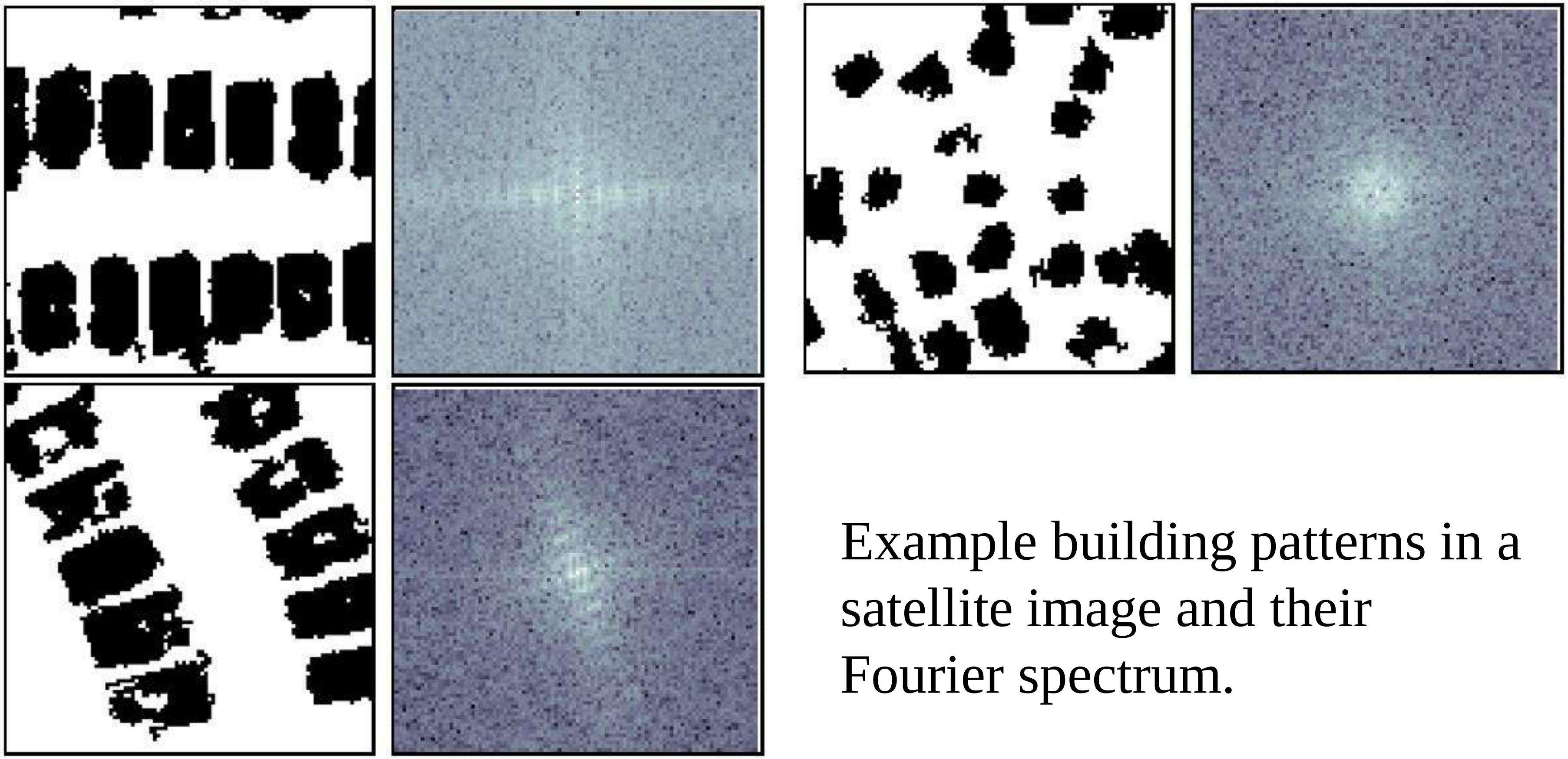


Figure 5.42: Four images (above) and their power spectrums (below). The power spectrum of the brick texture shows energy in many sinusoids of many frequencies, but the dominant direction is perpendicular to the 6 dark seams running about 45 degrees with the X-axis. There is noticable energy at 0 degrees with the X axis, due to the several short vertical seams. The power spectrum of the building shows high frequency energy in waves along the X-direction and the Y-direction. The third image is an aerial image of an orchard: the power spectrum shows the rows and columns of the orchard and also the “diagnonal rows”. The far right image, taken from a phone book, shows high frequency power at about 60^o with the X-axis, which represents the texture in the lines of text. Energy is spread more broadly in the perpendicular direction also in order to model the characters and their spacing.

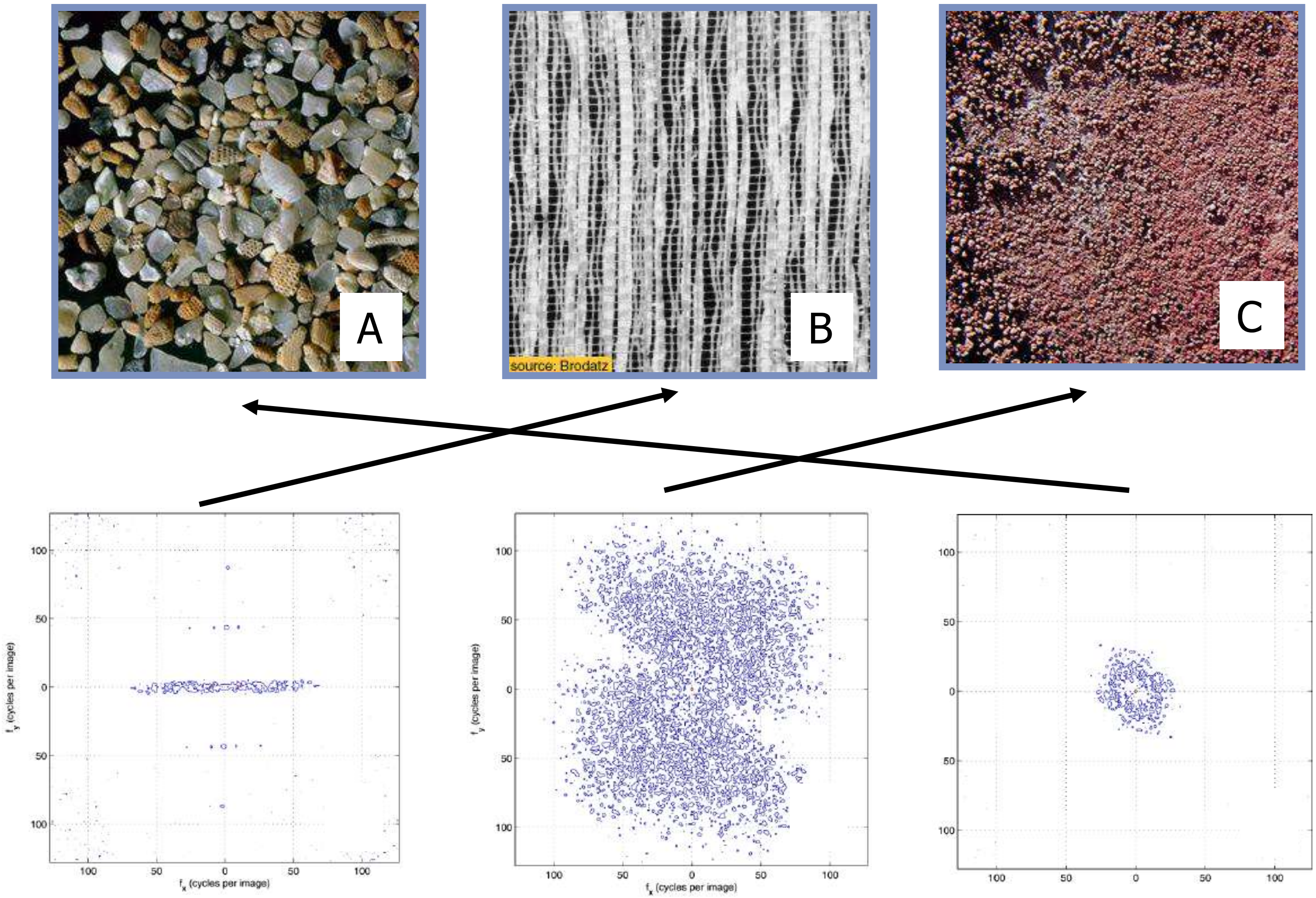
Adapted from Shapiro and Stockman

Examples



Example building patterns in a satellite image and their Fourier spectrum.

Test: Associate each image to its spectrum!



Adapted from Antonio Torralba

Filtering in the Frequency Domain

Blurring: Averaging / Lowpass Filtering

Blurring results from:

– Pixel averaging in the spatial domain:

  – Each pixel in the output is a weighted average of its neighbors.

  – Is a convolution whose weight matrix sums to 1.

– Lowpass filtering in the frequency domain:

  – High frequencies are diminished or eliminated

  – Individual frequency components are mul~~tiplied by~~ a nonincreasing function of ω such as $1/\omega = 1/\sqrt{(u^2+v^2)}$.

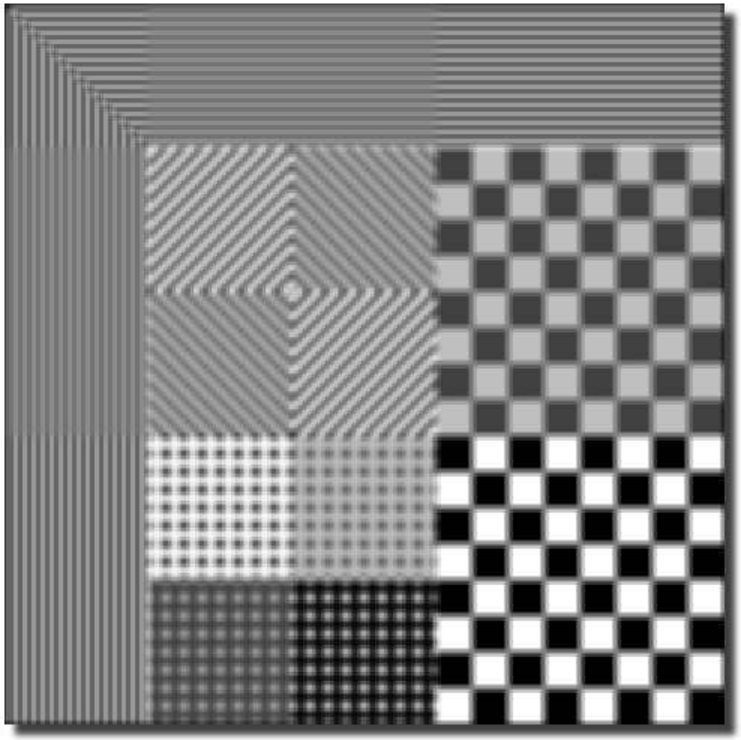
The values of the output image are all non-negative.

DLR

$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

First lecture (yesterday)

Convolution Examples: 5×5 Blur



Sharpening: Differencing / Highpass Filtering

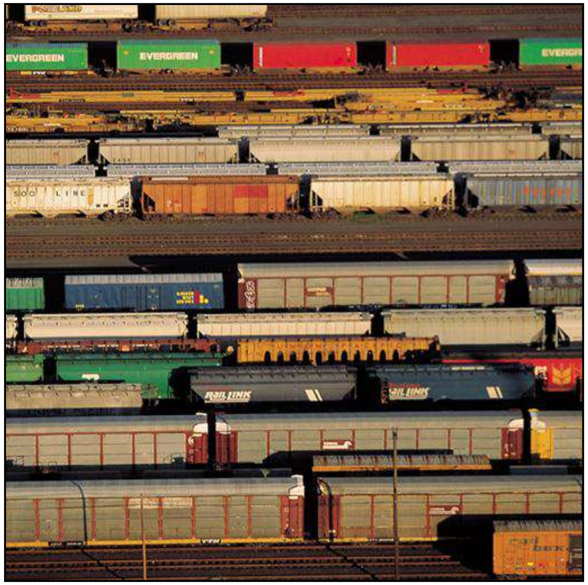
Sharpening results from adding to the image, a copy of itself that has been:

- Pixel-differenced in the spatial domain:
  - Each pixel in the output is a difference between itself and a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 0.
- Highpass filtered in the frequency domain:
  - High frequencies are enhanced or amplified.
  - Individual frequency components are multiplied by an increasing function of ω such as $\alpha\omega = \alpha\sqrt{(u^2+v^2)}$, where α is a constant.

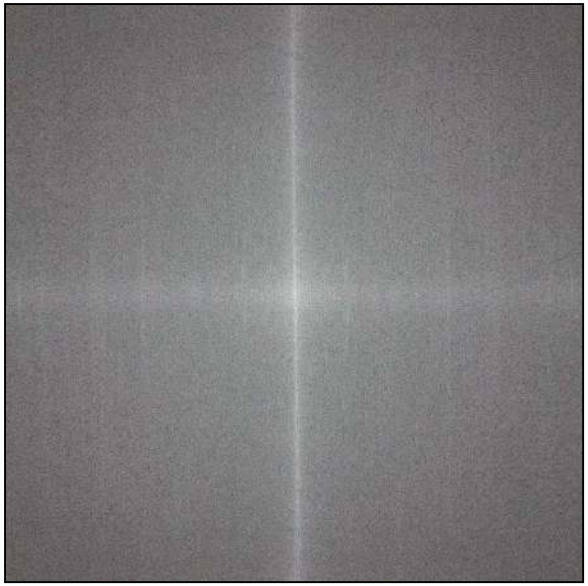
The values of the output image positive & negative.

Consider the image below:

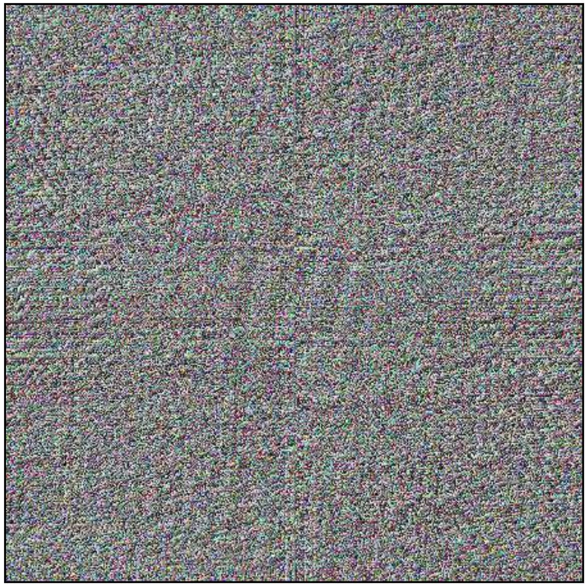
Power Spectrum and Phase of an Image



Original Image



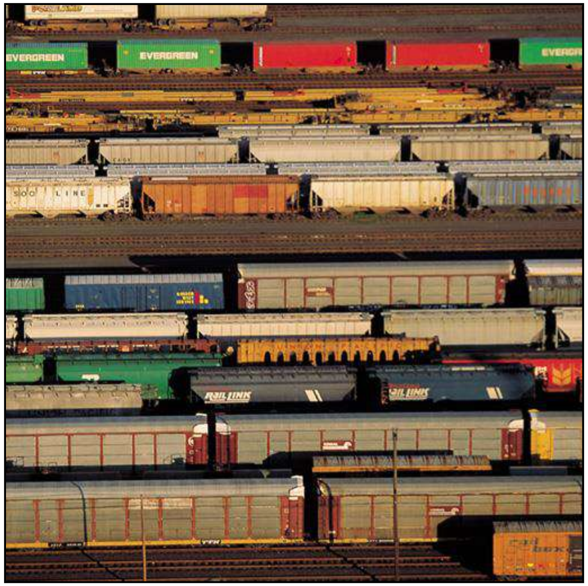
Power Spectrum



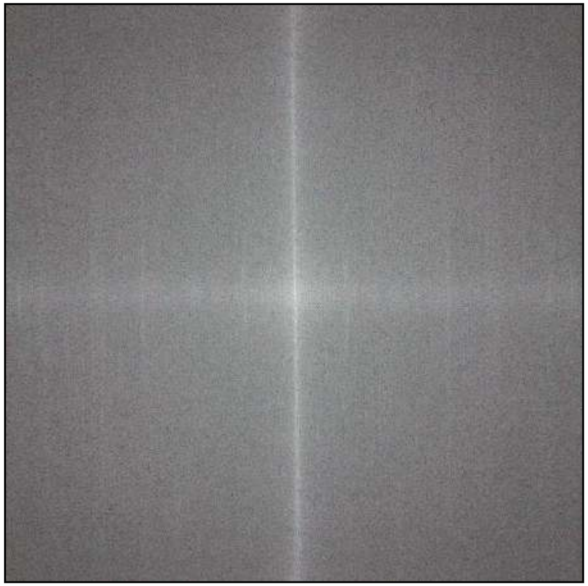
Phase

Ideal Lowpass Filter

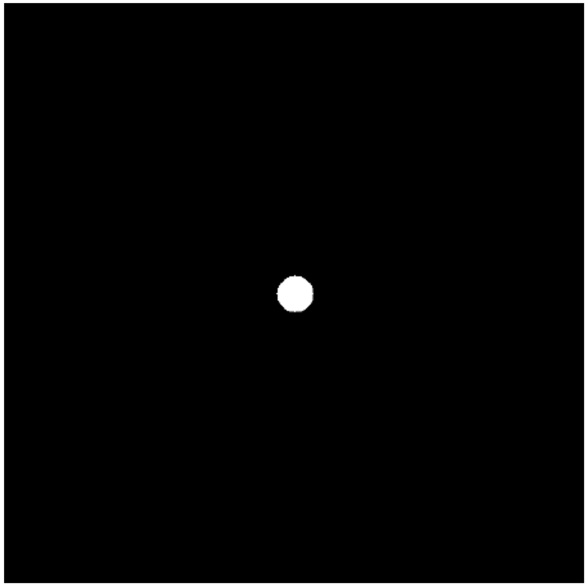
Image size: 512x512
FD filter radius: 16



Original Image



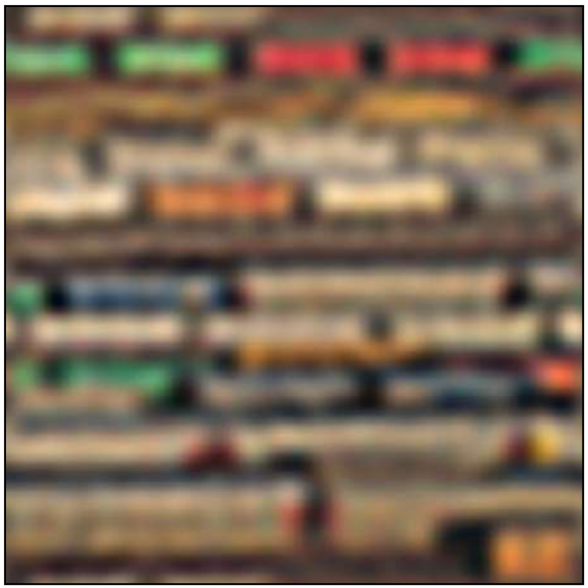
Power Spectrum



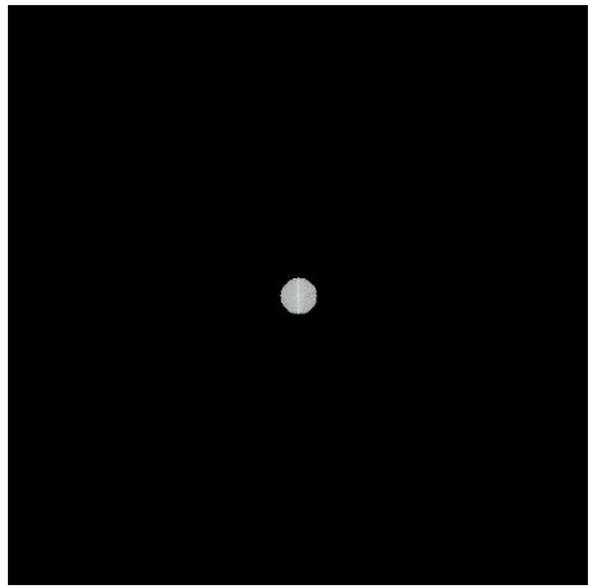
Ideal LPF in FD

Ideal Lowpass Filter

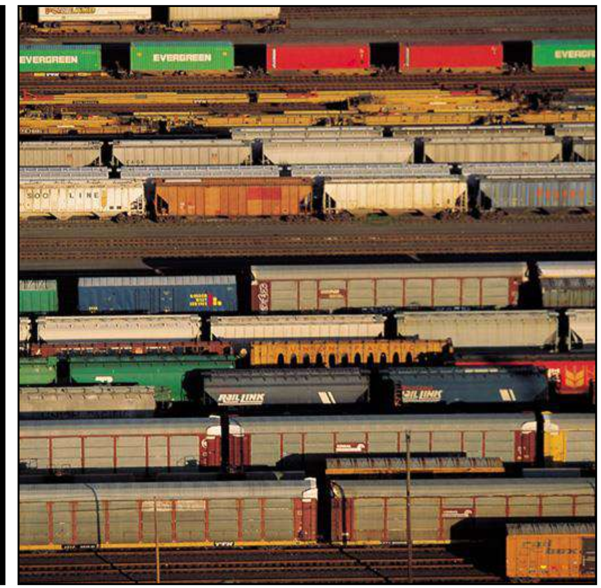
Image size: 512x512
FD filter radius: 16



Filtered Image



Filtered Power Spectrum



Original Image

DLR

Ideal Lowpass Filter

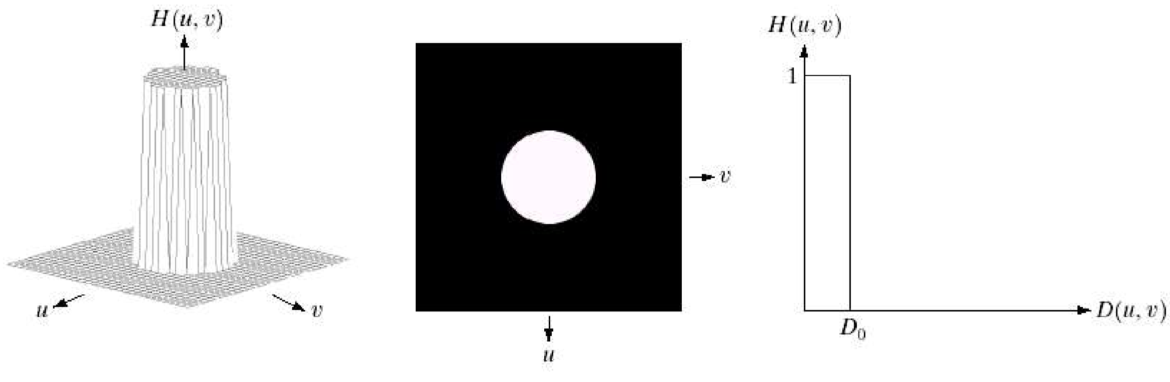
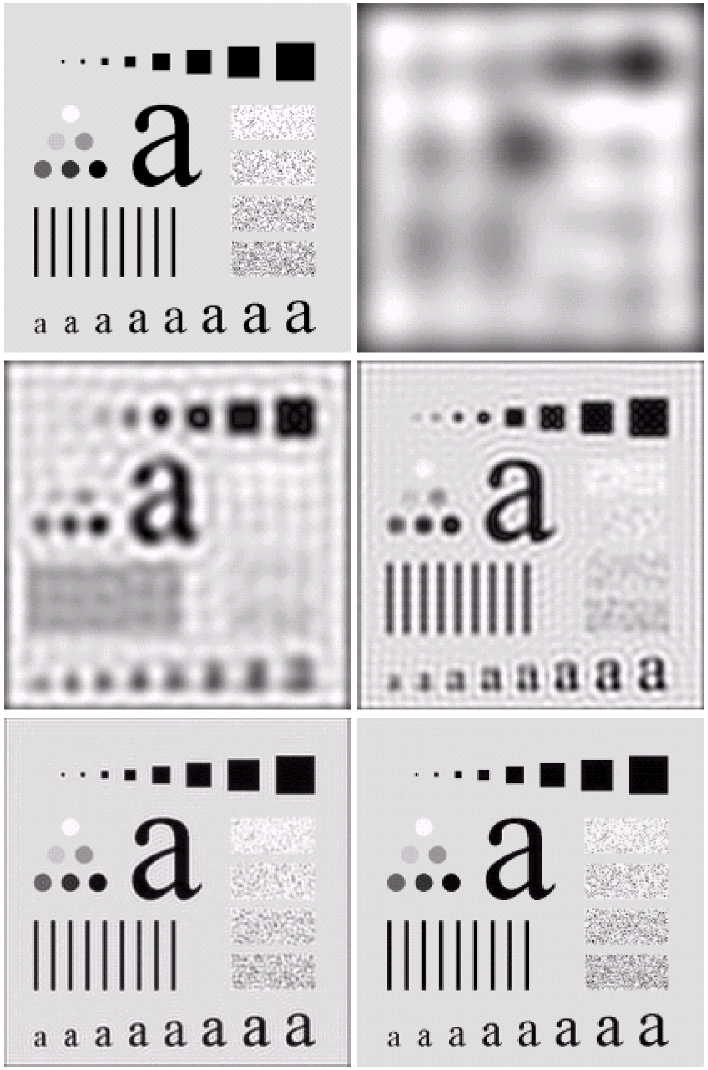


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

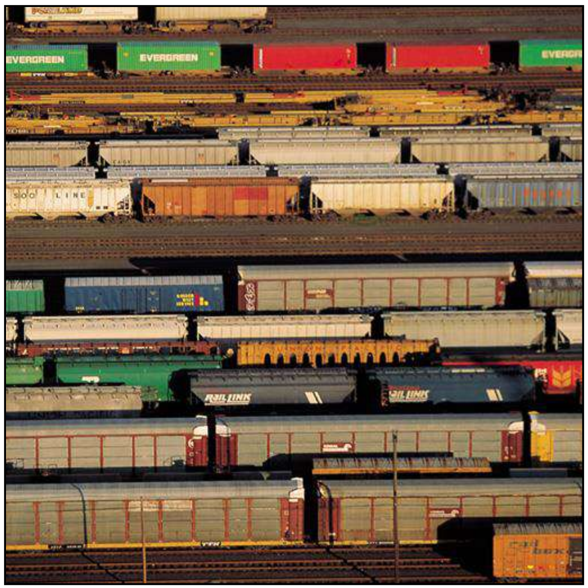


a b / c d / e f

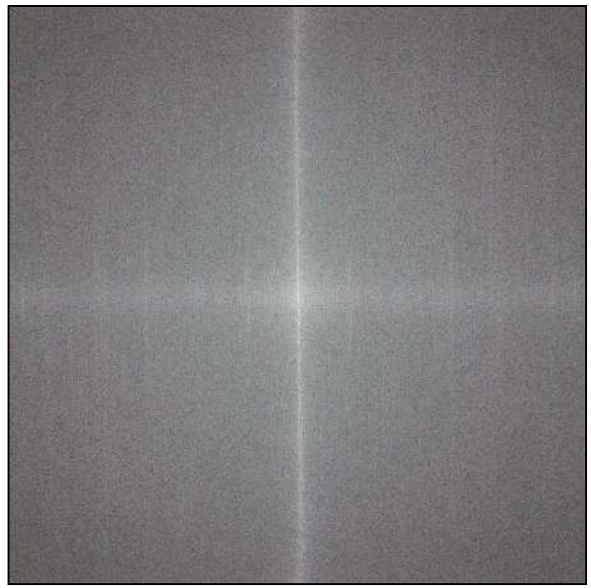
FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Ideal Highpass Filter

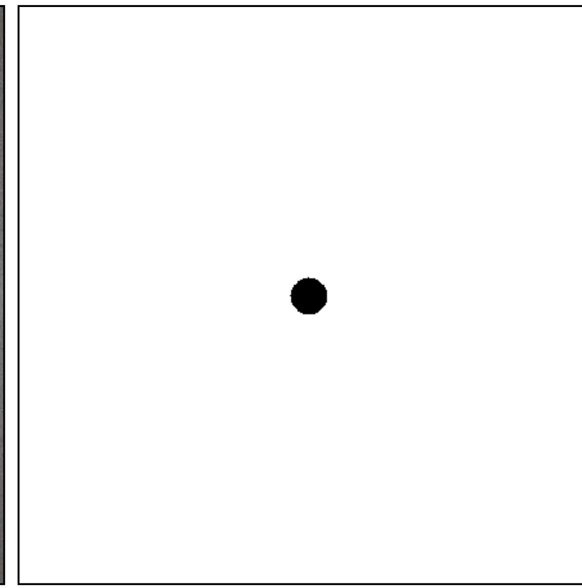
Image size: 512x512
FD notch radius: 16



Original Image



Power Spectrum



Ideal HPF in FD

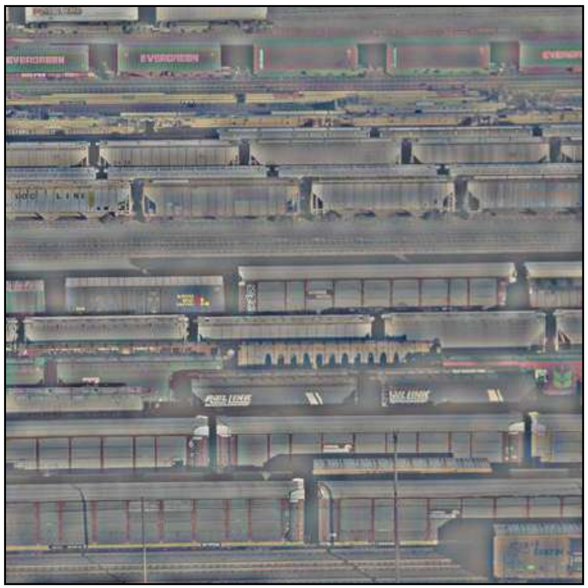
DLR

*signed image; 0 mapped to 128

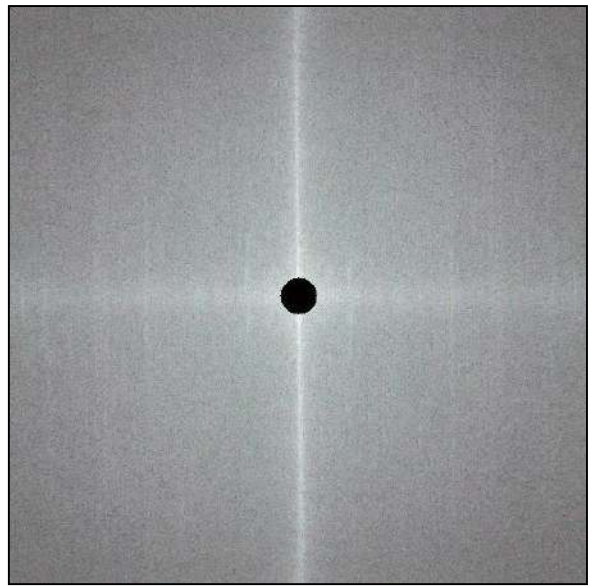
Why?

Ideal Highpass Filter

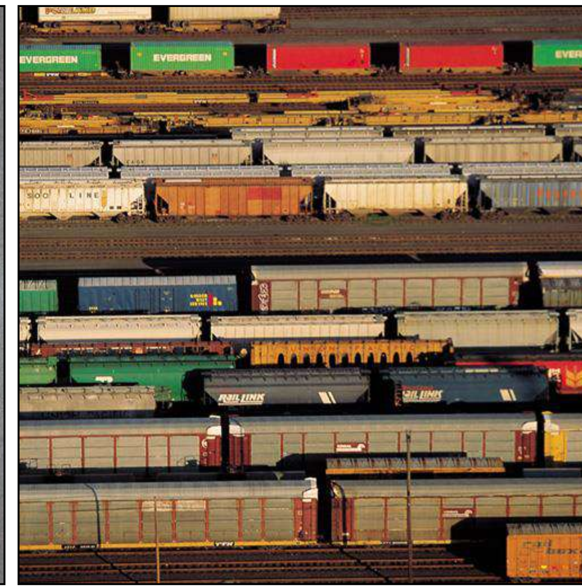
Image size: 512x512
FD notch radius: 16



Filtered Image*



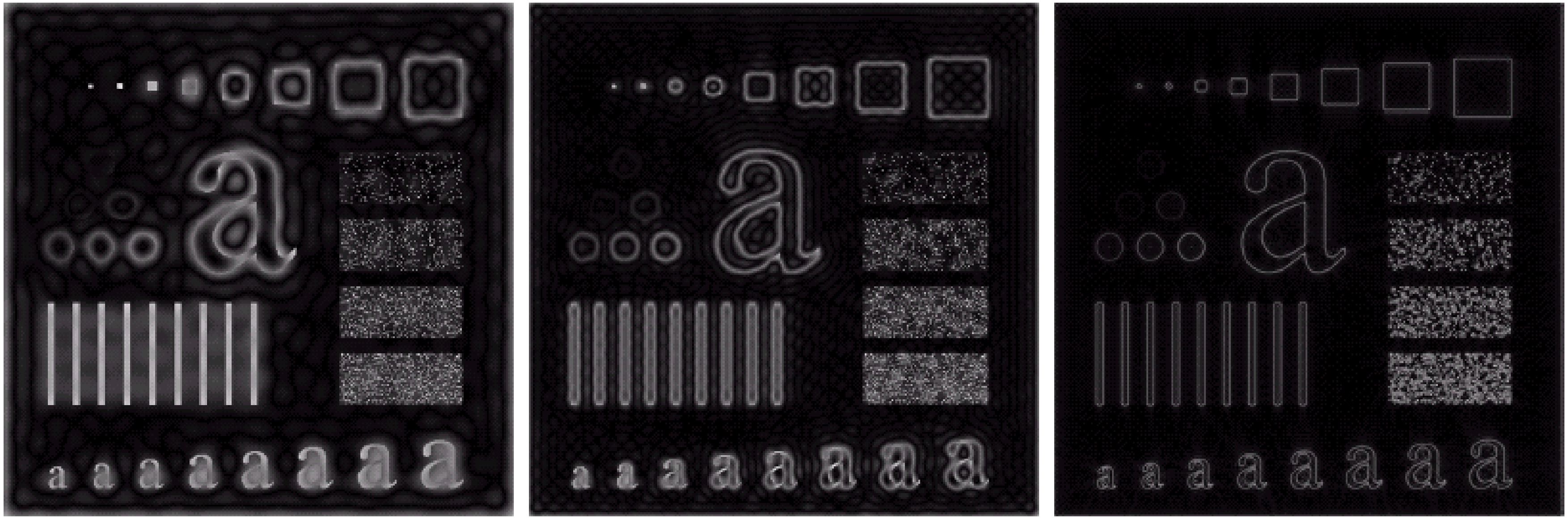
Filtered Power Spectrum



Original Image

DLR

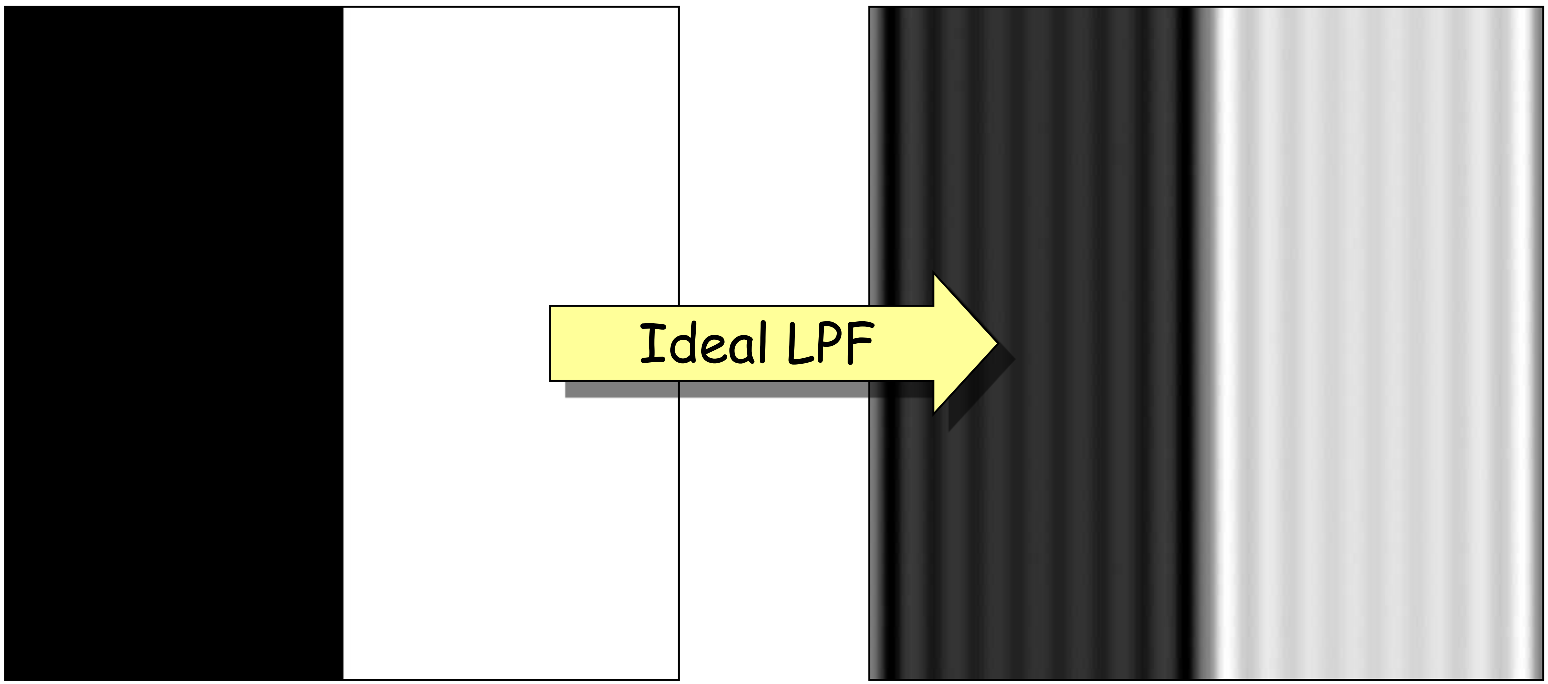
Ideal Highpass Filter



a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

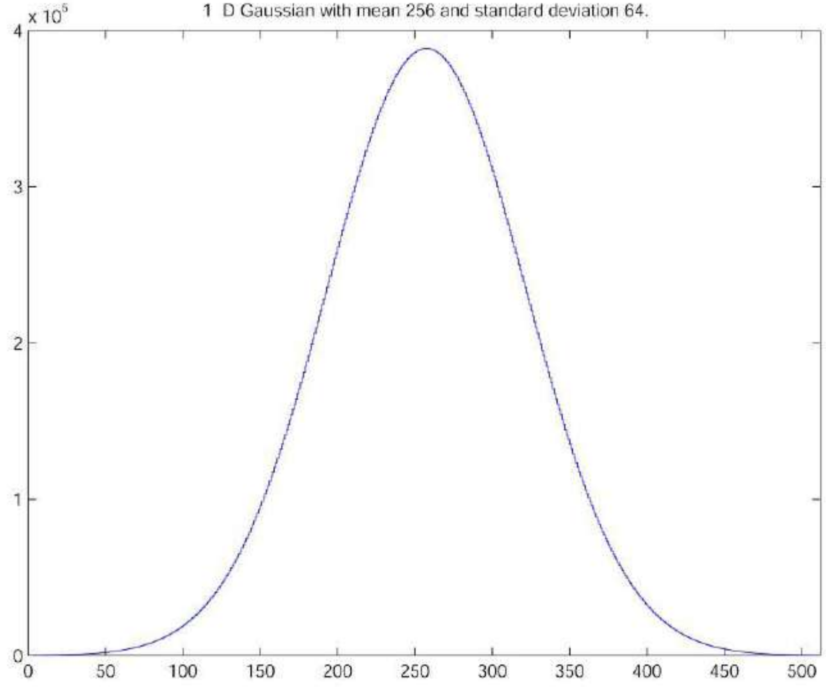
Ideal Filters Do Not Produce Ideal Results



Blurring the image above w/ an ideal lowpass filter…

…distorts the results with ringing or ghosting.

Optimal Filter: the Gaussian



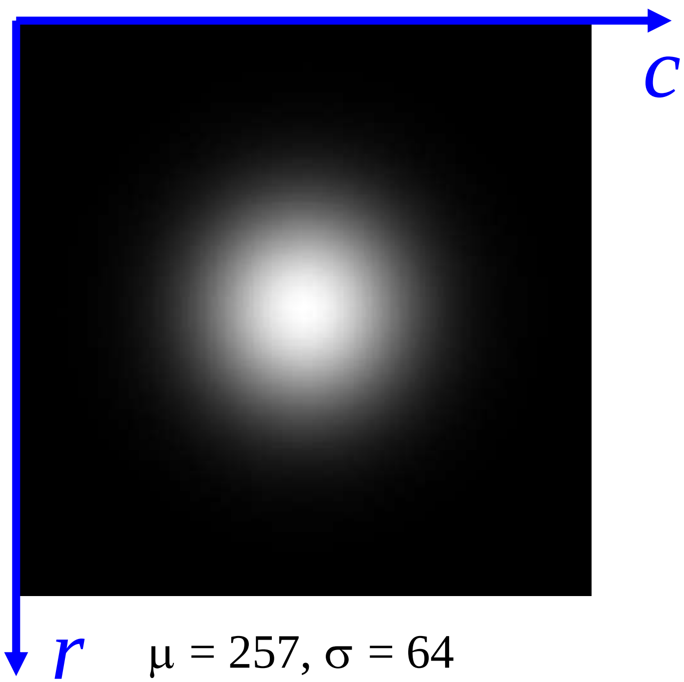
One-Dimensional Gaussian

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

DLR

Two-Dimensional Gaussian

$R = 512, C = 512$



$\mu = 257, \sigma = 64$

If μ and σ are different for r & c...

$$g(r,c) = g(r)g(c)$$

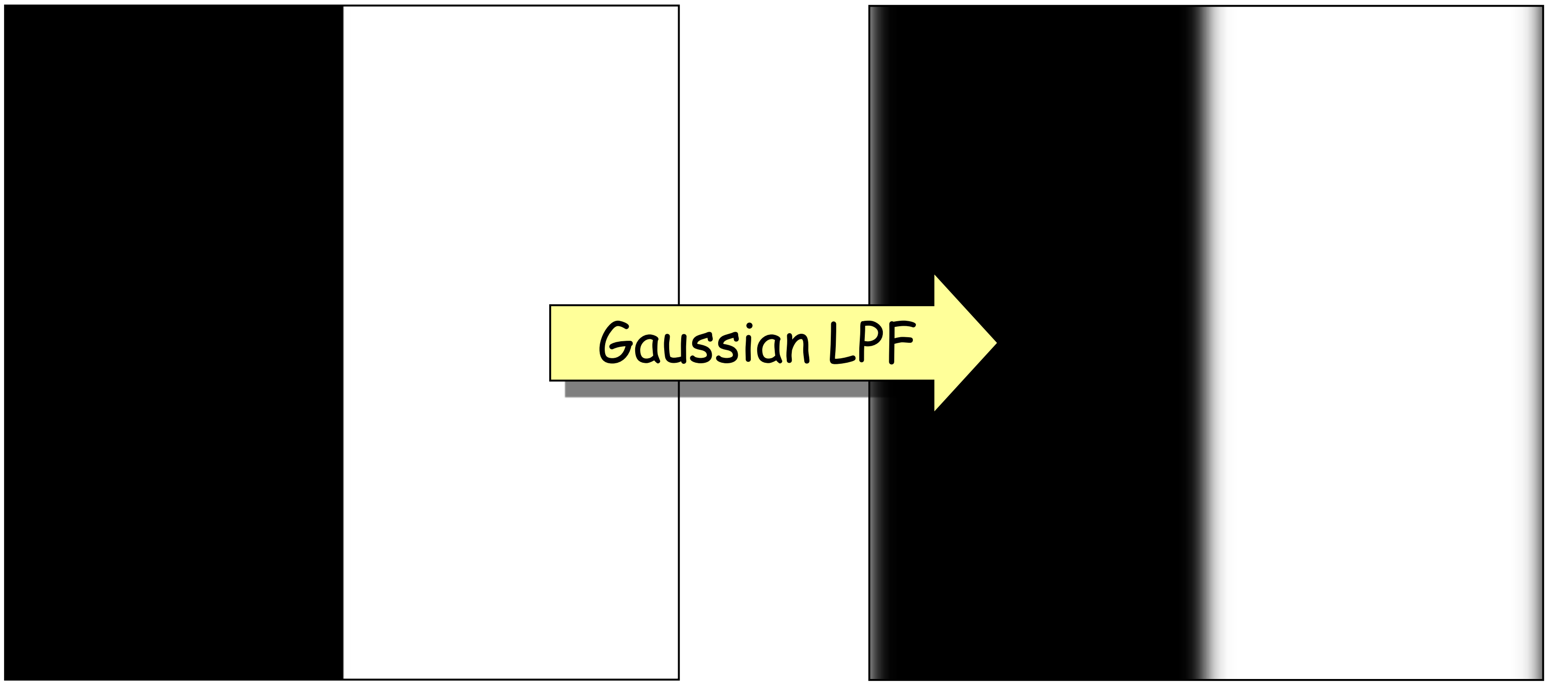
$$= \frac{1}{\sigma_r \sigma_c 2\pi} e^{-\frac{(r-\mu_r)^2}{2\sigma_r^2} - \frac{(y-\mu_c)^2}{2\sigma_c^2}}$$

$$= \frac{1}{\sigma_r \sigma_c 2\pi} e^{-\frac{\sigma_c^2 (x-\mu_r)^2 + \sigma_r^2 (y-\mu_c)^2}{2\sigma_r^2 \sigma_c^2}}$$

...or if μ and σ are the same for r & c.

$$g(r,c) = \frac{1}{\sigma^2 2\pi} e^{-\frac{(r-\mu)^2 + (c-\mu)^2}{2\sigma^2}}$$

Optimal Filter: The Gaussian

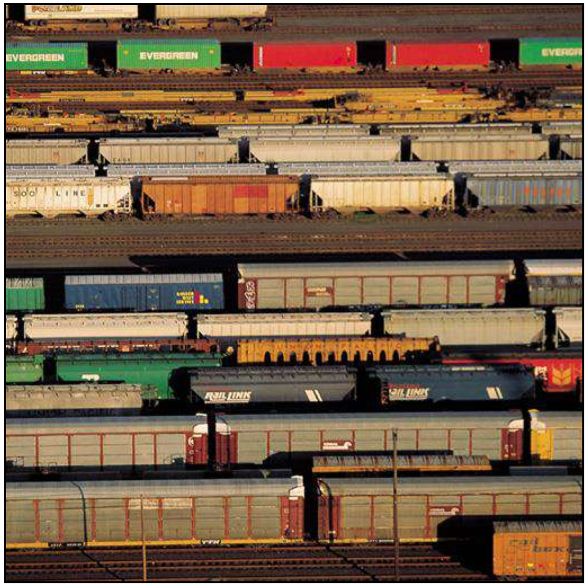


With a gaussian lowpass filter, the image above …

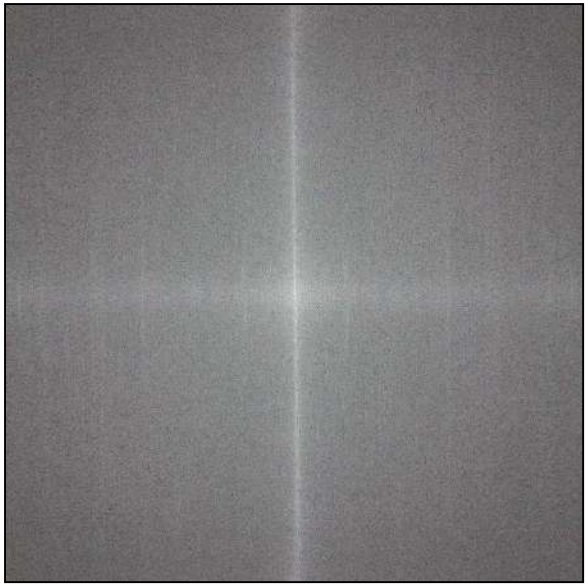
… is blurred without ringing or ghosting.

Gaussian Lowpass Filter

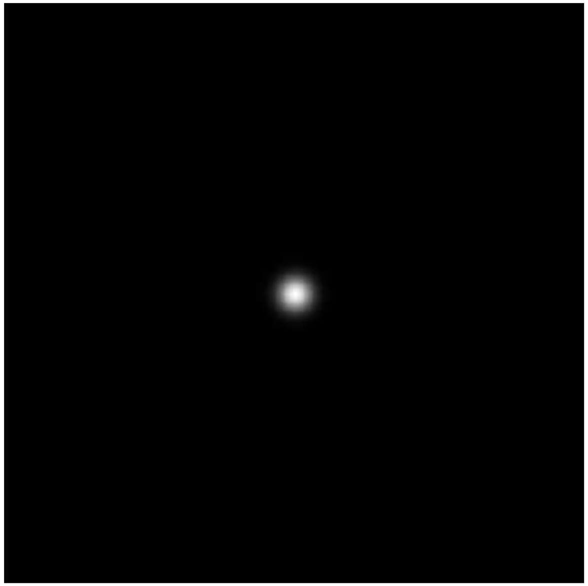
Image size: 512x512
SD filter sigma = 8



Original Image



Power Spectrum

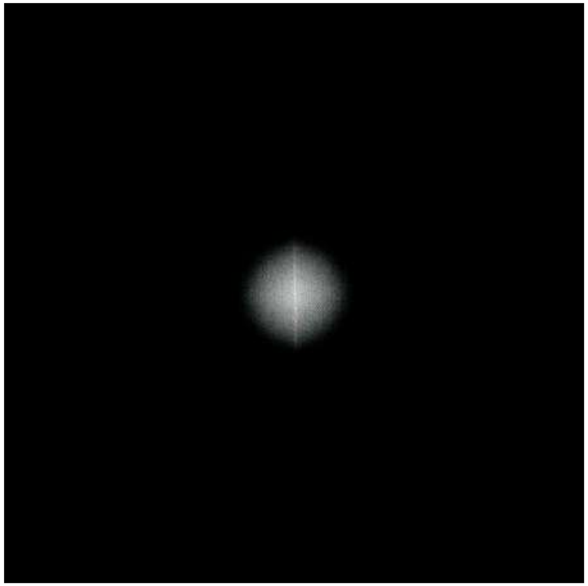


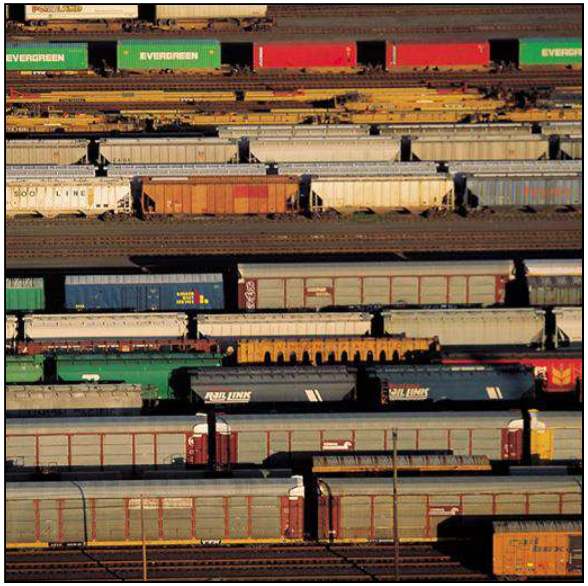
Gaussian LPF in FD

DLR

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 8



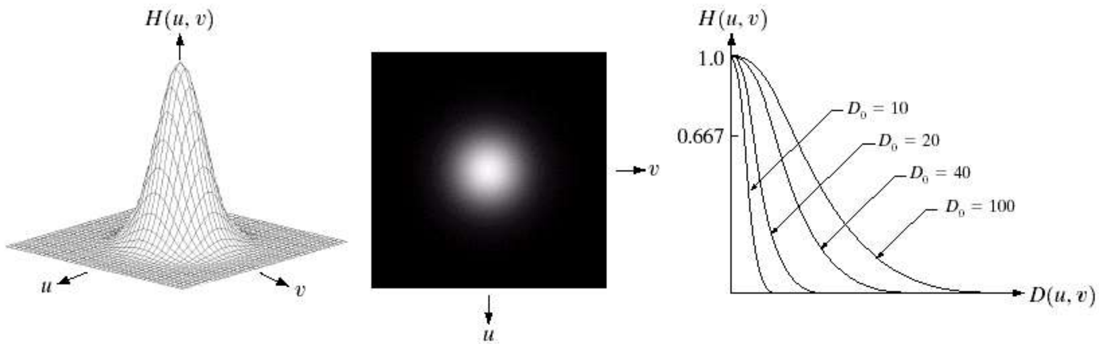


Filtered Image

Filtered Power Spectrum

Original Image

Gaussian Lowpass Filter



a b c

FIGURE 4.17 (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.

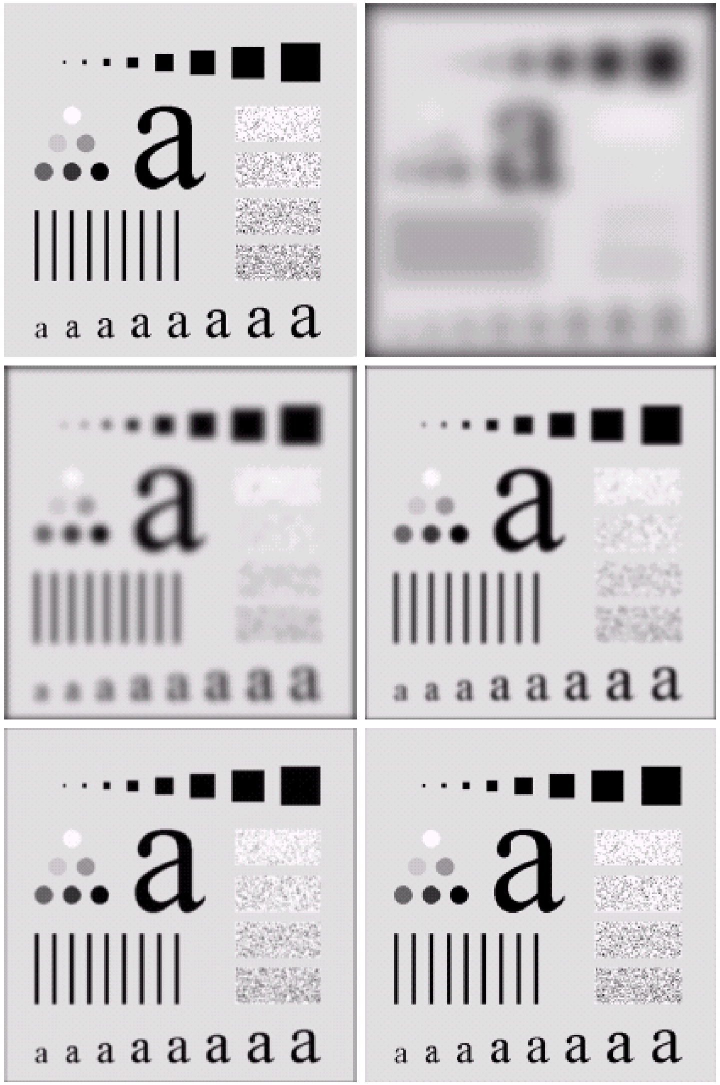
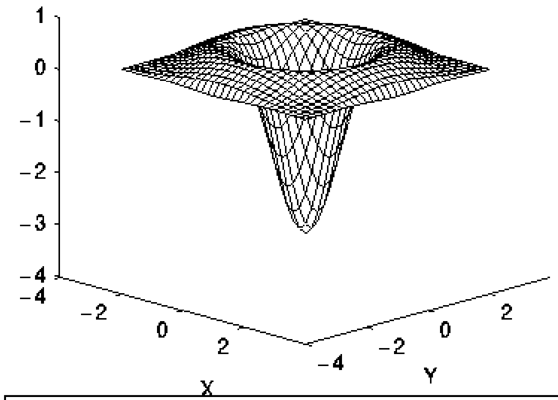
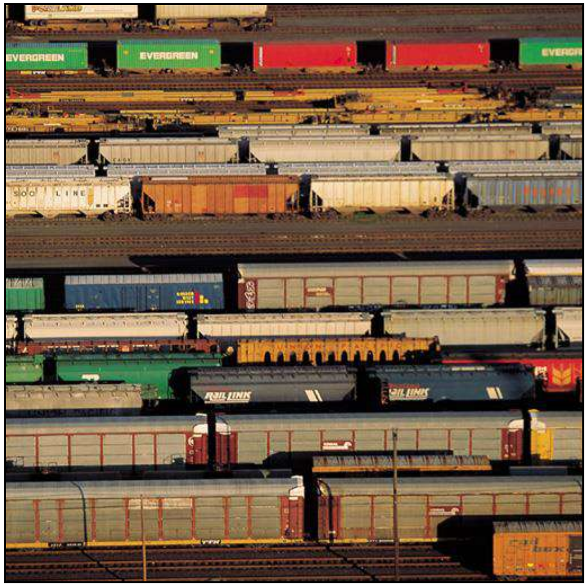


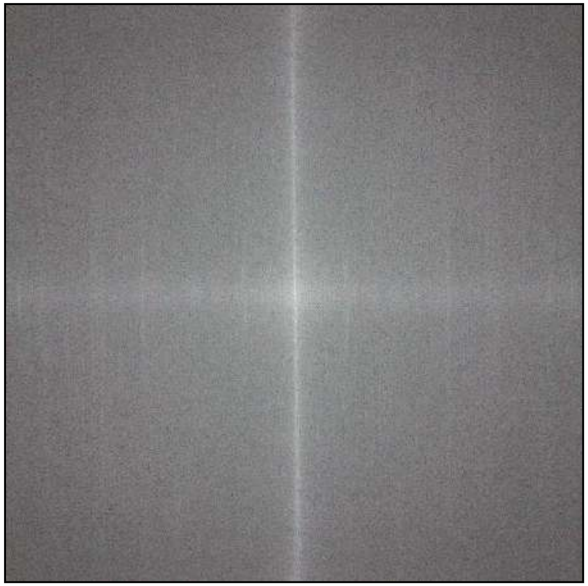
FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

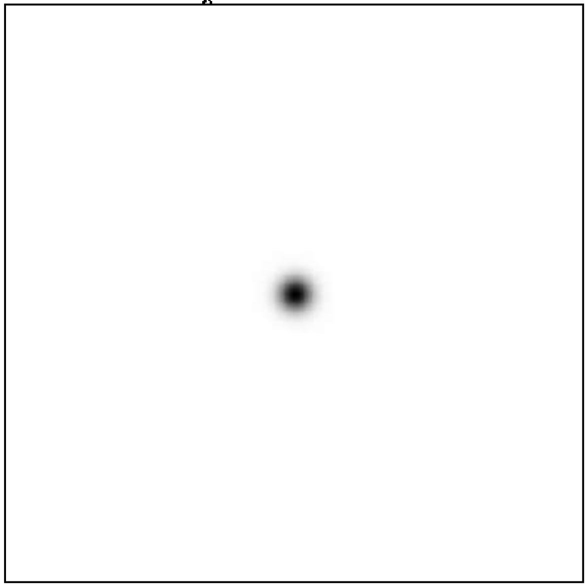
a b
c d
e f



Gaussian Highpass Filter







Original Image

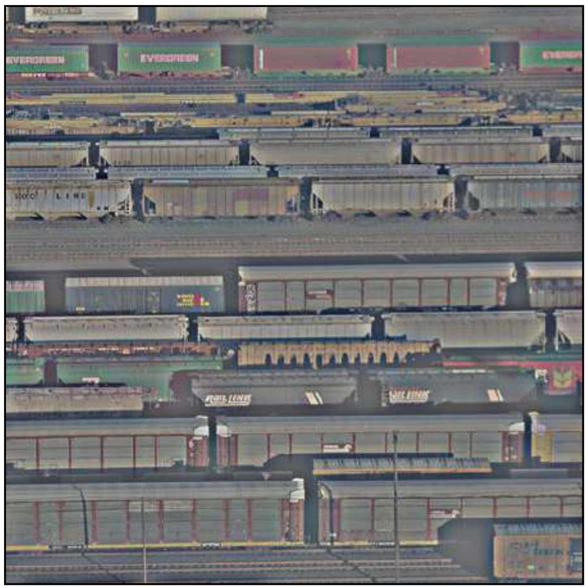
Fourier Spectrum

Gaussian HPF in FD

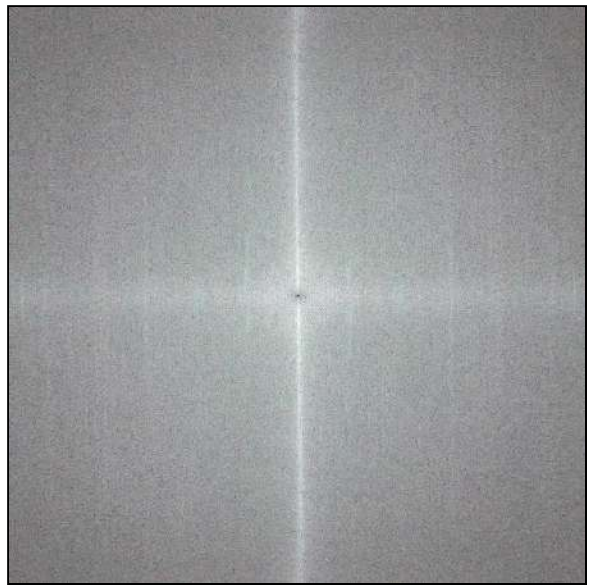
*signed image; 0 mapped to 128

Gaussian Highpass Filter

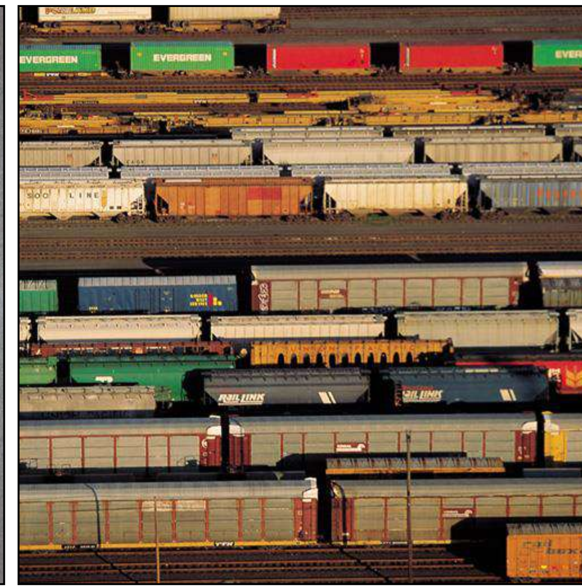
Image size: 512x512
FD notch sigma = 8



Filtered Image*



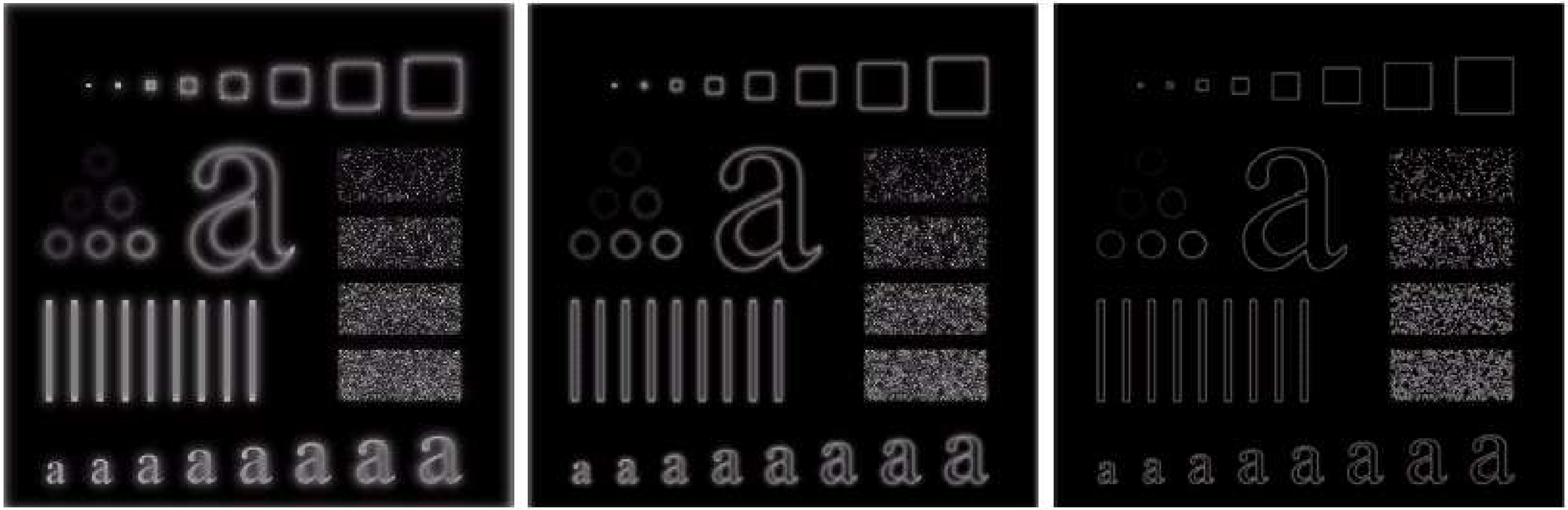
Filtered Power Spectrum



Original Image

DLR

Gaussian Highpass Filter



a b c

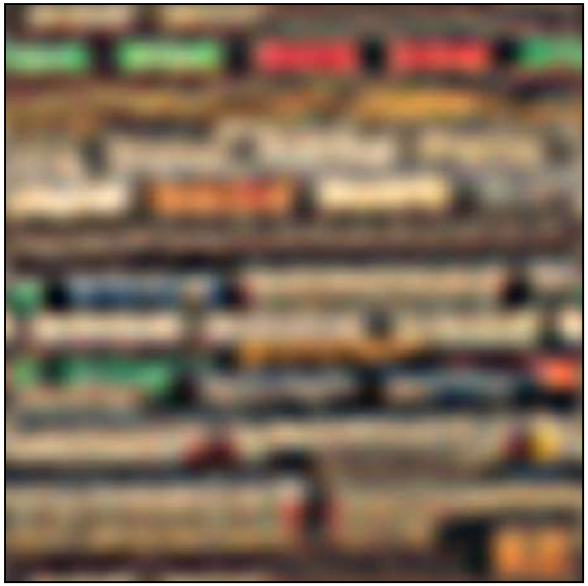
FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

DLR

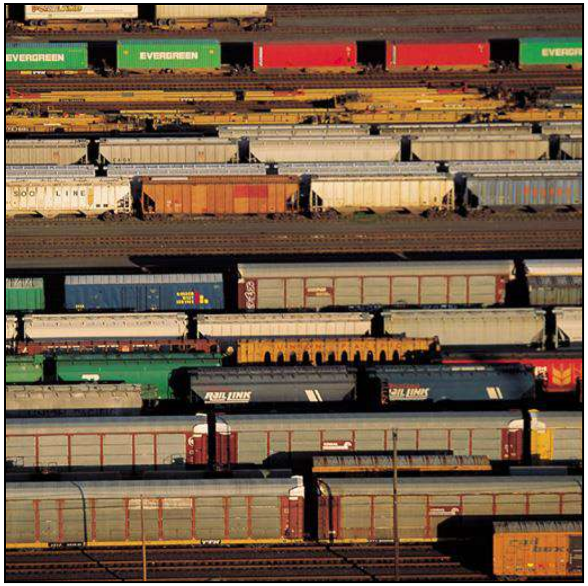
Adapted by Selim Aksoy from Gonzales and Woods

*signed image; 0 mapped to 128

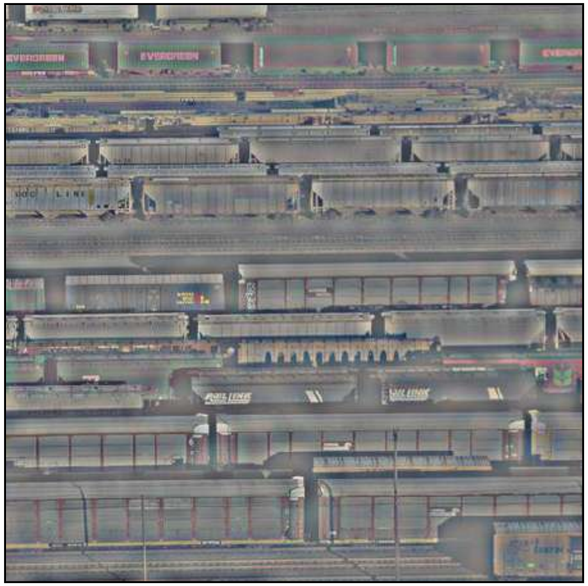
Comparison of Ideal and Gaussian Filters



Ideal LPF



Original Image

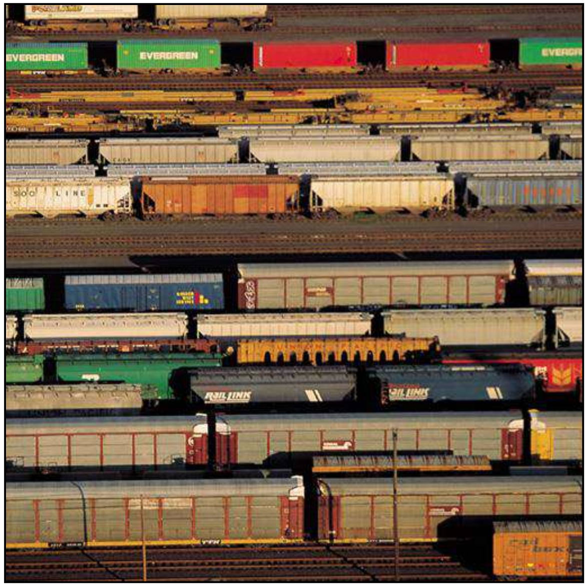


Ideal HPF*

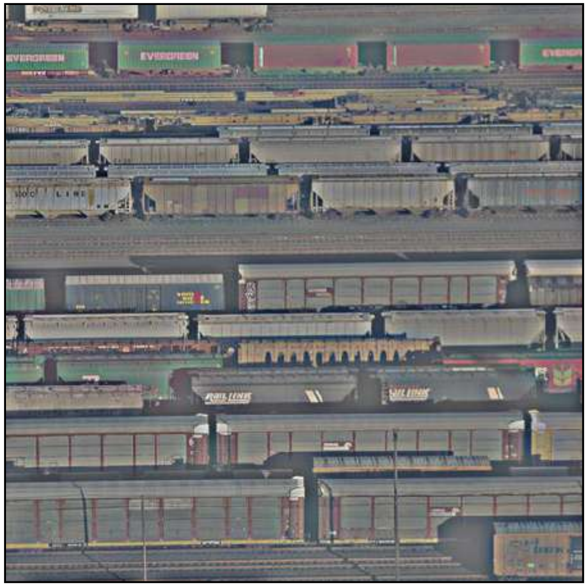
*signed image; 0 mapped to 128

Comparison of Ideal and Gaussian Filters

Gaussian LPF



Original Image

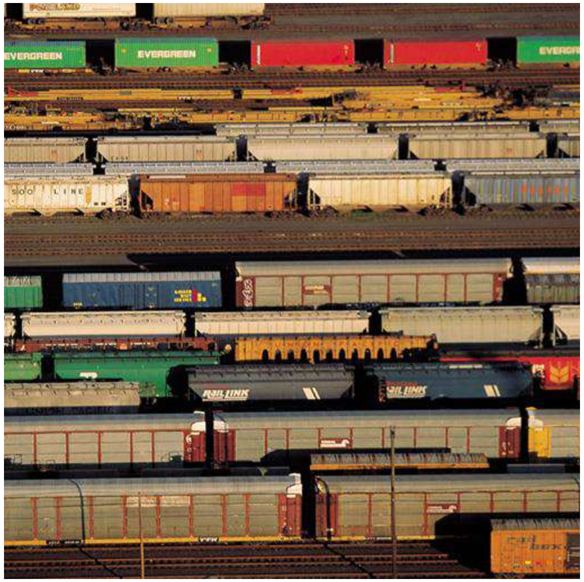


Gaussian HPF*

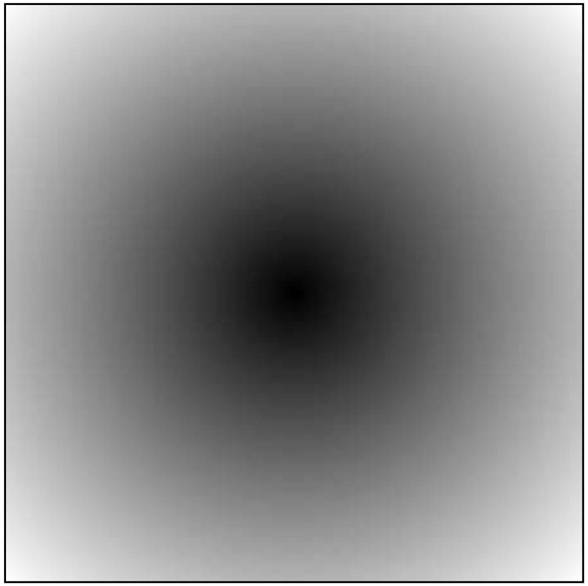
DLR

*signed image; 0 mapped to 128

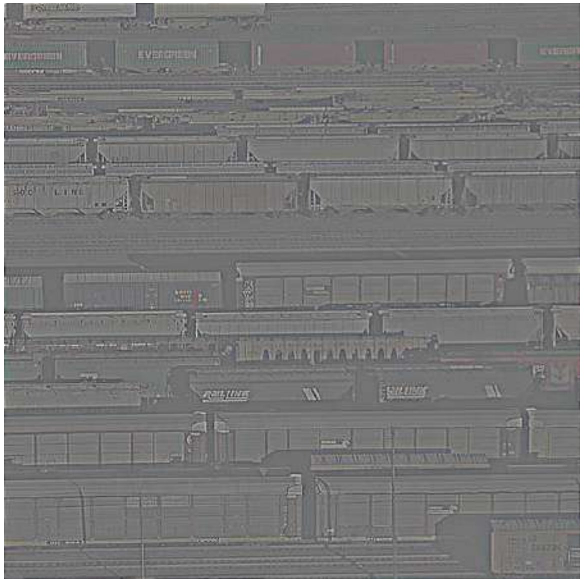
Another Highpass Filter



original image

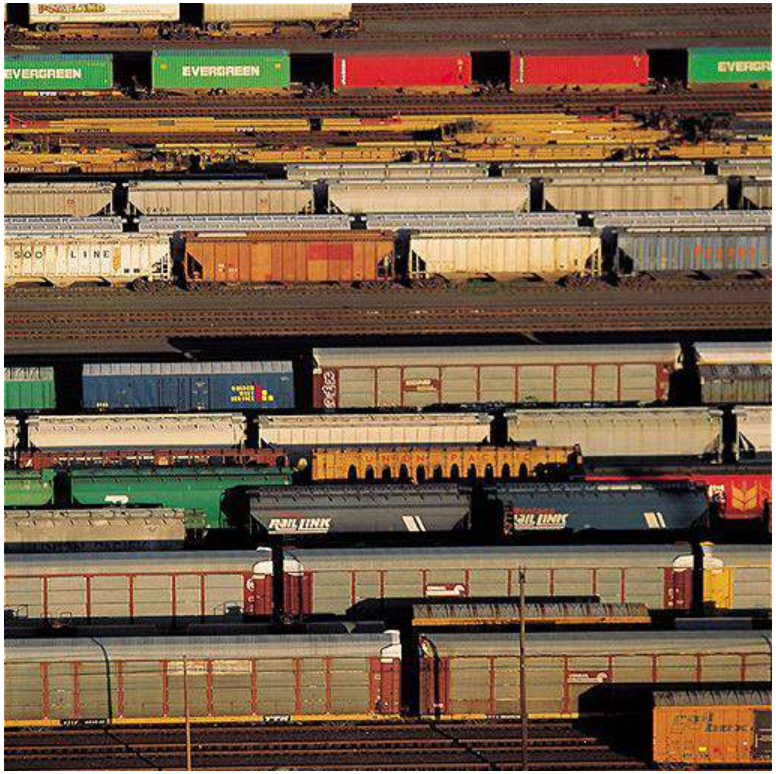


filter power spectrum

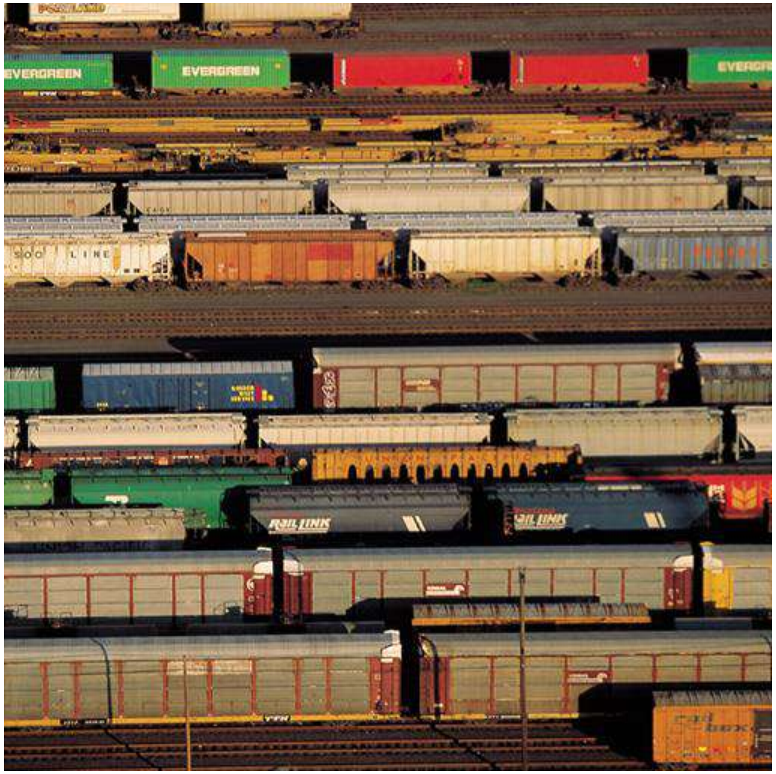


filtered image*

Original Image + Horiz. + Vert. Edges

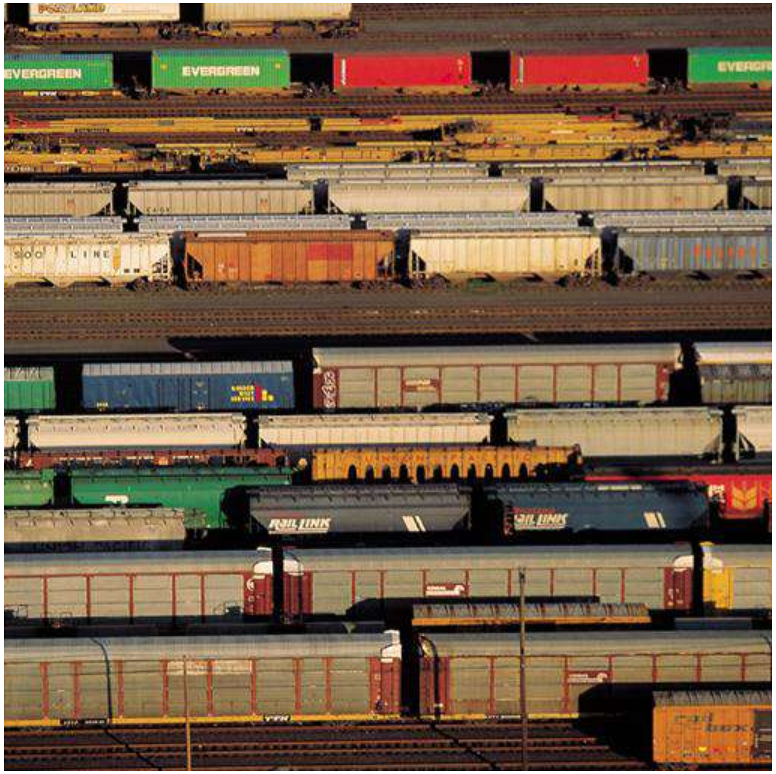


sharpened

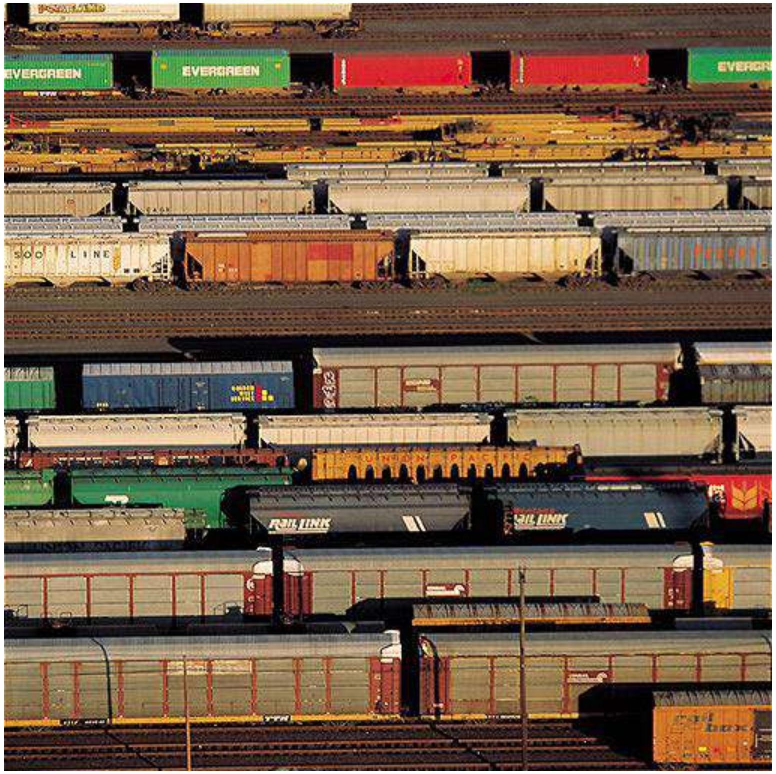


original

Original Image + Horiz. + Vert. Edges



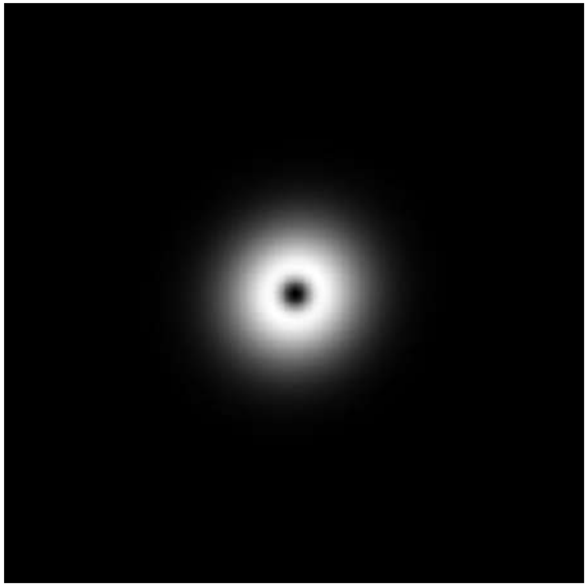
original

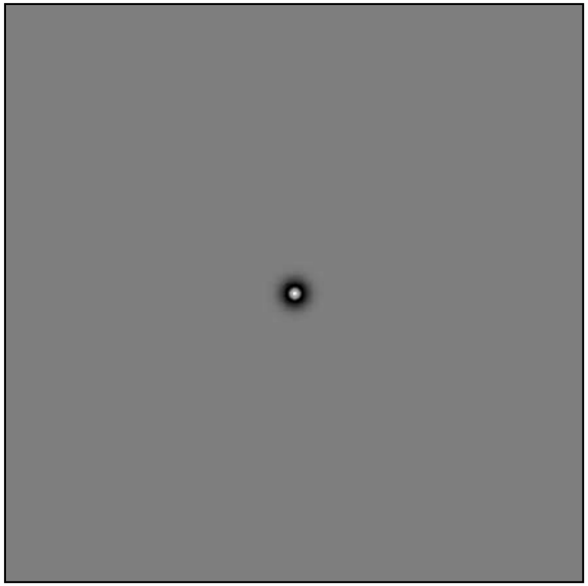


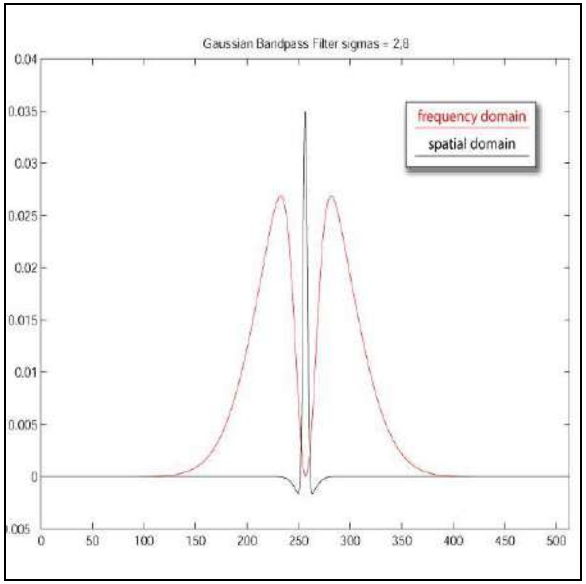
sharpened

Gaussian Bandpass Filter

Image size: 512x512
sigma = 2 - sigma = 8



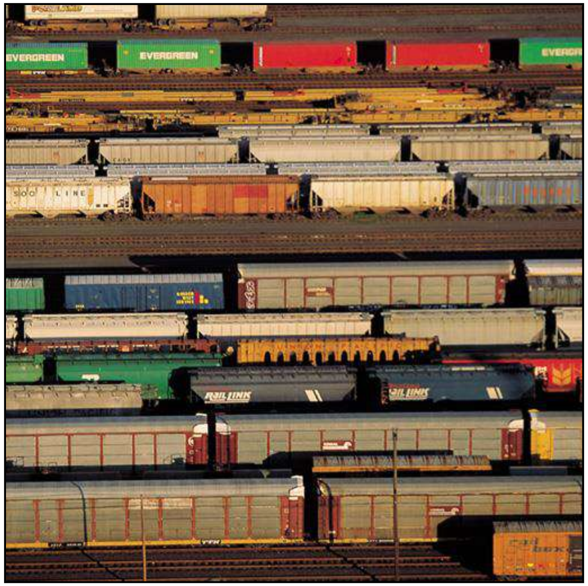




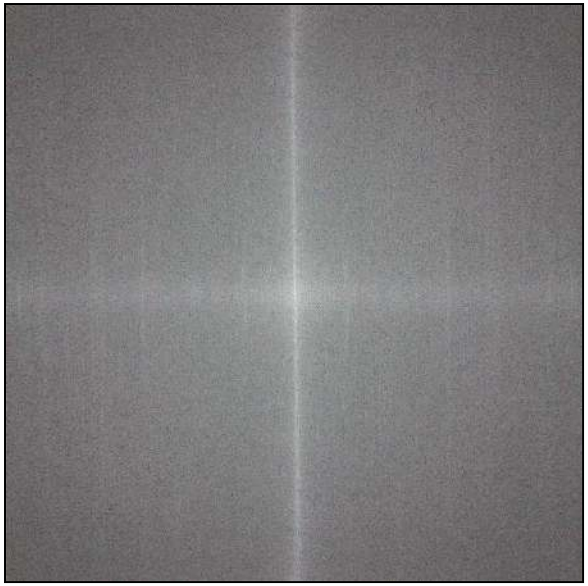
Fourier Domain Rep. | Spatial Representation | Central Profile

Gaussian Bandpass Filter

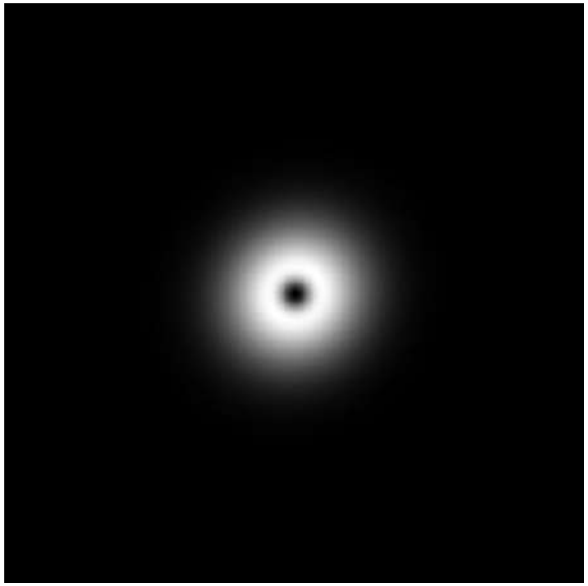
Image size: 512x512
sigma = 2 - sigma = 8



Original Image



Power Spectrum

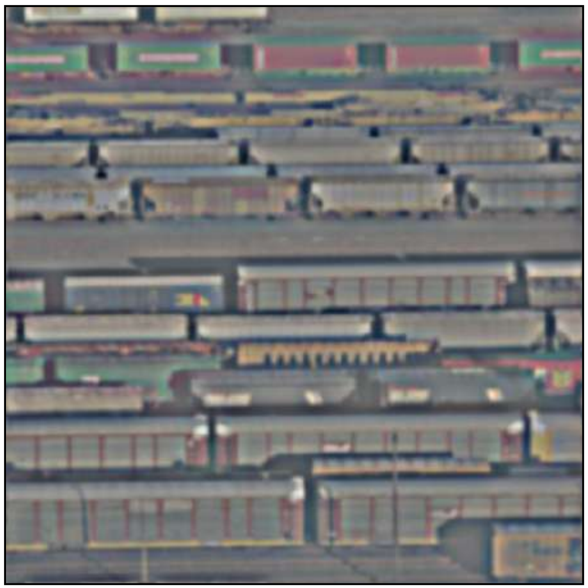


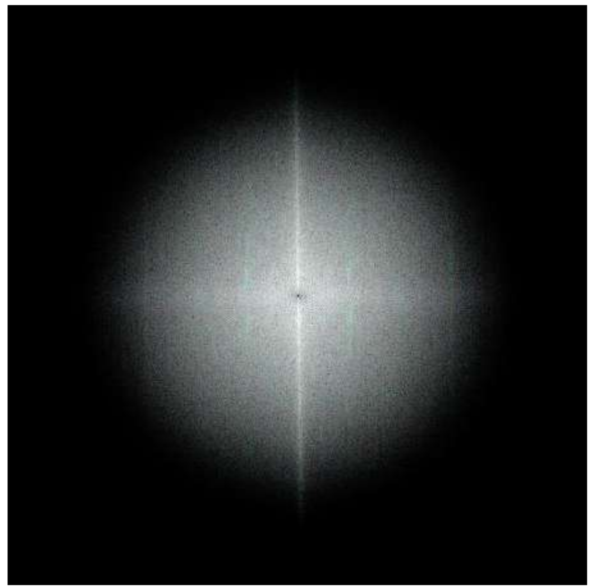
Gaussian BPF in FD

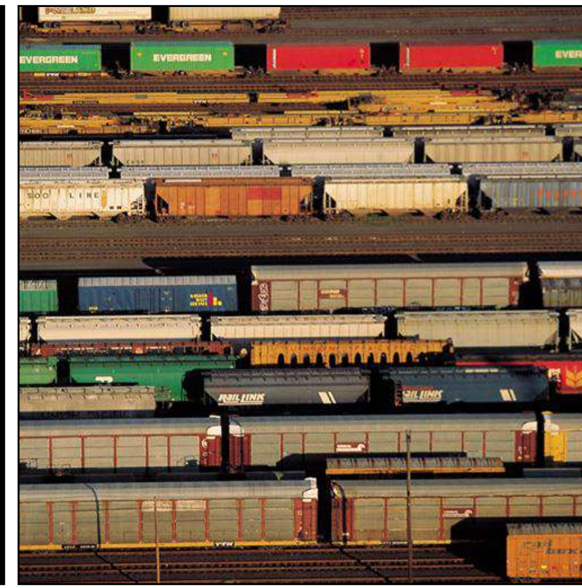
*signed image; 0 mapped to 128

Gaussian Bandpass Filter

Image size: 512x512
sigma = 2 - sigma = 8







Filtered Image*

Filtered Power Spectrum

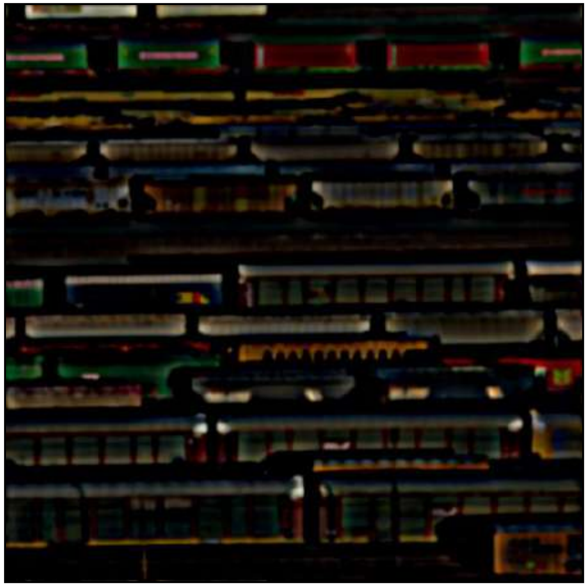
Original Image

DLR

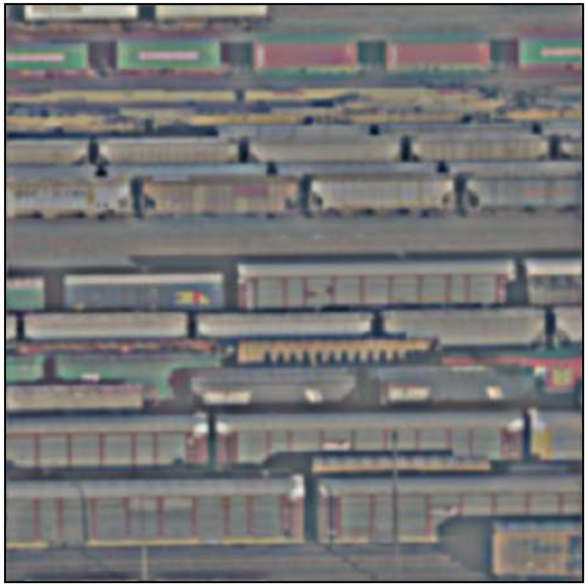
*signed image; 0 mapped to 128

Gaussian Bandpass Filter

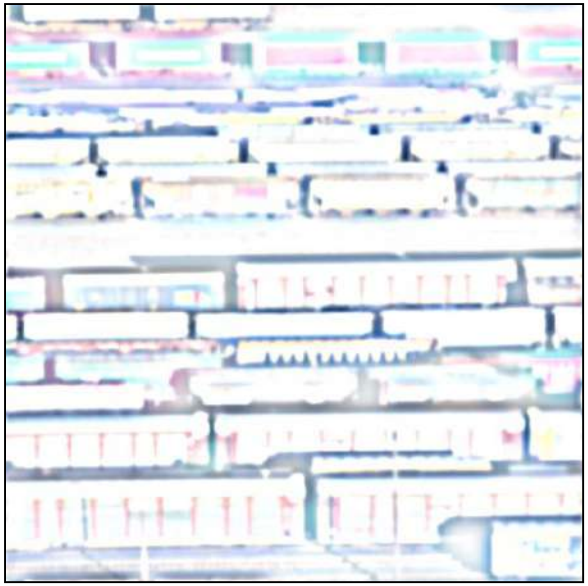
Image size: 512x512
sigma = 2 - sigma = 8



Positive Pixels



Filtered Image*



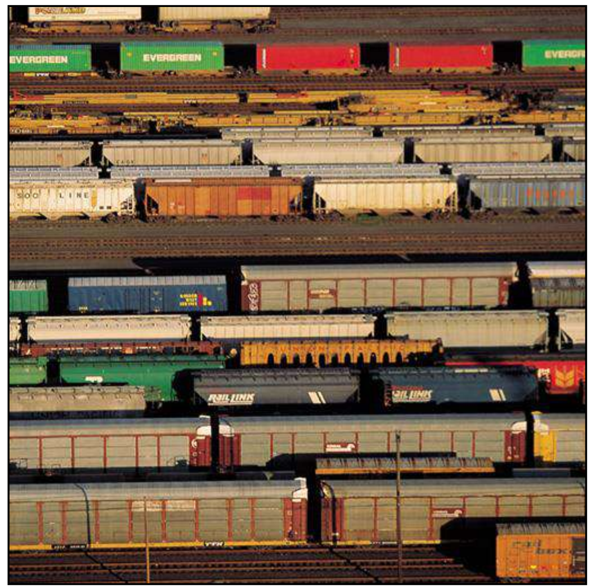
Negative Pixels

DLR

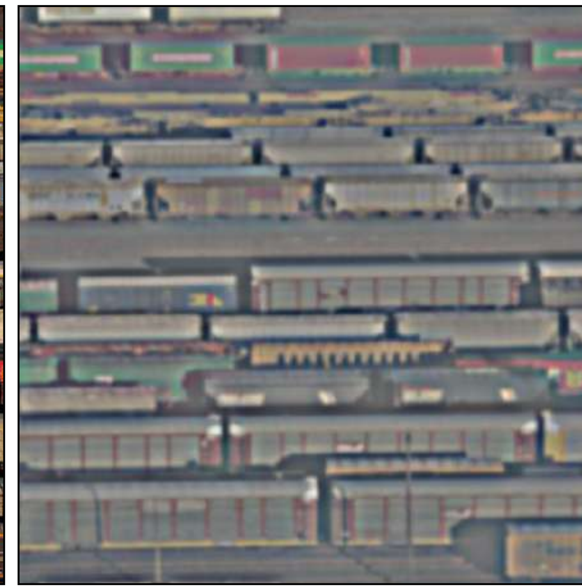
*signed image; 0 mapped to 128

Comparison of Ideal and Gaussian Filters

Ideal BPF*



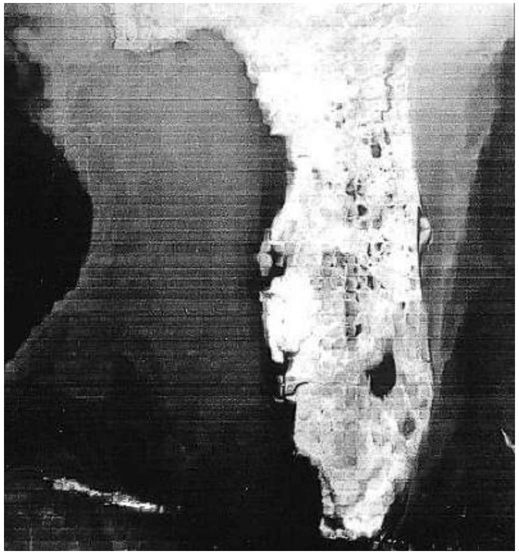
Original Image



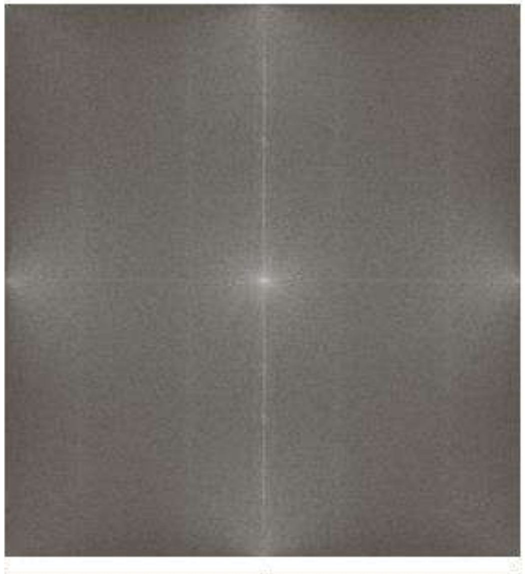
Gaussian BPF*

DLR

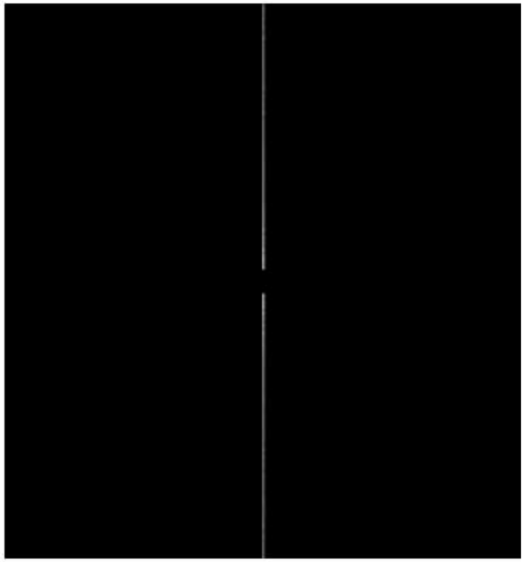
Removal of Horizontal Stripes



Original image *I*

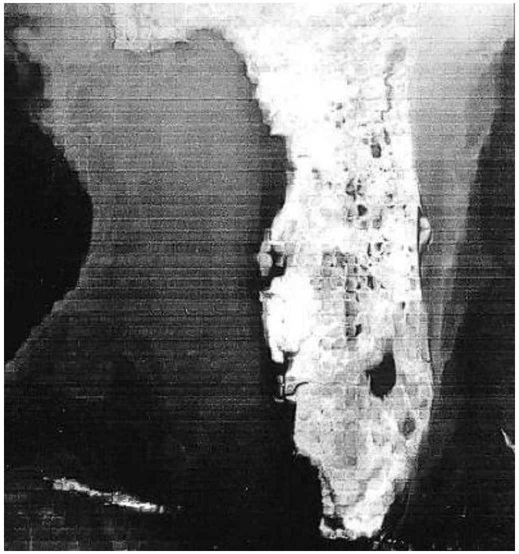


Log power spectrum

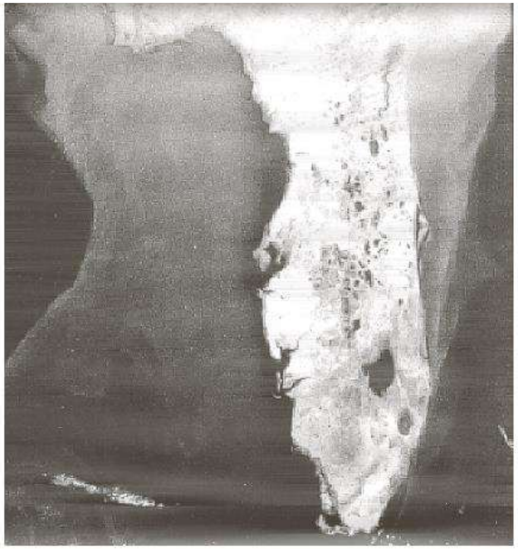


Filter *H*

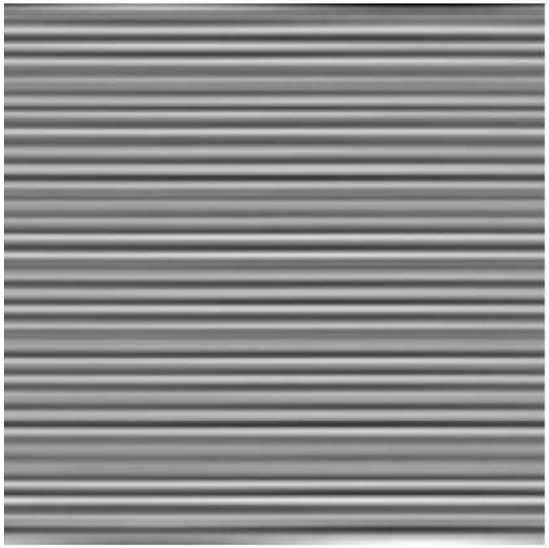
Removal of Horizontal Stripes



Original image *I*

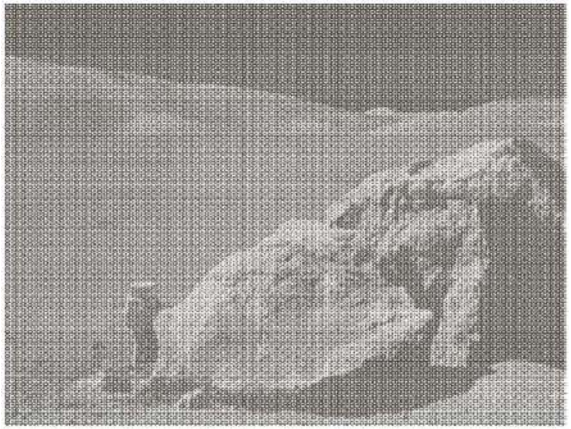


Filtered Image
$I*H_{reject}$

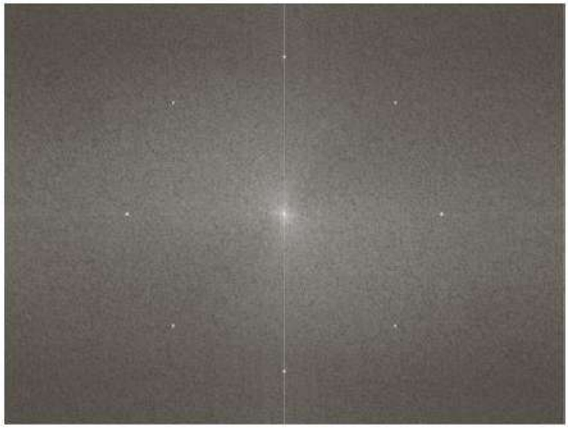


Inverse FT of *H*

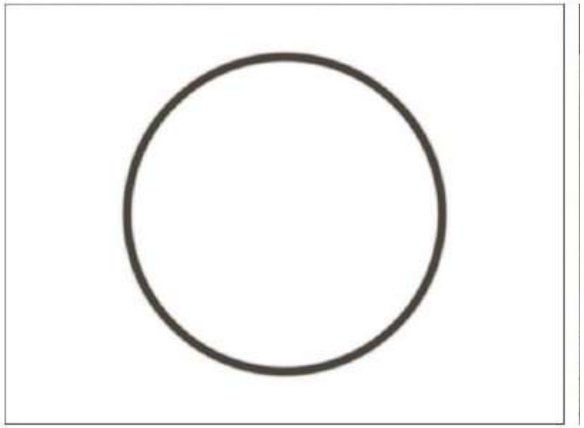
Bandreject Filter



Original image



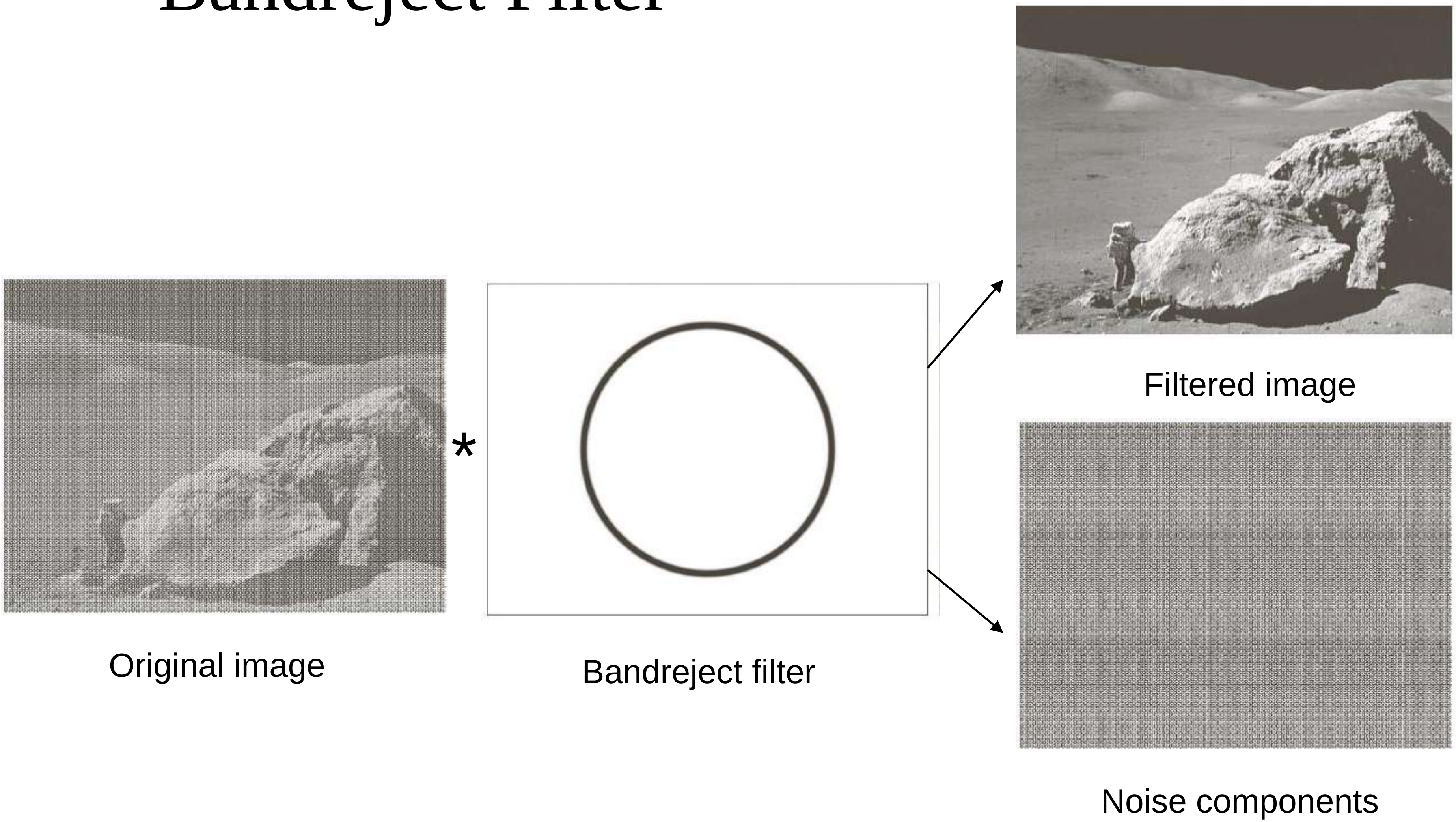
Log power spectrum



Bandreject filter

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Bandreject Filter



Summary: Global & Local Techniques
Histogram Stretching + Sharpening

Original RGB image

Histogram stretch

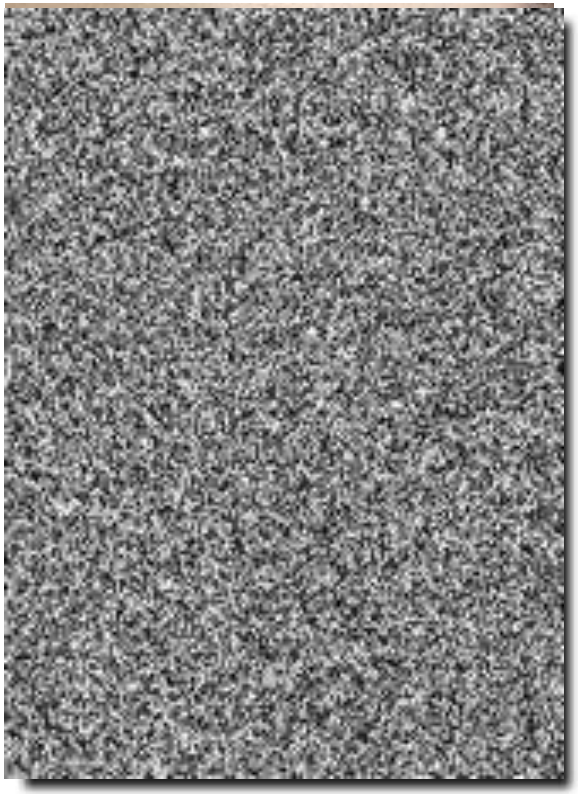
Image Sharpening

Warning → sharpening increases noise: do not sharpen a noisy image!

Effects of Noise on Enhancement of High Frequencies

original image

+



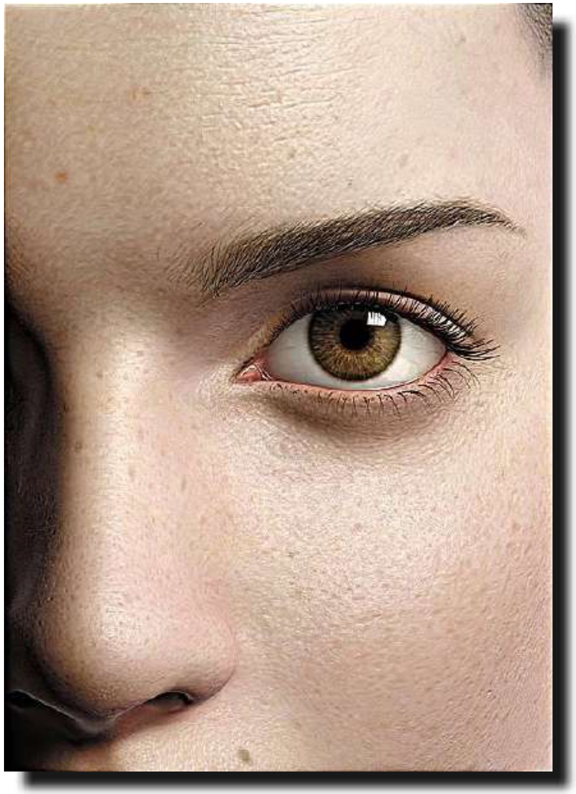
Random noise
(several times magnified!)

Effects of Noise on Enhancement of High Frequencies

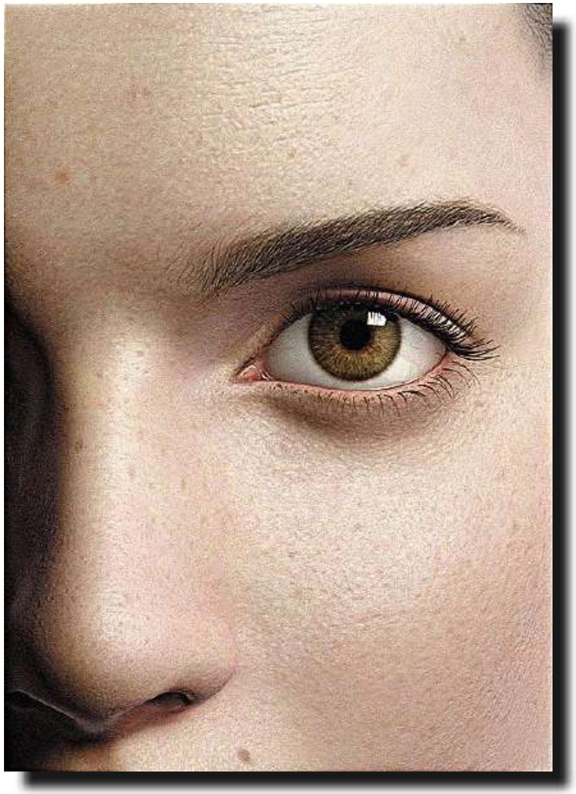
original image

noisy image

Effects of Noise on Enhancement of High Frequencies



HF enhanced original



HF enhanced noisy image

Effects of Noise on Enhancement of High Frequencies

original image

noisy image

Effects of Noise on Enhancement of High Frequencies

HF enhanced original

HF enhanced noisy image

Effects of Noise on Enhancement of High Frequencies

original image

HF enhanced original

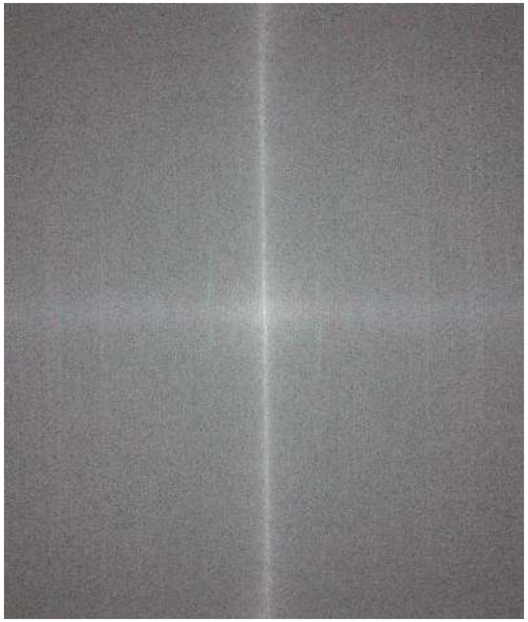
Effects of Noise on Enhancement of High Frequencies

noisy image

HF enhanced noisy image

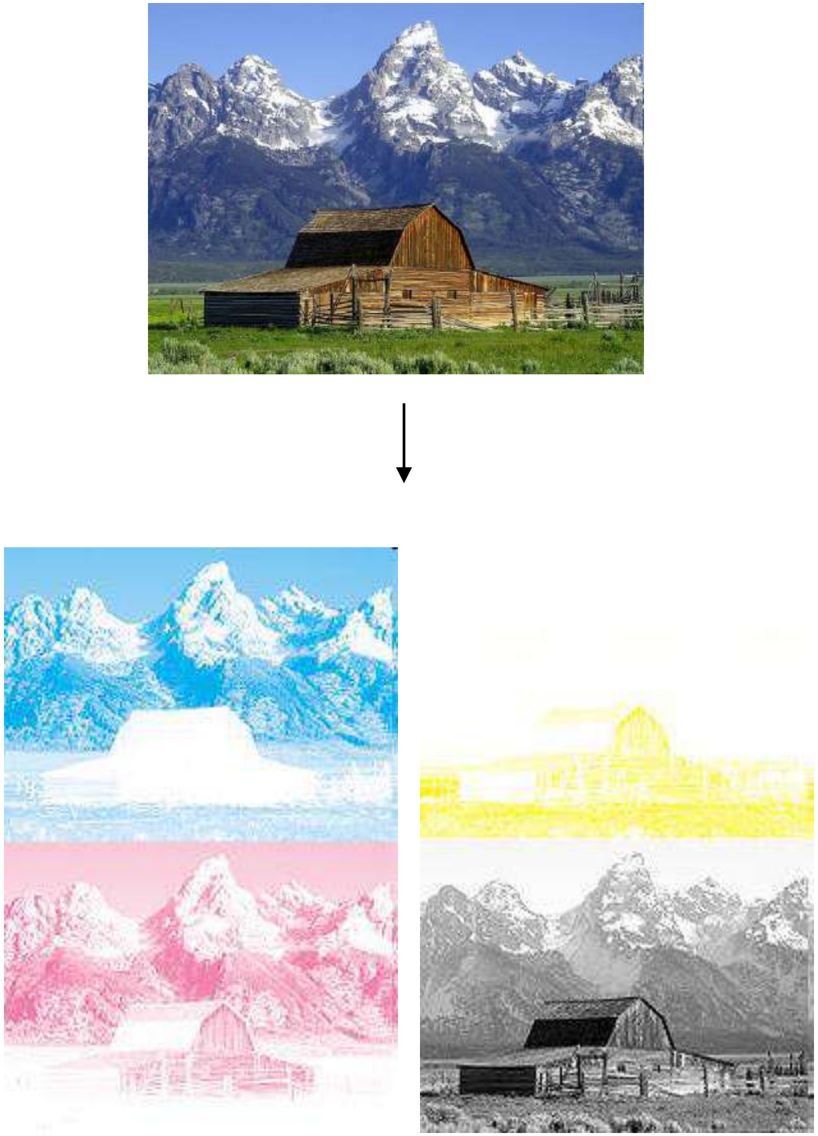
Some Interesting Applications

(Optional, we will probably stop here ☺)



CMYK color model

- Cyan-Magenta-Yellow is the standard color model for paper printing
- How are the colors superimposed?



$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

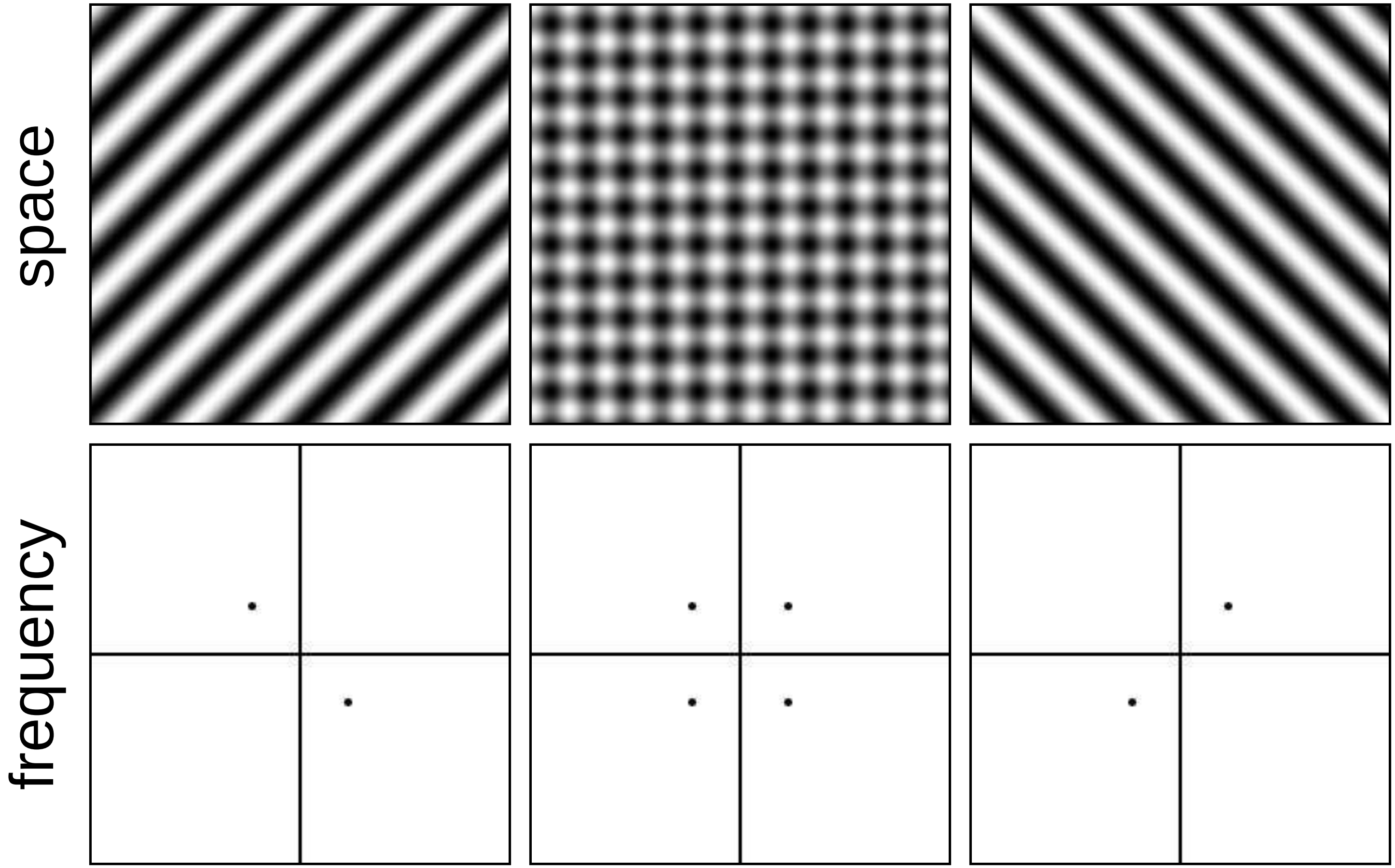
DLR

Processing Scanned Press-Printed Images

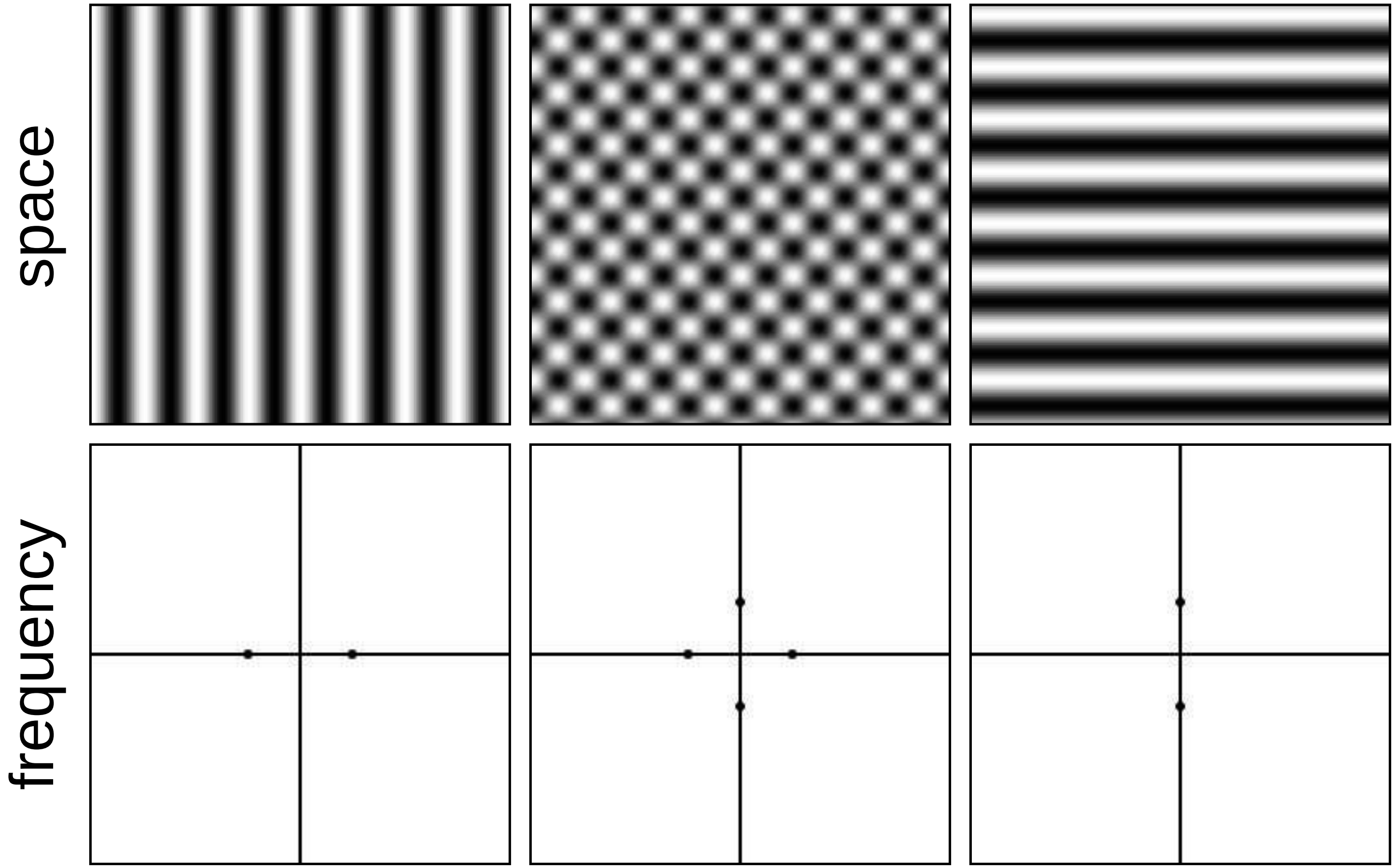
4-color printing:

1. A photograph or other color image is separated into four intensity band images: cyan, magenta, yellow, and black.
2. Each of these is multiplied by a halftone screen – a dot mask with a unique orientation.
3. Each of the resulting four images shows a pattern of dots whose individual sizes indicate the amount of ink to be applied at each point.
4. The four images are printed, one atop the other, in the corresponding color.

Halftone Screen

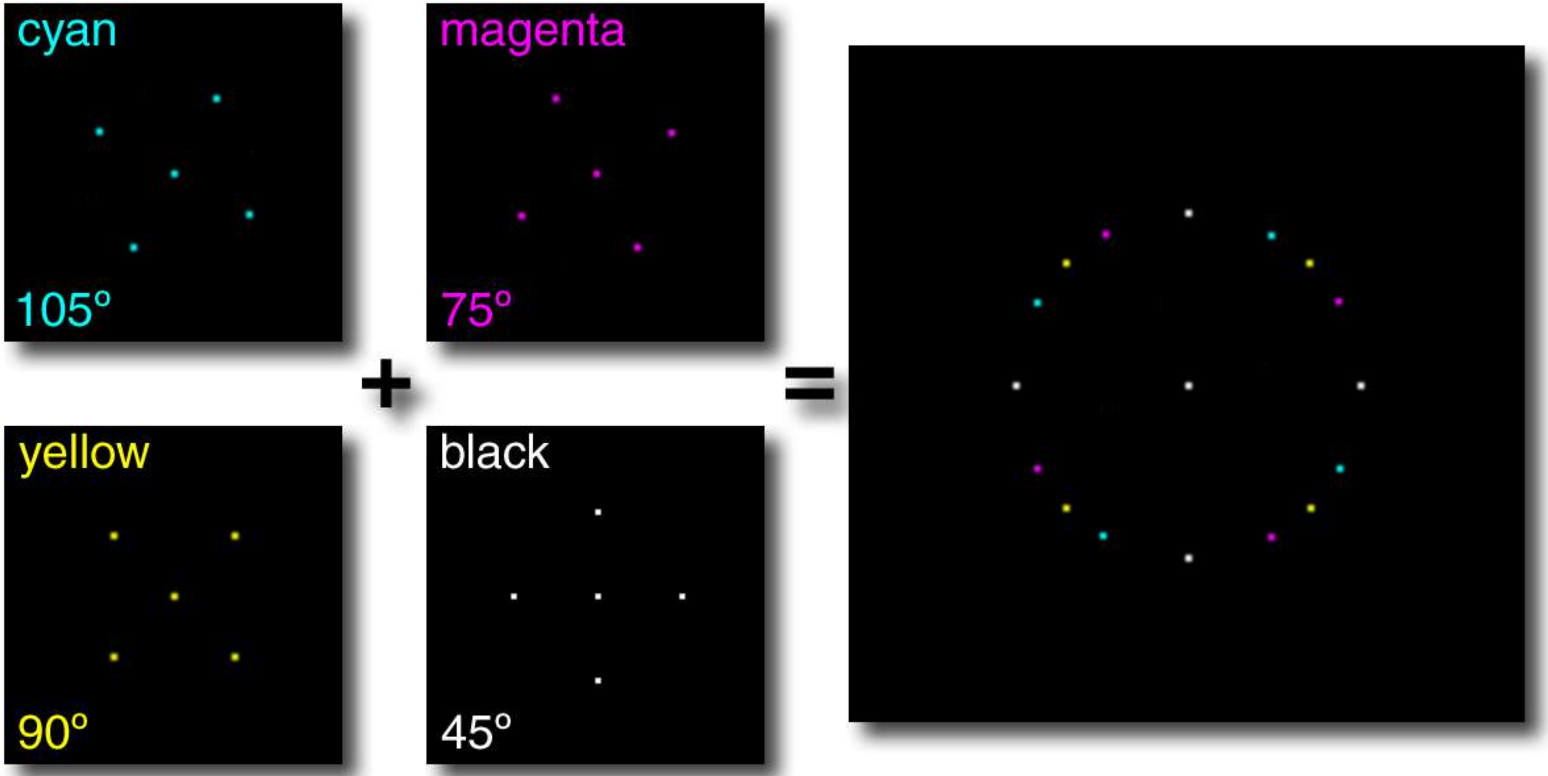


Halftone Screens



Each band has 2 perpendicular sinusoids + an impulse in the origin…

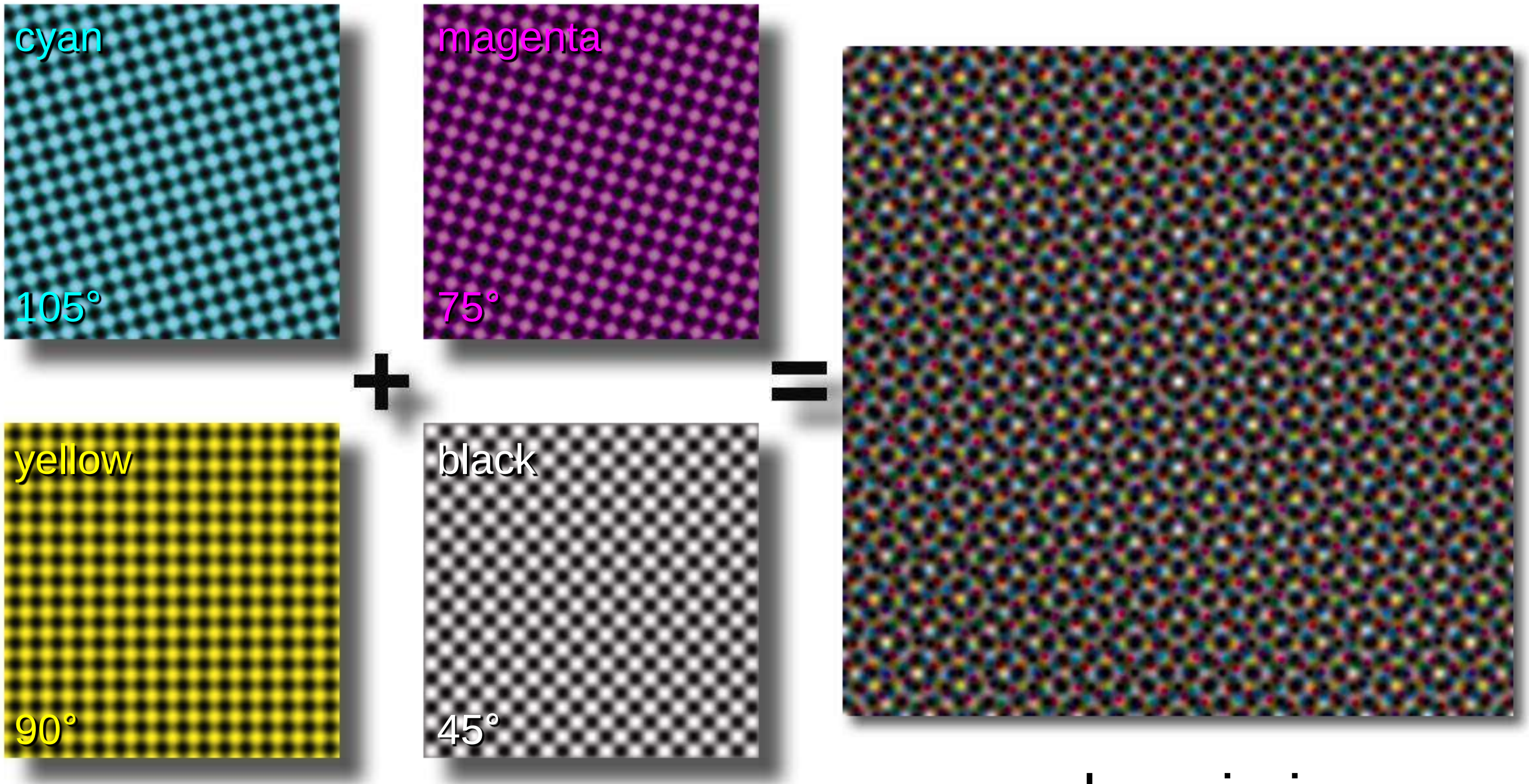
CMYK Standard Halftone Screens



Power Spectra

… which creates rectangular grids at 4 different angles.

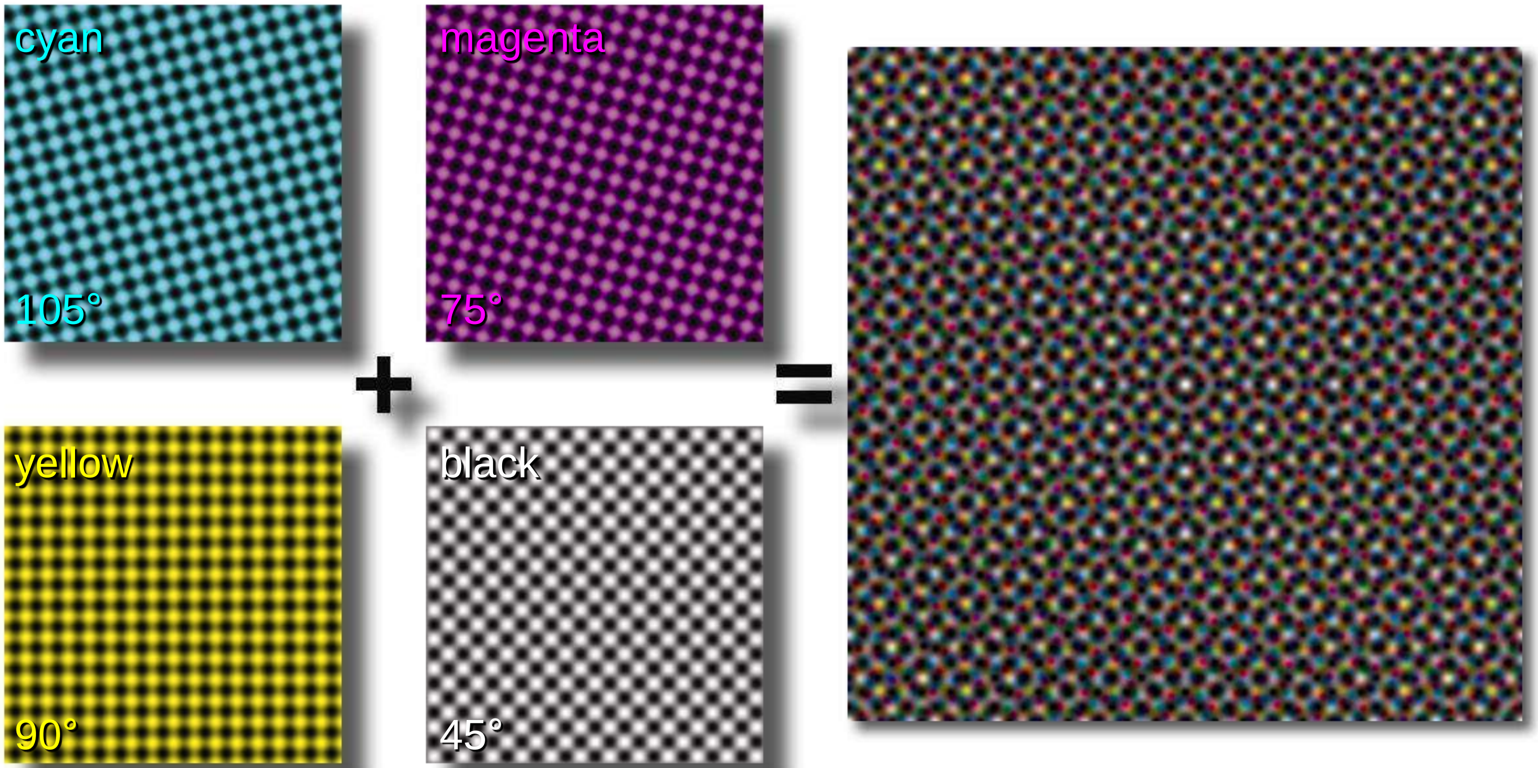
CMYK Standard Halftone Screens



space domain images

When the 4 are summed, the result is a “rosette” image.

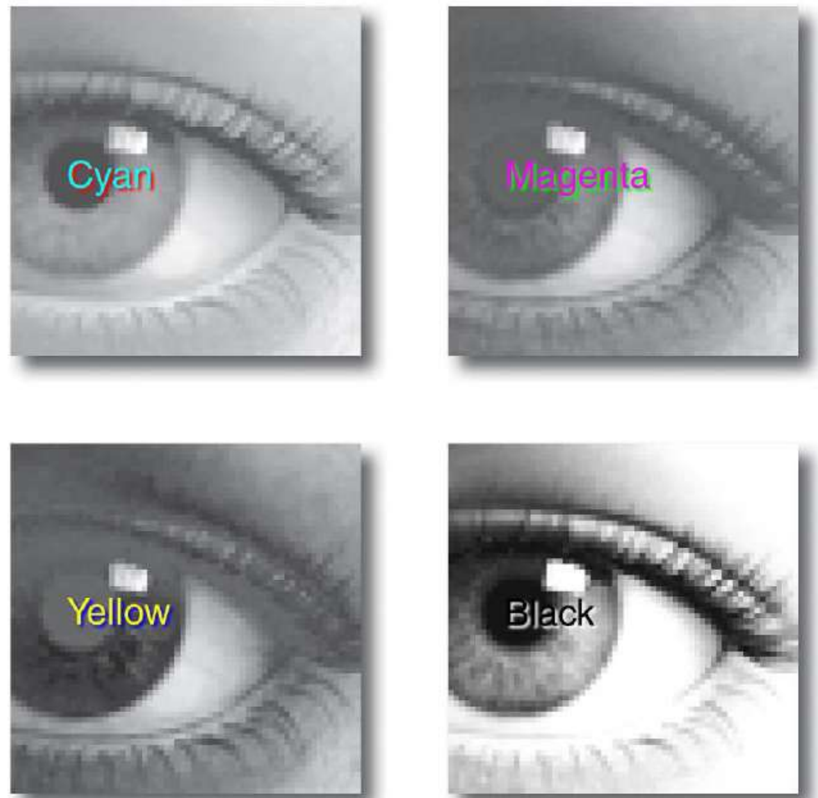
CMYK Standard Halftone Screens



space domain images

To print an image, it is separated into 4 color bands …

Example: Color Separation / Halftoning

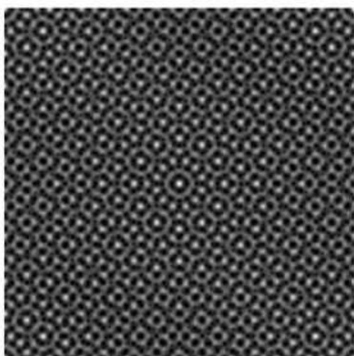


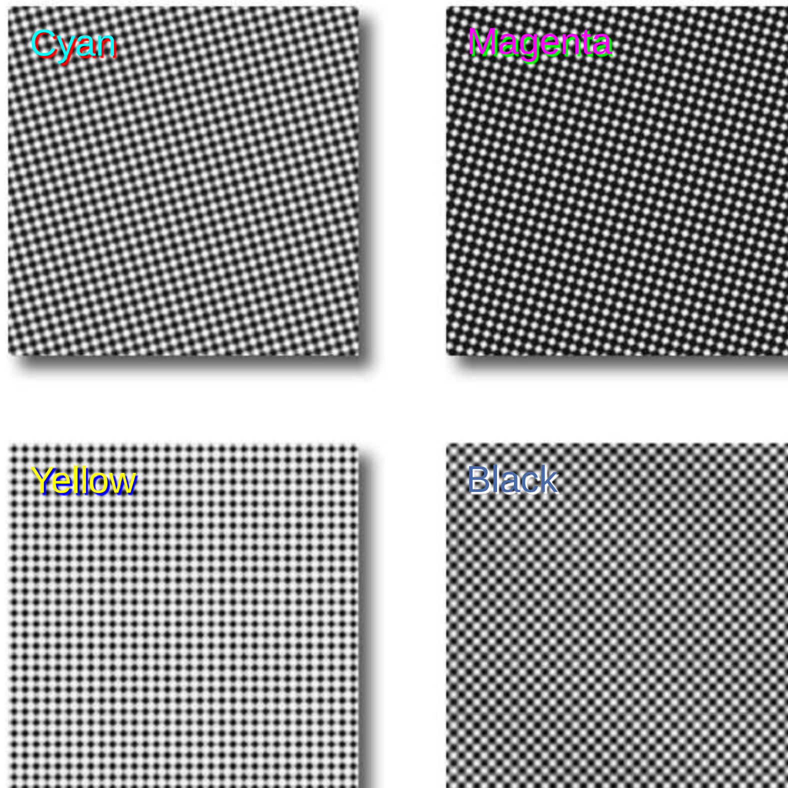
That is, an intensity image is created for each of the four color bands.

… each of which is multiplied by a corresponding screen.

Color Separation / Halftoning

Each intensity image is multiplied by a corresponding screen, then



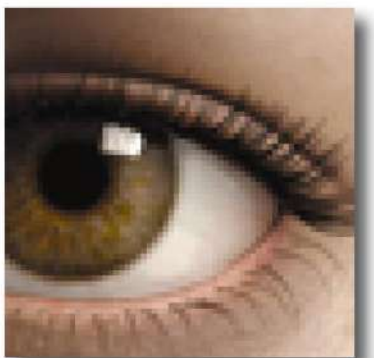


each screened image is printed in its own color on the same page.

DLR

To print an image, it is separated into 4 color bands …

Example: Color Separation / Halftoning

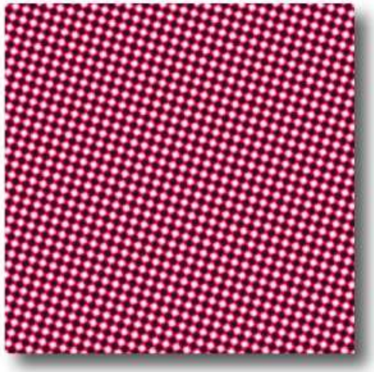


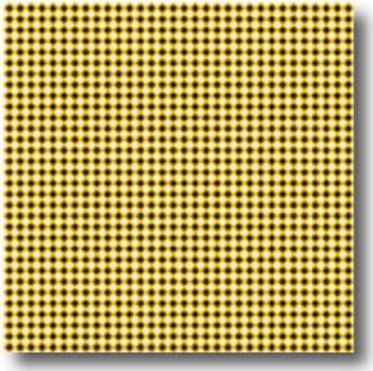
Here the bands tinted in their corresponding colors.

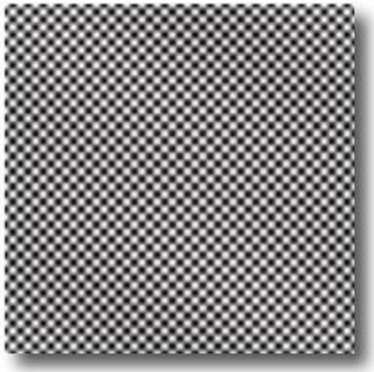
DLR

… each of which is multiplied by a corresponding screen …

Example: Color Separation / Halftoning







Here the screens tinted in their corresponding colors.

…to get dot patterns for printing.
The 4 are printed over each other
to get the final result.

Example: Color Separation / Halftoning

Halftone Dots

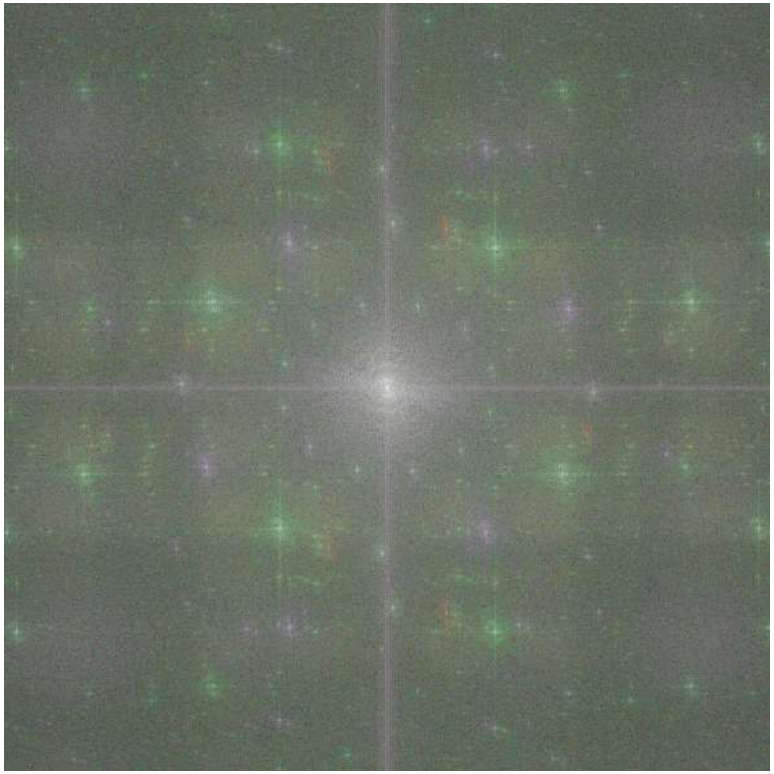
Image scanned (600 dpi) from a magazine

Detail: Circular patterns, the rosettes, are the result of the halftone dots.

DLR

Filtering Out Halftone Dot Distortion

original

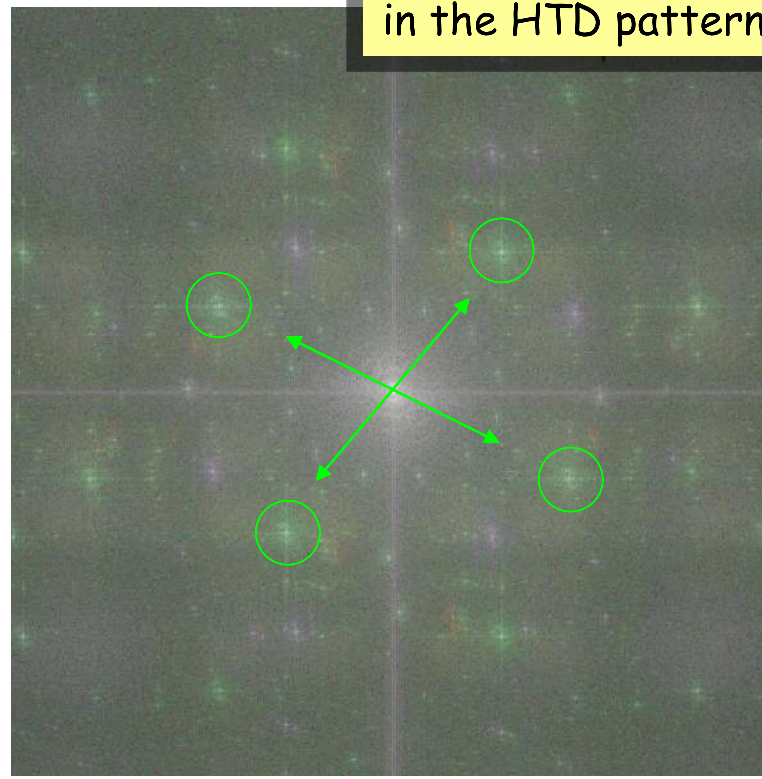


log power spectrum

Filtering Out Halftone Dot Distortion

Each pair of peaks corresponds to a sinusoidal sub-pattern in the HTD pattern.

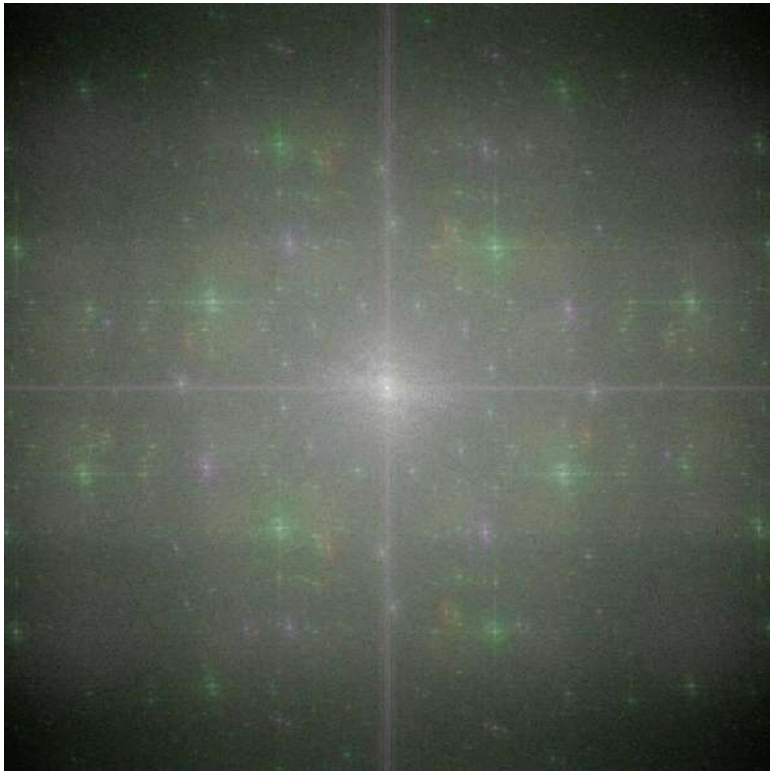
original



log power spectrum

Blurring with a Gaussian (σ = 1)

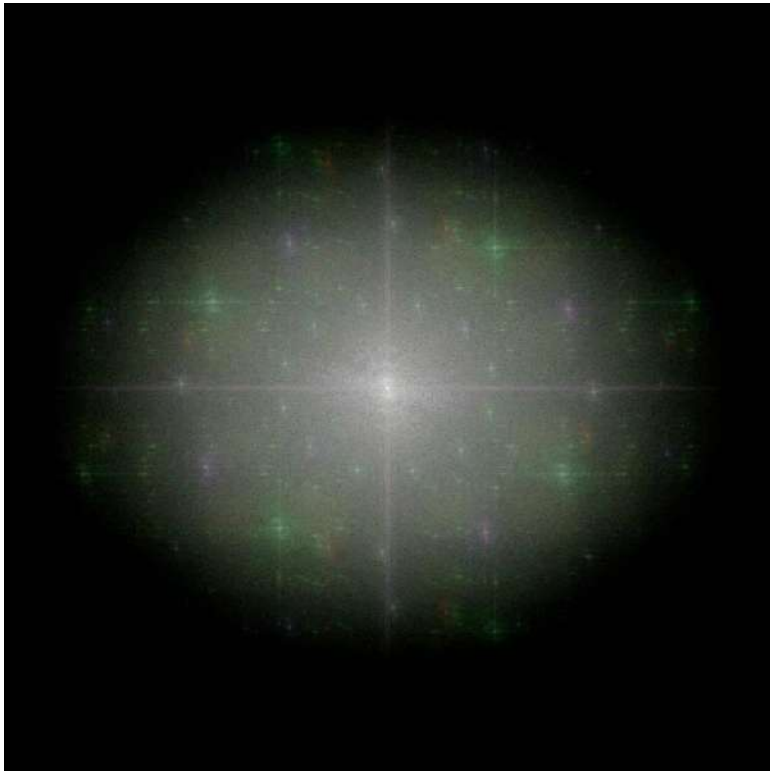
blurred image σ=1



log power spectrum σ=1

Blurring with a Gaussian (σ = 2)

blurred image σ=2

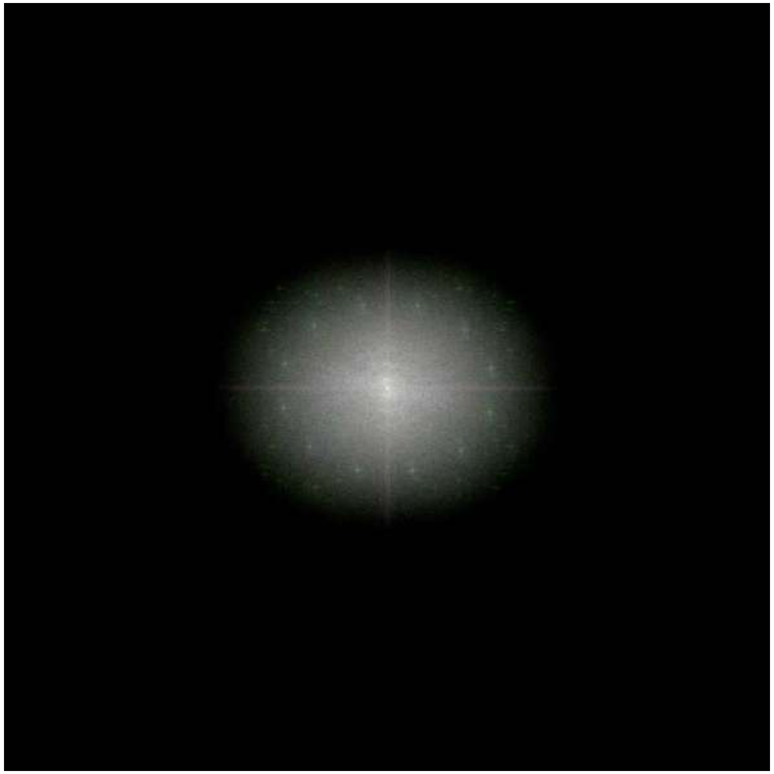


log power spectrum σ=2

DLR

Blurring with a Gaussian (σ = 4)

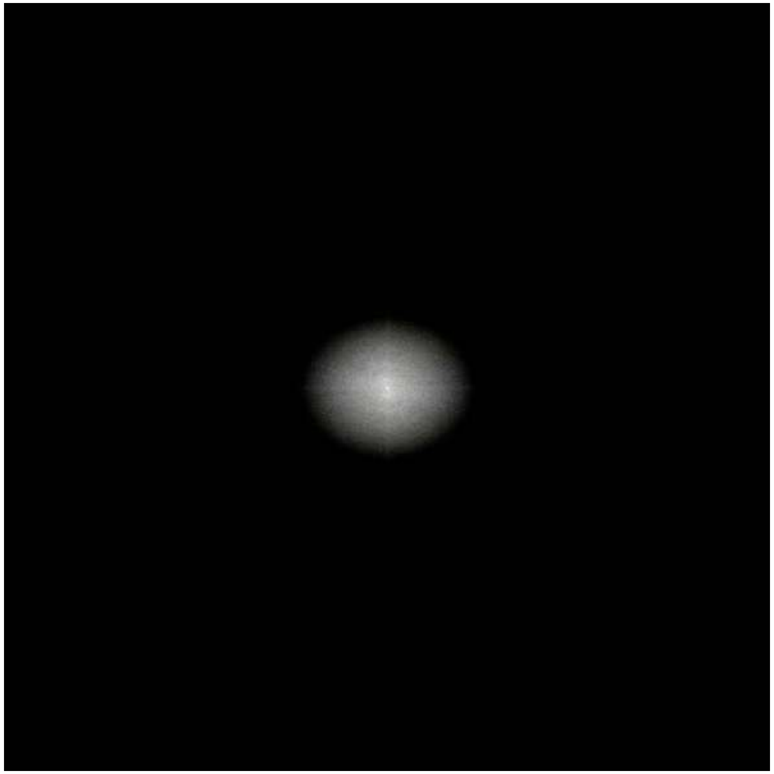
blurred image σ=4



log power spectrum σ=4

Blurring with a Gaussian (σ = 8)

blurred image σ=8



log power spectrum σ=8

Blurring with a Gaussian (σ = 1)

original

difference

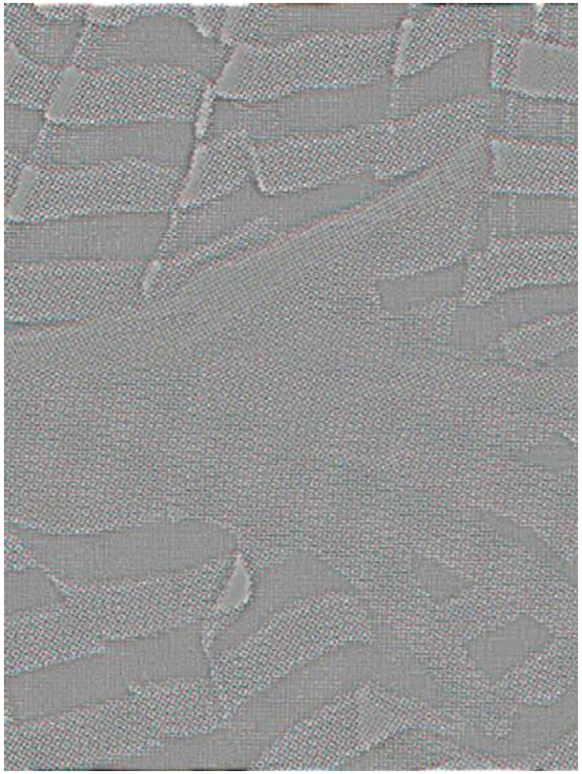
blurred σ = 1

middle gray = 0, normalized

DLR

Blurring with a Gaussian (σ = 2)

blurred σ = 2

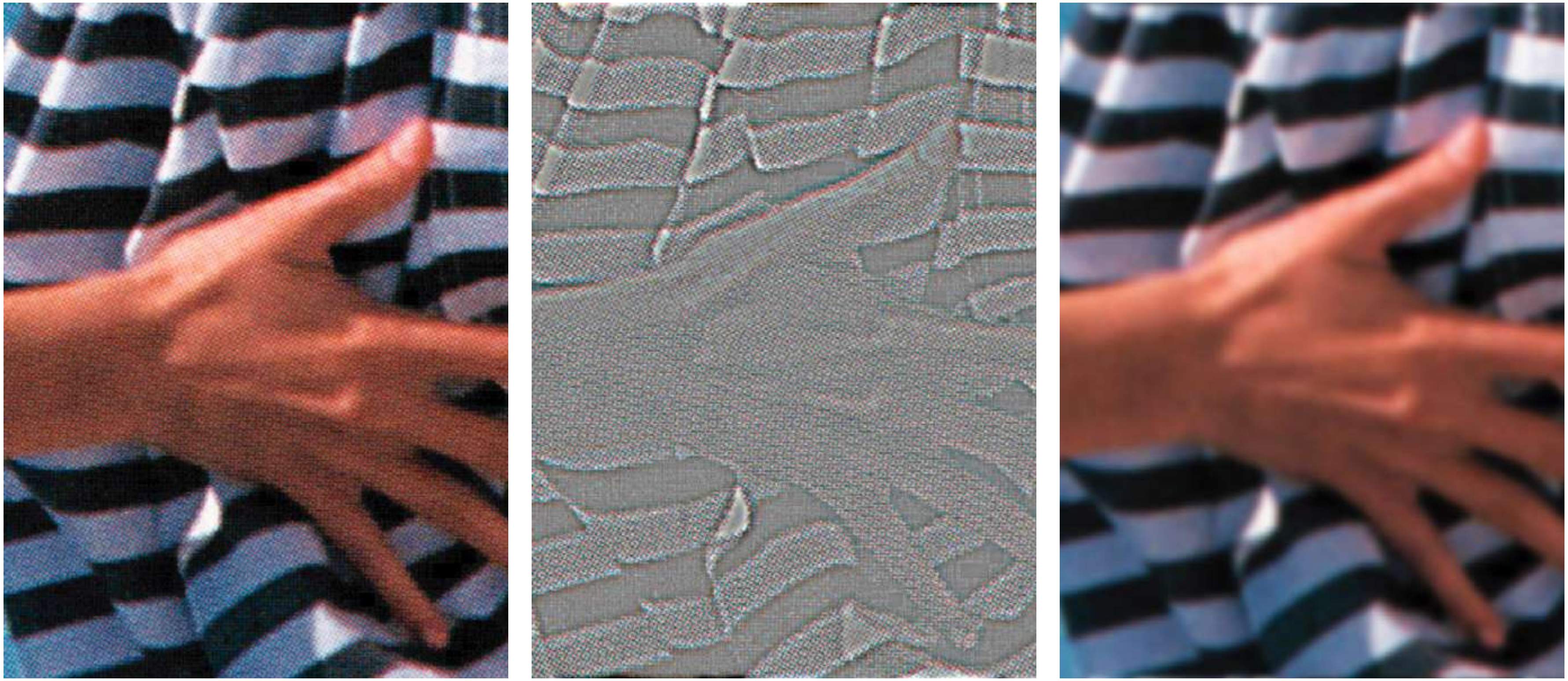


difference

middle gray = 0, normalized

original

Blurring with a Gaussian (σ = 4)

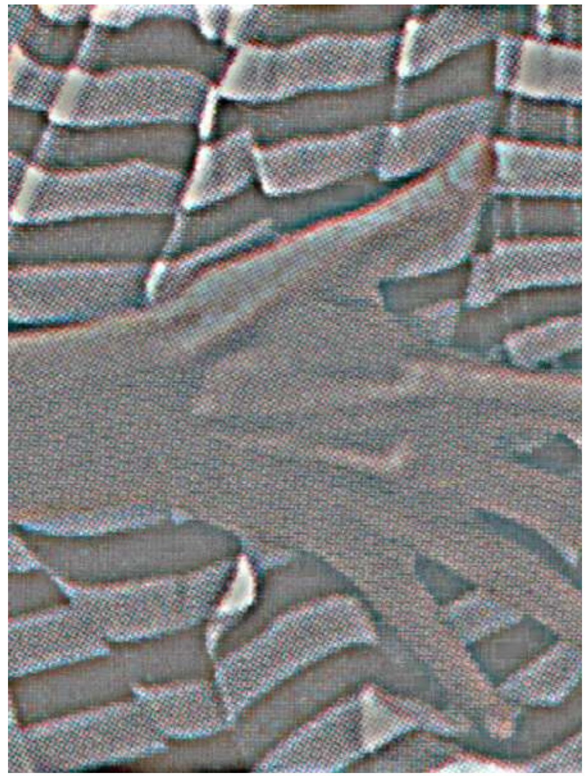


original | difference | blurred σ = 4

middle gray = 0, normalized

Blurring with a Gaussian (σ = 8)

blurred $\sigma = 8$



difference

original

middle gray = 0, normalized

DLR

Problem with Blurring to Reduce HTD Distortion

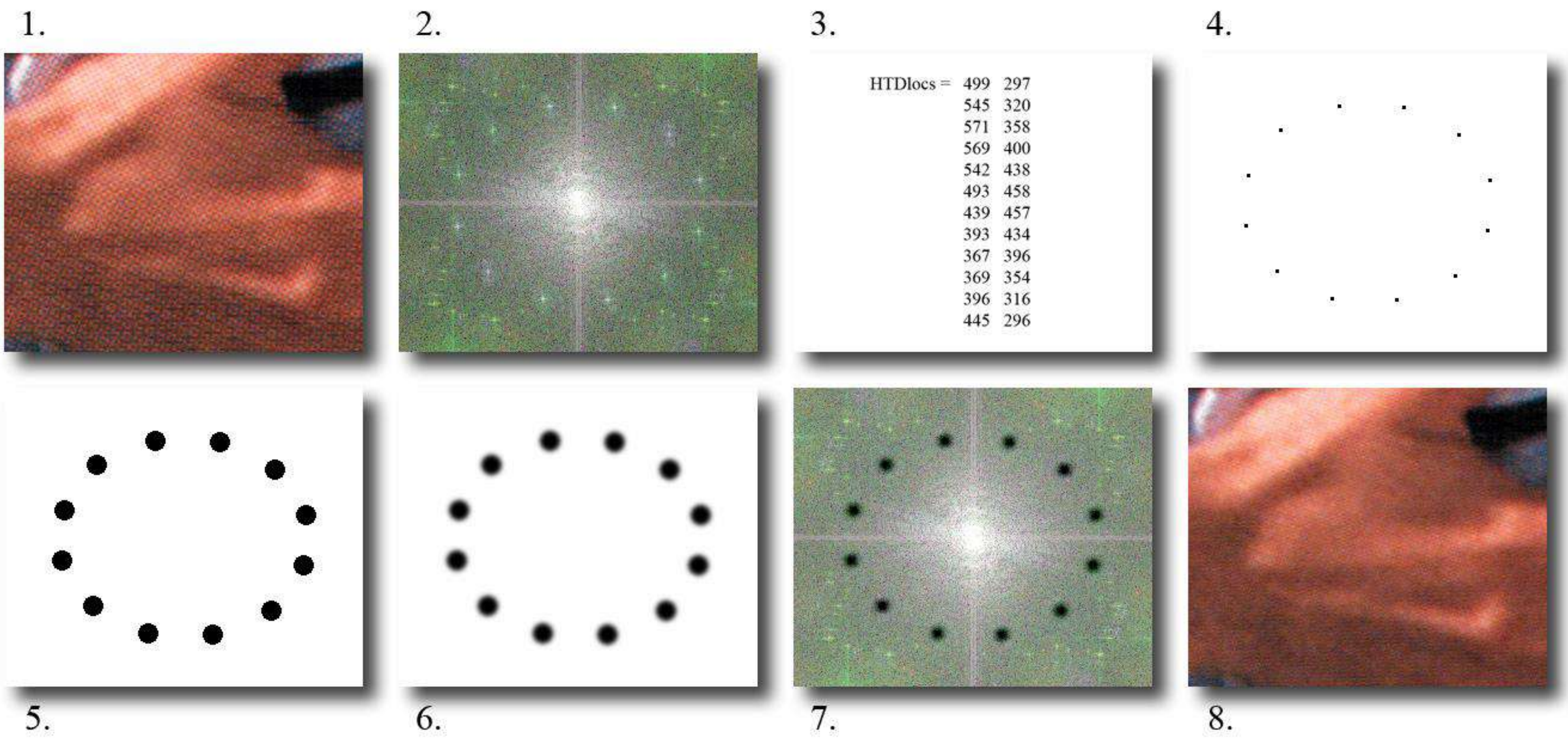
It blurs everything.

Better to remove the HTD frequency components selectively:

1. Read in the image.
2. Compute the log power spectrum of the image.
3. Find the locations of the HTD spectrum peaks.
4. Mark these on a mask.
5. Enlarge the points to regions that cover most of the energy.
6. Blur the mask for used as a notch filter.
7. Multiply the Fourier transform of the image by the mask.
8. Take the inverse Fourier transform of the result.

Remove HTD Distortion Selectively

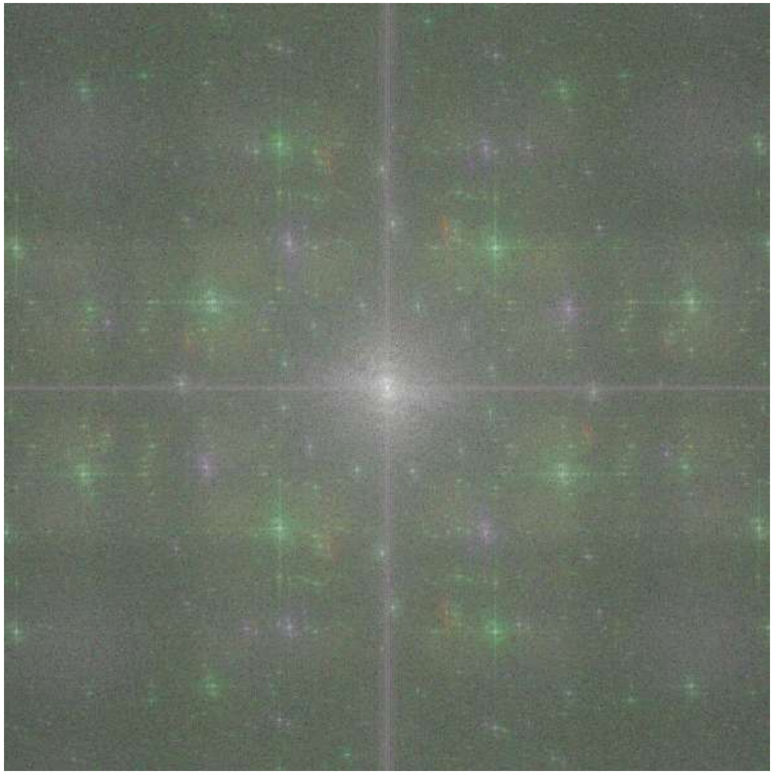
… through notch filtering.



1. Read in image; 2. Compute power spectrum; 3. Locate HTD frequency components; 4. Mark locs on a mask; 5. Enlarge points to regions; 6. Blur the mask; 7. Multiply FT of image by mask; 8. Take inverse FT of result;

Notch Filtering of Halftone Dot Distortion

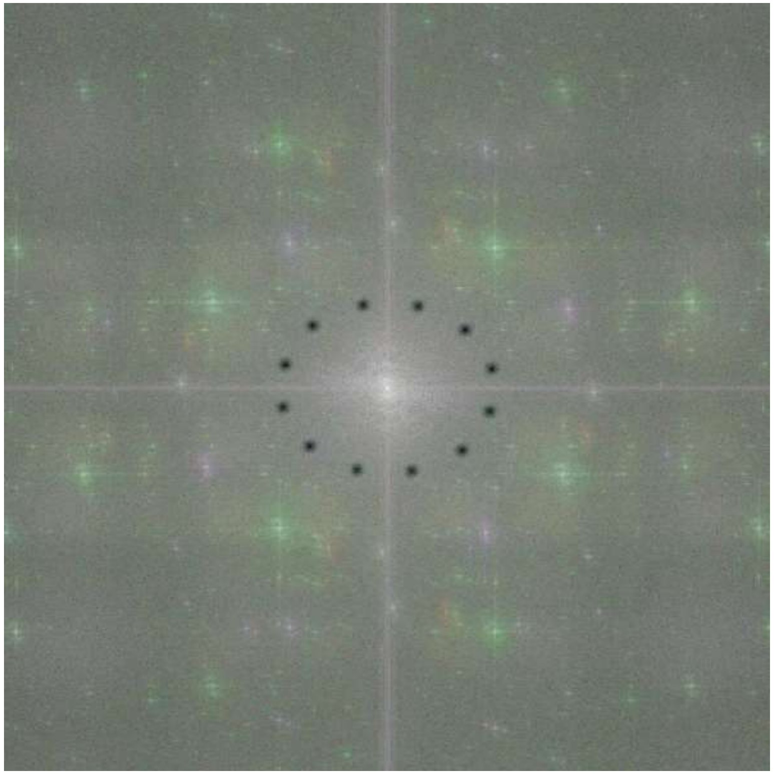
original



log power spectrum

Notch Filtering

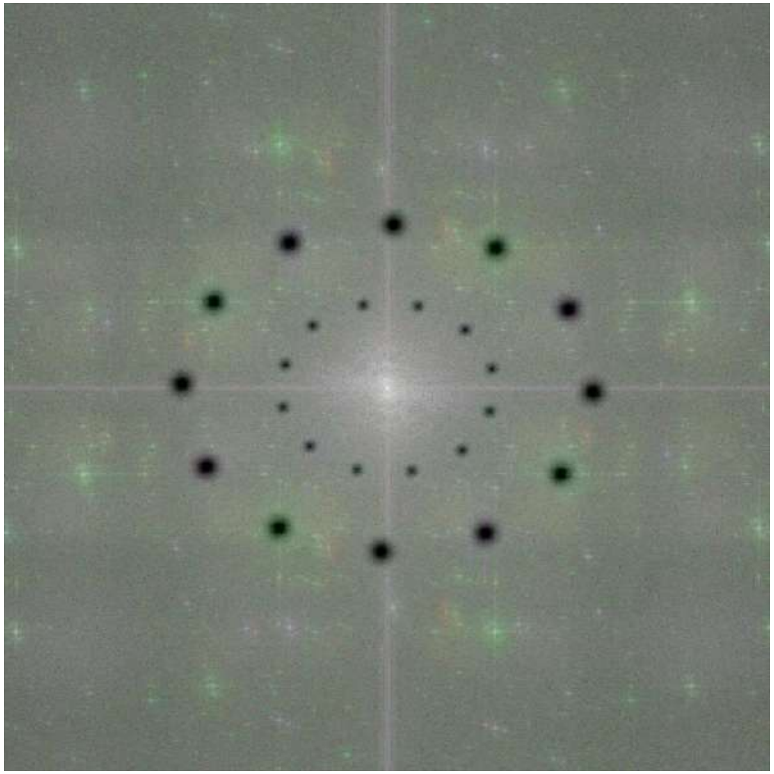
frequency masked 1



log power spectrum

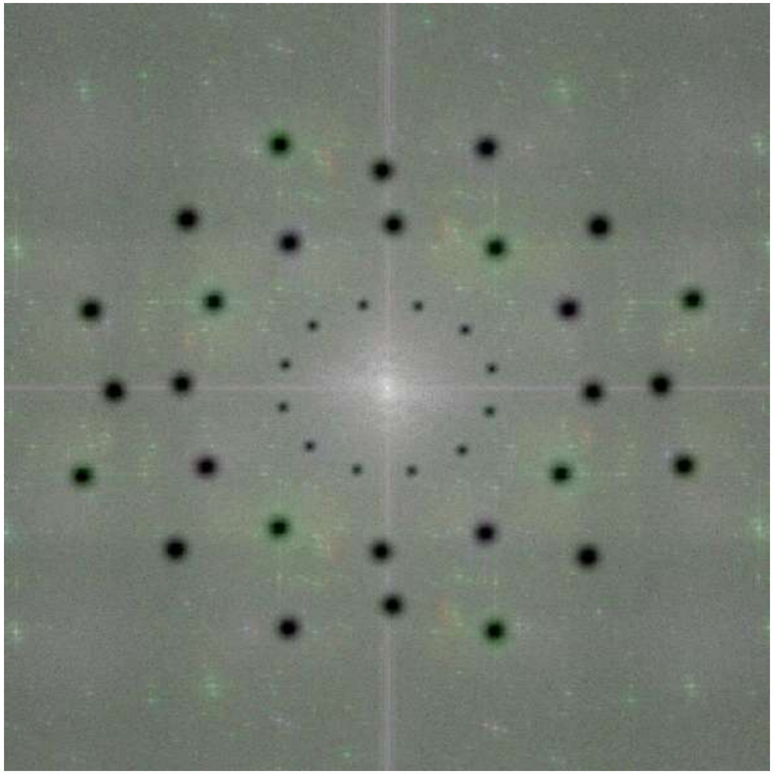
Notch Filtering


frequency masked 2


log power spectrum

Notch Filtering

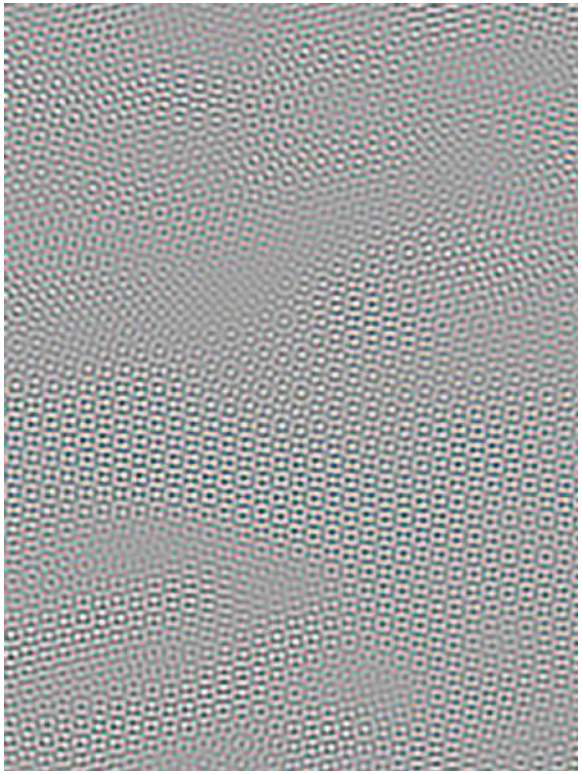
frequency masked 3



log power spectrum

Notch Filter Difference Images

original



difference

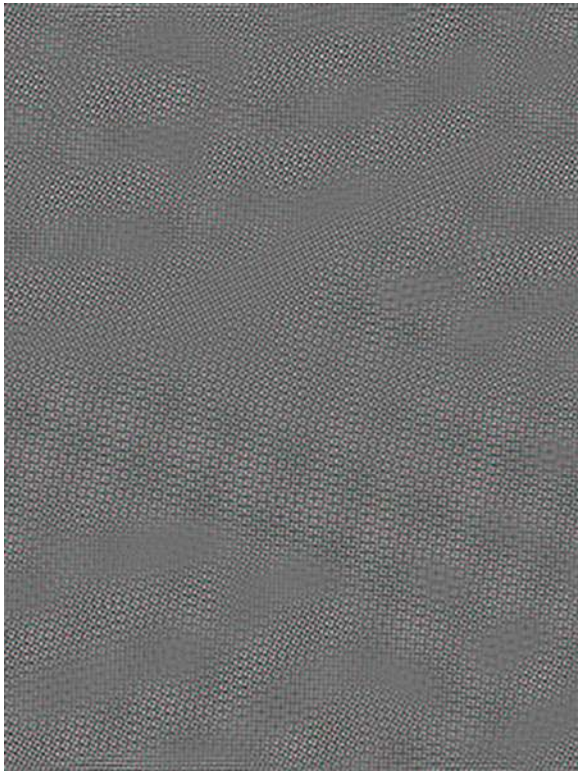
frequency masked 1

middle gray = 0, normalized

DLR

Notch Filter Difference Images

frequency masked 2



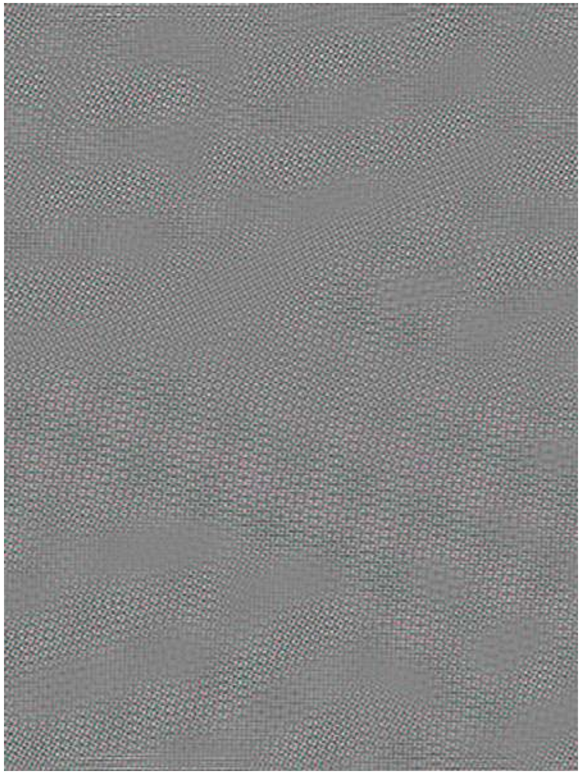
difference

original

middle gray = 0, normalized

Notch Filter Difference Images

original



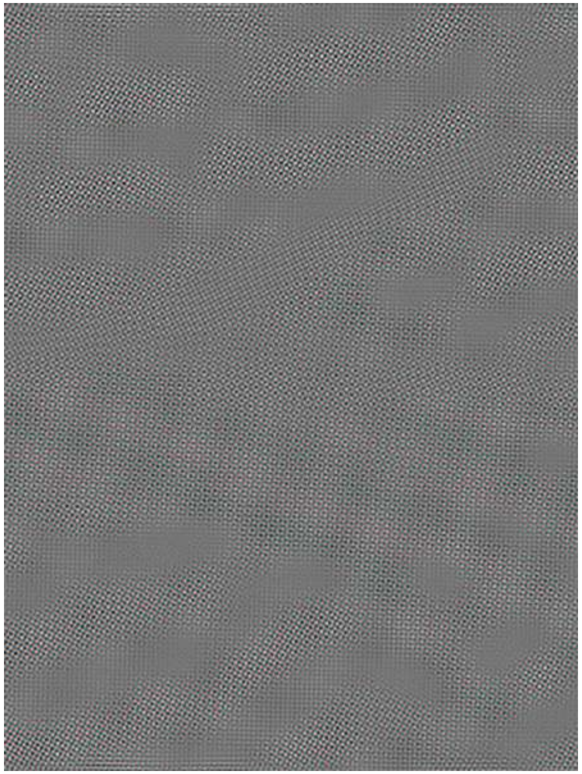
difference

frequency masked 3

middle gray = 0, normalized

Notch Filter Difference Images

frequency masked 1



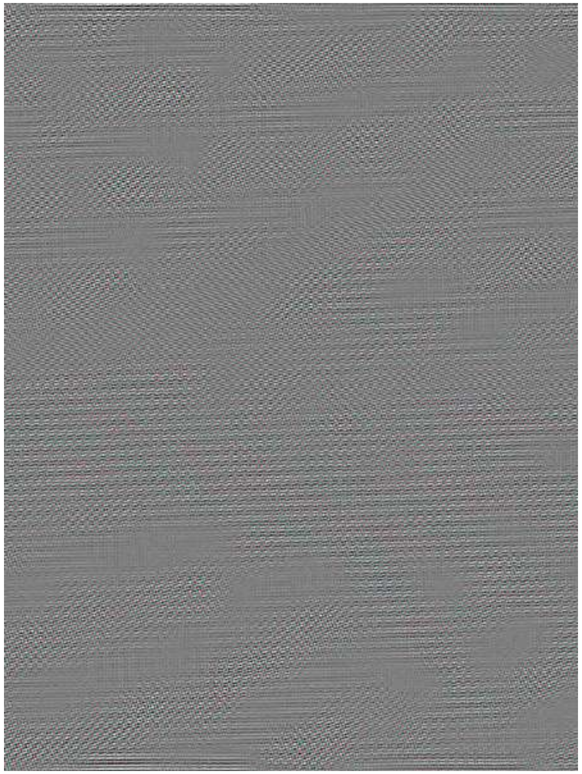
difference

frequency masked 2

middle gray = 0, normalized

Notch Filter Difference Images

frequency masked 2



difference

frequency masked 3

middle gray = 0, normalized

Noise Enhancement: Problem with Sharpening

– The spectra of most natural images fall-off toward the high frequencies.

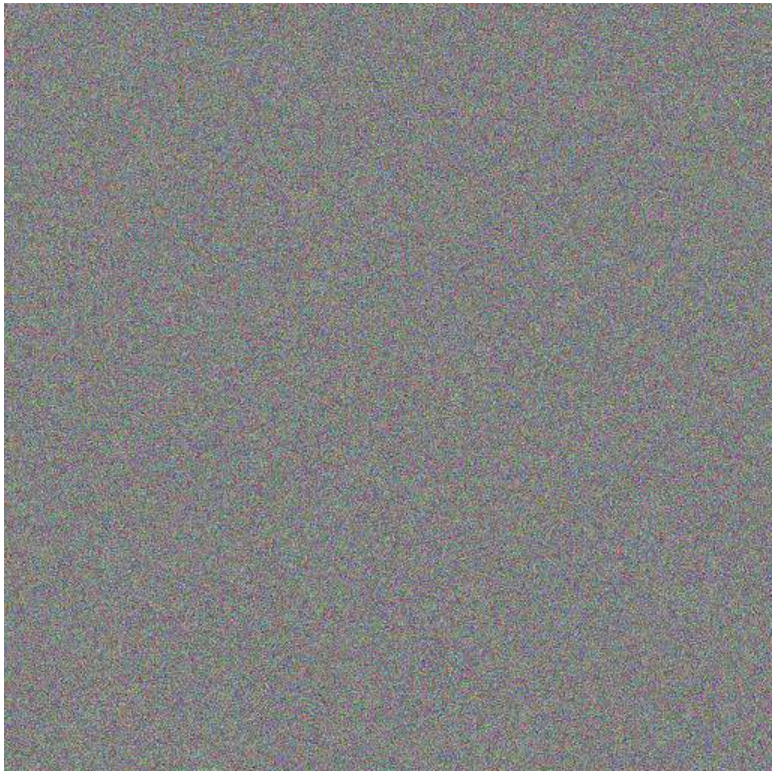
– IID noise has a flat spectrum.

– Therefore, at some relatively high frequency (HF) the energy in the noise is greater than that in the uncorrupted image.

– Sharpening multiplies the FT of the image by u and v (or by linear combinations of them) which, at HF, increases the noise more than the uncorrupted image.

Effects of Noise on Images

image



noise field

Effects of Noise on Images

What is this peak?

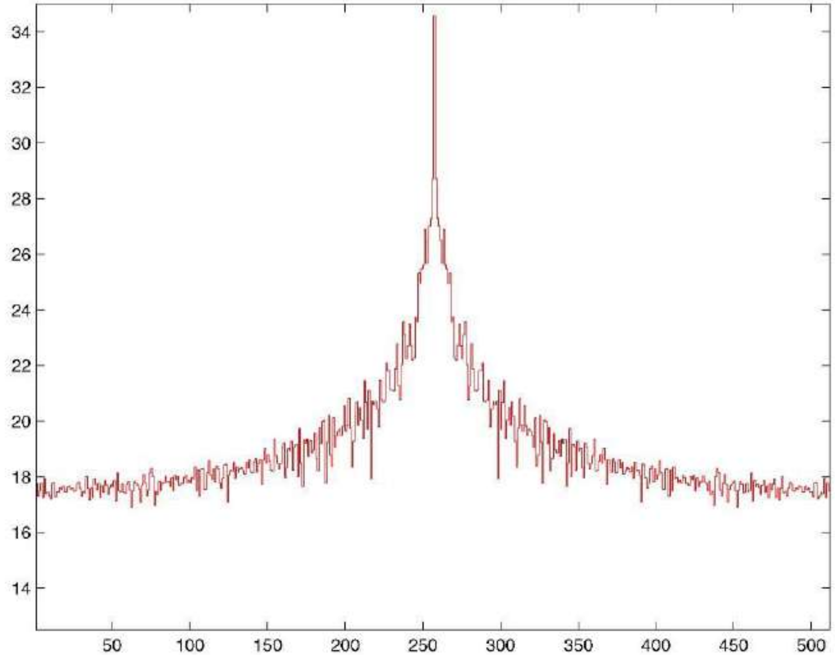
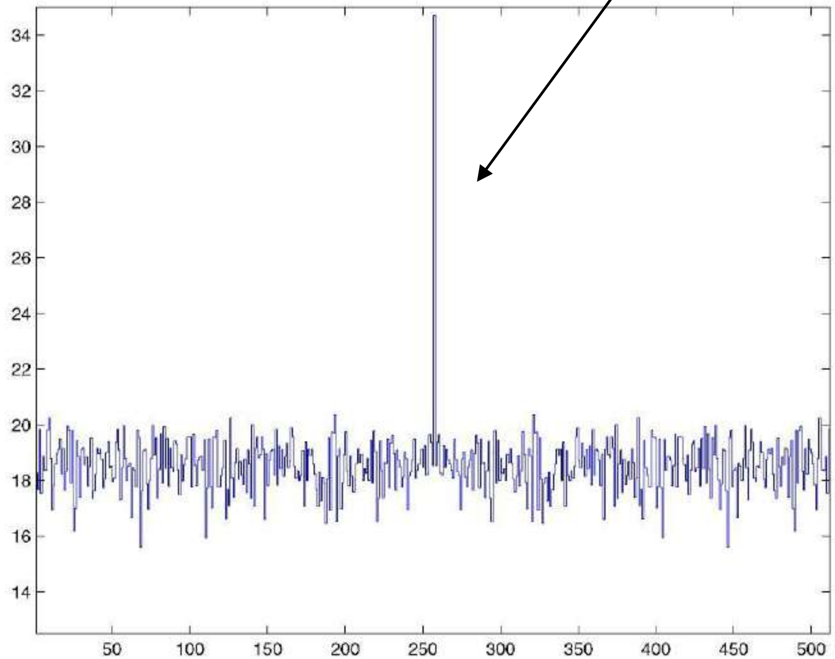


image center row log power spectrum



noise field center row log power spectrum

Recall: Fourier transform of an impulse at (0,0) is a constant.

Effects of Noise on Images

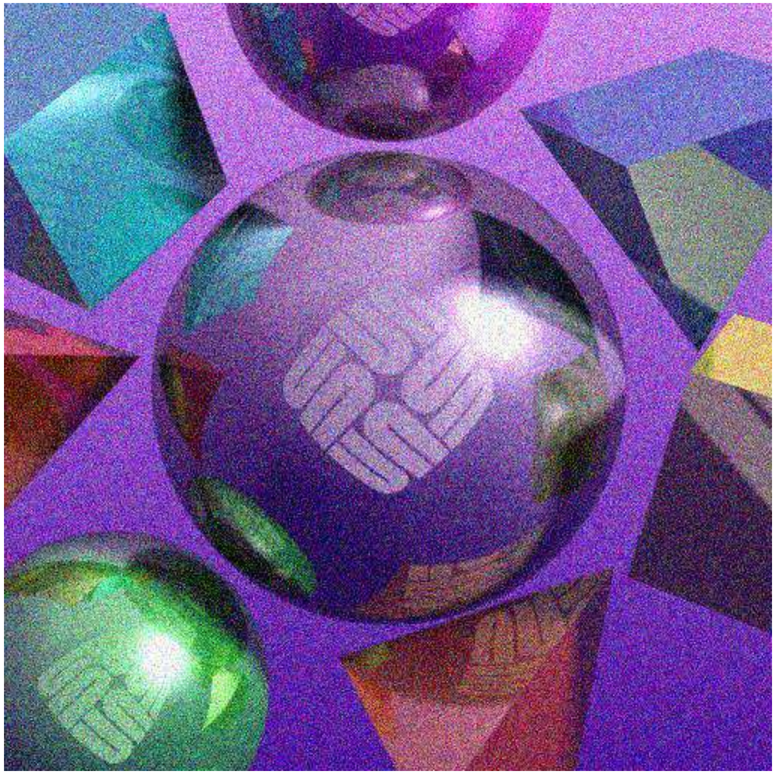


image + noise field

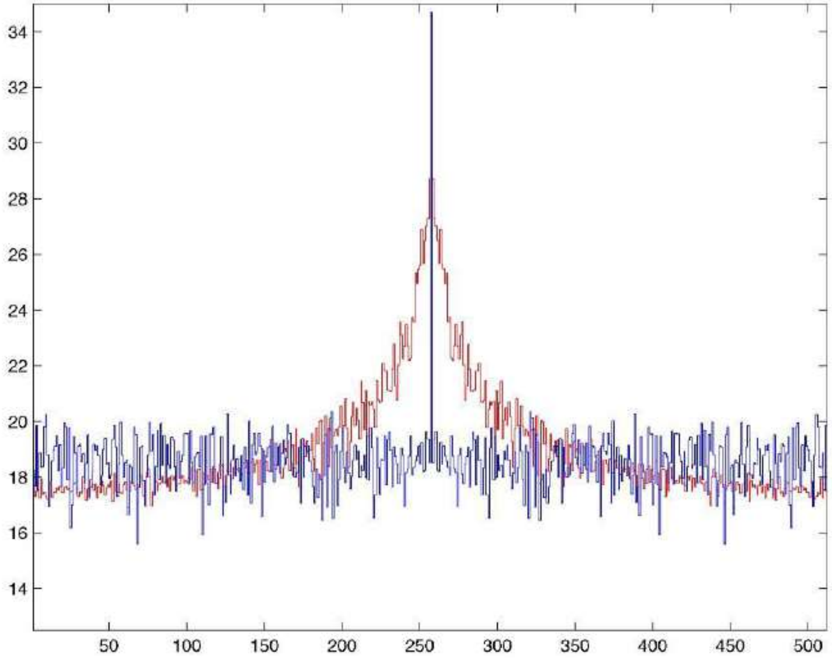
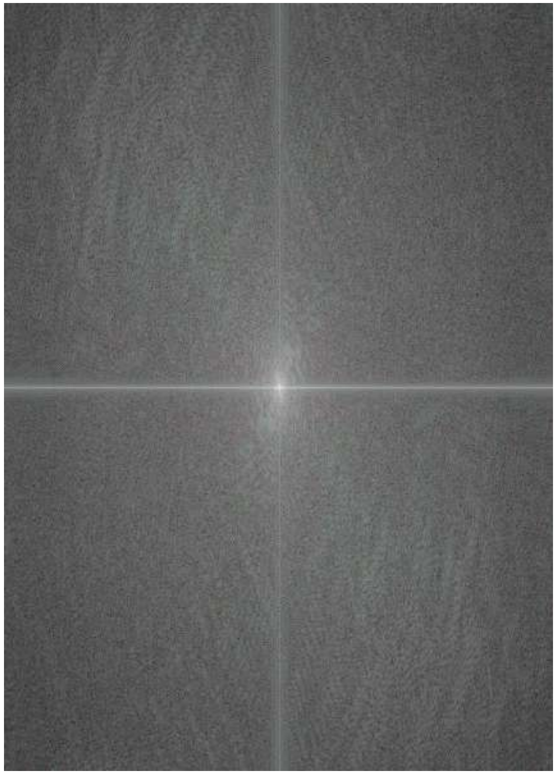
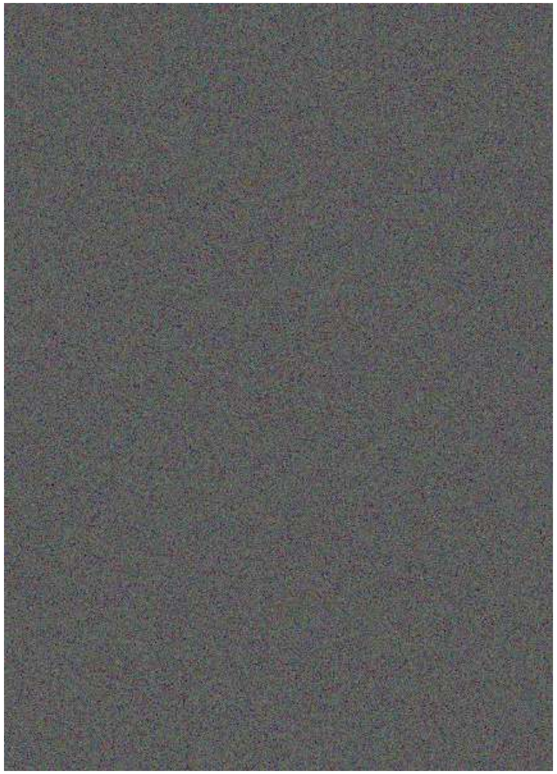


image + noise field center row log PS

Effects of Noise on Images (Power Spectra)

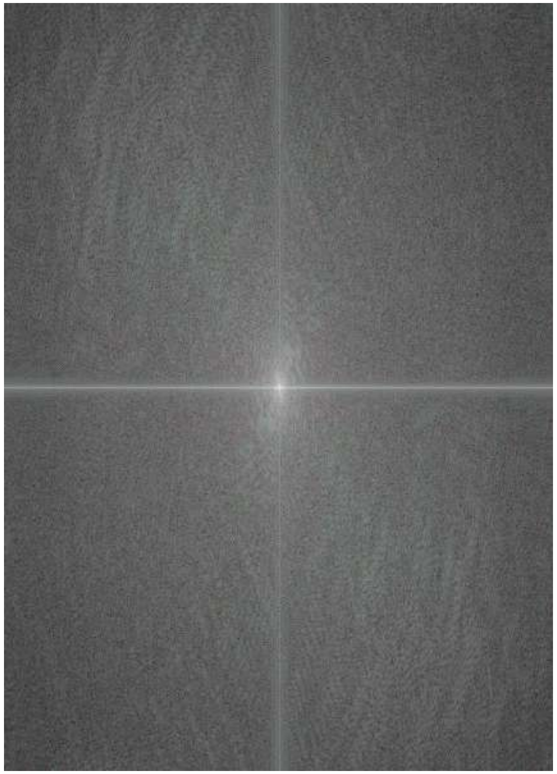


original image

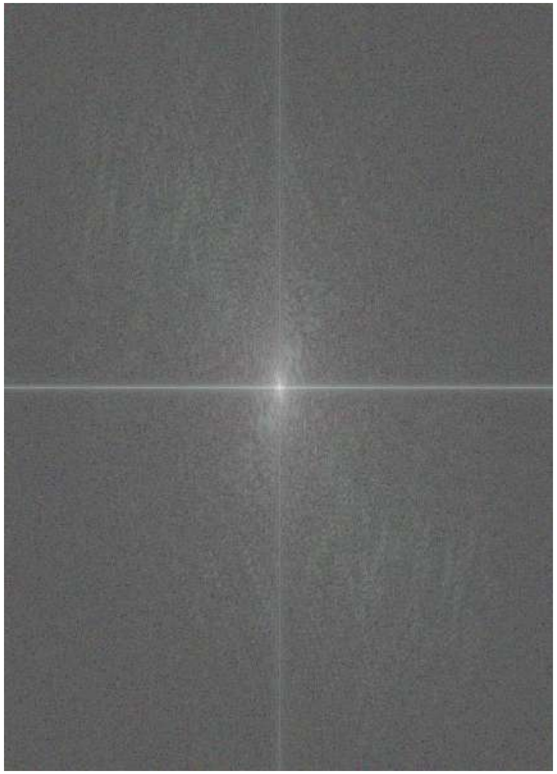


noise image

Effects of Noise on Images (Power Spectra)

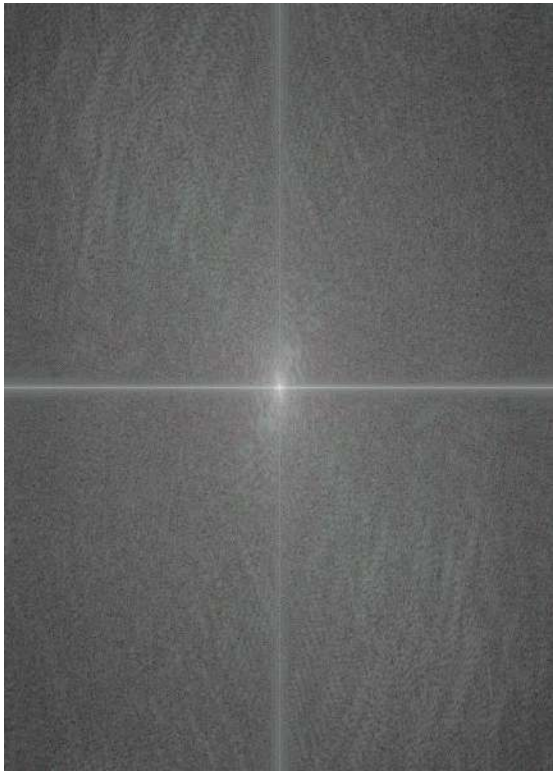


original image

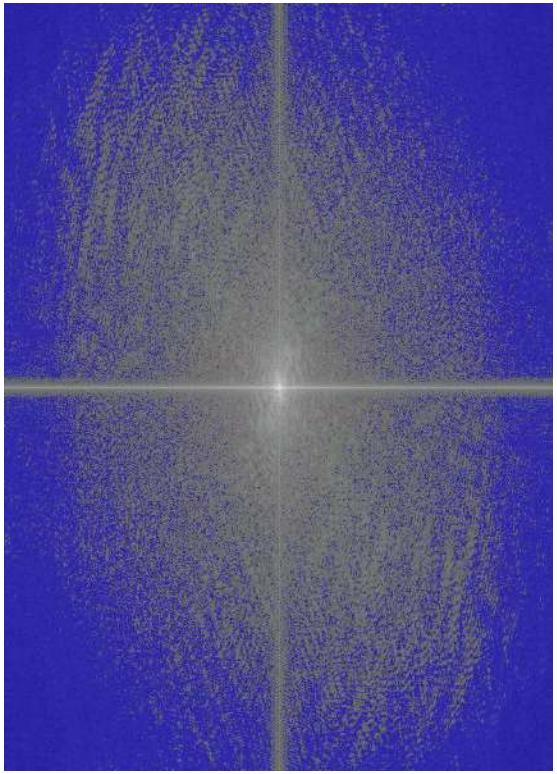


noisy image

Effects of Noise on Images (Power Spectra)

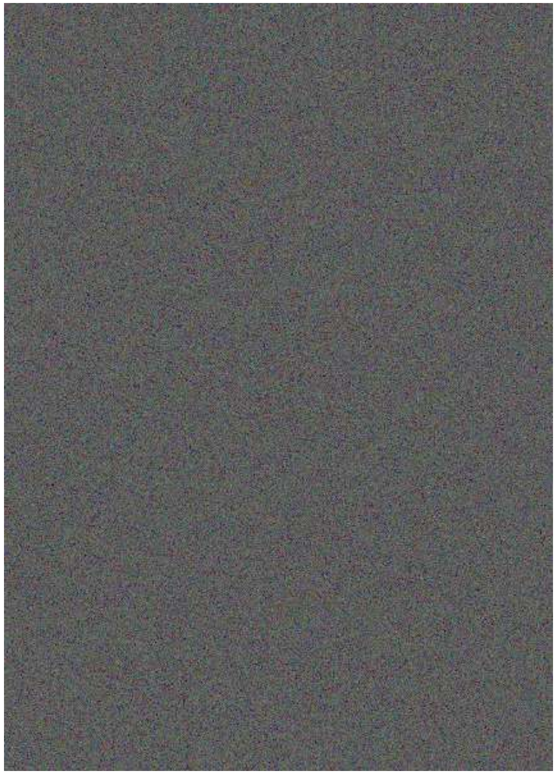


original image

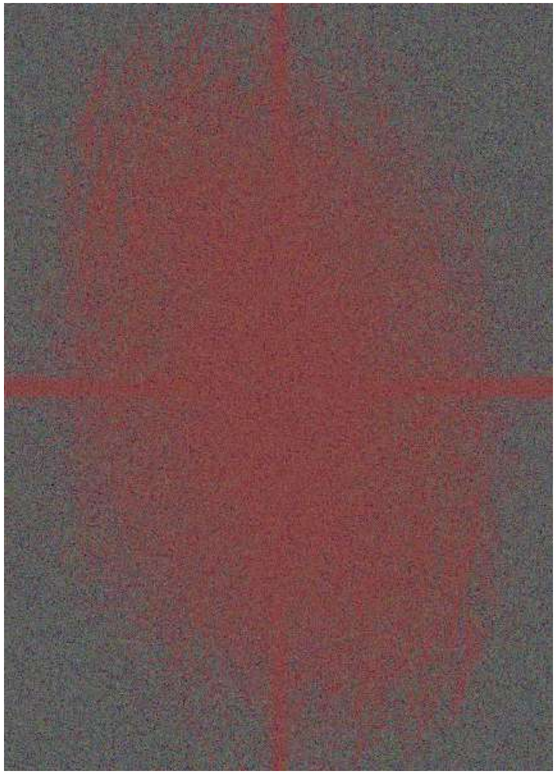


blue indicates noise > image

Effects of Noise on Images (Power Spectra)

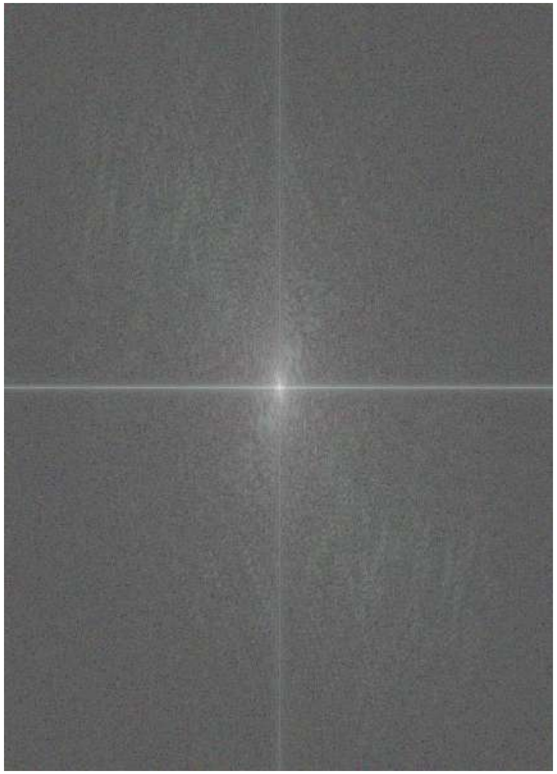


noise image



red indicates image > noise

Effects of Noise on Images (Power Spectra)



noisy image

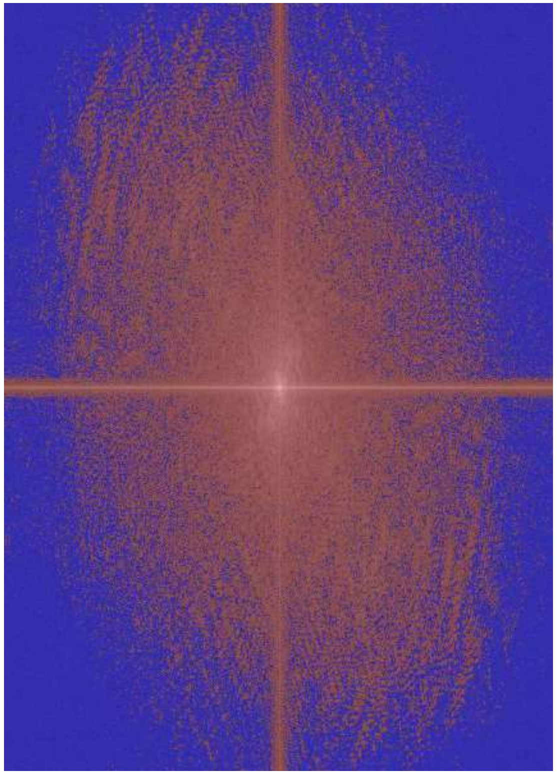


image & noise

Contextual Analysis of Image Elements

Daniele Cerra, German Aerospace Center (DLR)

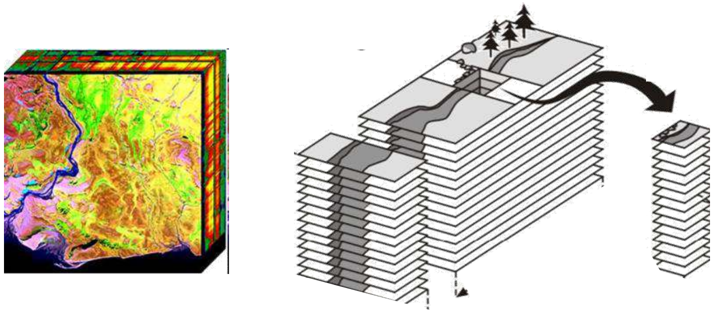
DLR

Summary

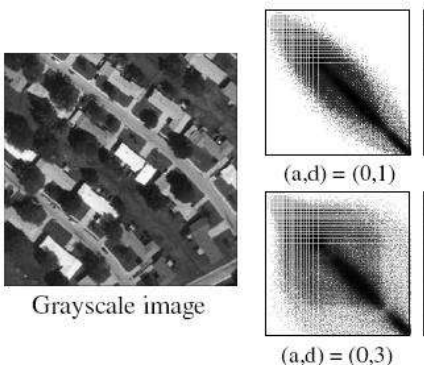
- Image Acquisition
- Image enhancement
- Sampling & Aliasing
- Image Features
  - Spectral Features
    - We will see more about this in the introduction to hyperspectral remote sensing
  - Features based on relations between pixels
- Image Clustering
- Image Classification

Which features can we extract from an image?

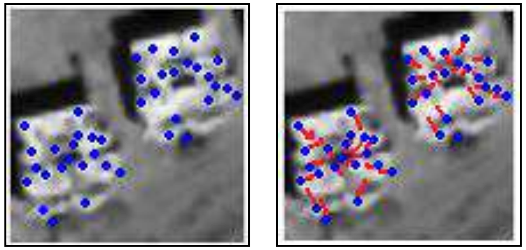
Pixel Value for each band



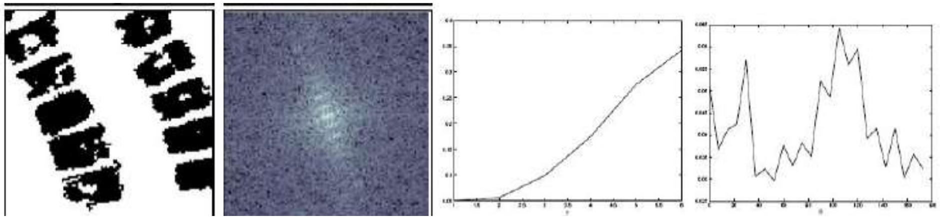
Texture Parameters



„Interesting“ Points



Power Spectrum Features



…and more!

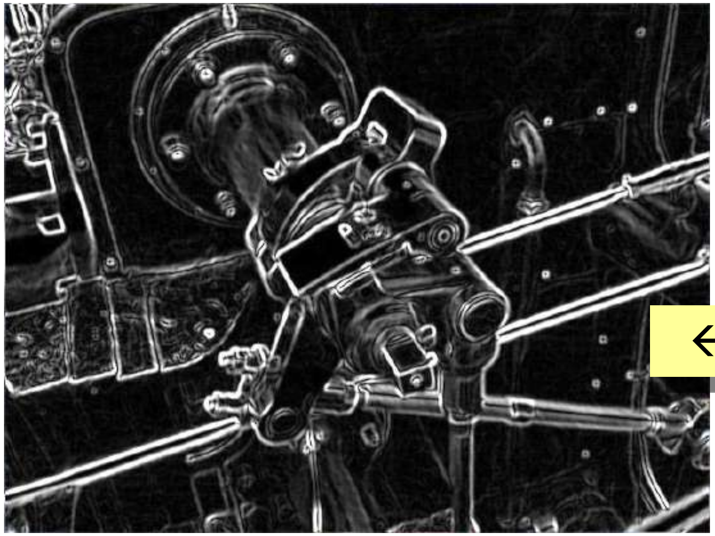
Context Analysis: Edge Extraction

Sobel operator

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel Operator

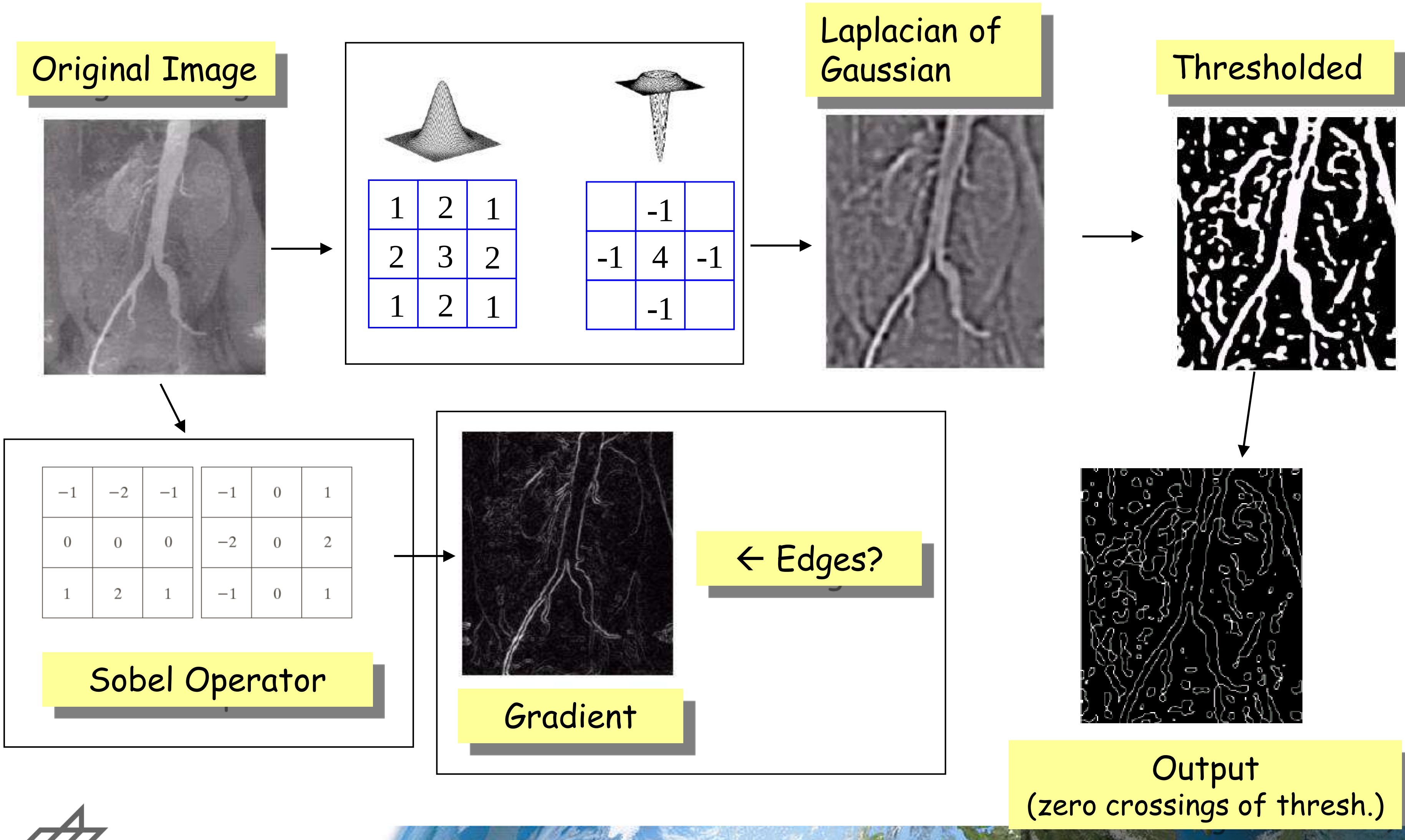


← Edges?

Gradient

DLR

Marr-Hildrith Edge detector

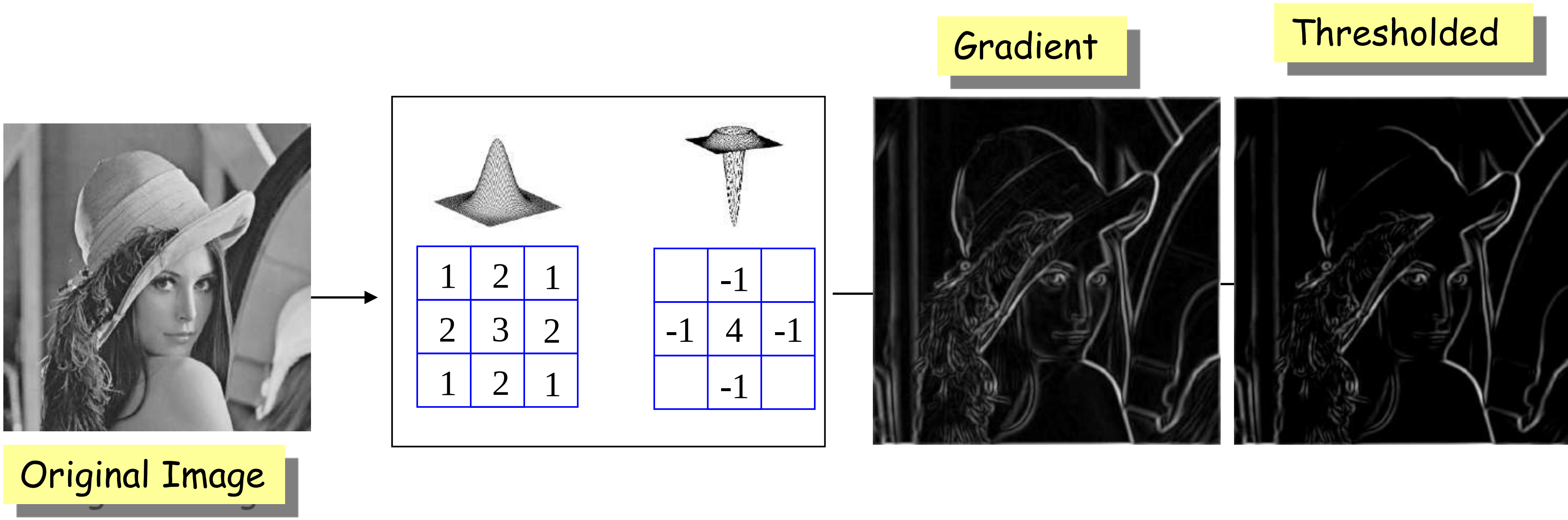


1	2	1
2	3	2
1	2	1

	-1	
-1	4	-1
	-1	

−1	−2	−1	−1	0	1
0	0	0	−2	0	2
1	2	1	−1	0	1

Canny Edge detector



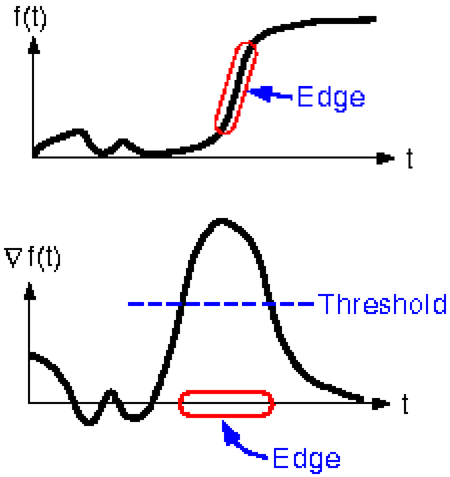
Canny edge detector

Original image

Magnitude of the gradient

Adapted from Chandra Kambhamettu

Canny edge detector

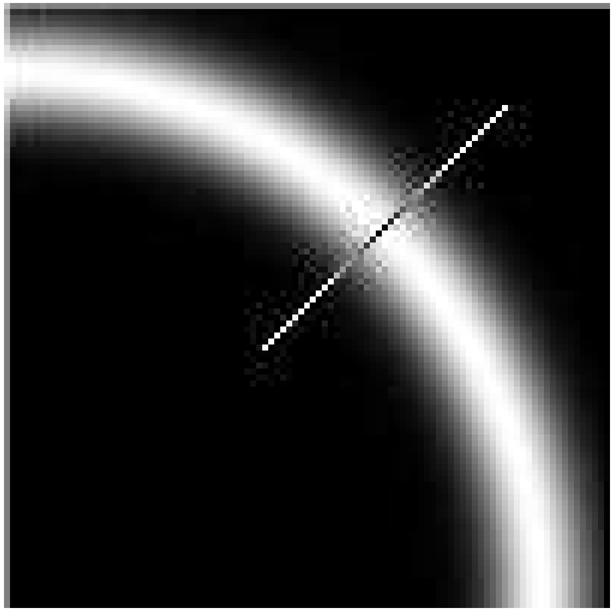


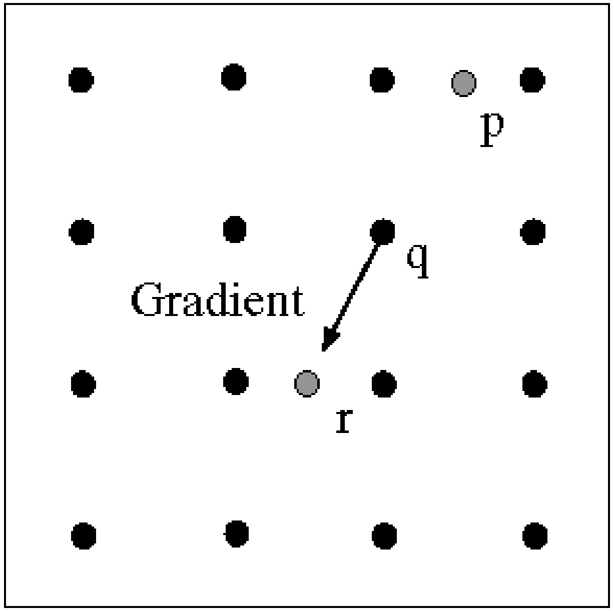
How to turn these thick regions of the gradient into curves?

Magnitude of the gradient

Canny edge detector

- Non-maxima suppression:
  - Check if pixel is local maximum along gradient direction.
  - Select single max across width of the edge.
  - Requires checking interpolated pixels p and r.
  - This operation can be used with any edge operator when thin boundaries are wanted.





Canny edge detector

courtesy of G. Loy

Original image

Gradient magnitude

Non-maxima suppressed

Canny edge detector

Problem: pixels along this edge did not "survive" the thresholding

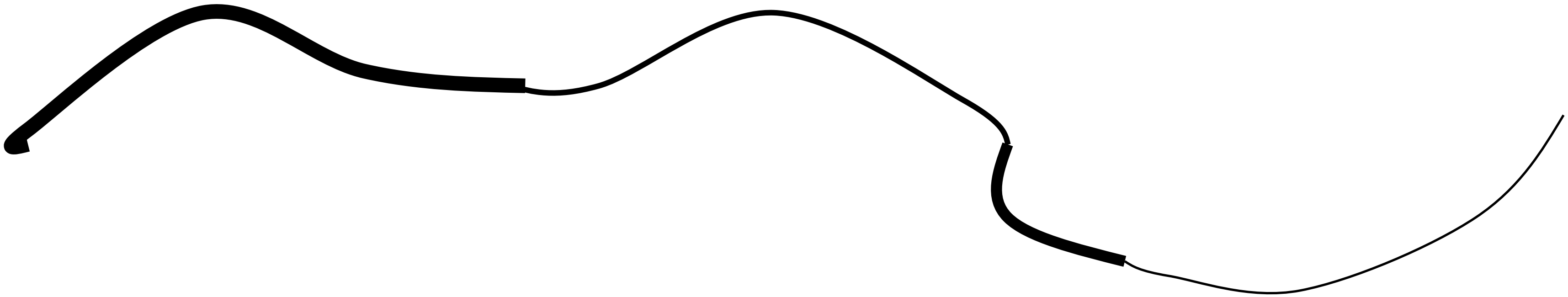
Original image

courtesy of G. Loy

Strong edges

Canny edge detector

- Hysteresis thresholding:
  - Use a high threshold Th to start edge curves, and a low threshold Tl to continue them.



- Select the pixels with value v > Th
- Then collect the pixels with value v > Tl that are connected to selected pixels

Canny Edge Detector: Final Result

Original image

gap is gone

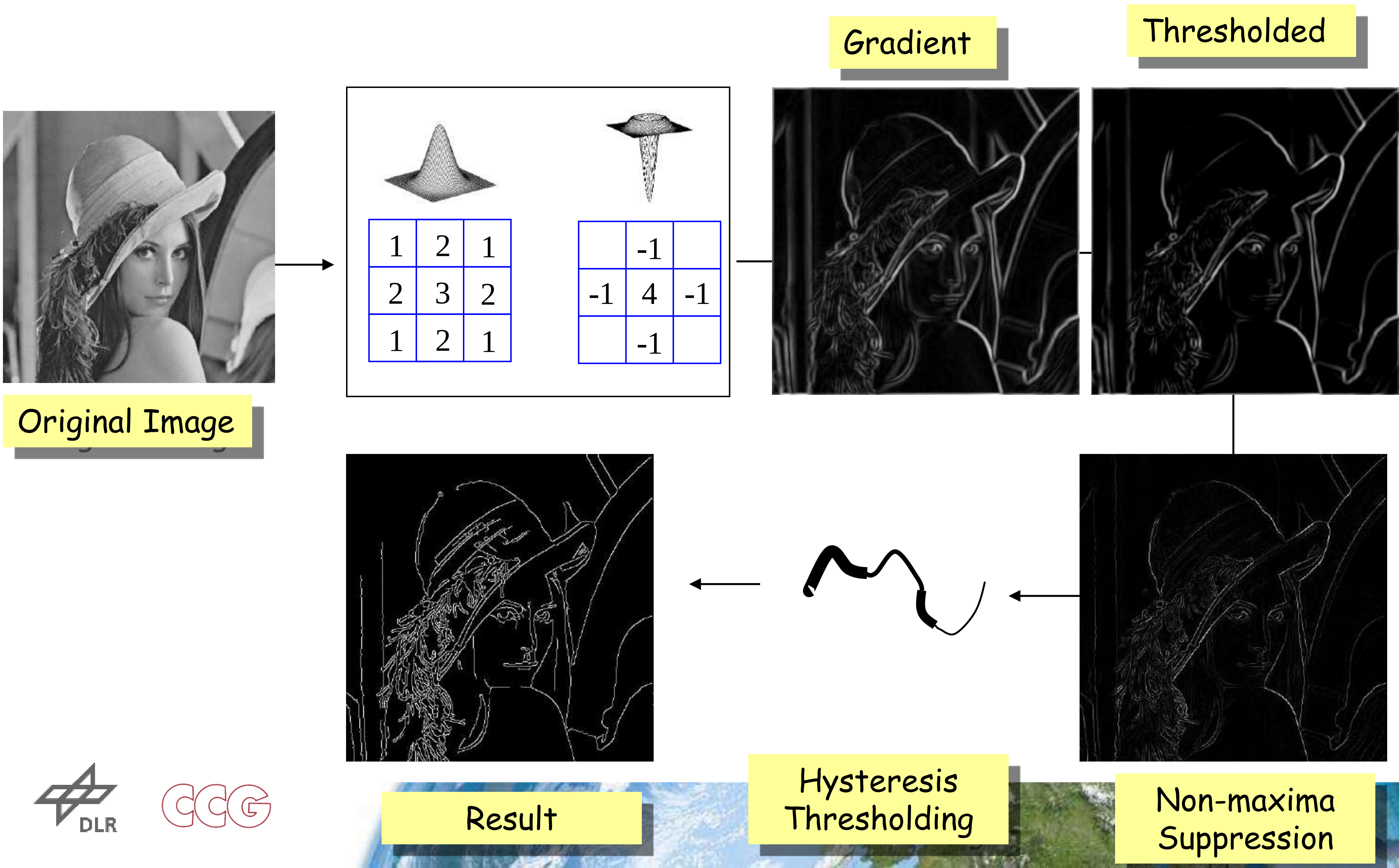
Strong + connected weak edges

Strong Edges (gradient > Thigh)

Weak Edges (gradient > Tlow)

courtesy of G. Loy

Canny Edge detector



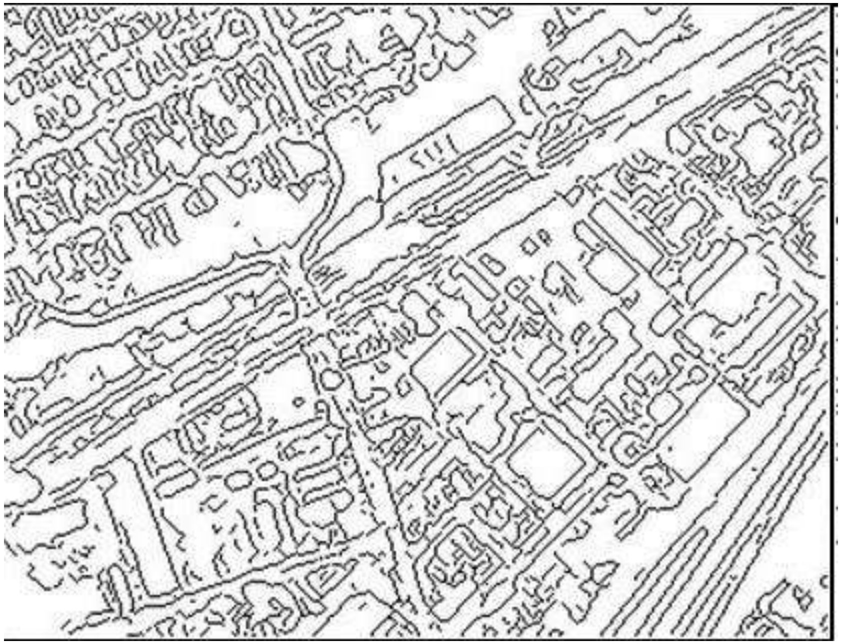
- Canny algorithm is very sensitive to its parameters, which need to be adjusted for different application domains.
    - Smoothing parameter σ
    - Threshold for strong edges
    - Threshold for weak edges


original


Canny with $\sigma = 1$


Canny with $\sigma = 2$

Canny Edge Detector in Remote Sensing



Texture

- An important approach to image description is to quantify its texture content.
- Texture gives us information about the spatial arrangement of the colors or intensities in an image.

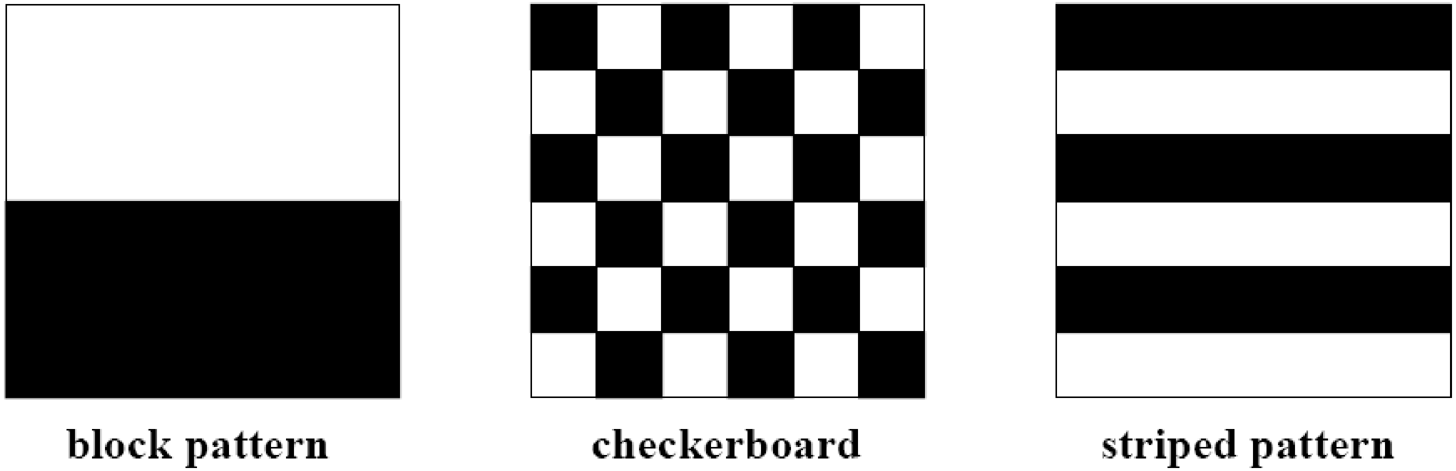
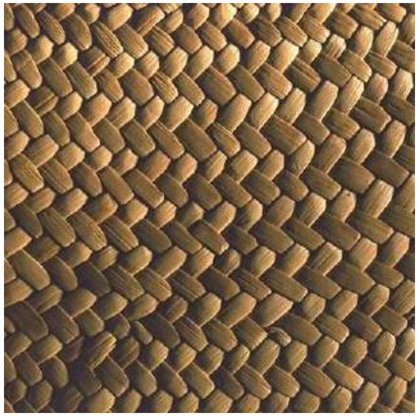


Figure 7.2: Three different textures with the same distribution of black and white.

Texture


Bark


Bark


Fabric


Fabric


Fabric


Flowers


Flowers

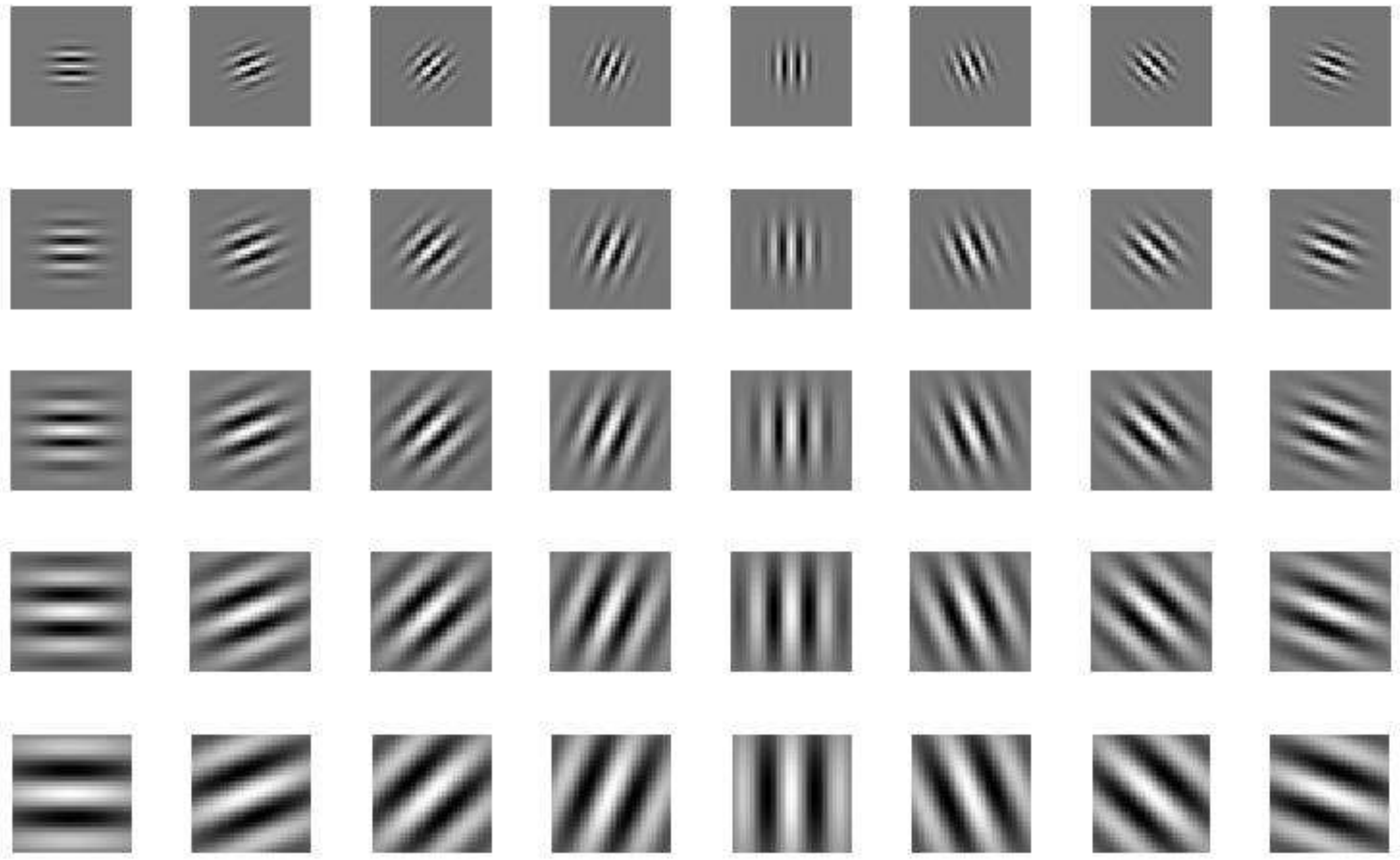

Flowers

Texture Analysis → Local Analysis of Pixels Distribution

Statistical moments

- The easiest thing we can do is to check the statistics of the histogram of a small window in the image
  - Mean
  - Standard deviation
  - Variance →
  - Kurtosis
  - Skewness…
- This gives us hints on the strength of the texture only
- How to characterize it better?

Gabor Texture Features

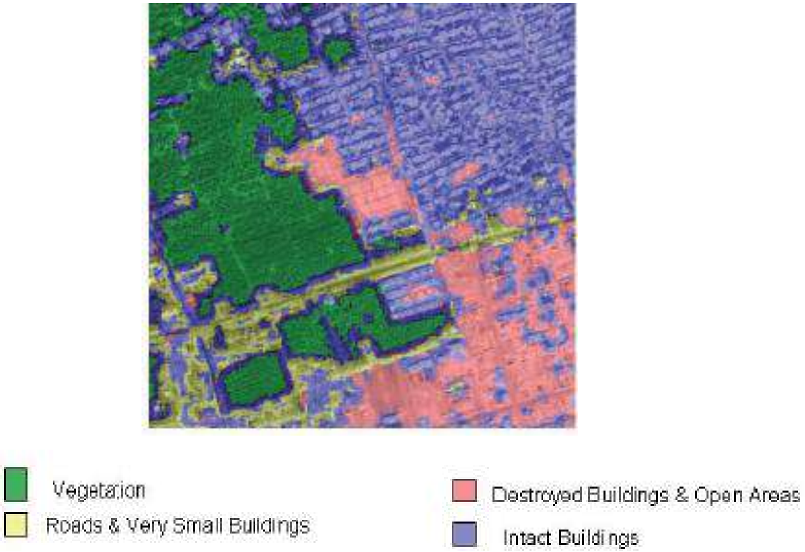


Sample filter bank

Texture Classification

Bam, Iran, suffered an earthquake in 2003

IKONOS image acquired in the aftermath of the earthquake

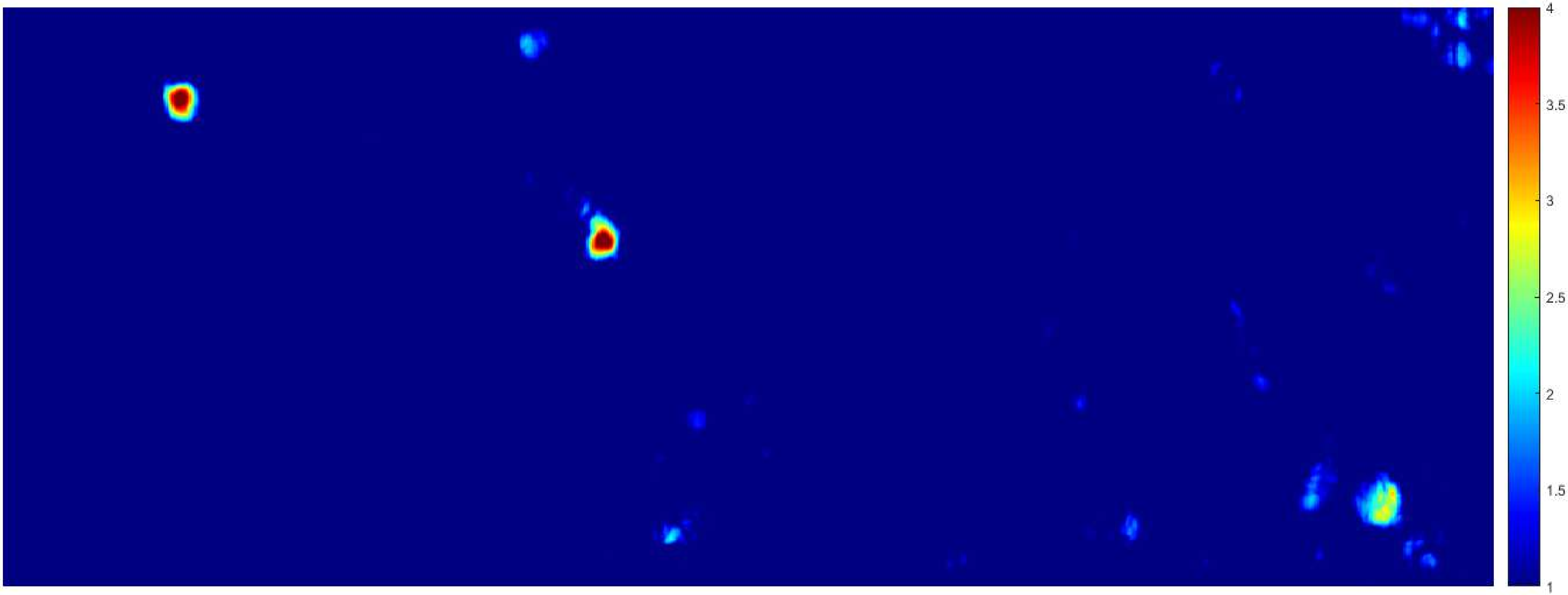


Classification obtained on the basis of the texture parameters only

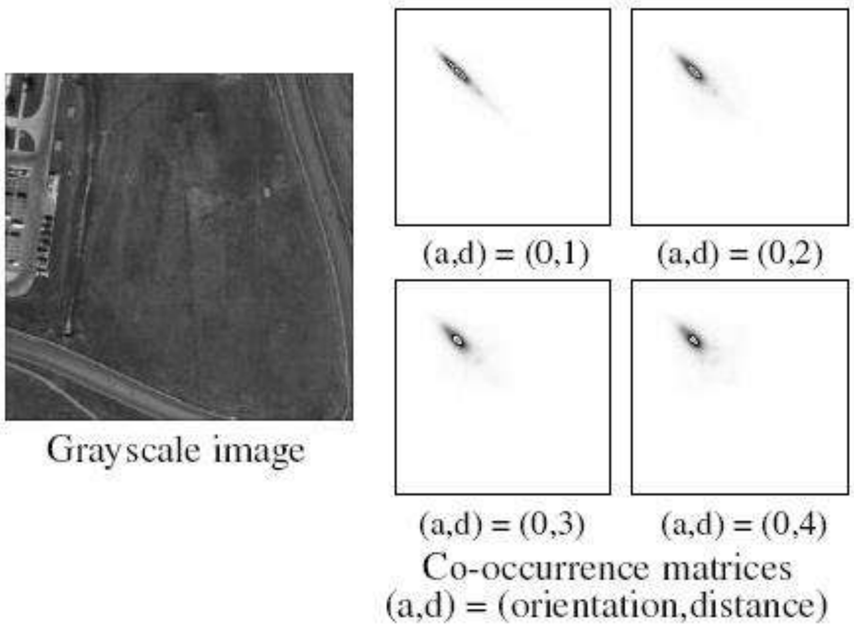
Change Detection based on Texture

Example: Palmyra – Temple of Bel: destroyed by IS (30.08.2015)

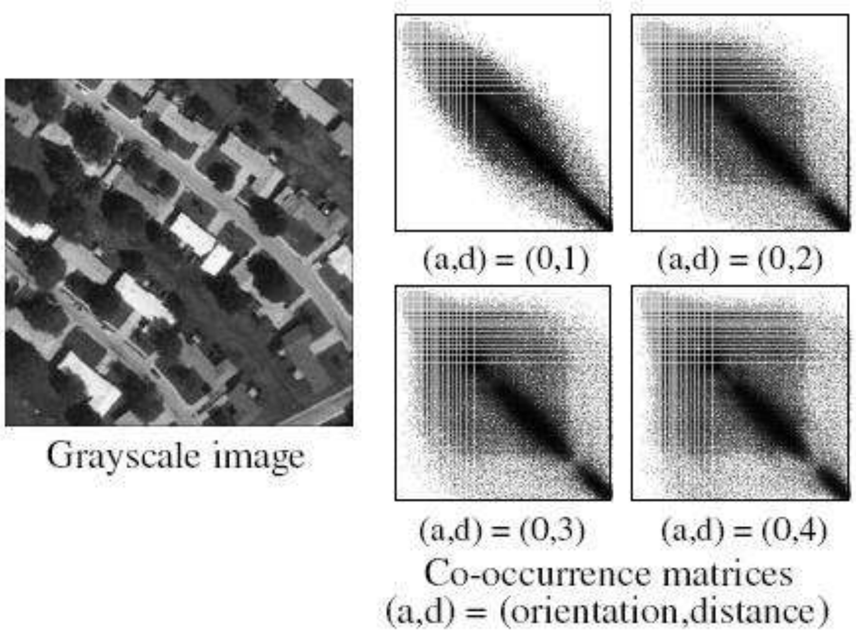
Palmyra: Difference of Gabor Features (based on texture values)



Co-occurrence matrices

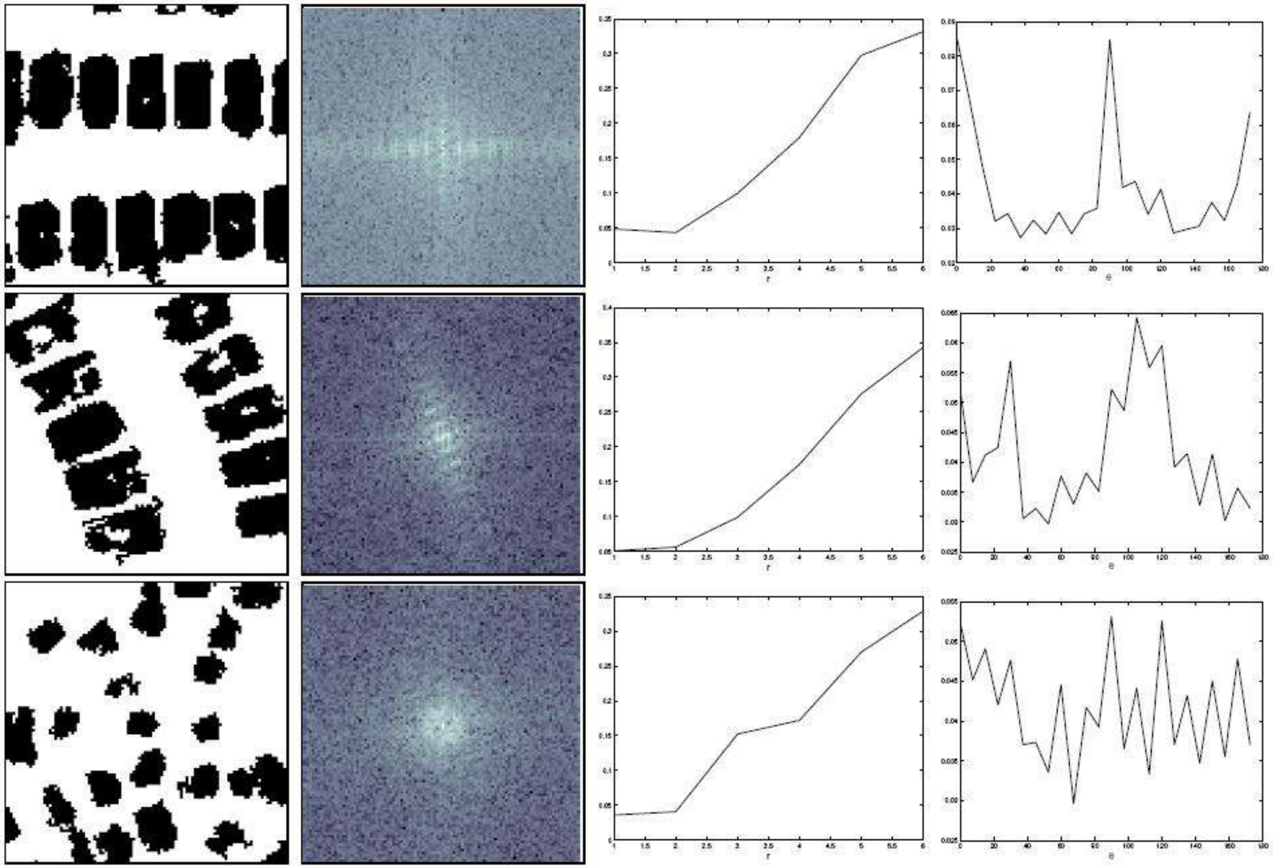


(a) Co-occurrence matrices for an image with a small amount of local spatial variations.



(b) Co-occurrence matrices for an image with a large amount of local spatial variations.

Fourier power spectrum

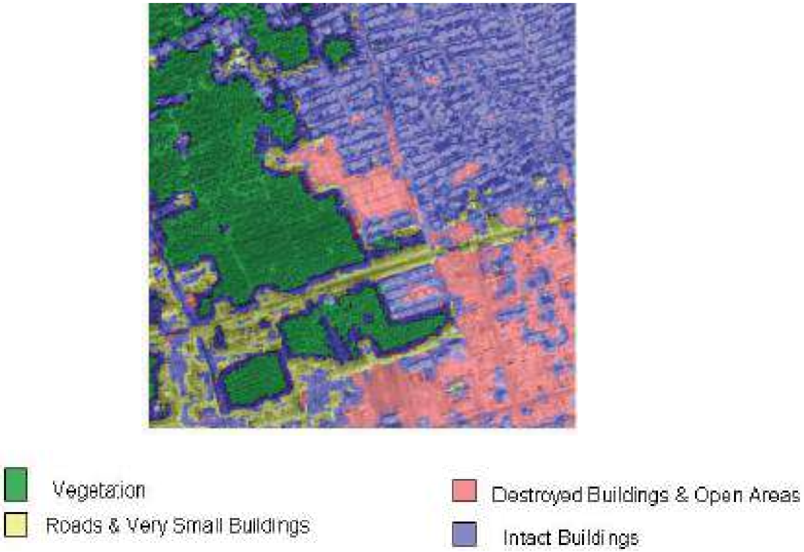


Example building groups (first column), Fourier spectrum of these images (second column), and the corresponding ring- and wedge-based features (third and fourth columns). X-axes represent the rings in the third column and the wedges in the fourth column plots. The peaks in the features correspond to the periodicity and directionality of the buildings, whereas no dominant peaks can be found when there is no regular building pattern.

Texture Classification

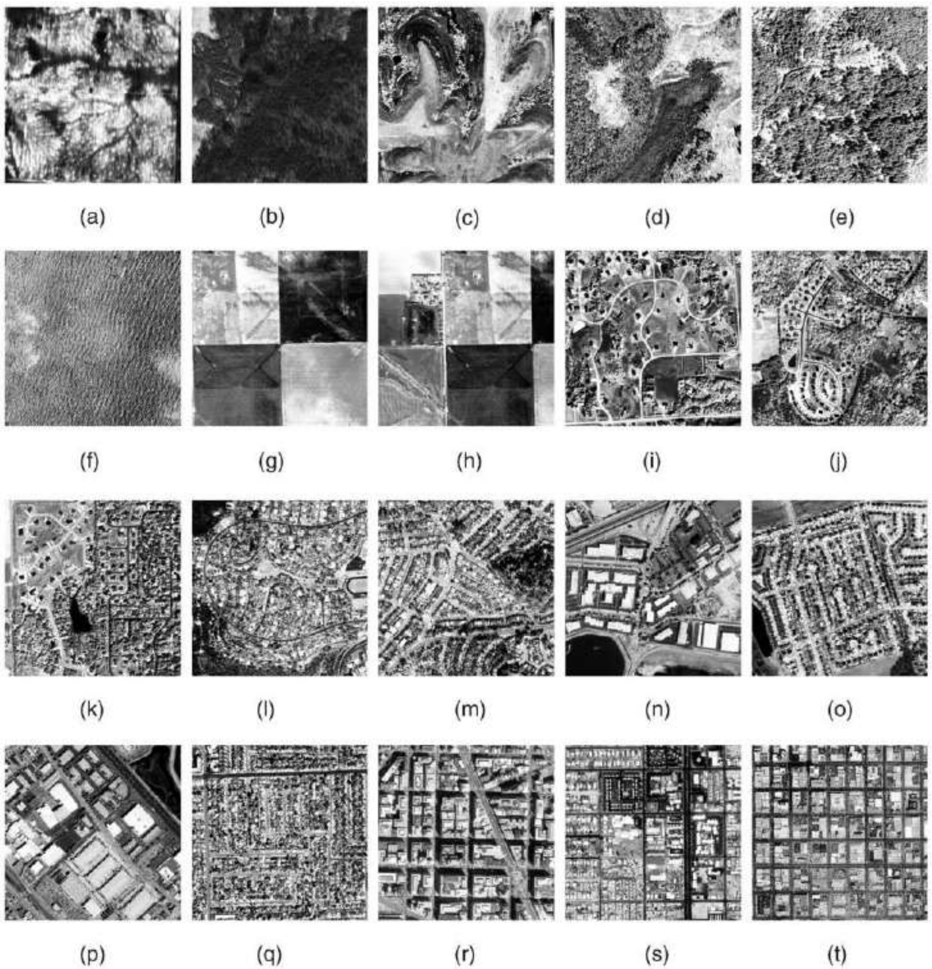
Bam, Iran, suffered an earthquake in 2003

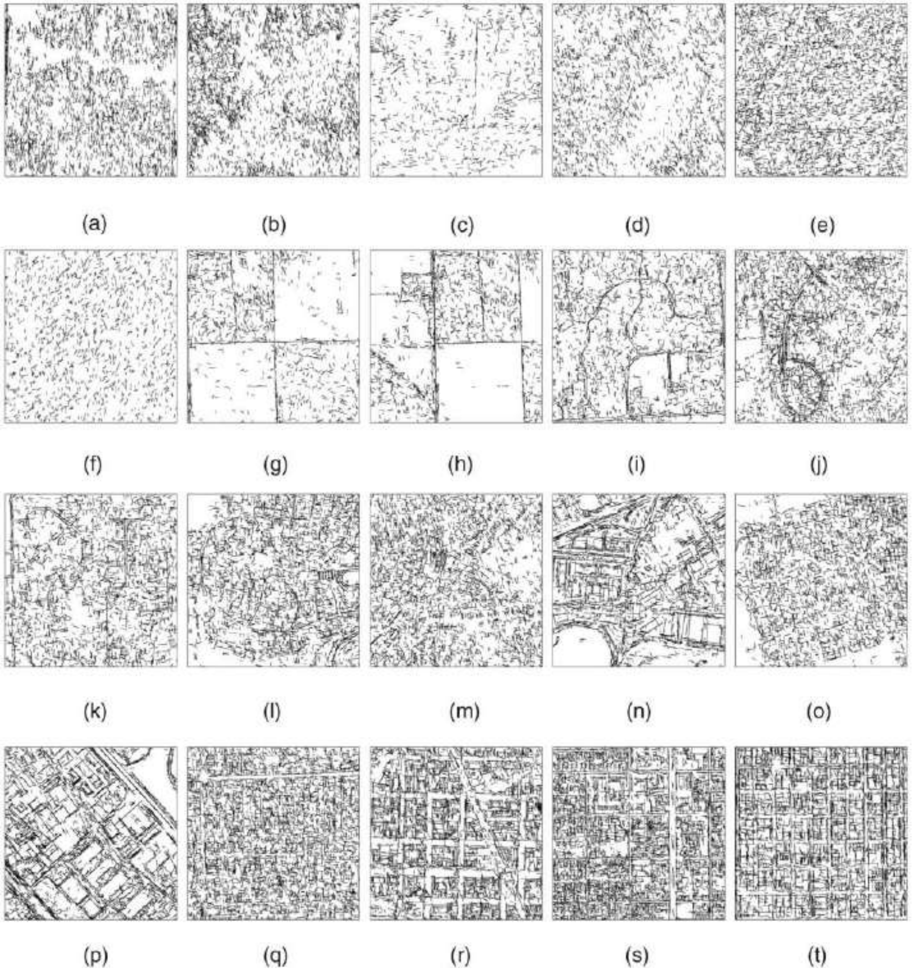
IKONOS image acquired in the aftermath of the earthquake



Classification obtained on the basis of the texture parameters only

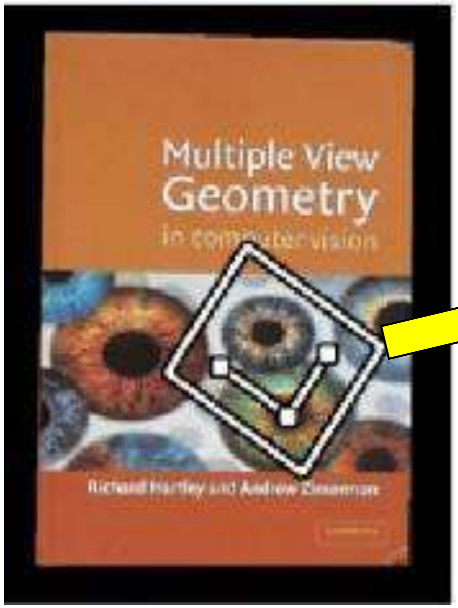
Edge texture

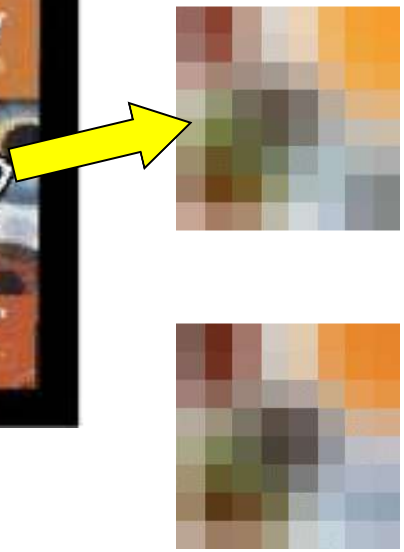




Satellite images sorted according to the amount of land development (left). Properties of the arrangements of line segments can be used to model the organization in an area (right).

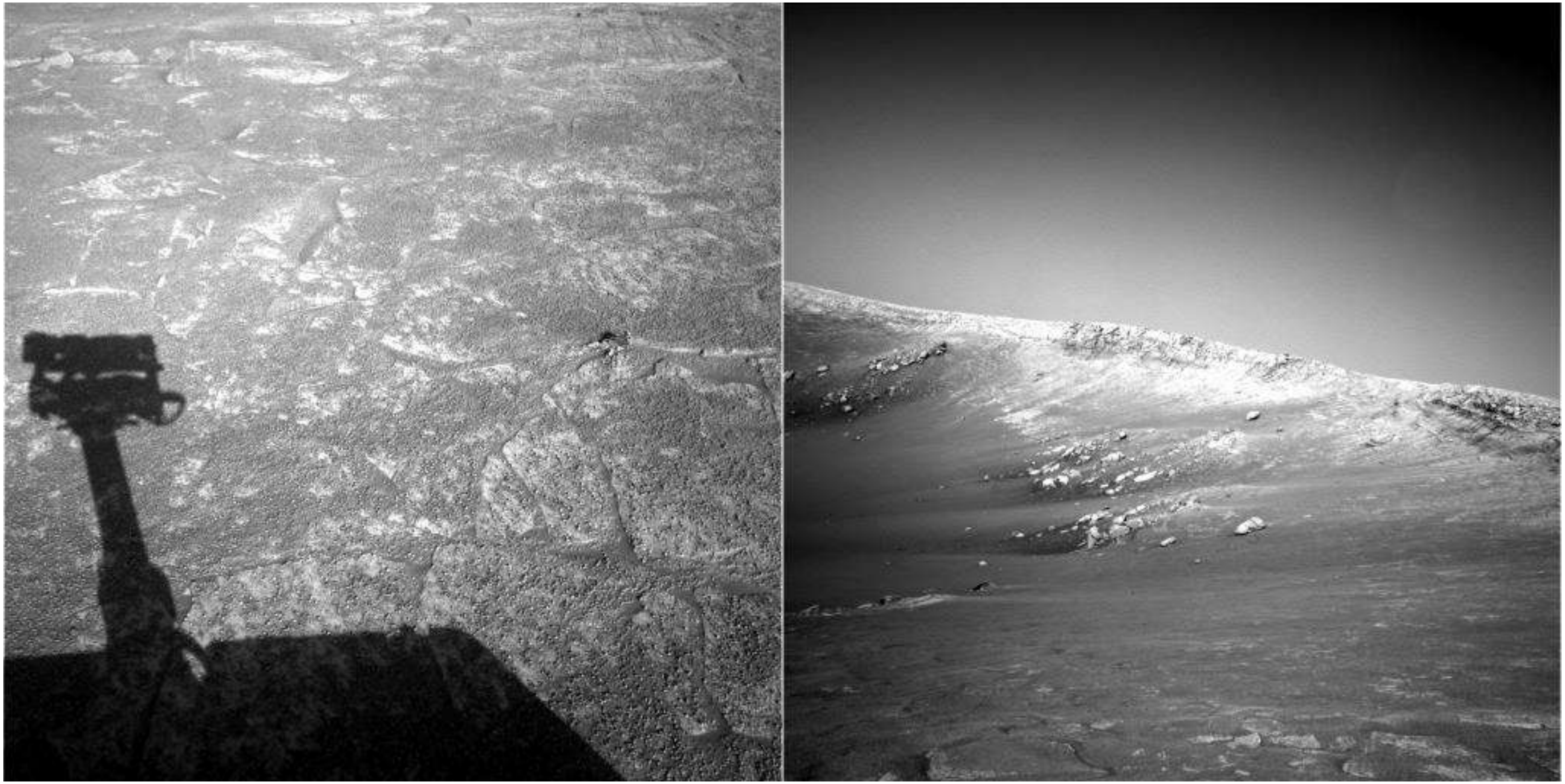
Matching based on "Interesting" Points





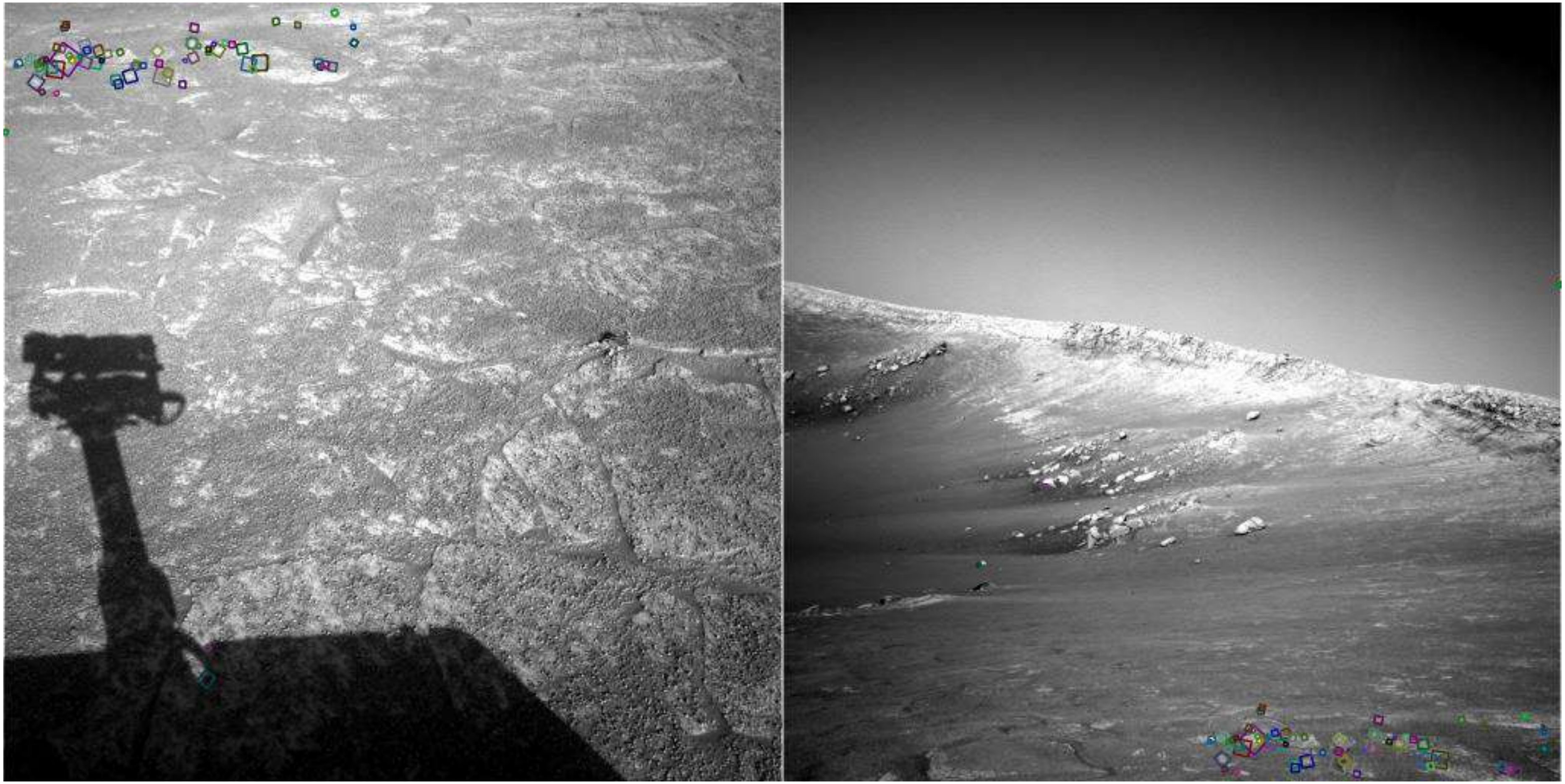
Object recognition: Find correspondences between feature points in training and test images.

Local Features Detectors for Image Matching



Two images from NASA Mars Rover: very hard matching case

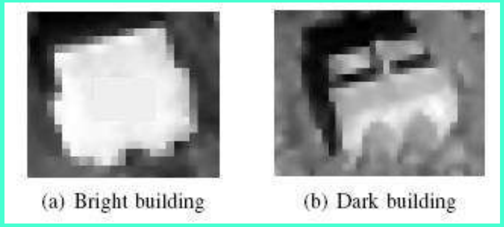
Local Features Detectors for Image Matching



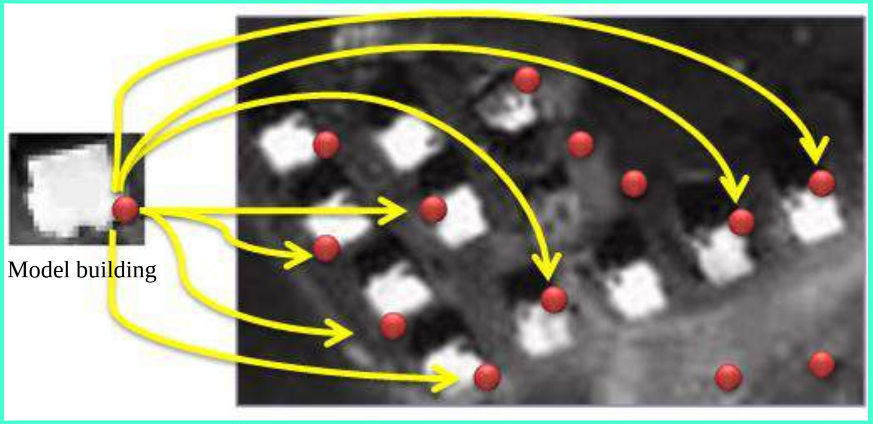
Two images from NASA Mars Rover: matching using local features

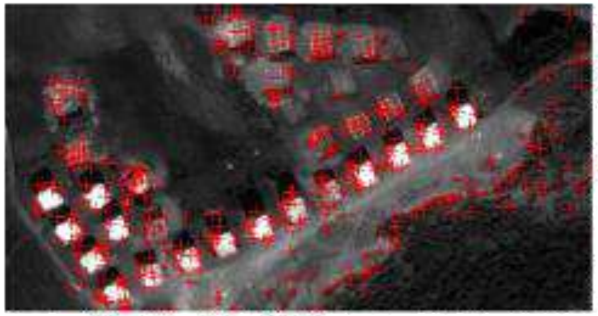
Remote Sensing: Building Detection

Model Building Database

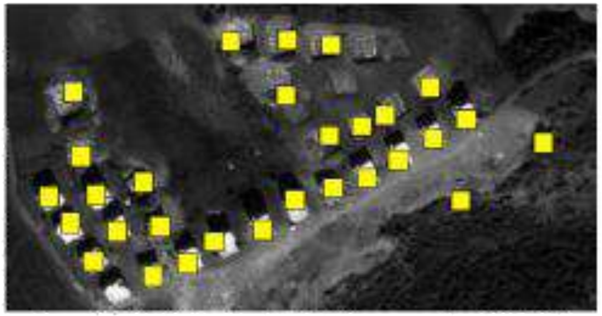


Each feature is matched with the most similar features



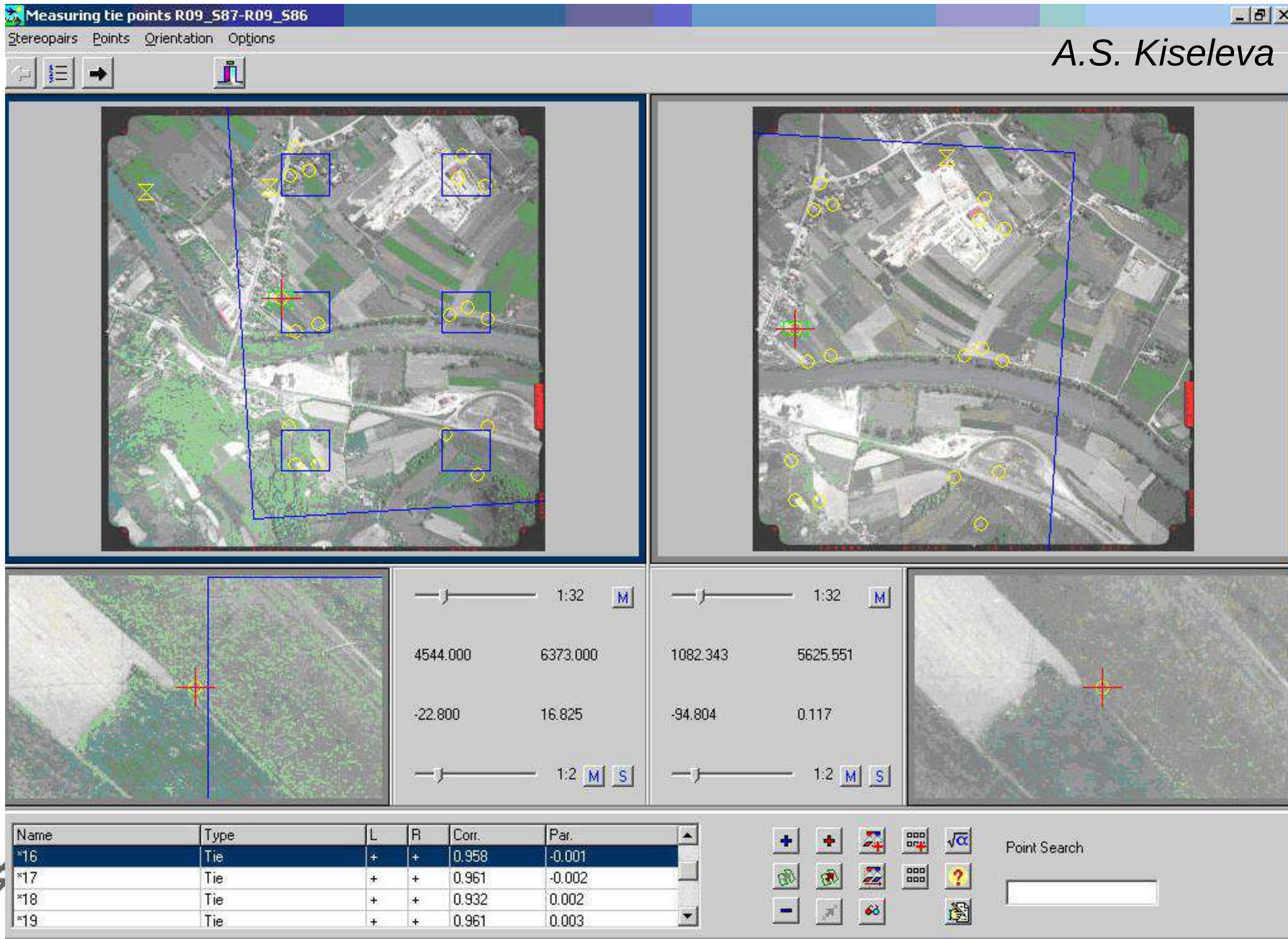


Detected „Interesting Points"



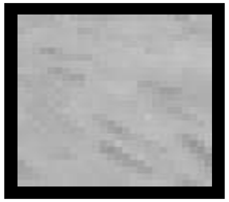
Detected buildings

What do you do when you choose Ground Control Points (GCP)?



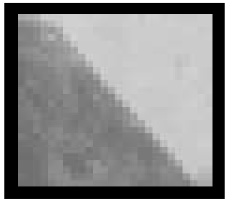
Local features

- What makes a good feature?
- We want uniqueness.
    - Look for image regions that are unusual.
    - Lead to unambiguous matches in other images.
- How to define “unusual”?



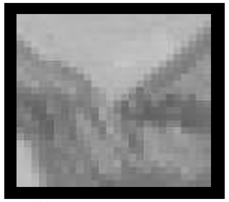
0D structure

not useful for matching



1D structure

edge, can be localized in 1D, not so good for matching



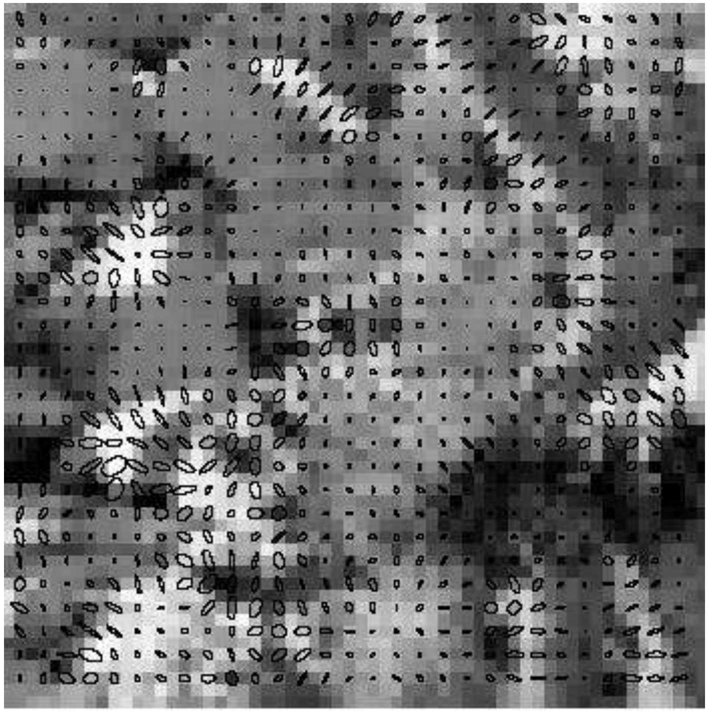
2D structure

corner, can be localized in 2D, good for matching

Which points make good features?

Good candidates are points with strong variations in all directions

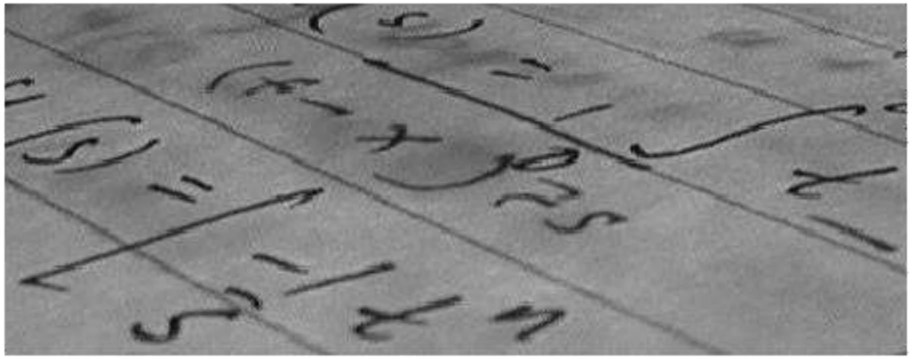
Full image

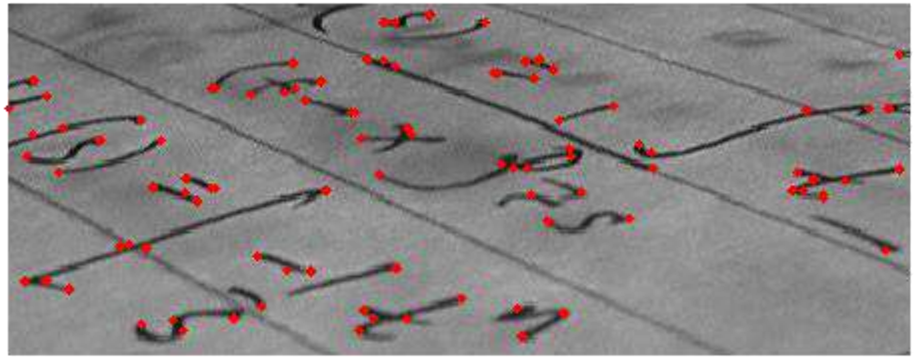


from Forsyth & Ponce

Detail of image with gradient covar-iance ellipses for 3 x 3 windows

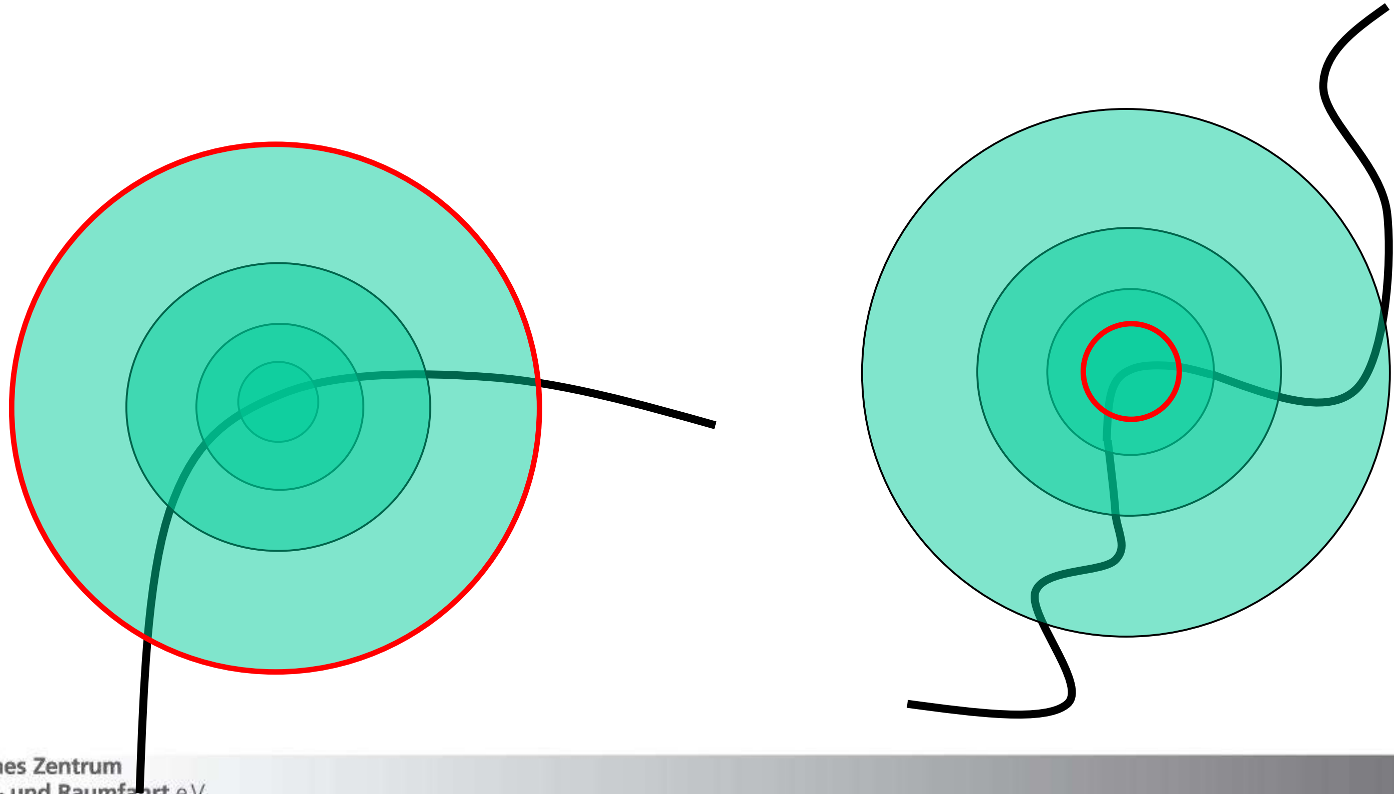
Sample Output of a Corner Detector



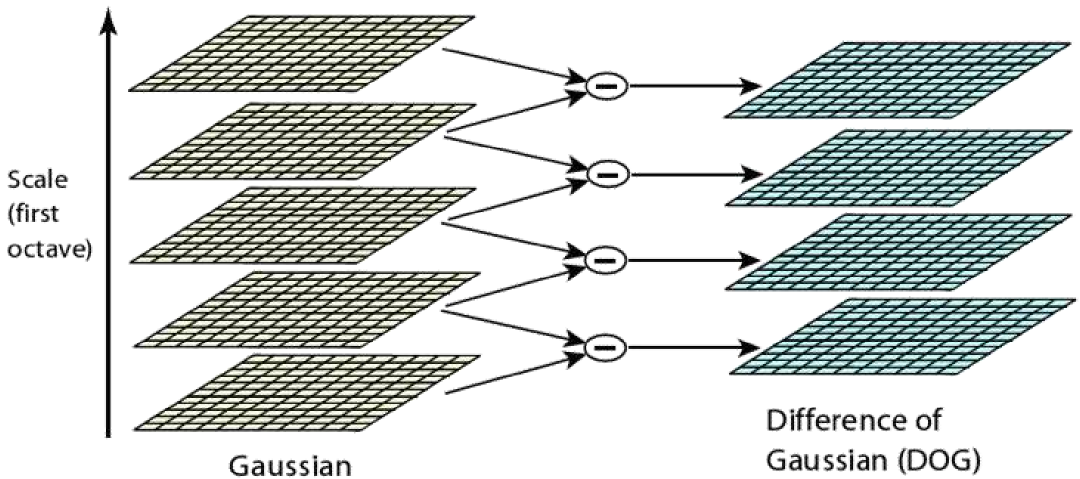


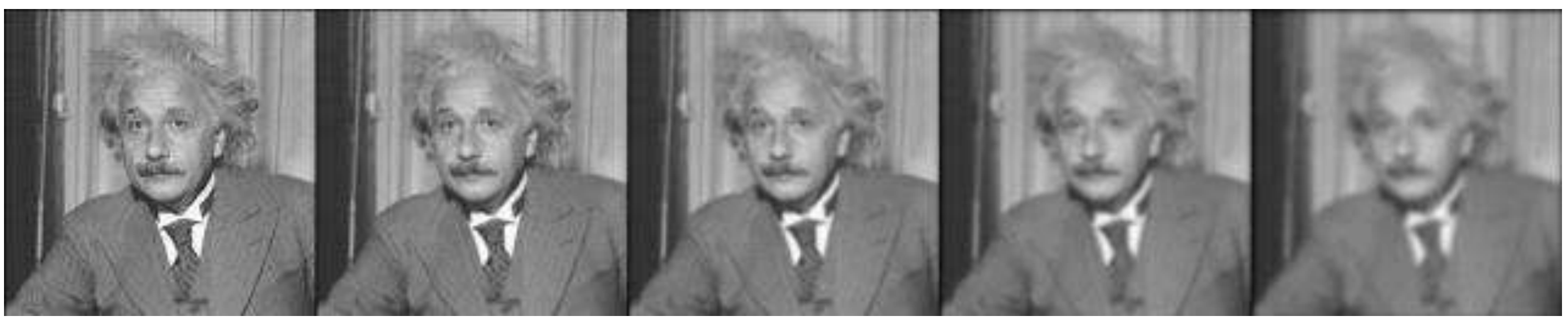
Scale Invariant Point of Interest Detection

- The problem: “corners” are dependent on the scale of the image
- How do we choose corresponding circles ***independently*** in each image?

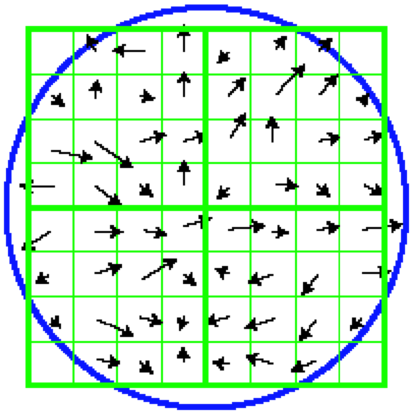


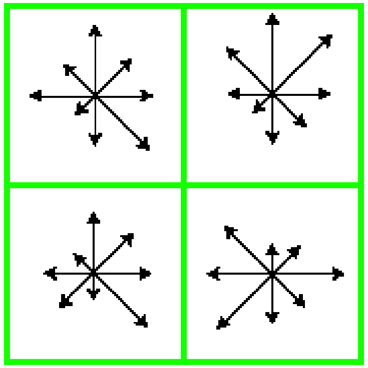
Look for “Corners” in Difference-of-Gaussians

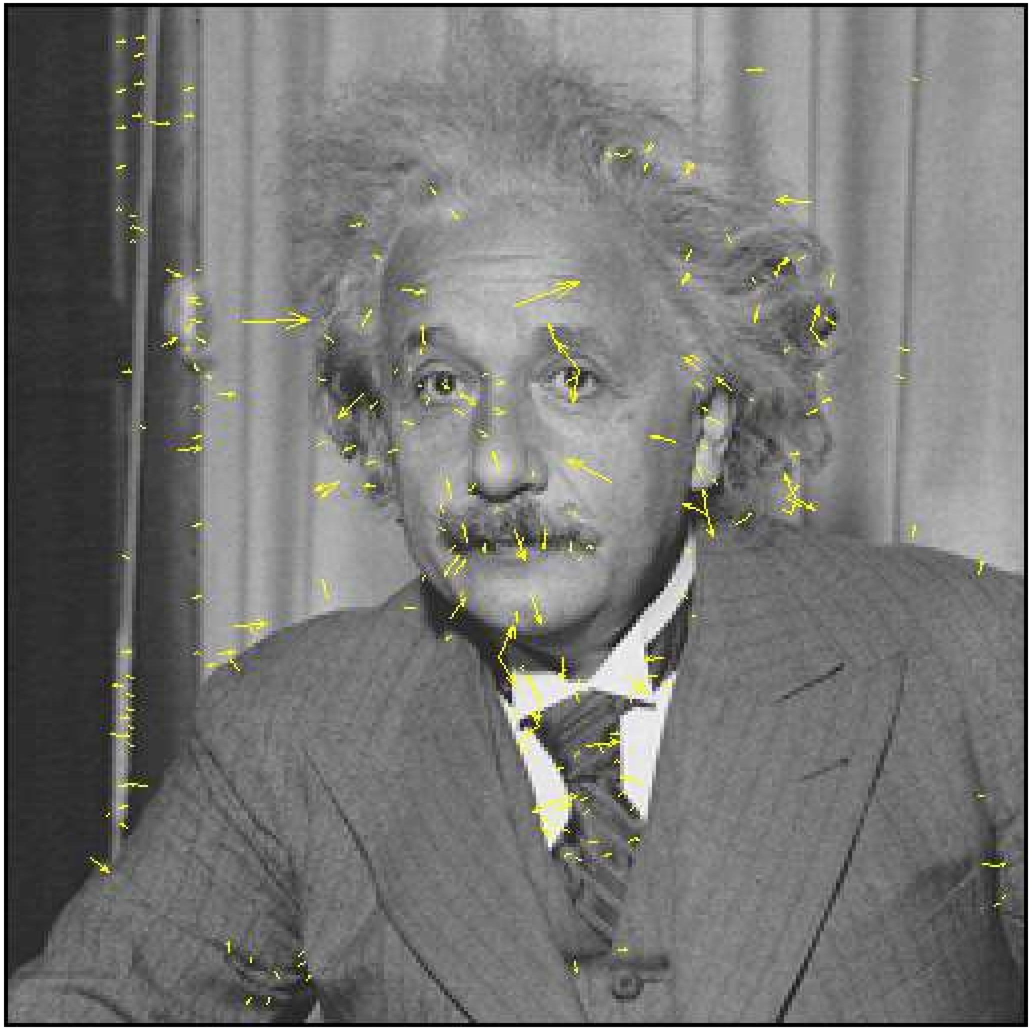




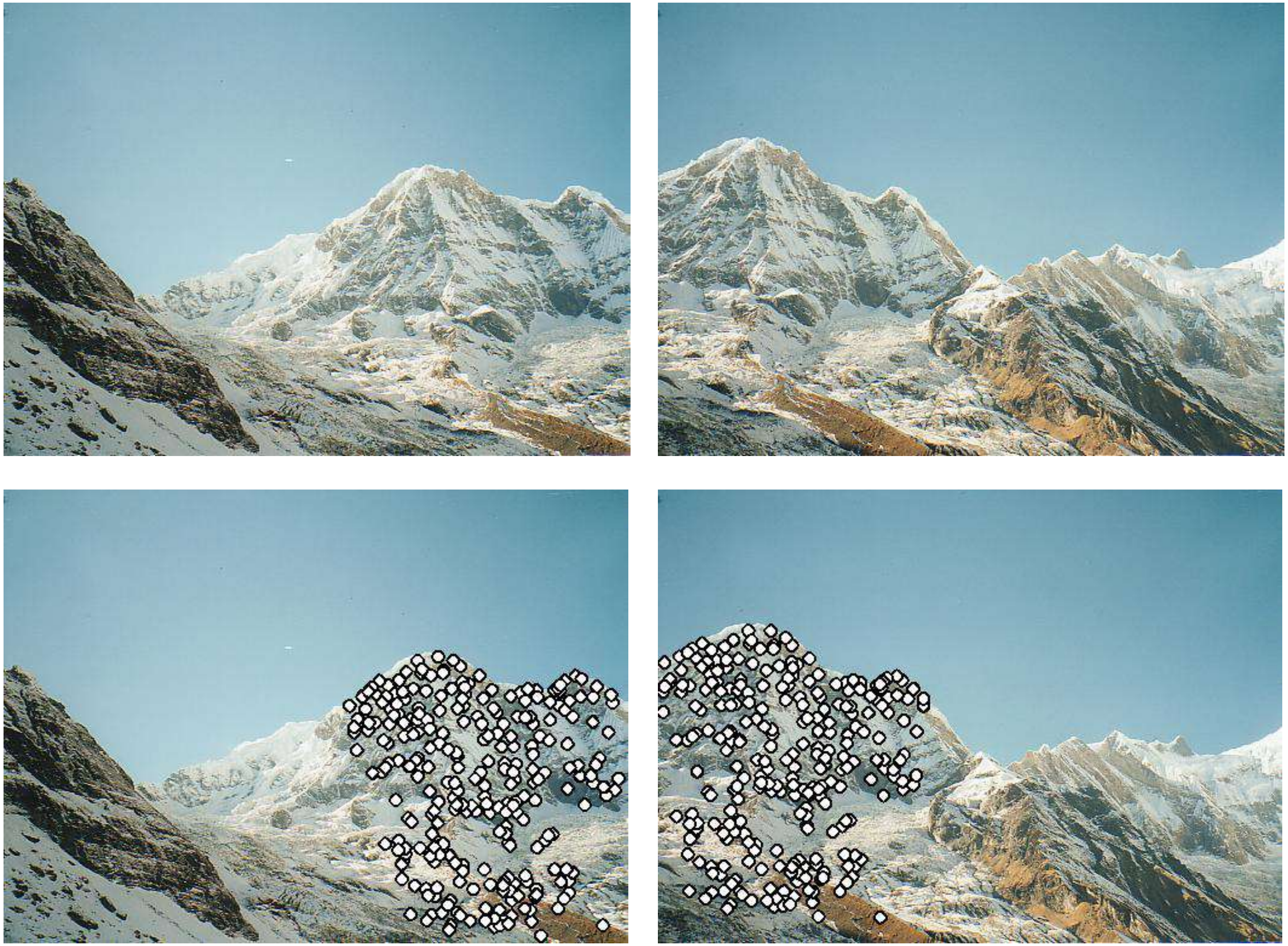
Scale Invariant Feature Transform (SIFT) descriptors



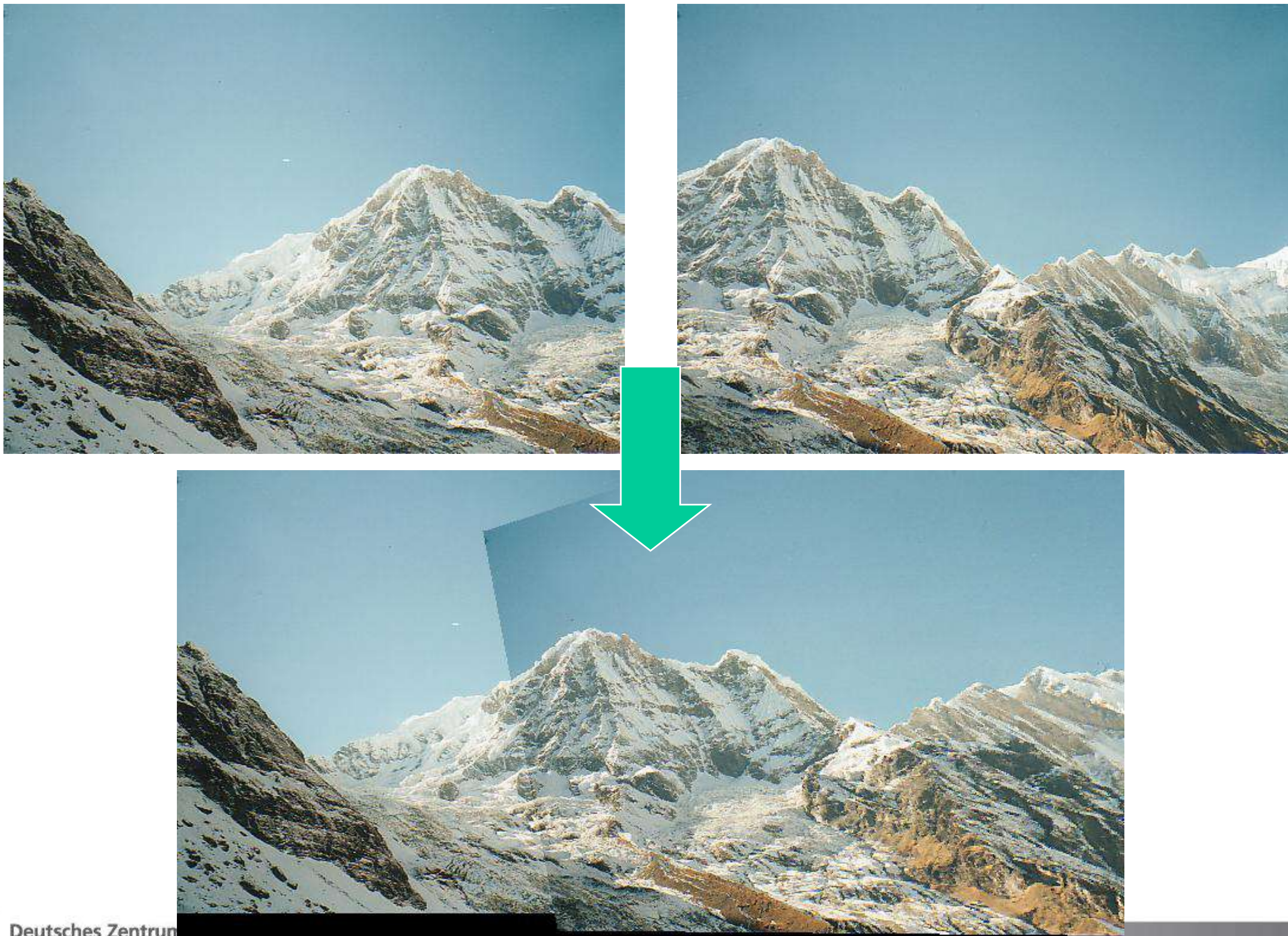




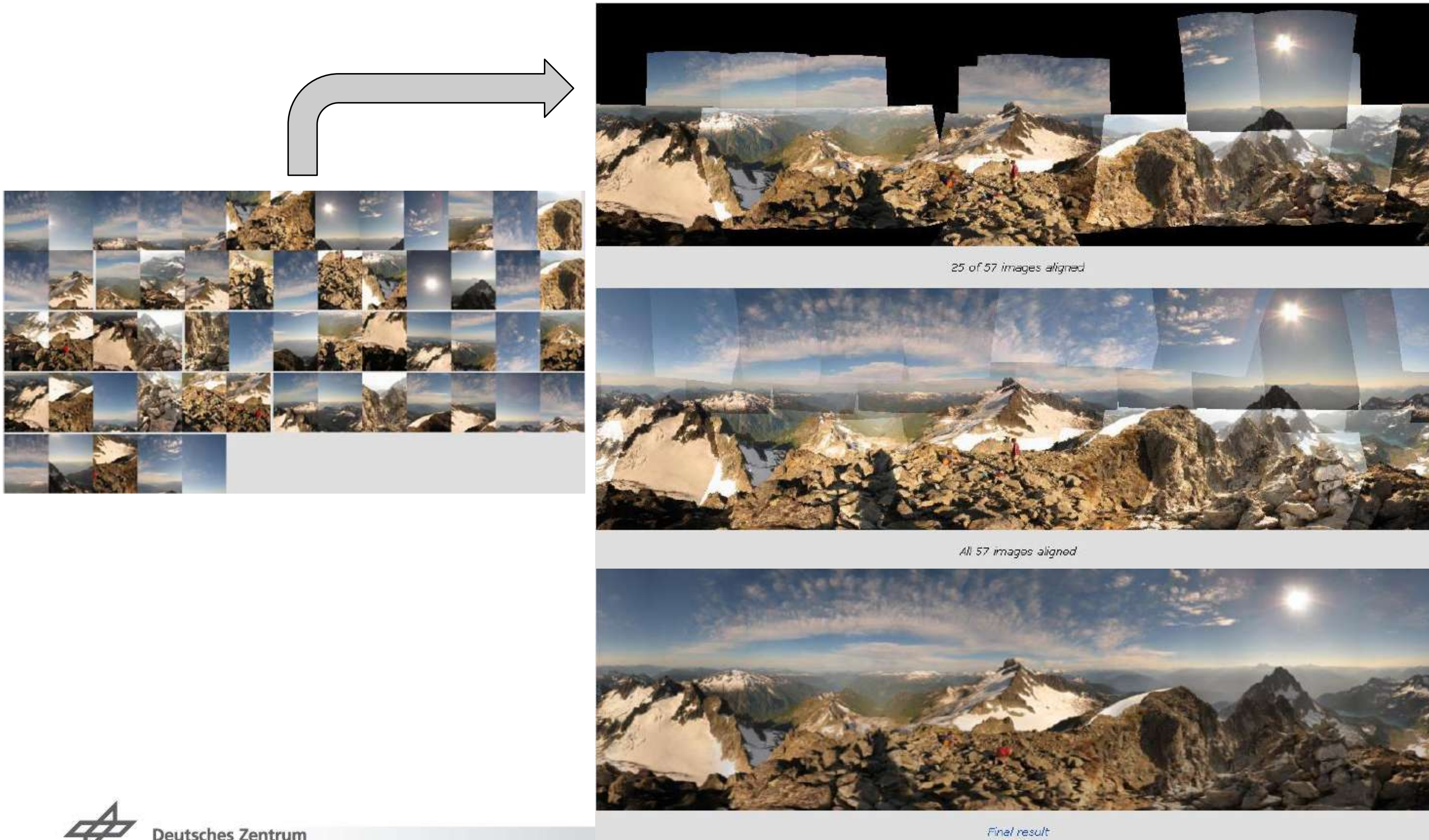
Matching examples



Matching examples



Examples: panoramas



Sony Aibo

- SIFT usage:
  - Recognize charging station
  - Communicate with visual cards
  - Teach object recognition

Photo tourism: exploring photo collections

- Joint work by University of Washington and Microsoft Research
  - http://phototour.cs.washington.edu/
  - http://research.microsoft.com/IVM/PhotoTours/
- Photosynth Technology Preview at Microsoft Live Labs
  - http://labs.live.com/photosynth/

Photo tourism: exploring photo collections

- Detect features using SIFT.

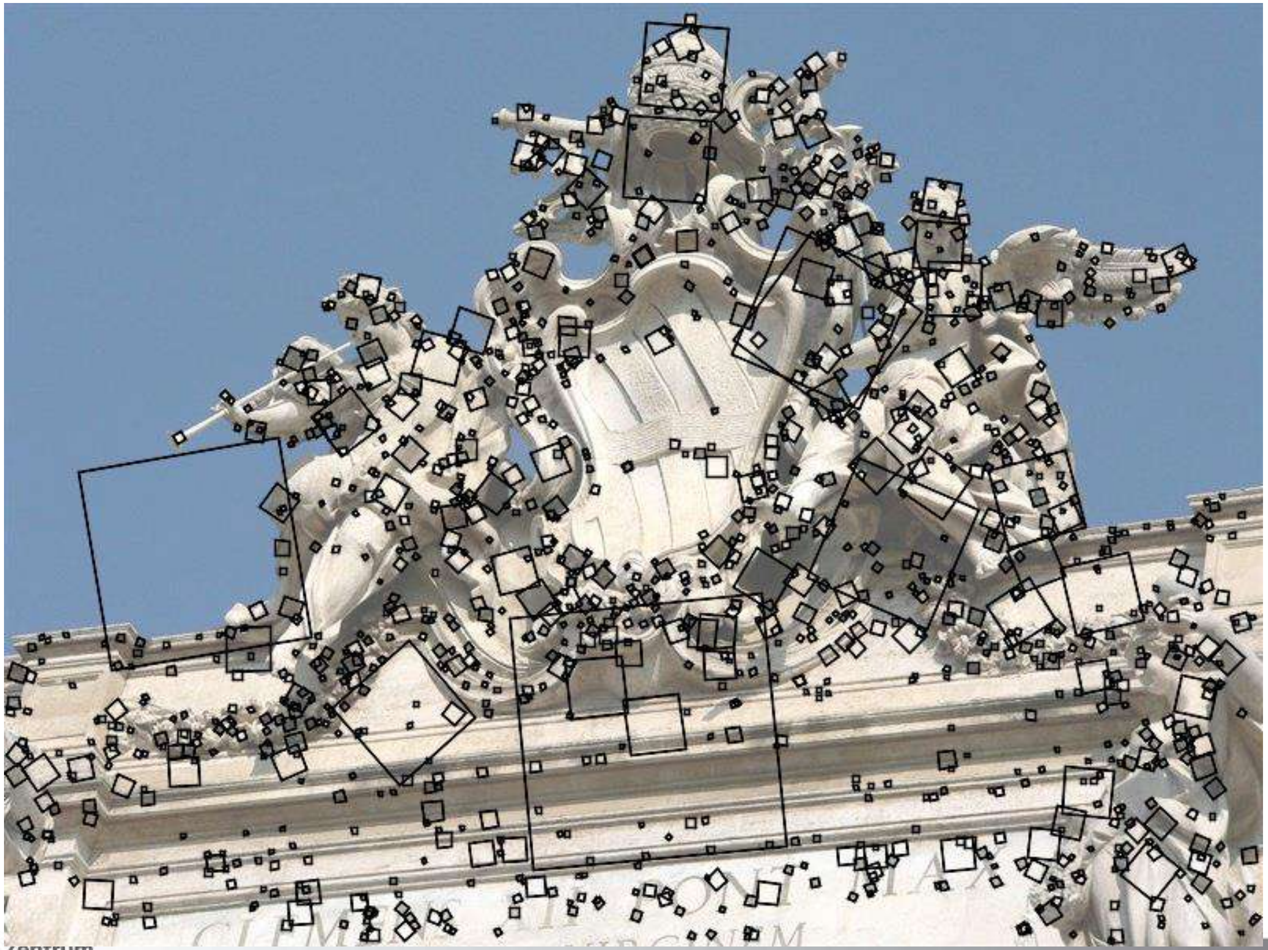


Photo tourism: exploring photo collections

- Detect features using SIFT.

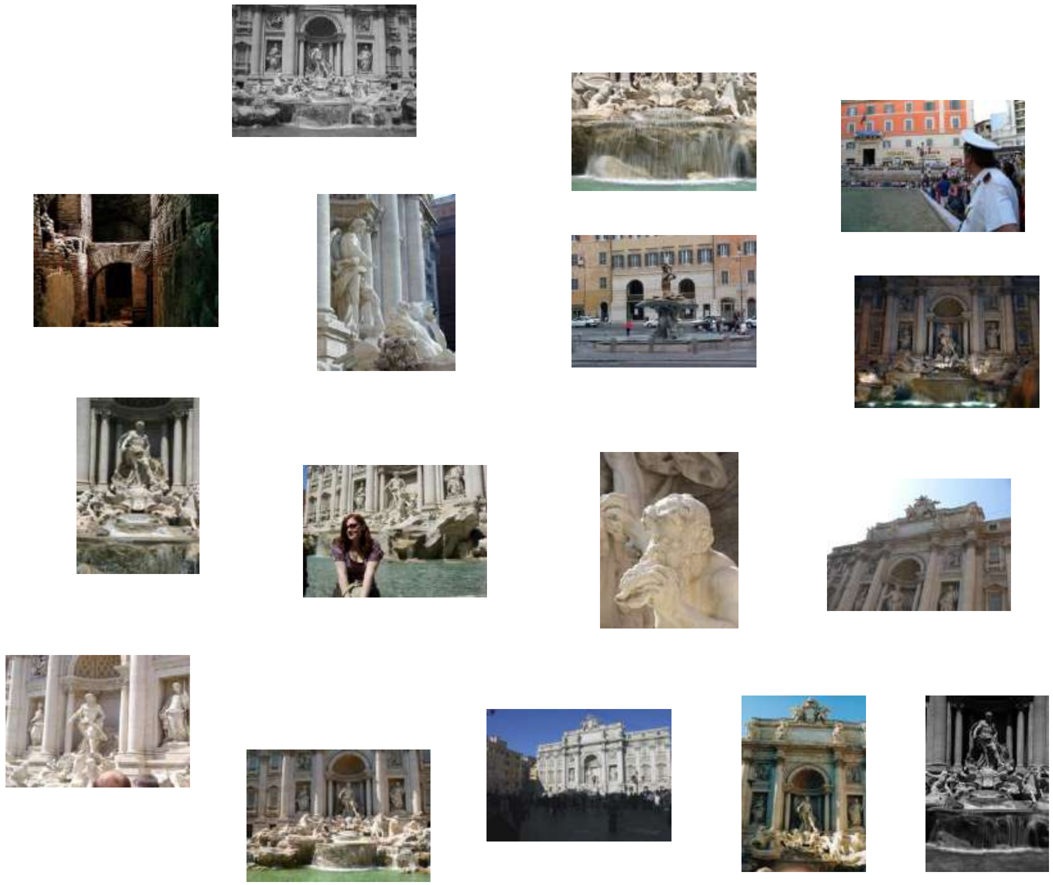


Photo tourism: exploring photo collections

- Detect features using SIFT.

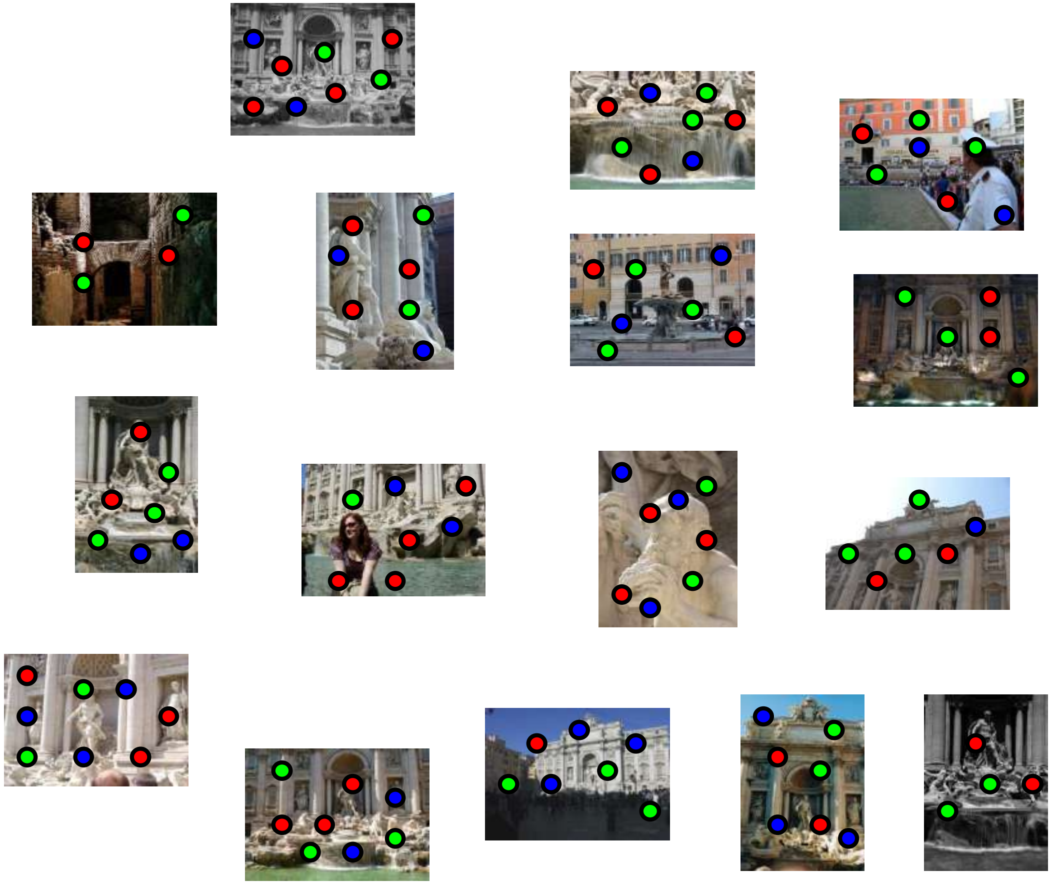


Photo tourism: exploring photo collections

- Match features between each pair of images.

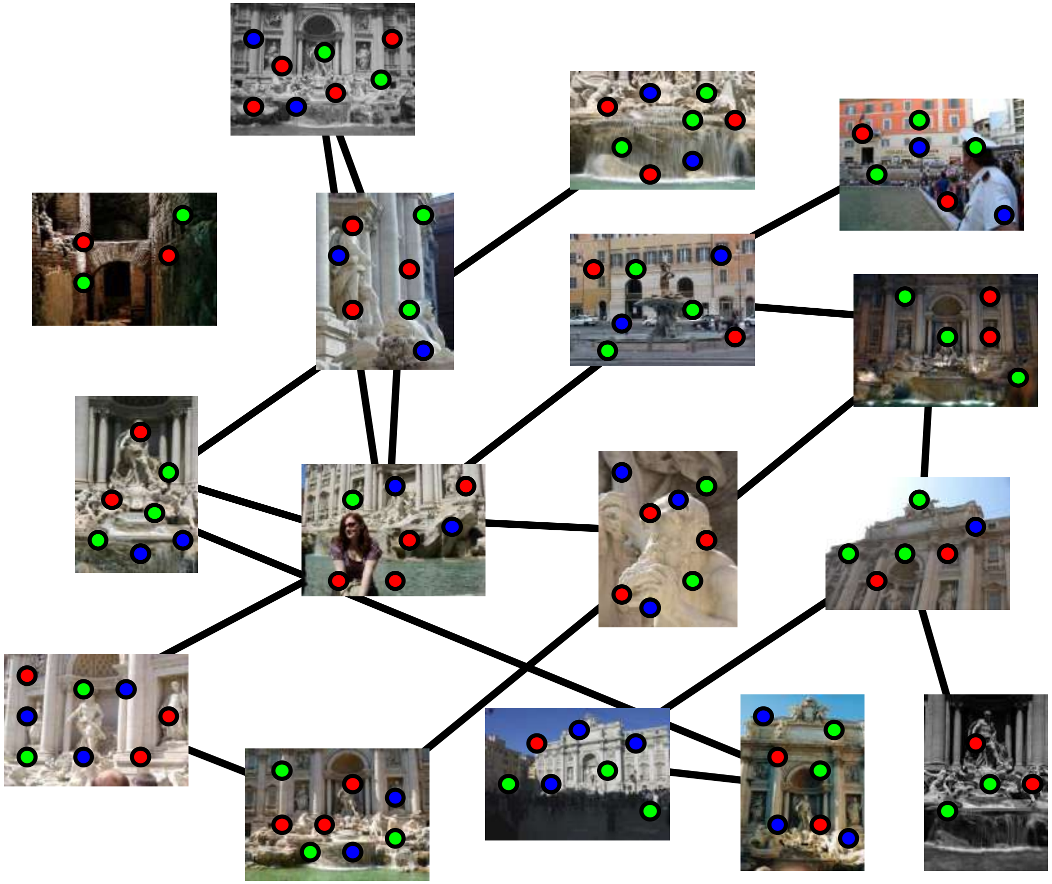


Photo tourism: exploring photo collections

- Link up pairwise matches to form connected components of matches across several images.

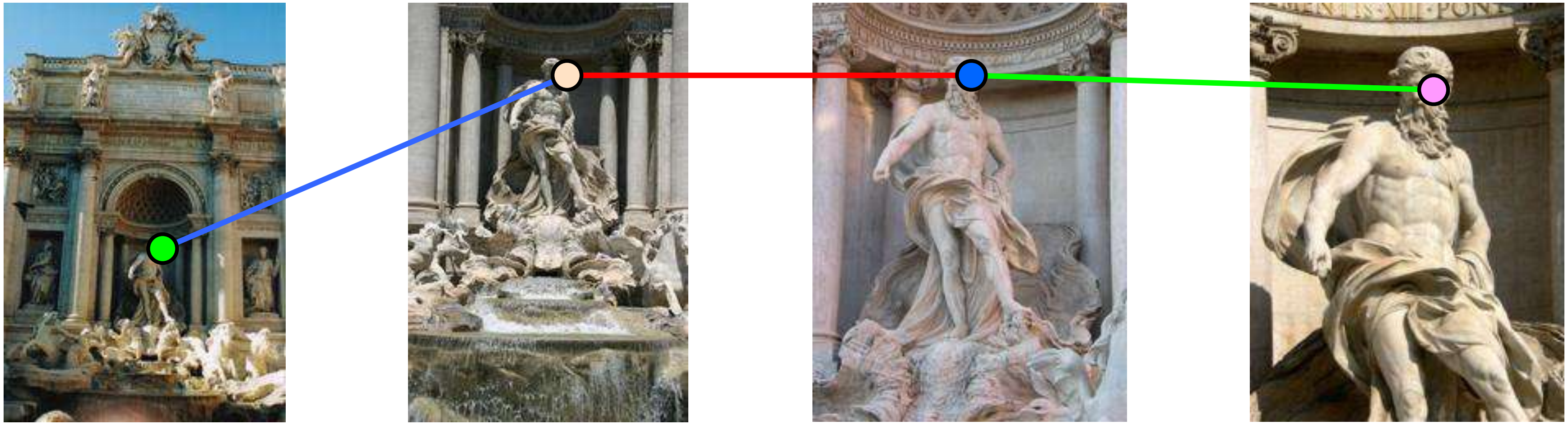


Image 1 Image 2 Image 3 Image 4

Photo tourism: exploring photo collections

Summary

- Image Acquisition
- Image enhancement
- Sampling & Aliasing
- Image Features
- Image Clustering
- Image Classification

What is Clustering?

- Organizing data into classes such that there is
  - high intra-class similarity
  - low inter-class similarity
- Finding the class labels and the number of classes directly from the data (in contrast to classification).
- More informally, finding natural groupings among objects.
- K-means: a very popular clustering algorithm

Algorithm *k-means*

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (usually randomly).

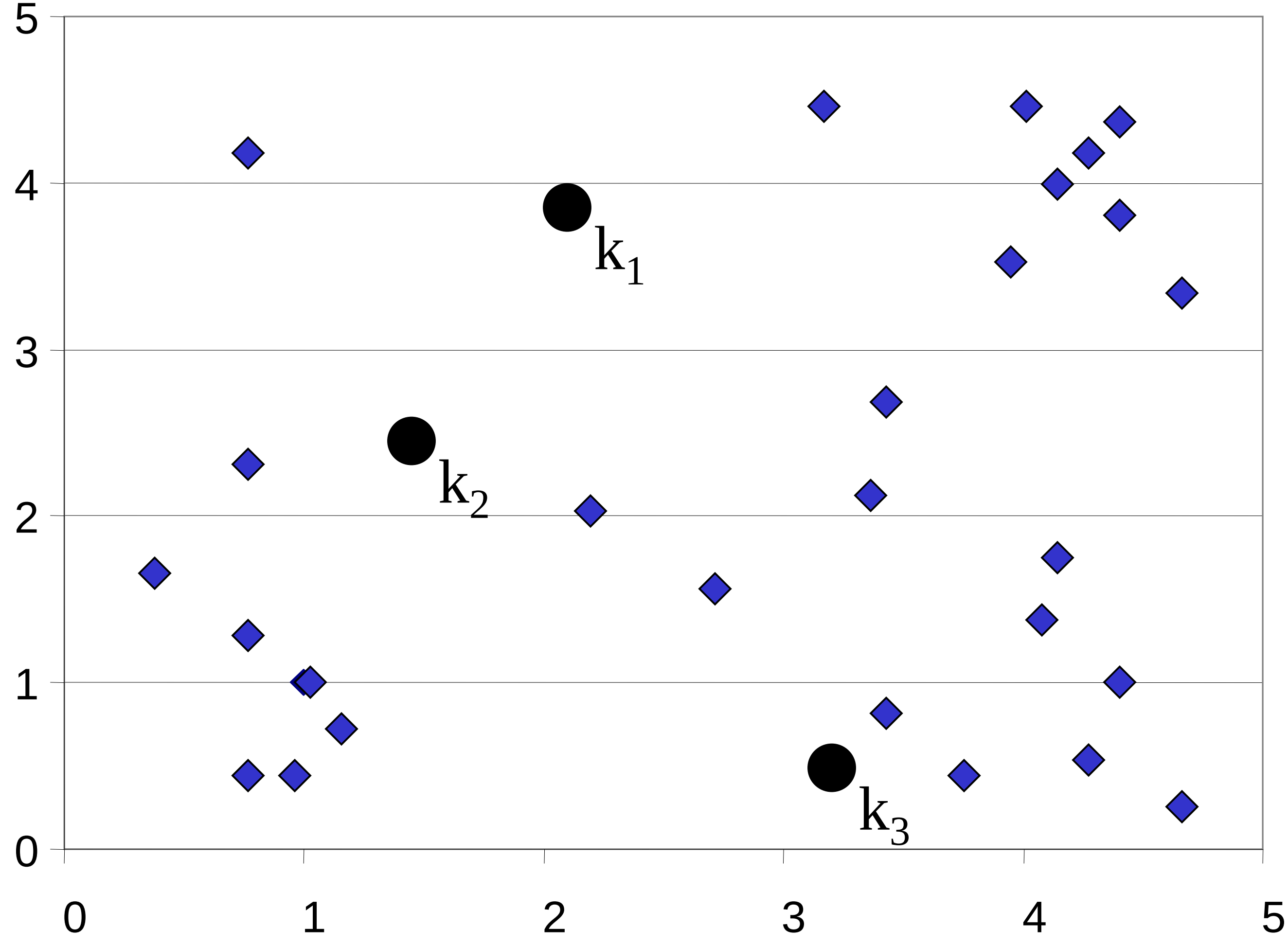
3. Decide the class memberships of the objects by assigning them to the nearest cluster center.

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

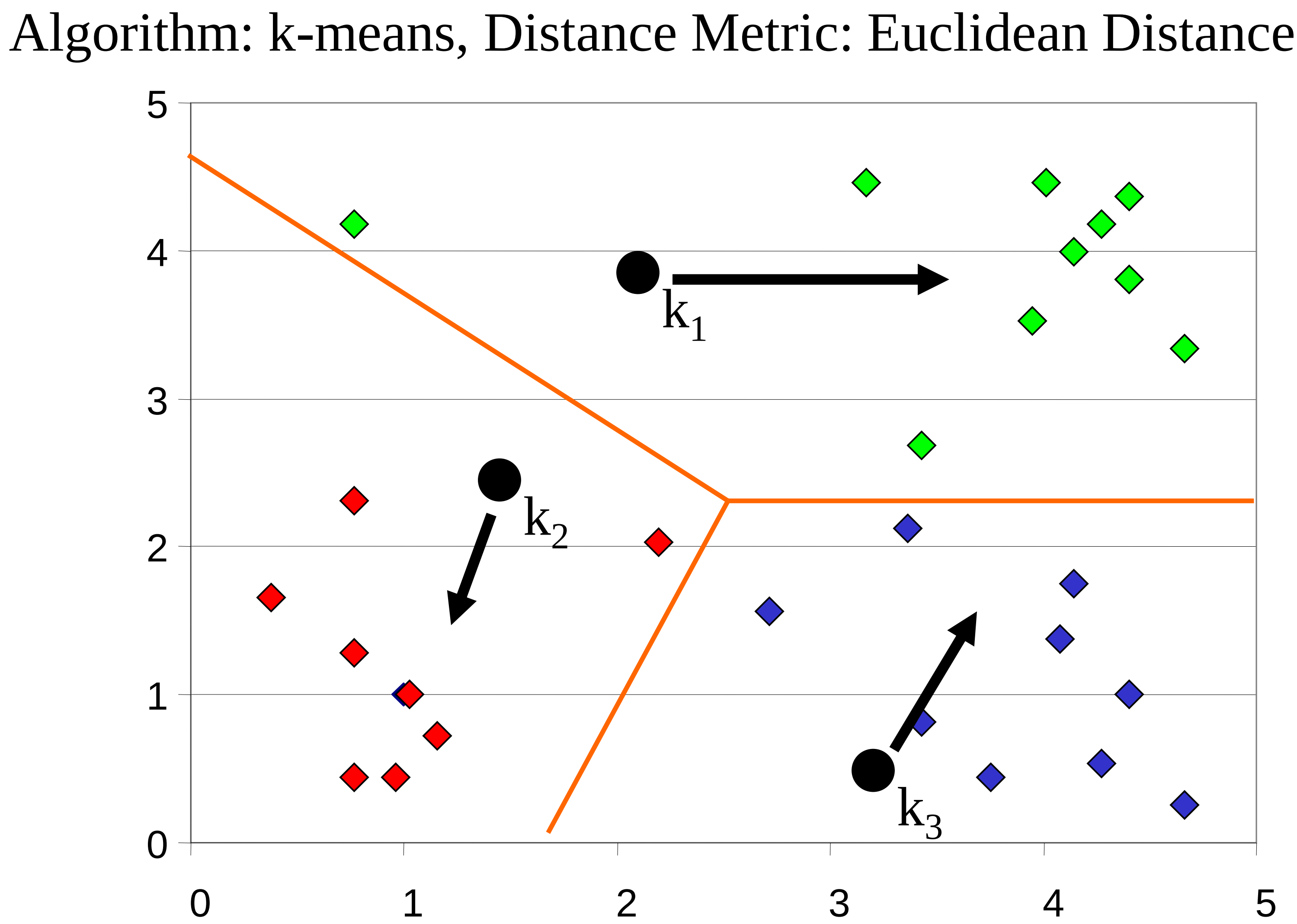
5. If none of the objects changed membership in the last iteration, exit. Otherwise goto 3.

K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance

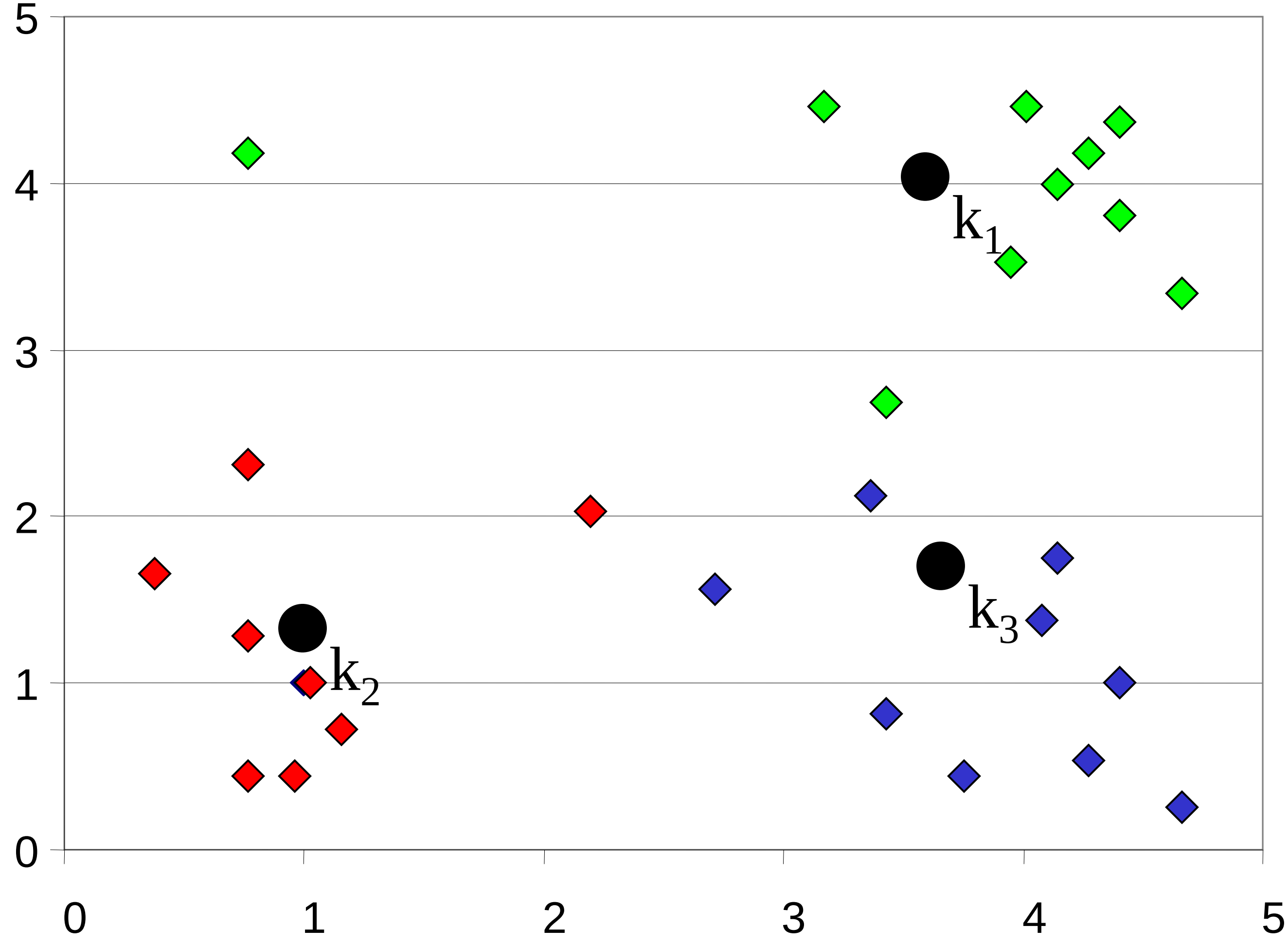


K-means Clustering: Step 2

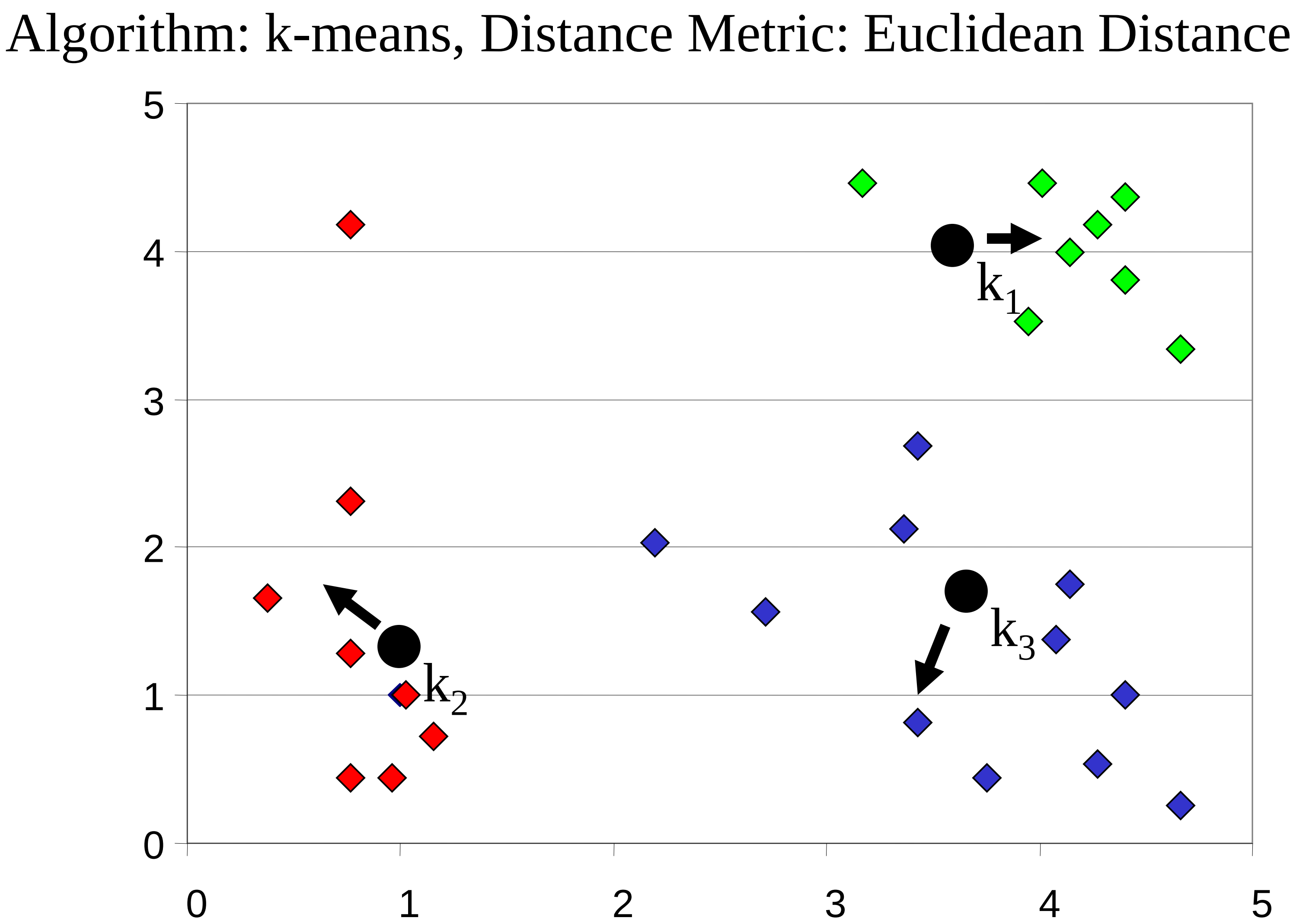


K-means Clustering: Step 3

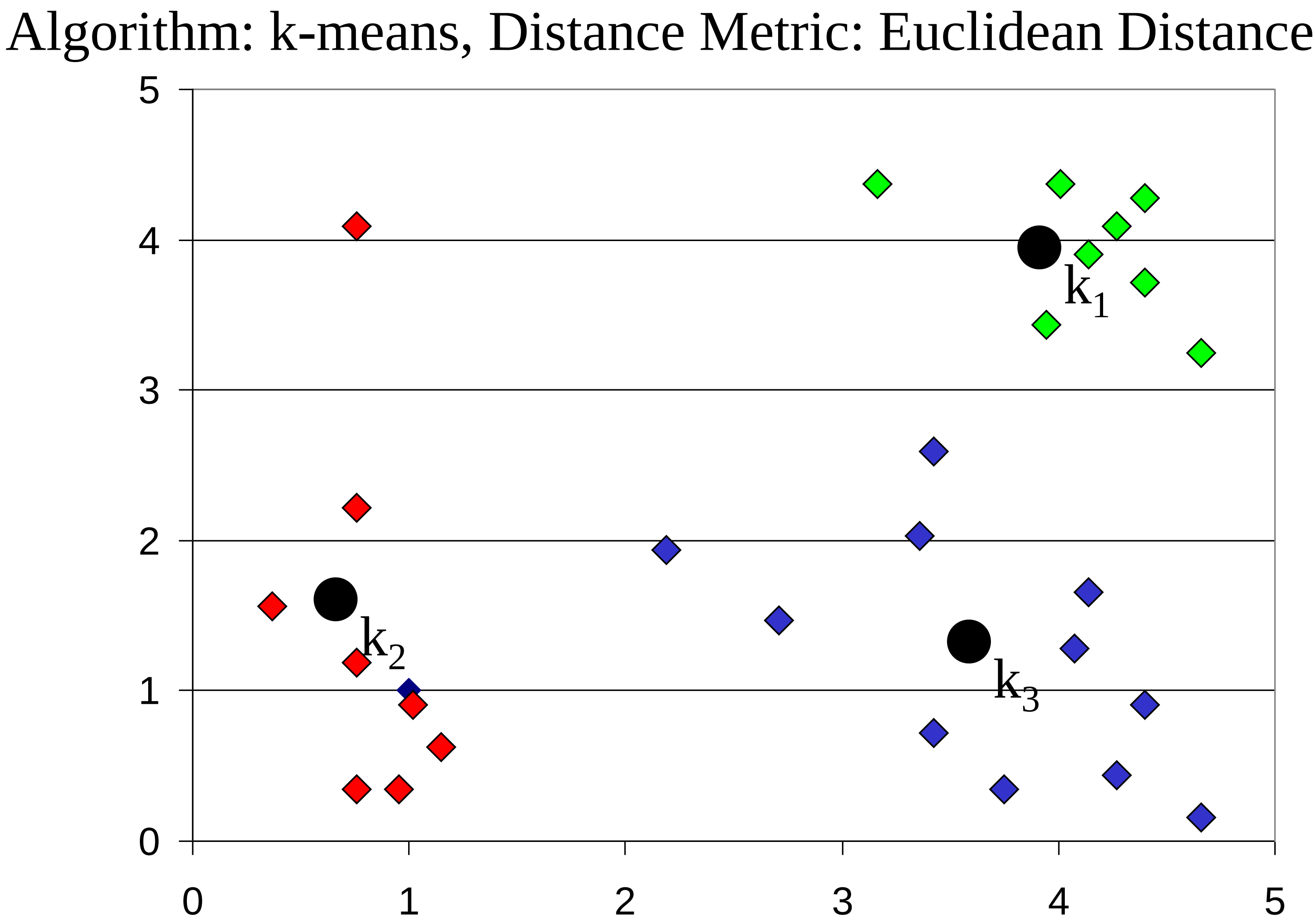
Algorithm: k-means, Distance Metric: Euclidean Distance



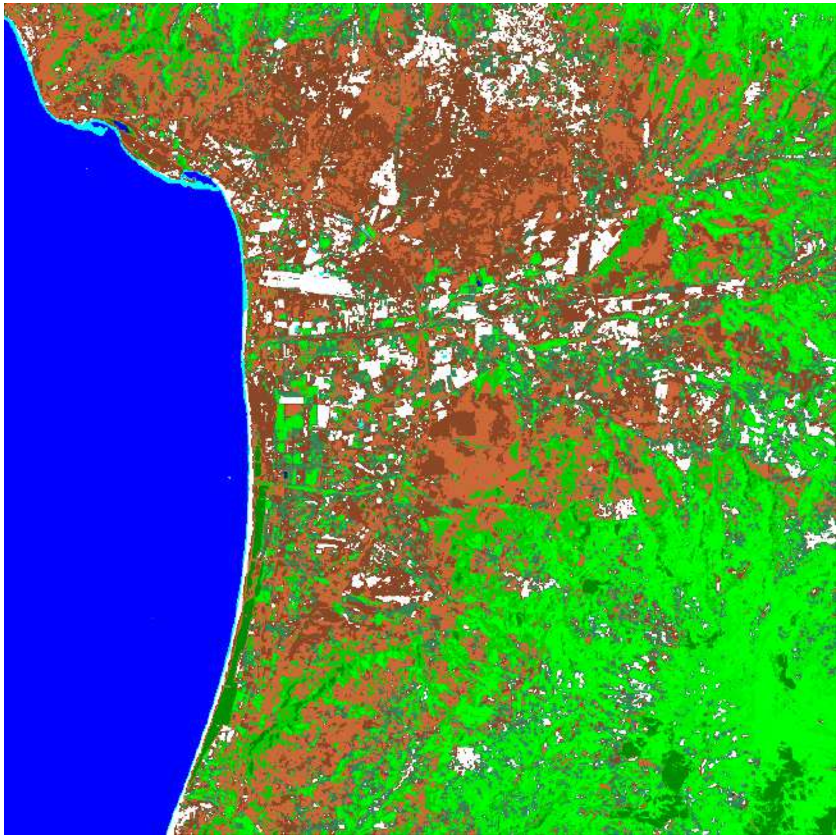
K-means Clustering: Step 4



K-means Clustering: Step 5



K-means Clustering (k = 7)



A (very short) Introduction to Classification & Clustering

DLR Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

What is Clustering?

- Organizing data into classes such that there is
  - high intra-class similarity
  - low inter-class similarity
- Finding the class labels and the number of classes directly from the data (in contrast to classification).
- More informally, finding natural groupings among objects.
- K-means: a very popular clustering algorithm

Algorithm *k-means*

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (usually randomly).

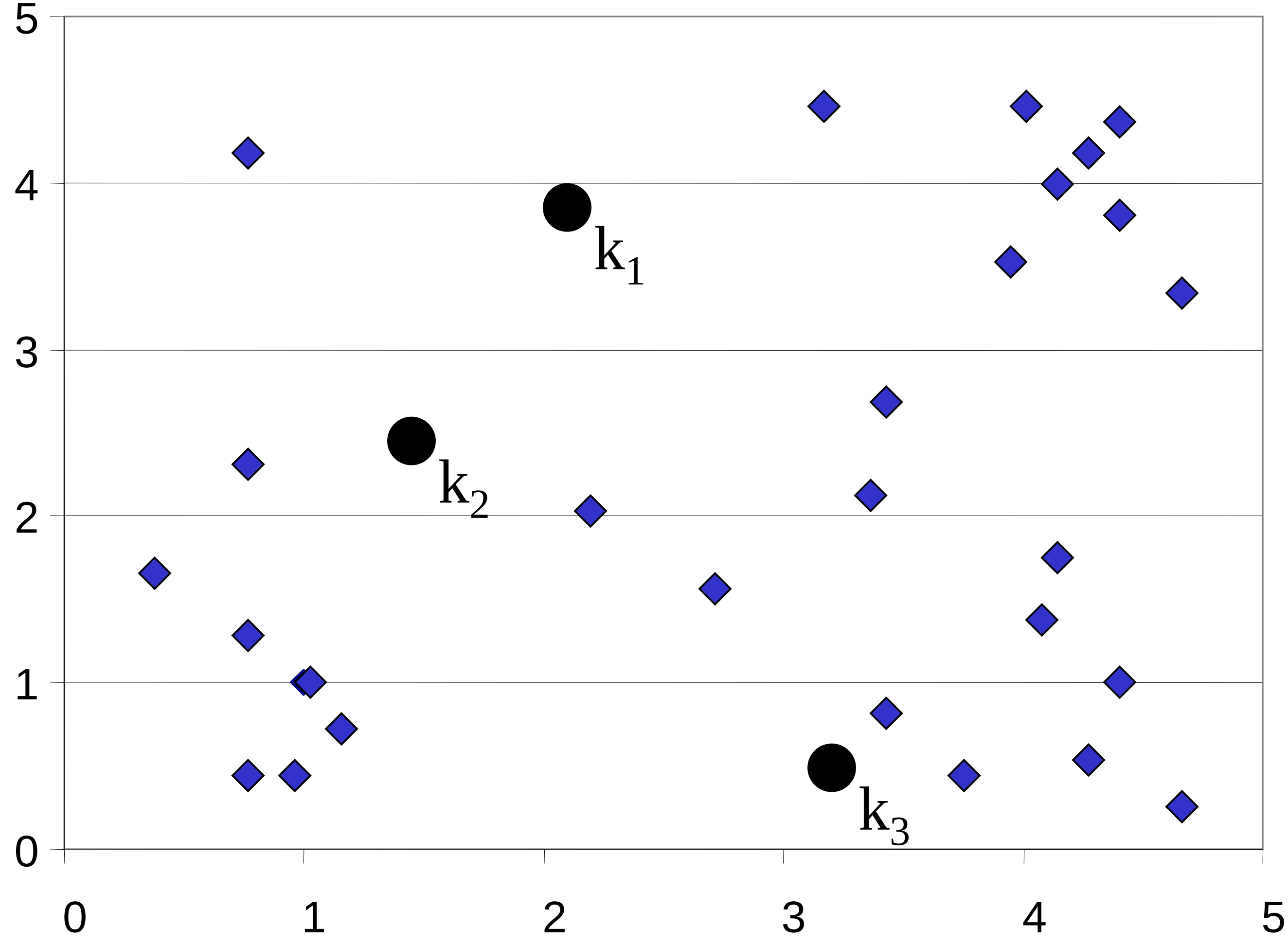
3. Decide the class memberships of the objects by assigning them to the nearest cluster center.

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

5. If none of the objects changed membership in the last iteration, exit. Otherwise goto 3.

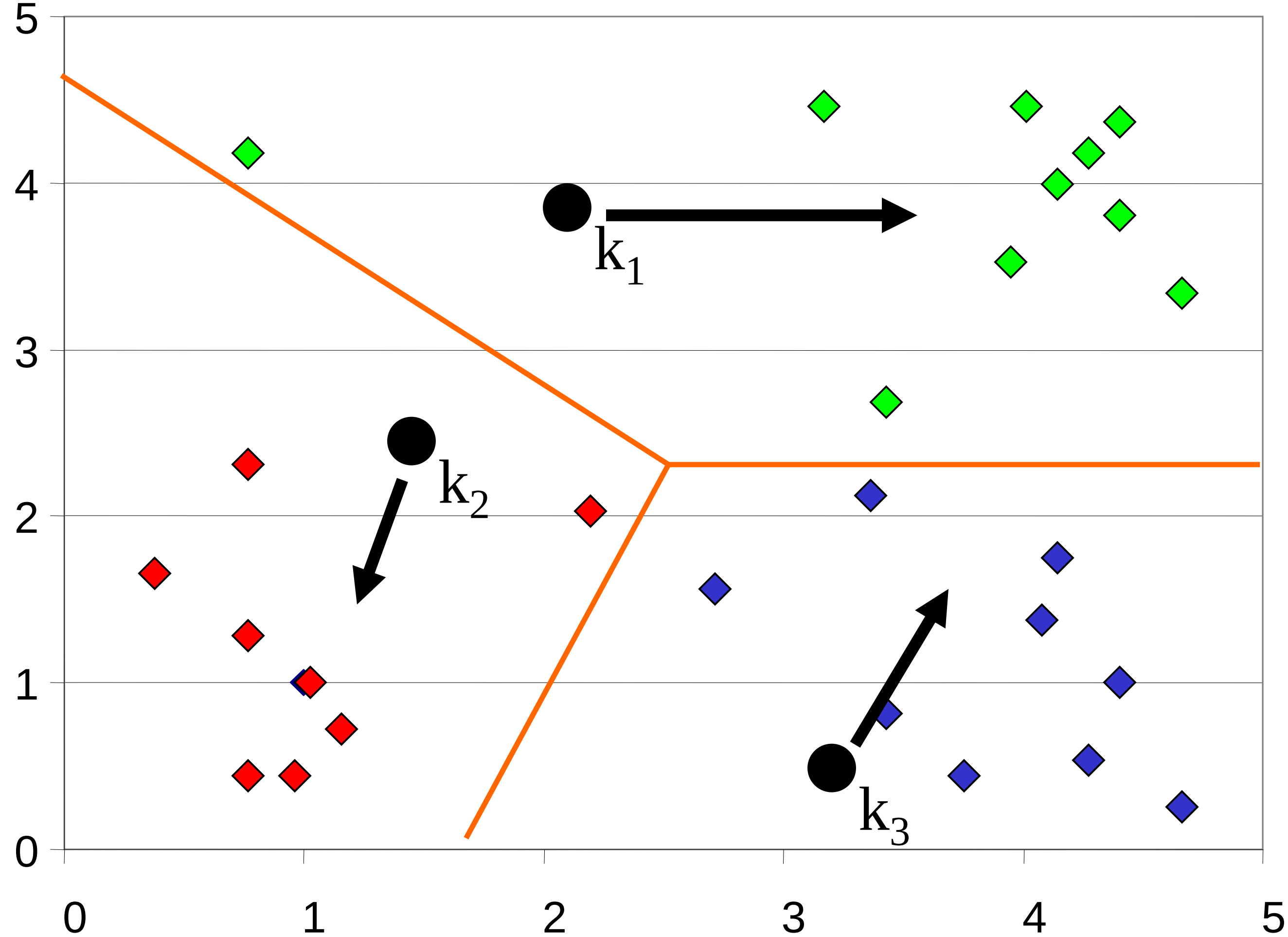
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



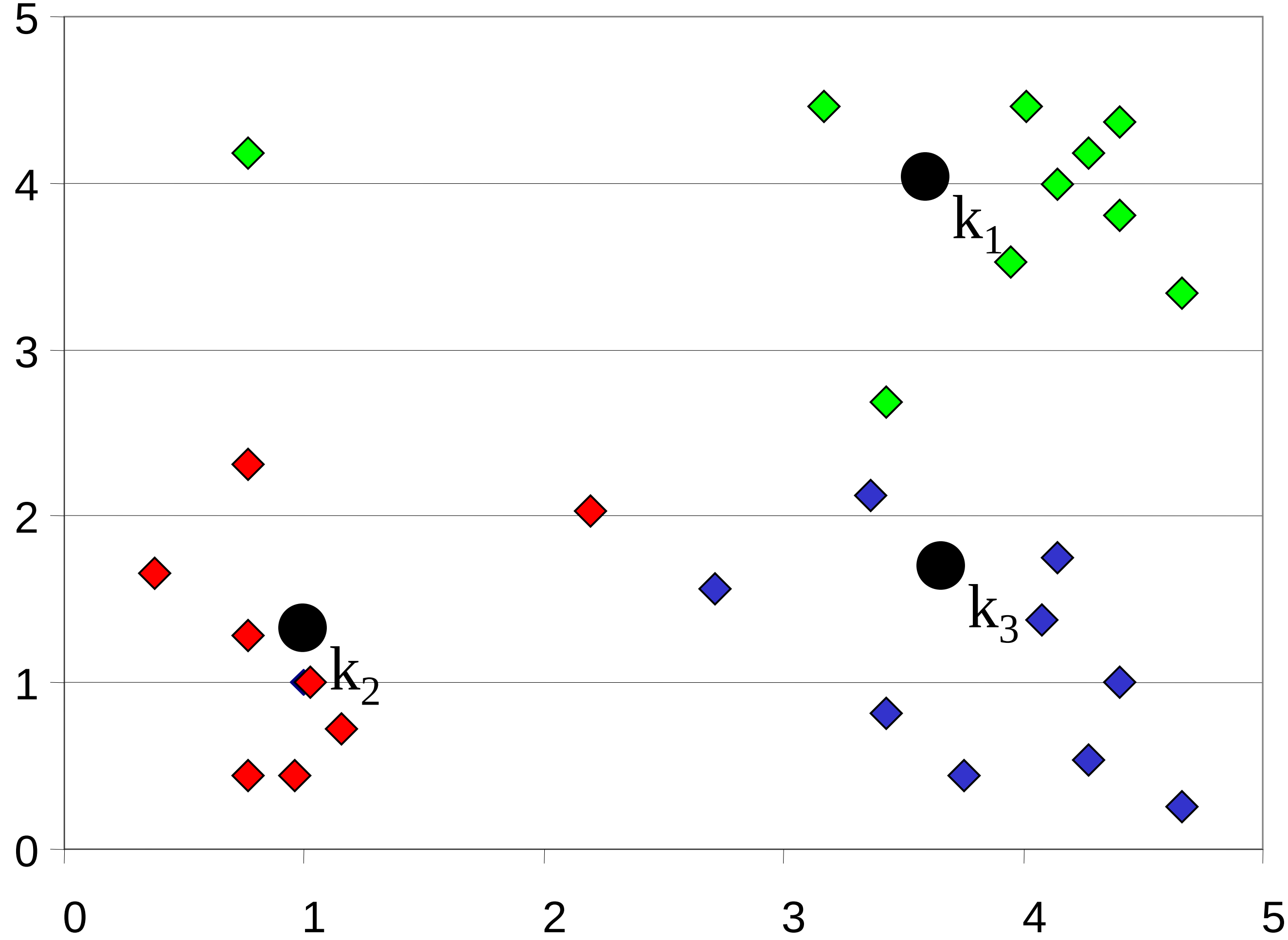
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance

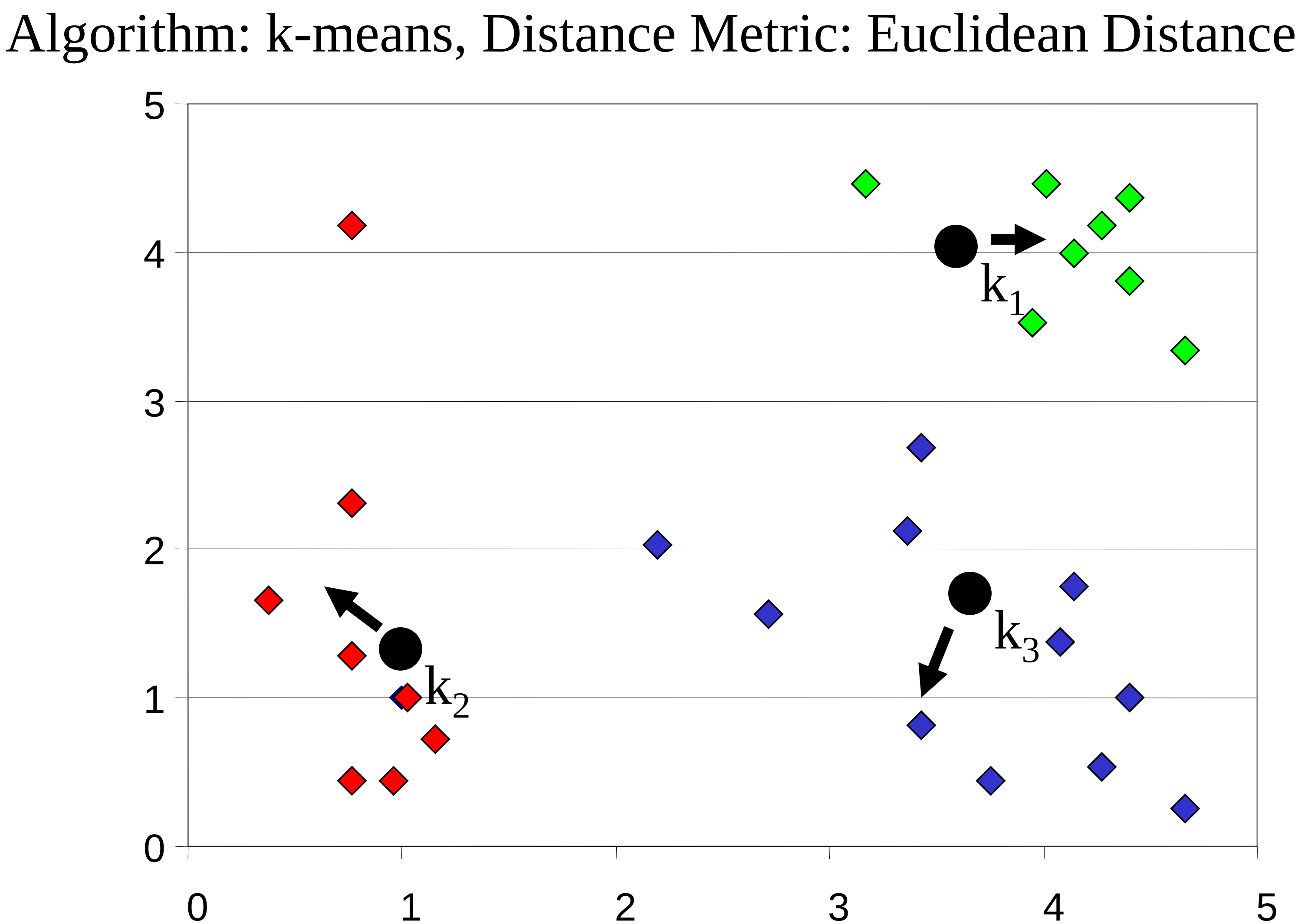


K-means Clustering: Step 3

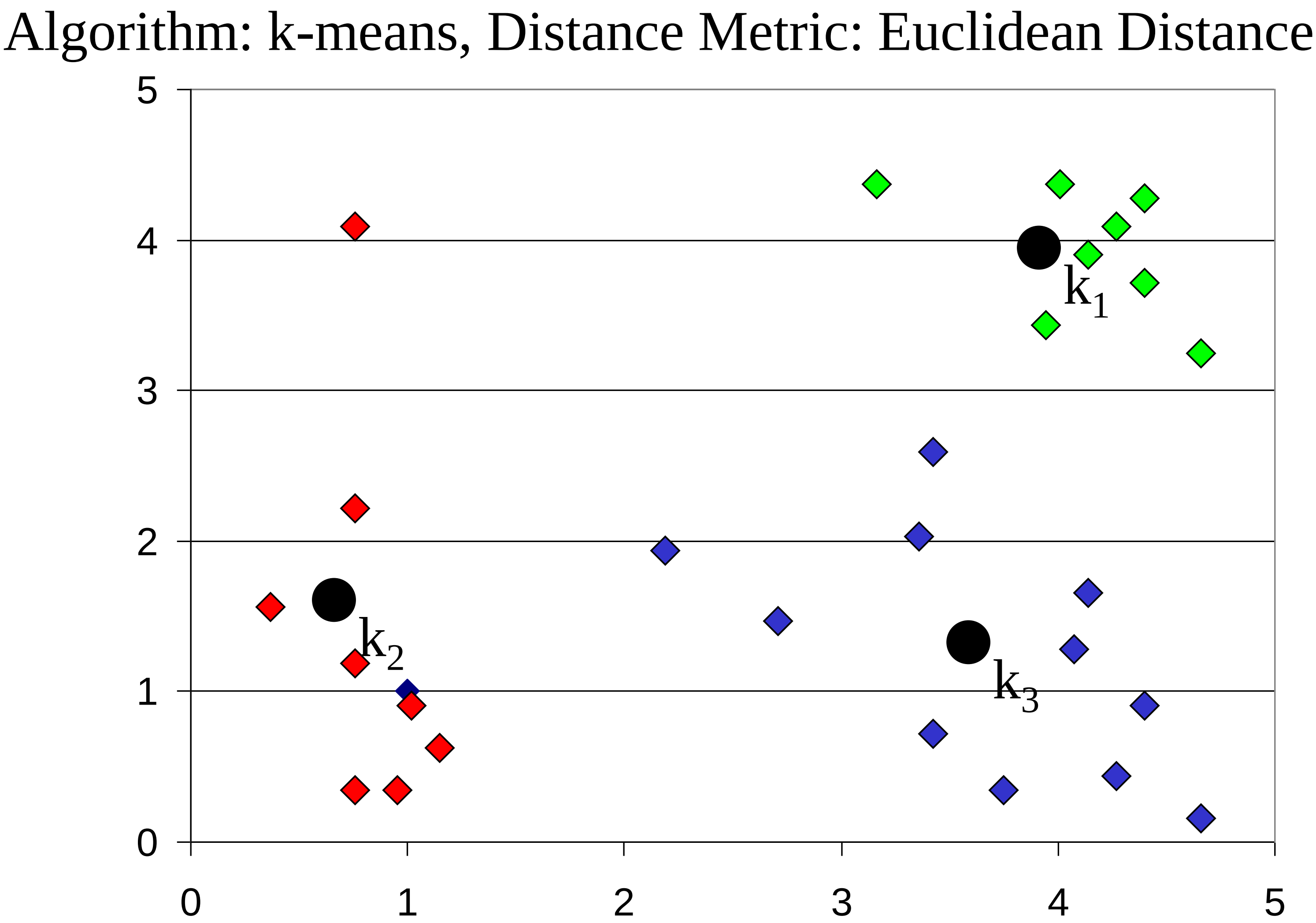
Algorithm: k-means, Distance Metric: Euclidean Distance



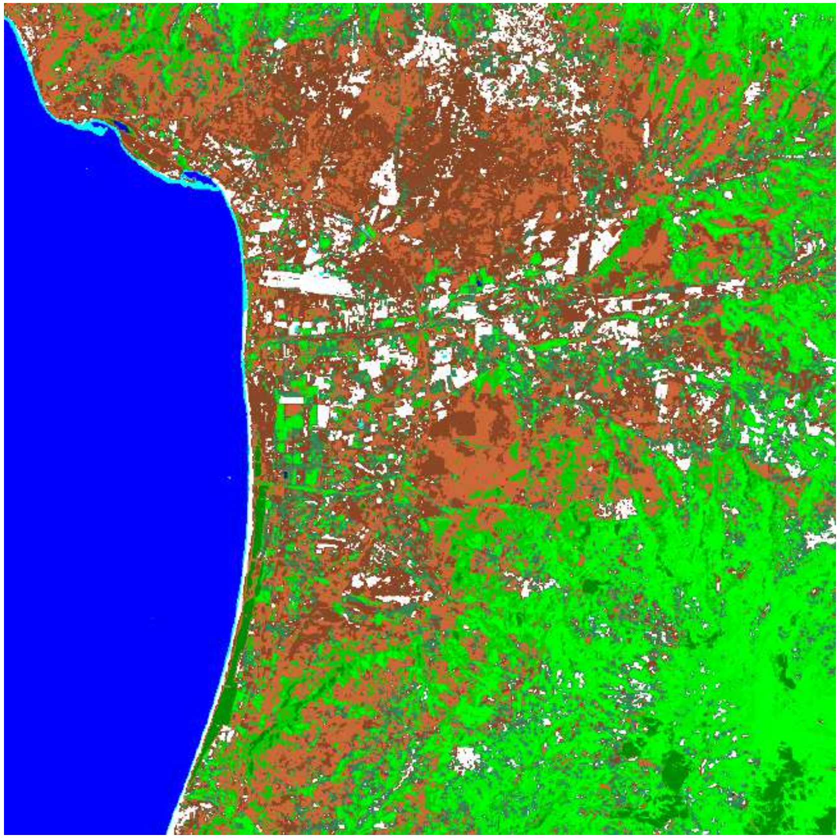
K-means Clustering: Step 4



K-means Clustering: Step 5



K-means Clustering (k = 7)



From Clustering to Image Segmentation

- If we divide an image into homogeneous clusters we obtain a segmentation of the image

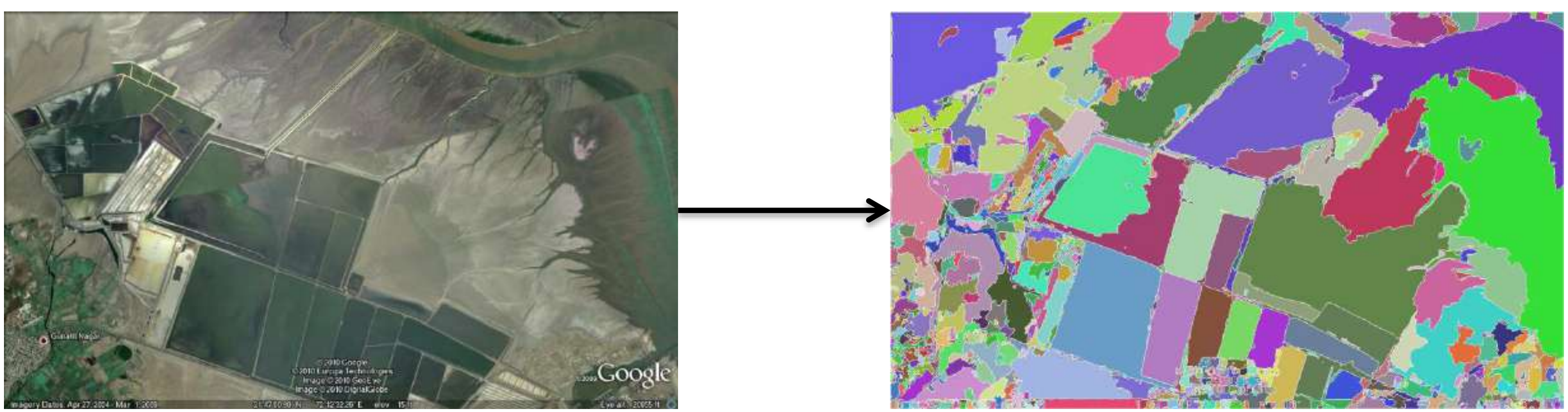


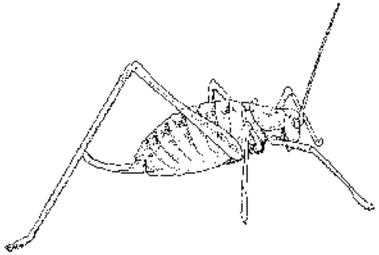
Image Segmentation

- Partitioning of an image in homogeneous regions (segments)
  - According to inter-pixel similarity
- Pre-processing step for machine recognition
  - Simplification of the image in something simpler to analyze
  - Decisions may then be taken on each segment rather than on each pixel
- Problems
  - Which characteristics make two pixels similar?
  - According to which criteria should we join the pixels into segments?

Results of image segmentation with varying number of segments

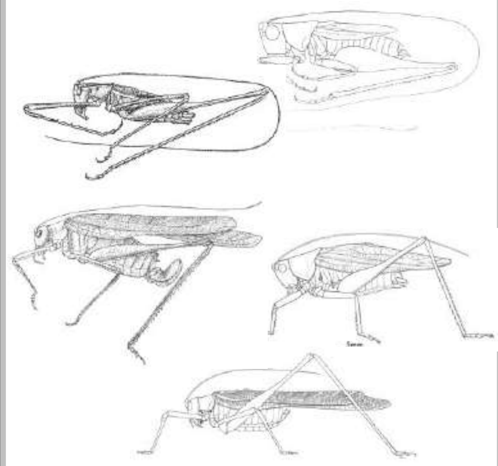
The Classification Problem
(informal definition)

Given a collection of annotated data. In this case 5 instances **Katydids** of and five of **Grasshoppers**, decide what type of insect the unlabeled example is.

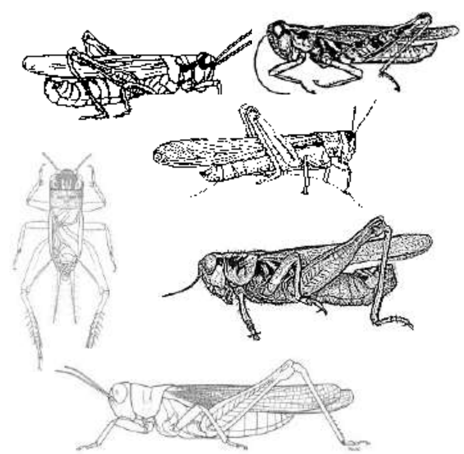


Katydid or **Grasshopper**?

Katydids



Grasshoppers

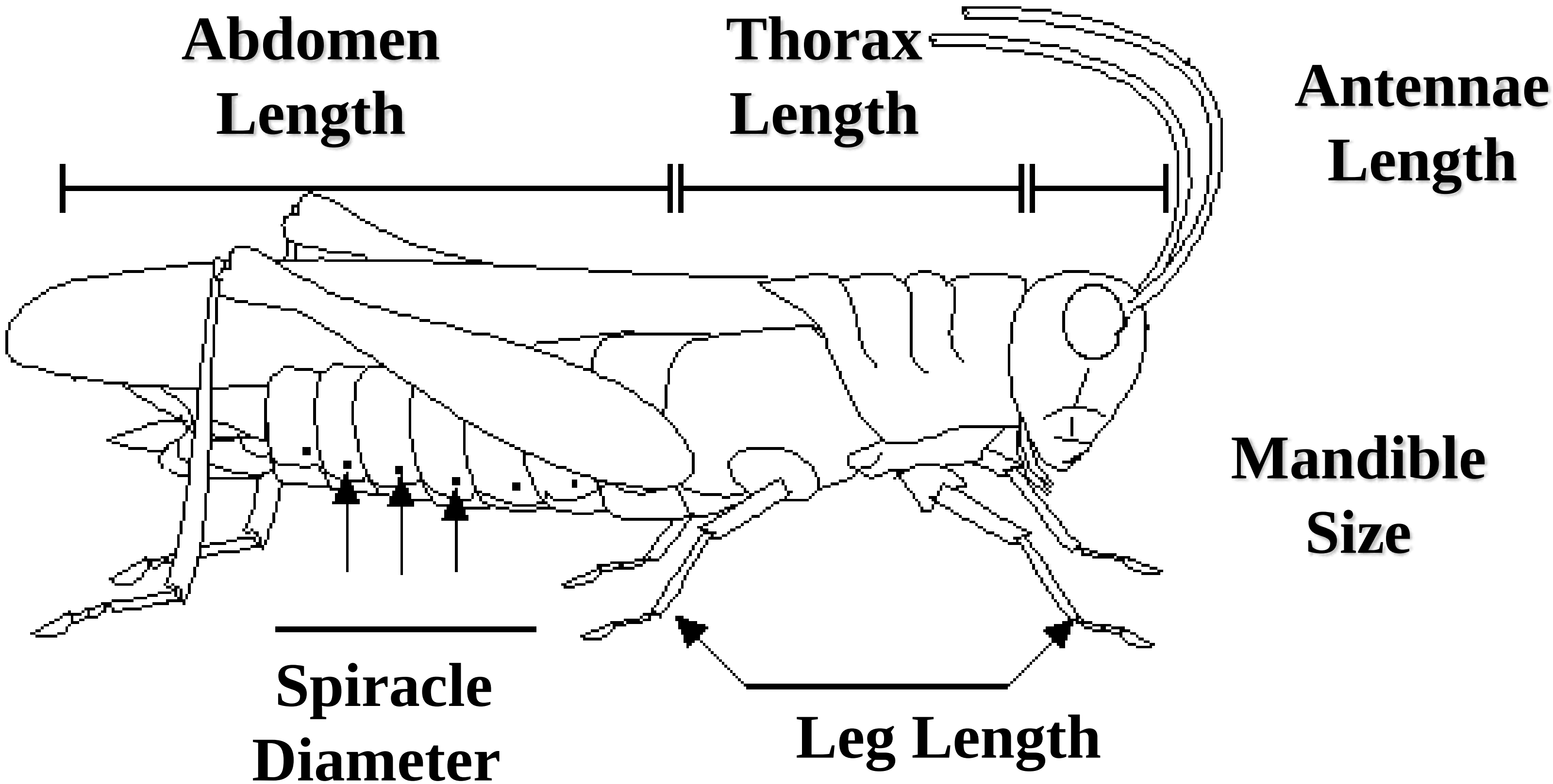


Keogh

For any domain of interest, we can measure *features*

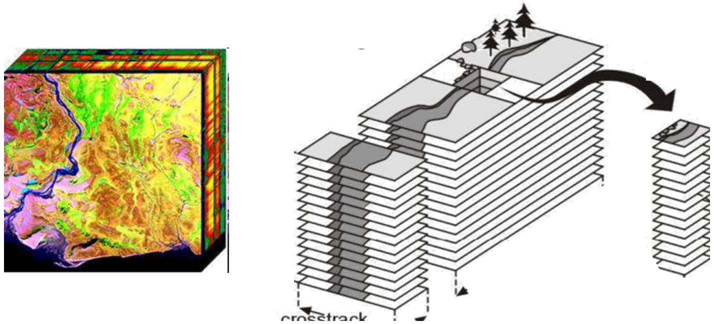
Color {Green, Brown, Gray, Other}

Has Wings?

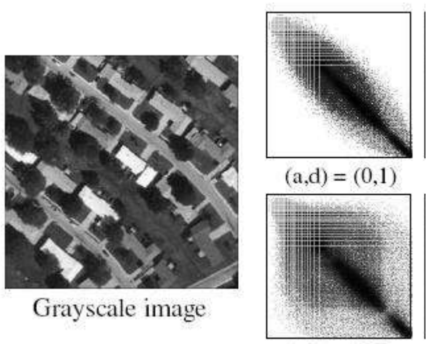


In the case of images, which features can we use?

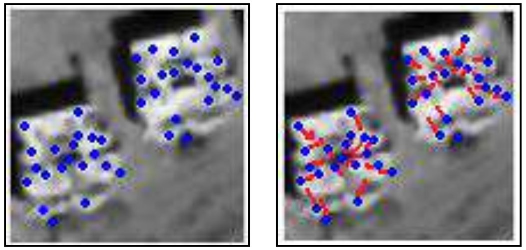
Pixel Value for each band
{RGB, NIR, SWIR, TIR, …}



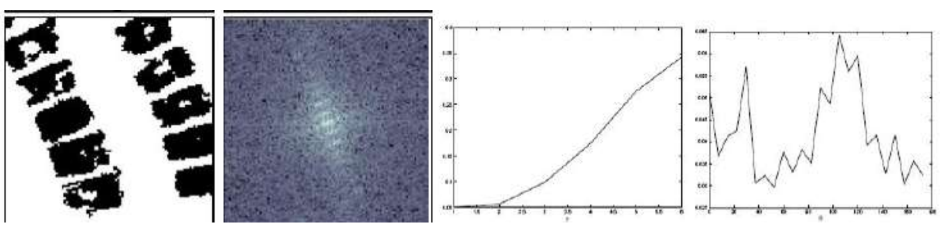
Texture Parameters



SIFT



Power Spectrum Features



…and more!

We can store features in a database.

The classification problem for images can now be expressed as:

• Given a training database (**My_Collection**), predict the **class** label of a previously unseen pixel

Pixel ID	Band 1	Band 2	Pixel Class
1	27	55	**Water**
2	80	91	**Vegetation**
3	9	47	**Water**
4	11	31	**Water**
5	54	85	**Vegetation**
6	29	19	**Water**
7	61	66	**Vegetation**
8	5	10	**Water**
9	83	66	**Vegetation**
10	81	47	**Vegetation**

My_Collection

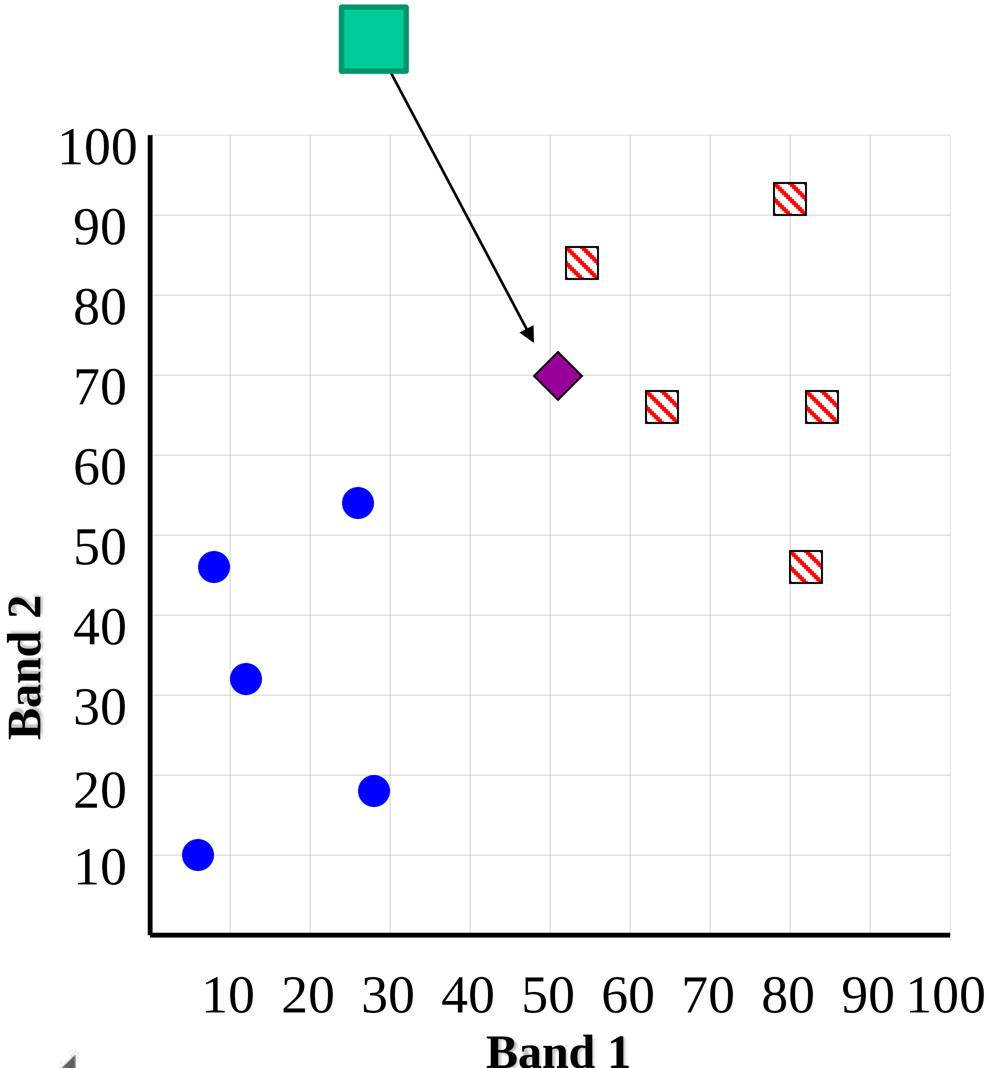
previously unseen pixel =

11	51	70	???????

Adapted from E. Keogh

previously unseen pixel =

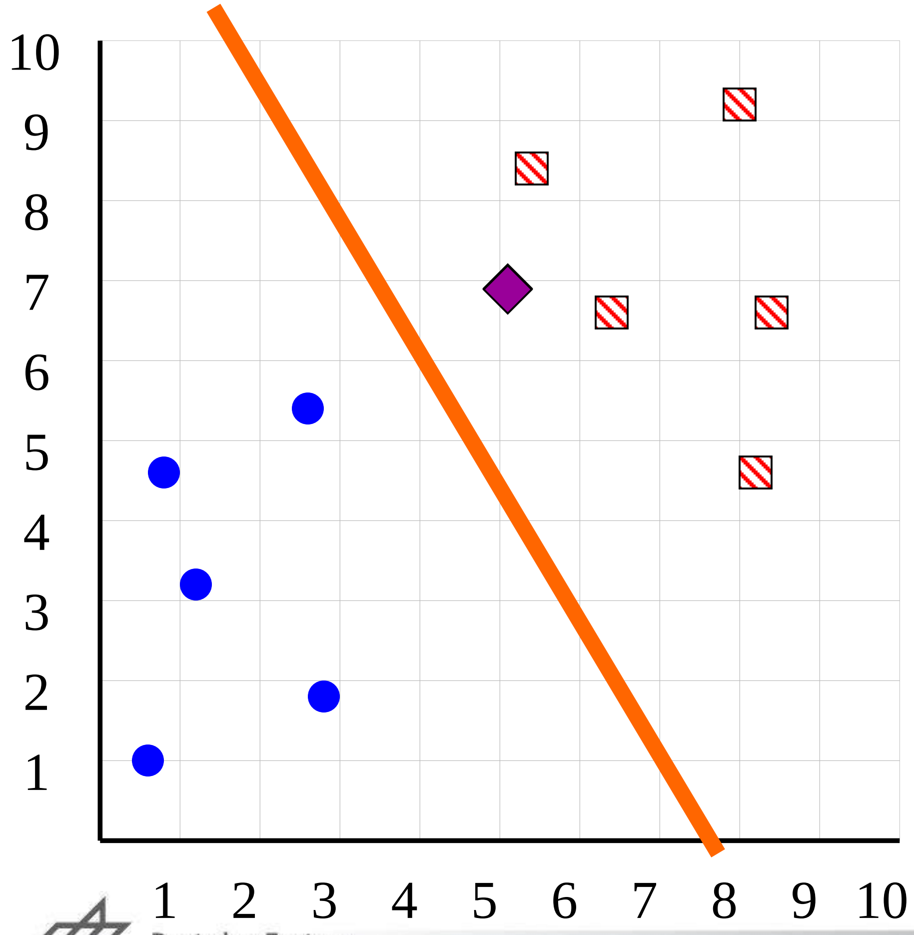
11	51	70	???????



- We can “project” the previously unseen pixel into the same space as the database.
- We have now abstracted away the details of our particular problem. It will be much easier to talk about points in space.

Vegetation
Water

Simple Linear Classifier

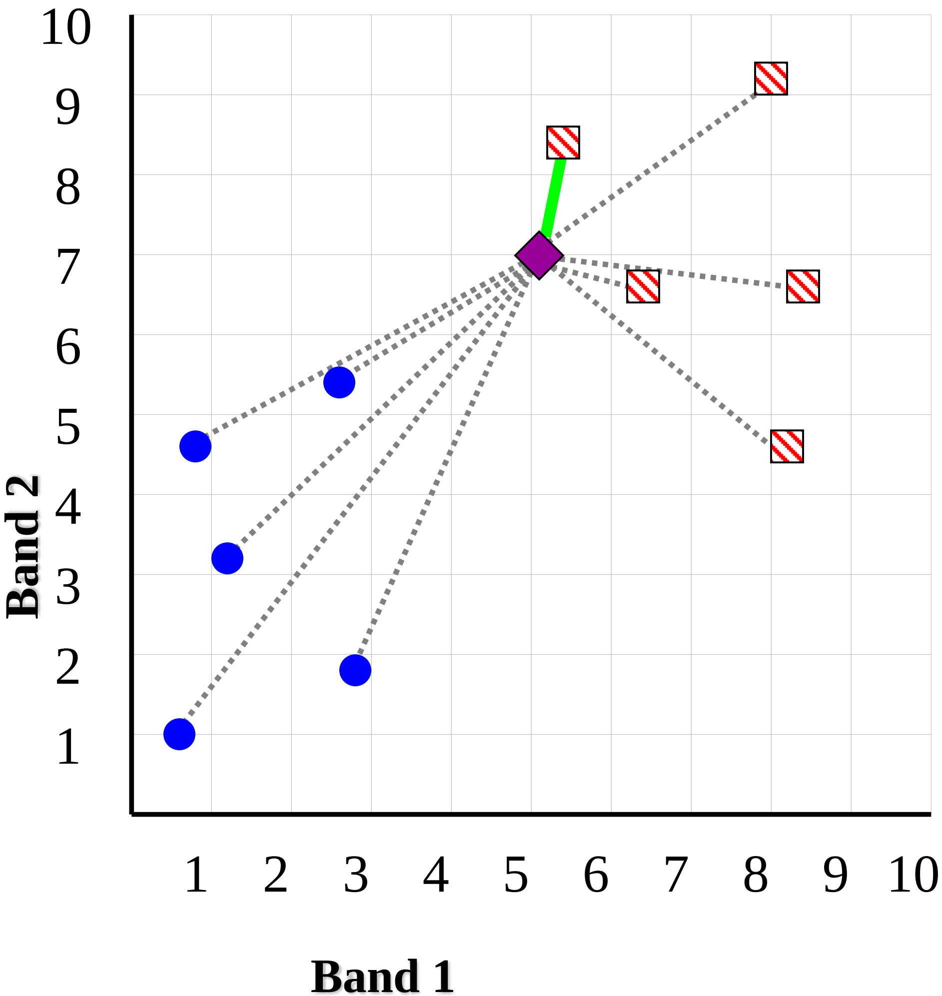


R.A. Fisher
1890-1962

If previously unseen pixel **above** the line
then
class is **Vegetation**
else
class is **Water**

Vegetation
Water

Nearest Neighbor Classifier



Evelyn Fix

1904-1965

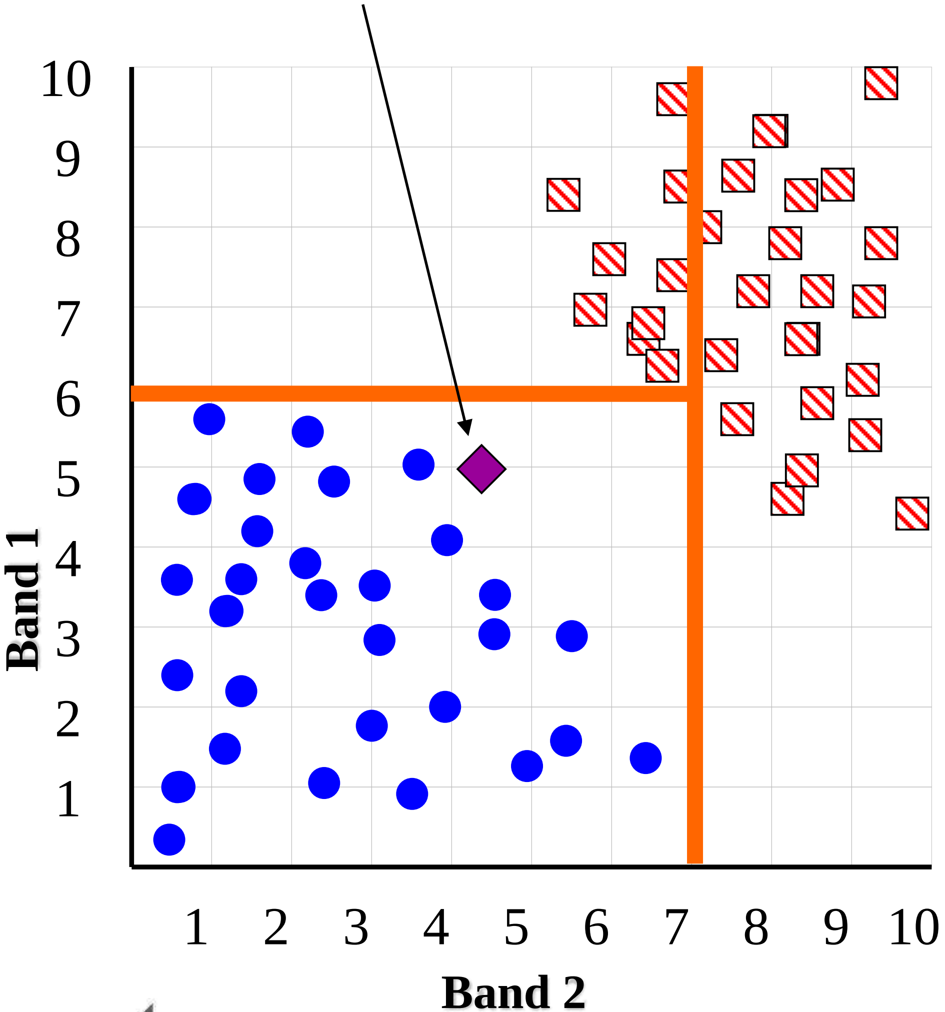
Joe Hodges

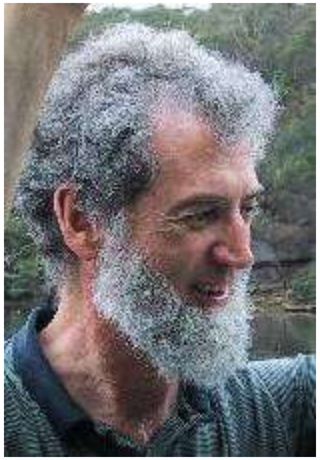
1922-2000

If the **nearest** instance to the previously unseen pixel **is Vegetation**
class is **Vegetation**
else
class is **Water**

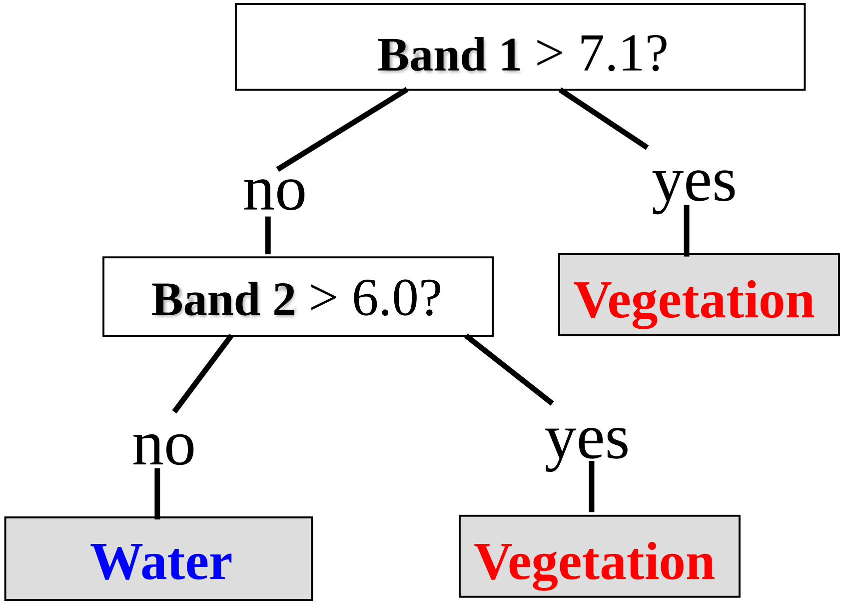
Vegetation
Water

Decision Tree Classifier



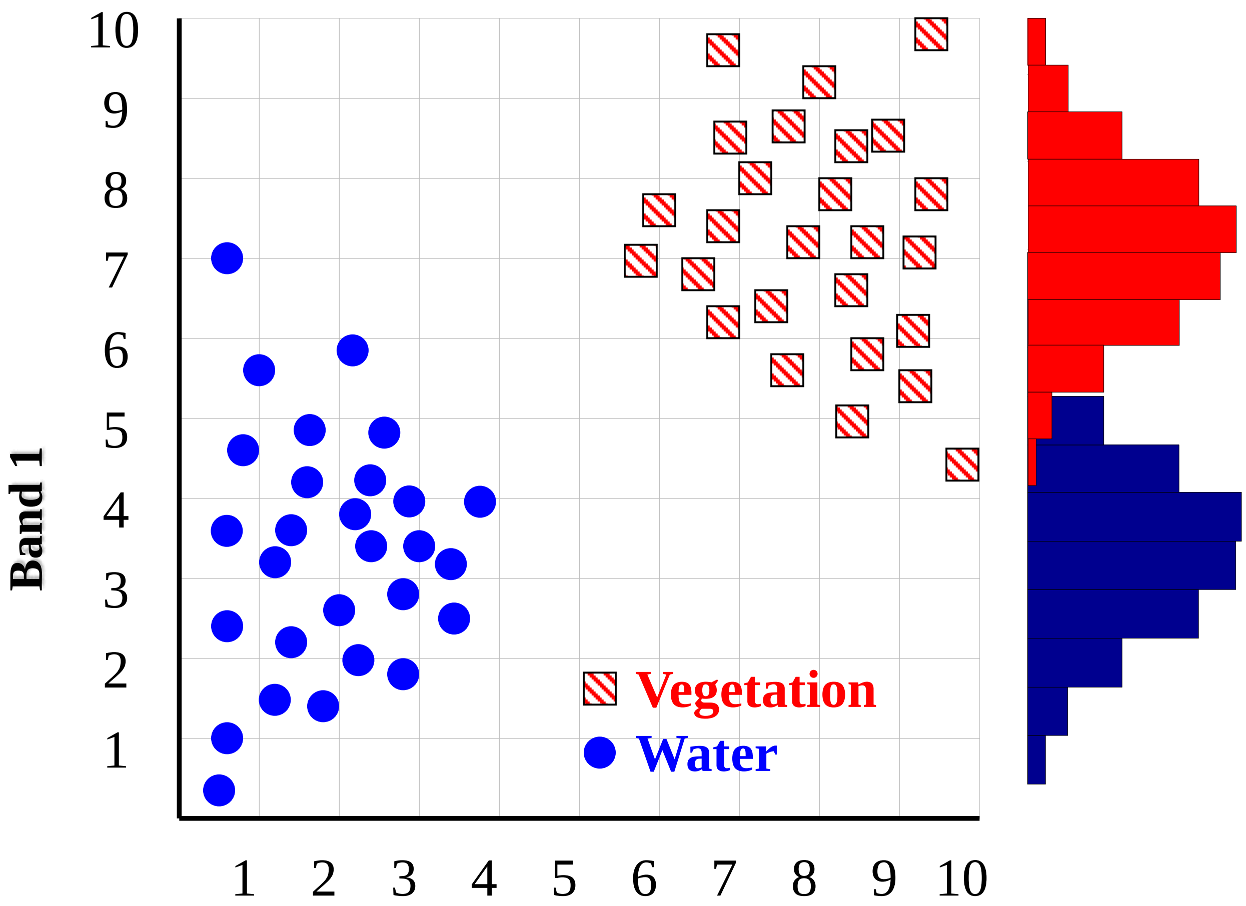


Ross Quinlan

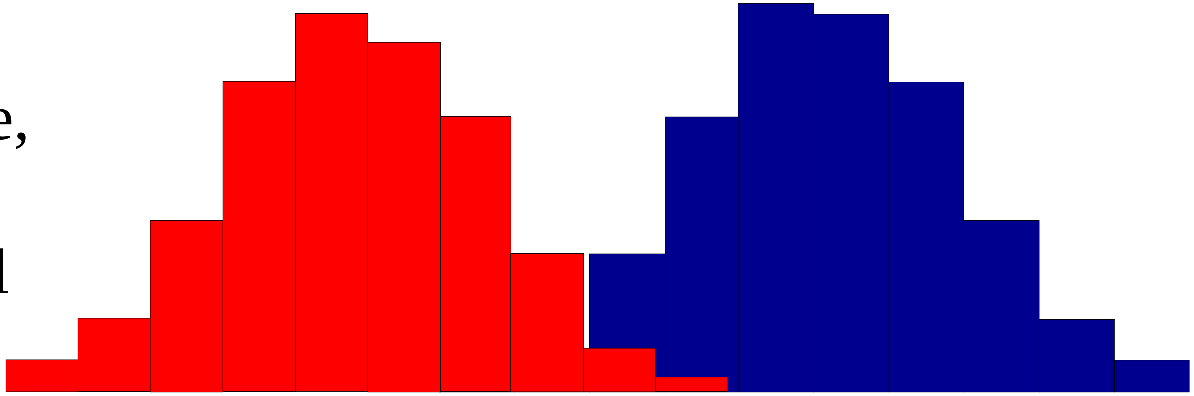


The Idea Behind the Maximum Likelihood Classifier

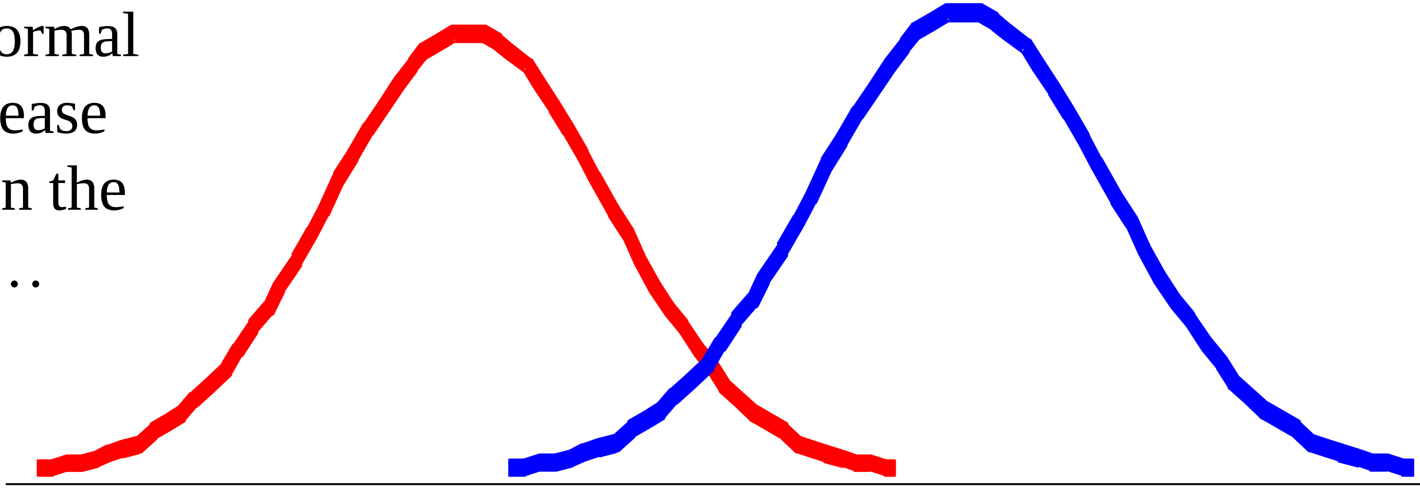
With a lot of data, we can build a histogram. Let us just build one for "Band 1" for now…



We can leave the histograms as they are, or we can summarize them with two normal (Gaussian) distributions.



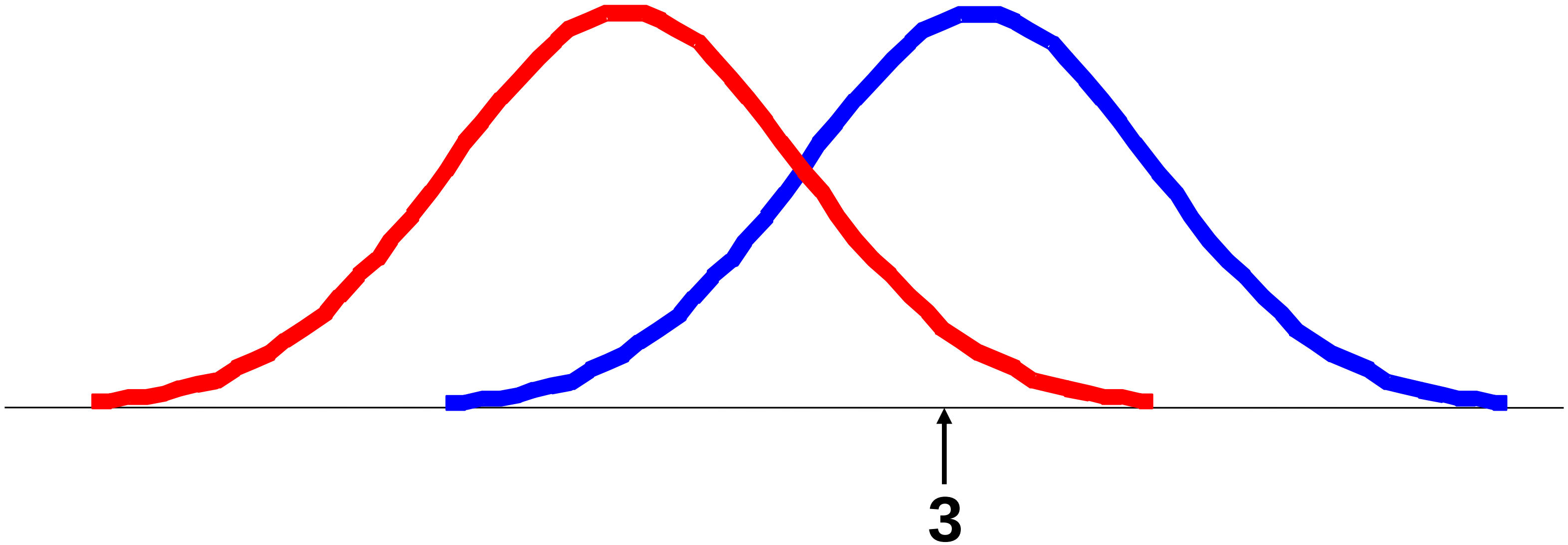
Let us use two normal distributions for ease of visualization in the following slides…



• We want to classify a pixel. Its brightness value in band 1 is equal to 3. How can we classify it?

• We can just ask ourselves, give the distributions of brightness values we have seen, if it is more *probable* that our pixel belongs to **Water** or **Vegetation**.
• There is a formal way to discuss the most *probable* classification…

$p(c_j \mid d)$ = probability of class c_j, *given* that we have observed d

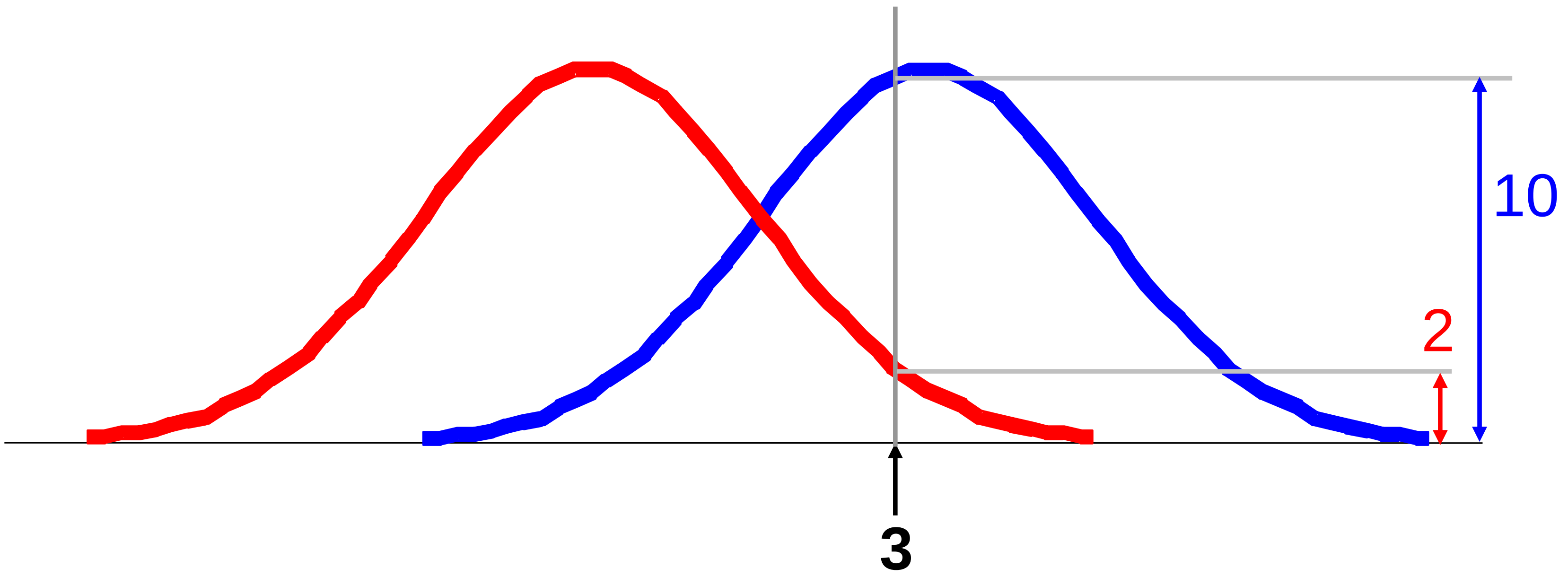


Value of band 1 is **3**

$p(c_j \mid d)$ = probability of class c_j, given that we have observed d

P(**Water** | **3**) = 10 / (10 + 2) = 0.833

P(**Vegetation** | **3**) = 2 / (10 + 2) = 0.166

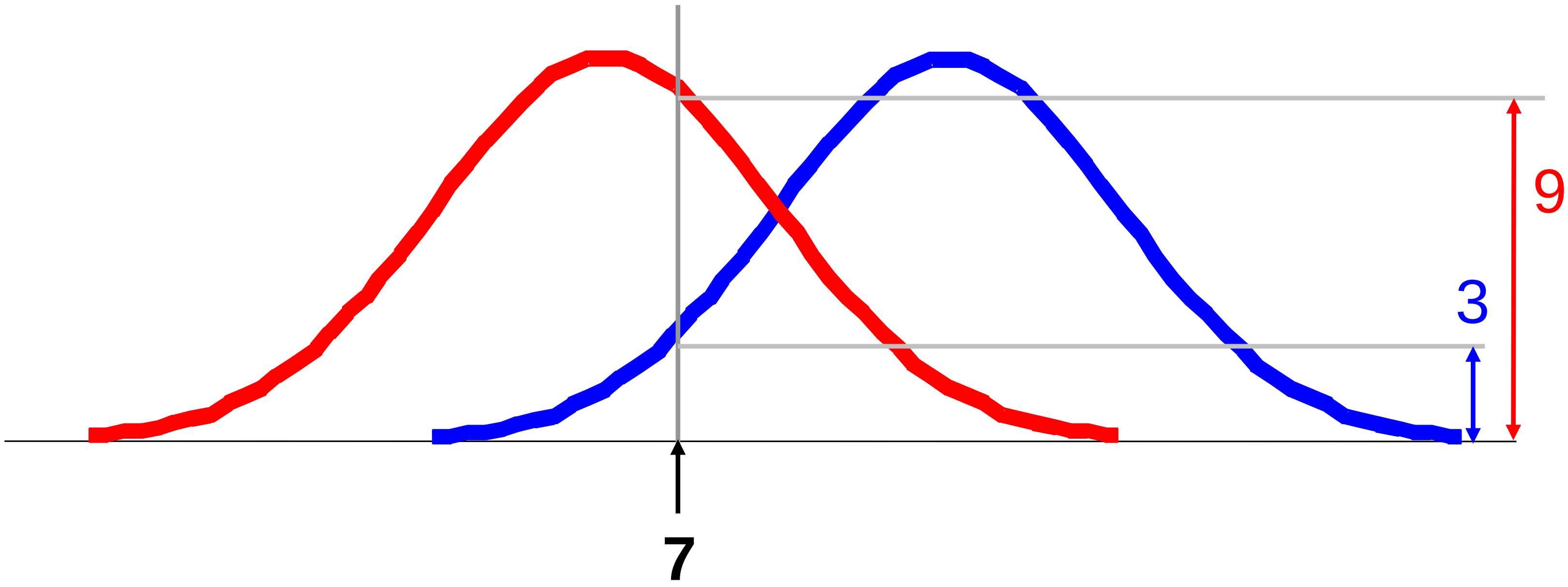


Value of band 1 is **3**

$p(c_j \mid d)$ = probability of class c_j, given that we have observed d

P(**Water** | **7**) = 3 / (3 + 9) = 0.250

P(**Vegetation**| **7**) = 9 / (3 + 9) = 0.750

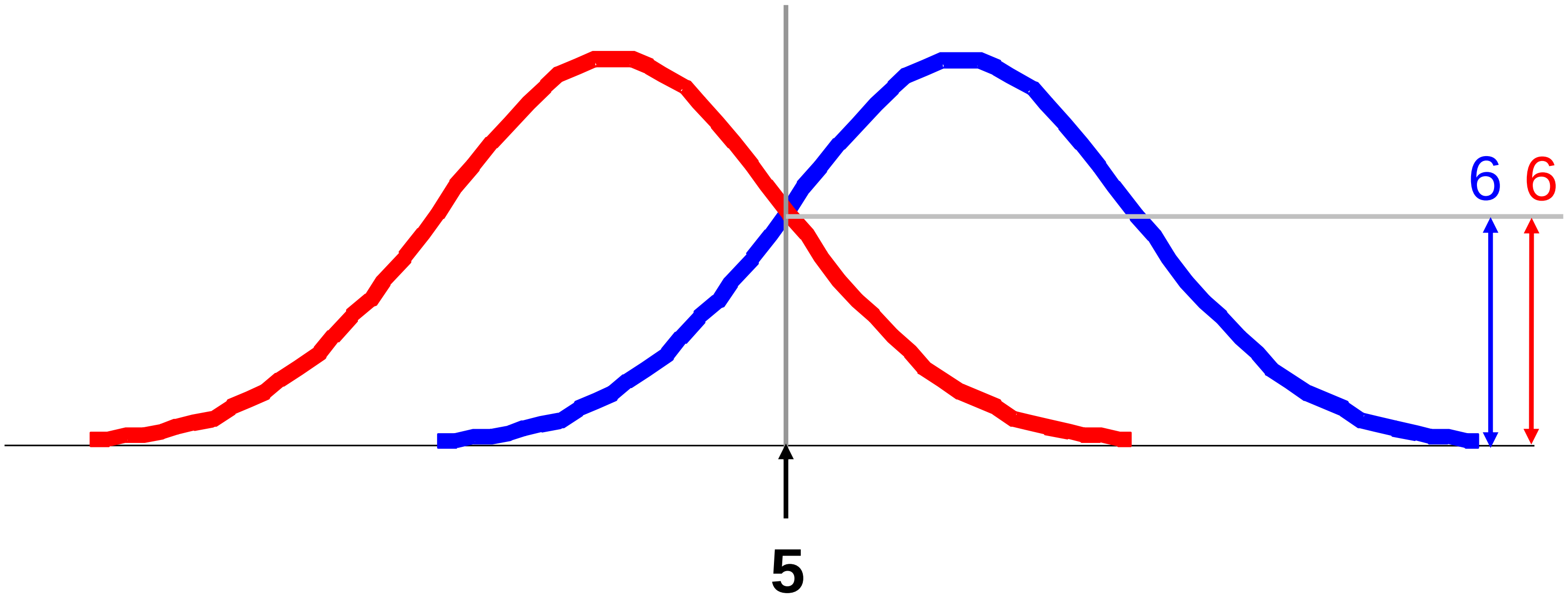


Value of band 1 is **7**

$p(c_j \mid d)$ = probability of class c_j, given that we have observed d

P(**Water** | **5**) = 6 / (6 + 6) = 0.500

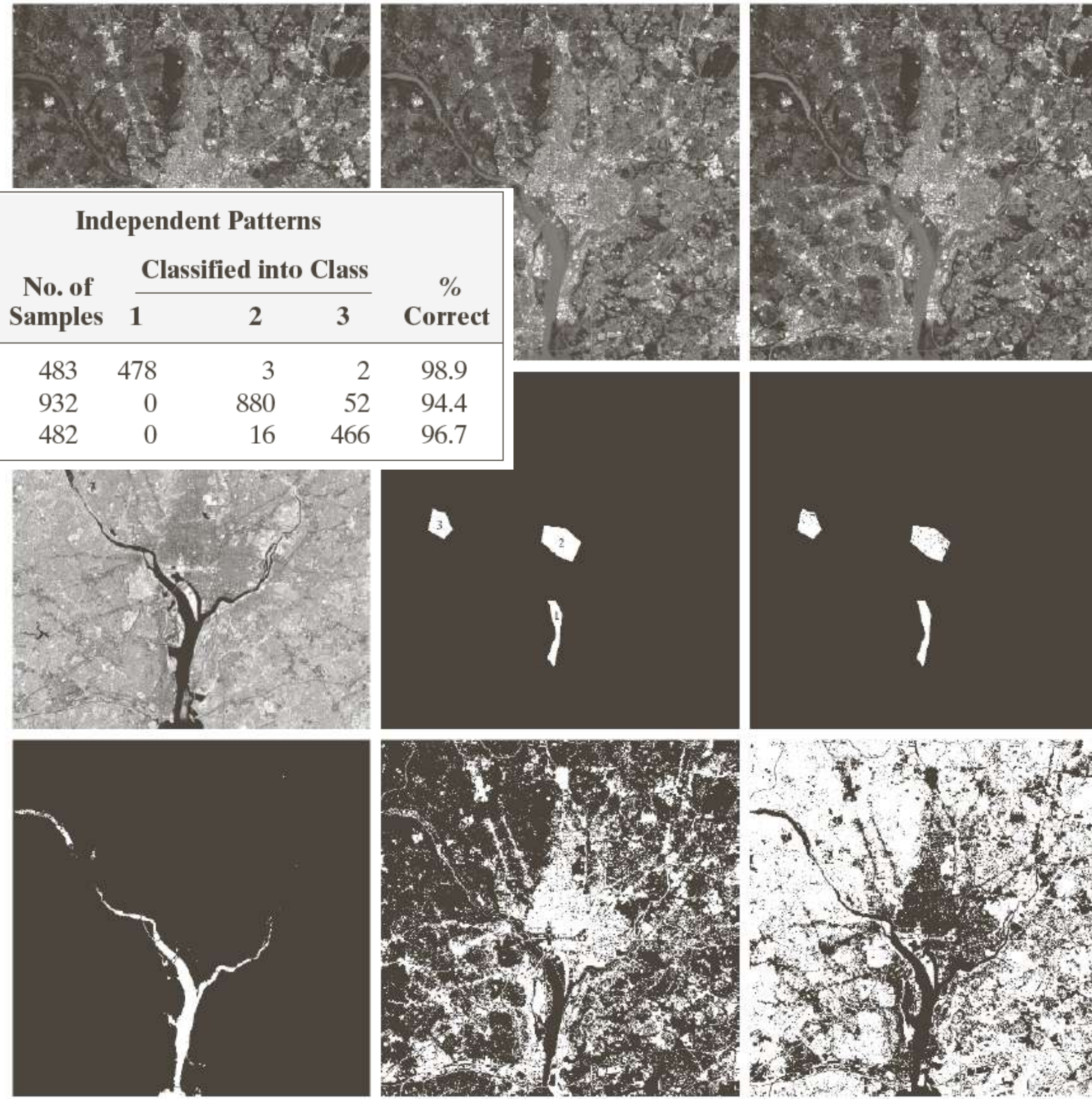
P(**Vegetation** | **5**) = 6 / (6 + 6) = 0.500



Value of band 1 is **5**

Sample Application

Training Patterns						Independent Patterns					
		Classified into Class						Classified into Class			
Class	No. of Samples	1	2	3	% Correct	Class	No. of Samples	1	2	3	% Correct
1	484	482	2	0	99.6	1	483	478	3	2	98.9
2	933	0	885	48	94.9	2	932	0	880	52	94.4
3	483	0	19	464	96.1	3	482	0	16	466	96.7



a b c
d e f
g h i

FIGURE 12.13 Bayes classification of multispectral data. (a)–(d) Images in the visible blue, visible green, visible red, and near infrared wavelengths. (e) Mask showing sample regions of water (1), urban development (2), and vegetation (3). (f) Results of classification; the black dots denote points classified incorrectly. The other (white) points were classified correctly. (g) All image pixels classified as water (in white). (h) All image pixels classified as urban development (in white). (i) All image pixels classified as vegetations (in white).

Support Vector Machine (SVM)

- Non-probabilistic binary linear classifier
- Finds an optimal hyperplane to separate the instances of the two classes
- Extends to patterns that are not linearly separable by mapping the data onto a higher-dimensionality space
- Resistent to outliers
- Handles very well high-dimensionality data such as hyperspectral images
- One of the most popular classification algorithms since the ´90s

SVM et al. – Story of a Success

BVM
Tsang et al.
2007

GVM
Mansoory.
2009

TVM
Wang, Vutetic,
2010

AVM
Wang, Wu,
2009

IVM
Hastie, Zhu,
2005

RVM
Tipping.
2001

SVM
Vapnik et al.
1992

BVM
Friedman et al.
1998

QVM
Wang, Wu,
2007

IVM
Lawrence
2003

DVM
Awad et al.
2004

CVM
Tsang et al.
2007

EVM
Chen et al.
2008

VVM
Minka et al.
2009

PVM
Zhang et al.
2007

LVM
Singer
2000

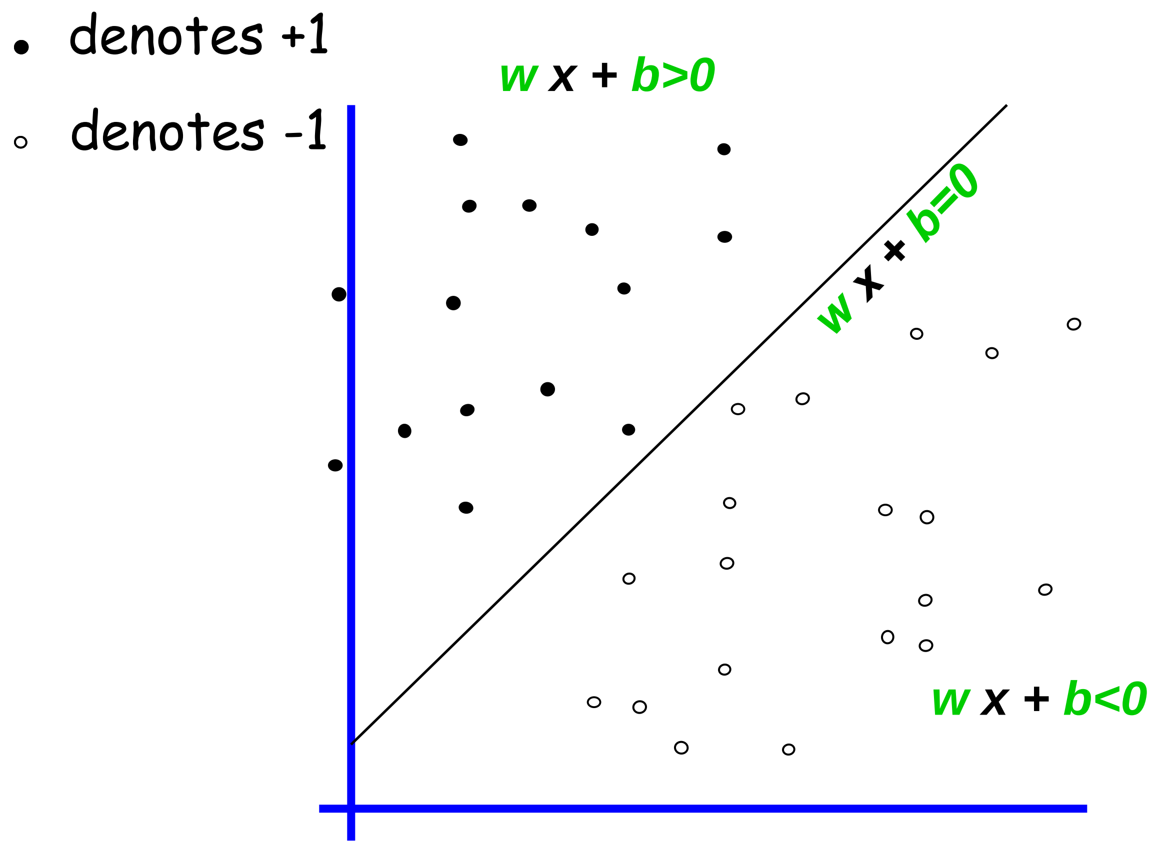
OVM
Yang.
2009

FVM
Li et al.
2006

XVM
Haffner,
2002

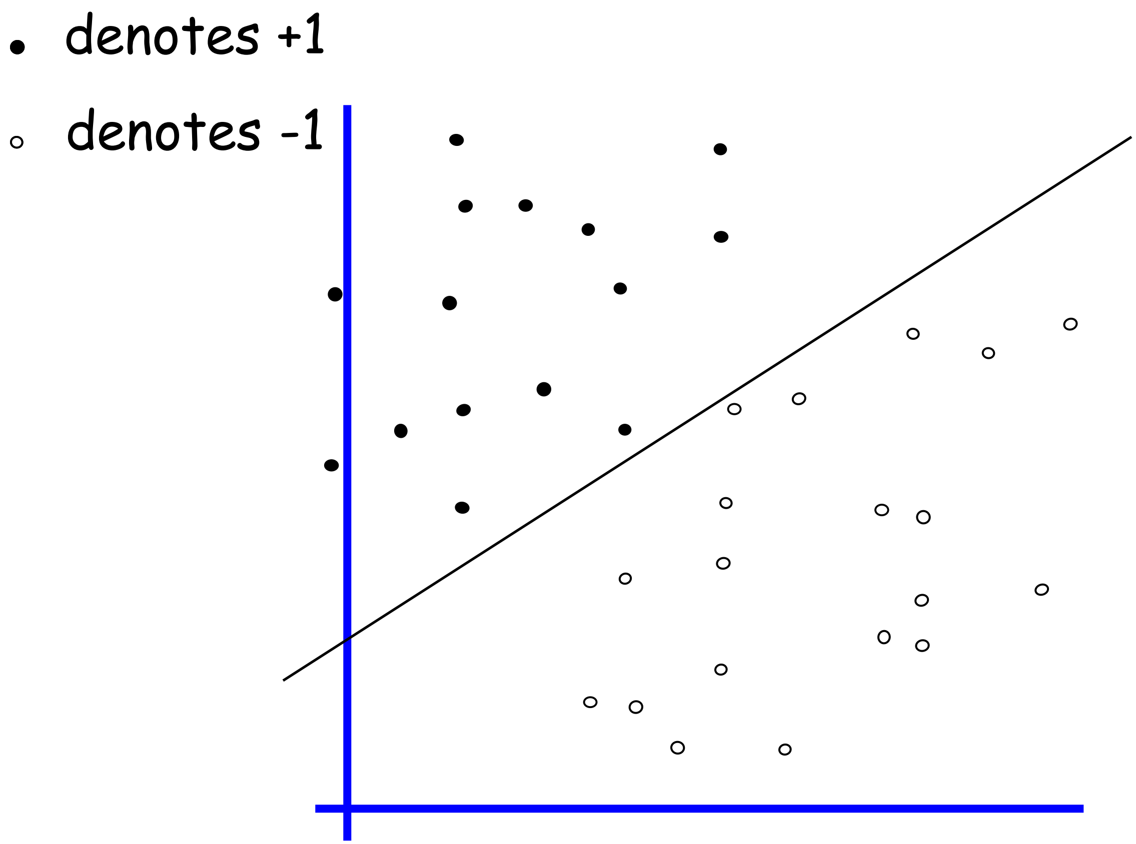
Adapted from Andreas Braun, KIT

Let's go back to linear classifiers…



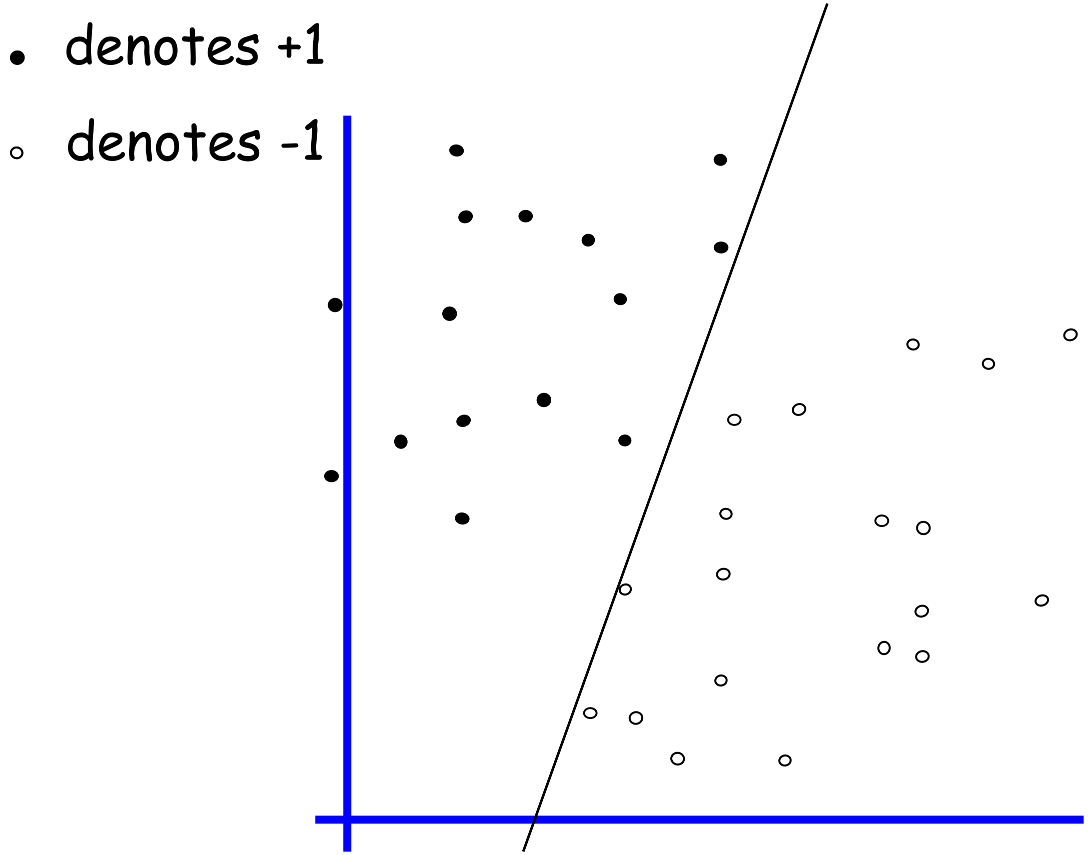
How would you classify this data?

Linear Classifiers



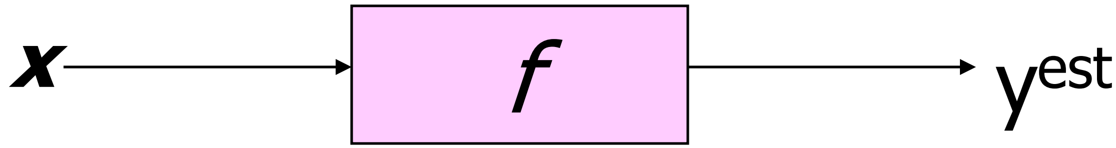
How would you classify this data?

Linear Classifiers



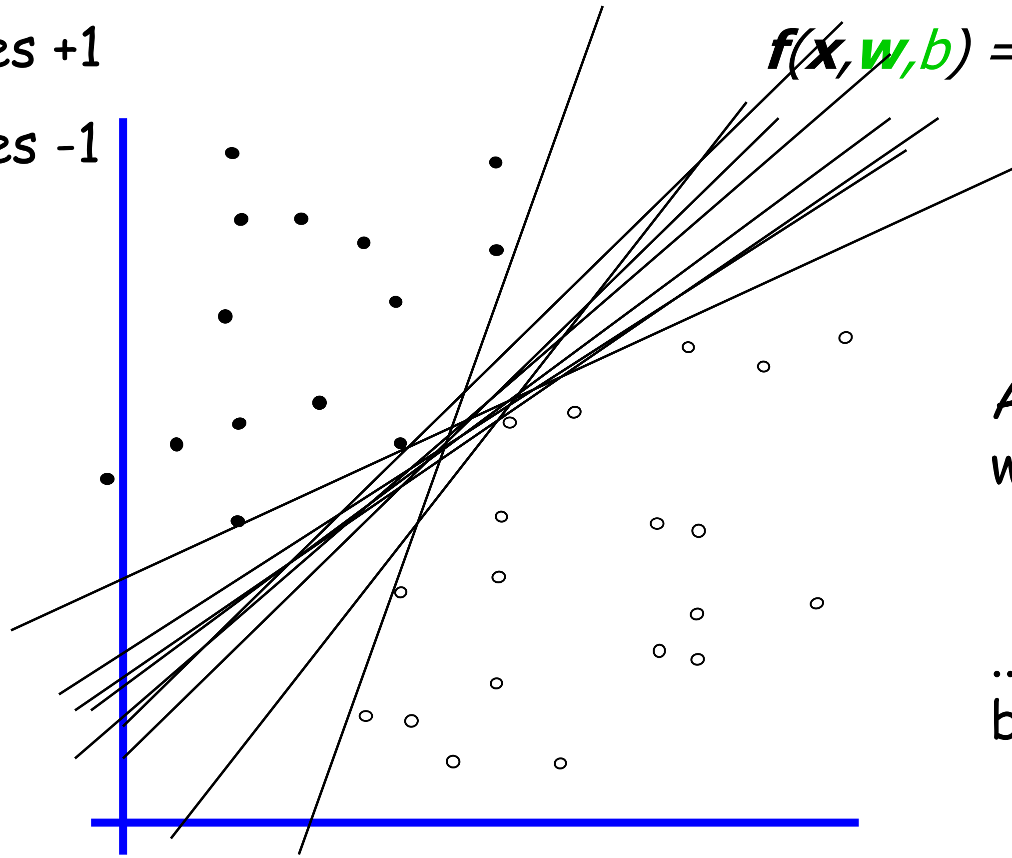
How would you classify this data?

Linear Classifiers



$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w}\,\mathbf{x} + b)$$

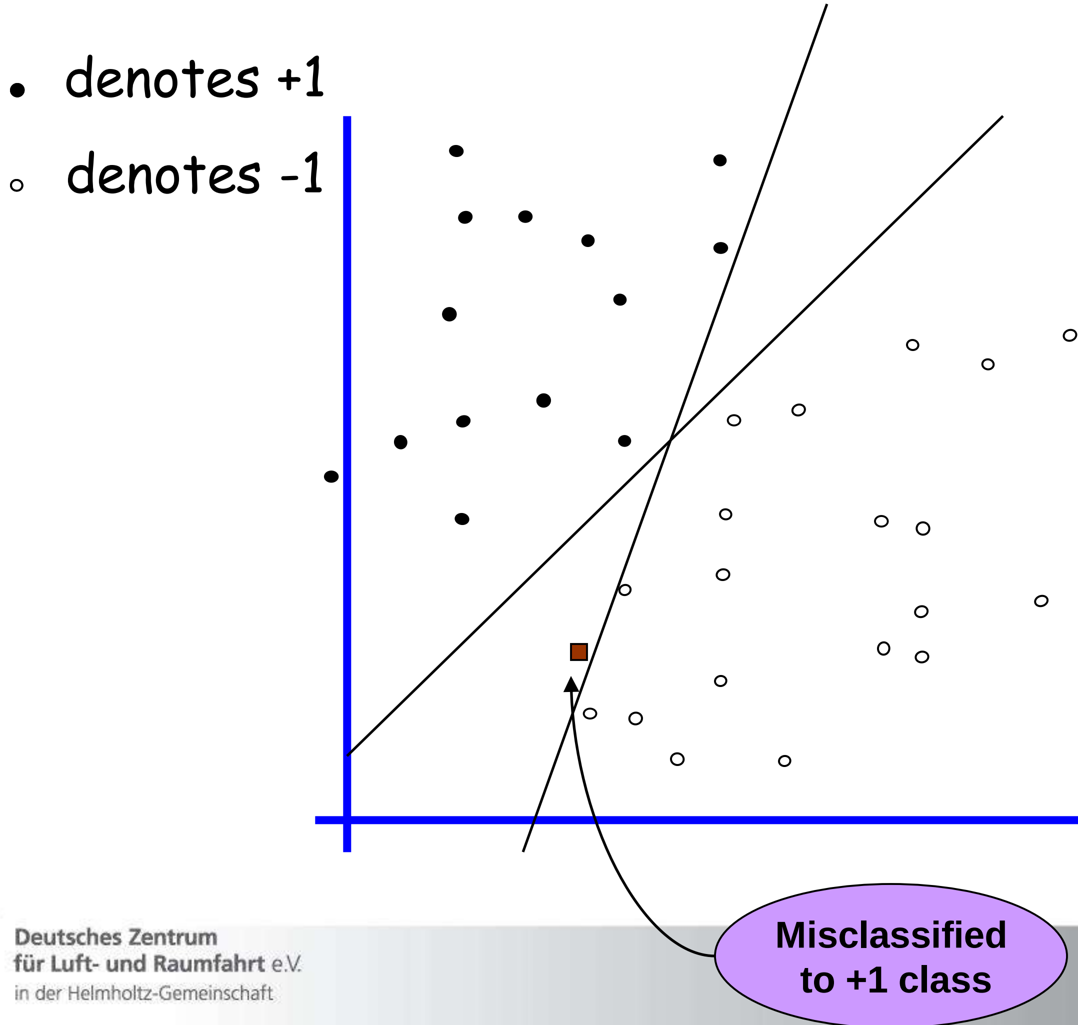
- ● denotes +1
- ○ denotes -1



Any of these would be fine..

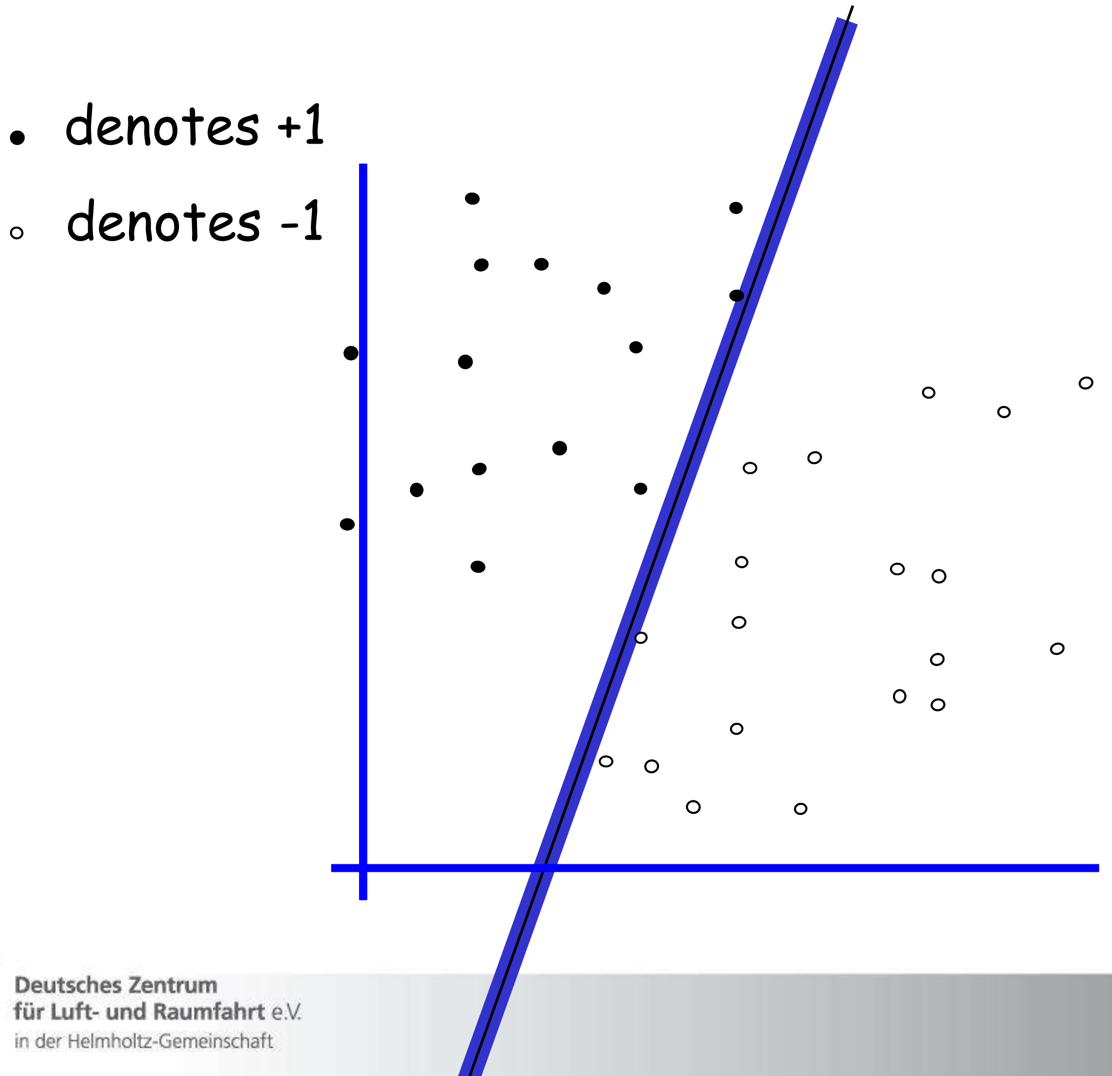
..but which is best?

Linear Classifiers



How would you classify this data?

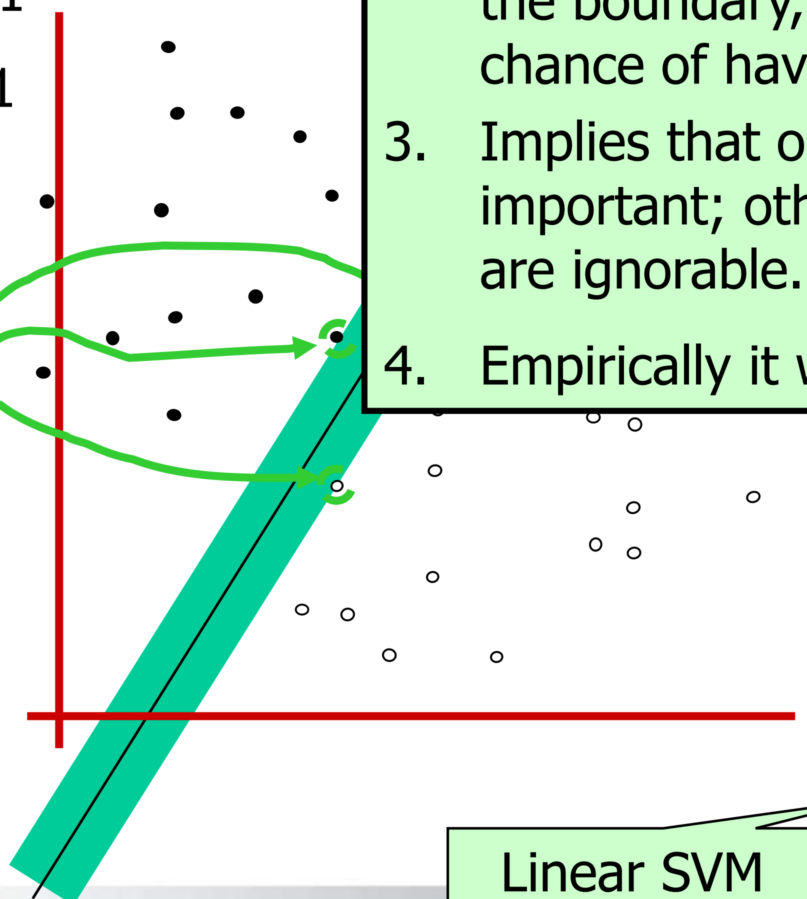
Classifier Margin



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin

- • denotes +1
- ◦ denotes -1



1. Intuitively this feels safest.
2. If we made a small error in locating the boundary, we have a smaller chance of having misclassifications.
3. Implies that only support vectors are important; other training examples are ignorable.
4. Empirically it works very very well.

Support Vectors are those datapoints that the margin pushes up against

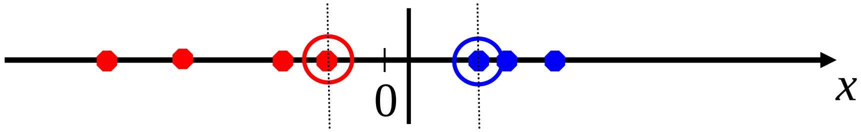
maximum margin.

This is the simplest kind of SVM (Called an LSVM)

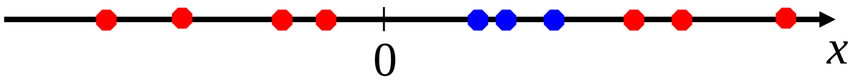
Linear SVM

Non-linear SVMs

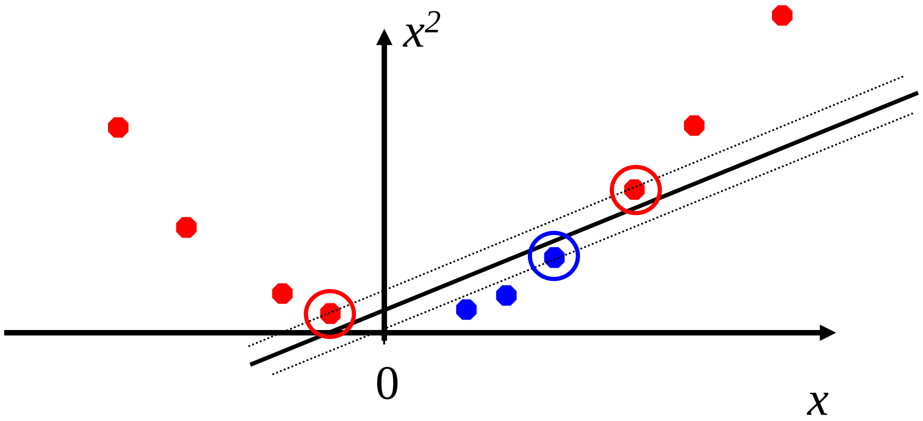
- Datasets that are linearly separable with some noise work out great:



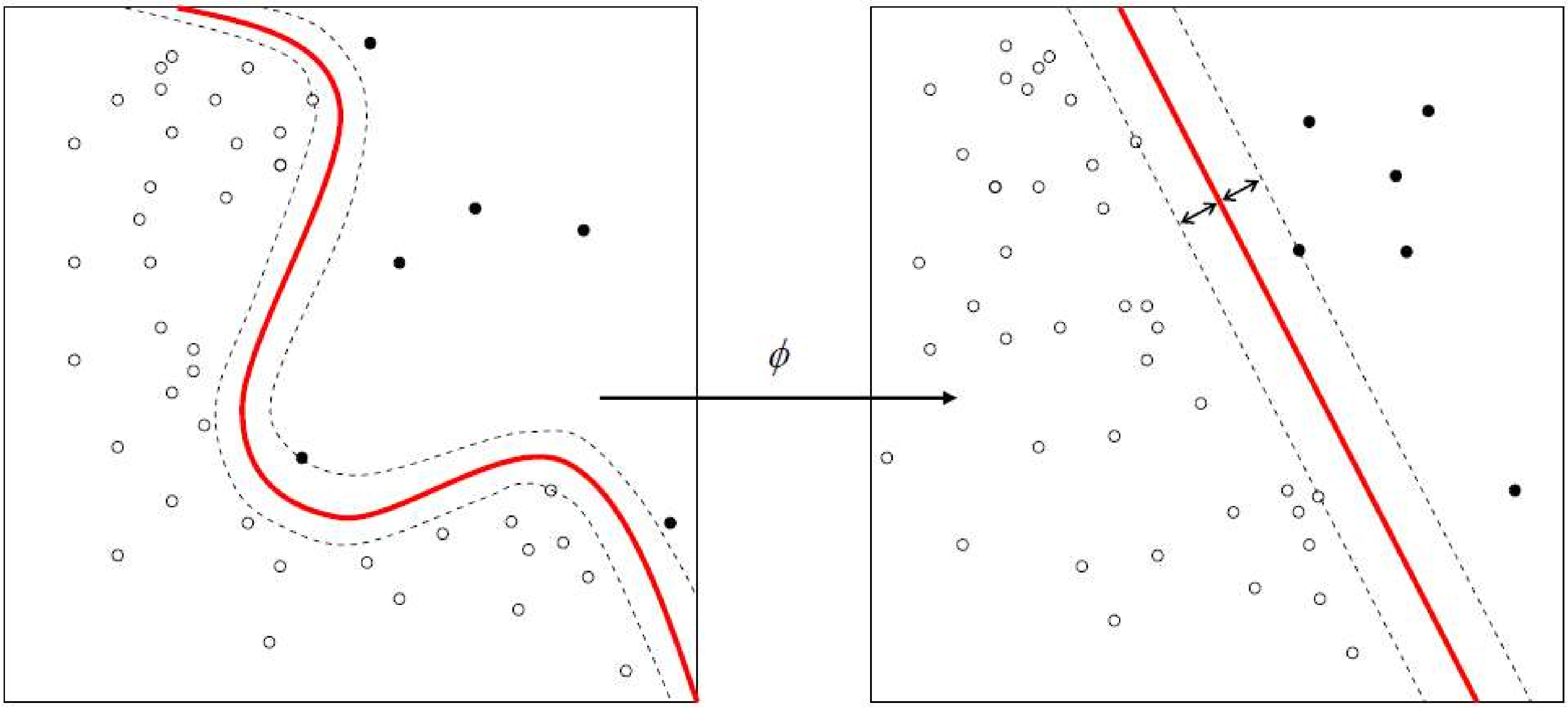
- But what are we going to do if the dataset is just too hard?



- How about… mapping data to a higher-dimensional space:



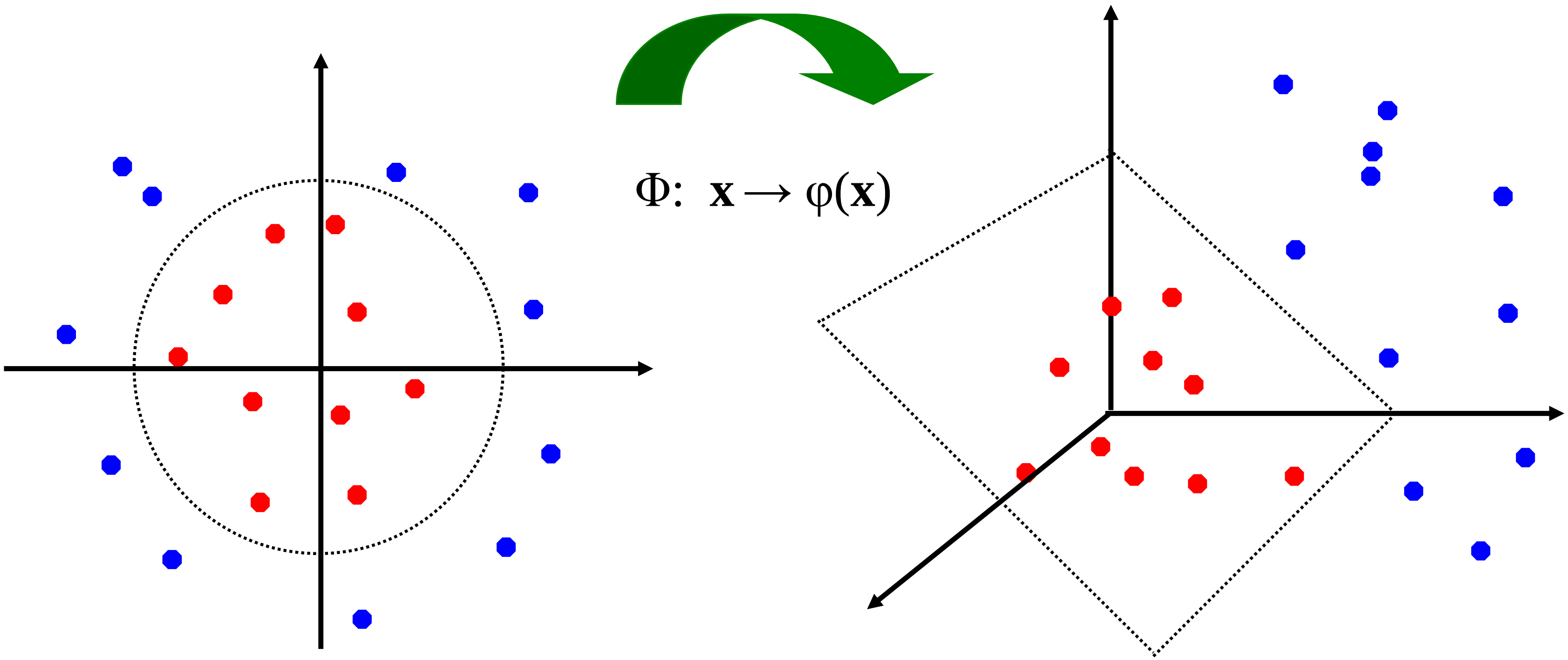
Non-linear SVM



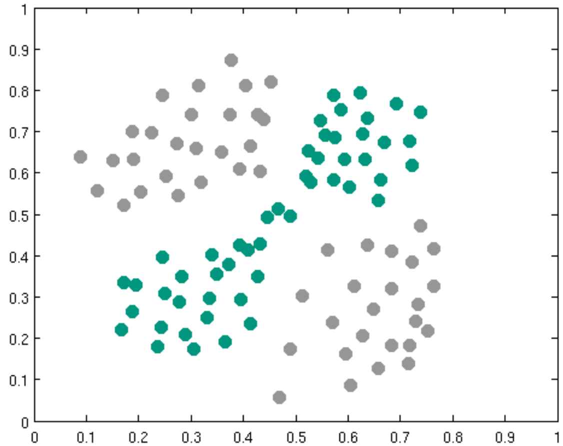
Kernel Trick

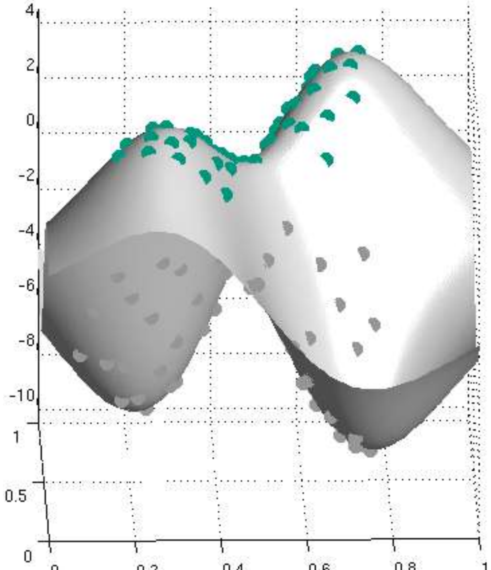
Non-linear SVMs: Feature spaces

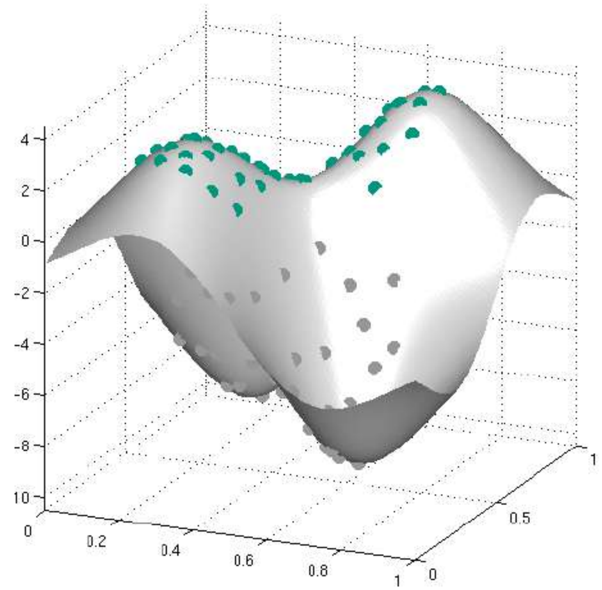
- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:

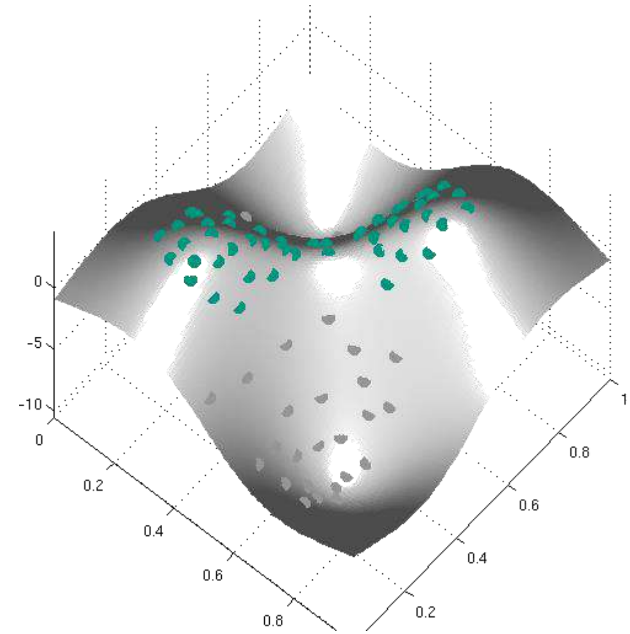


SVM Decisions Surface



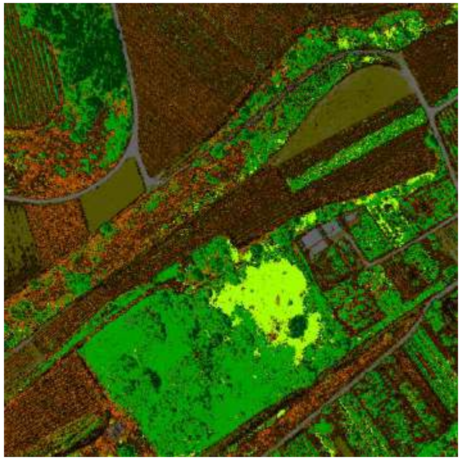


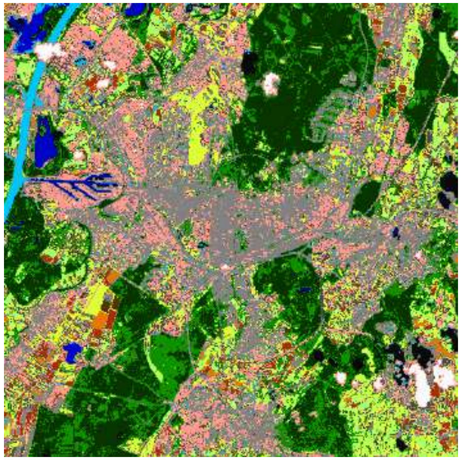


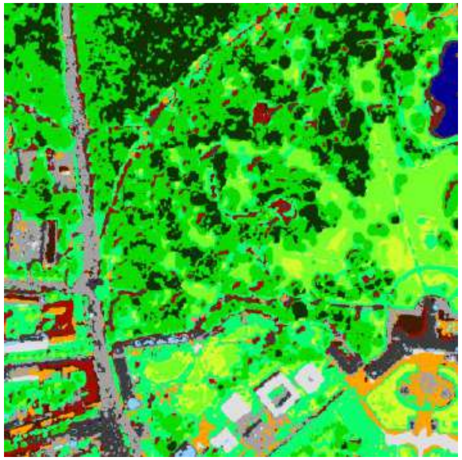


SVM-based Classification

RGB Picture | **Landsat** | **QuickBird**







Airborne Sensor | **Multispectral – low resolution** | **Multispectral - High resolution**

Disadvantages of SVM

- SVM is robust, works with any kind of data, and yields good classification results. Why should we take care with using an SVM classifier then?

→It does not give in output a probability density function

→It is designed only to separate two classes

→It often needs a high number of SVs (computation time)

Summary of Classification

We have briefly introduced 4 classification techniques:

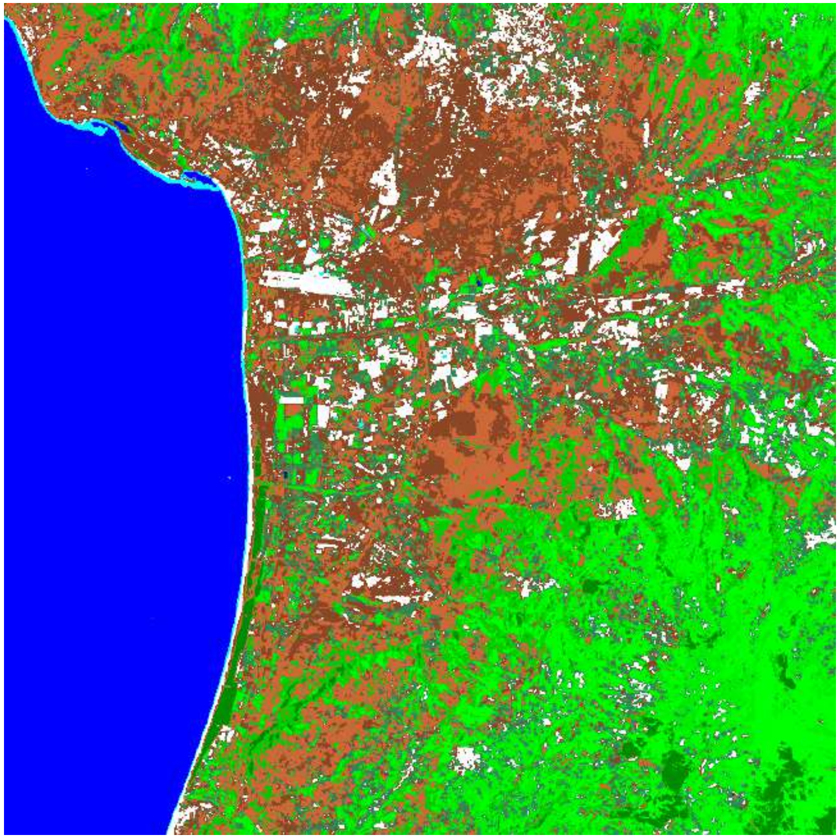
- Simple linear classifier
- Nearest neighbor
- Decision tree
- Maximum Likelihood

There are other, more sophisticated techniques:

- Support Vector Machines (briefly described if we have time)
- Neural Networks, Genetic algorithms..

In general, there is no one best classifier for all problems. You have to consider what you hope to achieve, and the data at hand…

How to Improve & Regularize our results?



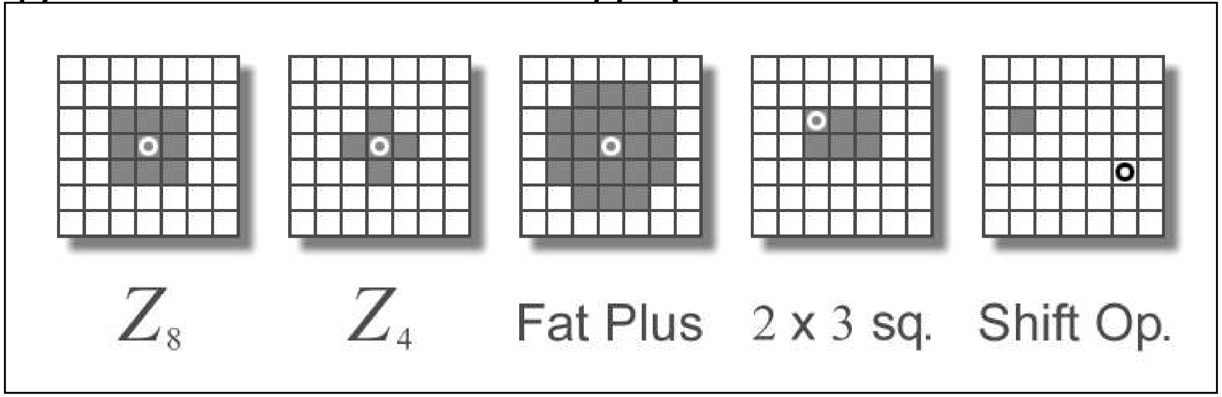
Morphological Operators

"Your x-ray showed a broken rib, but we fixed it with Photoshop."

Structuring Element (SE)

A structuring element is a small image – used as a moving window – whose support delineates pixel neighborhoods in the image plane.

Example SEs

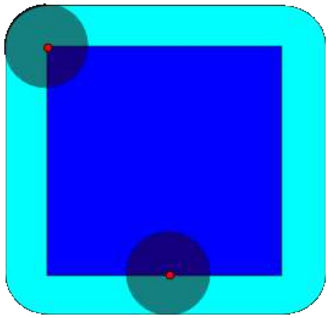


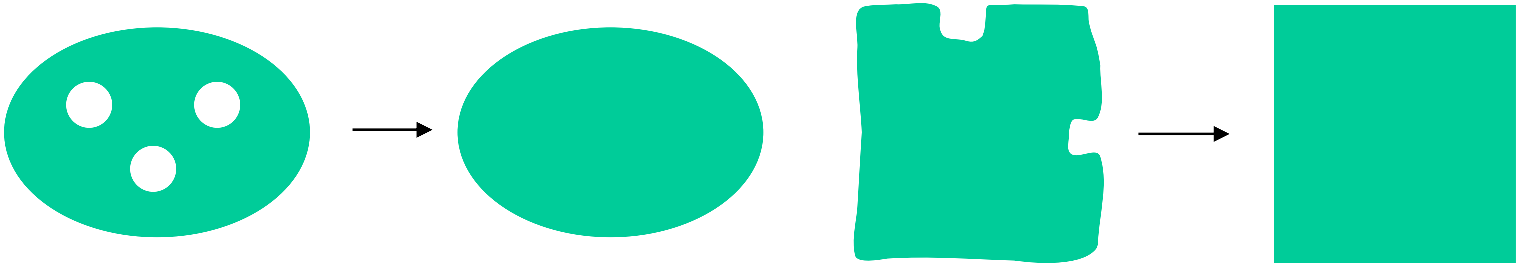
FG is gray ;
BG is white

It can be of any shape, size, or connectivity (more than 1 piece, have holes). In the figure the circles mark the location of the structuring element's origin which can be placed anywhere relative to its support.

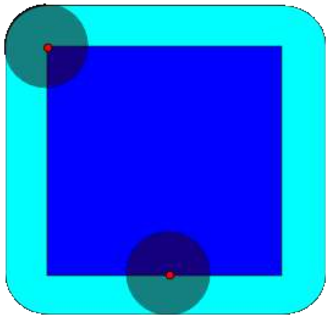
Dilation

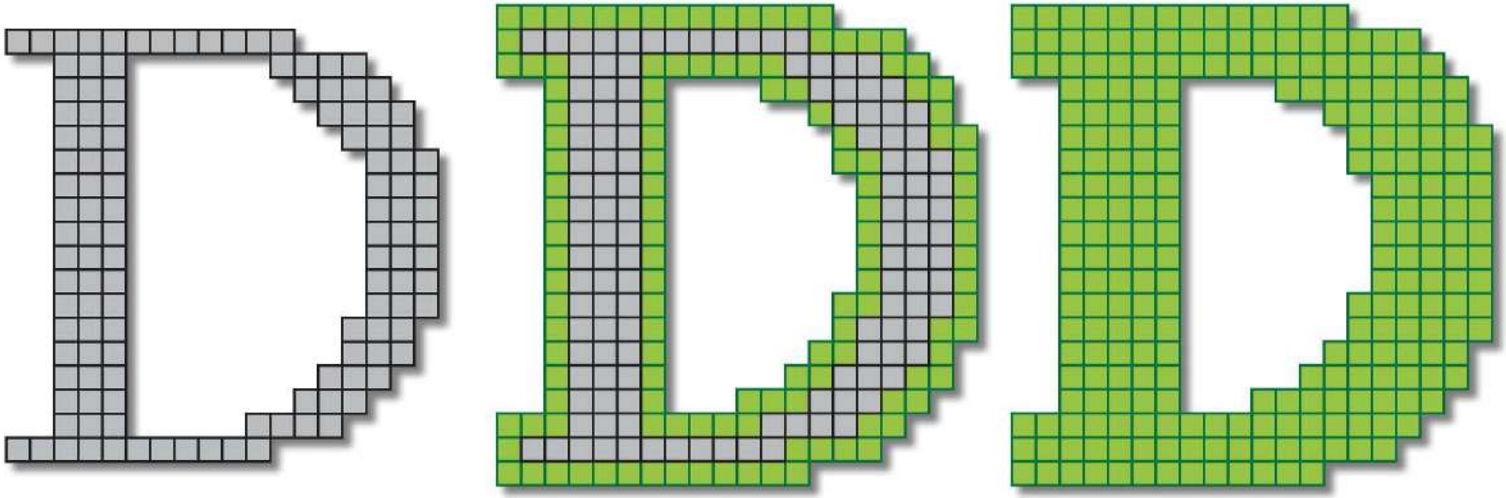
- Dilation expands the connected sets of 1s of a binary image.
- It can be used for
  - growing features
  - filling holes and gaps



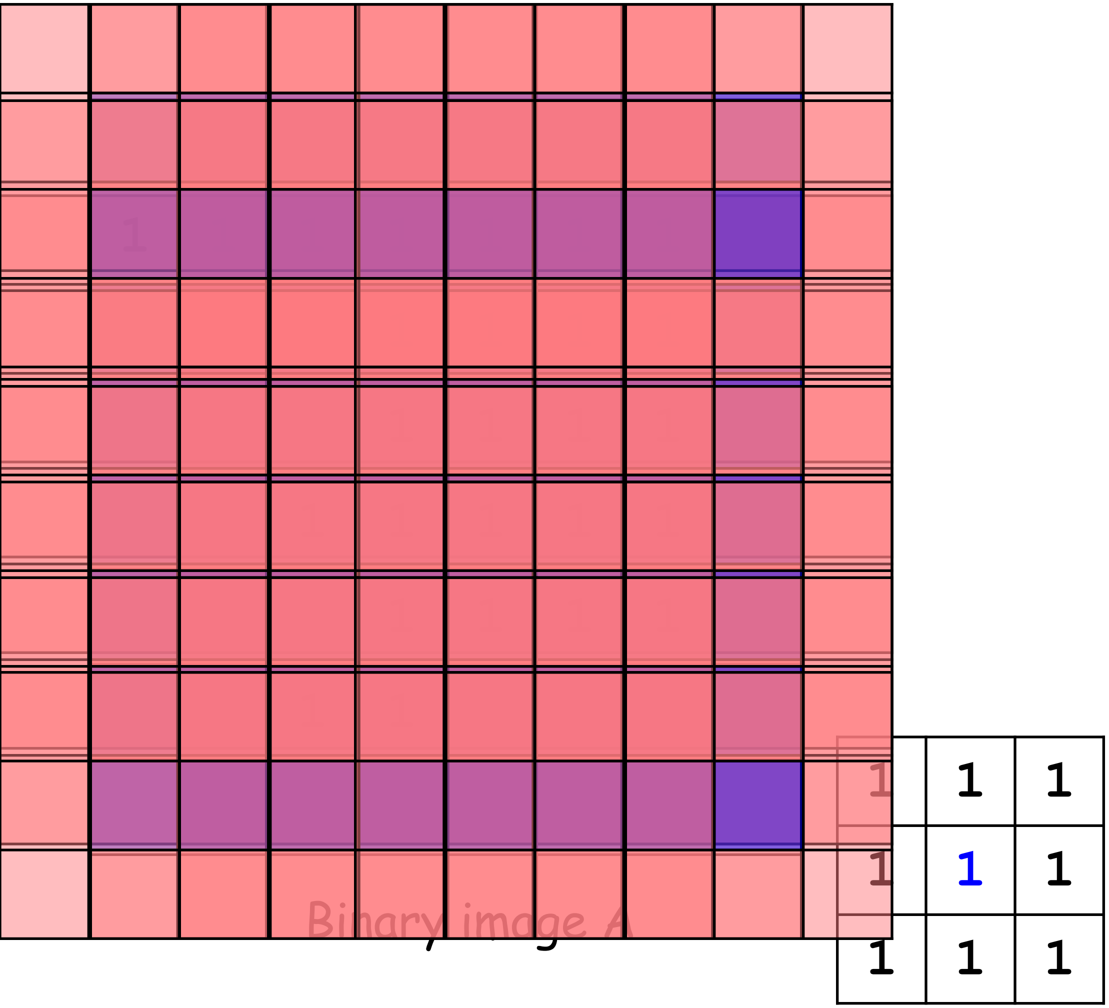


Dilation





Dilation

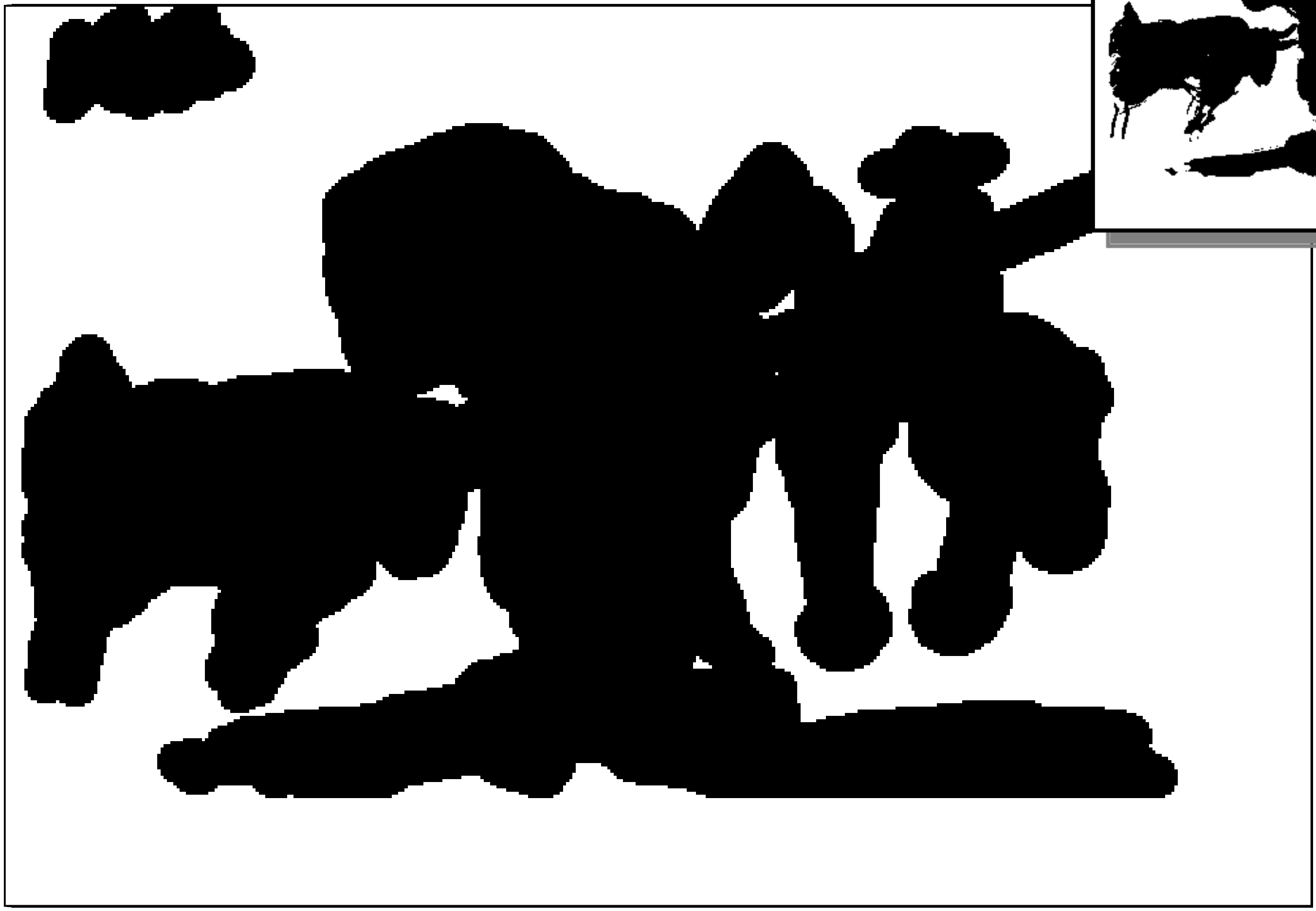


Structuring element B

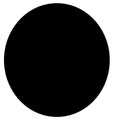
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	1	1	1	1	0	0	0

Dilation result

Dilation



Structuring Element

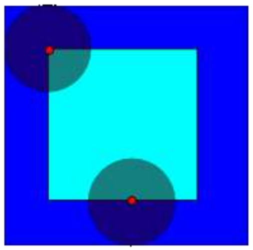


Pablo Picasso, *Pass with the Cape*, 1960

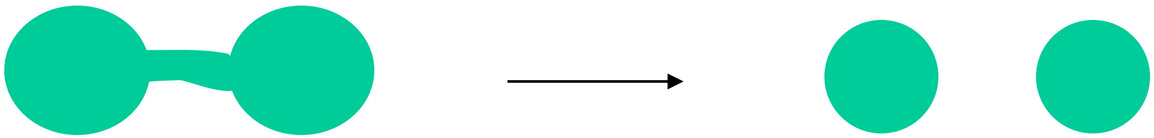
Adapted from John Goutsias, Johns Hopkins Univ.

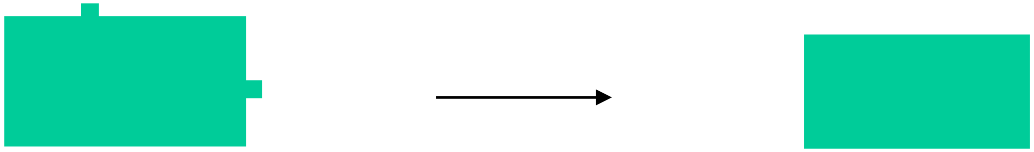
Erosion

- Erosion shrinks the connected sets of 1s of a binary image.
- It can be used for
  - Shrinking features

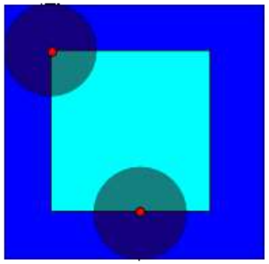


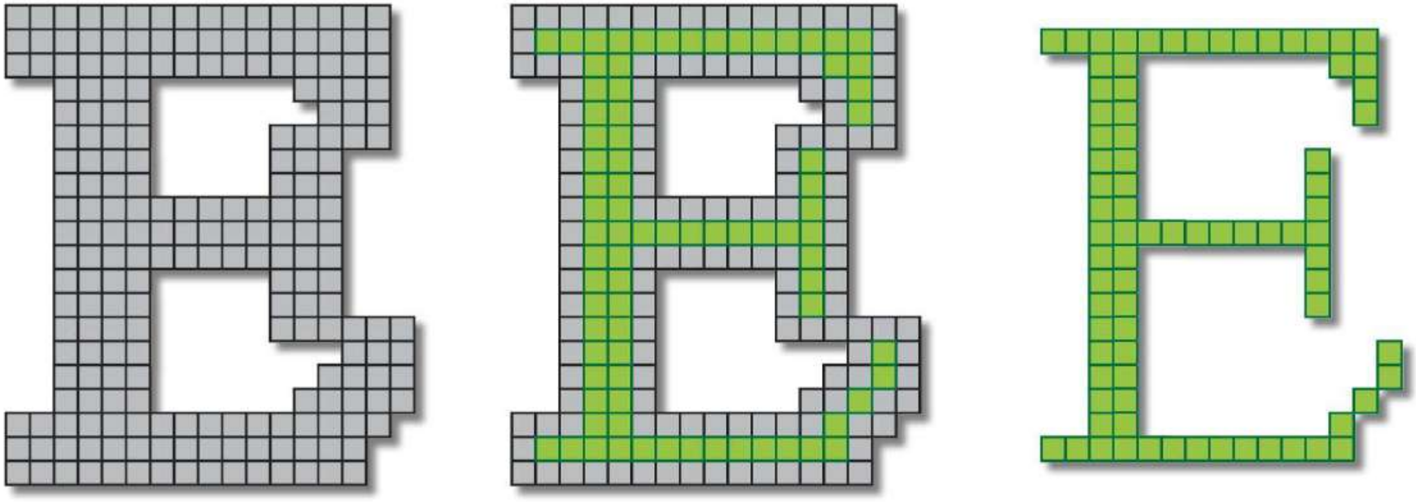
  - Removing bridges, branches and small protrusions



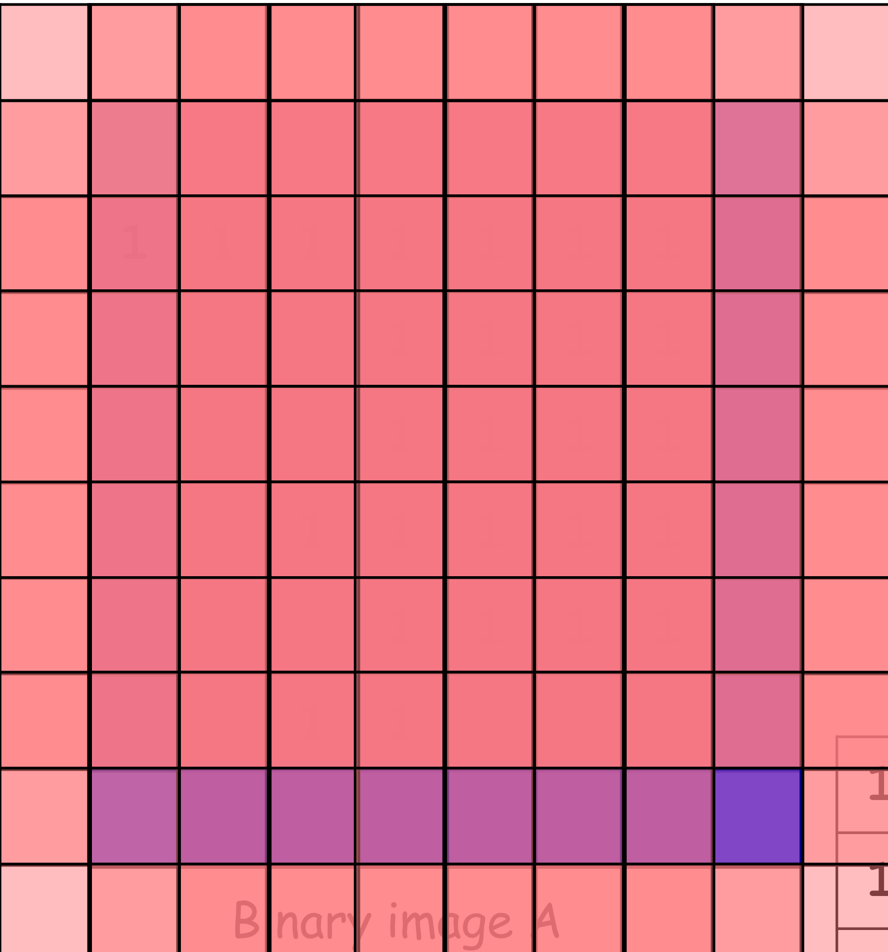


Erosion





Erosion



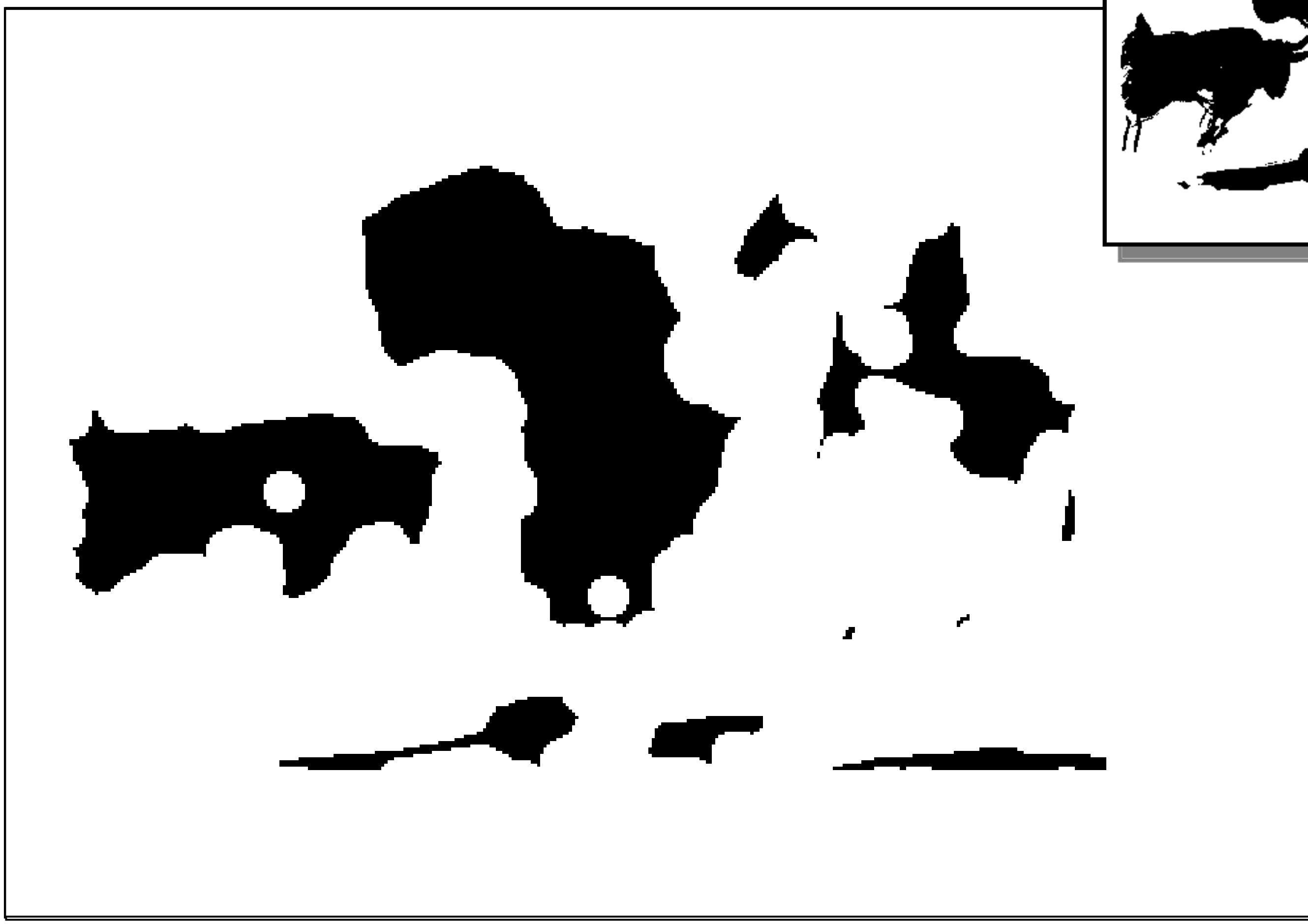
1	1	1
1	1	1
1	1	1

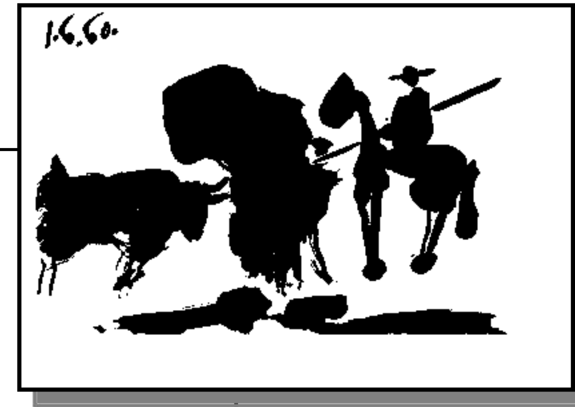
Structuring element B

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

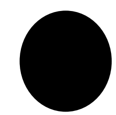
Erosion result

Erosion



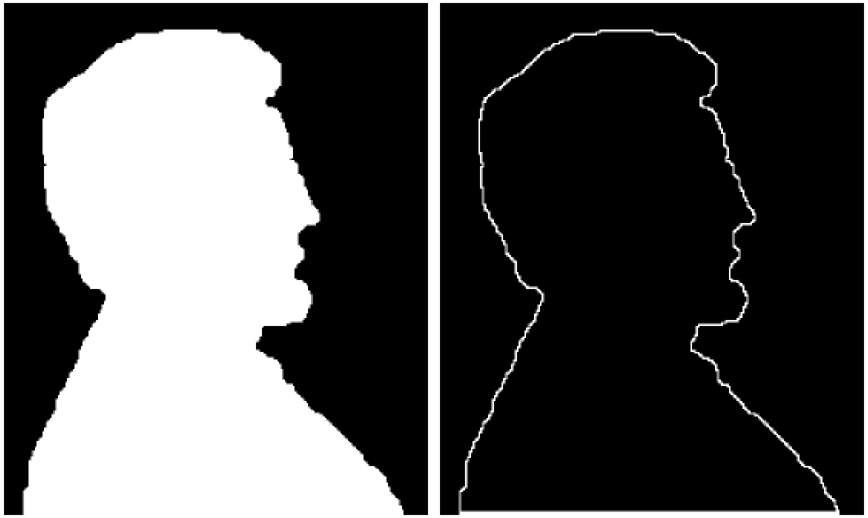


Structuring Element



Pablo Picasso, *Pass with the Cape*, 1960

Boundary extraction



a b

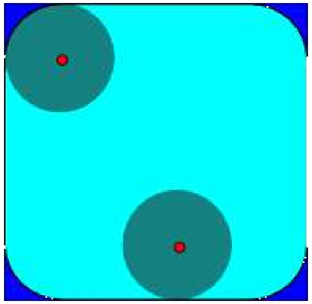
FIGURE 9.14 (a) A simple binary image, with 1's represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

$$Boundary(I) = I - (I \rightarrow erosion(Z))$$

$$\downarrow$$

$$Boundary(I) = I - (I \ominus Z)$$

Opening



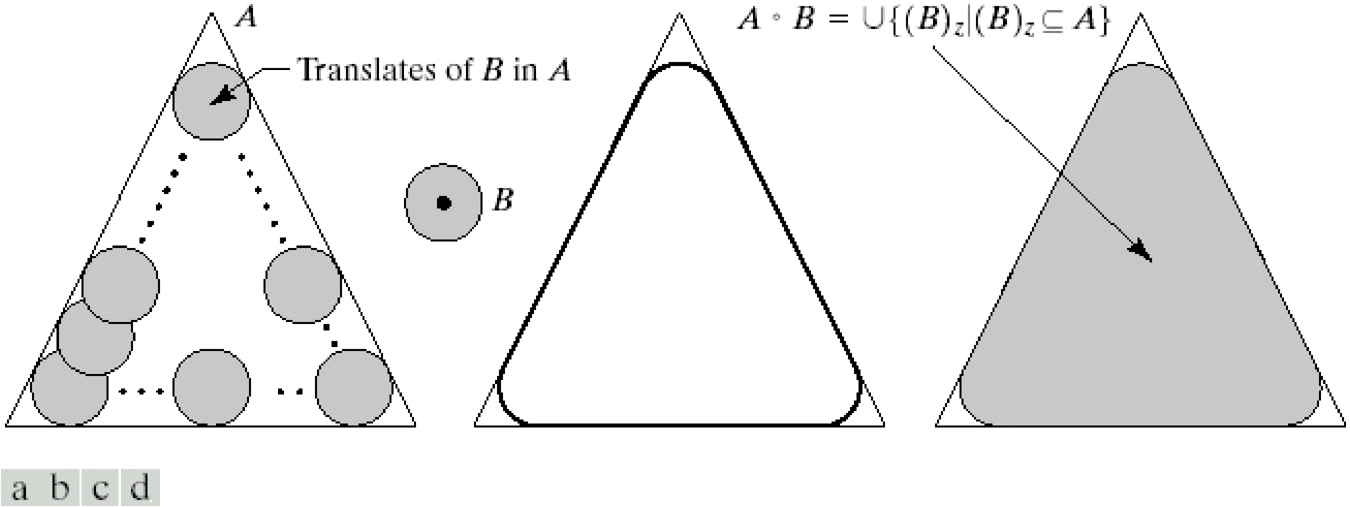


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

Erosion + Dilation → Opening

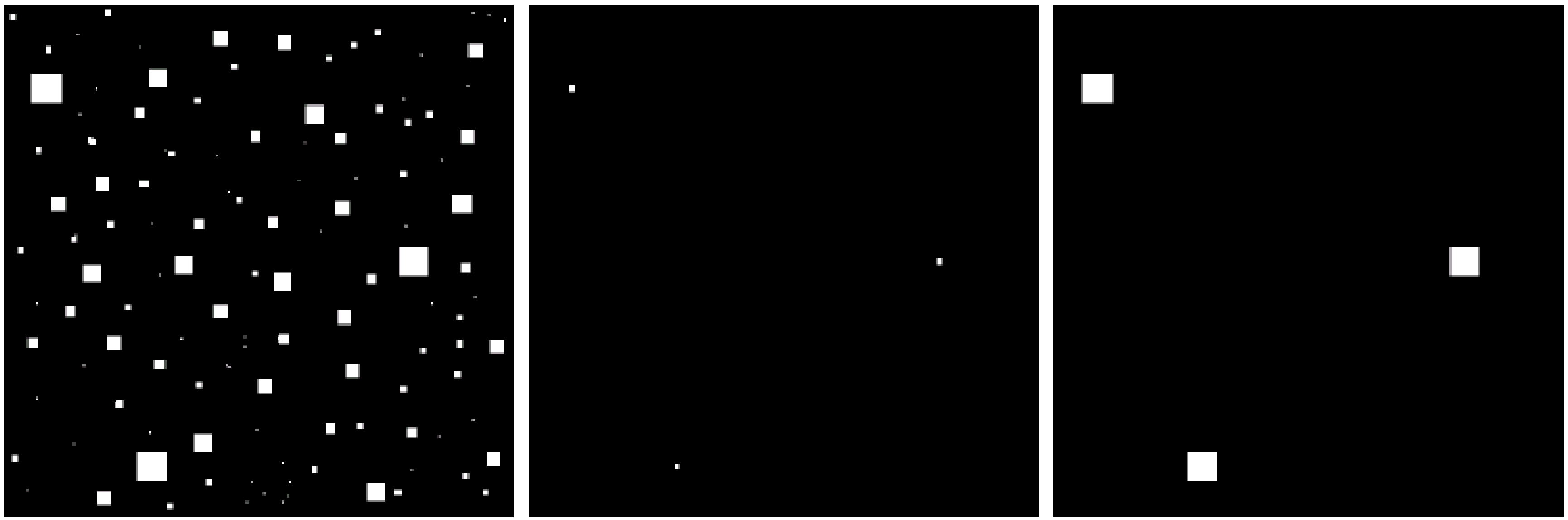
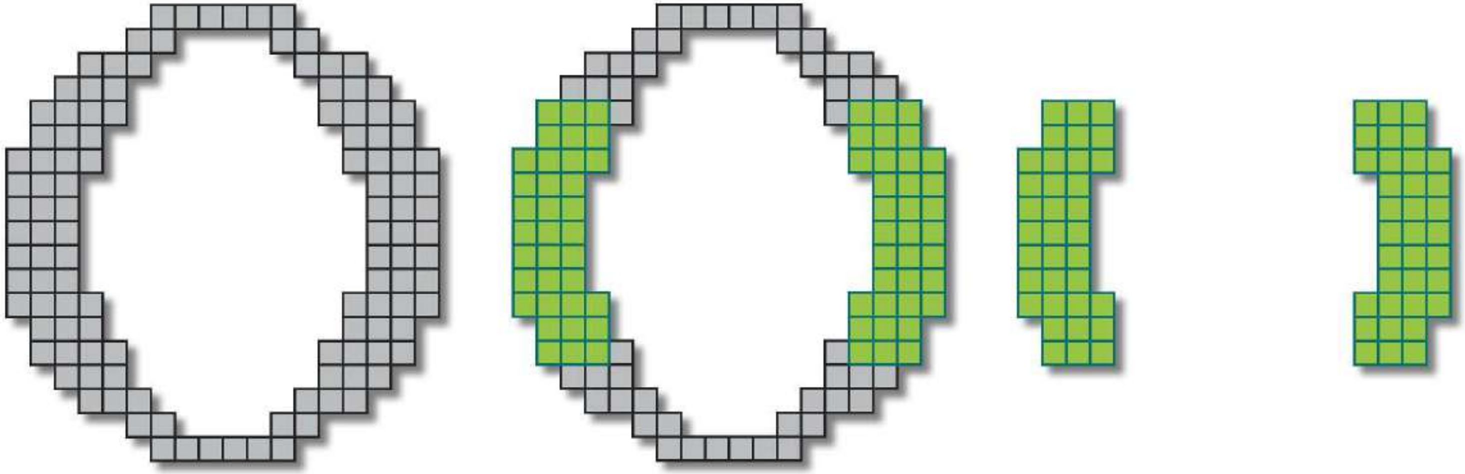


FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Opening



Opening

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	1				

Binary image A

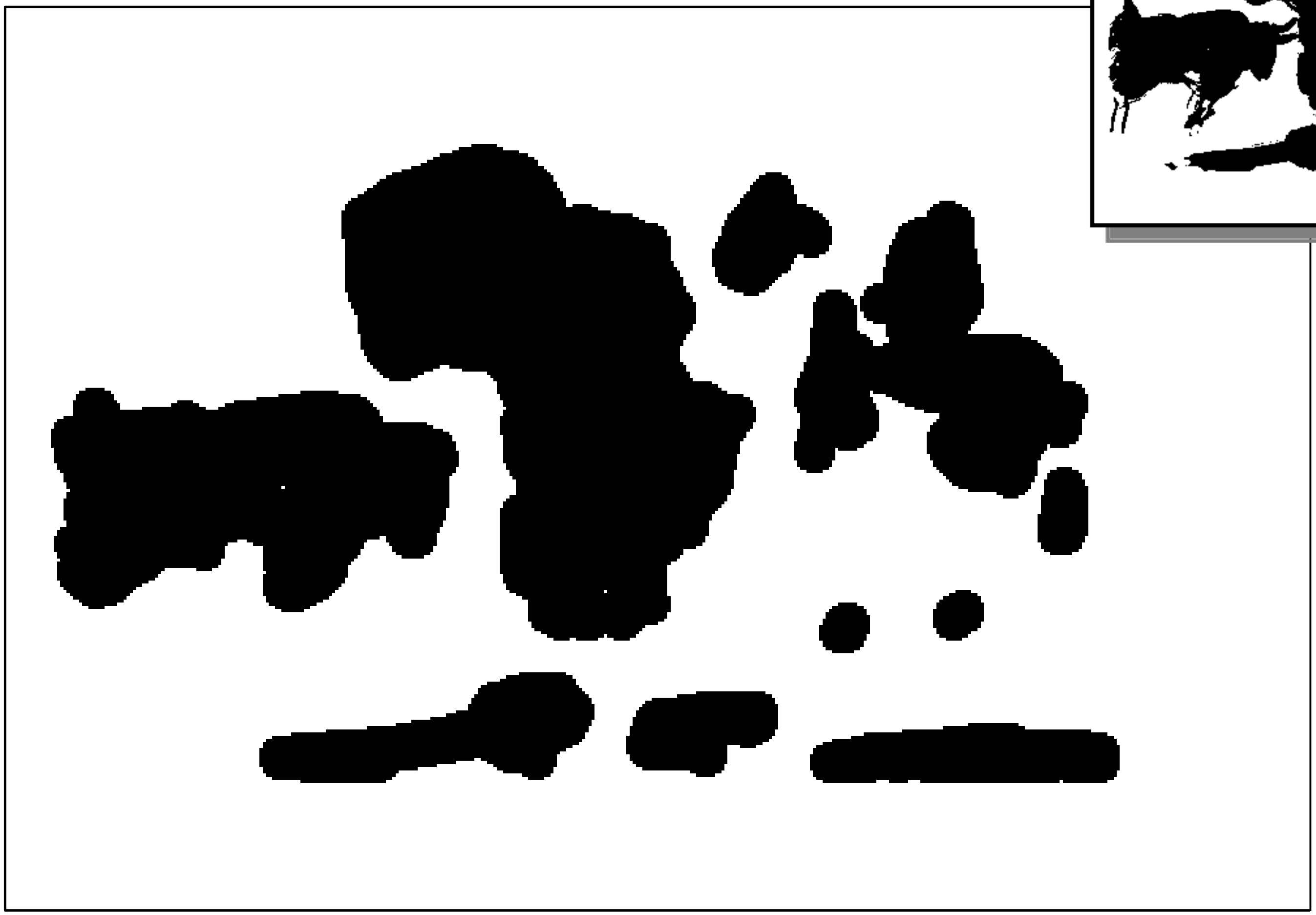
1	1	1
1	1	1
1	1	1

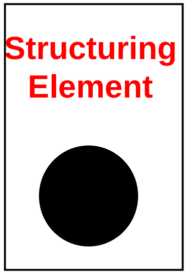
Structuring element B

			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	

Opening result

Opening

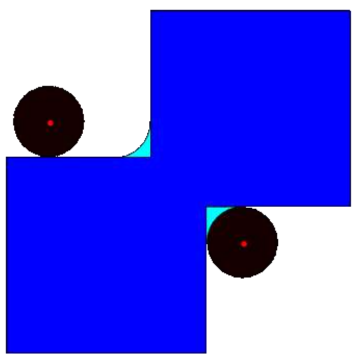




Pablo Picasso, *Pass with the Cape*, 1960

Adapted from John Goutsias, Johns Hopkins Univ.

Closing



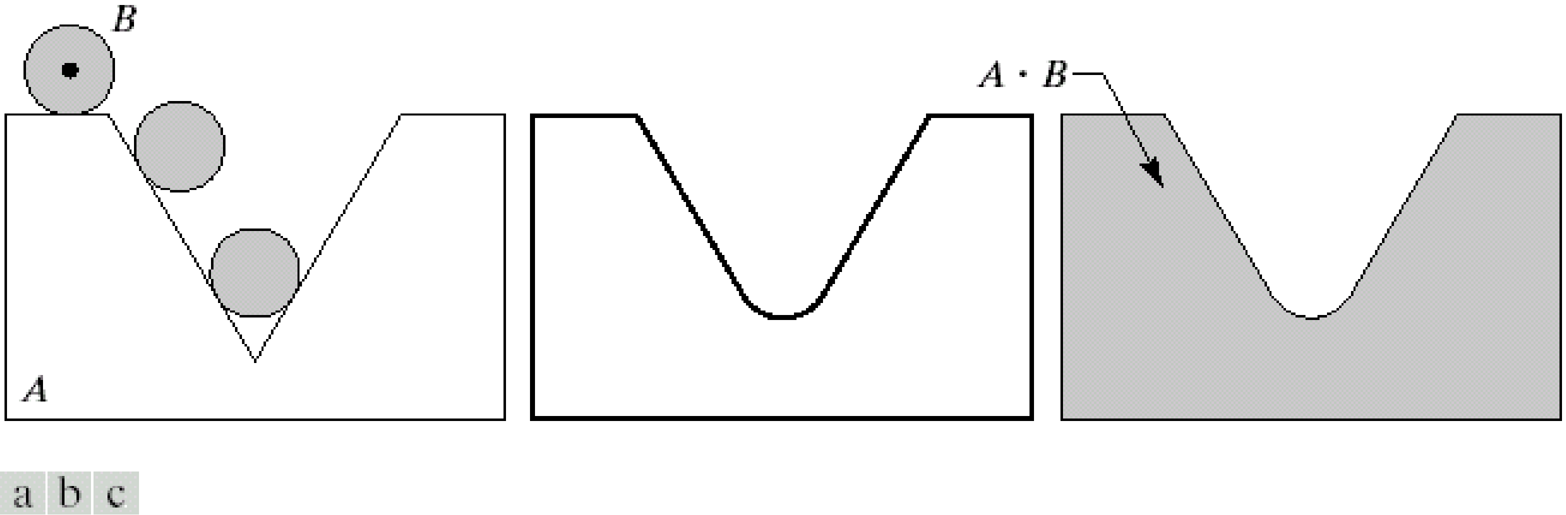
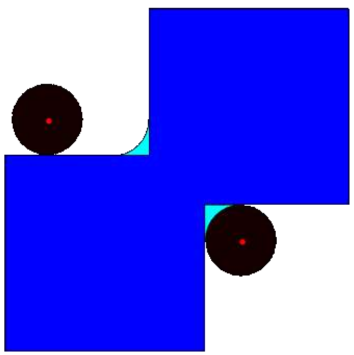
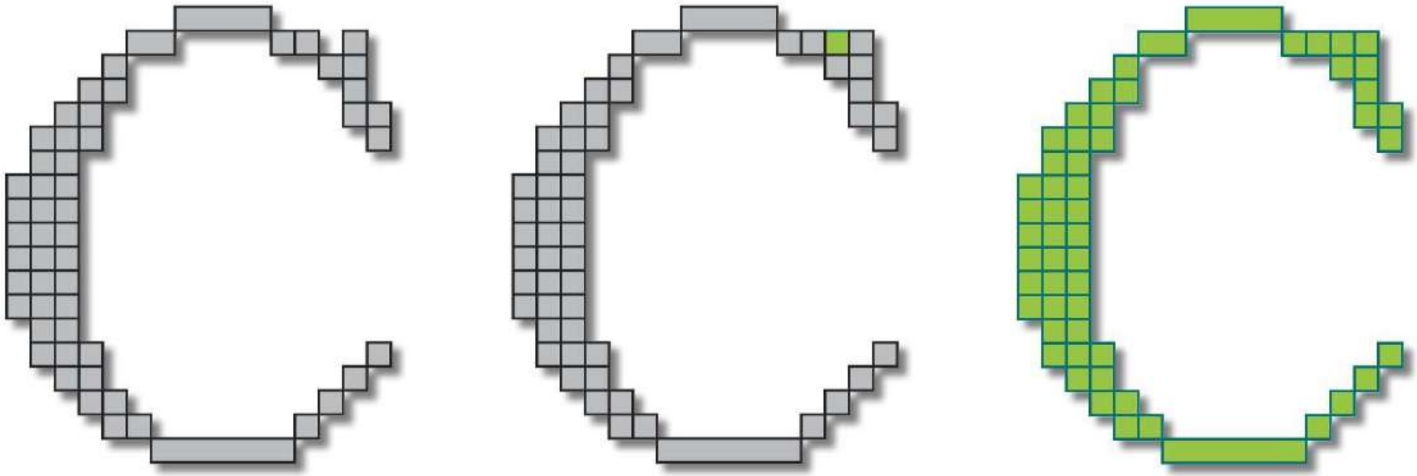


FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

Closing





DLR Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft

Closing

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	1				

Binary image A

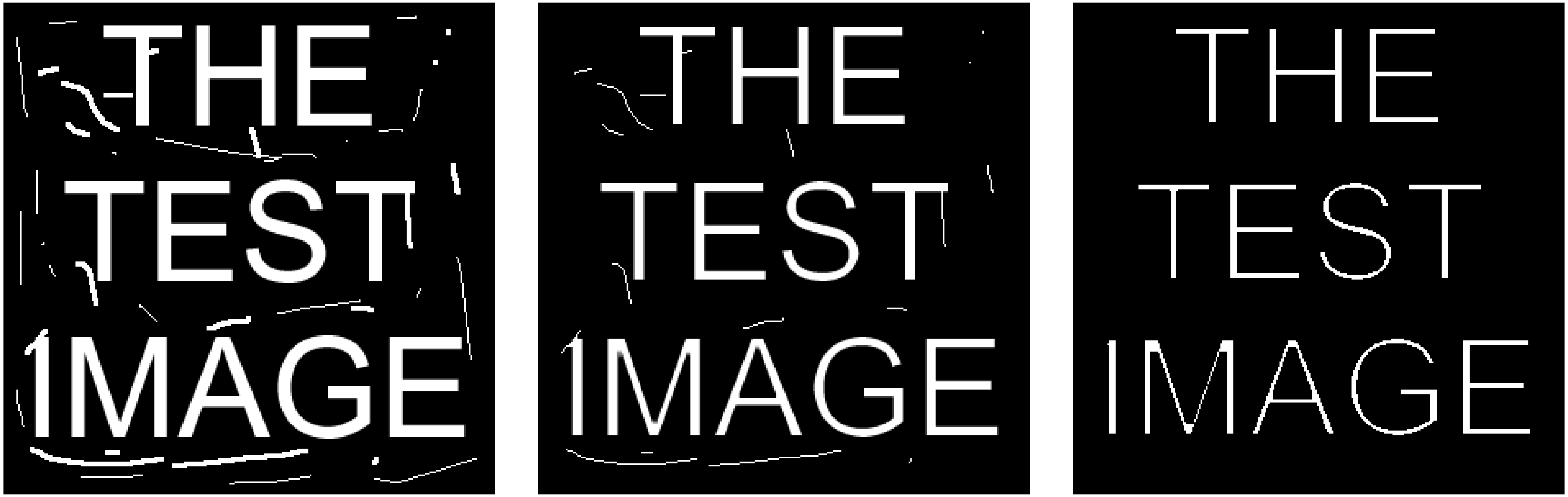
1	1	1
1	1	1
1	1	1

Structuring element B

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	1	1	
		1	1				

Closing result

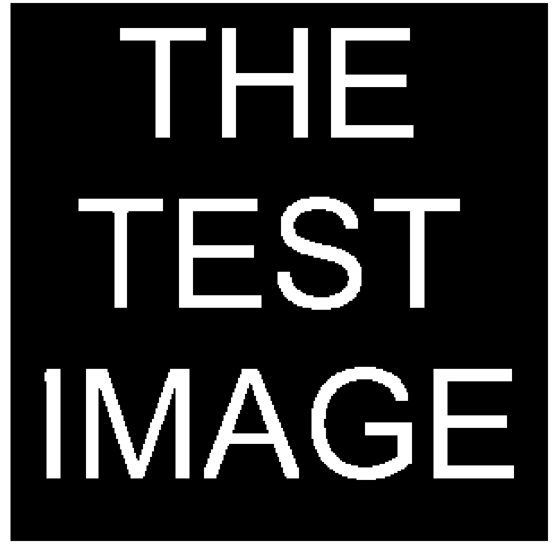
Examples



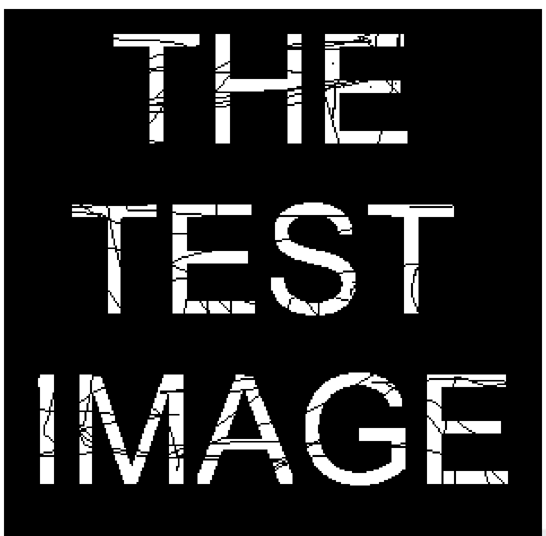
Original image Eroded once Eroded twice

Examples

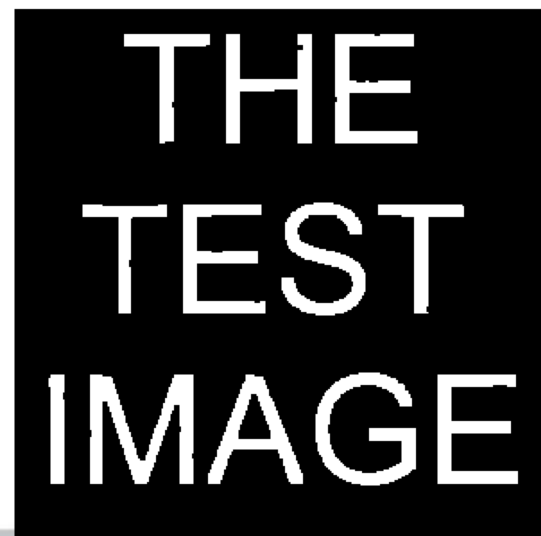
Original image



Opened twice



Original image

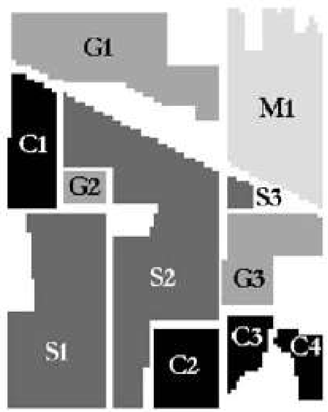


Closed once

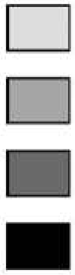
Examples: Remote Sensing

Original Data

Hyperspectral Image (1 band)



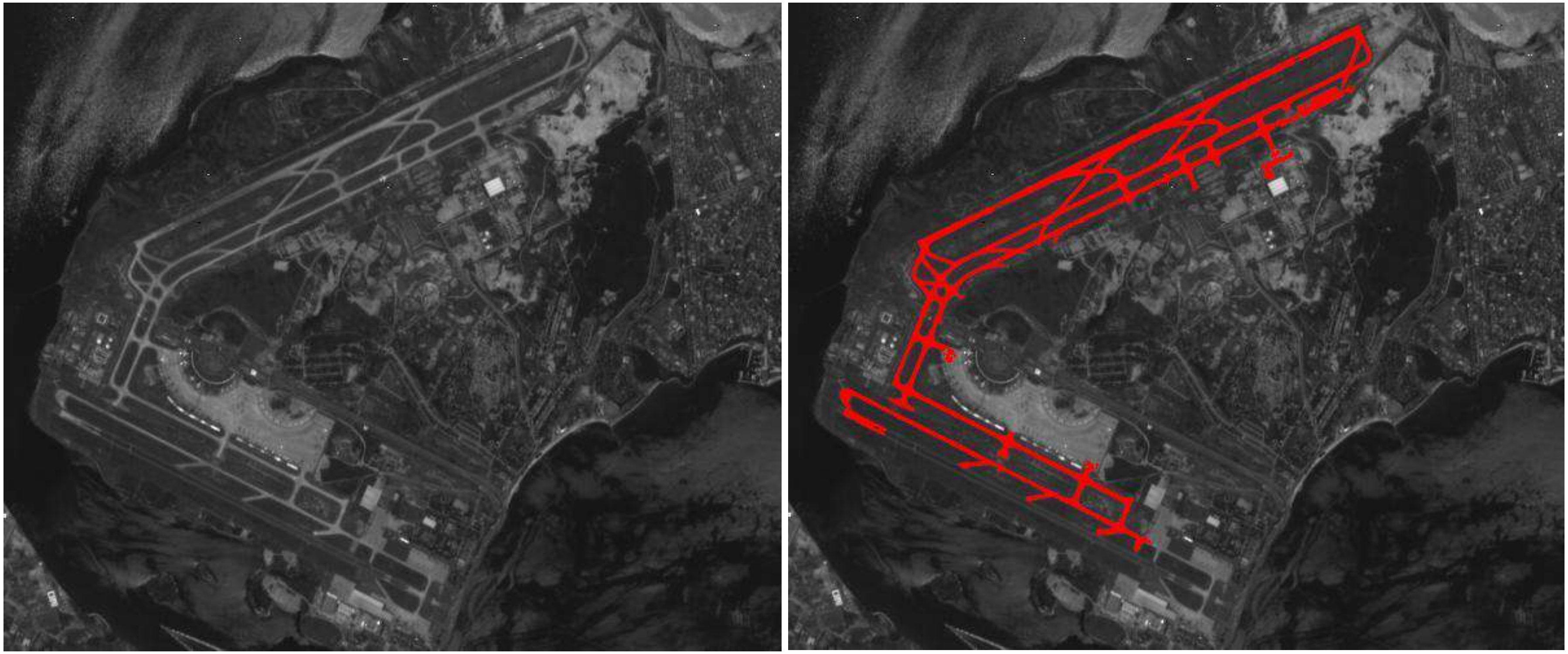
Ground Truth



Classification

After morphological filtering

Adapted from Plaza et al.



Detecting runways in satellite airport imagery

http://www.mmorph.com/mxmorph/html/mmdemos/mmdairport.html

Material from...

- Richard Alan Peters, university of Vanderbild (slides)
- Eamonn Keogh, university of California (slides)
- Digital Image Processing, Gonzalez & Woods (book)
- Selim Aksoy, university of Istanbul (slides)
- Daniele Cerra, DLR

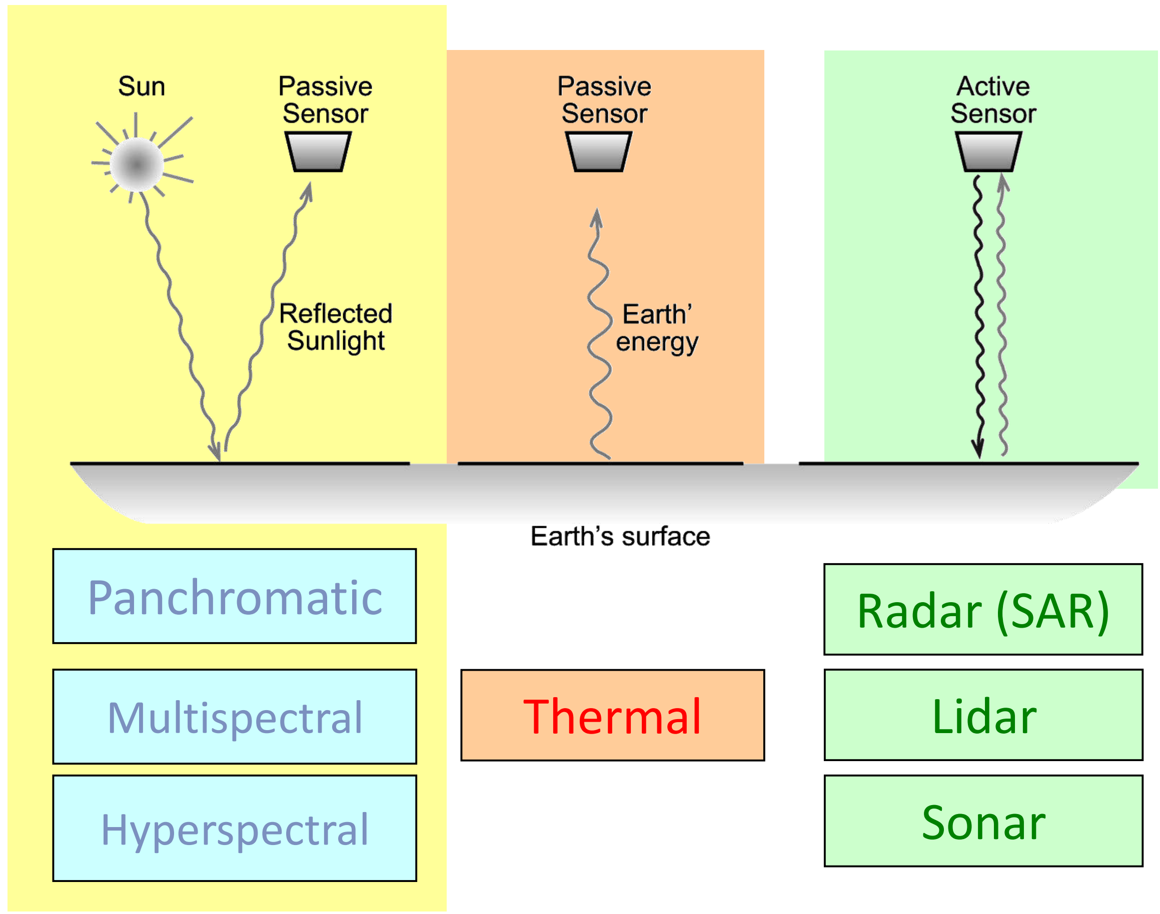
Hyperspectral Remote Sensing

Basic Principles

Daniele Cerra, German Aerospace Center (DLR)

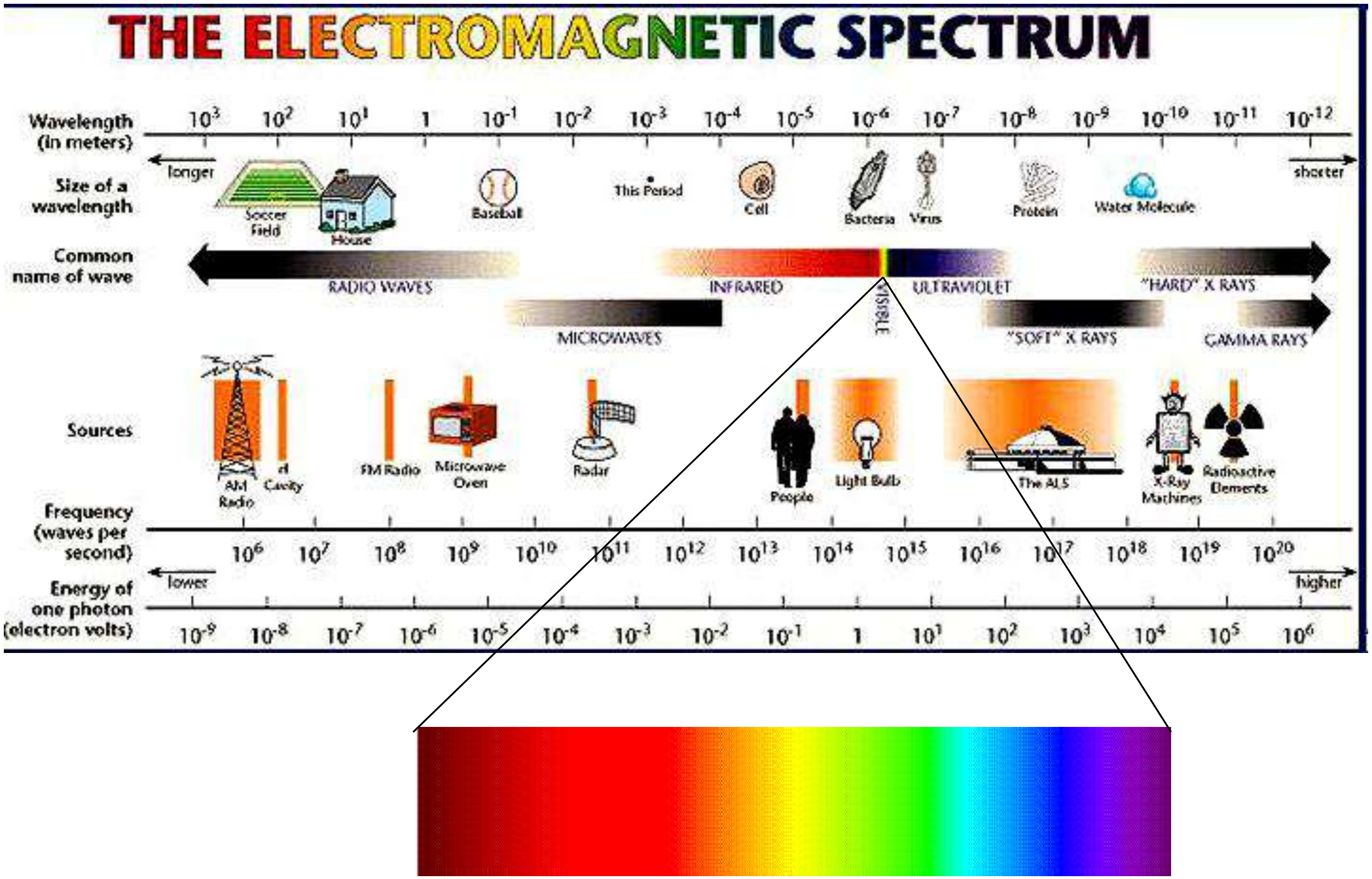
DLR

Sensors in Remote Sensing

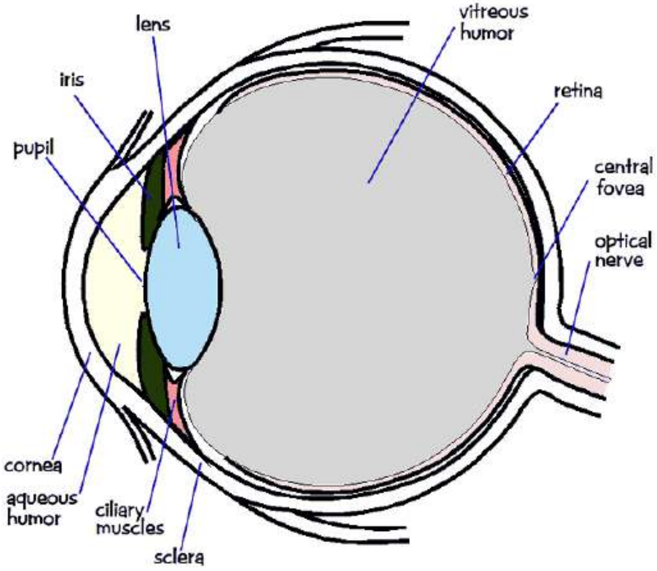


DLR

Electromagnetic (EM) spectrum



Human visual system



– Color perception

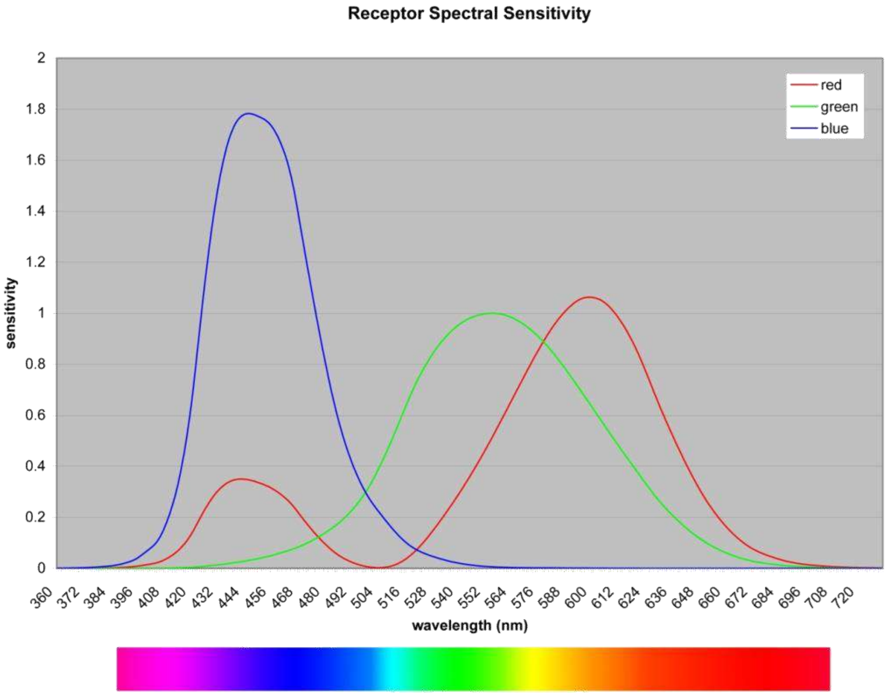
  – Light hits the retina, which contains photosensitive cells.

  – These cells convert the spectrum into a few discrete values.

Adapted from Steve Seitz, U of Washington

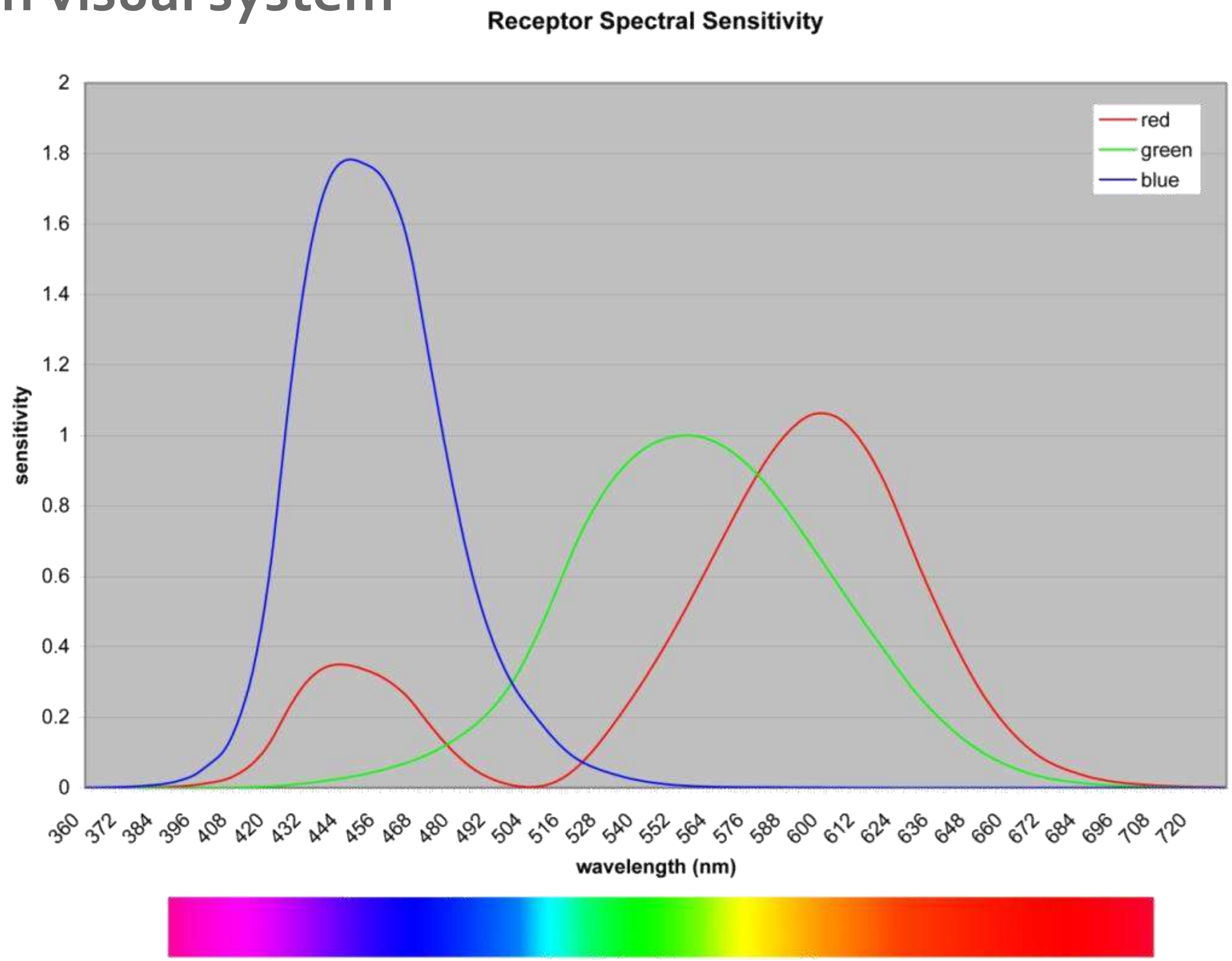
Human visual system

- Two types of photosensitive cells:
  - Cones
    - Sensitive to colored light, but not very sensitive to dim light
  - Rods
    - Sensitive to achromatic light
- We perceive color using three different types of cones.
  - Each one is sensitive in a different region of the spectrum.
    - 440 nm (BLUE)
    - 545 nm (GREEN)
    - 580 nm (RED)
  - They have different sensitivities



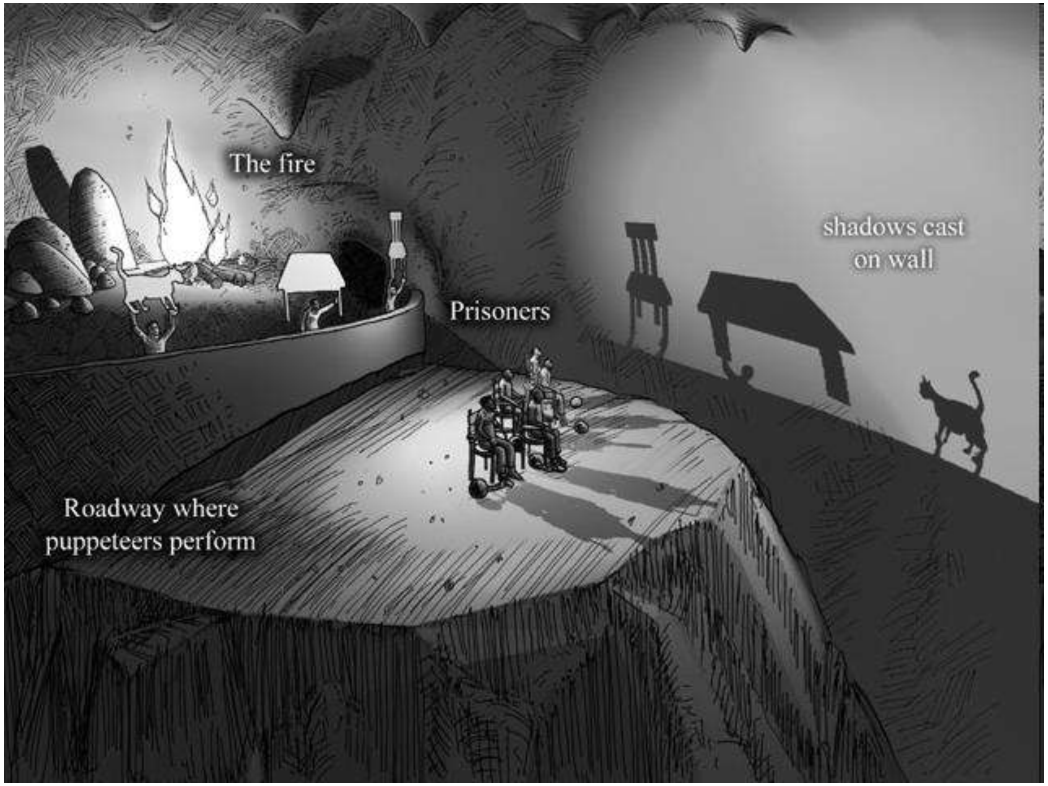
Adapted from Gonzalez & Woods

Human visual system



Adapted from Gonzalez & Woods

Can you trust your senses?

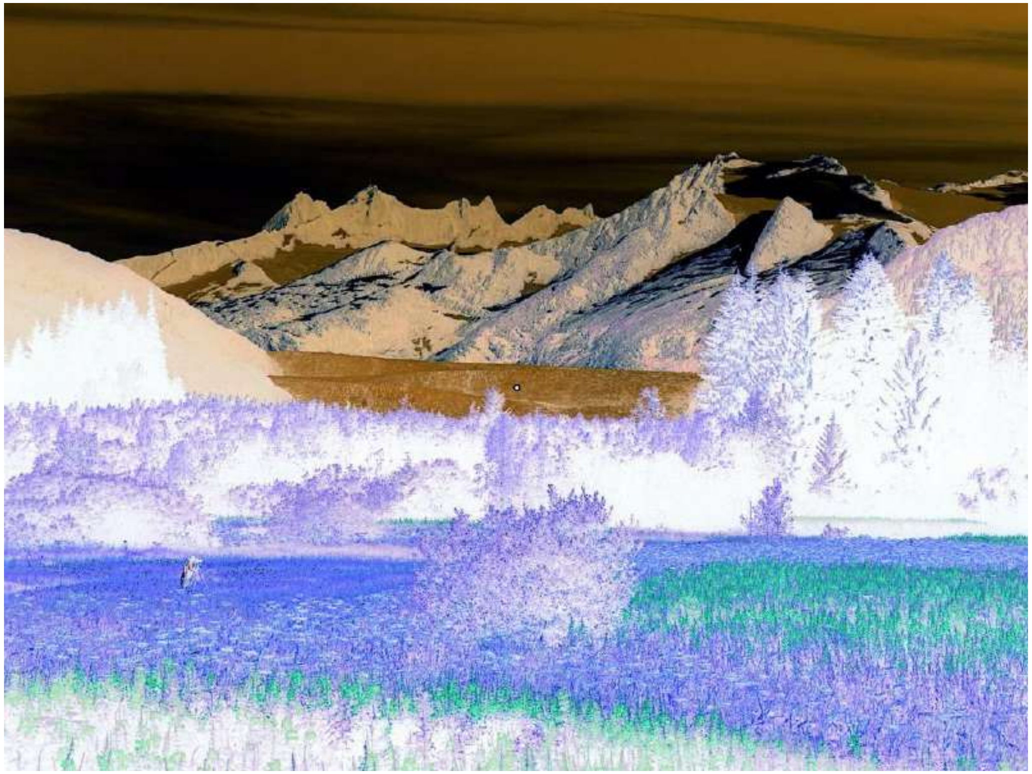


– Plato´s Myth of the Cave

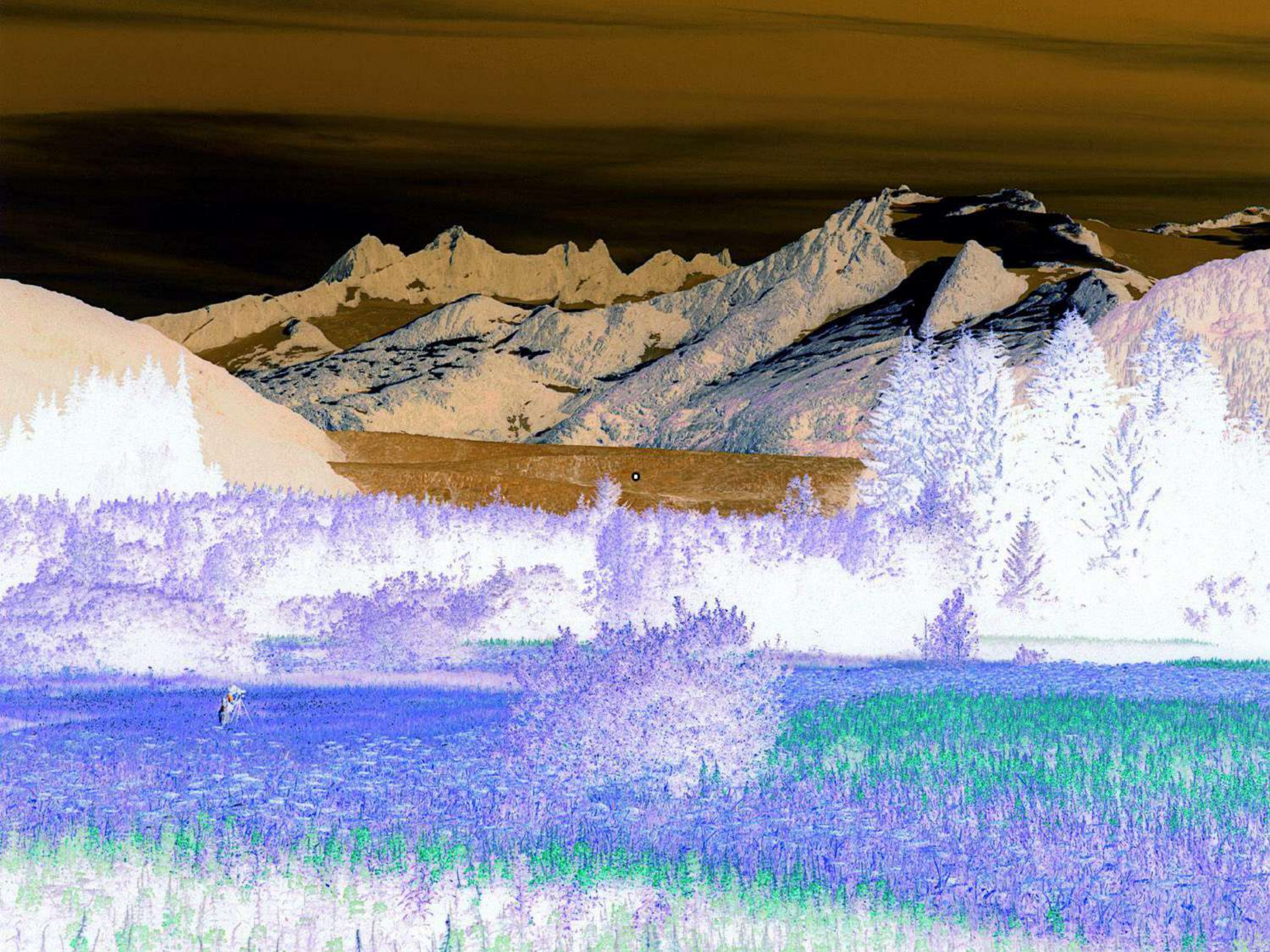
– What we see with our eyes is our „perception" of reality

Can you trust your senses?

Color Perception: The Afterimage Effect



Stare at the dot in the center of the image

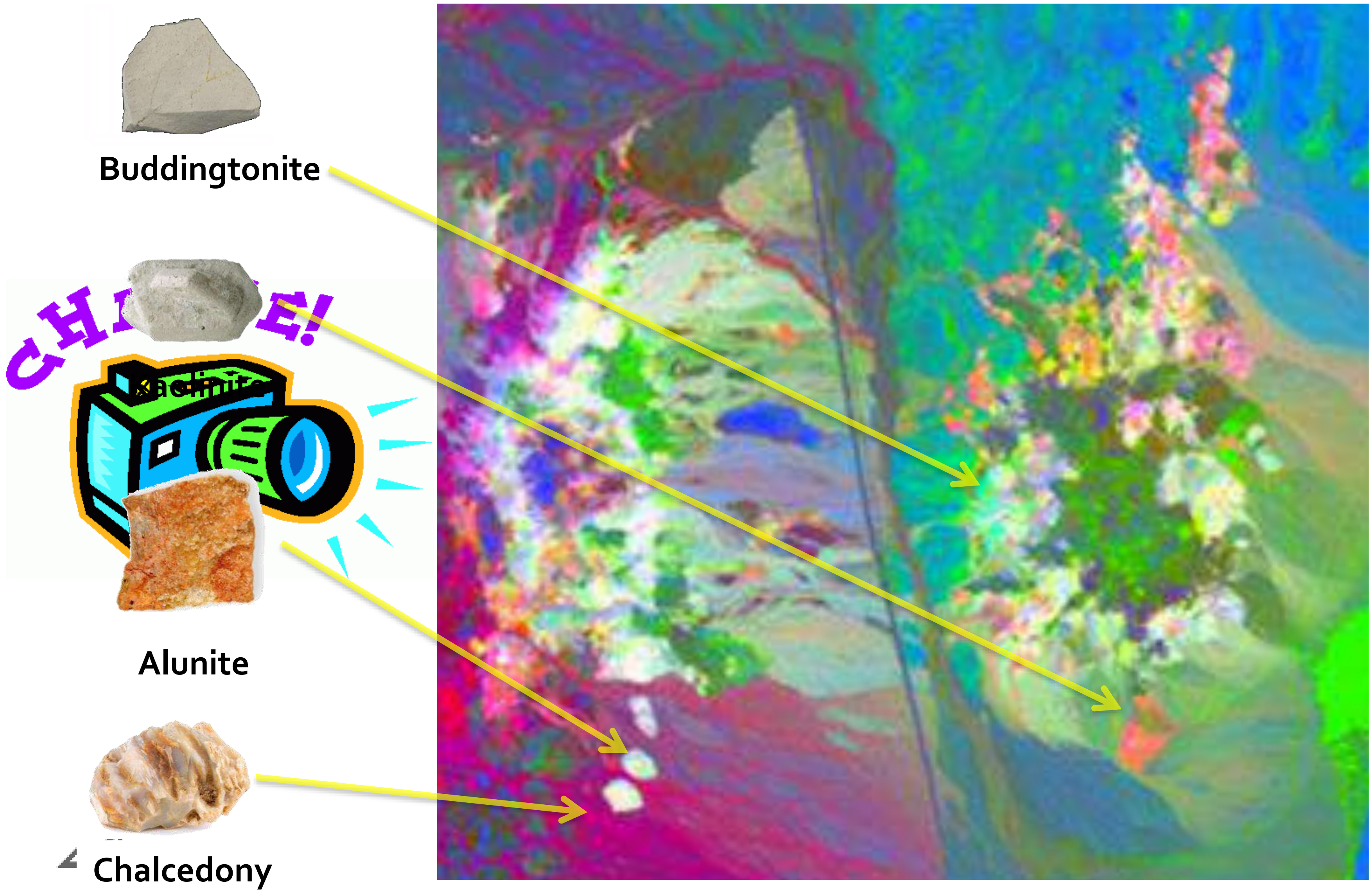


Color Perception: The Afterimage Effect

1. The color "negatives" saturate the local receptors
2. When the color is removed these receptors are "mute"
3. The gray tones only have contributions from the agonist (opposite) colors
   - Like the recoil after a gunshot

What is "real" is NOT only what we can see with our eyes!

Hyperspectral



Basic Principle

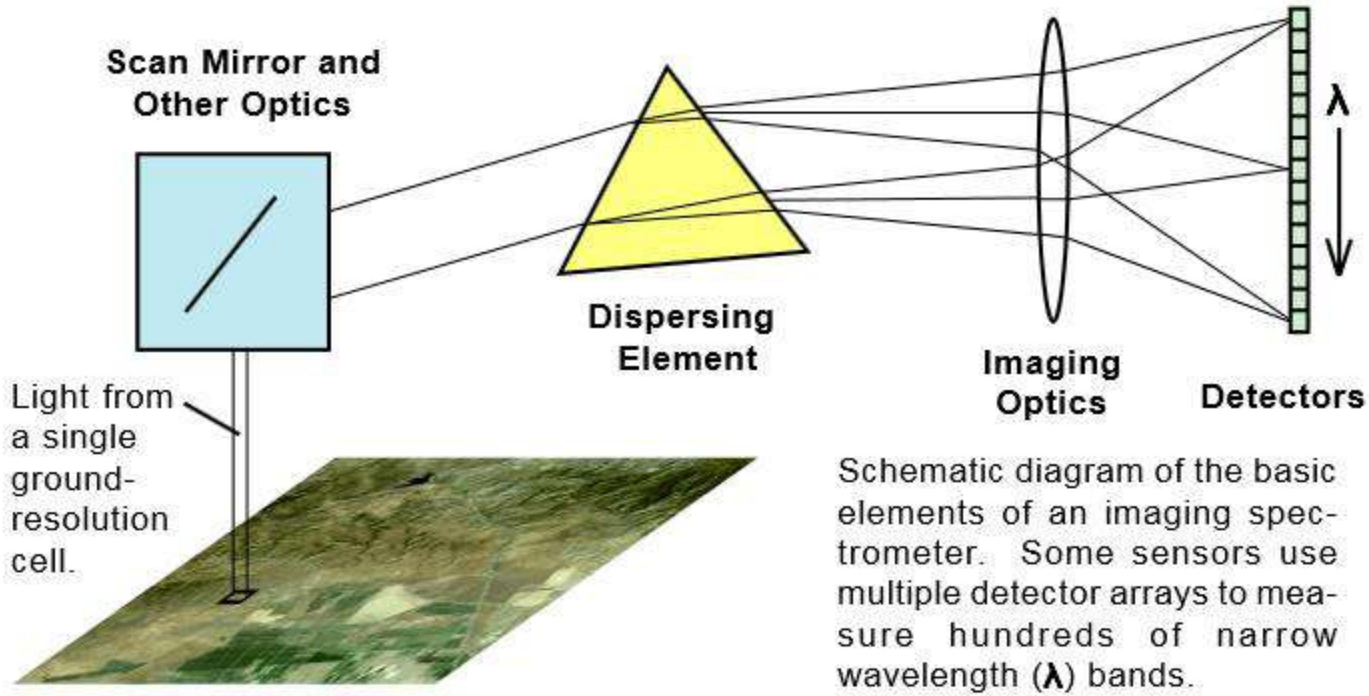
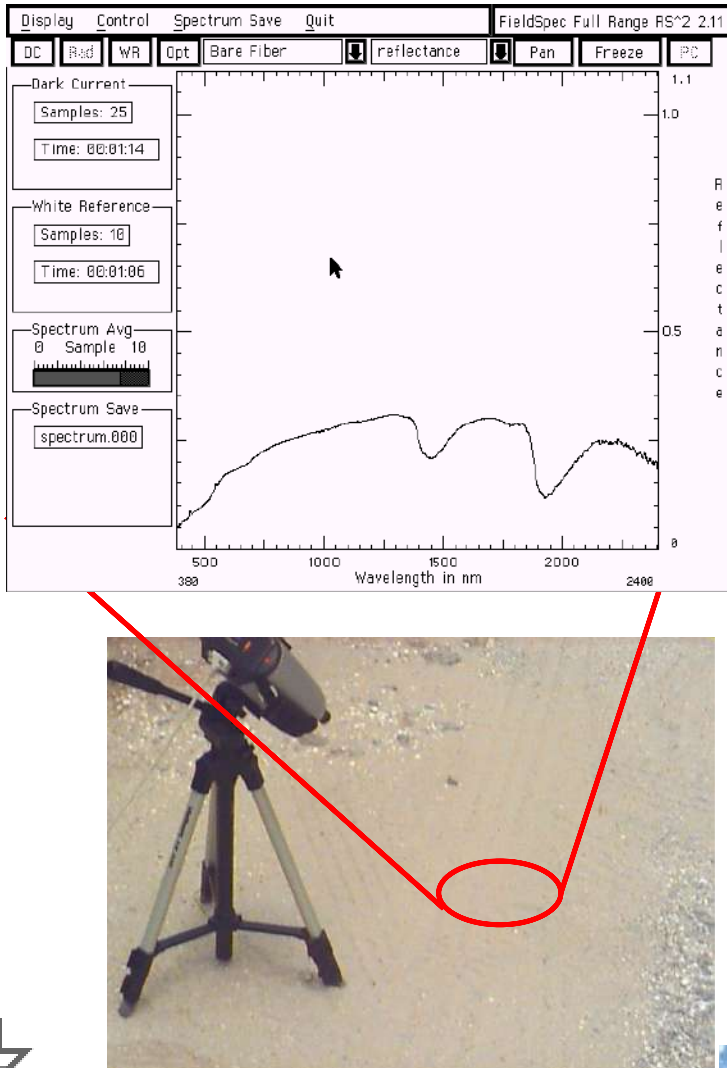
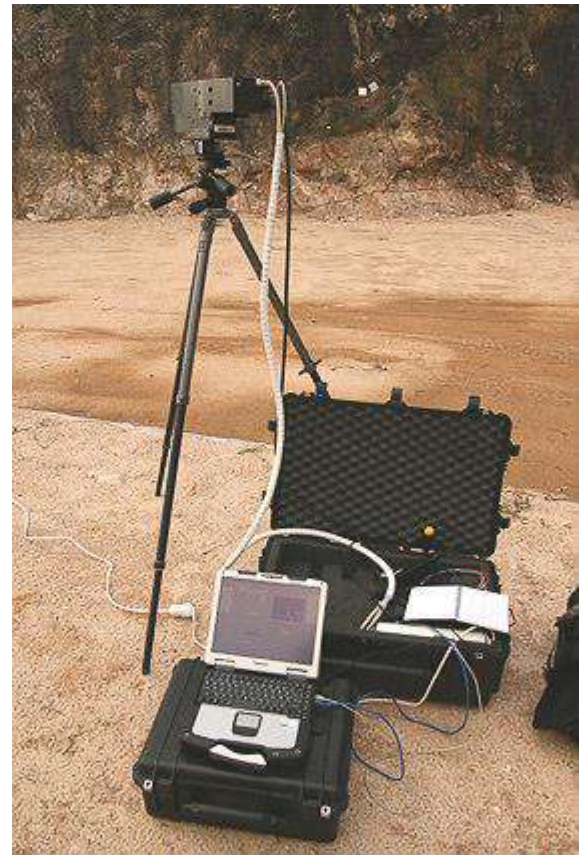


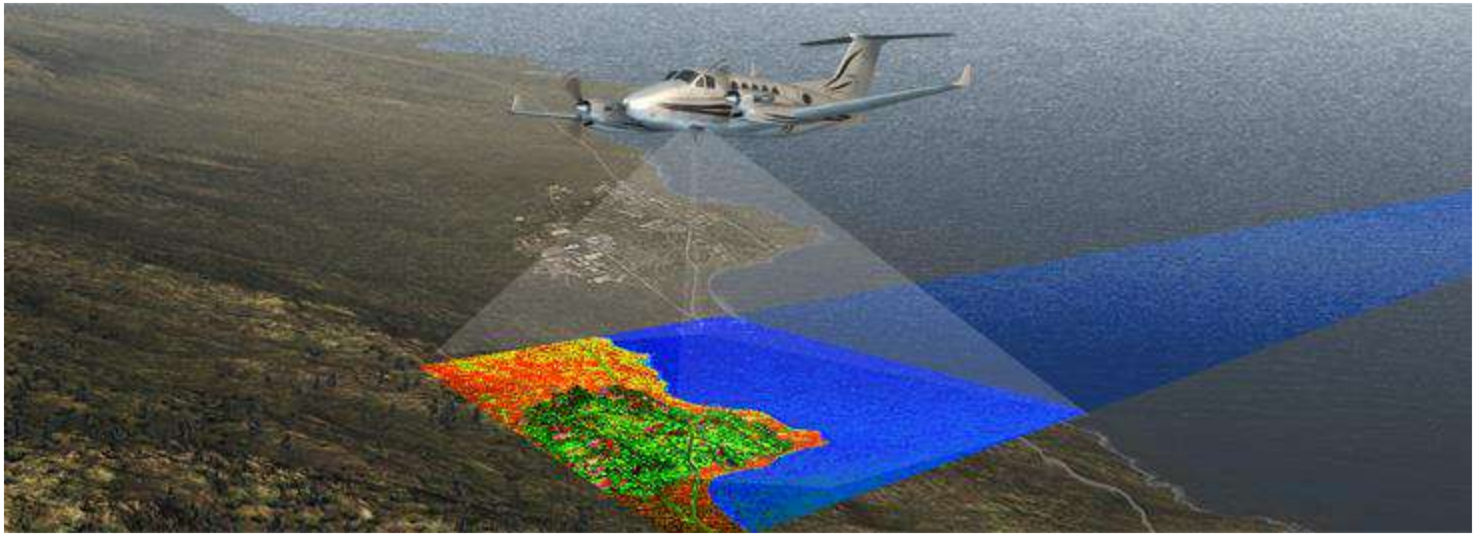
Image by TNTmips

Acquisition Systems

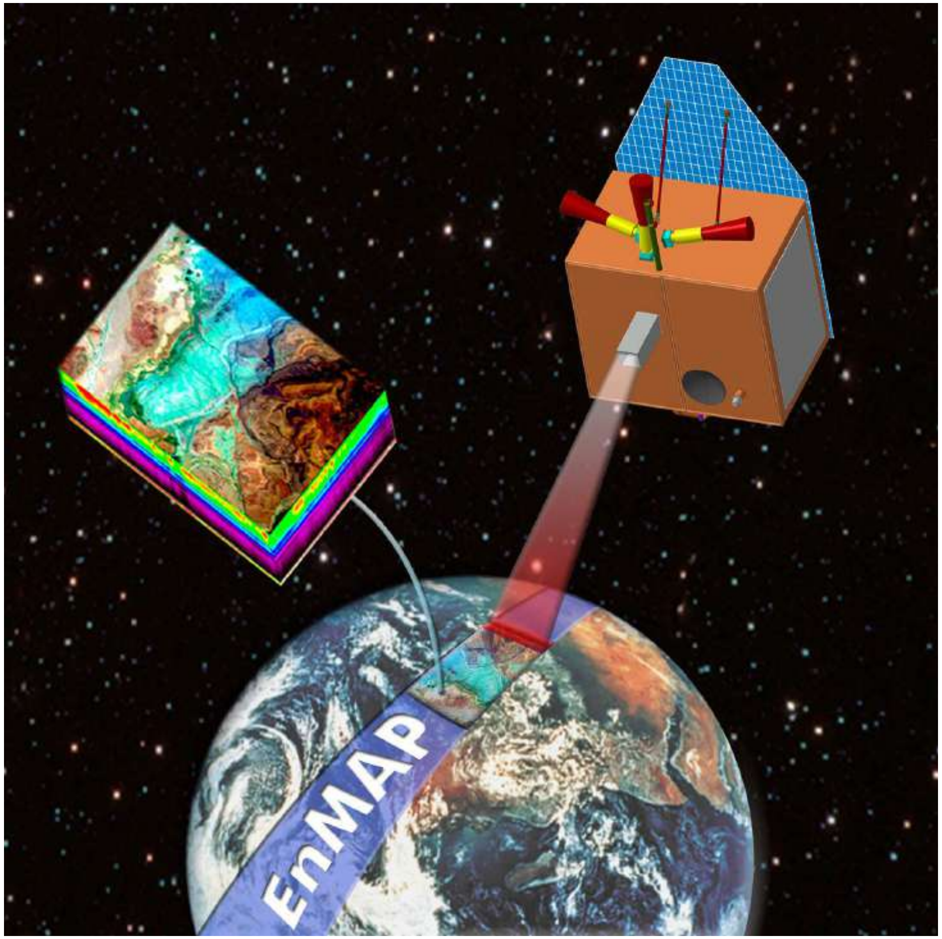




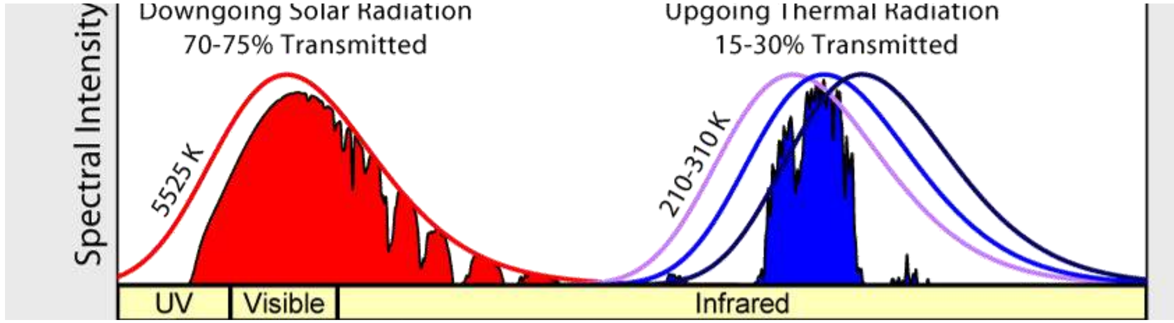
Acquisition Systems



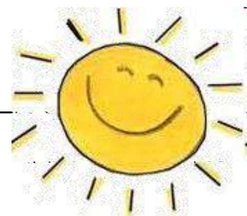
Acquisition Systems

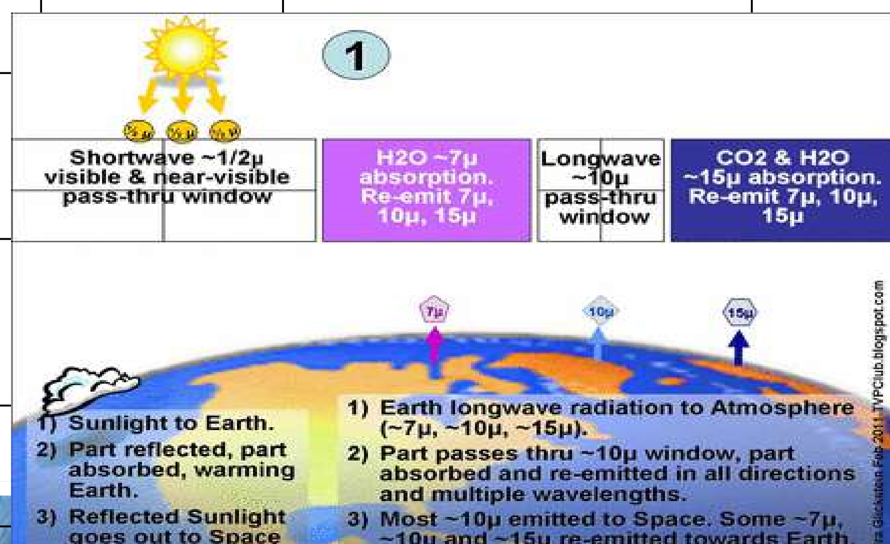


Radiation transmitted by the atmosphere



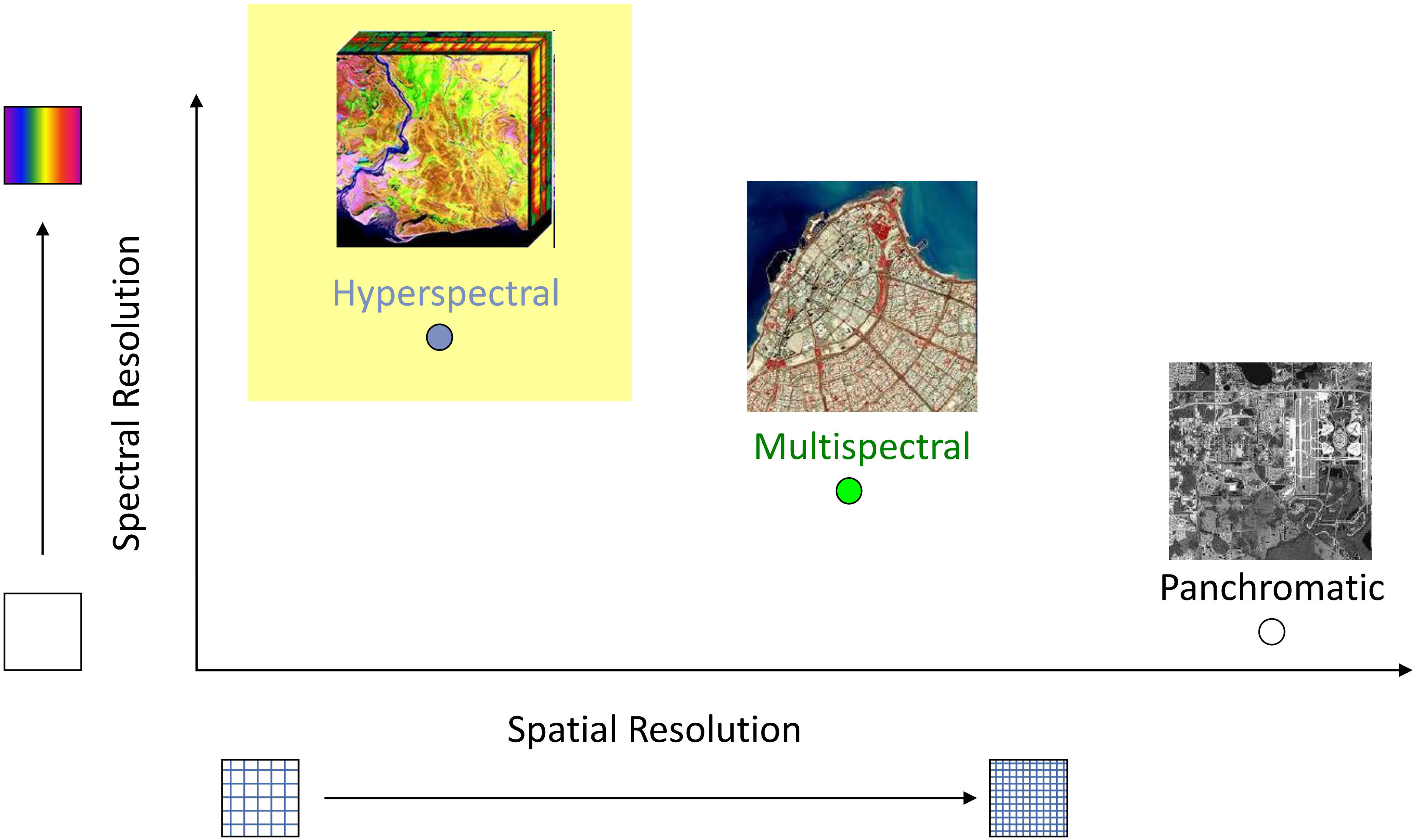
Sensor Type
Panchromatic
Multispectral
Hyperspectral
Thermal (HS)



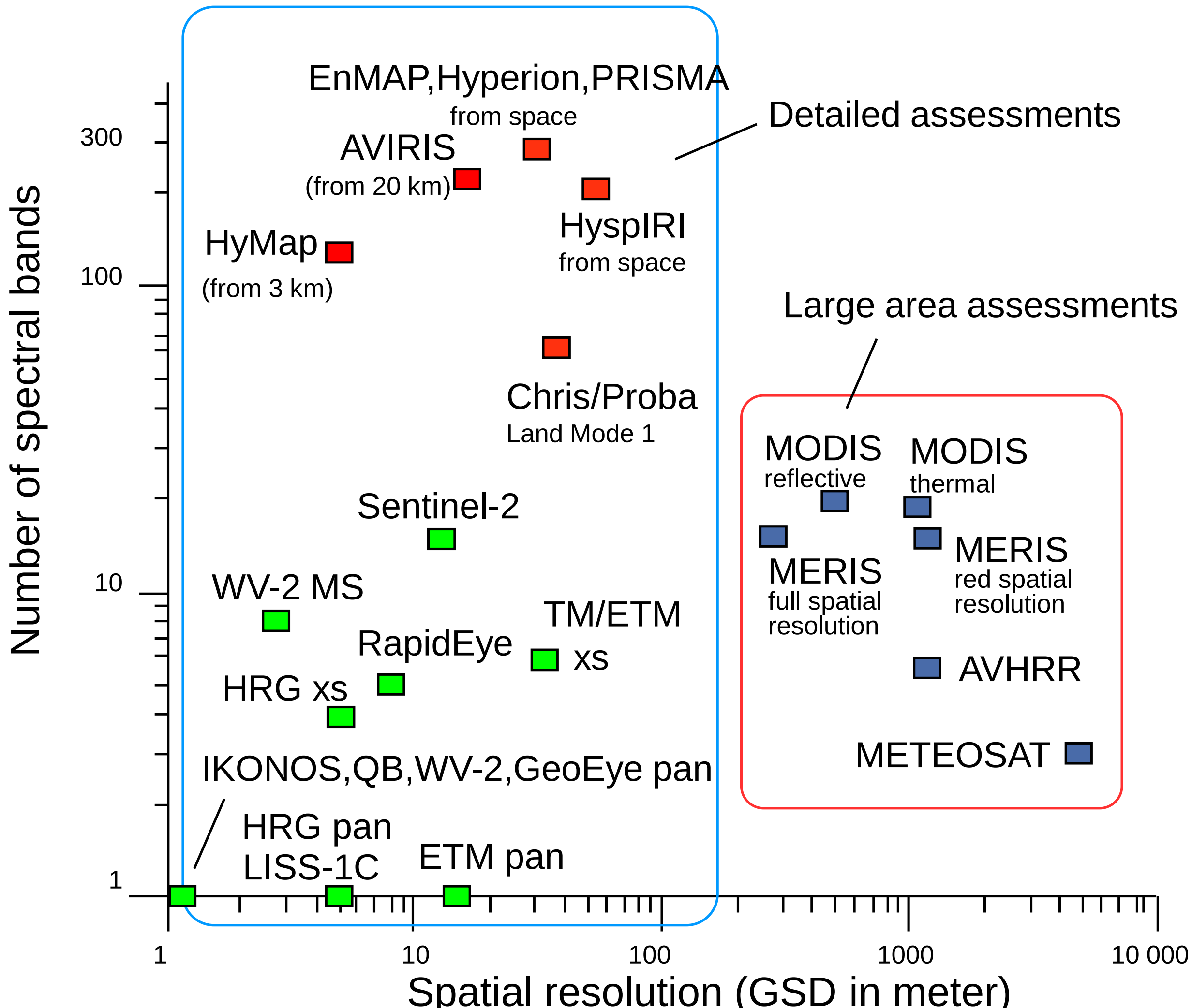


DLR

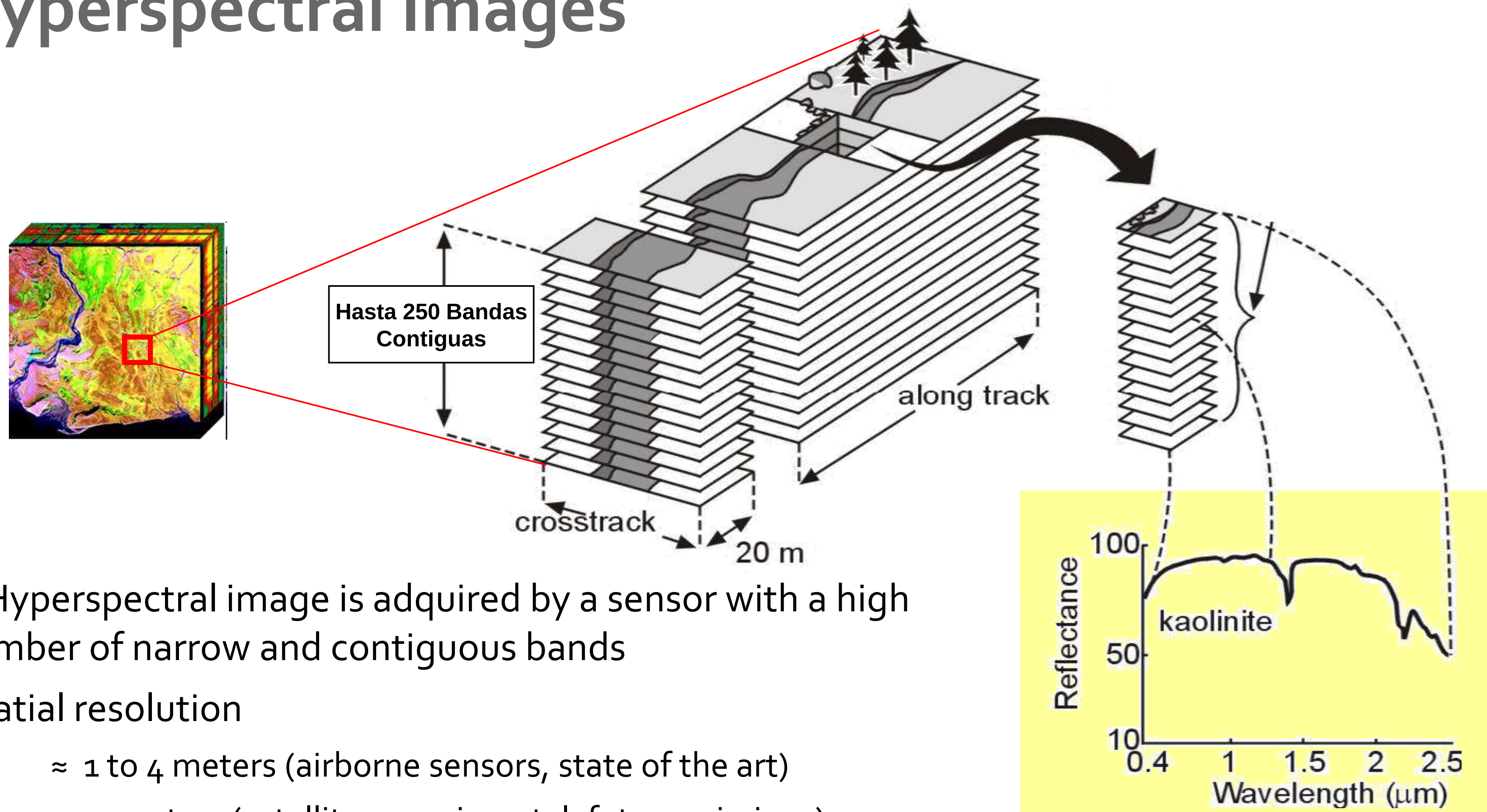
Optical Passive Sensors in Remote Sensing



Context of Optical EO Systems



Hyperspectral Images



- A Hyperspectral image is adquired by a sensor with a high number of narrow and contiguous bands
- Spatial resolution
  - ≈ 1 to 4 meters (airborne sensors, state of the art)
  - ≈ 30 meters (satellites, sperimental, future missions)
- Spectral range: usually 0.4 – 2.5 micrometers (µm)
- Each pixel has a characteristic spectrum
  - In this example it is related to a mineral (kaolinite) →

The spectrum of a pixel is represented by its values across all bands

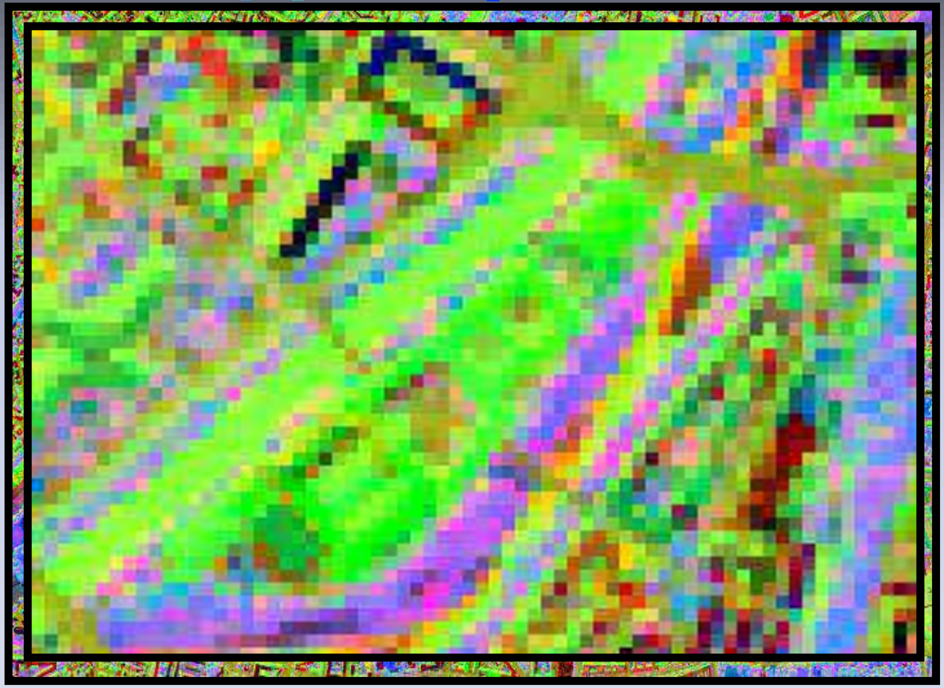
Why are spatial and ***spectral*** resolution inversely proportional?

In the city..

Zoom in!

Panchromatic

Hyperspectral



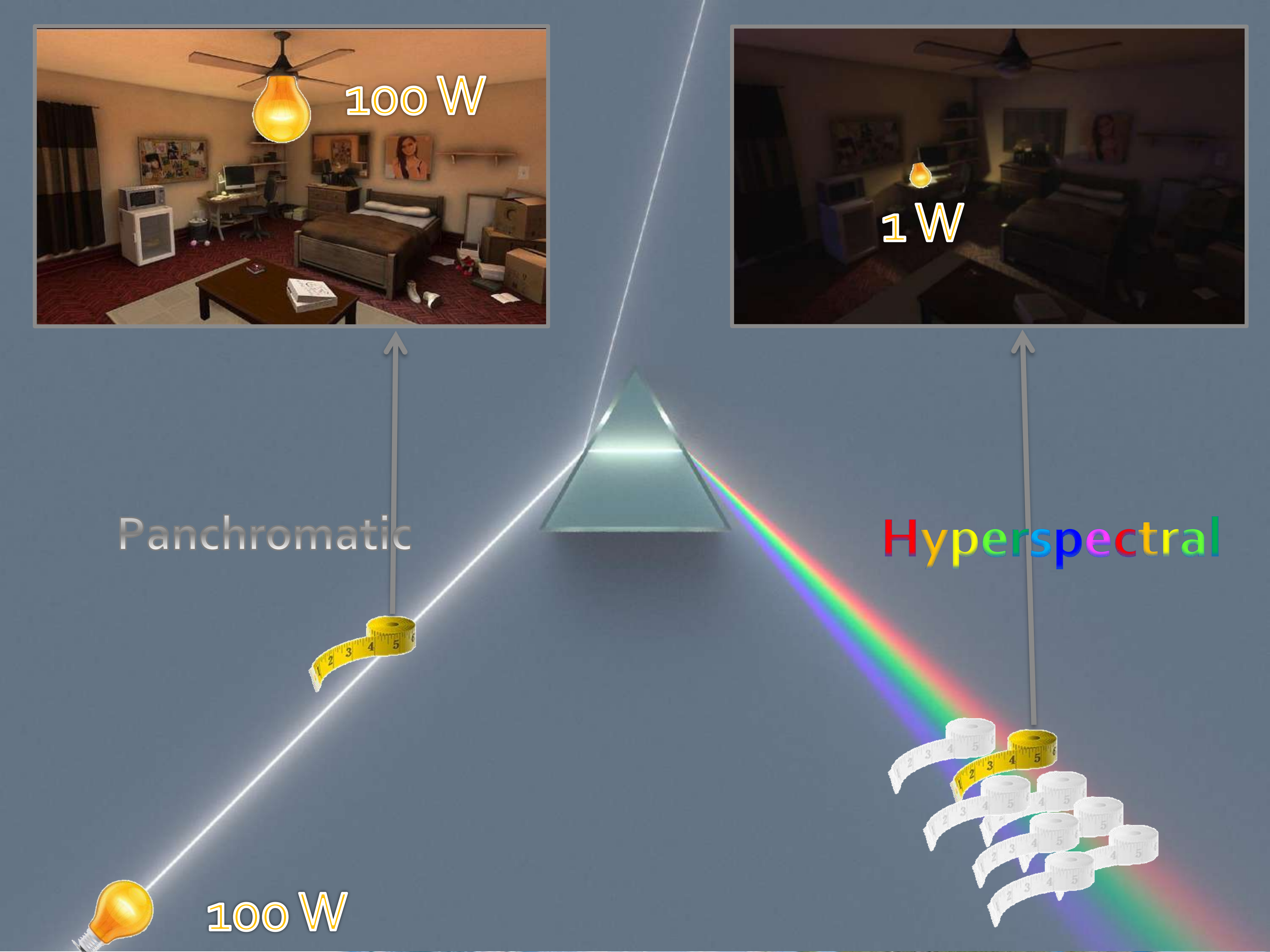

100 W
1 W
Panchromatic
Hyperspectral
100 W

Image Correction

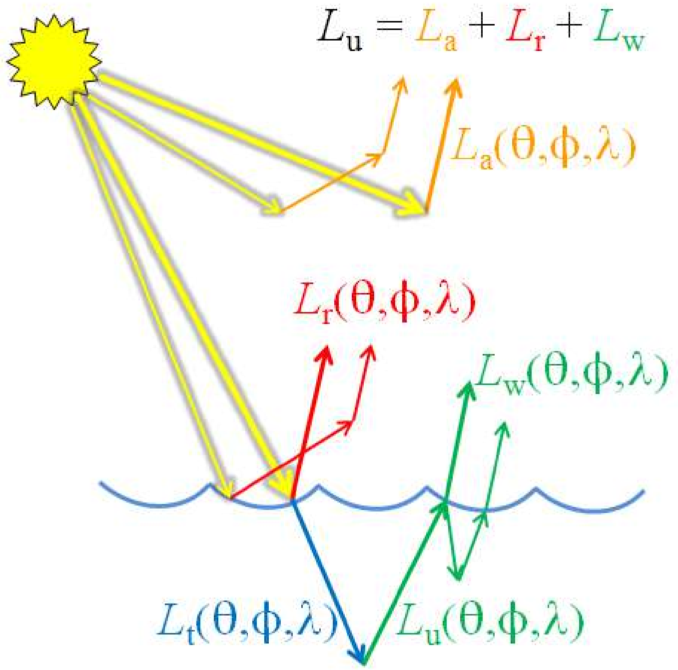
In the last episode..

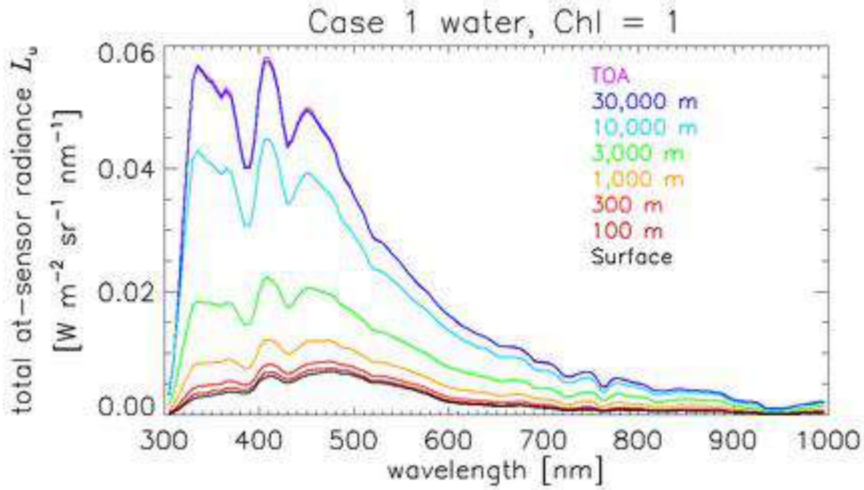
– Once our raw data are corrected, the image is formed and usually undergoes other correction steps....

  – Atmospheric Correction
  – Geometric Correction / Orthorectification..

Atmospheric Correction: Why is the sky blue?

– Atmospheric path radiance → *La*

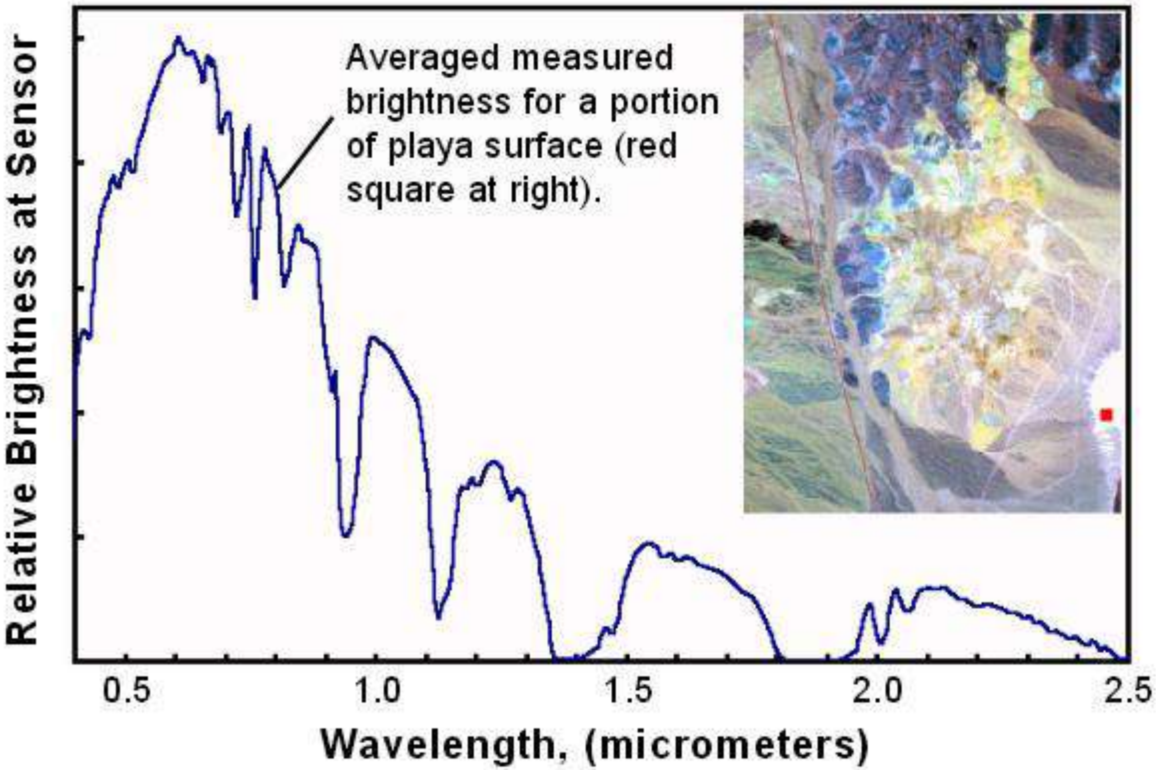




– Less important at long waves (infrared), more evident at short wavelengths

Figure by Ocean Optics

From radiance to reflectance



– If we want to know which fraction of the incoming solar energy is reflected by each band, we have to process the radiance values (amount of light/radiation measured in each band)

Figure by TNTmips

From radiance to reflectance

– The solar energy is not constant across all the bands! We must correct this

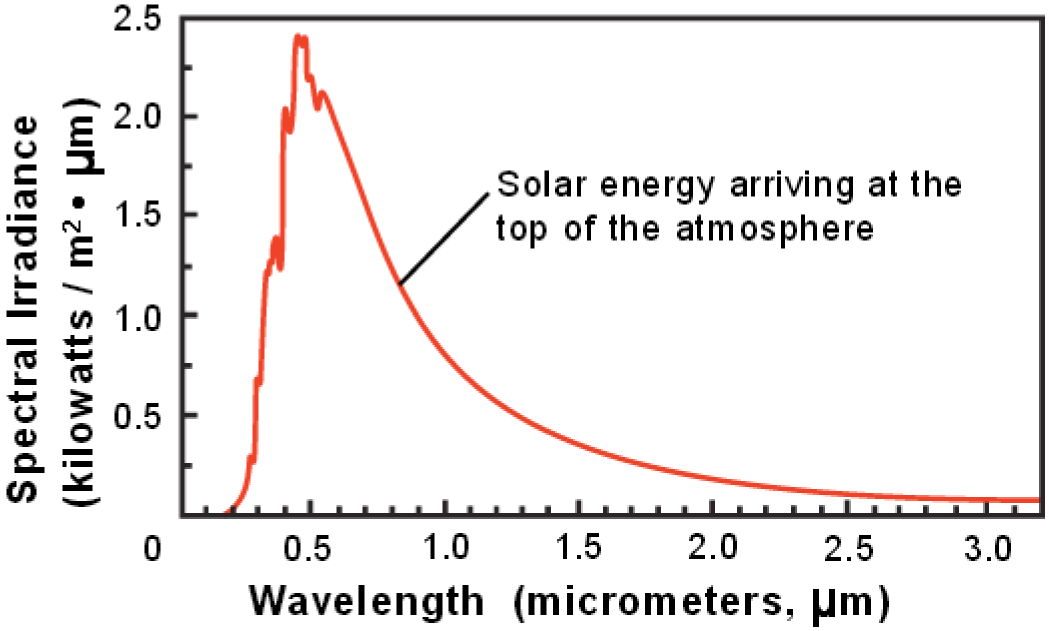
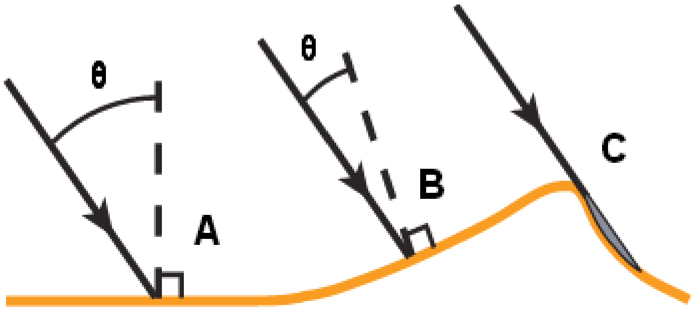


Figure by TNTmips

From radiance to reflectance

– Geometric effects / shadows

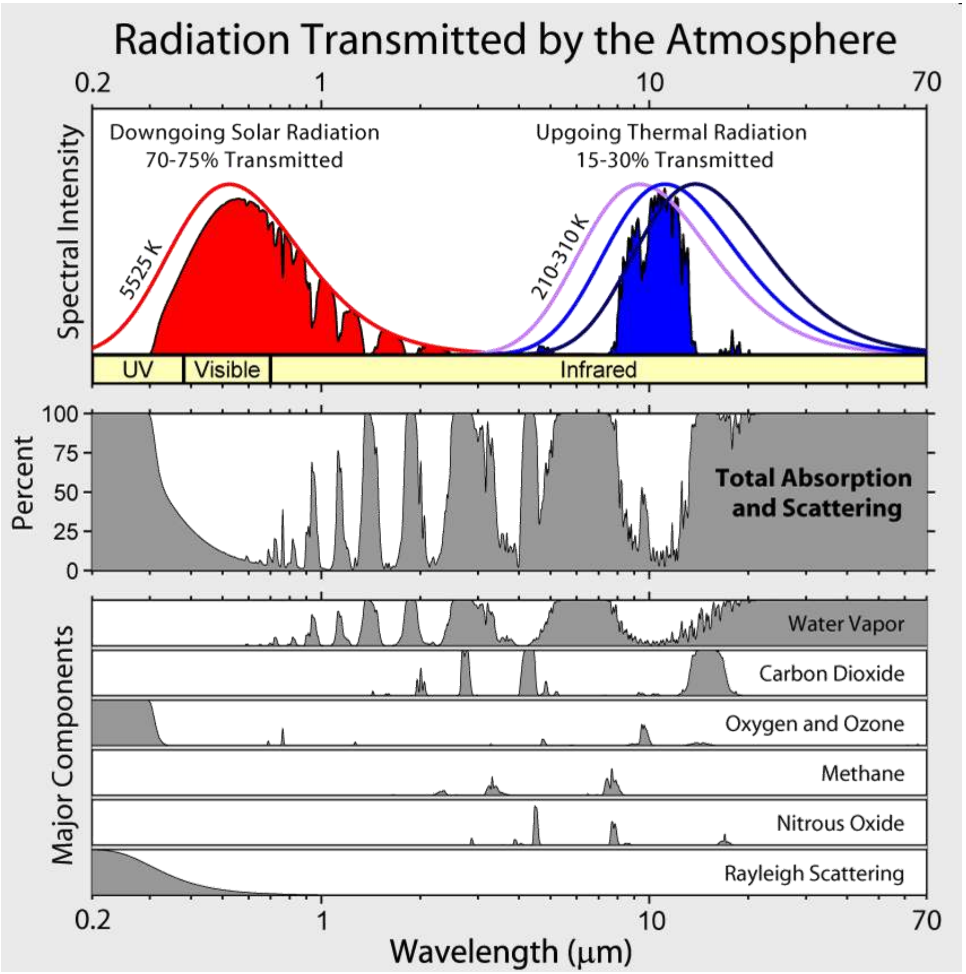


Illumination differences can arise from differing incidence angles (θ) as for **A** and **B**, or from shadowing (**C**).

Figure by TNTmips

From radiance to reflectance

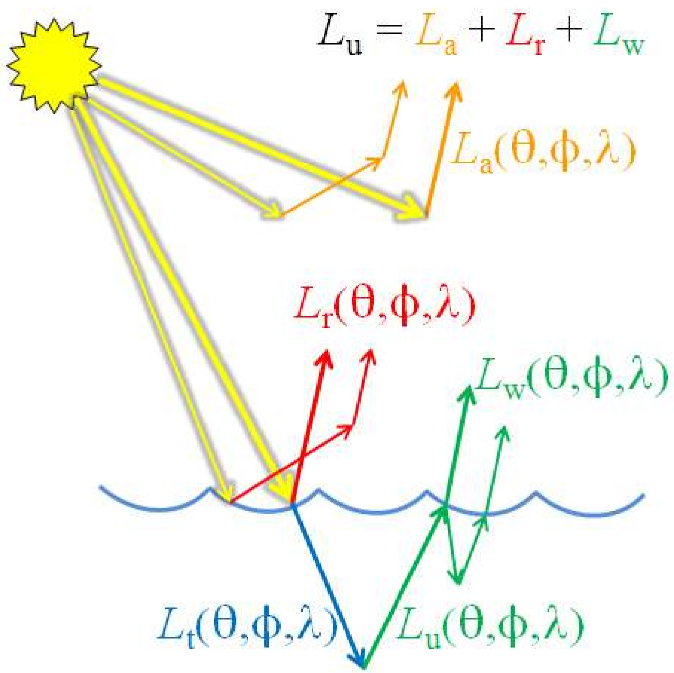
– Atmospheric effects

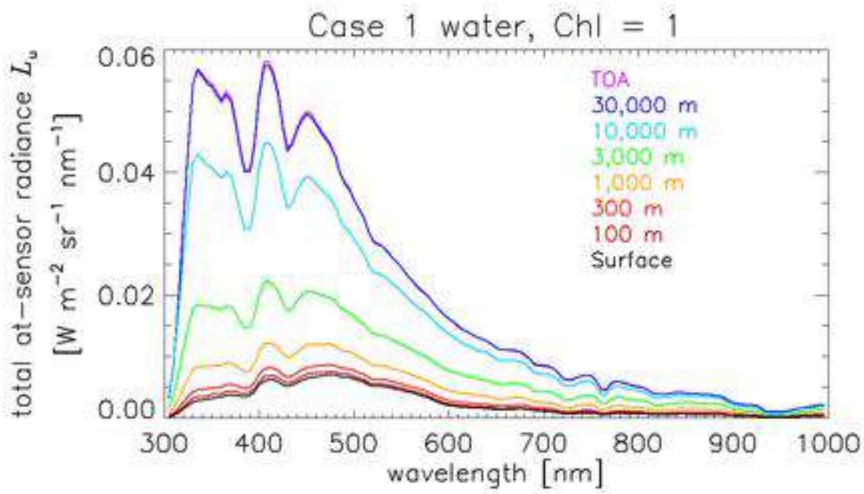


DLR

Why is the sky blue?

– Atmospheric path radiance → *La*

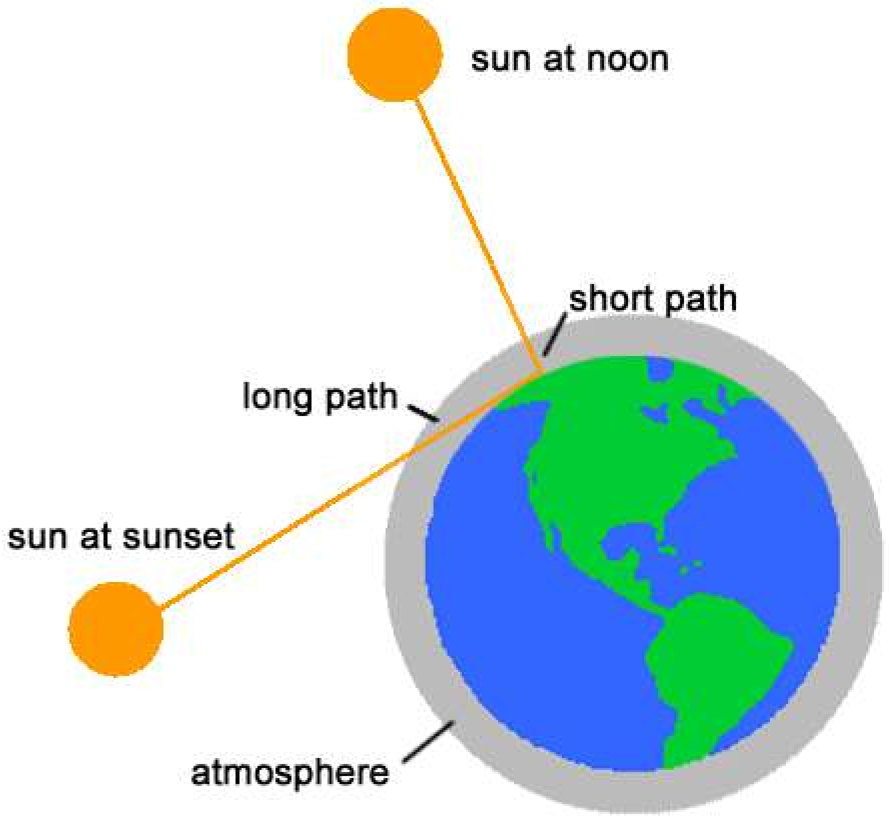




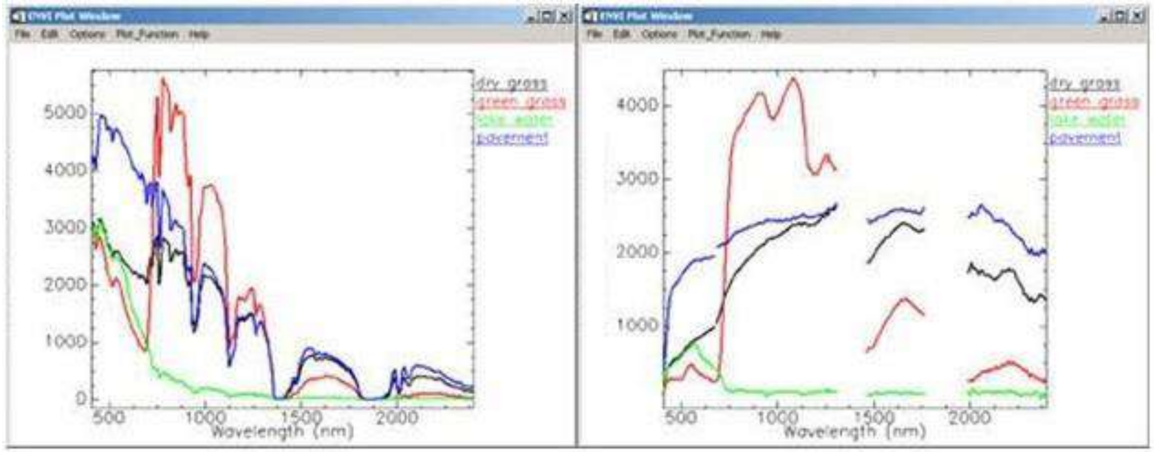
– Less important at long waves (infrared), more evident at short wavelengths

Figure by Ocean Optics

…and why is it red at sunset?



From radiance to reflectance



– After correcting all these aspects, we can convert each pixel value into the fraction of reflected energy for each band (from 0 to 1).

– To do this there are a lot of different methods

  – We are not going to see them in detail

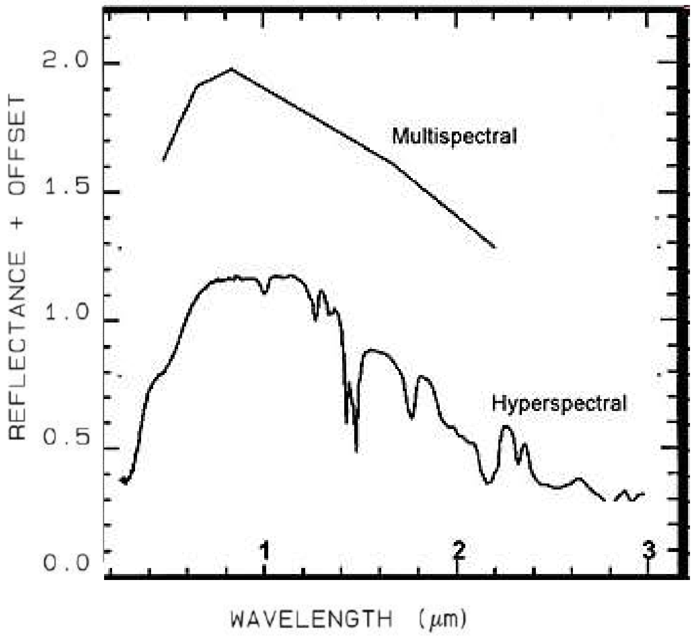
– It is not mandatory to do this (only if we need to work with physical values)

– For statistical operations we can also use the data in radiance

DLR

Figure courtesy of Exelis VIS

What we cannot see in Multispectral images

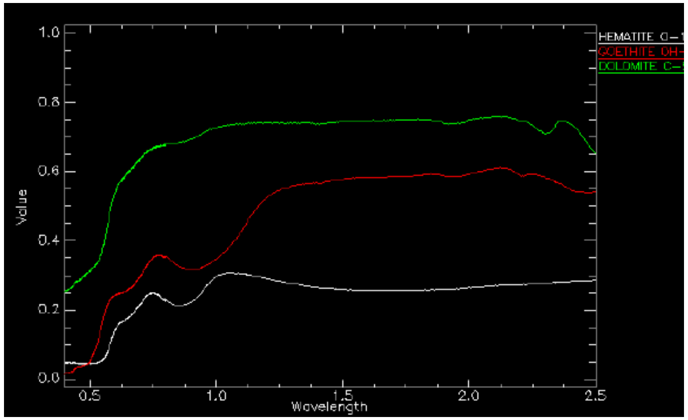


Landsat (7 bands)

- Laboratory (up to 1000 bands)
- Hyperspectral images (up to 250 bands)

- The main characteristic of hyperspectral sensors: their bands are contiguous
  - It is not just the number of bands they contain!
  - We are going to see an application with a sensor having only 15 bands!
  - The important thing is to represent a material with a continuous curve in a given area of the spectrum

Spectral Signatures



Each material can be identified through its characteristic spectral signature

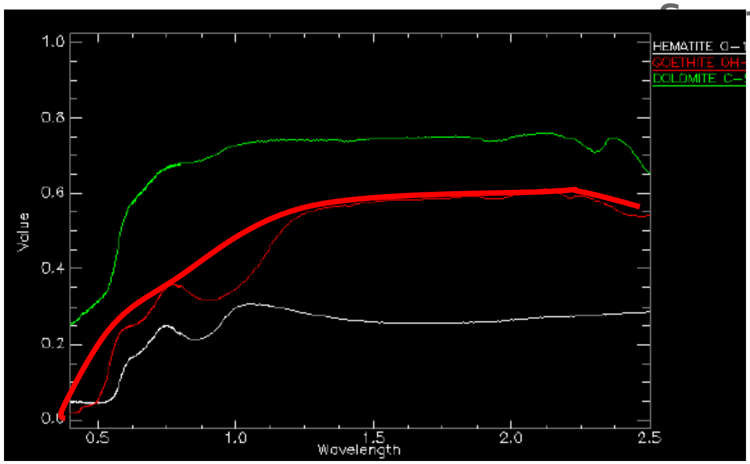
- In this example 3 spectra of minerals adquired in laboratory

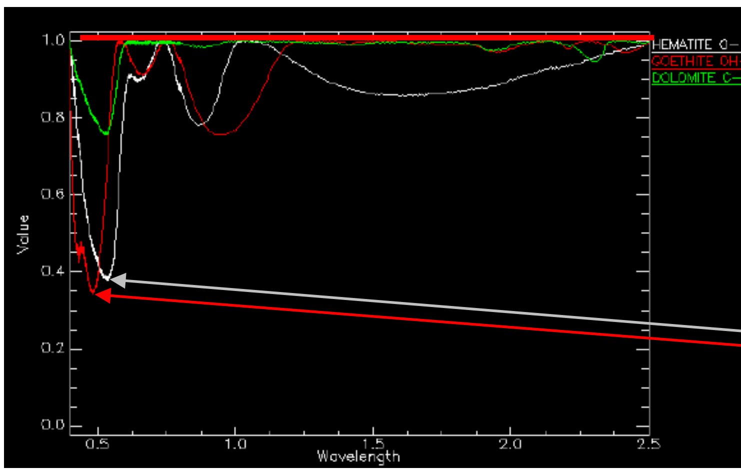
– Different members in each class (in this case different kinds of rocks):

- Cannot always be identified by the "level" of the curves
  - In an image these depend on illumination conditions
- They are usually identified by small variations in frequency of the maxima and minima of the slope (derivative) of the curve

tral Signatures

Spectra and Spectral Signatures





After Continuum Removal

- Most of the information is in the absorbing bands (less reflected energy)
- Spectra can be represented in an alternative way to highlight this
- Continuum removal: the general shape of the spectra is subtracted
- Absorbing bands become more evident
- This helps in distinguishing the classes of interest for some applications

DLR

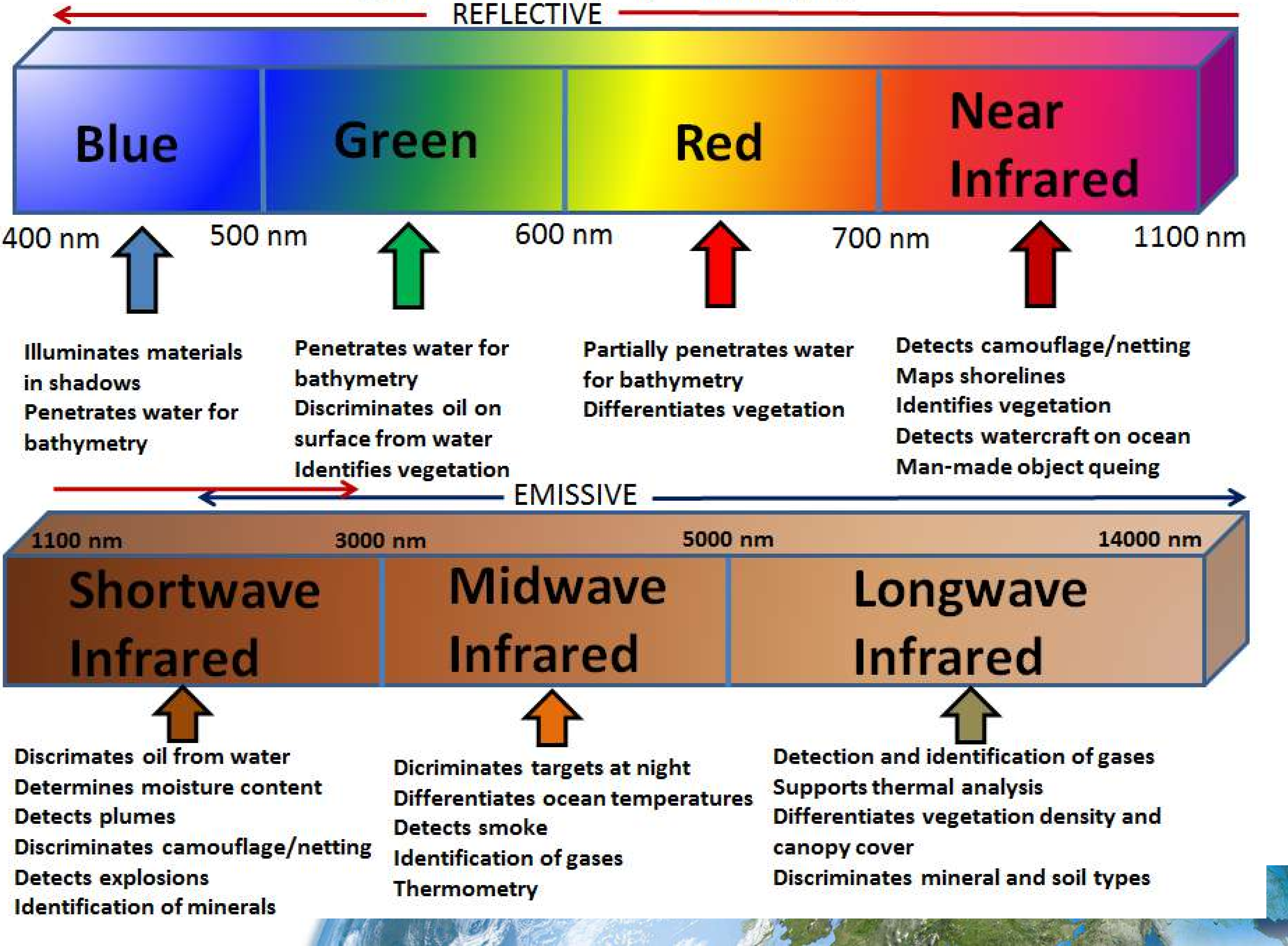
Applications of Hyperspectral Images

Daniele Cerra, German Aerospace Center (DLR)

DLR

Slide by Mark Elowitz

Applications of Hyperspectral Imaging



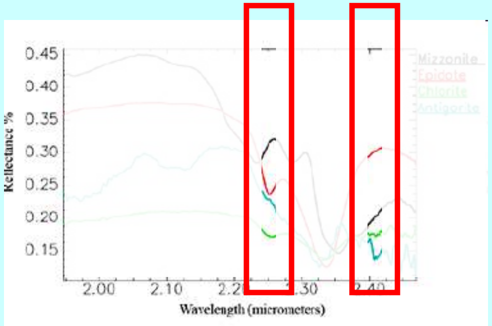
Applications of Hyperspectral Images

Do I really need all the bands?

We can distinguish two „families“ of applications

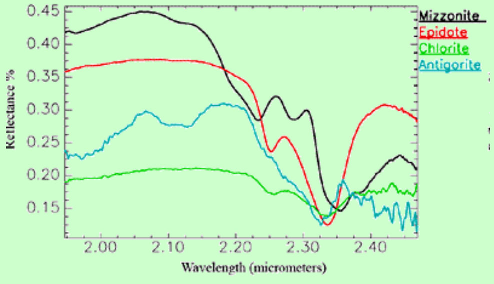
We only may need two or three bands

- Water Quality
- Vegetation health
- Gas leaks from gasoducts

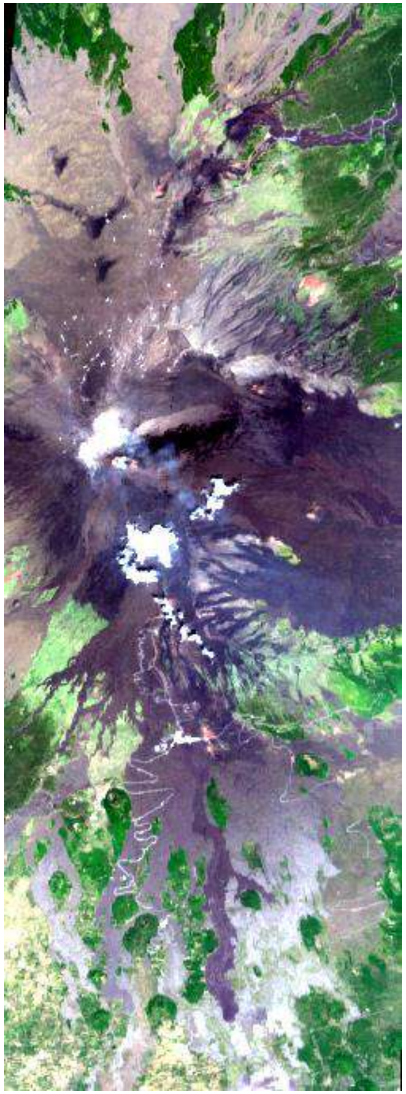


We consider the full spectrum of each pixel

- Mineralogy
- Acid mine drainage



Really all the bands?



Several sources of noise in HS data

Coming from the sensor/introduced in the preprocessing steps

Atmospheric absorption & interferences

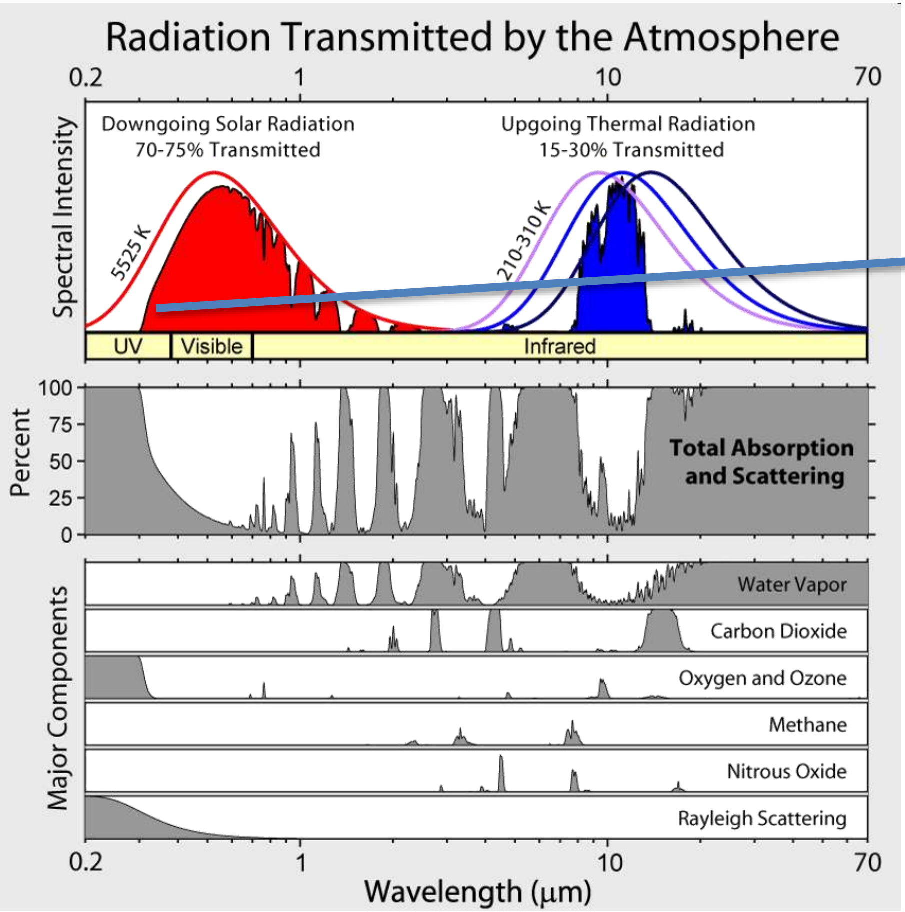
Thermal noise

Electronic failures…

Hyperion Etna dataset RGB false color composite

Animation of all the 133 bands in the dataset (0.4 – 2.5 μm)

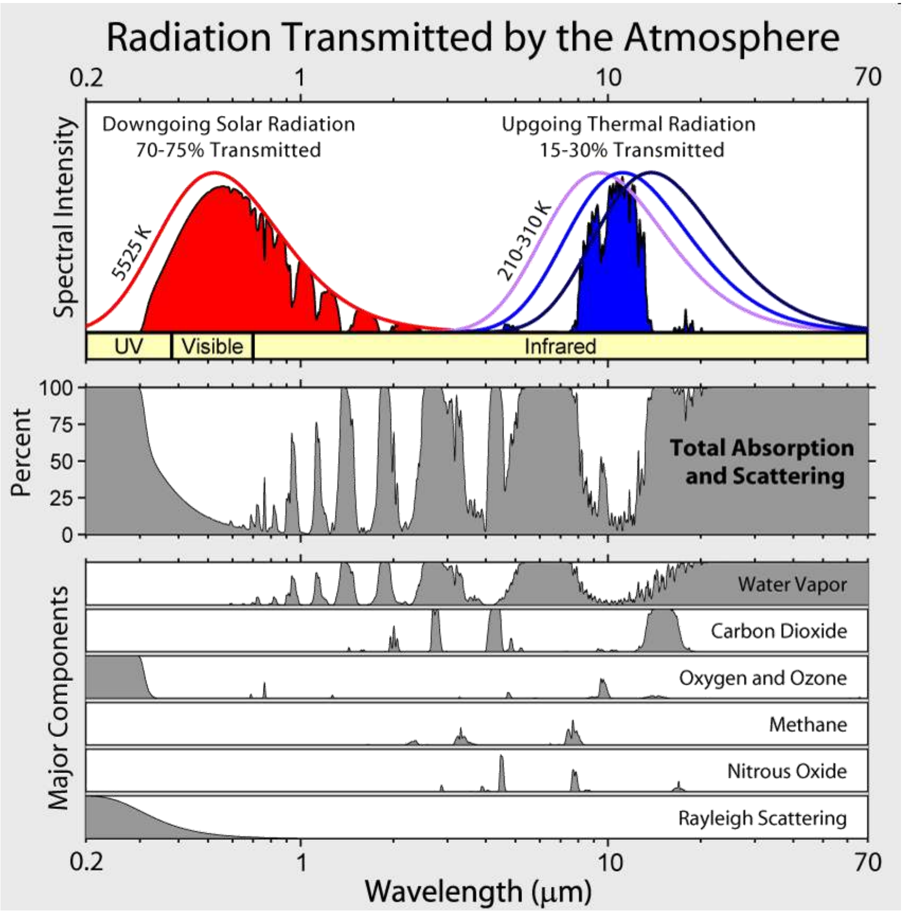
Example: bands at the edge Ultraviolet-Visible Light

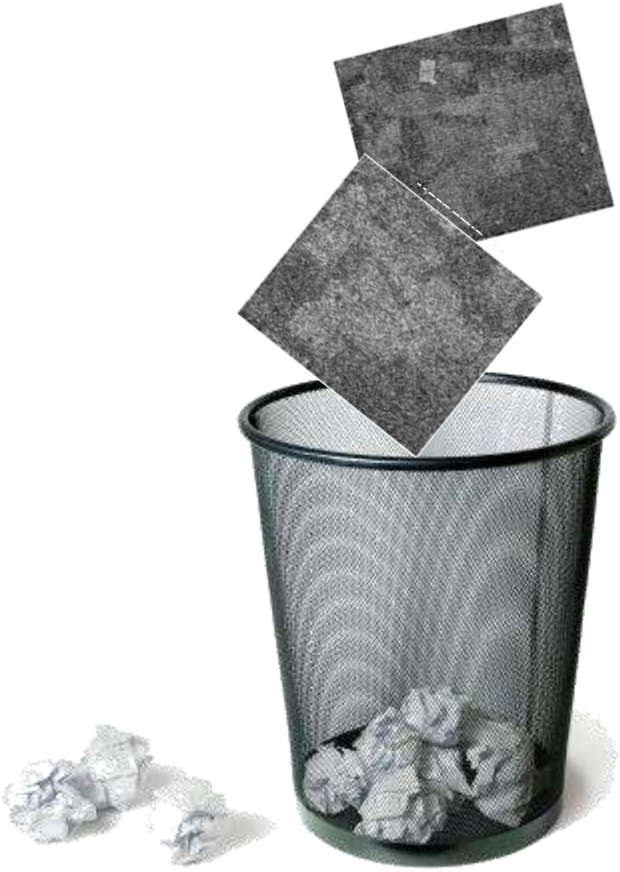


AVIRIS Salinas dataset, 380 nm

DLR

These bands are usually discarded!



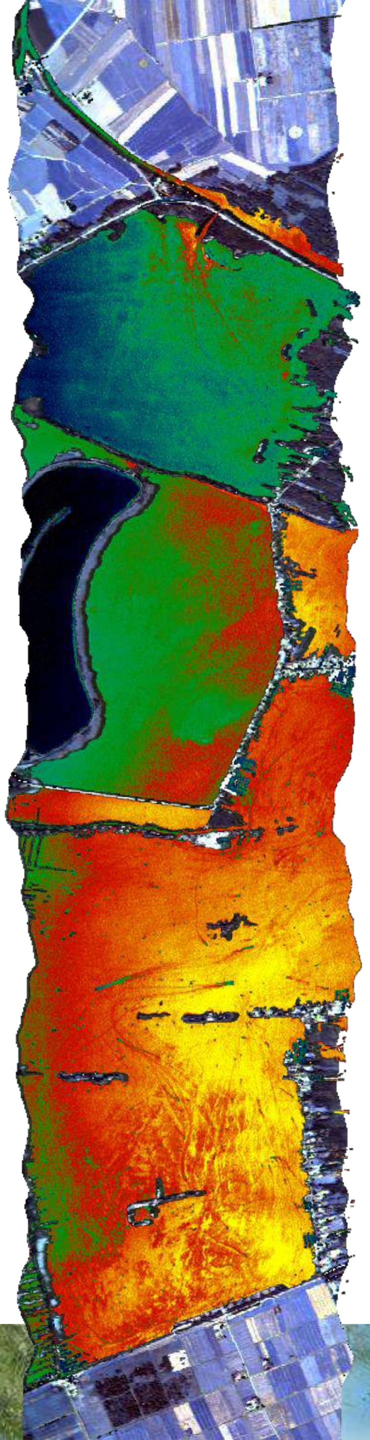


Water quality

- Reflectance of water in visible freqs:
  - Reflected light
  - Bottom reflectance
- Suspended matter in water has higher reflectance
  - Mineral Sediments
  - Clorophyll
- Measuring Clorophyll-*a*
  - Estimation of alga (seaweed) biomass
  - Anomalous values indicate alga blooms
  - Normally algas mono-species which fishes don't eat
  - Not eaten algas drop on the bottom, removing oxygen from there
  - Water quality drops down

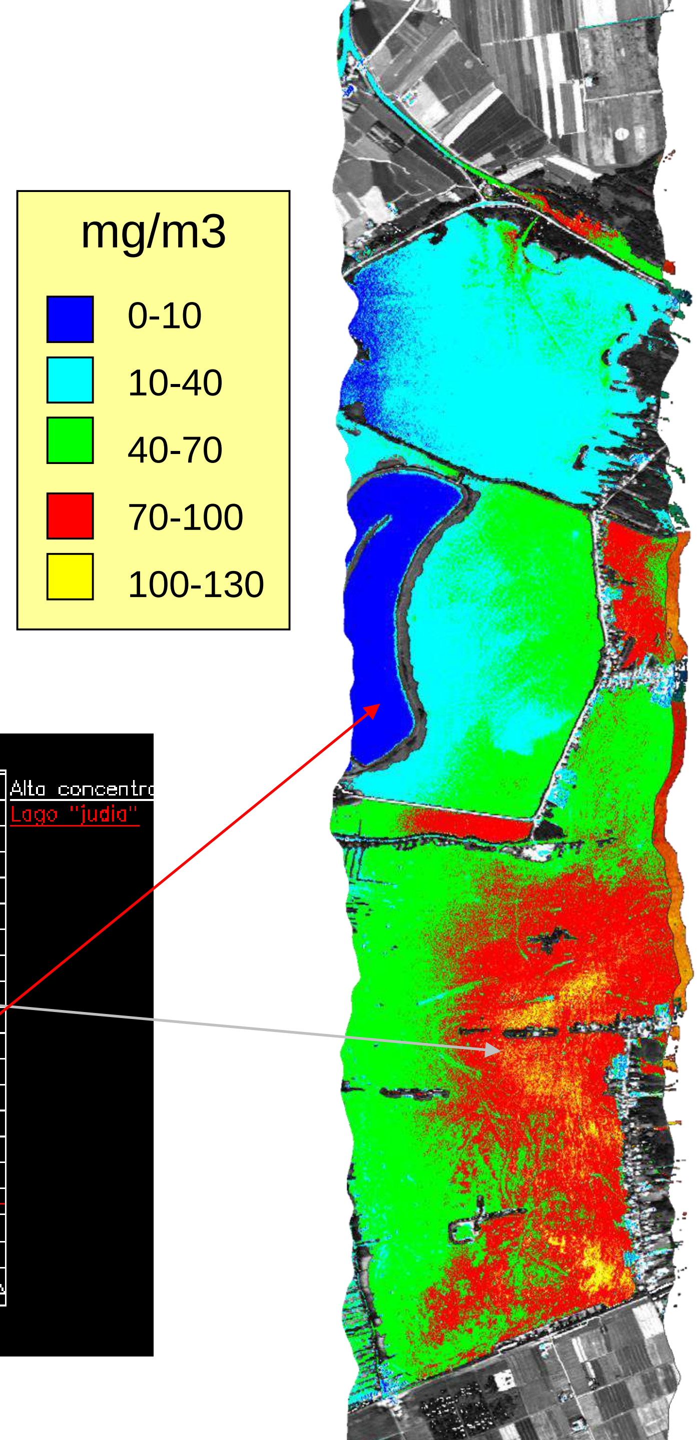
Water quality

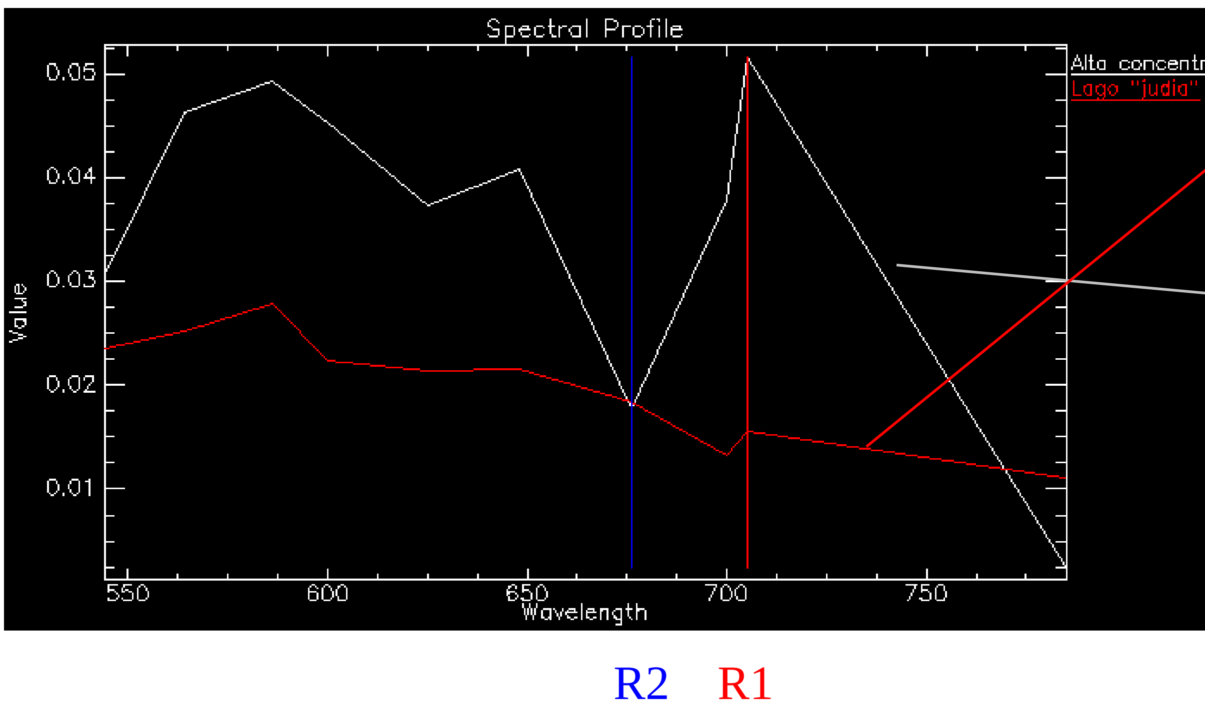
- Compact Airborne Spectrographic Imager (CASI)
- A hyperspectral airborne sensor which operates in the visible and near infrared frequencies
  - Fewer bands than other sensors (around 20)
- We want to investigate the quality of water reserves in this image acquired on the lakes of Vechstreek, Netherlands
- To measure the clorophyll in a pixel x we apply the following equation:
  - $\mathrm{Clor}(x) = 90\,(\mathrm{R1}(x)/\mathrm{R2}(x)) - 70$
  - Where R1(x) y R2(x) are the reflectances for x around 702 and 675 nm, respectively



Water Quality

- The water from the channel with a higher clorophyll concentration flows in the lake
- The bean-shaped lake is a drinkable water reserve
  - It must be kept free from Algae blooms





What Information can we get from Band Ratios?

Examples for Landsat images

Separate water from ice

$$\text{Ratio} = \frac{\text{Band 4}}{\text{Band 5}}$$

Spot hidrotermally altered rocks

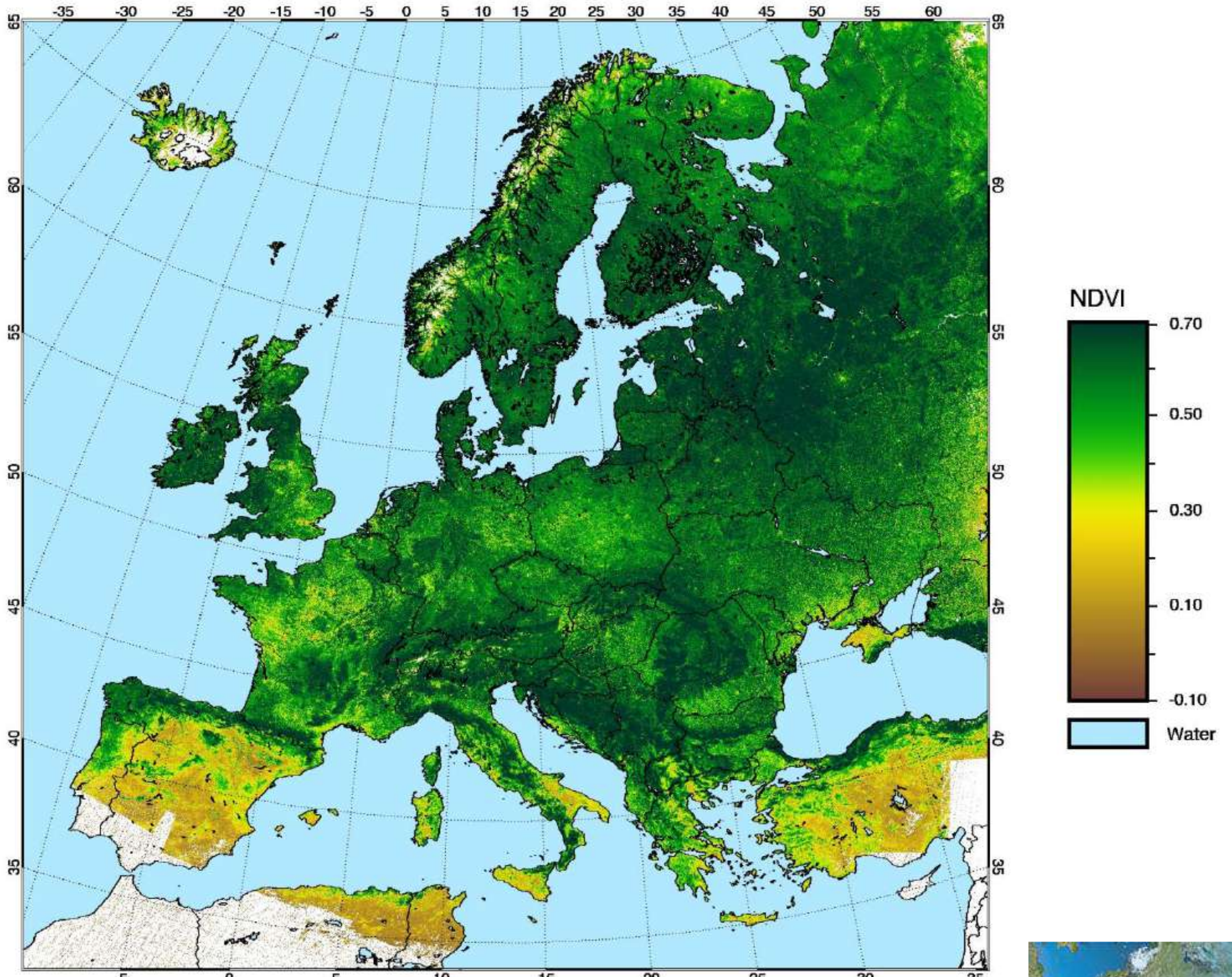
$$\text{Ratio} = \frac{\text{Band 3}}{\text{Band 2}}$$

Highlight Urban Area

$$\text{Ratio} = \frac{\text{Band 3}}{\text{Band 4}}$$

DLR

Normalized Difference Vegetation Index (NDVI) „THE“ Band Ratio

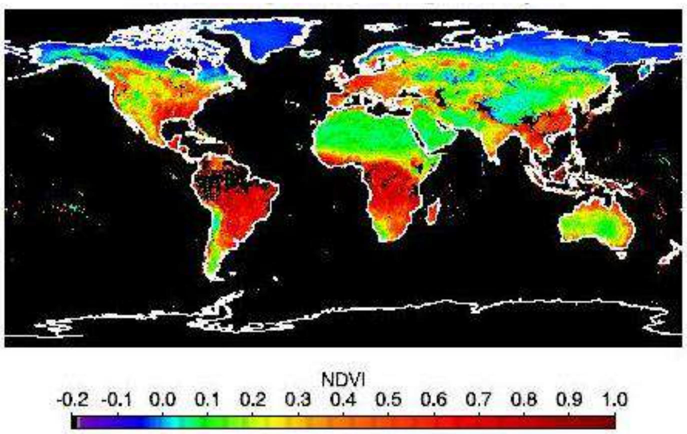


NDVI

Near Infrared Band

Red Band

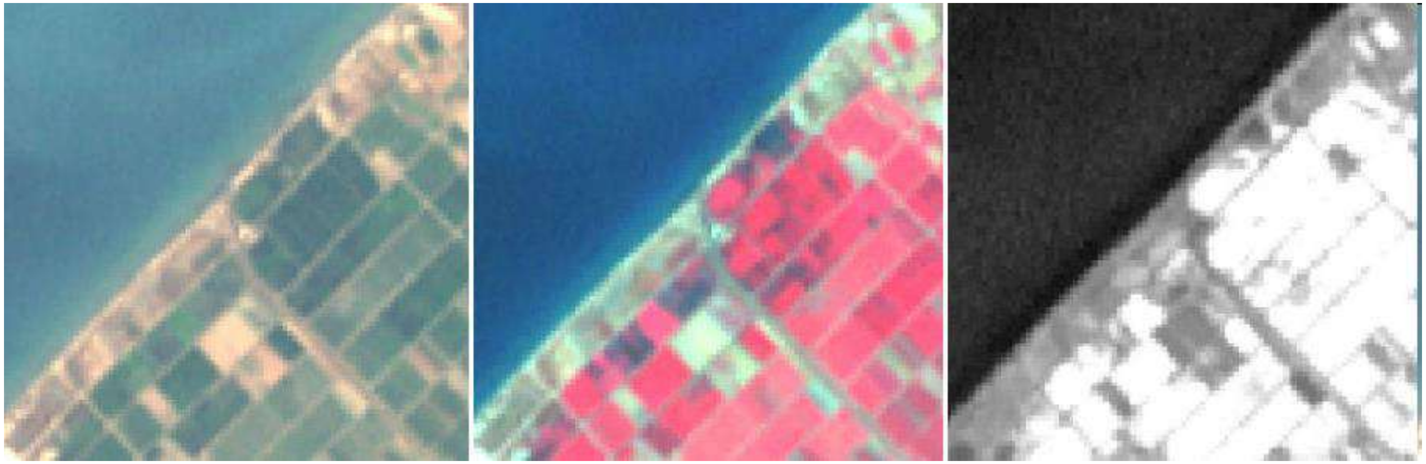
$$NDVI = \frac{NIR - R}{NIR + R}$$



- Normalized Difference Vegetation Index
- Probably the most well-known and used band ratio
- -1 ≤ NDVI ≤ 1
- Used to:
    - Detect vegetation
    - Create masks to restrict the image analysis to areas of interest
    - Roughly estimate green biomass

DLR

NDVI, Landsat Example



True Color

Bands 321

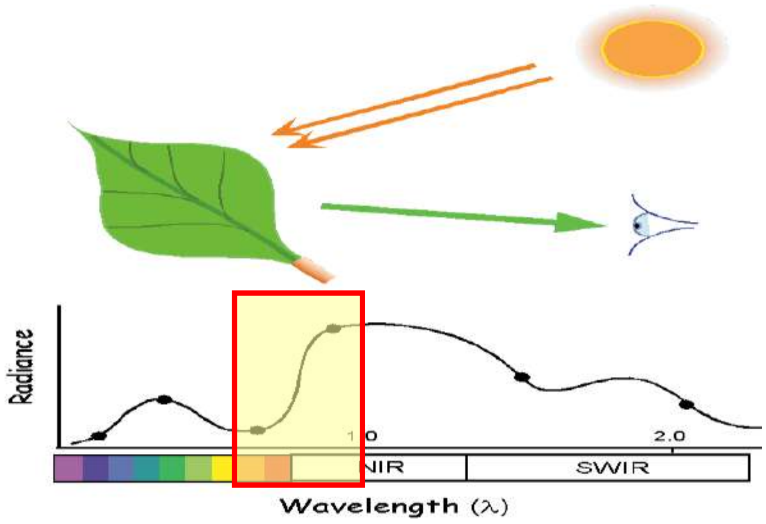
False Color

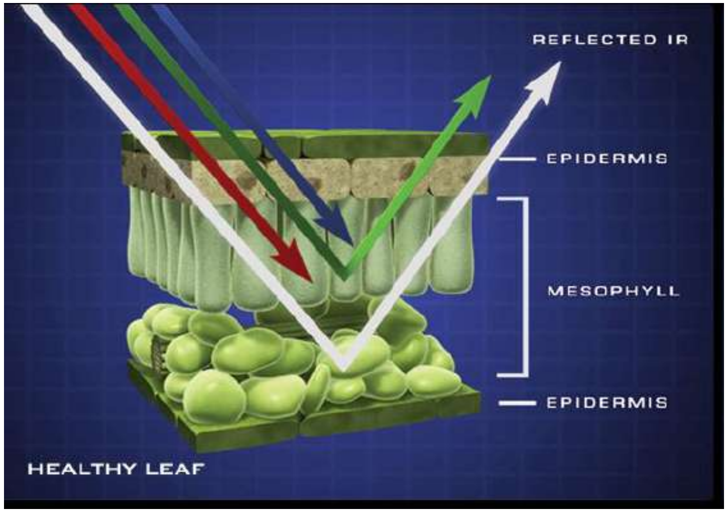
Bands 432

NDVI

$$\frac{\text{Bands } 3-4}{\text{Bands } 3+4}$$

Vegetation & Spectra



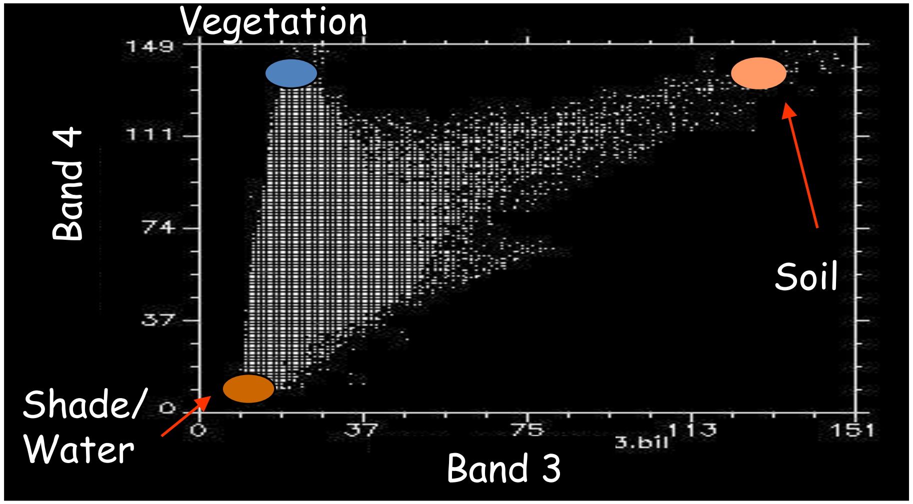


$$NDVI\,(Landsat) = \frac{Band\,4 - Band\,3}{Band\,4 + Band\,3}$$

Chlorophyll strongly absorbs visible light (0.4 to 0.7 μm), with max absorption at 0.7 μm → Landsat band 3

Cell Structure strongly reflects Near-IR (0.7 – 1.1 μm) → Landsat band 4

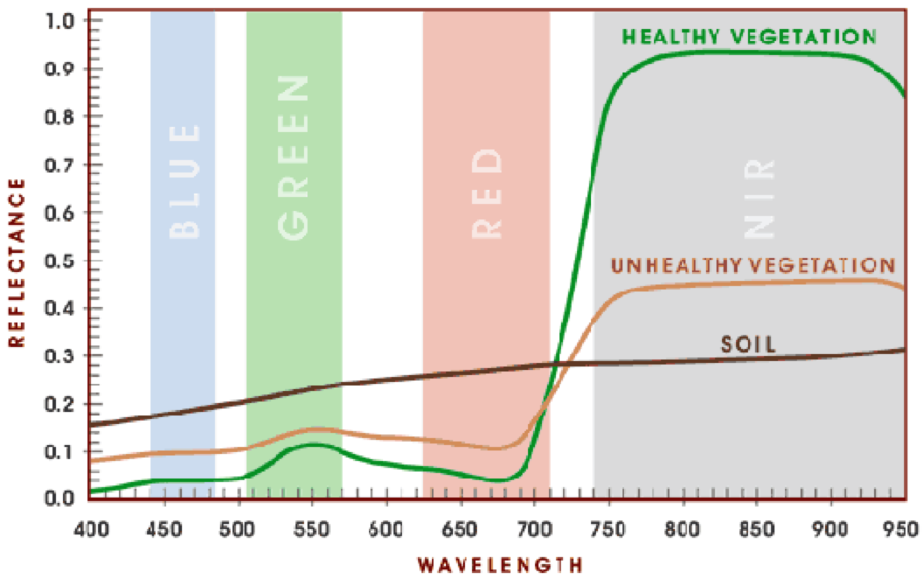
NDVI & Landsat

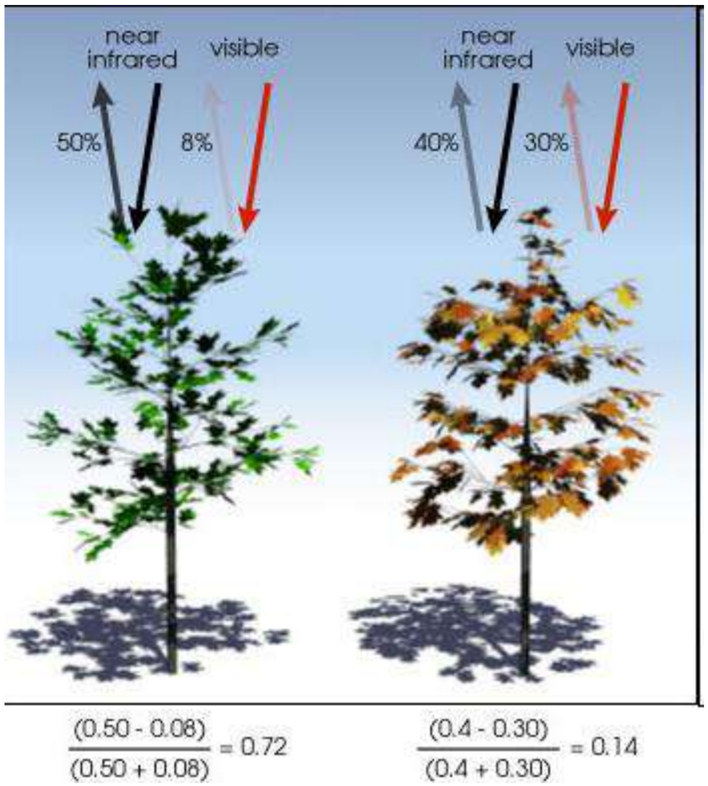


Scatter Plot Band 3 vs. Band 4 for a natural scene in a Landsat image

NDVI derives from empirical observations!

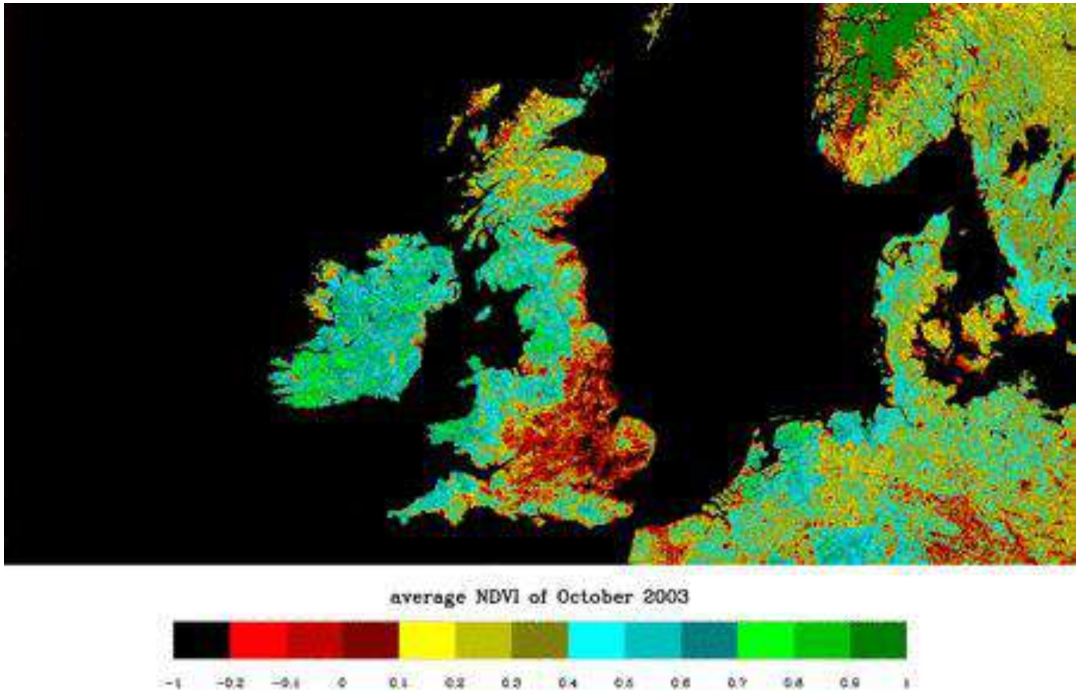
NDVI & Vegetation‘s Health





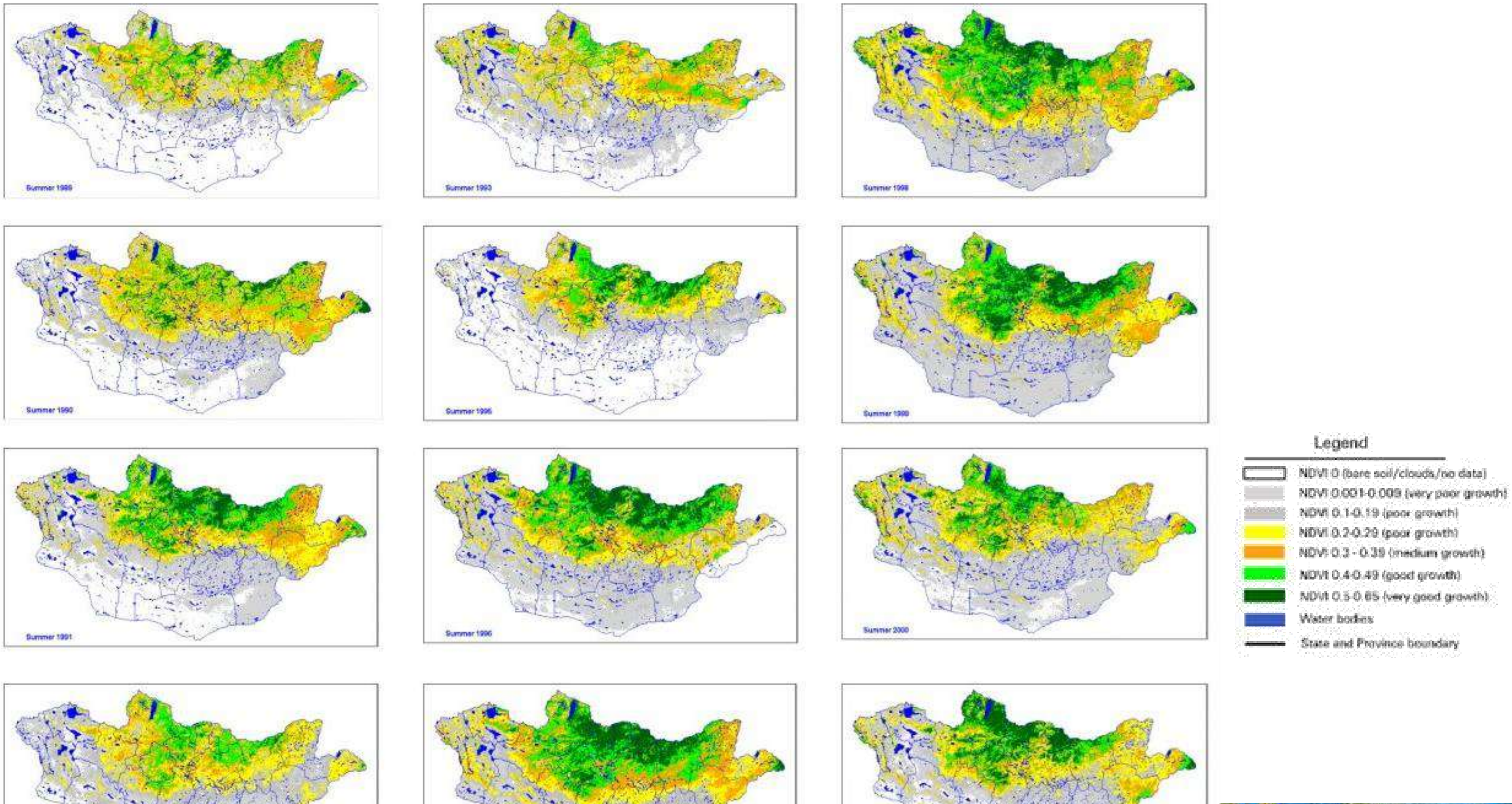
NDVI is related to the health status of the vegetation

NDVI in Different Seasons…



..and in different years!

Time-series NDVI images between 1989-2001, Mongolia. NOAA/AVHRR



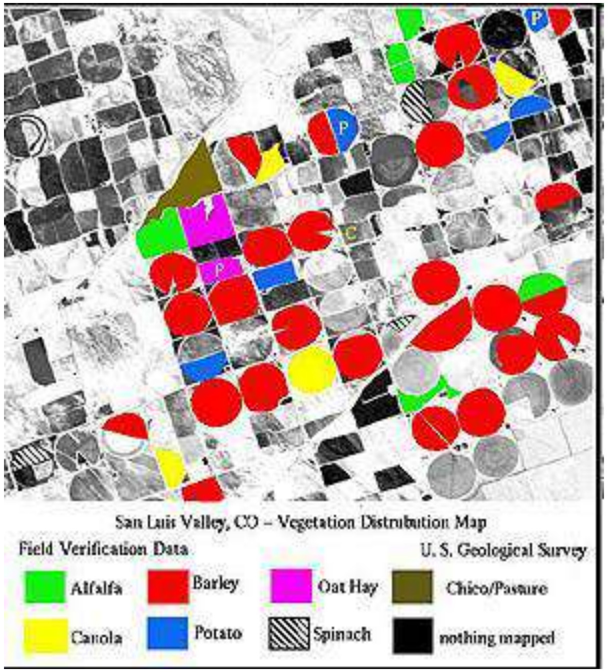
Hyper- vs. Multispectral: Vegetation Analysis

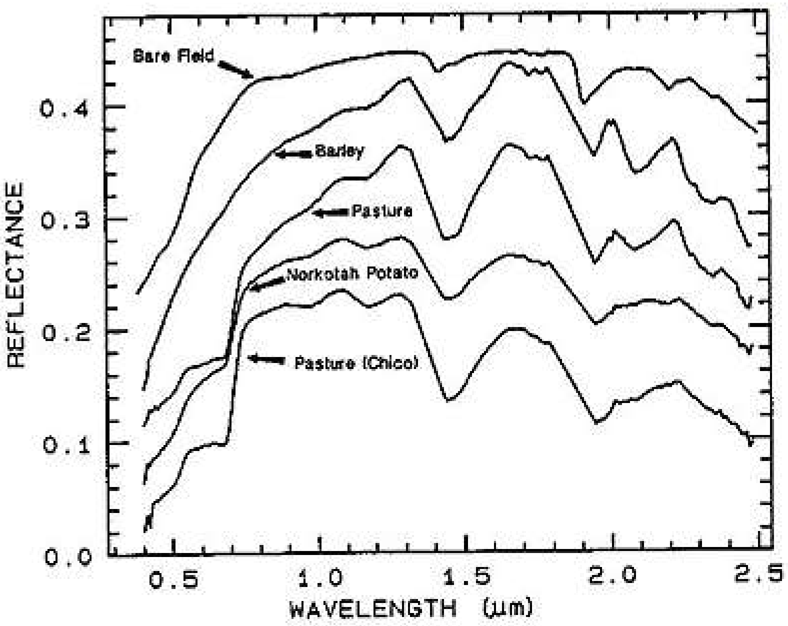
Limitations of NDVI

- NDVI can only measure the vegetation biomass only at surface and late growth stages
  - Gives 2D information rather than 3D
  - Difficult to obtain from NDVI a good estimation of the biomass (volume in cubic meters of forests/vegetation in general)
- NDVI cannot predict the amount of nitrogen concentration in the vegetation
  - Key parameter to understand at early stages if the health status of crops is worsening
  - NDVI is limitated for the task of vegetation health estimation

But in Hyperspectral Images we can do more!

images courtesy of NASA

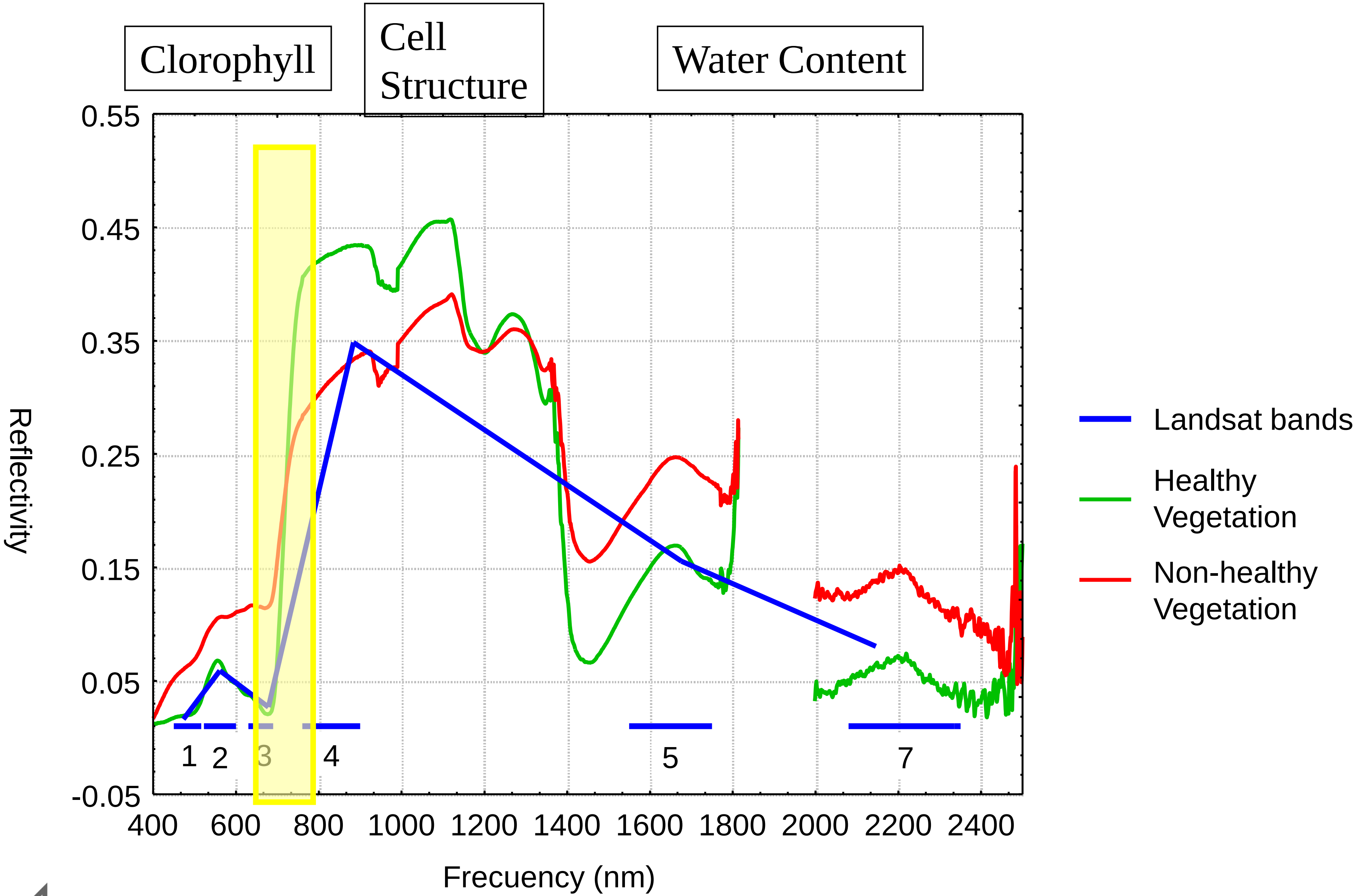




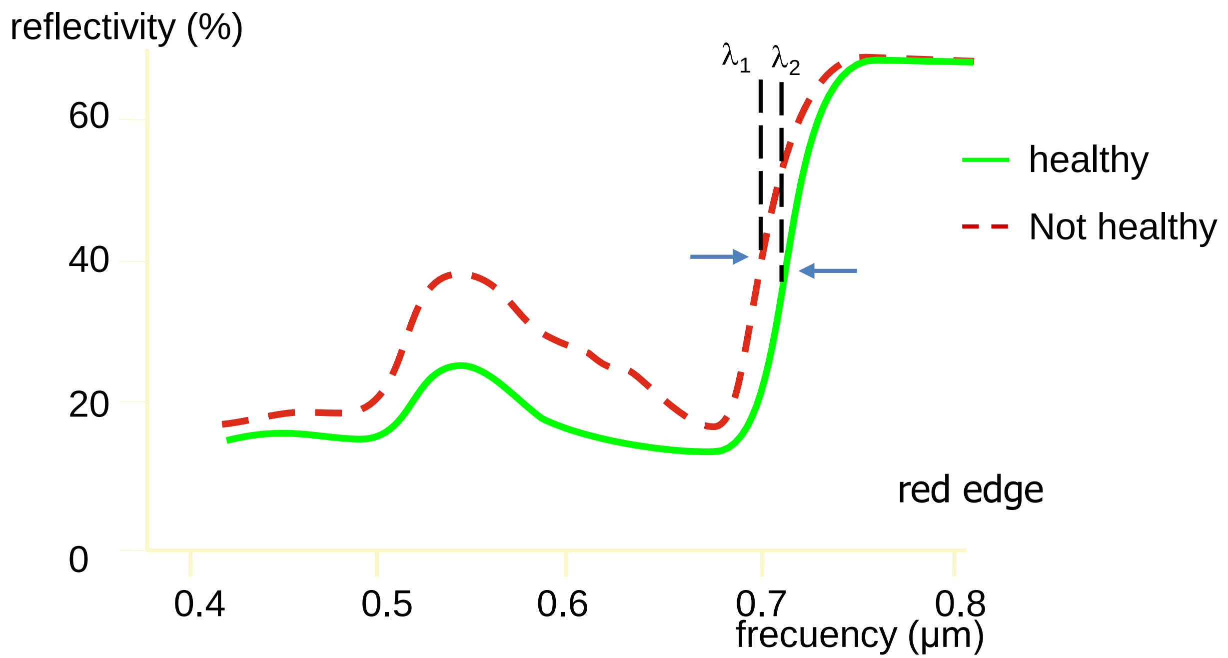
- Analyzing the spectra it is possible to extract information for each field/crop
- For vegetation the spectral range between red and near infrared is of special interest (around 700 nm)
- We are looking for a steep increase in reflectivity in this area (red edge position)
- More about it later!

DLR

Spectral signatures of vegetation: beyond NDVI



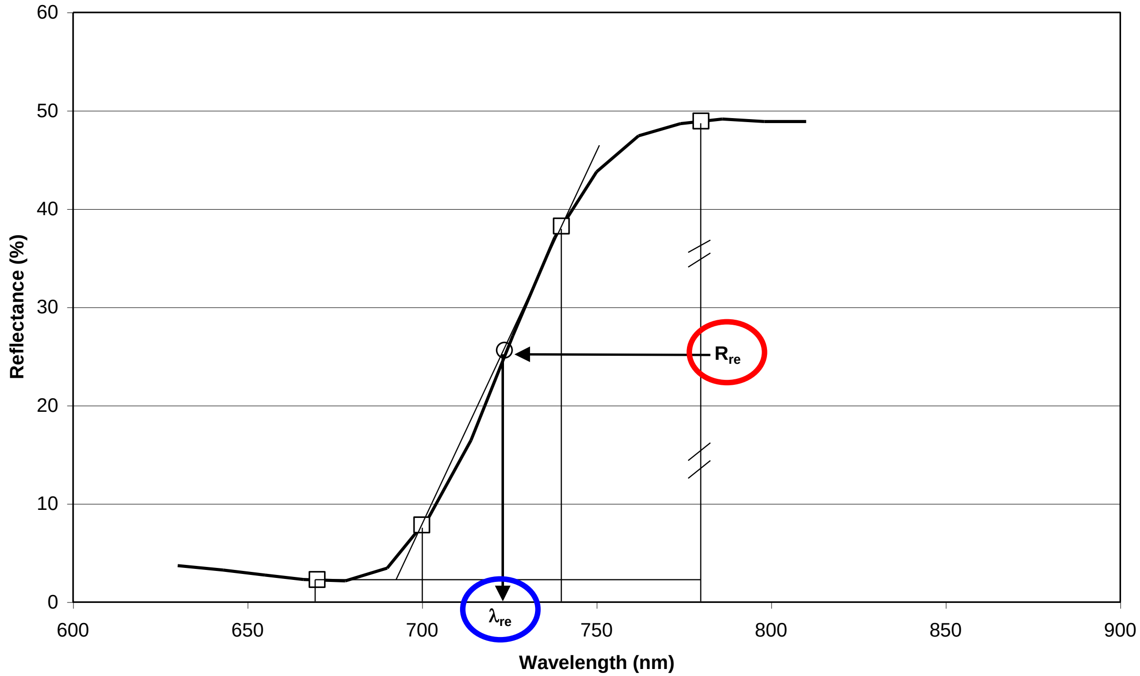
Near Infrared: the Red Edge



- Transition between absorption into red and high reflectance in the near infrared portions of the spectrum
- The **red edge** is the spectral range in which this change is observable (flexion point in the curve)
- It depends on the amount on clorophyll in the plant and nitrogen in the soil
- A displacemente to the left of the red edge characterizes ill vegetation
  - Scarce clorophyll in leaves
  - “Breathing” problems of the plant

DLR

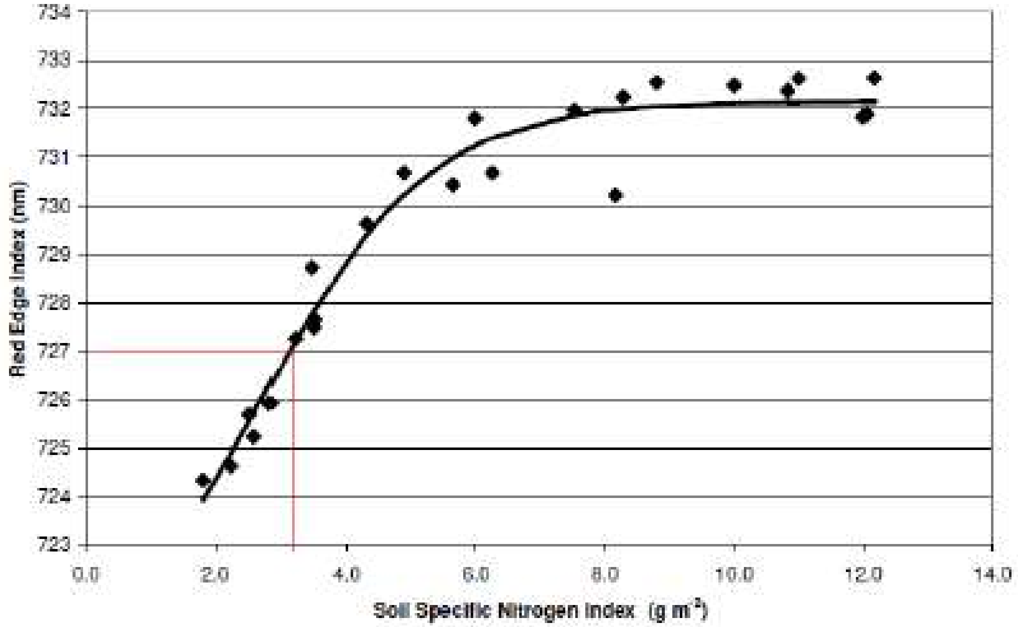
How to compute the red edge position?



- We need again only 4 bands among the available ones
- First we compute the reflectivity in the inflection point in the spectrum x
  - $RE(x) = (R1(x) + R2(x)) / 2$
  - Where $R1(x)$ and $R2(x)$ are the reflectance values of x around 670 and 780 nm
- Afterwards we compute the red edge frequency position by the following equation:
  - $\lambda = 700 + 40 ((RE(x) - R3(x)) / (R4(x) - R3(x)))$
  - Where $R3(x)$ is the reflectance of x at 700 nm and $R4(x)$ at 740 nm

DLR

For which red edge values is the vegetation not in good health?

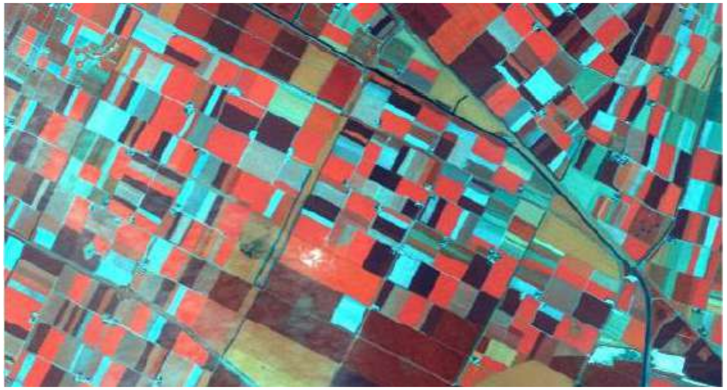


Red edge index as function of the Soil Specific Nitrogen Index for a potato crop

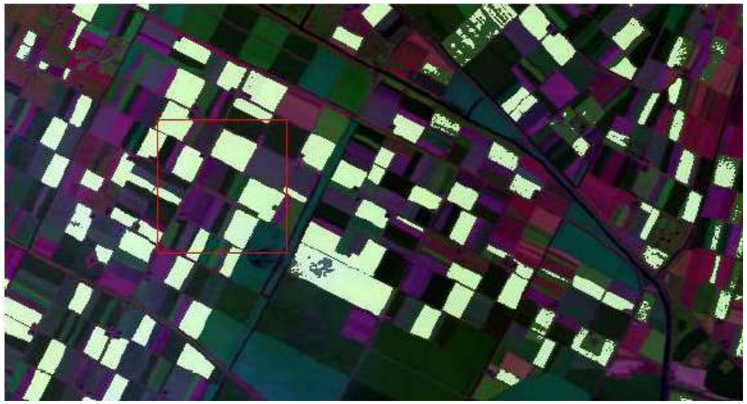
- Lack of nitrogen in the soil indicates respiratory problems of the plants
  - For potato fields this happens for values < 3.5
  - We have these values for red edge values < 727

DLR

Vegetation Health



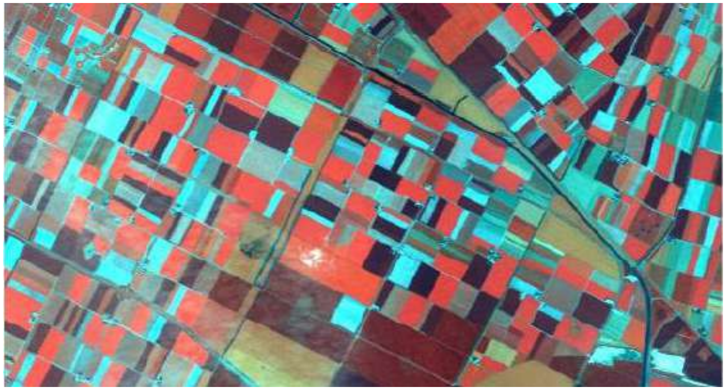
Crops



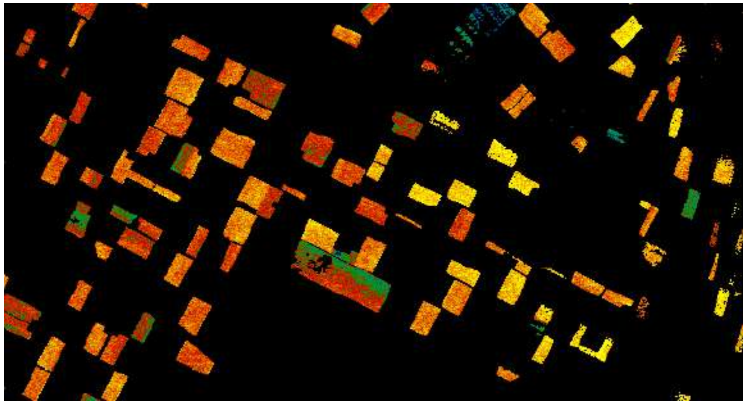
Detection of potato fields

- We want to see which potato fields are in good health
- Let’s compute the red edge position in these fields and check where these values are < 727

Vegetation Health



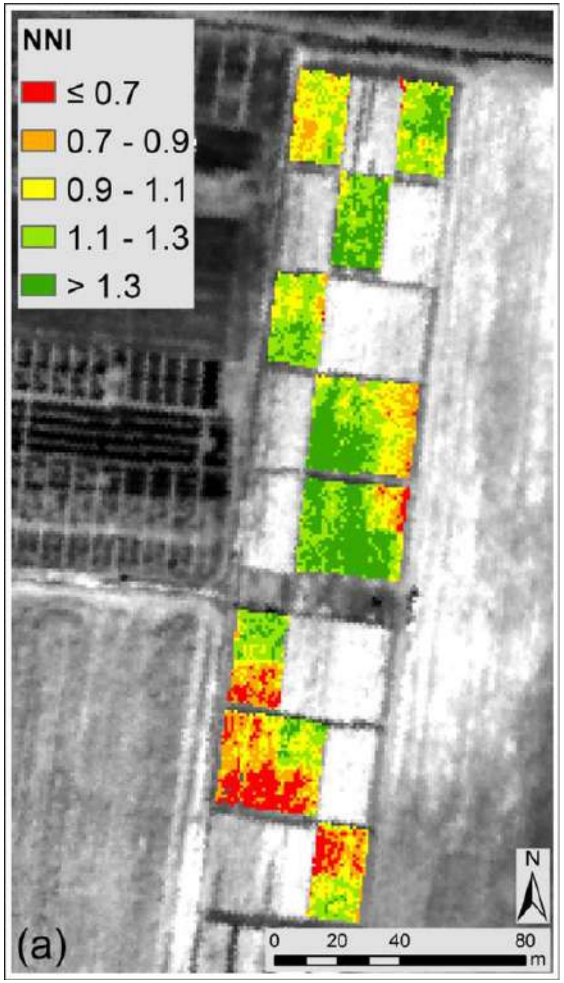
Crops



Red edge values in potato fields

- The fields in blue/green are not healthy
  - Red edge position < 727 nm
- Fields in orange/yellow are very healthy

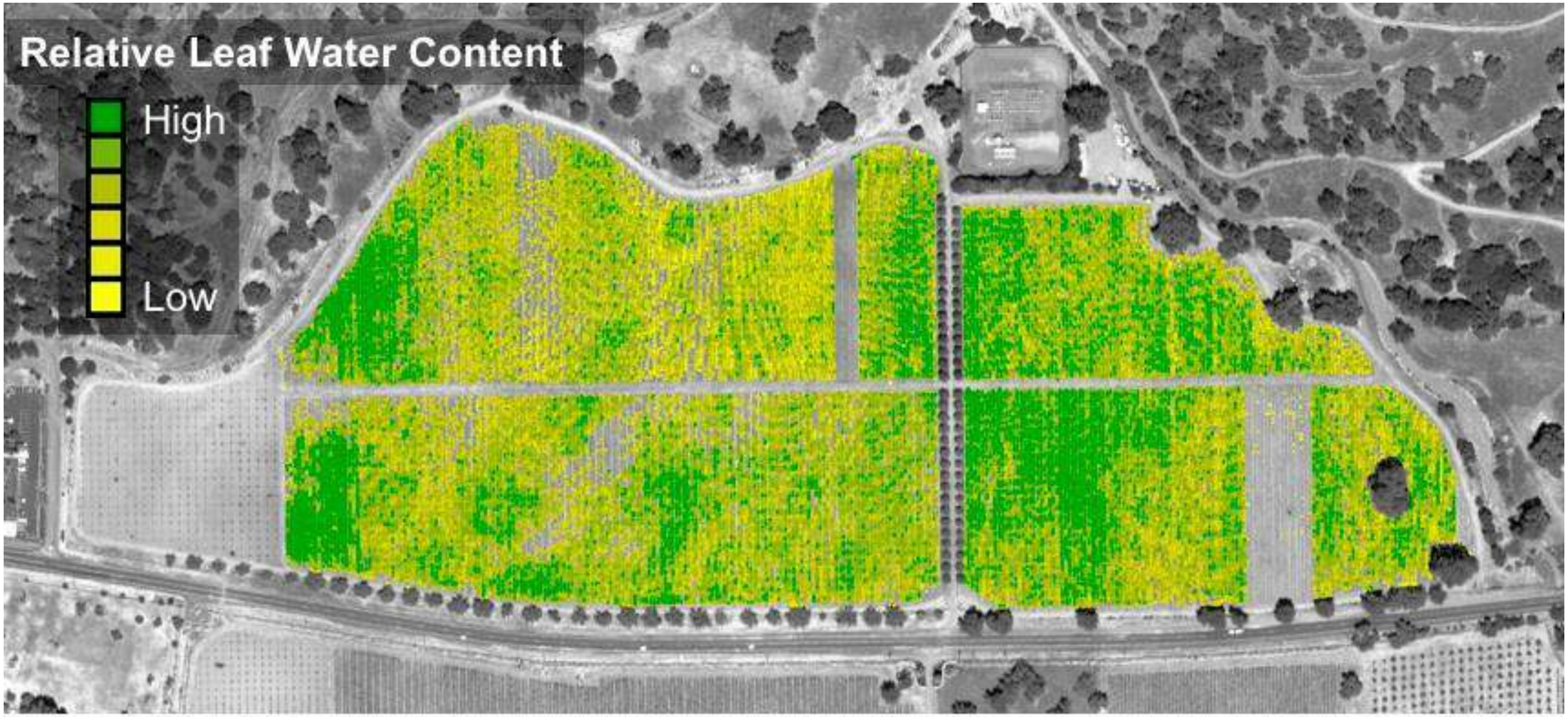
Corn Fields



Chiara Cili et al.

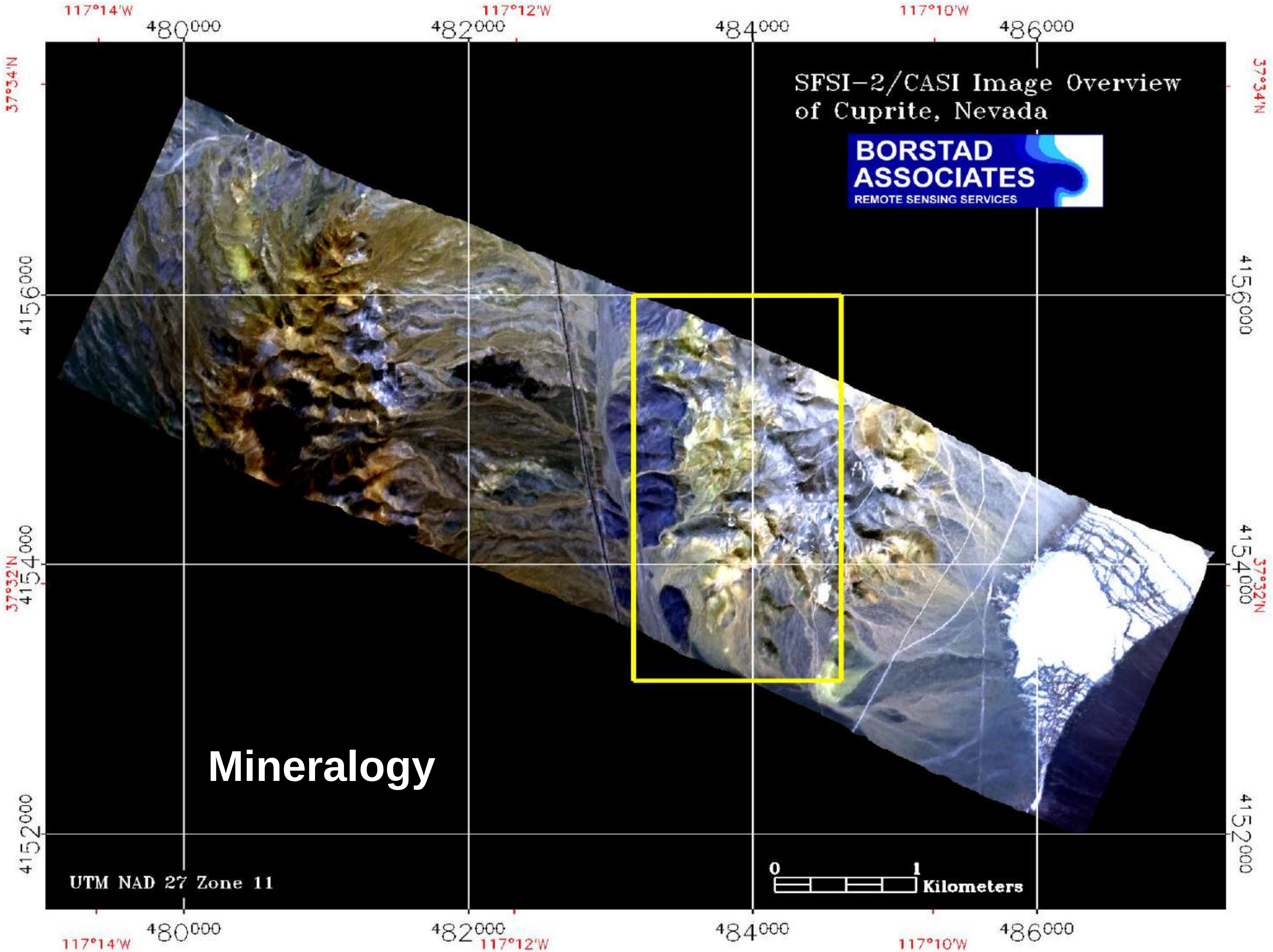
DLR

Other Vegetation Parameters: Relative Leaf Water Content



Normalized Difference Infrared Index (NDII) was used to estimate relative leaf water content. This index responds to reflectance changes at 1649 nanometers, a small water absorption feature.

data courtesy of SpecTIR

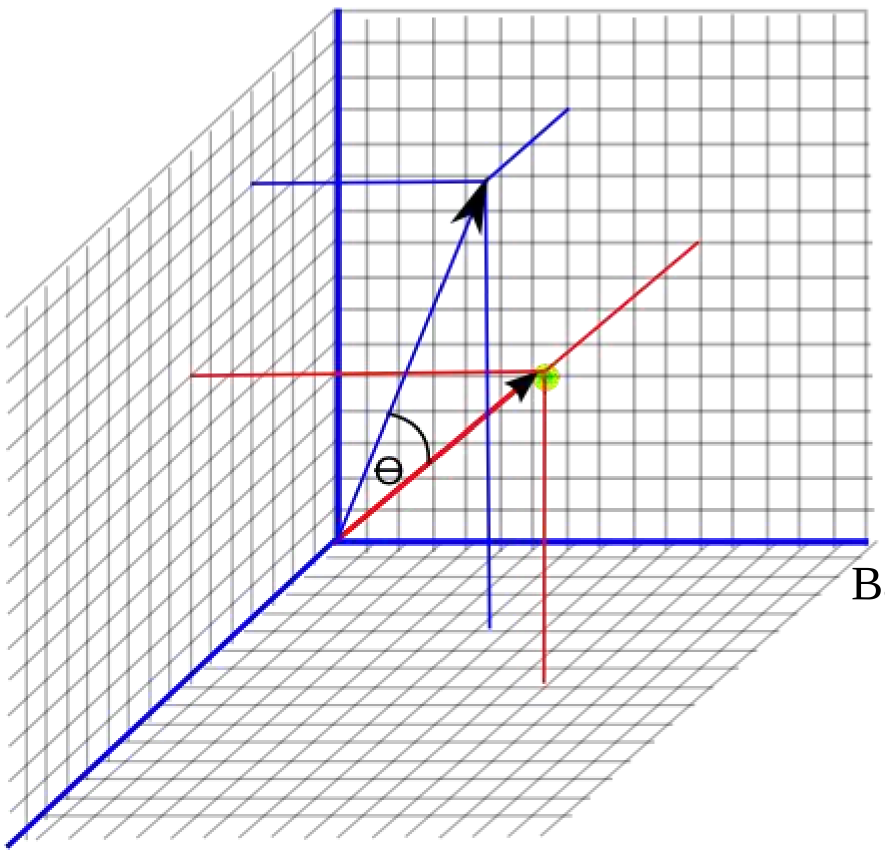


117°14'W
117°12'W
117°10'W
480000
482000
484000
486000
37°34'N
37°32'N
4156000
4154000
4152000
SFSI-2/CASI Image Overview of Cuprite, Nevada
BORSTAD ASSOCIATES
REMOTE SENSING SERVICES
0
1
Kilometers
UTM NAD 27 Zone 11

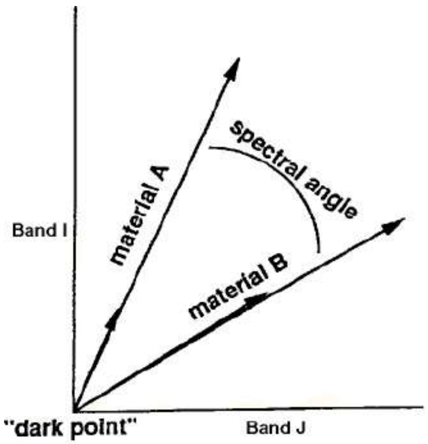
Mineralogy

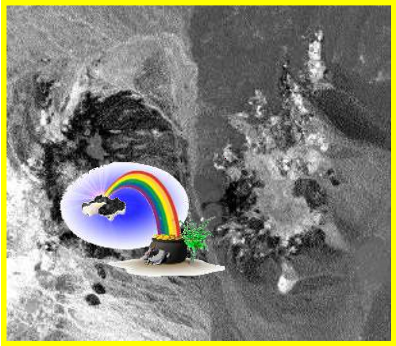
Mineralogy: Spectral Distances

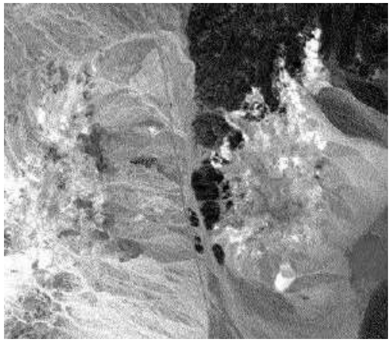
- We are interested in mountainous areas in which potentially we can find **gold mines**
- We are going to look fo alunite, a mineral which indicates the possible presence of gold
- We need a pixelwise classification of the area
- We need therefore to quantify how two spectra are similar to find similar spectra to alunite
- Spectral Angle Mapper (SAM)
  - Independent of illumination conditions (vector length)
  - Often the preferred choice to measure the similarity between two spectra

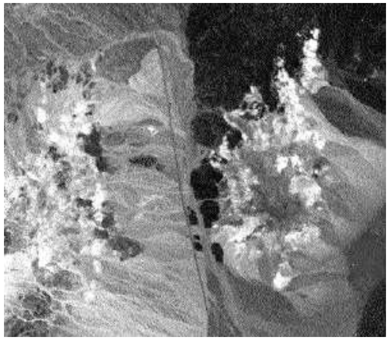


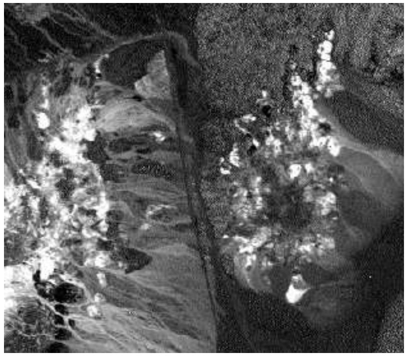
Spectral Angle (SAM)

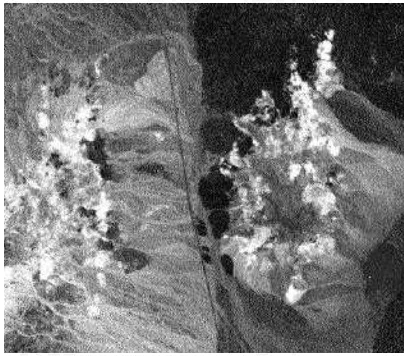


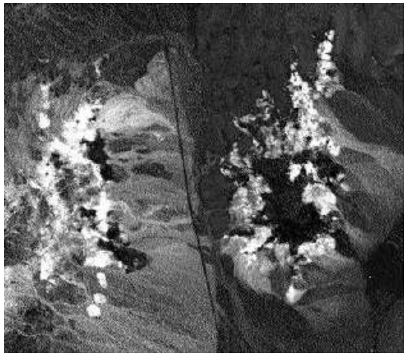

Alunite

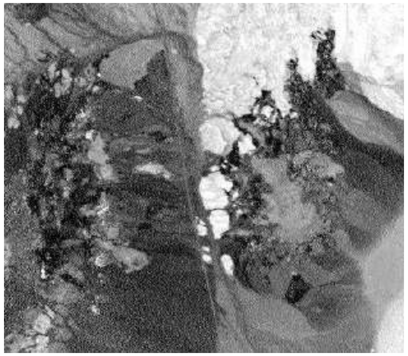

Buddingtonite


Calcite

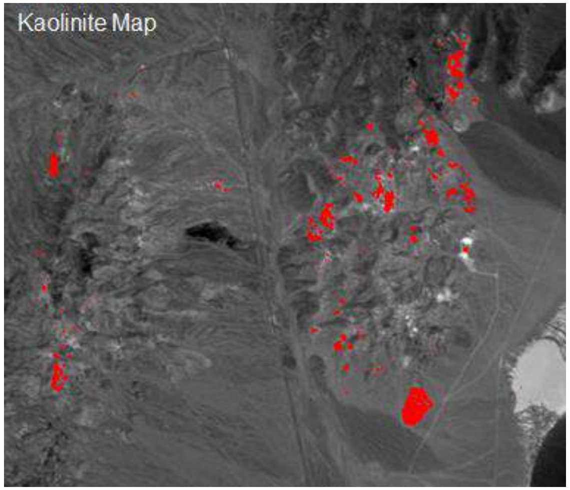

Zeolites

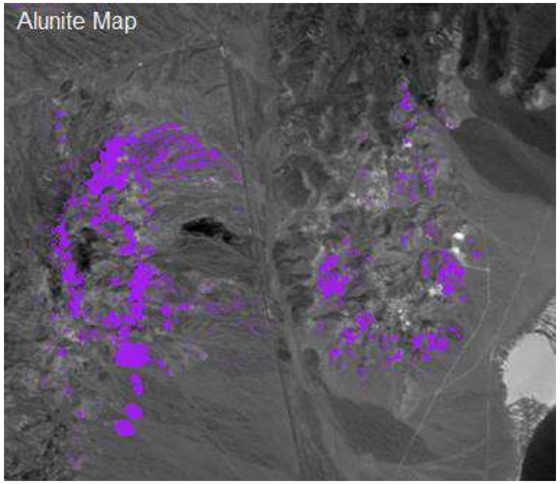

Illite

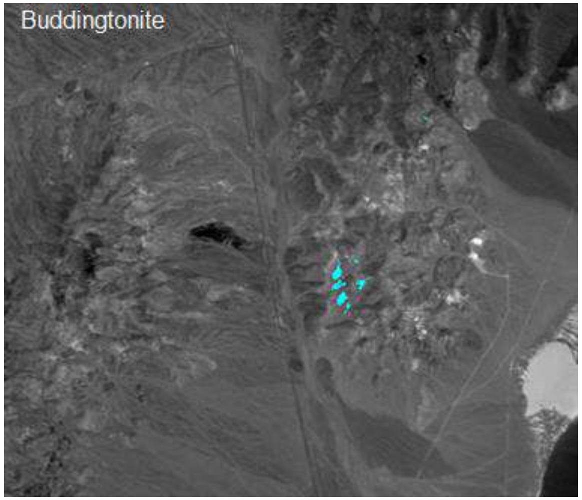

Silica


Kaolinite

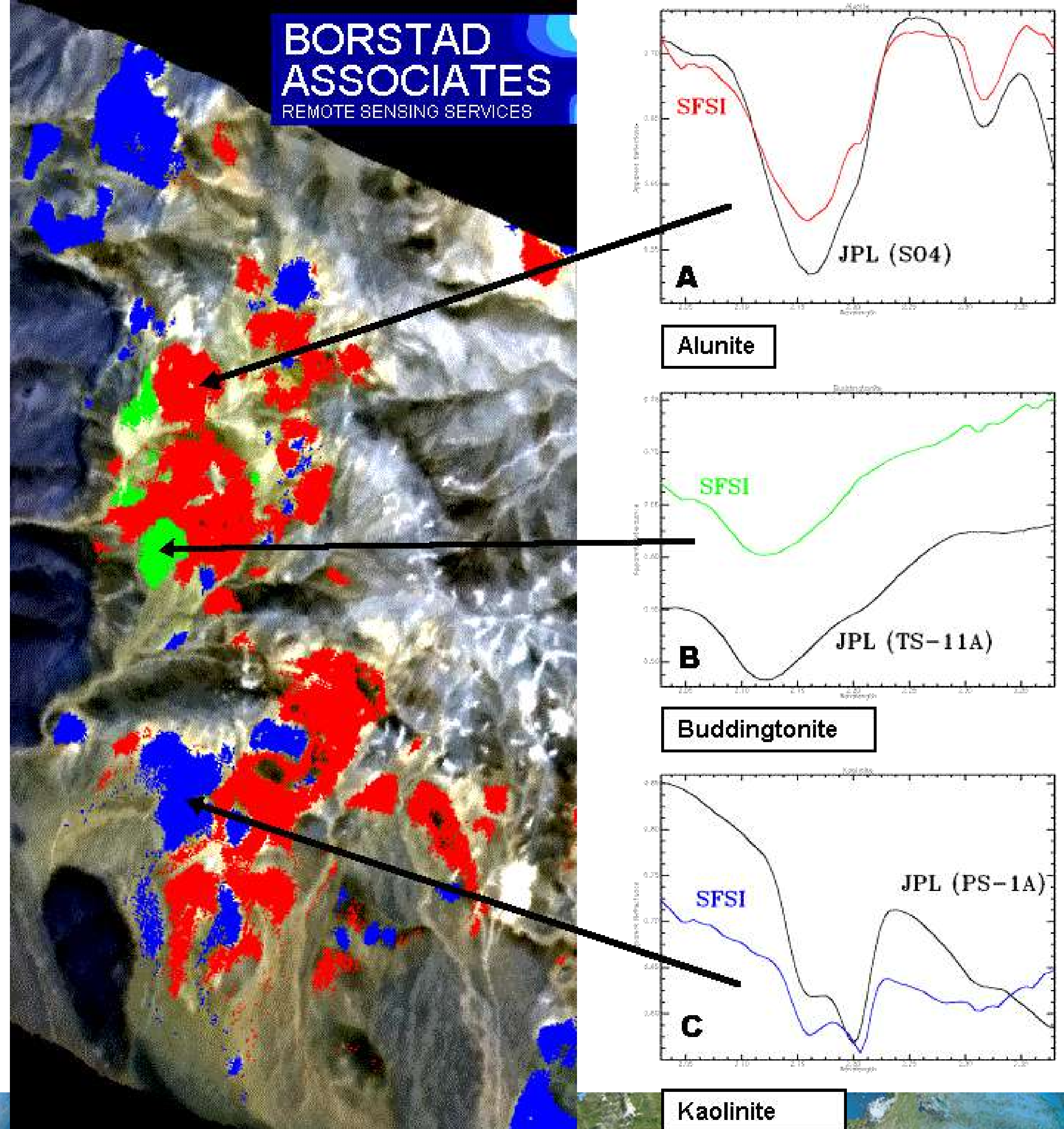
Maps for specific materials







- Image SFSI (SWIR Full Spectrum Imager)
- Clasification based on SAM
- Minimum distances between a pixel on the ground and a spectrum adquired in laboratory
- We should start to look for gold within „red“ areas
- Many spectral libraries are freely available from:
  - NASA (JPL)
  - USGS



DLR

Detection of Fires and Burned Areas

- Analyzing specific bands we can derive parameters in areas which suffered damages from fire

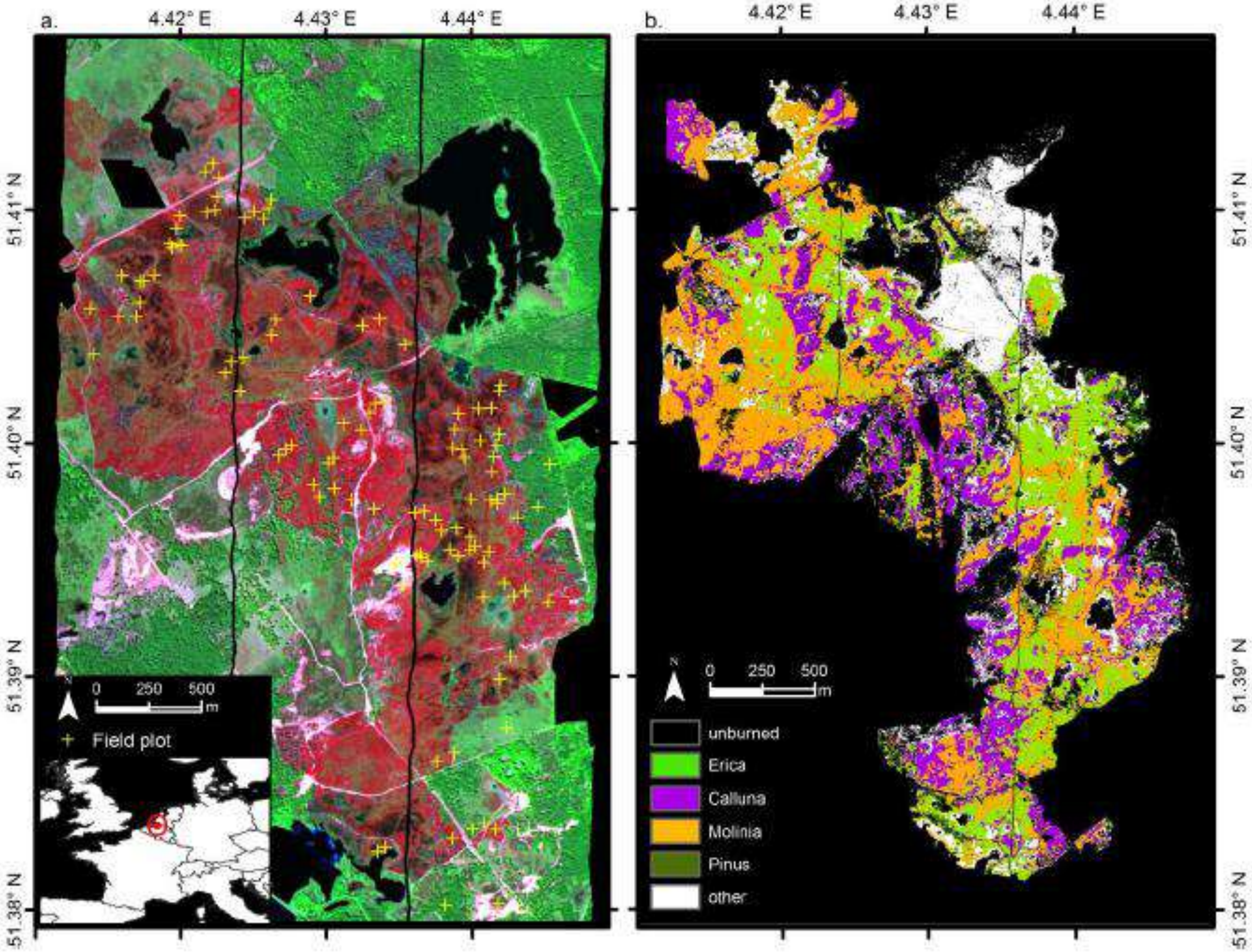
Burned Area Index	BAI	$BAI = \frac{1}{(0.1+R)^2 + (0.06+NIR)^2}$
Normalized Burn Ratio	NBR	$NBR = \frac{NIR-LSWIR}{NIR+LSWIR}$
Char Soil Index	CSI	$CSI = \frac{NIR}{LSWIR}$
Mid-Infrared Burn Index	MIRBI	$MIRBI = 10\ LSWIR - 9.8\ SSWIR + 2$

The indices were calculated using the bands with following central wavelengths for each spectral region: B: 499 nm; G: 552 nm; R: 699 nm; NIR: 801 nm; SSWIR: 1302 nm; LSWIR: 2332 nm.

DLR

Case of Study: Belgium

Birgen Haest, Lennert Schepers et al.



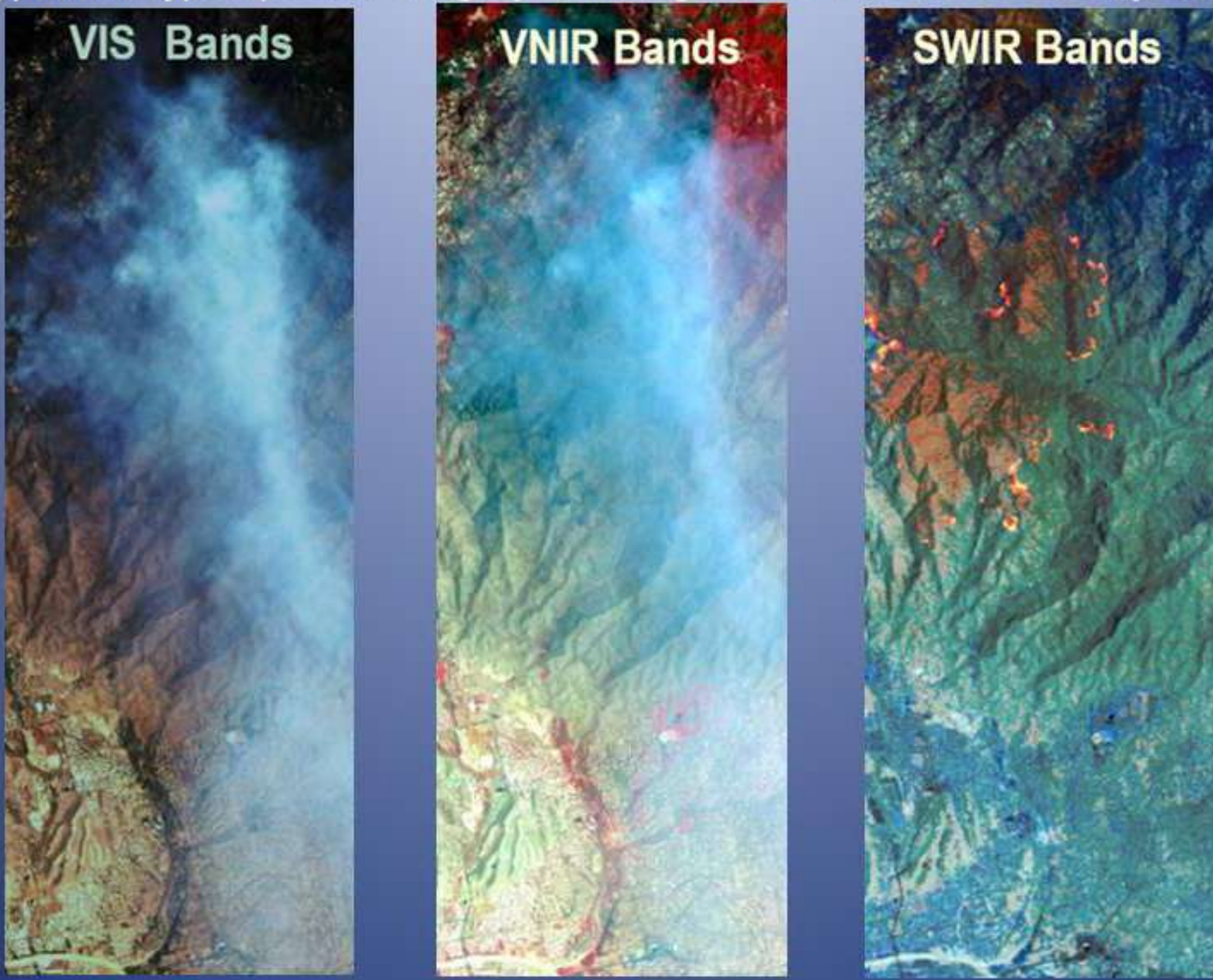
DLR

Case of Study: Belgium

Birgen Haest, Lennert Schepers et al.

DLR

Hyperion Hyperspectral Imagery Showing Tucson Wildfires – 3 July 2003

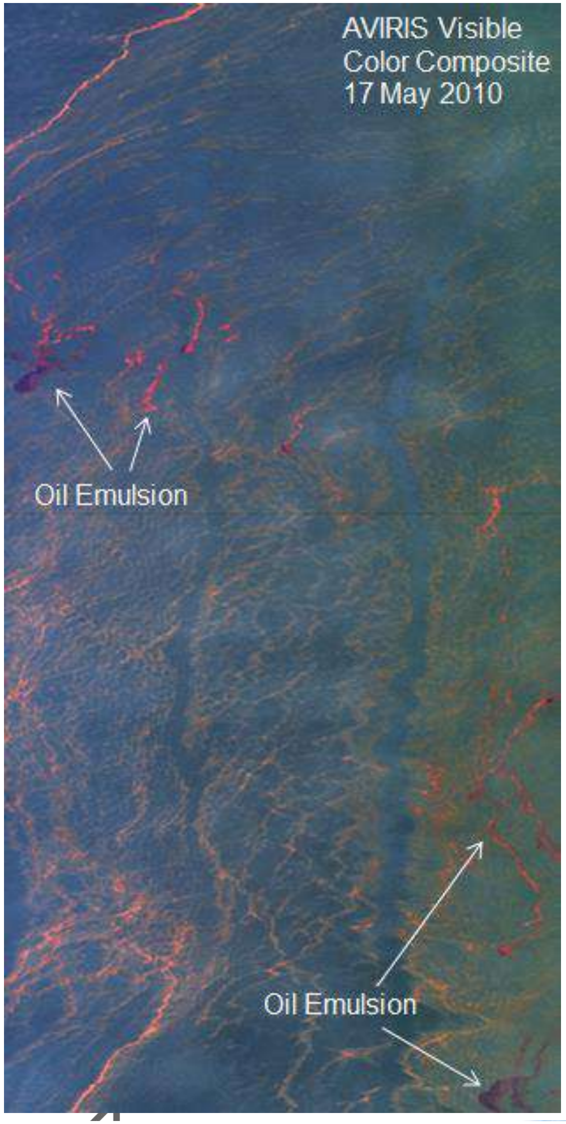


$$BI = \frac{\rho_{1100} - \rho_{2200}}{\rho_{1100} + \rho_{2200}}$$

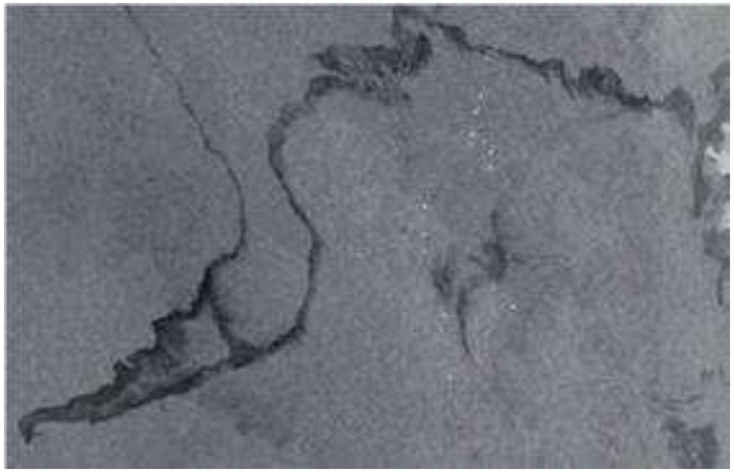
The Burn Index (BI) is used to detect burn scars due to wildfires. It has the Advantage over the NDVI in that it uses wavelengths that are transparent to smoke (1.1 and 2.2 microns).

Slide by Mark Elowitz

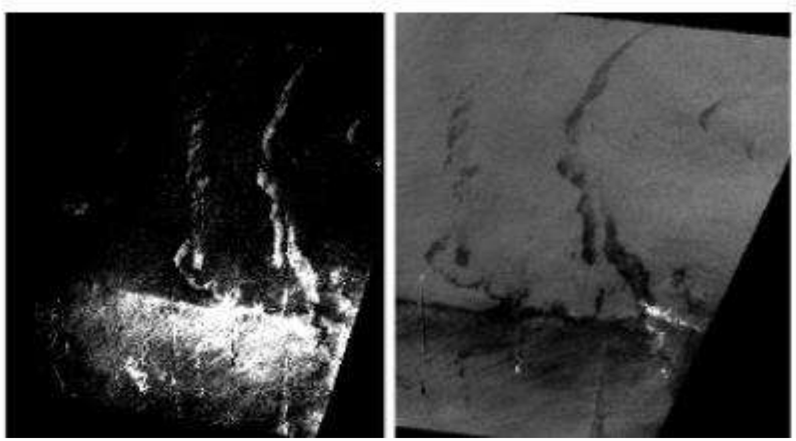
Oil Spill



RGB, Gulf of Mexico Oil spill, 2010

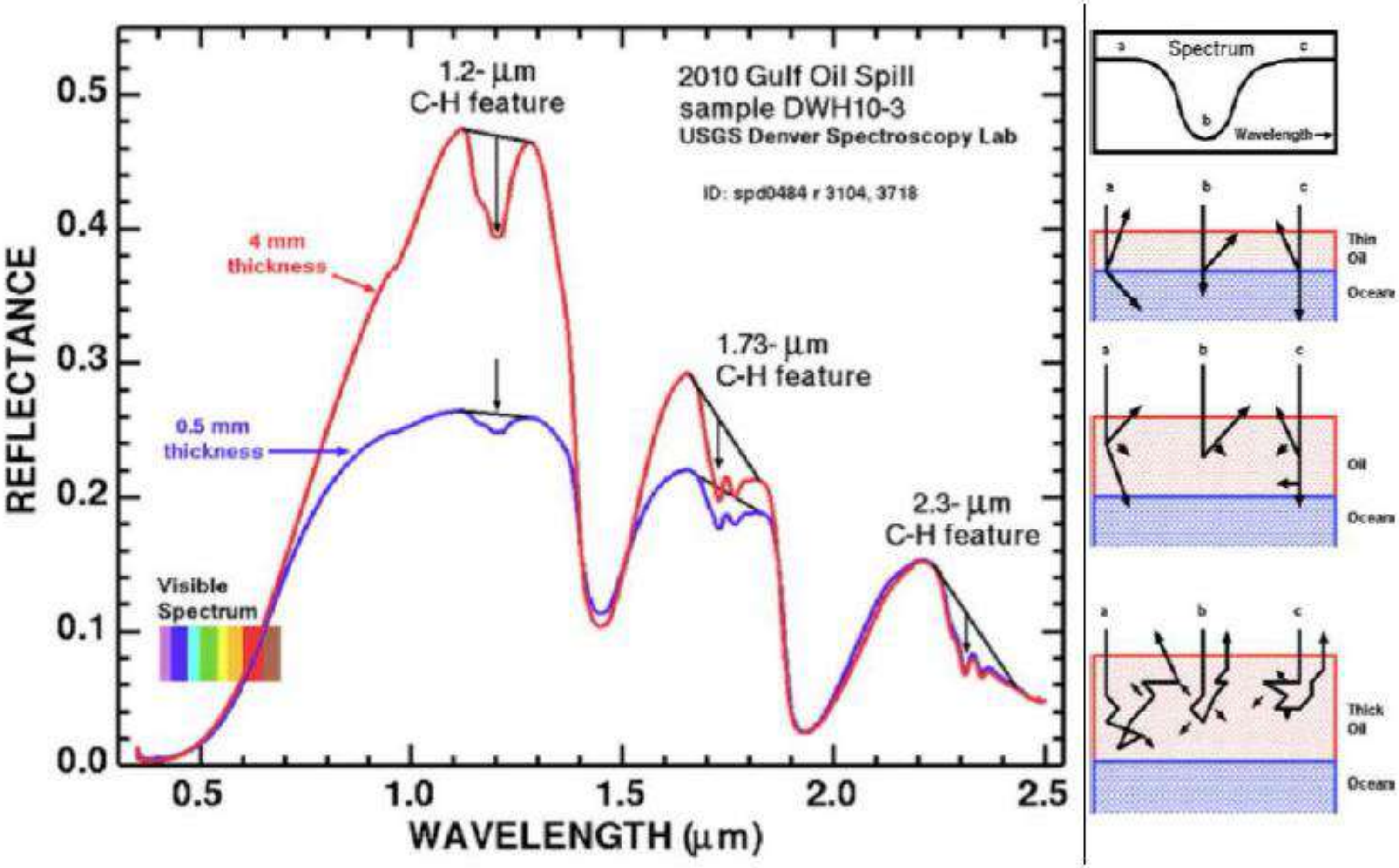


Radatr image of the Prestige disaster in Galicia, Spain



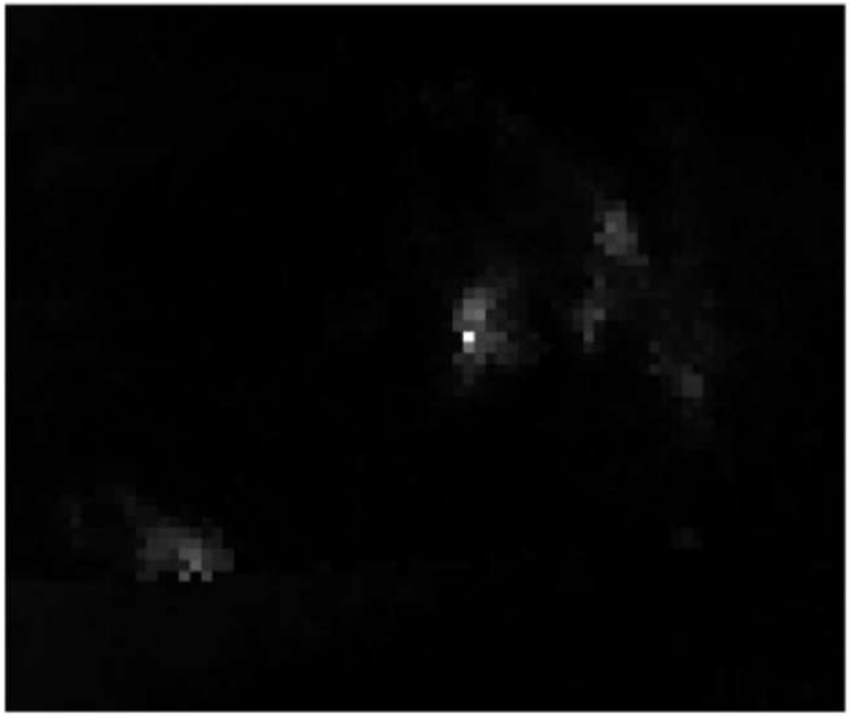
by Dimitris Sykas

Oil Spill



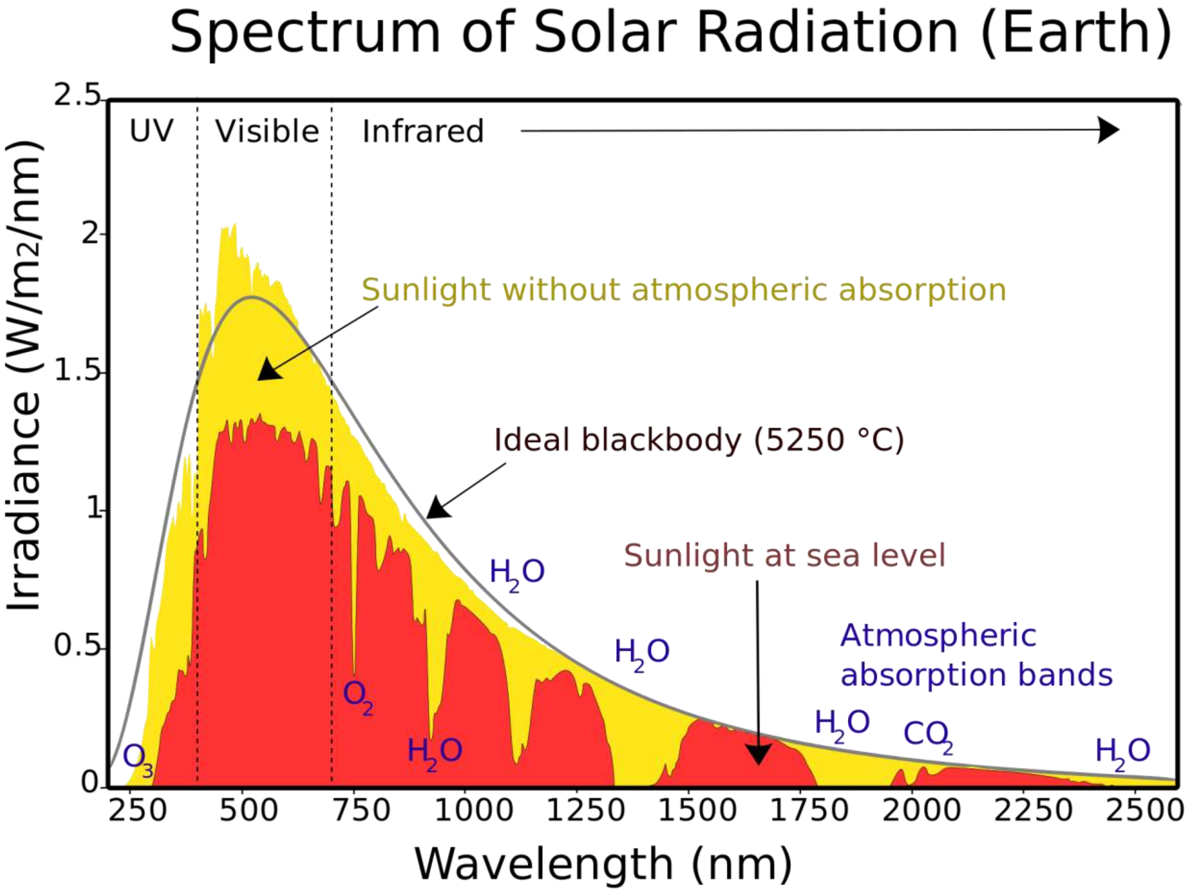
DLR

Oil Spill



Map of oil-spill relative thickness

Ozone Mapping and Profiler Suite (OMPS)

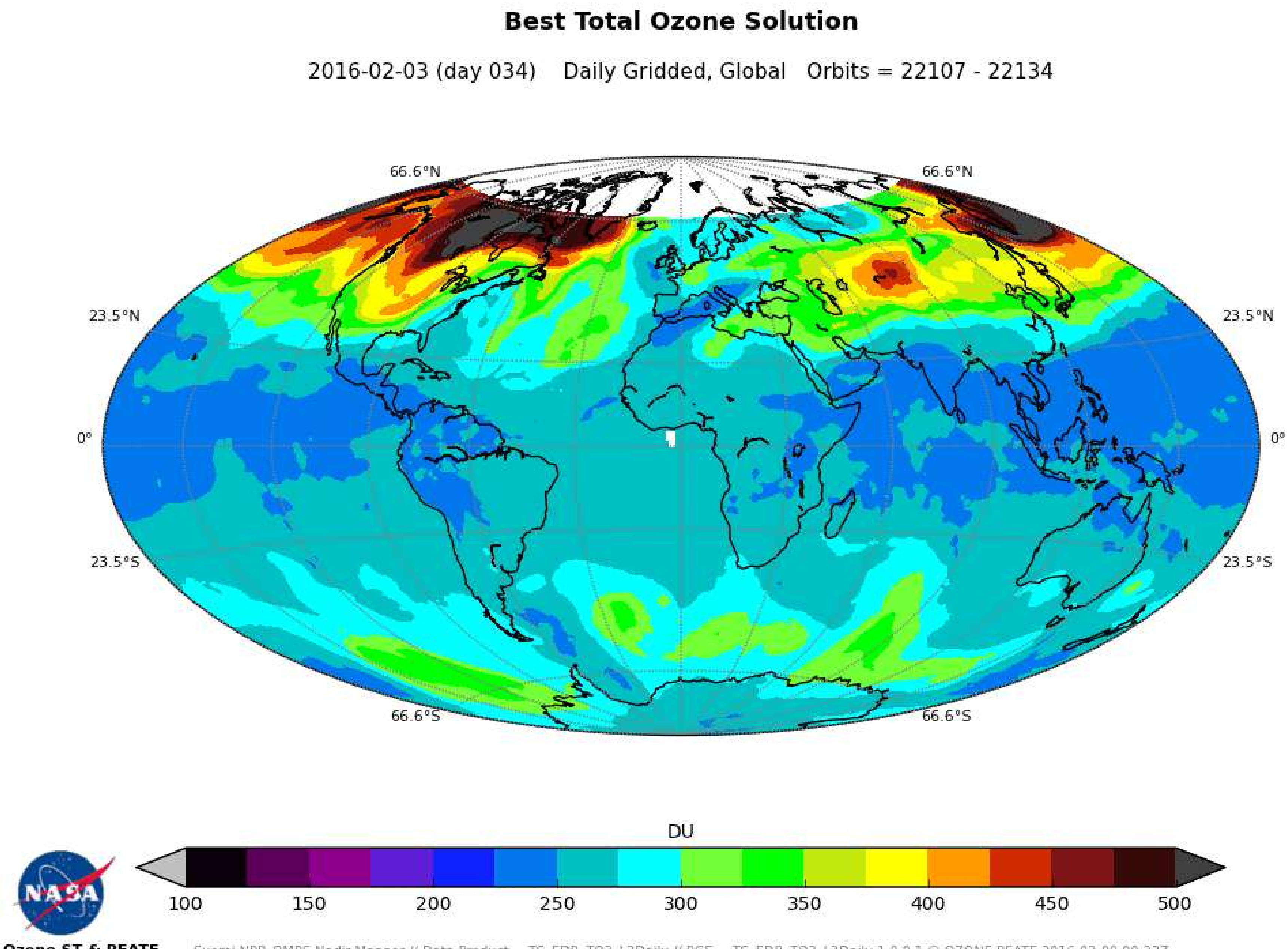


Spectral range: 290 to 1000

DLR

OMPS: a recent result (total atmospheric ozone column)

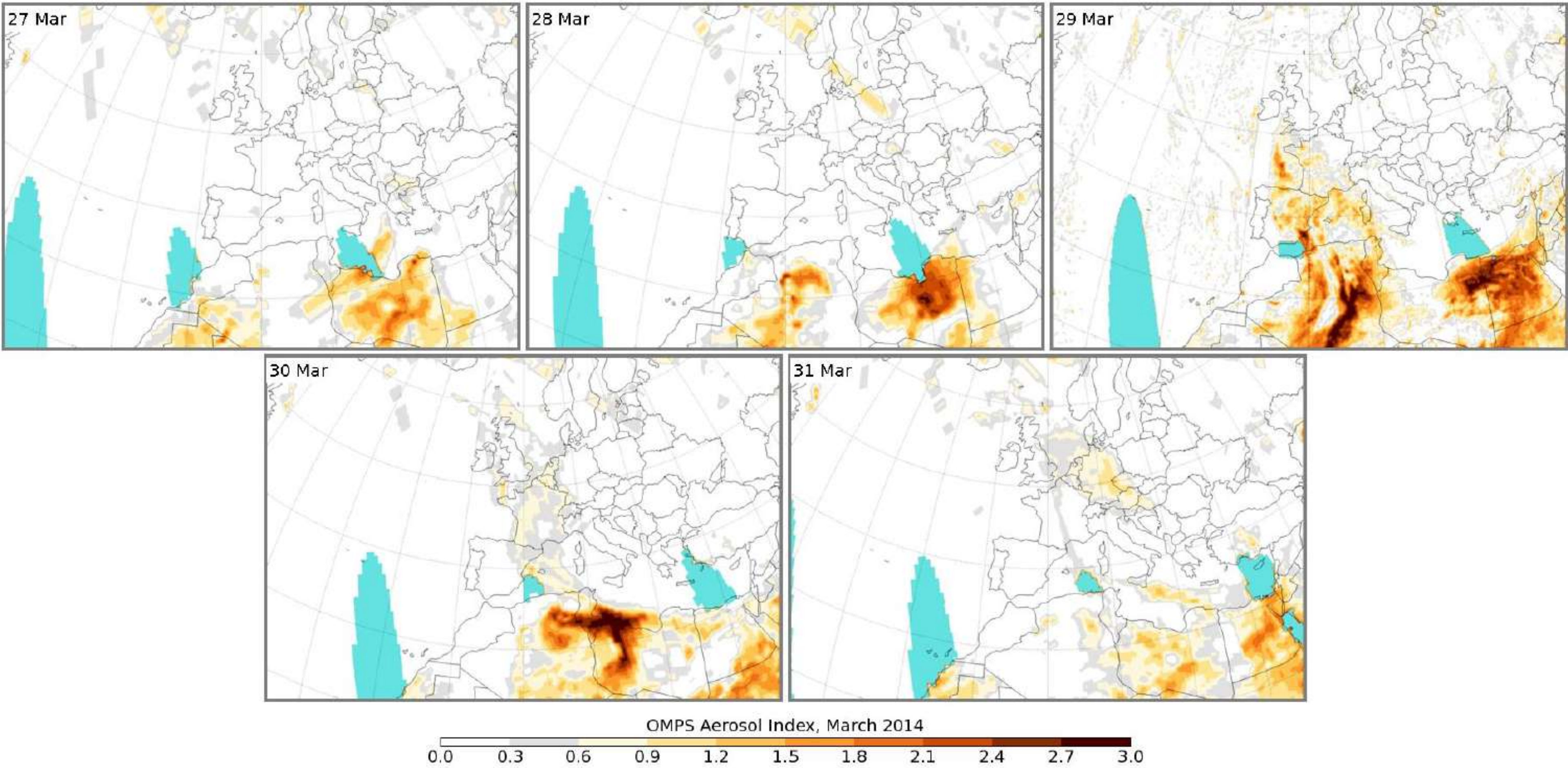
Source: NASA



Source: NASA

OMPS: a recent result (aerosol index 2015)

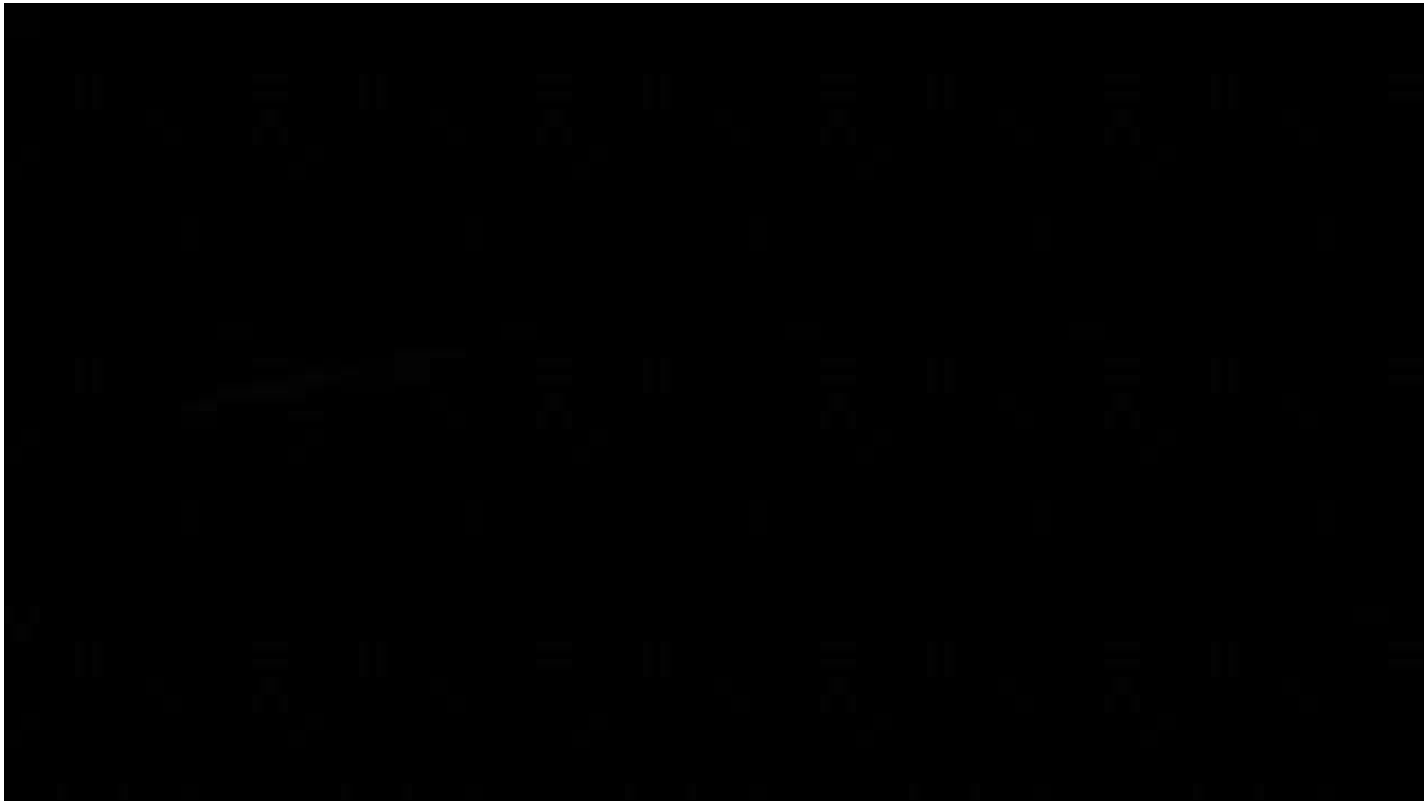
Question: what can you see?



DLR

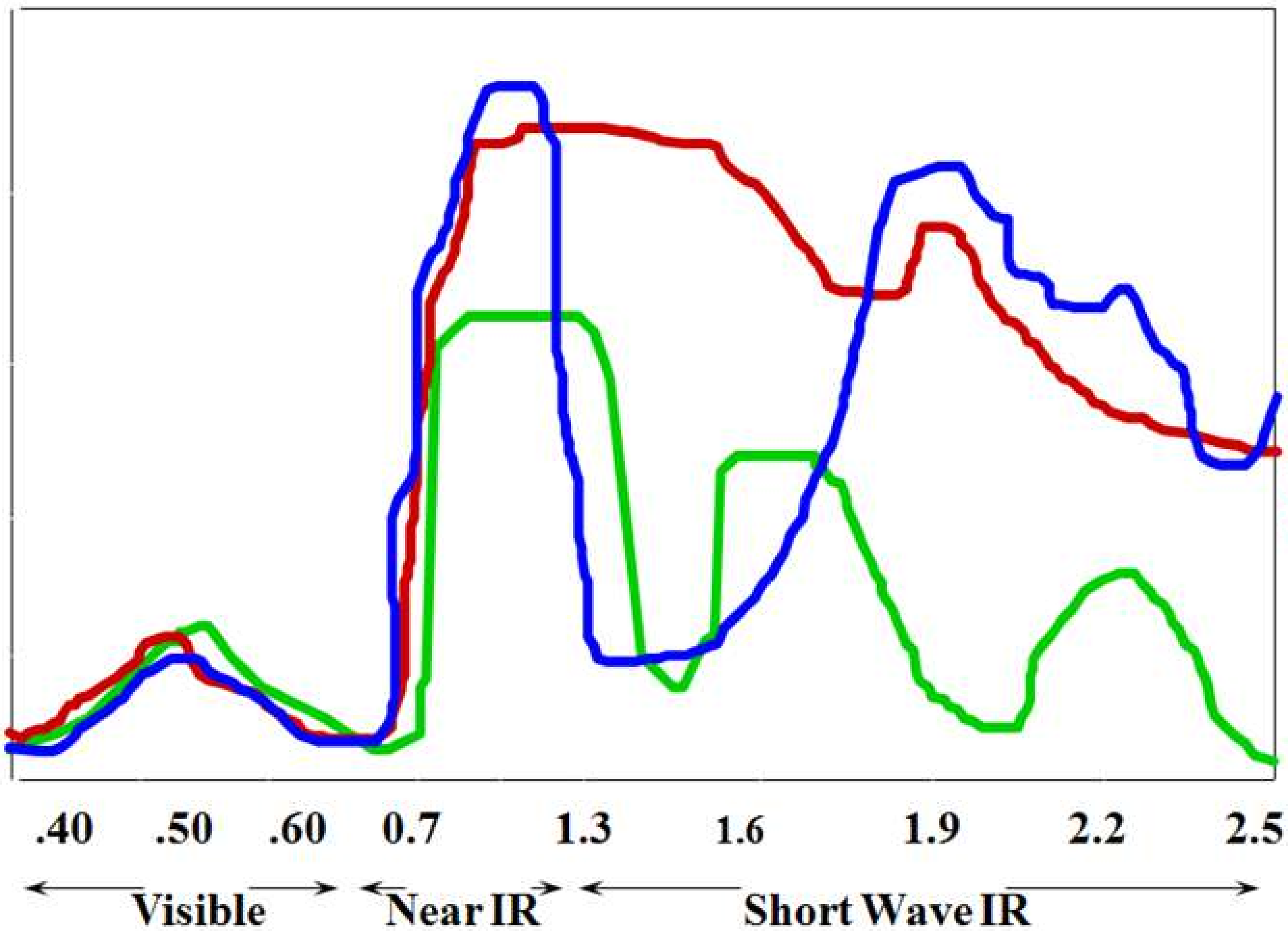
Source: NASA

OMPS: An interesting application for aerosol measurements



Adapted from Mark Elowitz

Detection of Camouflage



Support to Military Operations: the case of Osama Bin Laden

Abbottabad Compound, 2005

Source: http://www.globalsecurity.org

Analysis of materials from hyperspectral data

“the MH-60 helicopters made their way to Abbottabad […] Aboard were Navy SEALs along with tactical signals, intelligence collectors, and navigators using highly classified hyperspectral imagers.”

Why?

1. Identify materials: walls, roofs, gates,..?
2. Important targets can be marked by undetectable chemical agents. The hyperspectral sensor reveals where these targets are (if there is no occlusion)

Environmental Application: Acid Mine Drainage

- Acid mine drainages are waters with high acidity and dissolved metals content
- Result of the reaction between water and sulphide minerals
- The sulphides are oxidated when exposed to air and moved in large amounts (when a mine is exploited)
- In the USA for example 10.000 km of rivers contain metals such as cadmium, copper and arsenic
- Major environmental contamination between the 1940‘s and 1980‘s

Foto: Tat Wild

DLR

Acid Mine Drainage: Consequences

Problems in the reproduction of aquatic flora and fauna

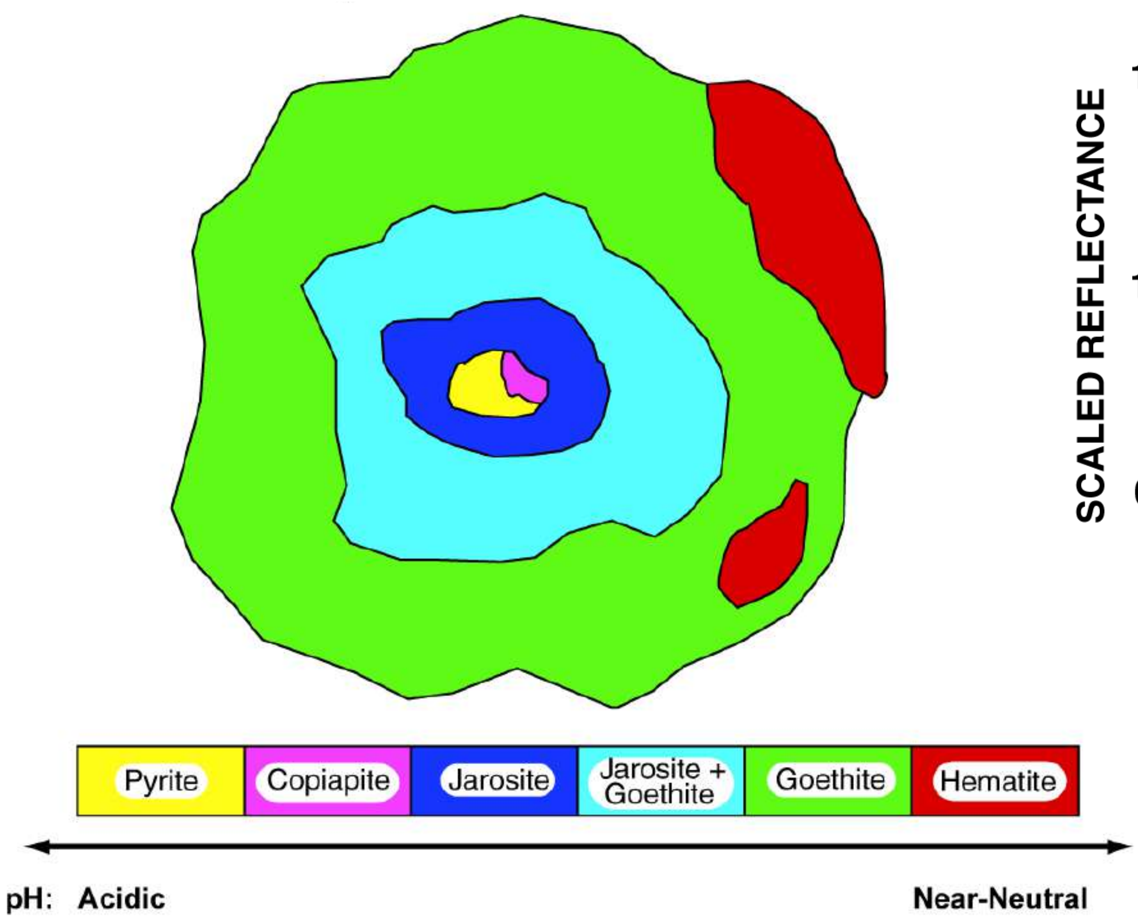
Damage to ecosystems

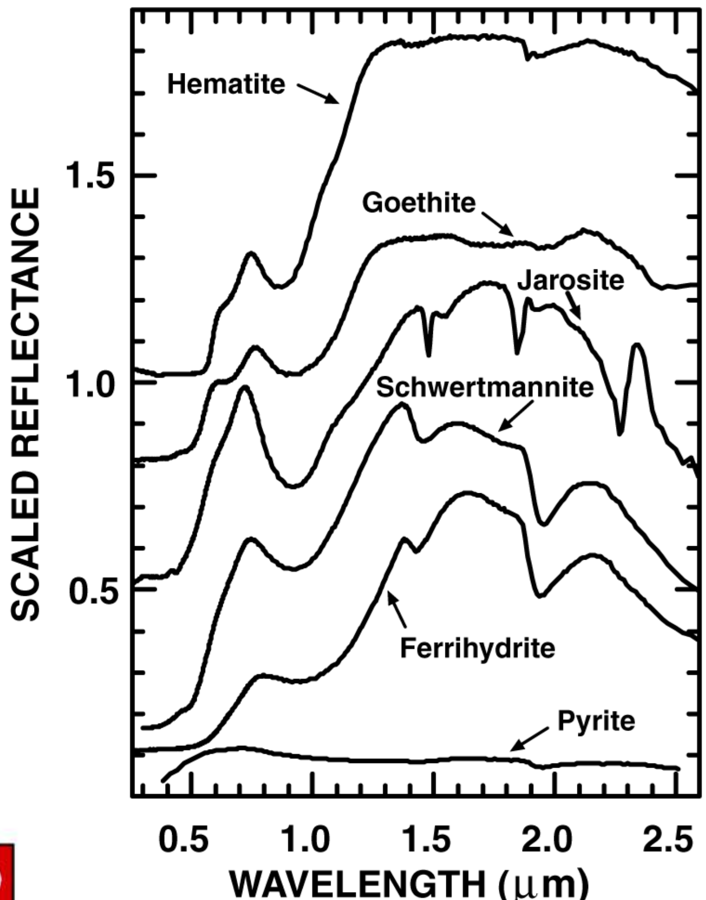
Corrosion of bridges bases

Contamination of drinking water

Map for contamination from acid mine drainage

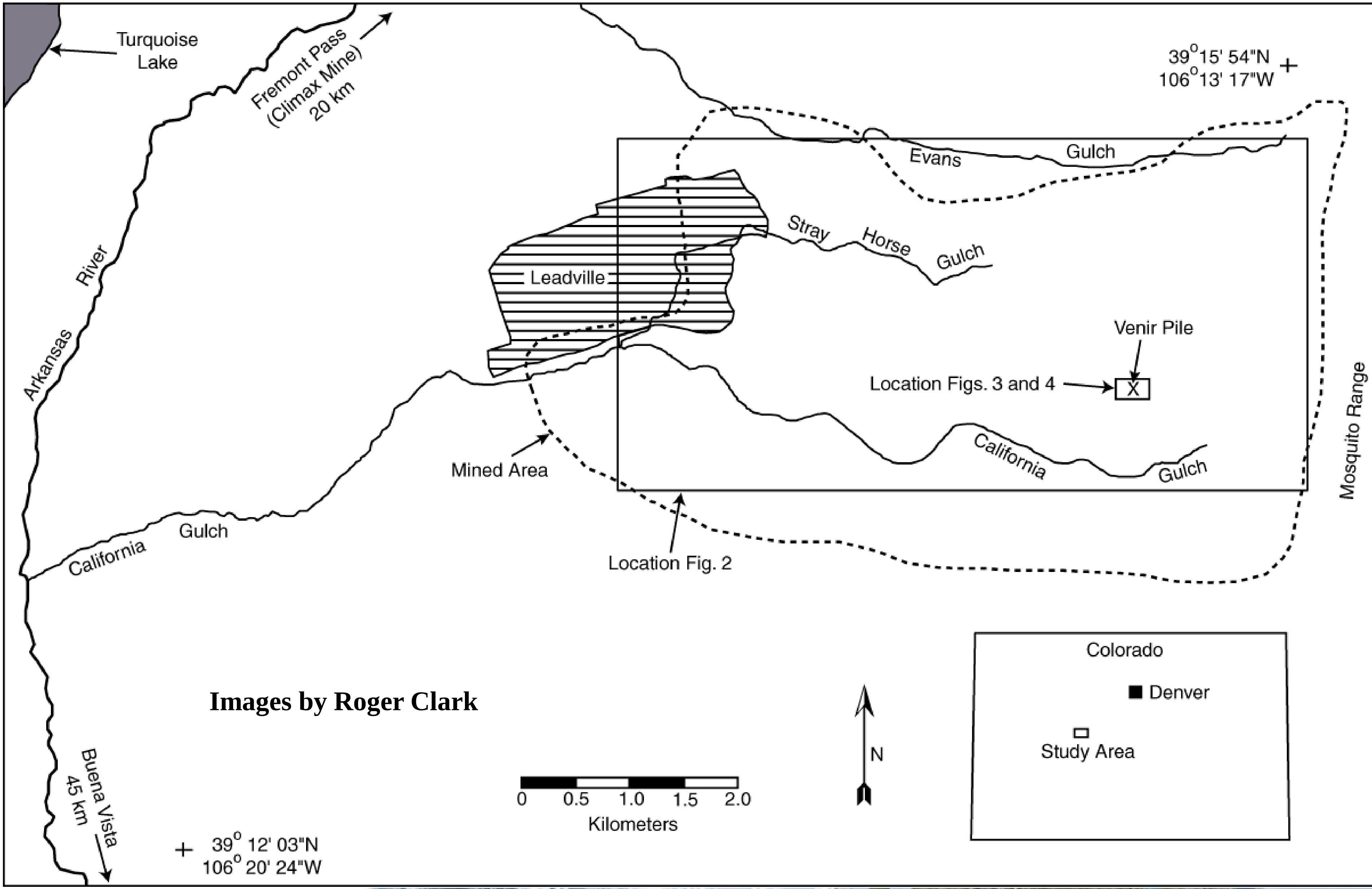
- Sulphides are associated to different metals





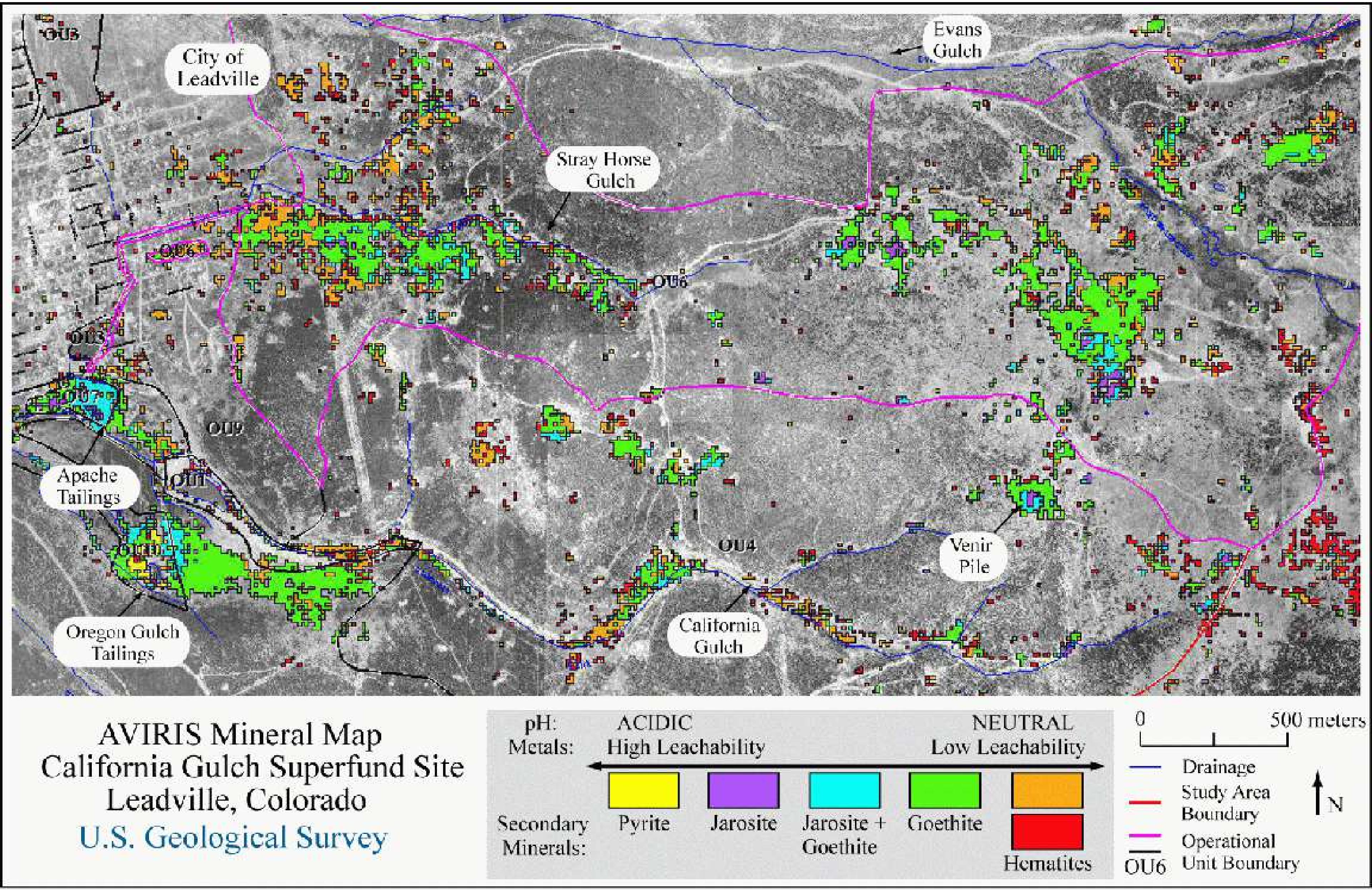
Images by Roger Clark

Contamination of mining sites in Leadville, Colorado: study area



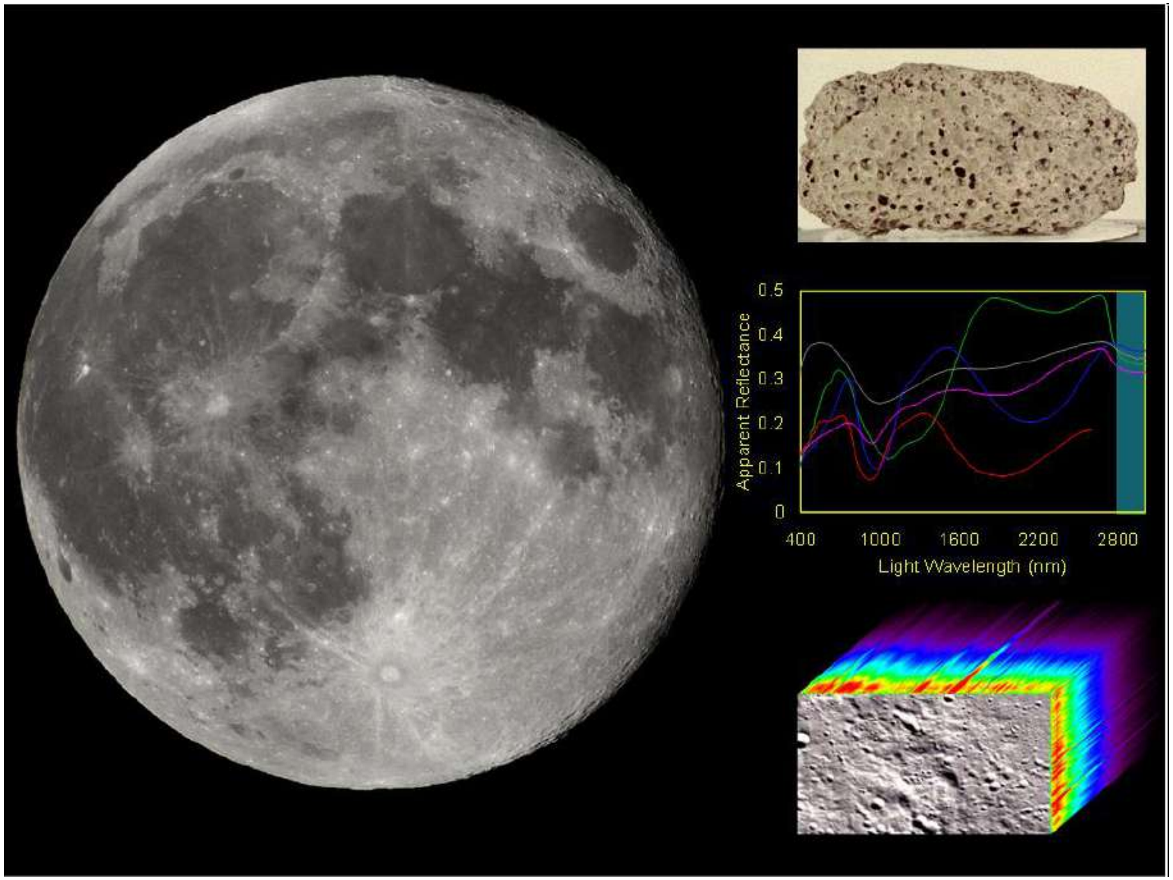
Contamination of mining sites in Landville

Images by Roger Clark



images by NASA

M3 (Moon Mineralogy Mapper)
A hyperespectral mission on the moon



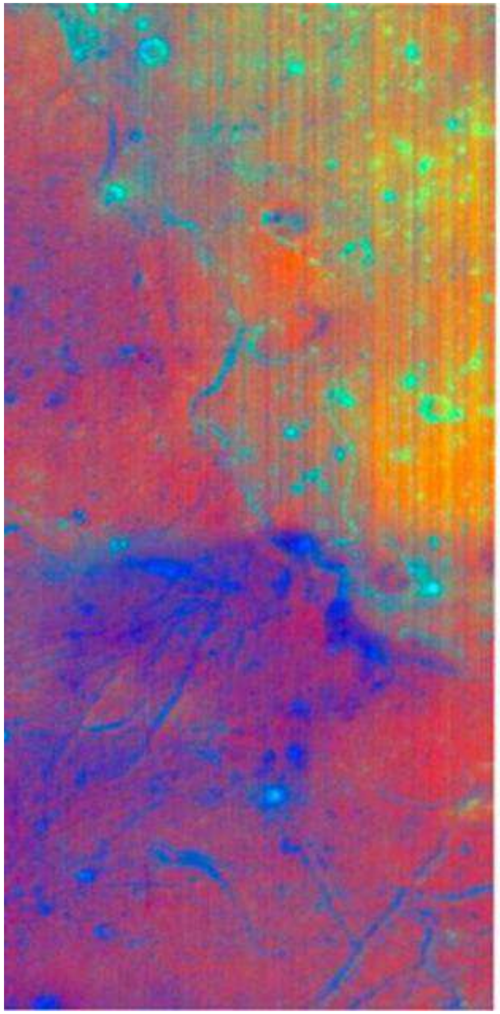
DLR

Mineralogy of the moon

– Transitions between red and blue show variations in the composition of the rocks

– In green zones rich of iron

– Vertical lines are artifacts (information not present in reality deriving from interferences or other sources)



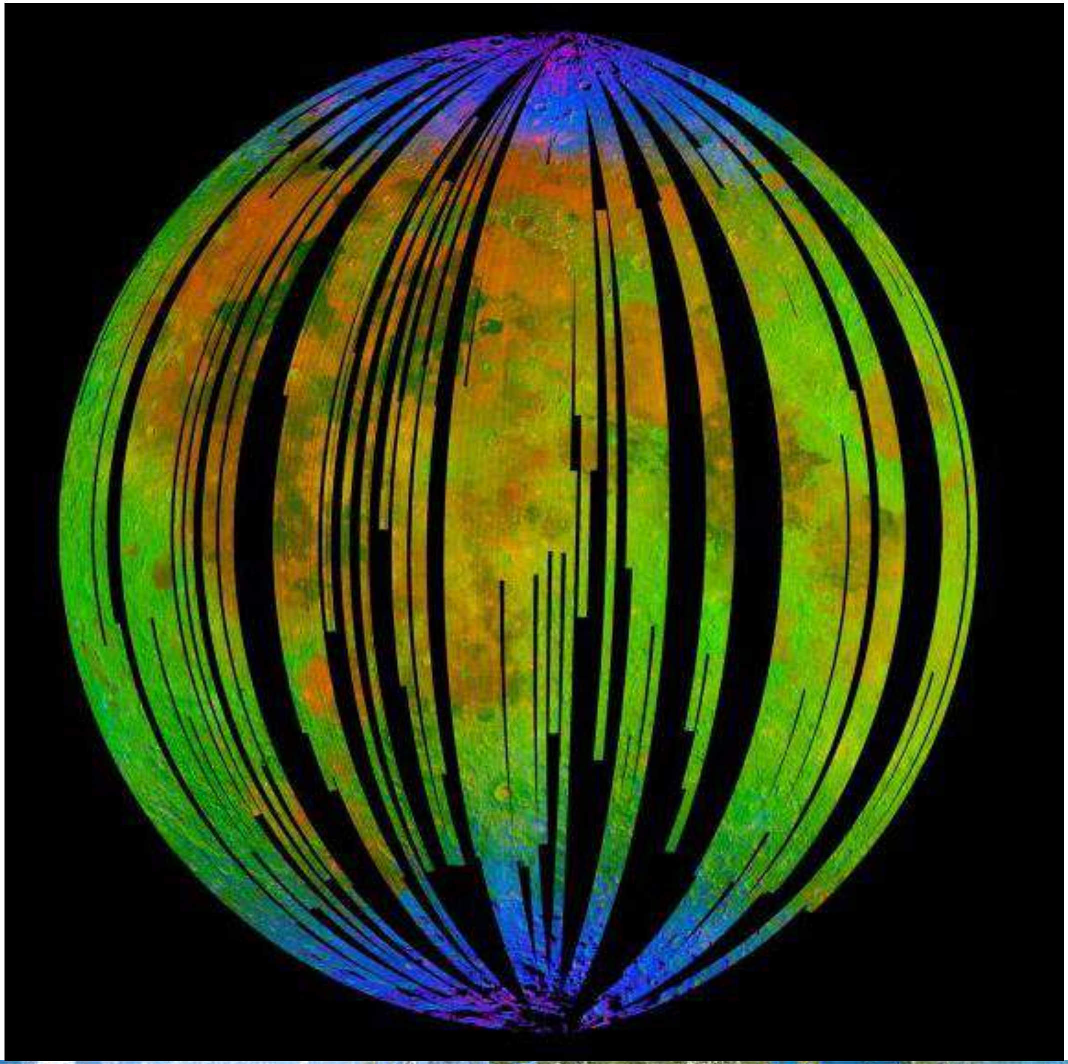
Map basad on Hyperespectral Data

Image acquired in the thermal infrared

DLR

Water on the moon?

- A 10 years long debate
- In blue zones in which evidence for the presence of water has been found
- Available for the first spacemen set to colonize our satellite
- To extrac a liter of water it will be needed to process one ton of rocks ☺



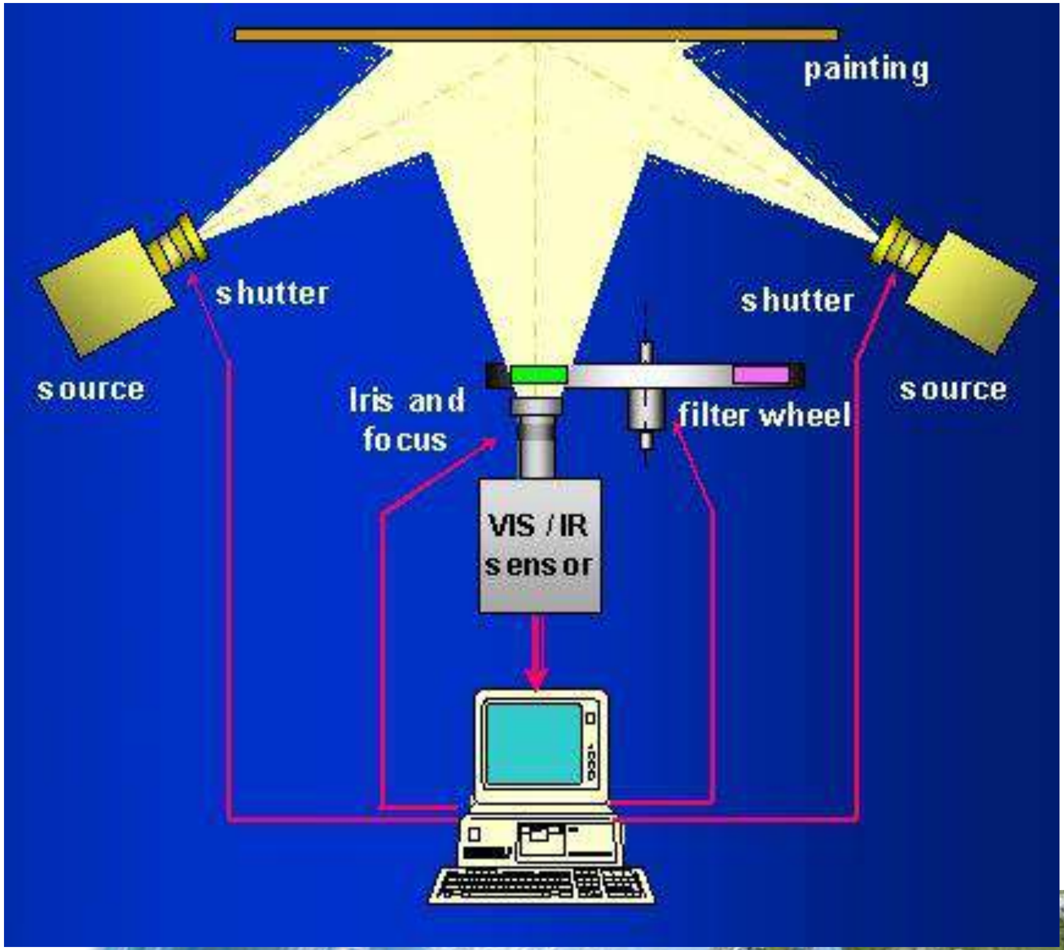
Hyperspectral Imaging applications in art and archaeology

ADAPTED FROM OMER PAPARO, 2013

DLR

Spectral imaging systems

– Working scheme:



DLR

Spectral imaging systems

– Measurement at the Uffizi Gallery, Florence, Italy - Leonardo room

Pigment identification

- Introduction – What are paintings made of?
  - A pigment is a colored material ground into a fine powder
    - After the grinding it is suspended in some type of media that acts as a binder to hold the dry pigments pigment together
      - E.g. linseed oil for oil paints
  - Over the eras, many different pigments were used

Pigment identification

– E.g., the late gothic palette

Hieronymus Bosch's Palette (1450-1516)

1. Yellow Ochre
2. Lead Tin Yellow
3. Azurite
4. Vermilion
5. Carmine Lake
6. Malachite
7. Copper resinate
8. Lead White
9. Ivory Black

Pigment identification

– E.g., the early Italian Barocco palette

Caravaggio (1571-1610)
Palette
1. Umber
2. Yellow Ochre
3. Red ochre
4. Vermilion
5. Lead White
6. Carbon black
7. Lead tin yellow
8. Copper resinate

DLR

Revealing hidden information

- For paintings:
  - Maximum penetration of most paints can be achieved at wavelengths of around 2 µm
  - At wavelengths around 1-2 µm, the common drawing materials, namely iron gall ink and sepia, become invisible
  - Can use this to see underdrawings and preparatory sketches

Revealing hidden information

– A Byzantine icon at 640nm (a) and 1000nm(b)

DLR

Revealing hidden information

– Pablo Picasso –
"The Tragedy"

SLIDE BY OMER PAPARO

Revealing hidden information

– The optimal spectral window to visualize such features varies with the material used as well as the thickness of the paint layer

•Man, ~1100nm

•Horse, ~1350nm

•Sketch, ~1600nm

SLIDE BY OMER PAPARO

DLR

Revealing hidden information

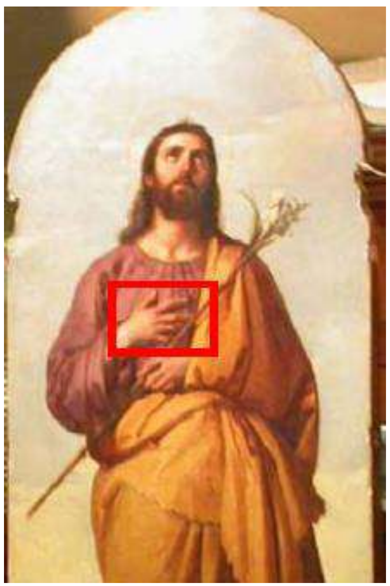
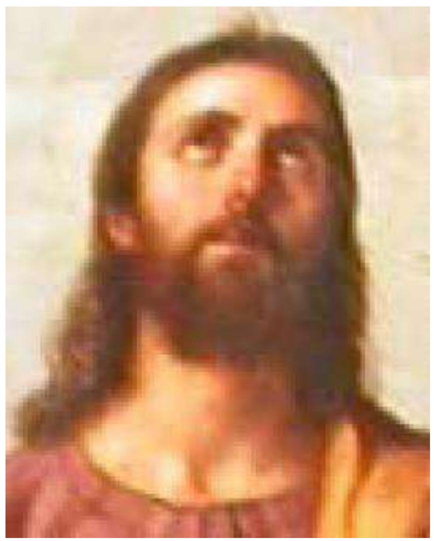
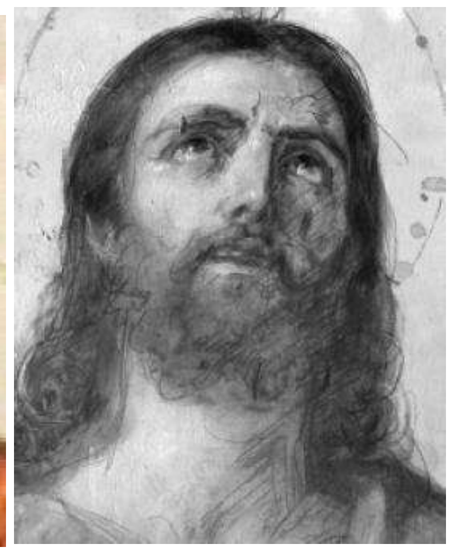
– A painting by Sellaio


520nm


885nm


RGB

Revealing hidden information

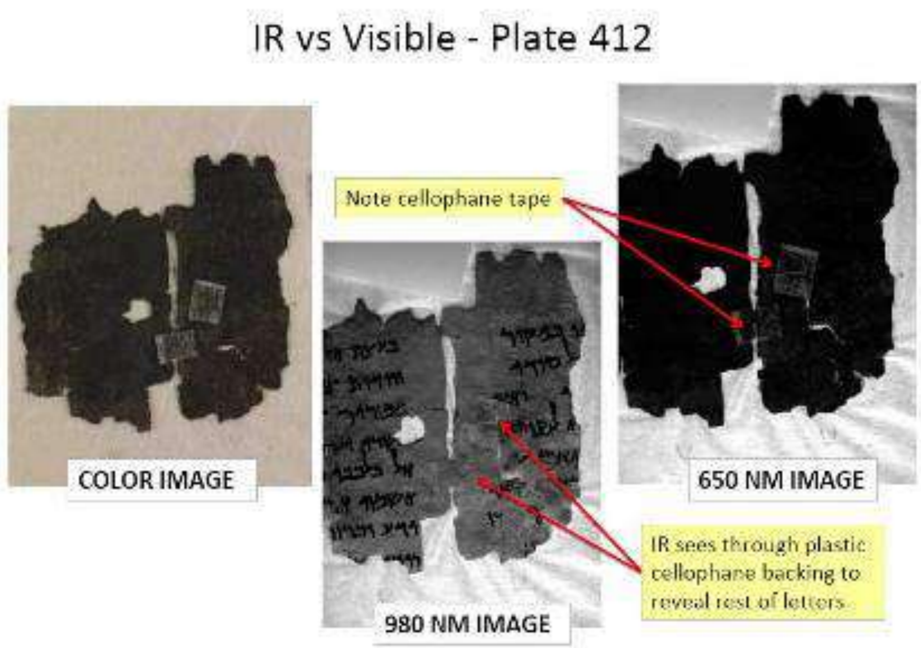
DLR

Revealing hidden information

- Studying archaeological manuscripts
  - "Soft media" ancient documents (i.e. documents written on soft materials such as leather or papyrus) are often unreadable
    - The carbon-black ink is faded beyond recognition
    - The carbon-black ink indistinguishable from the surface
    - Not to mention the document itself is found in shreds

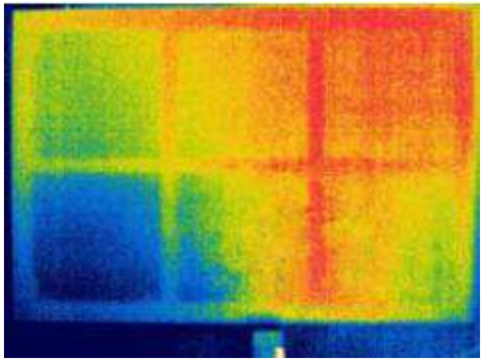
Revealing hidden information

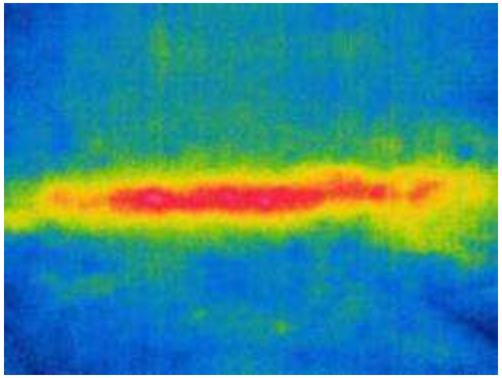
- Studying archaeological manuscripts
  - Can use IR to read previously invisible texts and scripts
    - The dead sea scrolls can only be seen through IR light



Art conservation

- Conserving Paintings
  - Conservation monitoring
    - Can identify continual damage to paintings, for example
      - From a lamp in front of the painting
      - From a pipe going through the ceiling





Food

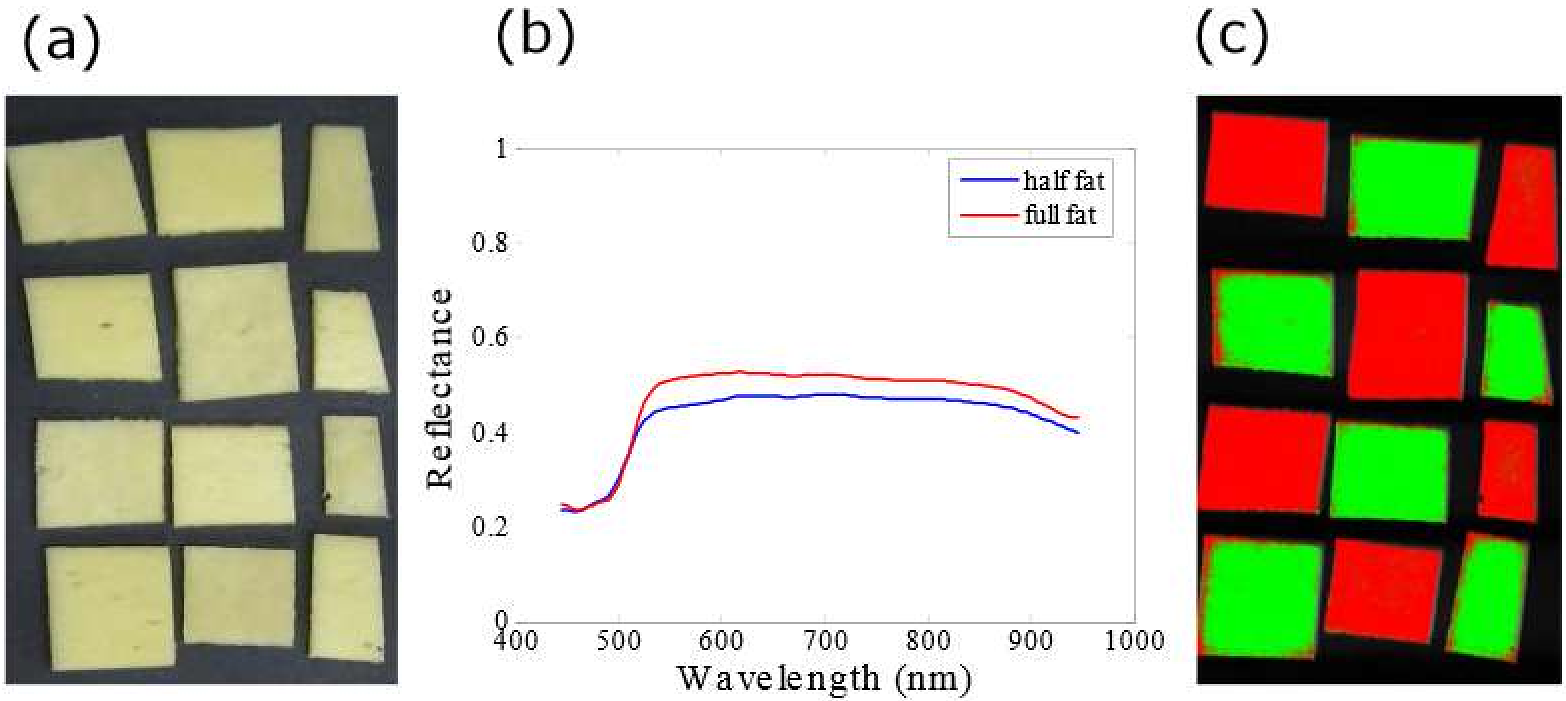


Figure 5a: RGB image of cheese samples studied; (b) Mean spectral of half and full fat cheese samples; (c) cheese classification map (red = half fat, green = full fat) obtained using Spectral Angle Mapper algorithm.

A. Mc Gowen

Food

Cooked chicken (RGB)

Detection of blood spots (in red)

Food

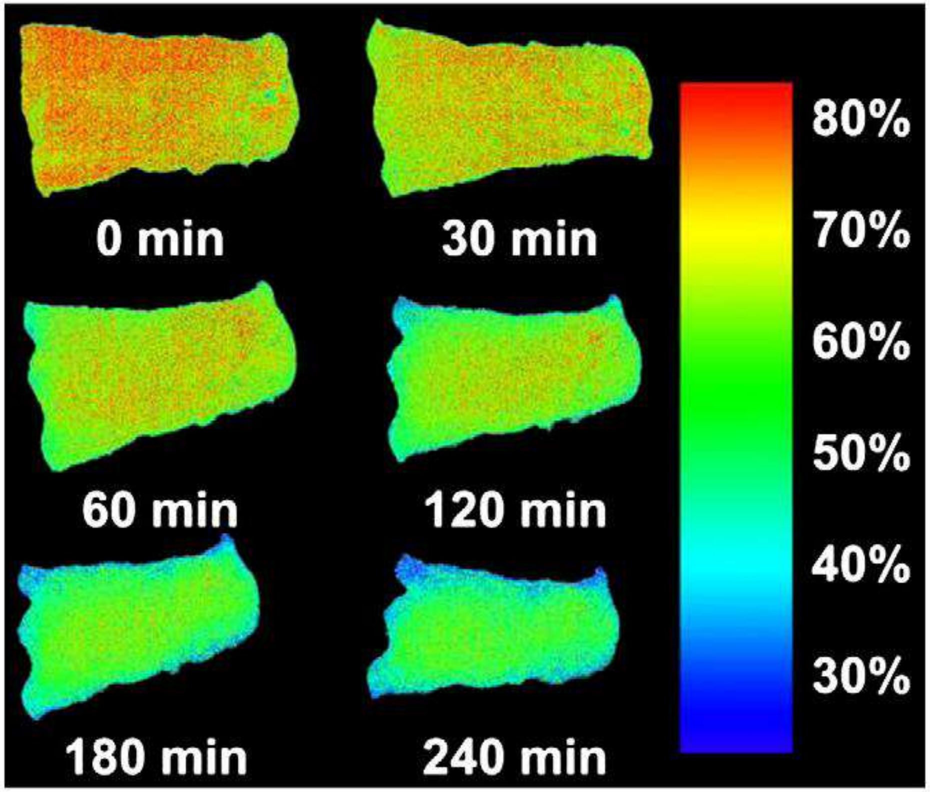
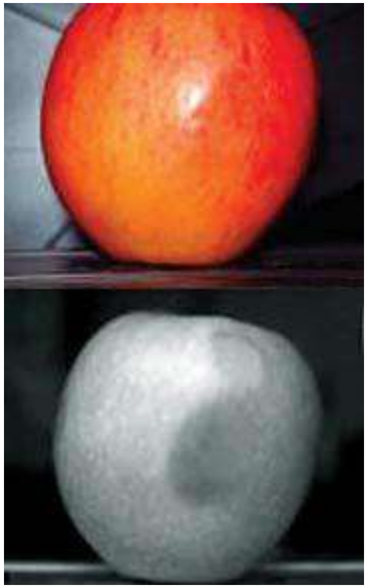


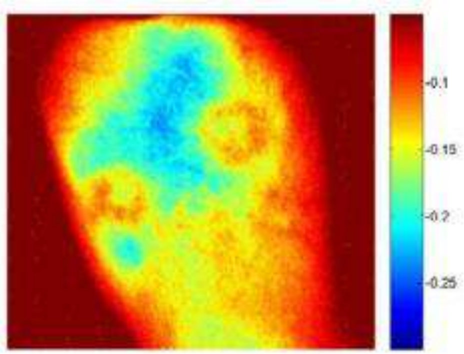
Fig. 3. Visualization of water distribution of beef slice during dehydration (Wu et al., 2013).

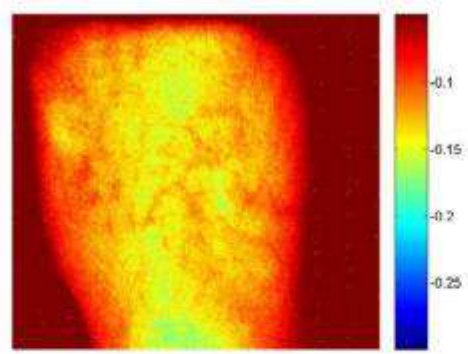
Food



RGB & Infrared picture of an apple
(invisible defects)

Medicine & Health





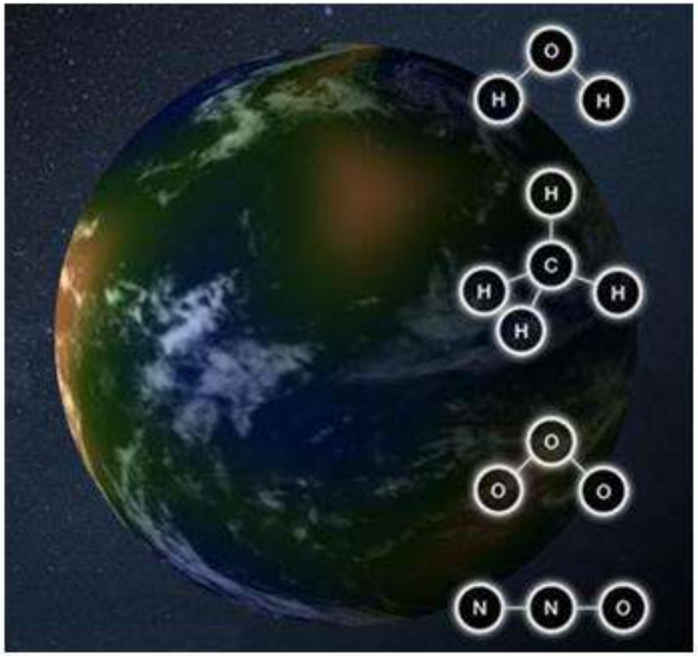
- Tissue sample analysis
- Blood analysis
- Chemical samples
- Diagnostics (e.g. skin diseases, cancer)
- Skin characterization
- Cosmetics
- Fluorescence imaging spectroscopy

- *This image shows the bilirubin levels in bruised skin after 66 hours (left) and 180 hours (right) based on analysis of HySpex data. [Lise L. Randeberg, NTNU]*

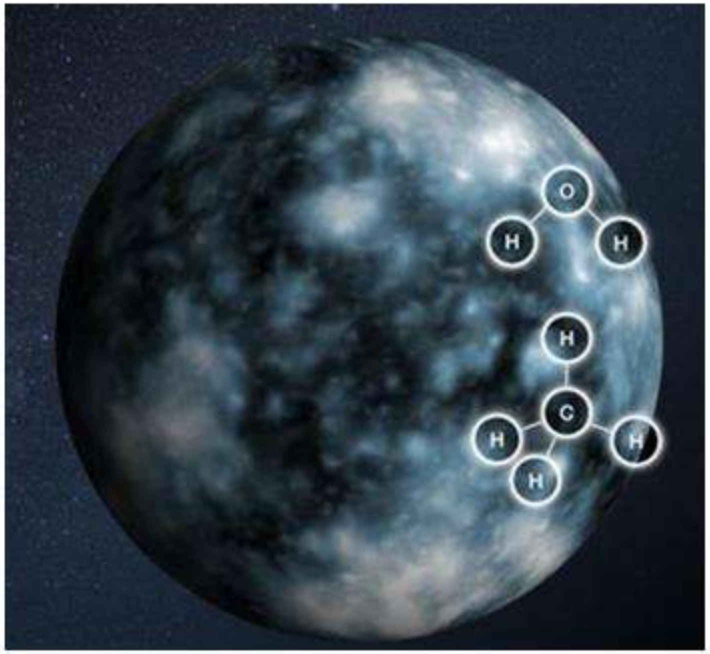
DLR

Are we alone in the universe?

Imaging Spectroscopy – Biosignatures on Earth-like Exoplanets



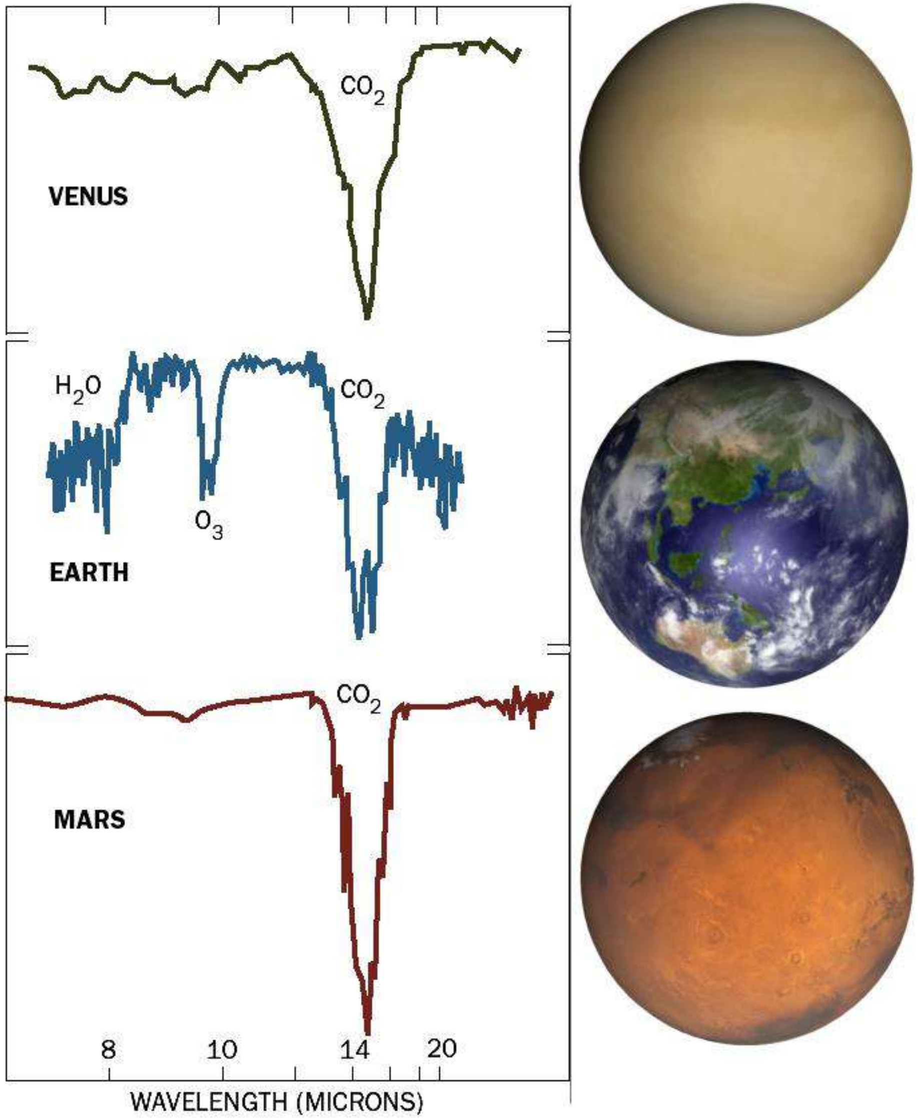
A hypothetical earth-like planet that shows water, ozone, nitrous oxide, and methane in its spectrum could be inhabited by plant life, bacterial life, and intelligent life. The presence of ozone indicates that oxygen must also exist in the atmosphere, since ozone is created from UV radiation reacting with oxygen.



A hypothetical planet showing methane and water in its atmosphere suggest that the planet is a good candidate for the evolution of life, assuming it does not already exist. Both plant life and bacterial life would be expected based on the biosignatures.

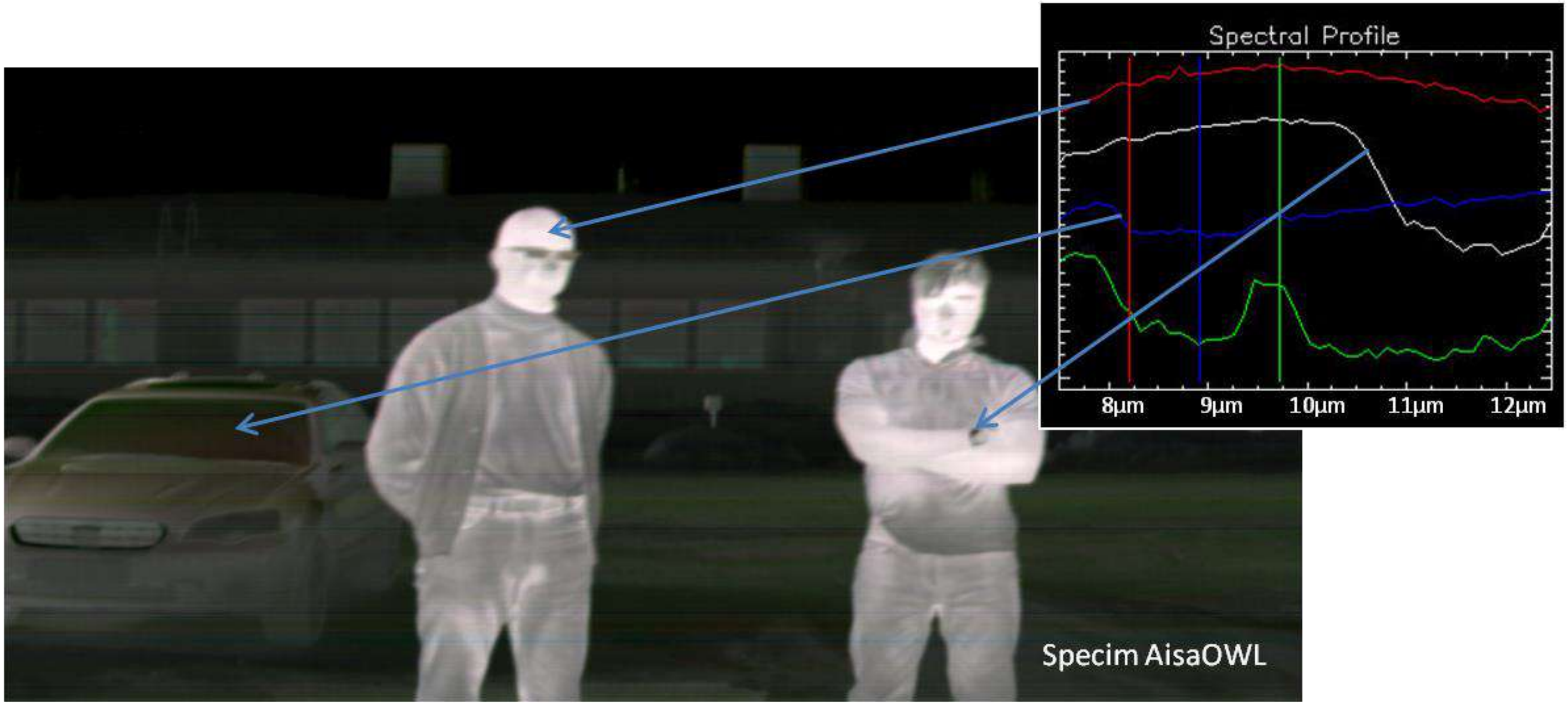
DLR

…and in our solar system?



DLR

Thermal Hyperspectral: Surveillance Systems



Source: http://www.photonik.de

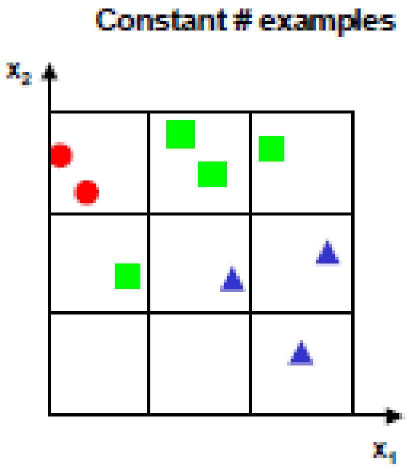
Tutorial on Principal Components Analysis

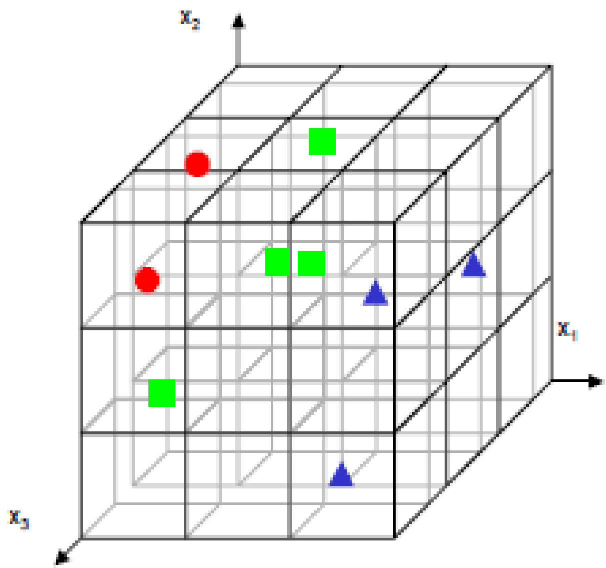
Daniele Cerra, DLR

DLR

Curse of Dimensionality

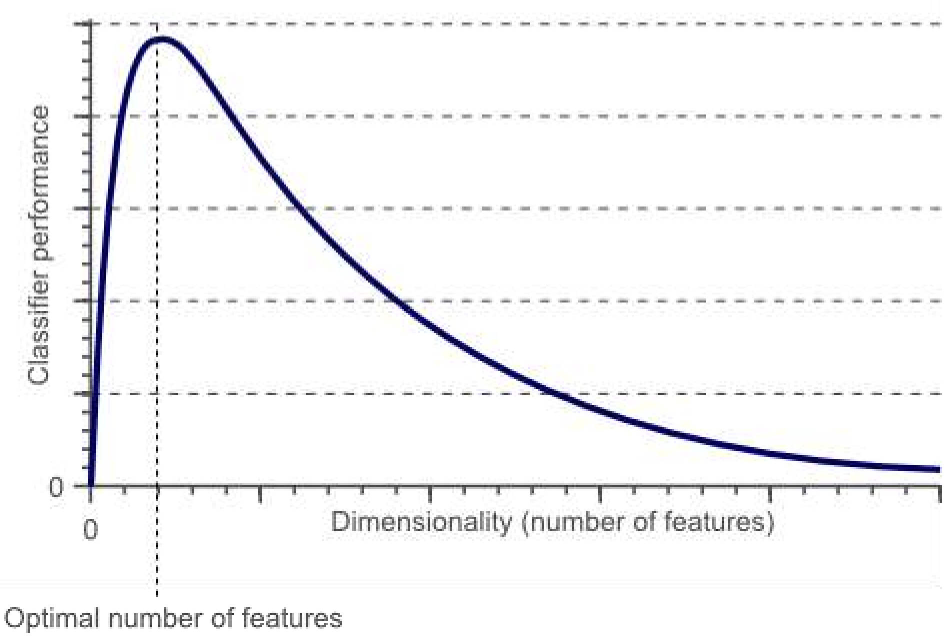
– Classification problem: 3 classes



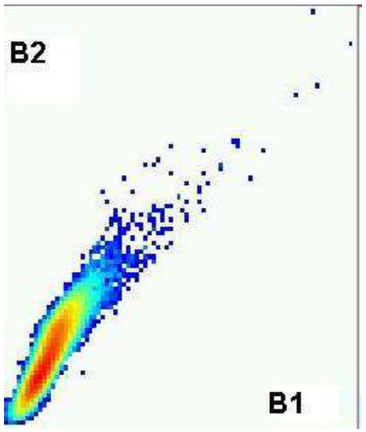


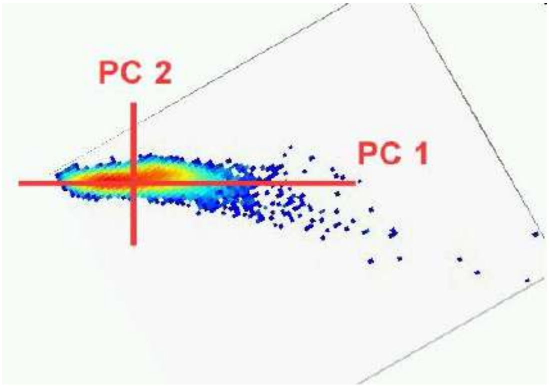
Curse of Dimensionality

– Classification problem



Really the whole spectrum? Second part

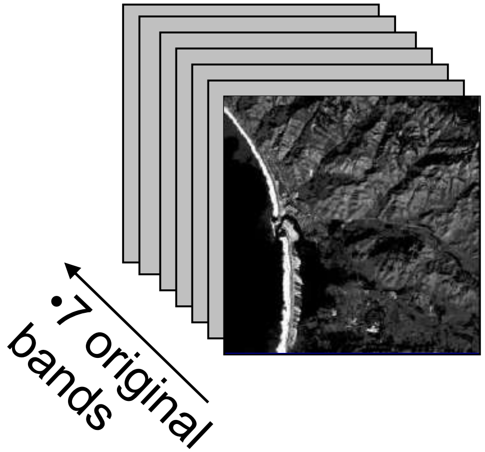


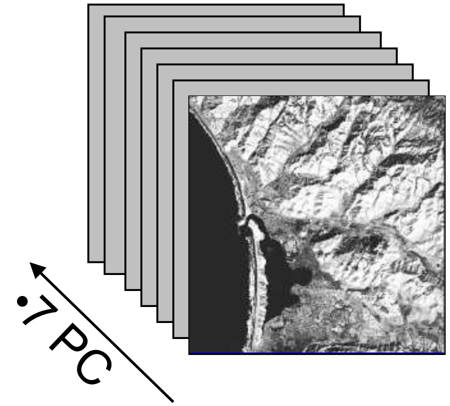


- Correlation between bands
  - Information redundancy

- We "change" the bands to have independent information in the bands
- By rotating the feature space
- Then we select new "bands" with high variance only

Principal Components Analysis





A typical Landsat Image

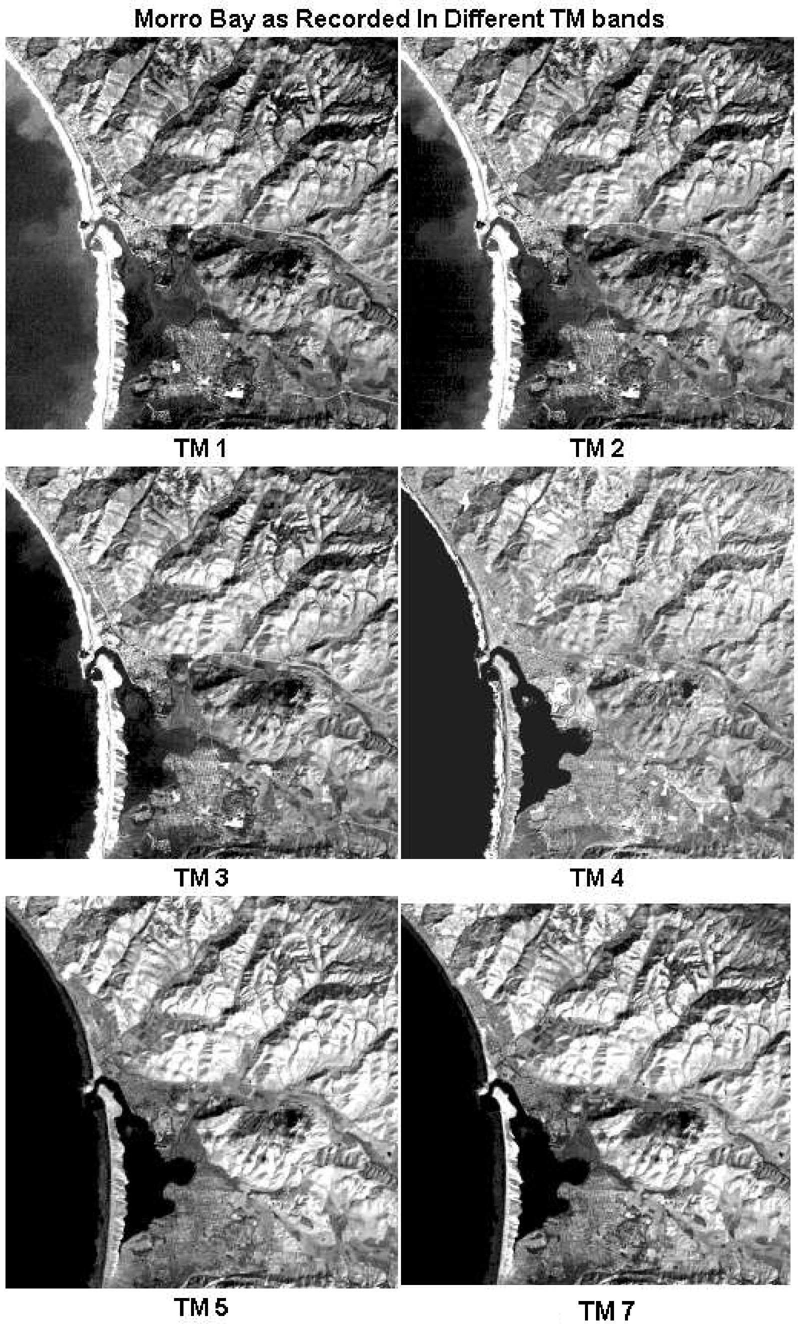
•Morro Bay, California, USA

•RGB Combination

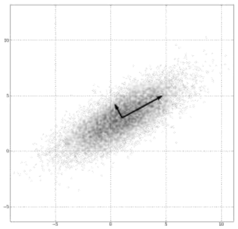
•(First three bands from the Landsat TM scene)

Let‘s have a look…

- A typical Landsat image has 7 bands
  - Blue
  - Green
  - Red
  - Near Infrared
  - Shortwave Infrared
  - Thermal Infrared
  - Shortwave Infrared2
- Do these bands look really different?
- How much redundant information is there?



Principal Components Analysis

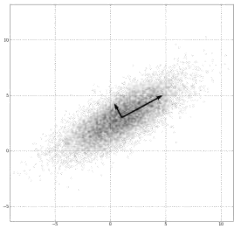


– *Principal Components Analysis* (PCA) is a technique used to reduce multidimensional data sets to lower dimensions
  – It describes *n*-dimensional data with a set of *p* synthetic variables, with $p < n$
  – The new variables are uncorrelated and are called *Principal Components* (PC)
  – This process leads to some information loss
    – PCA ensures that this loss is minimal

– Also known as:
  – Karhunen-Loève transform
  – Hotelling transform
  – Proper Orthogonal Decomposition (POD)

DLR

Principal Components Analysis



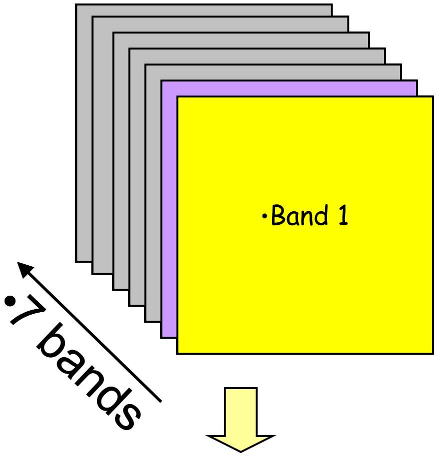
- PCA is widely used in remote sensing → dimensionality reduction aids data exploration

- It reveals the internal structure of the data by ignoring not relevant information
  - It highlights similarities and differences within the data

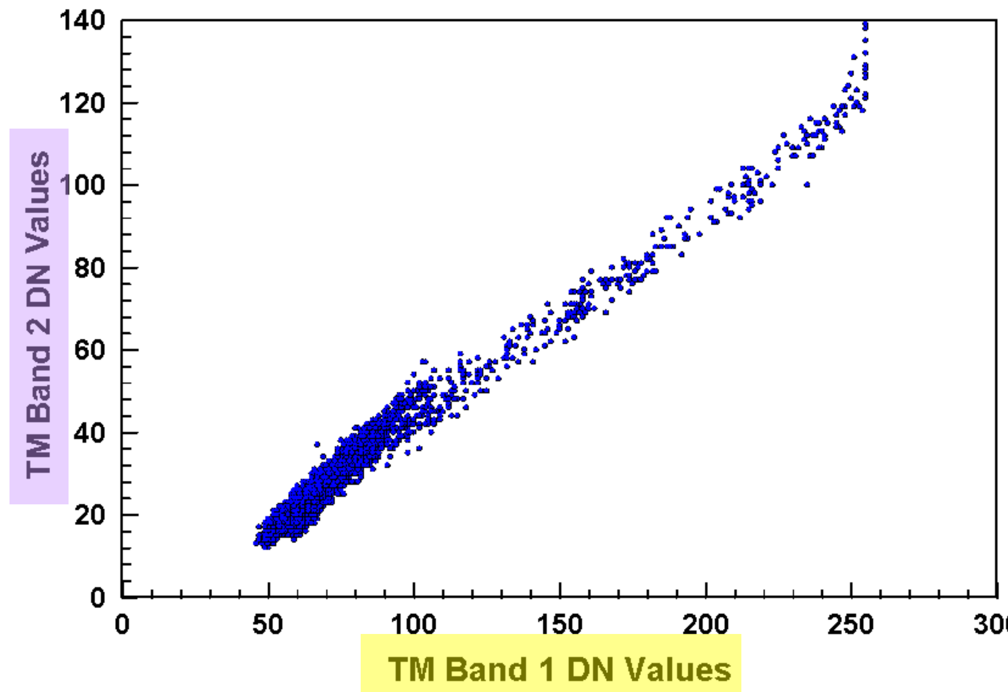
- First of all, let's see how PCA can be useful…

Let's compare some bands…

Morro Bay (California, USA) Landsat Scene



•RGB Combination



•Band 1

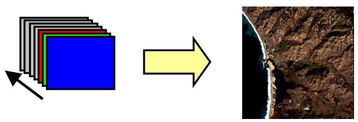
•Band 2

- What do you understand from the scatter plot?
- Can we predict the value of band 2 knowing band 1?

Dimensionality reduction: why?

– How to visualize multidimensional data?

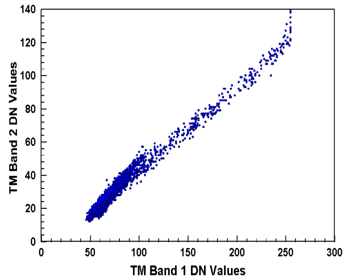
  – In the previous example, out of a 7-band image only 3 bands could be visualized



– ↑ Data → ↑ Information? Not always..

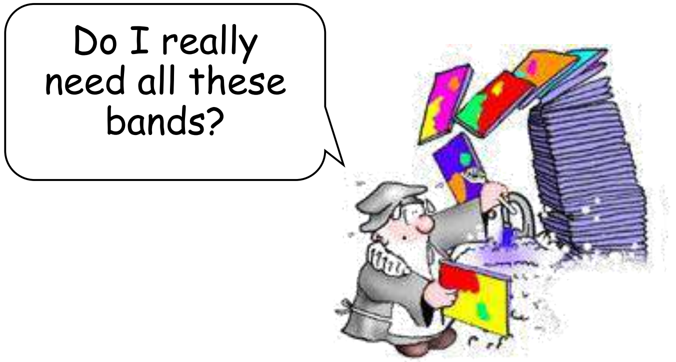
  – Redundancies

  – In the previous plot we can predict the value of band 2 on the basis of band 1



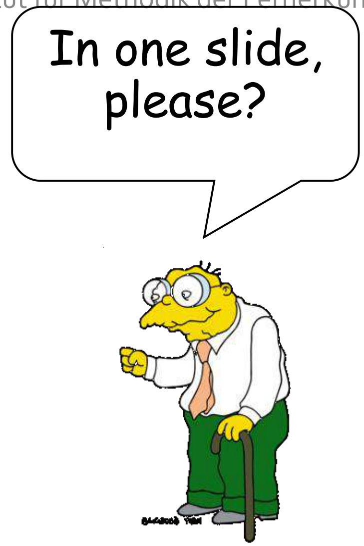
– We would like each band to contain relevant information

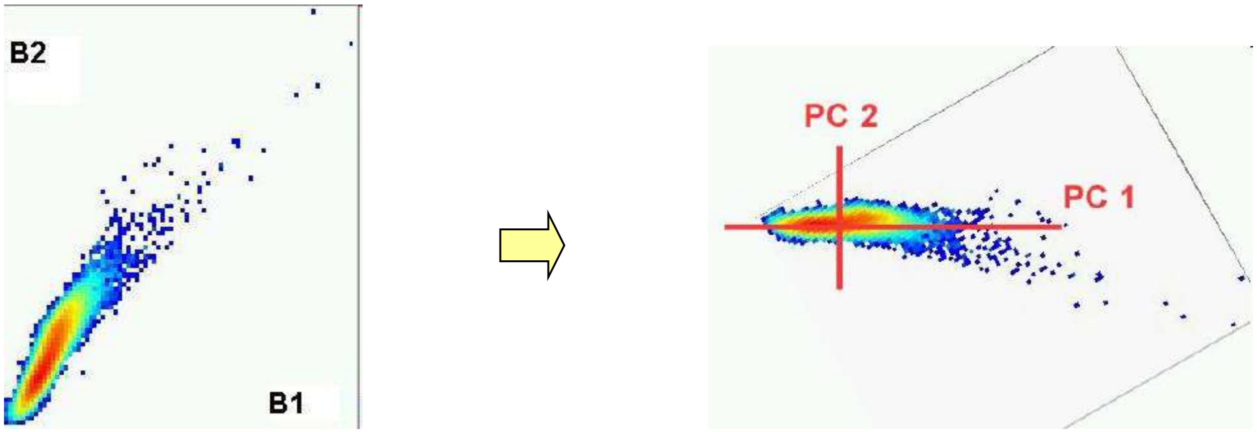
  – A decorrelation of the bands may help at analyzing the images



DLR

How does PCA work?





– PCA is a methodology for transforming a set of correlated variables into a new set of uncorrelated variables

– Achieved through a rotation of the original dimensions/axes to new orthogonal axes

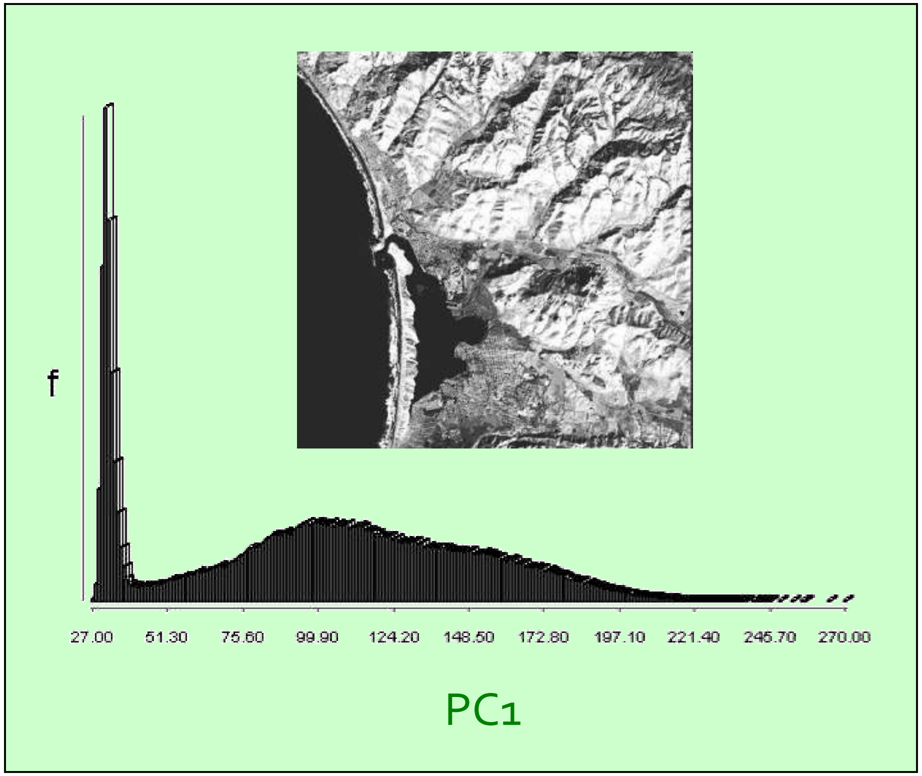
– The rotation is performed in order to have maximum variability in each new dimension

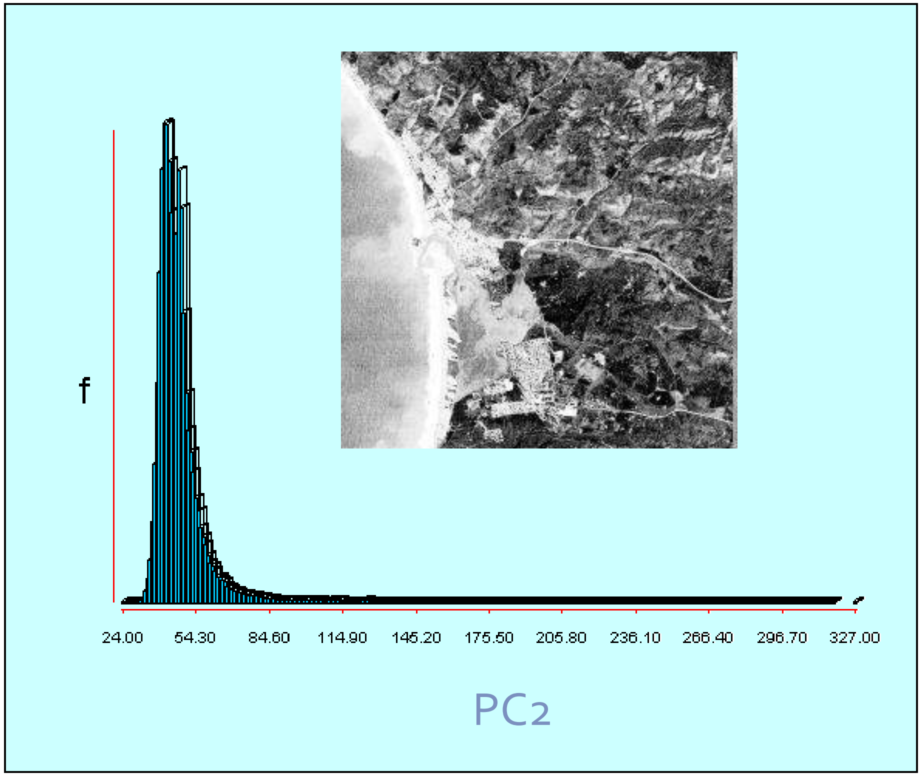
– No correlation between new variables

DLR

Histograms of First and Second Principal Components

Morro Bay Landsat Scene

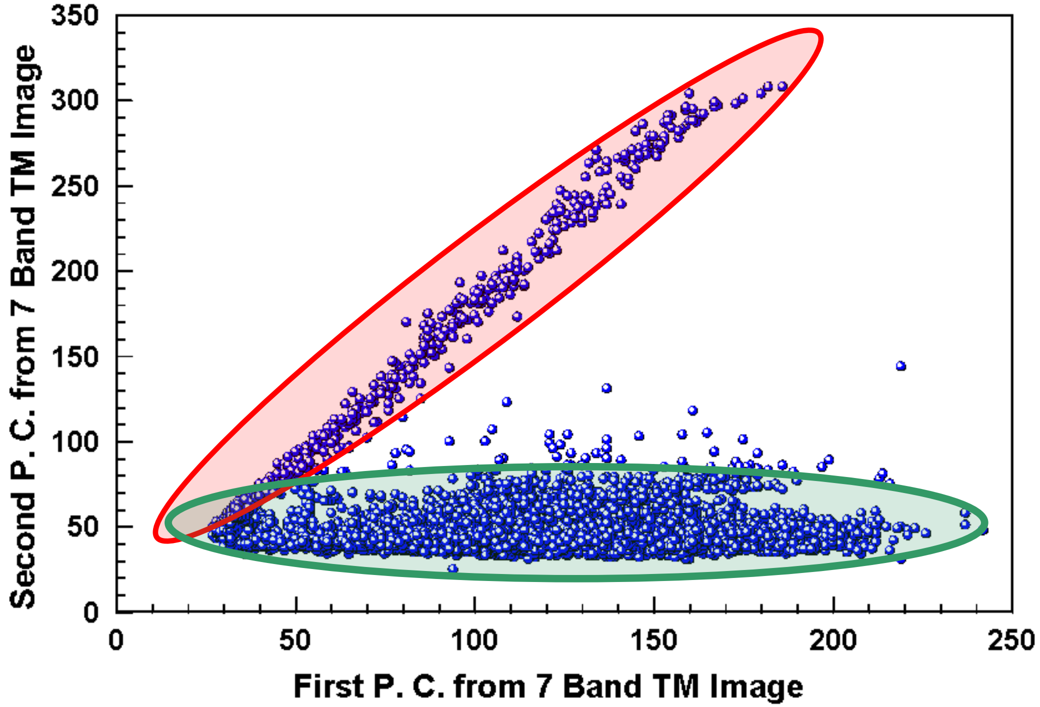


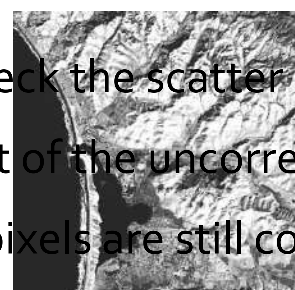


Check both the histograms and the images:

Which principal component contains more information, PC1 or PC2?

Plot of First vs. Second Principal Component





Check the scatter plot:

In which area are most of the uncorrelated data to be found?

Which pixels are still correlated?

Each component has its characteristics…

RGB Combination

PC 1

Close to what we would expect for a b/w picture of the scene

Max Information

PC 2

Several features can be spotted in the sea

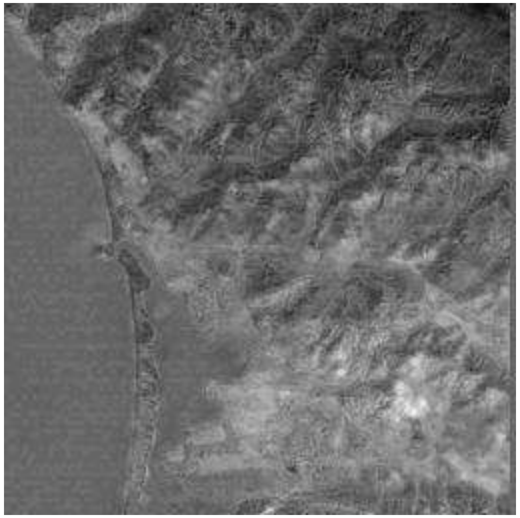
PC 3

Bright and dark gray for two classes of vegetation

DLR

Each component has its characteristics…

RGB
Combination



PC 4

Still some patterns in medium gray over the mountains

PC 6

This component appears noisy

Informational content ↓

Different PC = different information!

The main keyword for PCA is…

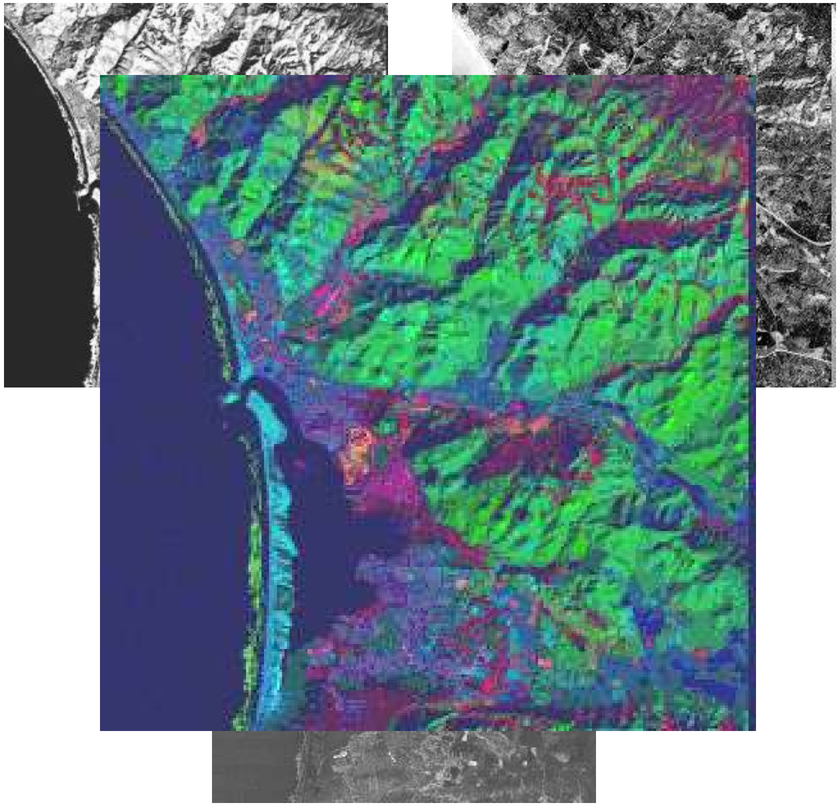
DECORRELATION!

DLR

Two Different Band Combinations

RGB Combination

(First three bands from the Landsat scene)



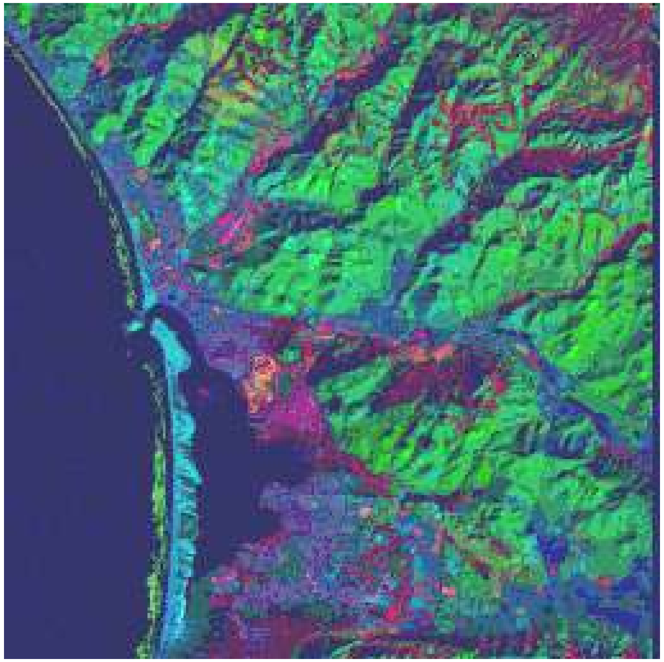
Combination of 3 PC

The information available in the Principal Components can be better revealed by combining them visually in a color composition

DLR

Two Different Band Combinations

RGB Combination

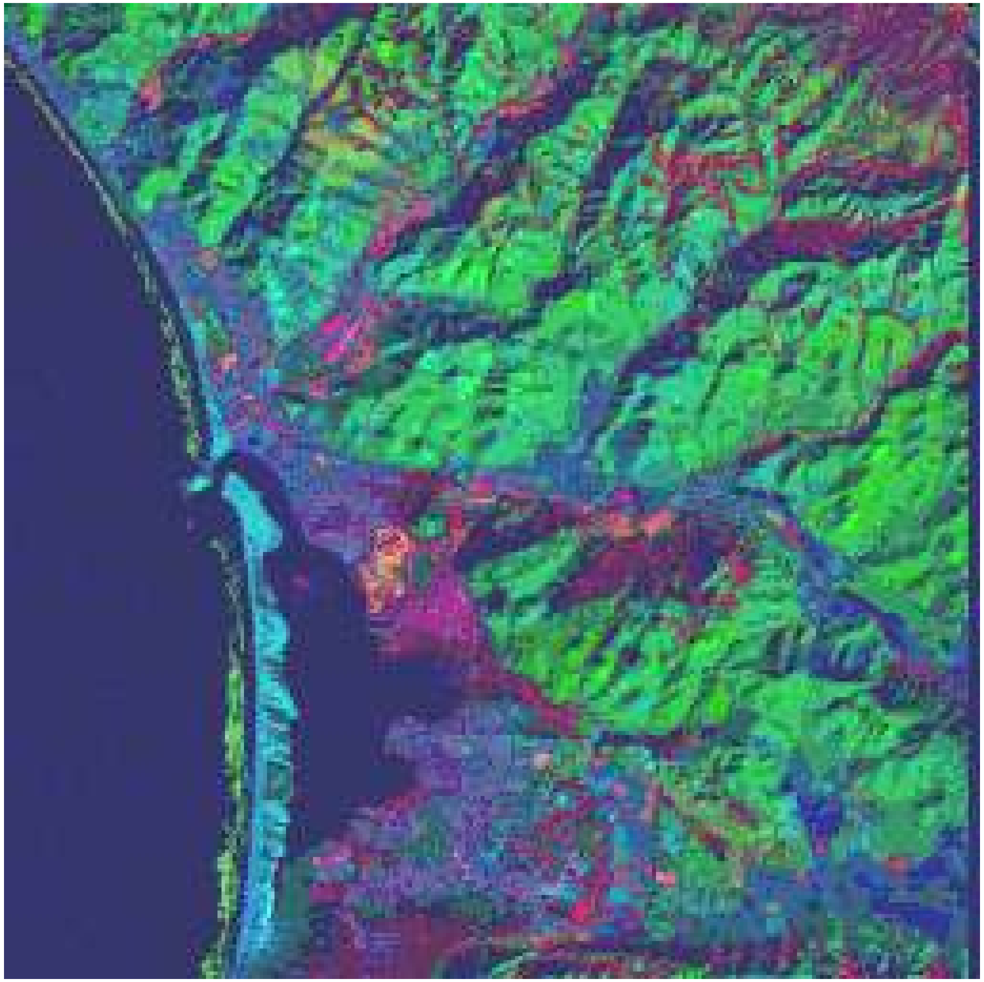


Three Principal Components

Which picture contains more information?

How many kinds of terrain can you spot in each one?

We can now identify many different areas…



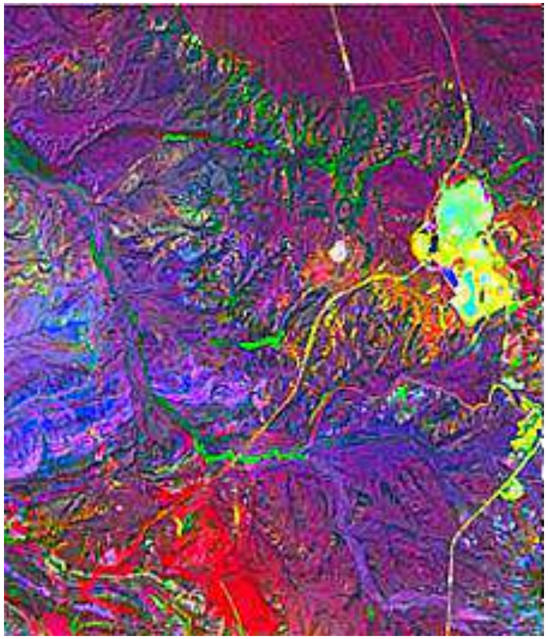
- Beach Bar
- Wave Breakers
- Vegetation1
- Vegetation2
- Golf Course
- Urban Area
- Shadows
- Sea
- Mountains (bright slopes)
- ….

DLR

A more dramatic example

A second Landsat scene

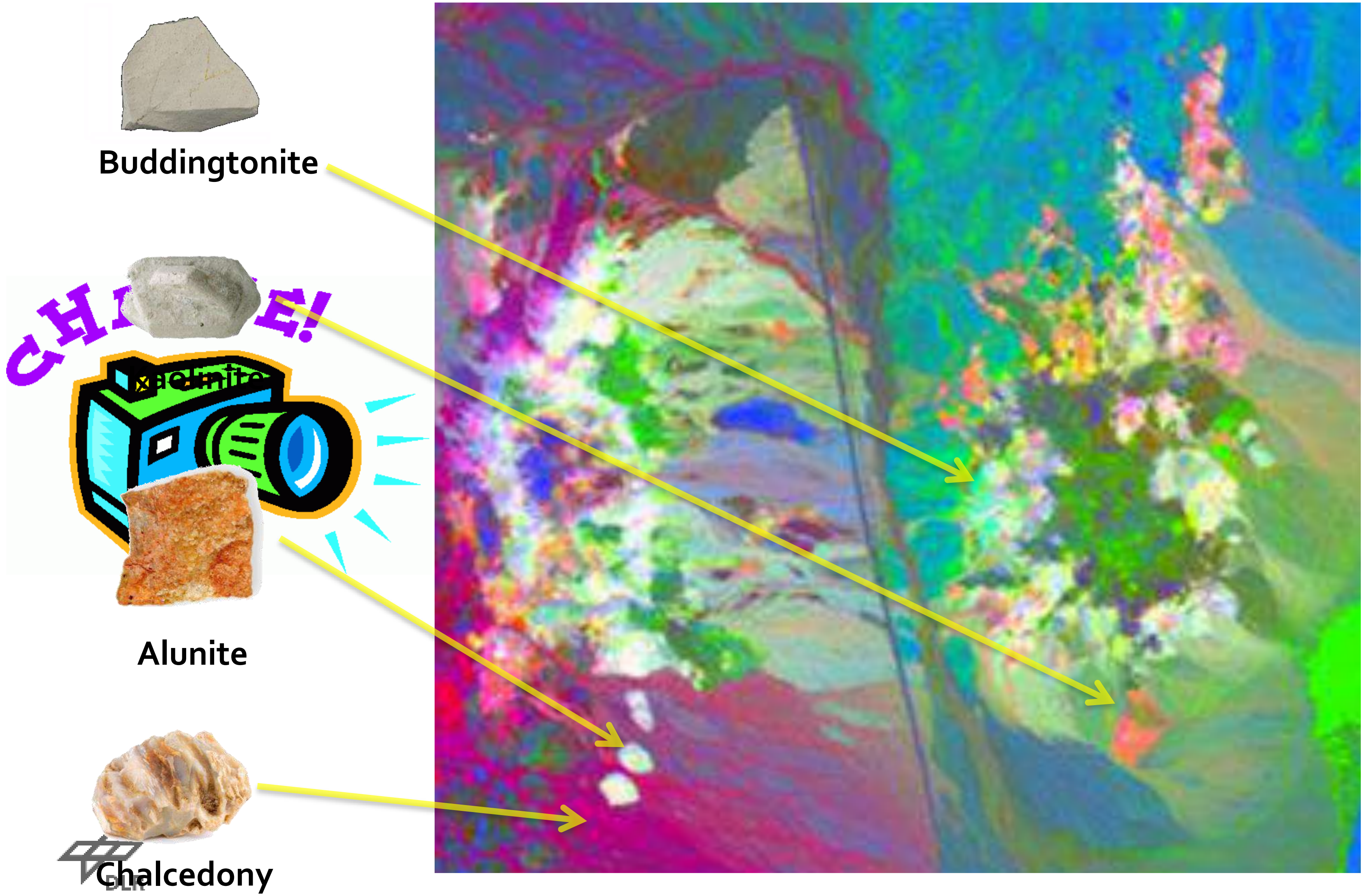
RGB Combination



Three Principal Components after Decorrelation Stretch (DS)

DS= Emphatization of the differences in color between the pixels

Hyperspectral



How do we get there?

- How do we express a principal component as a linear combination of the image bands?

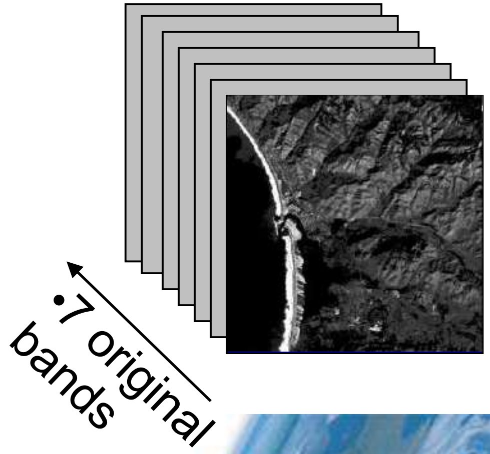
- A pixel p(i,j) at row i, column j is a vector of 7 bands b1…b7:

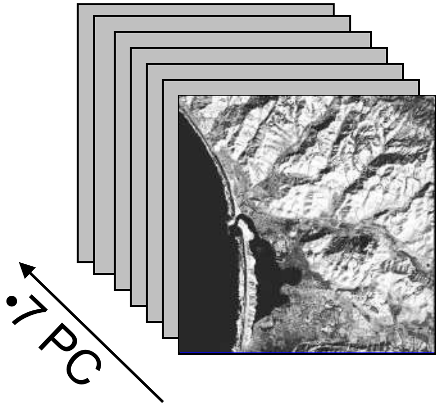
  p(i,j) = [b1(i,j),b2(i,j),b3(i,j),b4(i,j),b5(i,j),b6(i,j),b7(i,j)]

- Then a pixel of a PC can be expressed as:

  PC1(i,j) = [a(1,1)b1(i,j), a(1,2)b2(i,j), a(1,3)b3(i,j), a(1,4)b4(i,j), a(1,5)b5(i,j), a(1,6)b6(i,j), a(1,7)b7(i,j)]

  - How can we find these a(m,n) indices for each band and each PC?

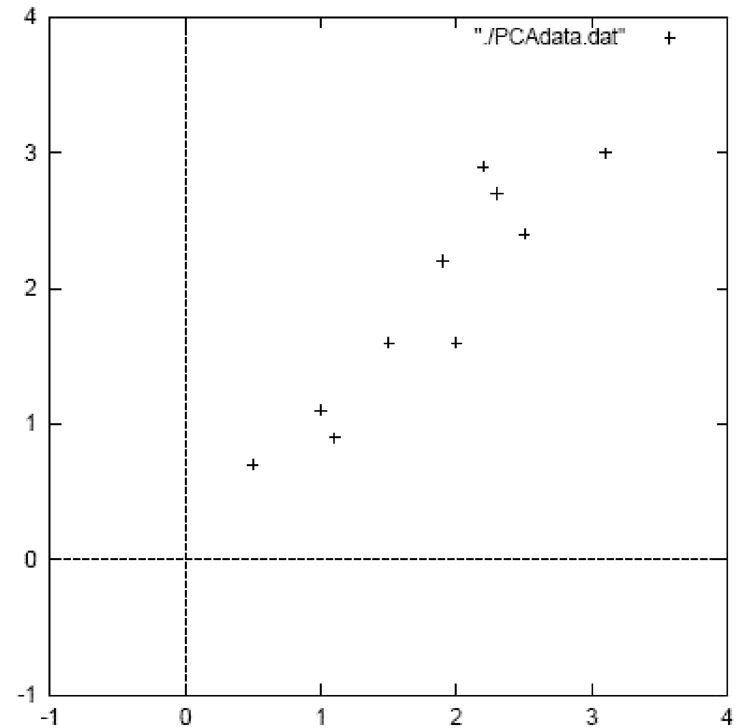




Let‘s see it through an Example…

Data

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

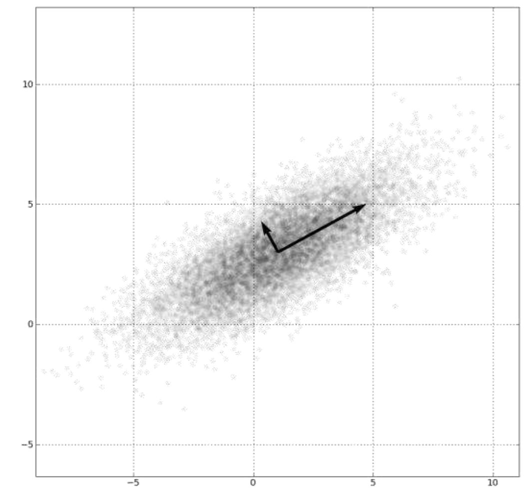


– Let‘s analyze this simple 2-dimensional dataset

  – Easy to visualize and to work with

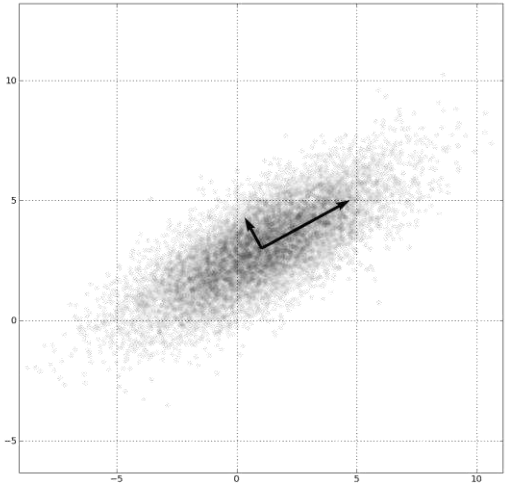
– The same procedure can be applied on the 7 dimensional Landsat scene, as well as on *n*-dimensional data (as long as *n* is finite)

PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

PCA Step by Step



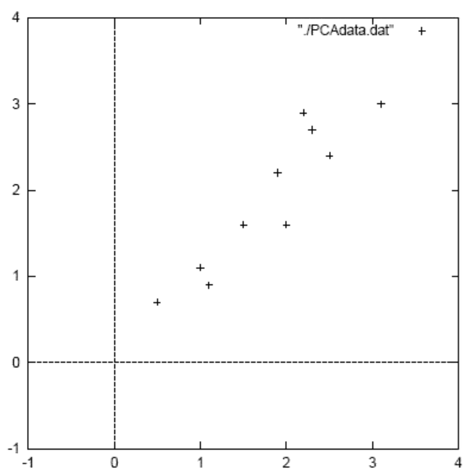
1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

DLR

Organize the Dataset

– Represent the data with a *m* x *n* matrix *M*

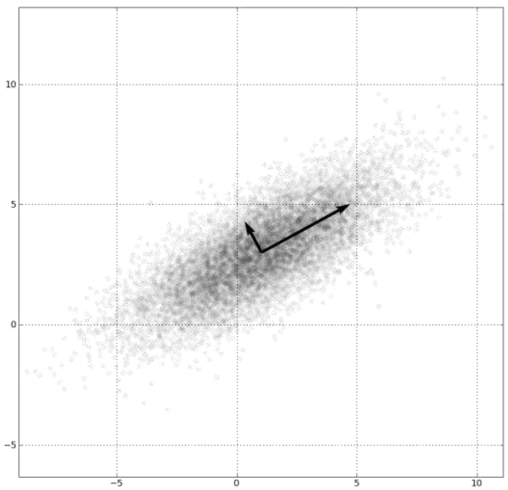
- *m* variables (in our case *x* and *y*)
- *n* observations per variable



$$M = \begin{pmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2 & 1.6 \\ 1 & 1.1 \\ 1.5 & 1.6 \\ 1.1 & 0.9 \end{pmatrix}$$

DLR

PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

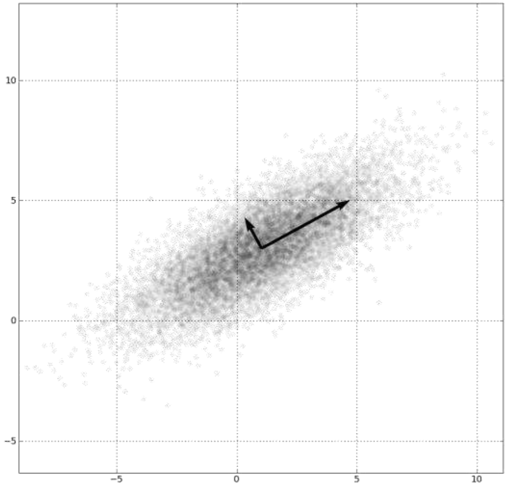
Subtract the mean

- Let $\underline{x}$ and $\underline{y}$ be the means of the *x* and *y* variables, respectively
  - For every *x* value: $x = x - \underline{x}$
  - For every *y* value: $y = y - \underline{y}$
- The mean of the data set is now zero
- Subtracting the mean makes next variance and covariance calculation easier by simplifying their equations
- The variance and co-variance values are not affected by the mean value

$$M = \begin{pmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2 & 1.6 \\ 1 & 1.1 \\ 1.5 & 1.6 \\ 1.1 & 0.9 \end{pmatrix} \Rightarrow \begin{pmatrix} 0.69 & 0.49 \\ -1.31 & -1.21 \\ 0.39 & 0.99 \\ 0.09 & 0.29 \\ 1.29 & 1.09 \\ 0.49 & 0.79 \\ 0.19 & -0.31 \\ -0.81 & -0.81 \\ -0.31 & -0.31 \\ -0.71 & -1.01 \end{pmatrix}$$

DLR

PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

What is the covariance?

- The covariance $Cov(x,y)$ between two variables x and y measures how much x and y change together

- There are two extreme cases:

  1. The variables are *independent*: knowing the value of x does not help in estimating the value of y → $Cov(x,y) \approx 0$
  2. The link between the variables is so strong that we can recover the values of y only by knowing the values of x → $Cov(x,y) = Max$

- Normally, this mutual dependance is somewhere in between

- High $Cov(x,y)$ → High correlation → When x is positive/negative, so is y
  - If the mean of x and y has been set to 0 as in the previous example

DLR

What is a covariance matrix?

– If $\underline{x}$ and $\underline{y}$ are the mean values of x and y we can think of the covariance as the average product of the deviations of x and y from the mean:

$$Cov(x, y) = average\left[(x - \underline{x})(y - \underline{y})\right]$$

- For the 2-dimensional case we can write in a matrix the covariances of any combination of the two variables

$$CovM(x, y) = \begin{pmatrix} Cov(x, x) & Cov(x, y) \\ Cov(y, x) & Cov(y, y) \end{pmatrix}$$

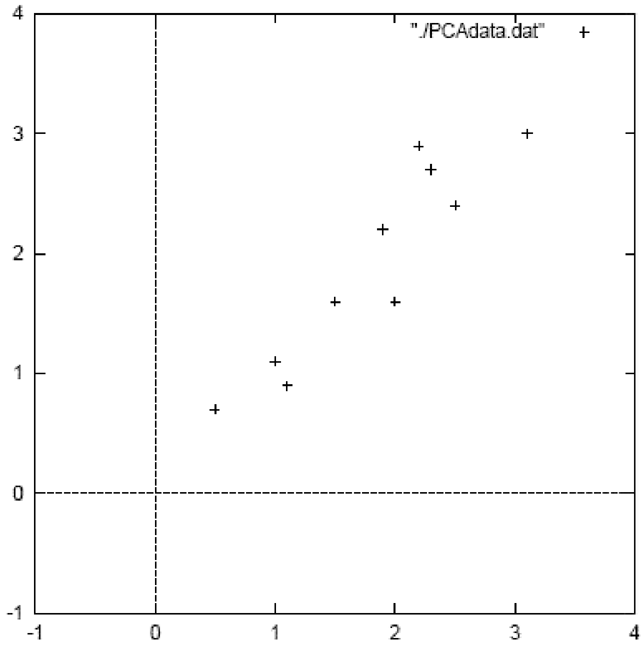
- Where $Cov(i,i)$ is the covariance of a variable with itself
    - Better known as variance σi^2 of i

DLR

Compute the covariance matrix

$$CovM(x,y) = \begin{pmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{pmatrix}$$

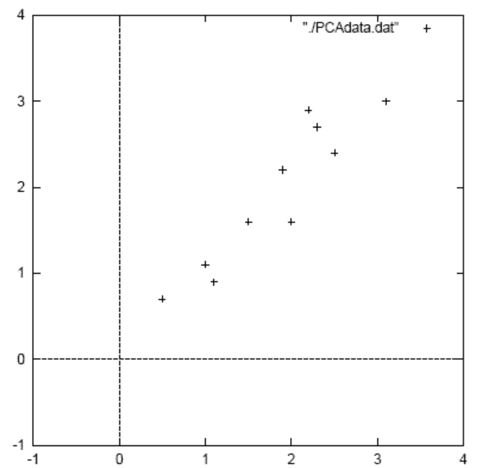
- Let's focus on the non-diagonal elements
  - Related to the mutual dependence of the variables
  - This information cannot be found in the values of the diagonal containing the variances
- In this case we are interested in *Cov*(*x*,*y*)
  - It is equal to *Cov*(*y*,*x*) since the covariance matrix is always symmetric
- What kind of value do you think *Cov*(*x*,*y*) will assume for the data distribution in the figure?



Compute the covariance matrix

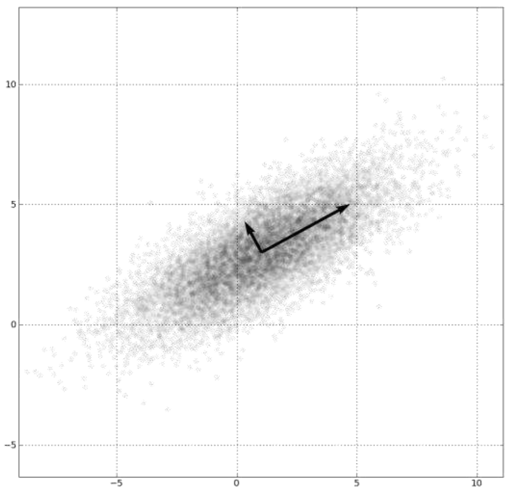
$$CovM(x,y) = \begin{pmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{pmatrix} = \begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

- *Cov*(*x*,*y*) is positive and comparable to the variances of *x* and *y*
- The two variables are strongly correlated!
- We expect them to vary together



DLR

PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

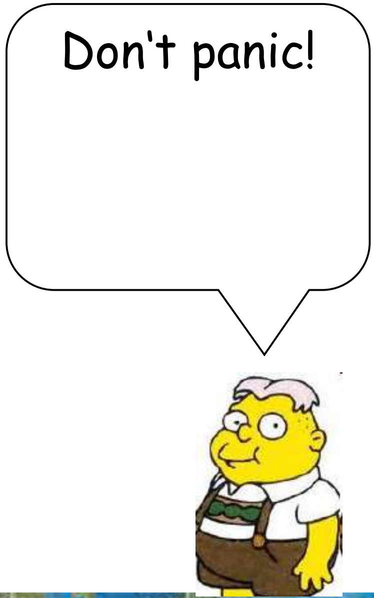
What are eigenvectors and eigenvalues?

– A vector *v* is an eigenvector for a matrix *M* if and only if

$$Mv = \lambda v$$

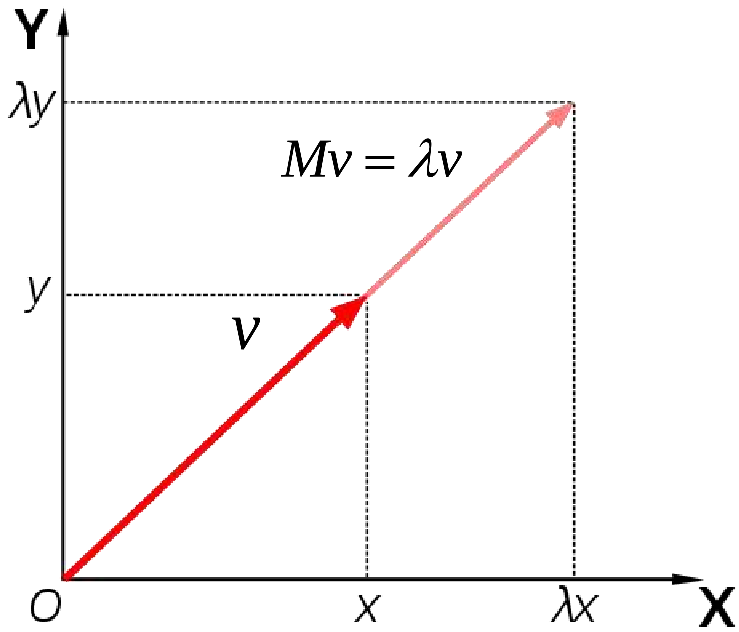
- Where λ is the eigenvalue related to the specific eigenvector *v* and is a scalar
- This means that *v* does not change if it is multiplied by *M*
  - The multiplication by the scalar λ „stretches“ the vector, but its direction is unaffected

- Eigenvectors are also known as characteristic vectors



DLR

What are eigenvectors and eigenvalues?



Example: here *v* is an eigenvector for the matrix *M*, as the result of the multiplication *Mv* does not change the direction of *v*.

Spot the eigenvector!

- Given the matrix $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

- And the two vectors $x = \begin{bmatrix} 3 \\ -3 \end{bmatrix}$ $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

x did not change after being multiplied by A!

Which one is an eigenvector?

- HINT!!

$$A\mathbf{x} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -3 \end{bmatrix} = \begin{bmatrix} 2 \cdot 3 + 1 \cdot (-3) \\ 1 \cdot 3 + 2 \cdot (-3) \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \end{bmatrix}$$

$$Ay = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 + 2 \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

And what is the eigenvalue of x?

It is 1!

The vector remained unchanged

Eigenvectors and eigenvalues

– Any *n* x *n* covariance matrix *A*, being symmetric, has *n* real eigenvectors

– It can be factorized as:

$$A = Q\Lambda Q^{-1}$$

- Q → matrix composed by the eigenvectors of A
- Λ → diagonal matrix containing the eigenvalues $\lambda 1 \ldots \lambda n$
- The eigenvectors can be chosen to be orthogonal
- They can form a new orthogonal basis → they can be thought of a new set of uncorrelated variables to represent the data!

DLR

Eigenvectors and eigenvalues

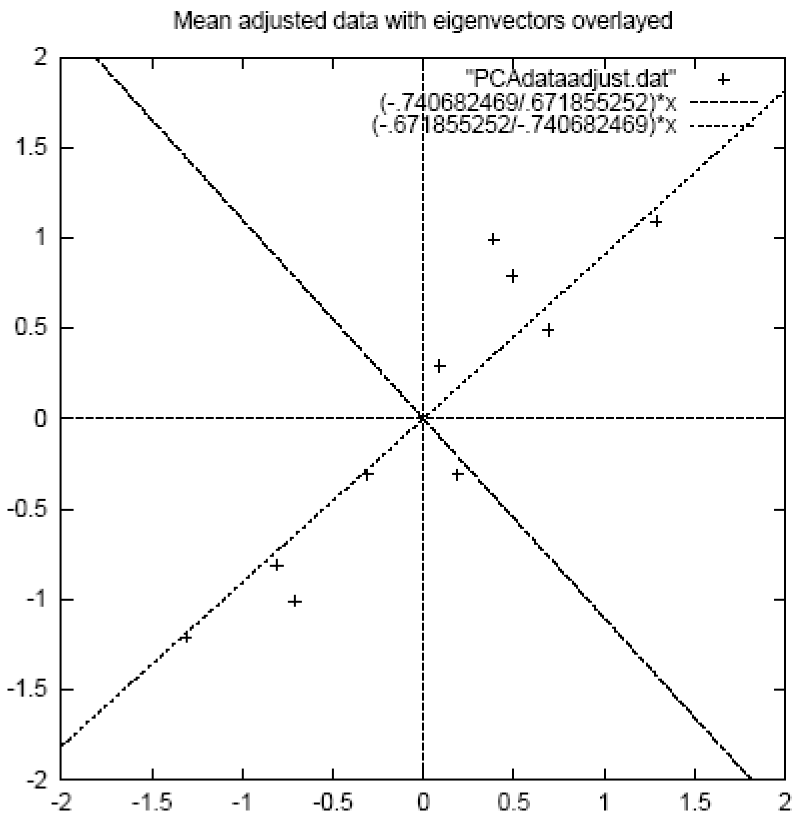
– Now we can compute the eigenvectors *Q* and eigenvalues Λ for our covariance matrix…

$$CovM(x,y) = \begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

$$Q(x,y) = \begin{pmatrix} -0.735 & -0.677 \\ 0.678 & -0.735 \end{pmatrix}$$

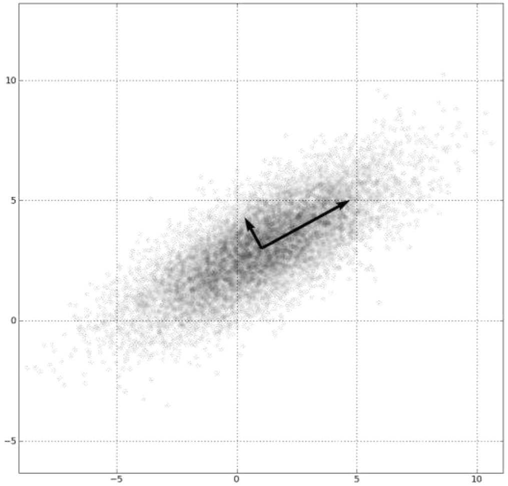
$$\Lambda(x,y) = \begin{pmatrix} 0.049 \\ 1.284 \end{pmatrix}$$

Let‘s project them back...



- Eigenvectors are plotted as diagonal dotted lines on the plot
- They are perpendicular to each other
- One of the eigenvectors goes through the middle of the points, like drawing a line of best fit
- The second eigenvector gives us the distance of the points from the first eigenvector
- It contains the second, less important aspect of the data

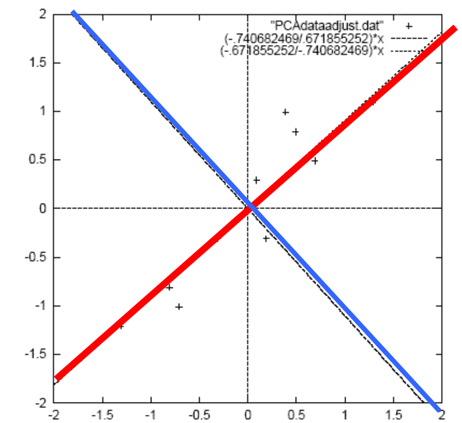
PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

Sort the eigenvectors

- The eigenvector with the highest eigenvalue is the principal component of the data set
  - It contains the highest amount of information on the data



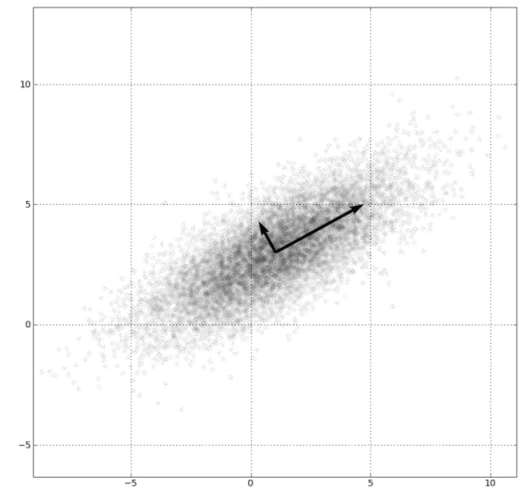
- In our example, it is "in the middle" of the data

- If we sort the eigenvectors from highest to lowest eigenvalue we have them in order of significance

$$\Lambda(x,y) = \begin{pmatrix} 0.049 \\ 1.284 \end{pmatrix} \begin{matrix} \cdot 2 \\ \cdot 1 \end{matrix} \quad \Longrightarrow \quad Q(x,y) = \begin{pmatrix} \overset{2}{-0.735} & \overset{1}{-0.677} \\ 0.678 & -0.735 \end{pmatrix}$$

DLR

PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

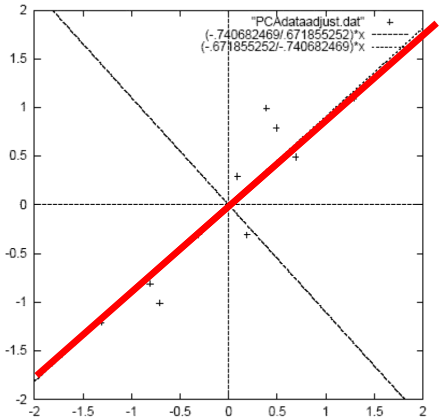
•Select a subset of the eigenvectors

– You can now decide to ignore less meaningful components

  – Eigenvectors with low eigenvalue

– Dimensionality reduction is achieved

  – Data compression is also achieved

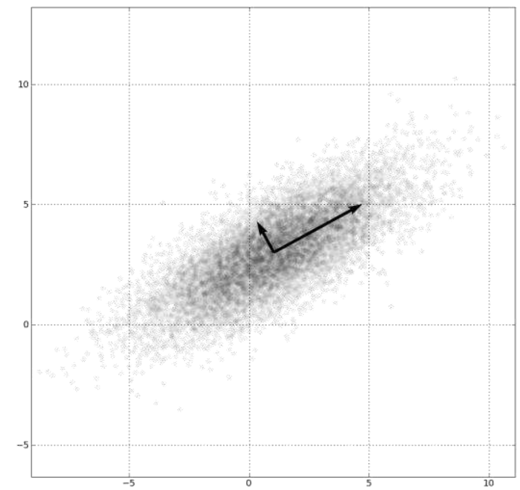


– Some information is lost, but as few as possible

$$\Lambda(x,y)=\begin{pmatrix}0.049\\1.284\end{pmatrix}\begin{matrix}\cdot 2\\ \cdot 1\end{matrix}\quad\Longrightarrow\quad Q(x,y)=\begin{pmatrix}\overset{2}{-0.735} & \overset{1}{\boxed{-0.677}}\\ 0.678 & \boxed{-0.735}\end{pmatrix}$$

•We can choose only the first component!

PCA Step by Step



1. Organize the data set
2. Subtract the mean
3. Compute covariance matrix
4. Find eigenvectors and eigenvalues for the covariance matrix
5. Sort the eigenvectors
6. Select a subset of the eigenvectors as basis vectors
7. Project the values unto the new basis

•Deriving the new data

– We can multiply our old data by our chosen set of eigenvectors

– We obtain a new representation for the data

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

•New representation of the data using both PCs

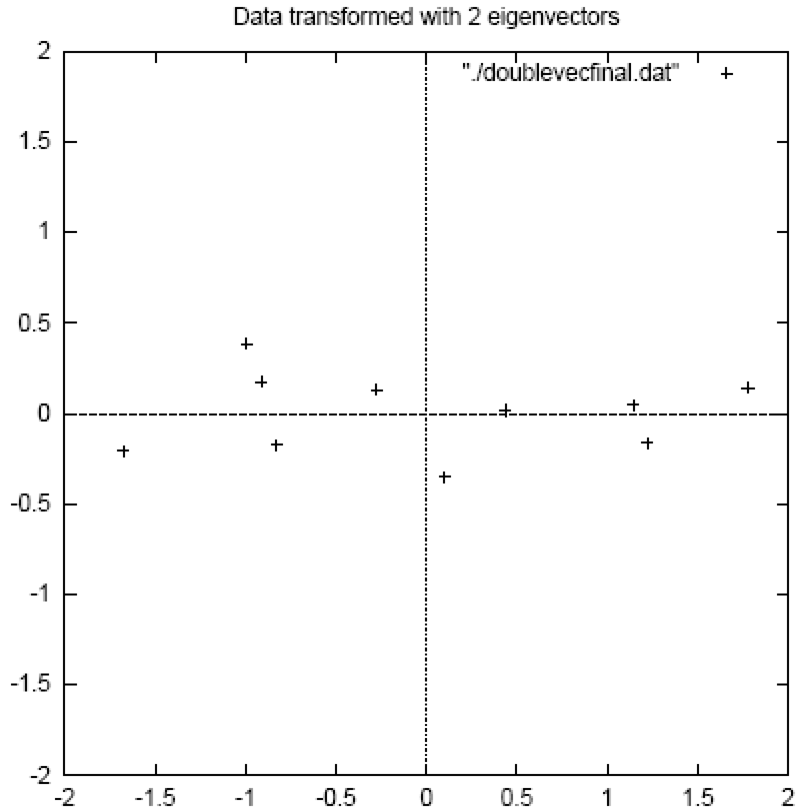
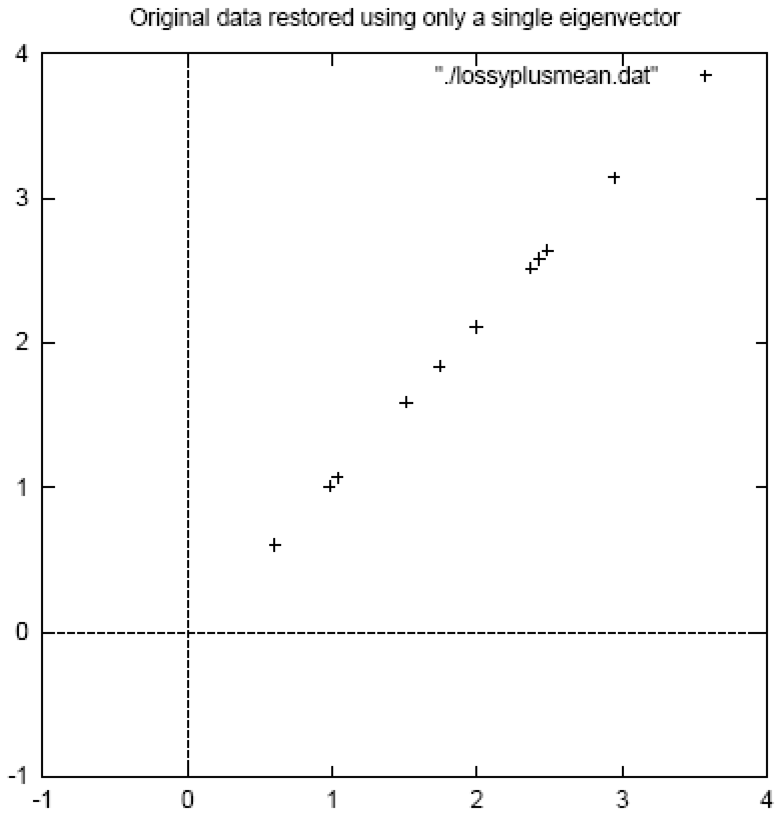


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

What if we use only the first PC?

x

-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056



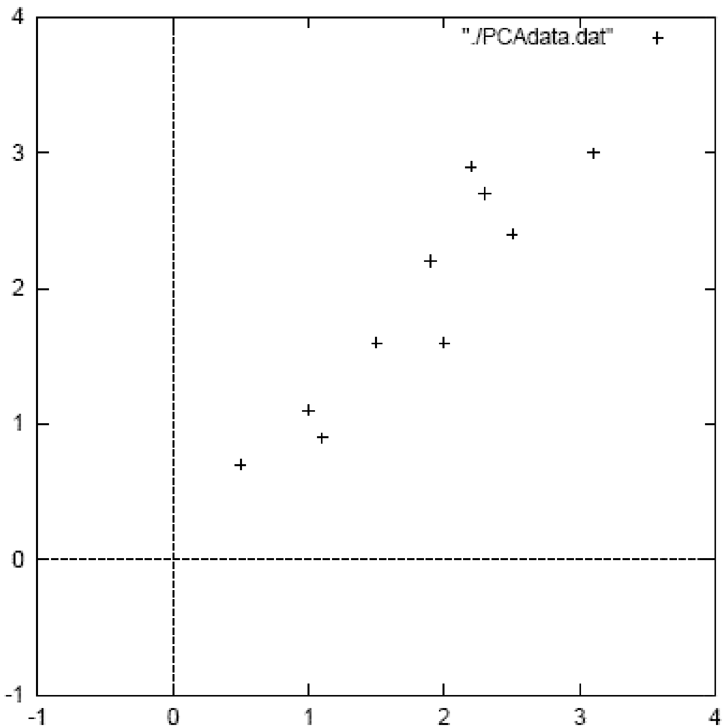
After adding back the mean values subtracted in the first steps

←

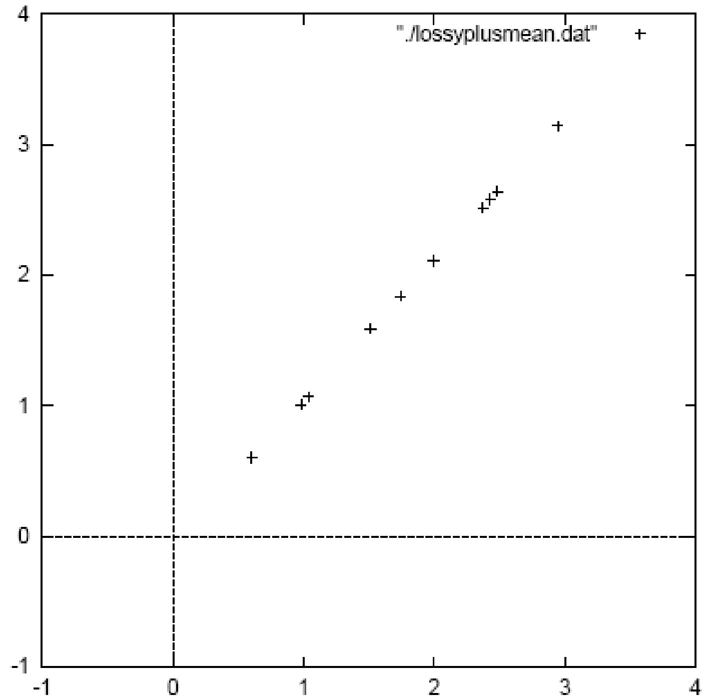
Figure 3.5: The reconstruction from the data that was derived using only a single eigen-vector

DLR

How much information are we keeping?

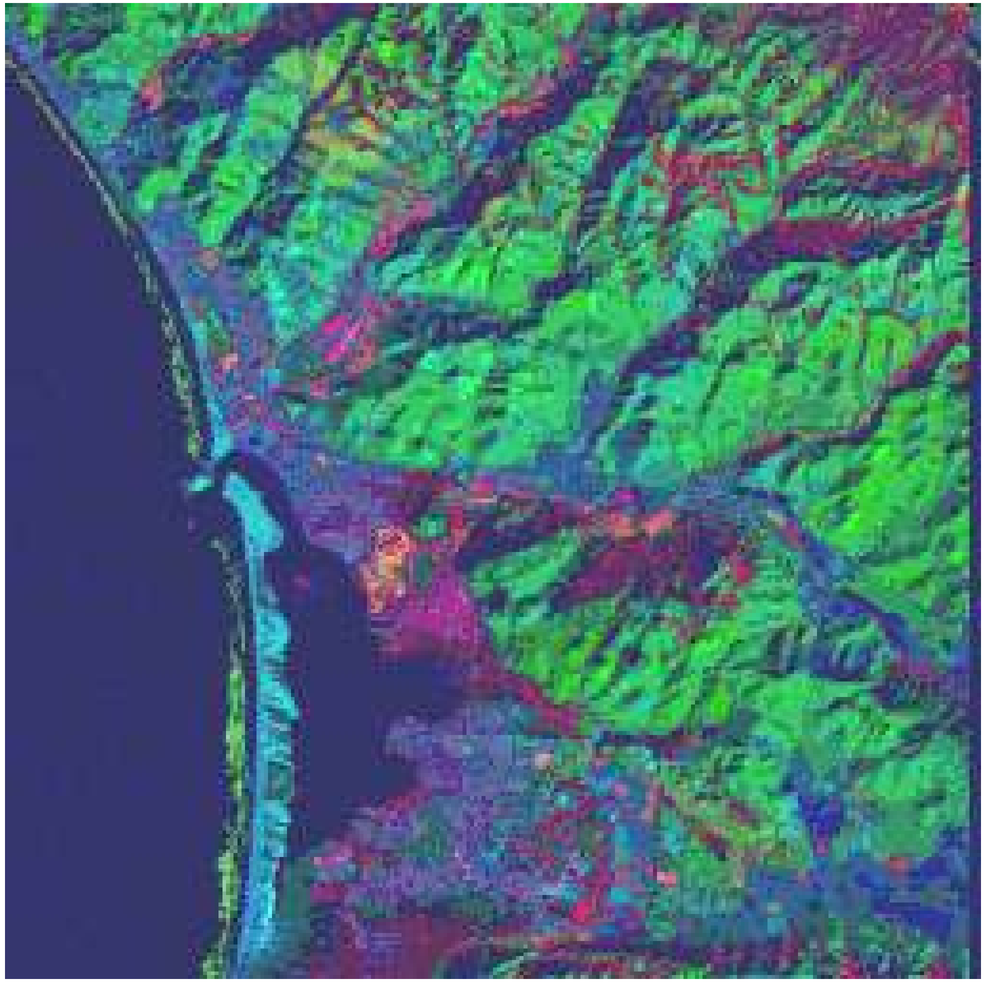


•Original 2D Data



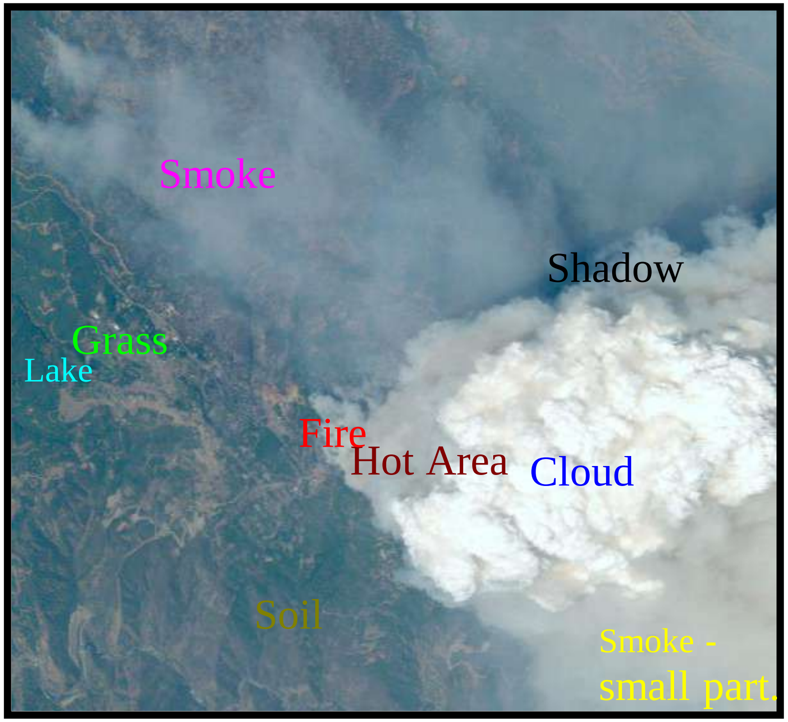
•Data reconstructed on the basis of only 1 Principal Component

•That‘s how we got here!



- Beach Bar
- Wave Breakers
- Vegetation1
- Vegetation2
- Golf Course
- Urban Area
- Shadows
- Sea
- Mountains (bright slopes)
- ....

One Last Example

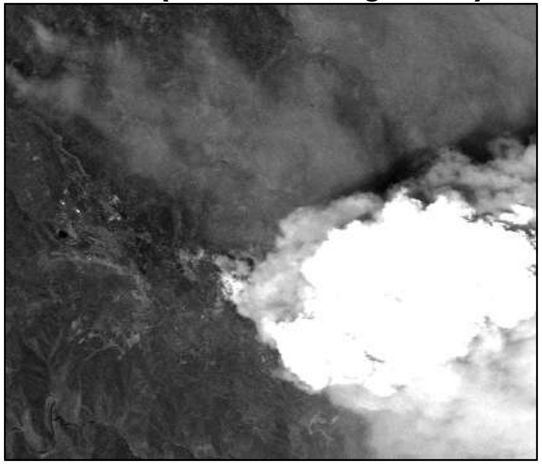


AVIRIS sensor RGB, Linden, CA , 20-Aug-1992

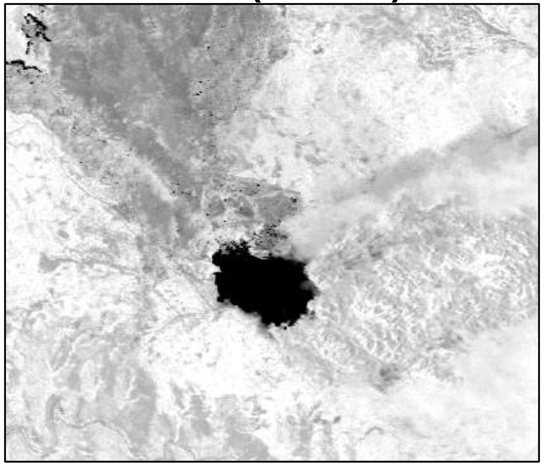
(Hsu, et al. in Frontiers of Remote Sensing Information Processing, WSP 2003)

Three Principal Components

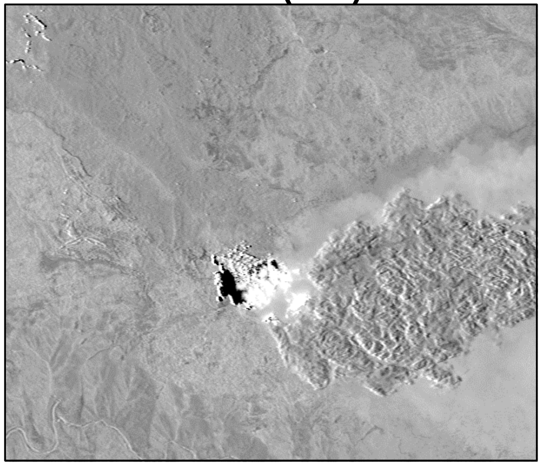
1st PC (Clouds/background)



2nd PC (Hot area)



5th PC (Fire)

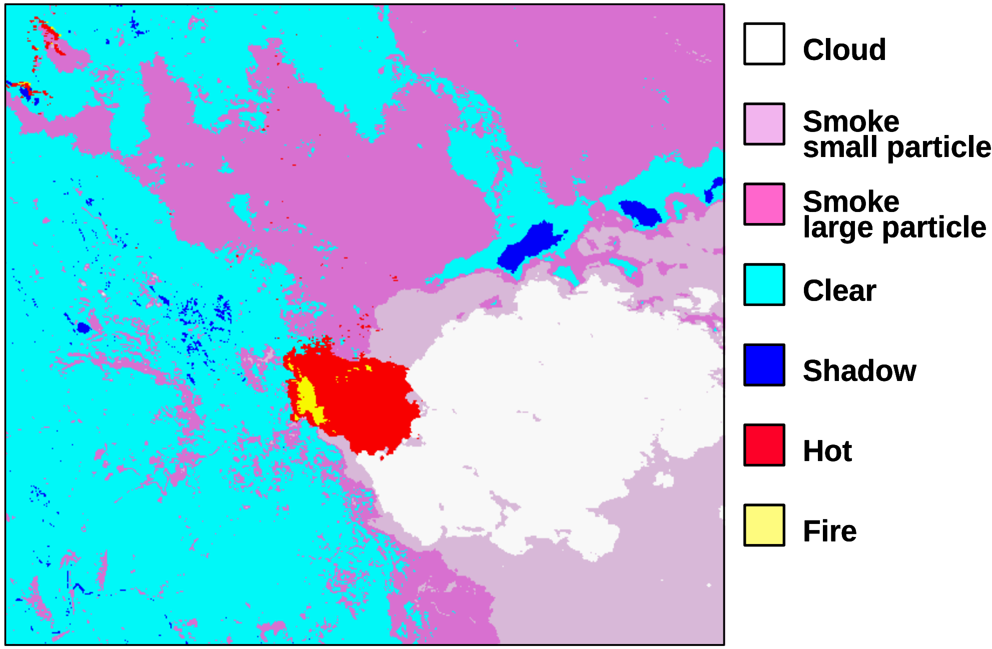


The 1st component again resembles a b/w picture of the area

The 2nd highlights an area in which we have a thermal anomaly

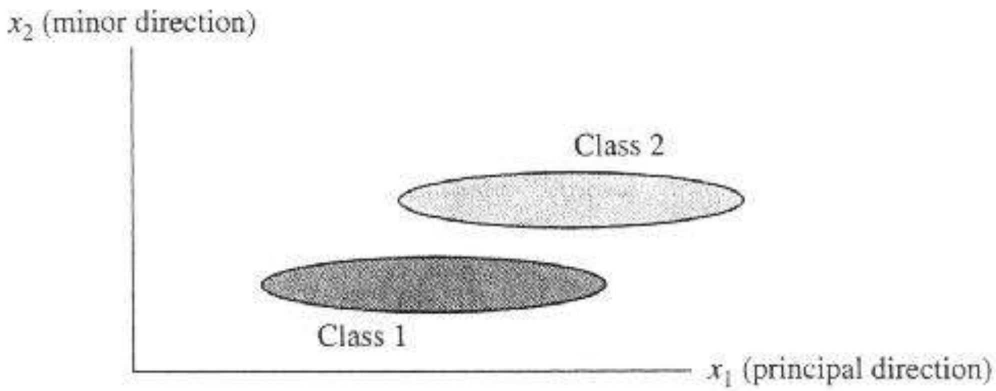
The 5th shows the cause of the anomaly (fire), which was hidden in the true color composition

Classification using the 3 PCs



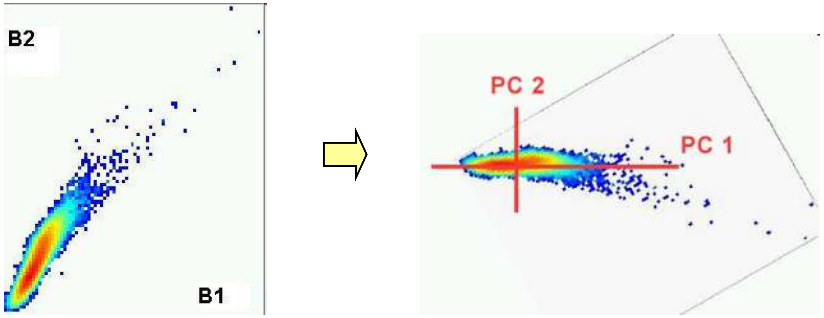
All major atmospheric and surface features can be identified

PCA is NOT Always Optimal!



– What happens if *x1* and *x2* are our first two PCs in this example?

Questions



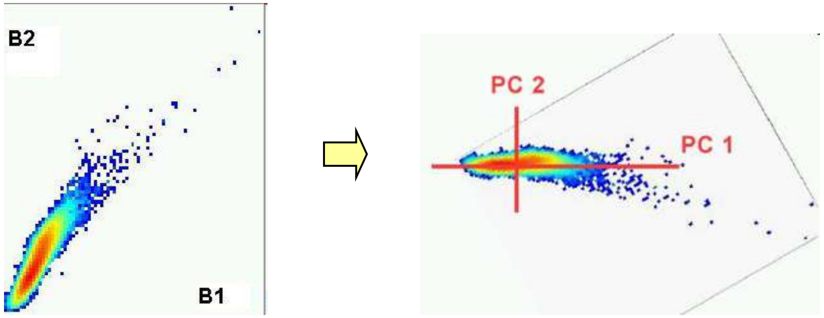
– What is the relation between the eigenvectors of the covariance matrix and the principal components?

– At what point in the PCA process can we decide to compress the data?

– Why are the principal components orthogonal?

– How many different covariance values can you calculate for an n-dimensional data set?

Conclusions



– PCA can be viewed as a projection of the observations onto orthogonal axes contained in the space defined by the original variables

– The first new variable (PC1) contains the maximum amount of variation → max information

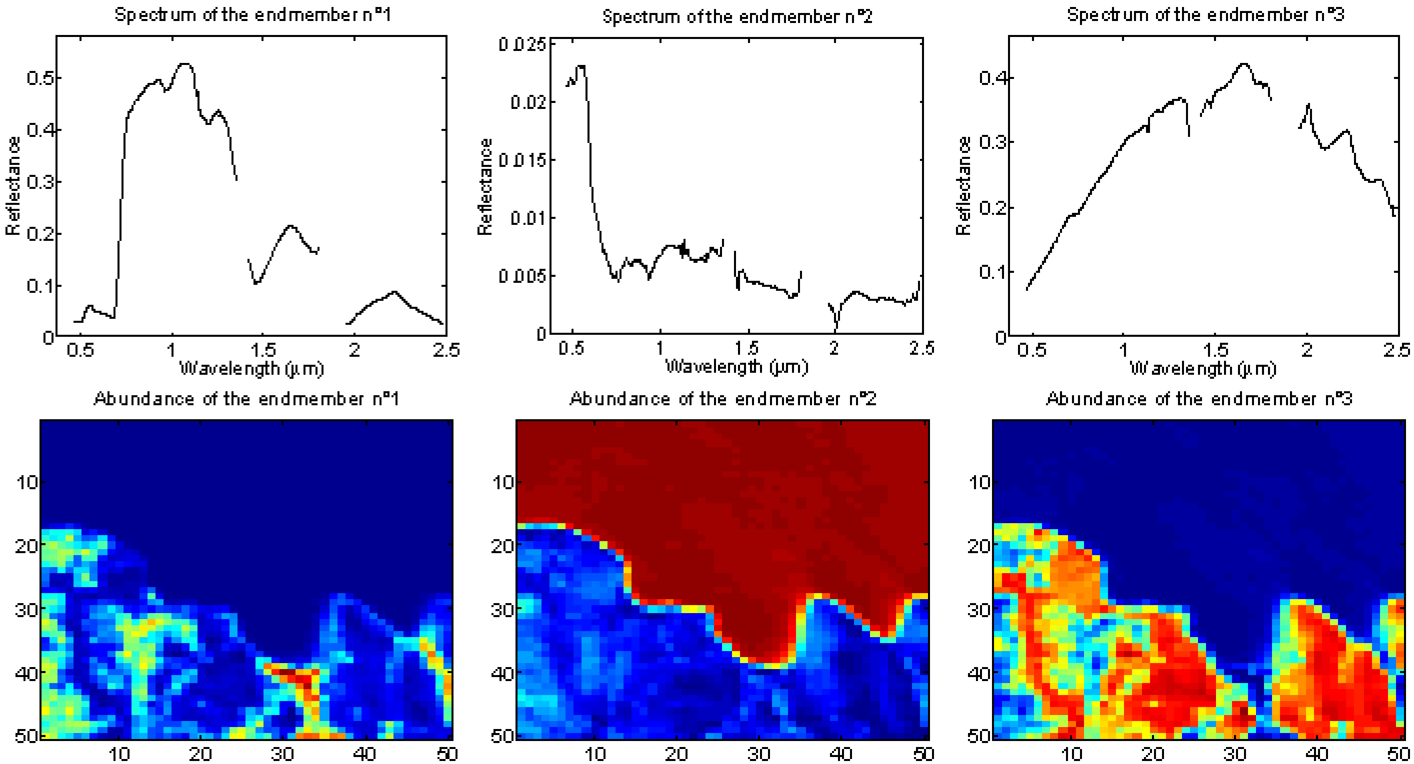
– The remaining components PC2..PCn are sorted according to their informational content, i.e. to their variance (which is not equal to the variance of the variables!!)

– The rotation is a linear combination of the original bands

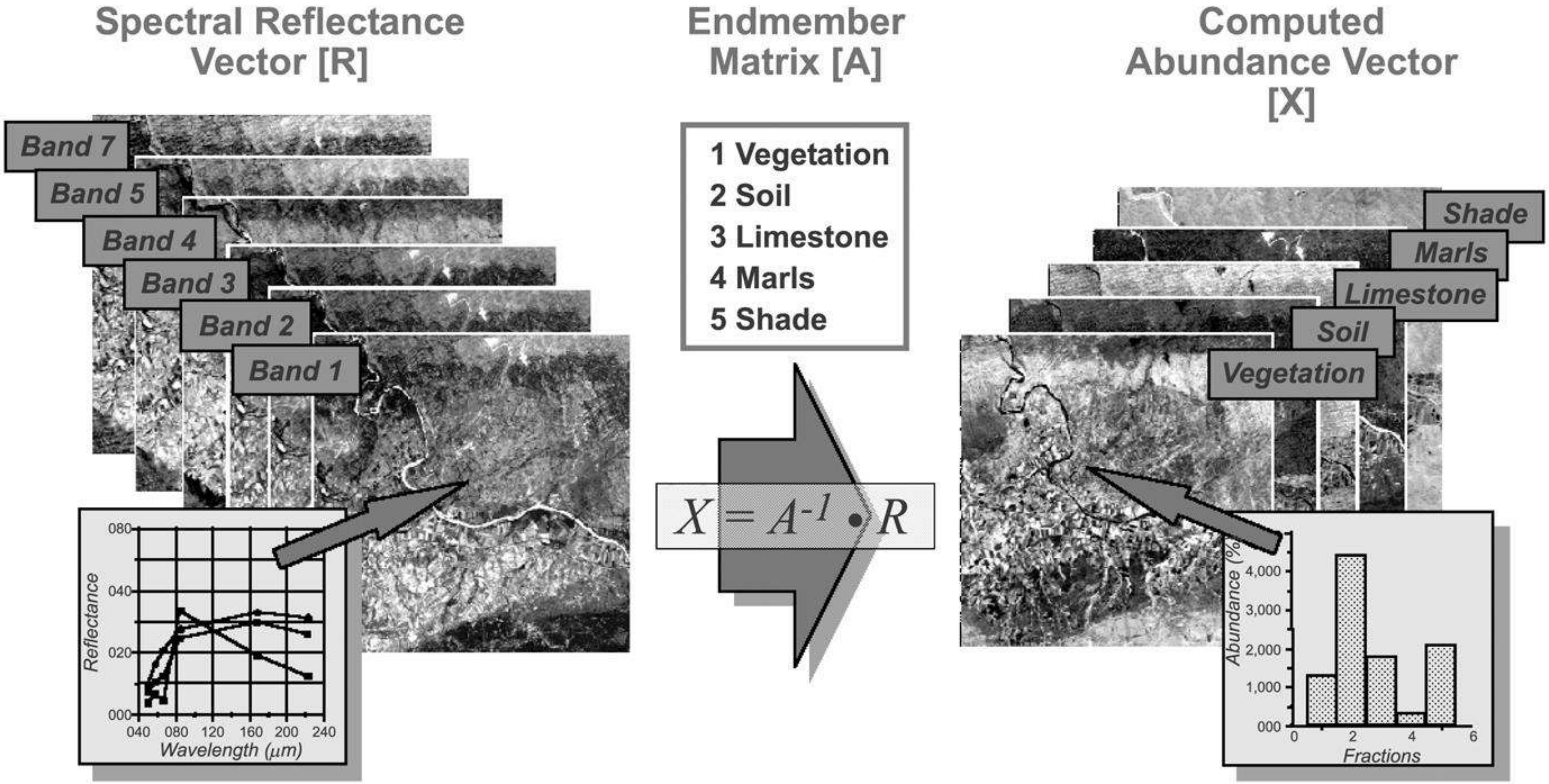
→ No information loss, original data can be recovered

→ The last components can be ignored, achieving data reduction

Spectral Unmixing



Spectral Unmixing



5th Workshop on
Hyperspectral Image and Signal Processing : Evolution in Remote Sensing

25 - 28 June 2013, Gainesville, Florida, USA

whispers 2013
Gainesville, Florida, USA

IEEE GRSS HYPERCOMP Hyperspectral Computing Laboratory

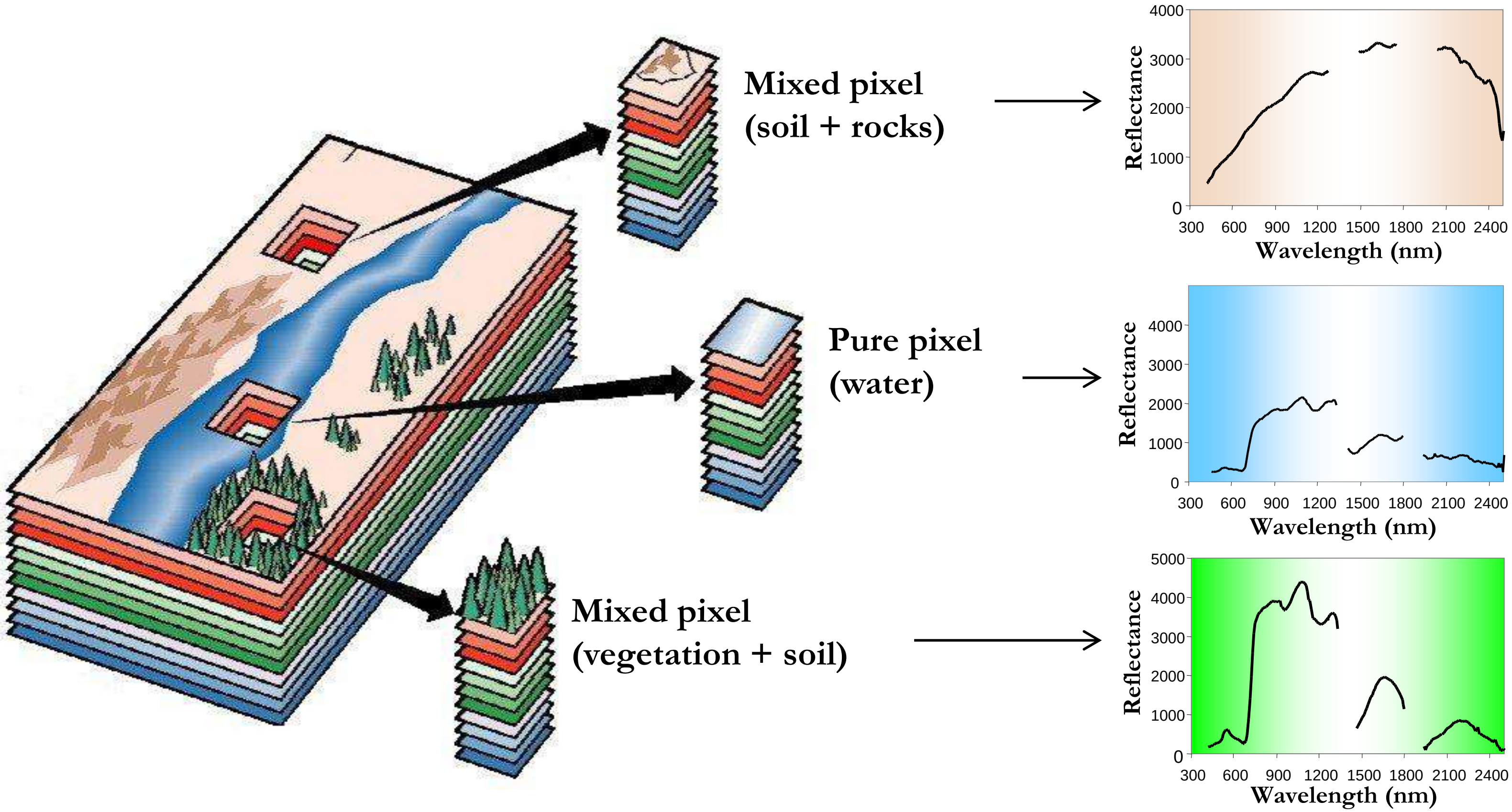
Spectral Unmixing of Hyperspectral Data

Antonio J. Plaza
Hyperspectral Computing Laboratory (HyperComp)
Department of Technology of Computers and Communications
Escuela Politécnica de Cáceres, University of Extremadura, Cáceres, Spain
E-mail: aplaza@unex.es – URL: http://www.umbc.edu/rssipl/people/aplaza

Contents

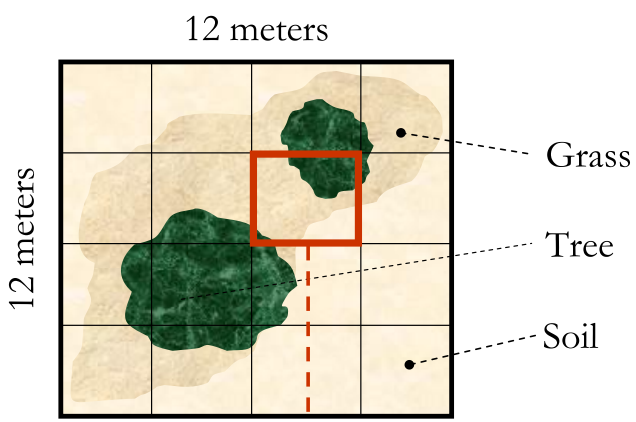
Introduction to spectral unmixing

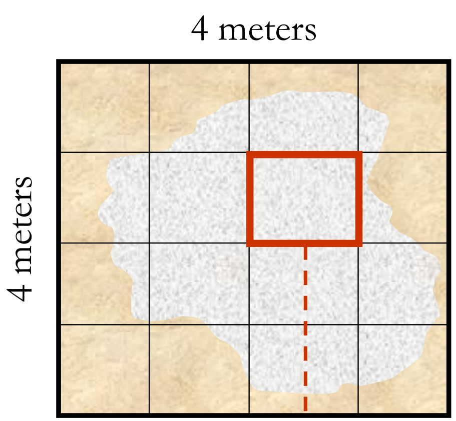
- Mixed pixels are frequent in remotely sensed hyperspectral images due to insufficient *spatial resolution* of the imaging spectrometer, or due to *intimate mixing effects*.
- The rich spectral resolution available can be used to unmix hyperspectral pixels.



Introduction to spectral unmixing

- Mixed pixels can also be obtained with high spatial resolution data due to intimate mixtures, this means that increasing the spatial resolution does not solve the problem.
- The mixture problem can be approached in *macroscopic* fashion, this means that a few macroscopic components and their associated abundances should be derived.
- However, intimate mixtures happen at microscopic scales, thus complicating the analysis with nonlinear mixing effects.





Macroscopic mixture:

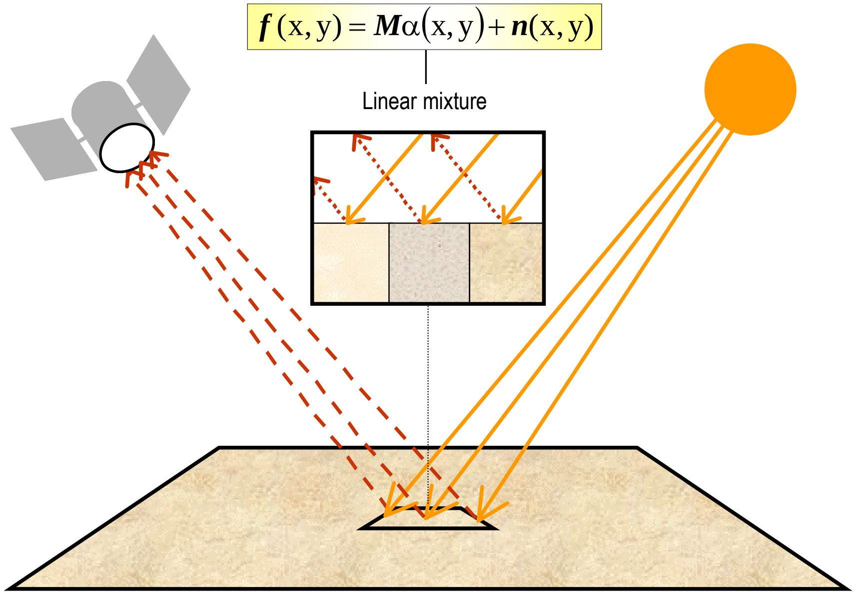
15% soil, **25%** tree, **60%** grass in a 3x3 meter-pixel

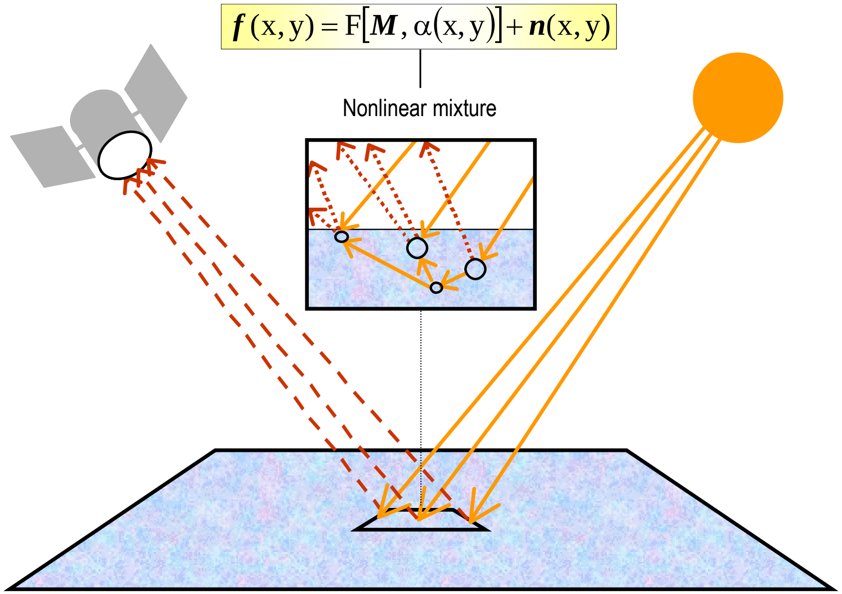
Intimate mixture:

Minerals intimately mixed in a 1-meter pixel

Introduction to spectral unmixing

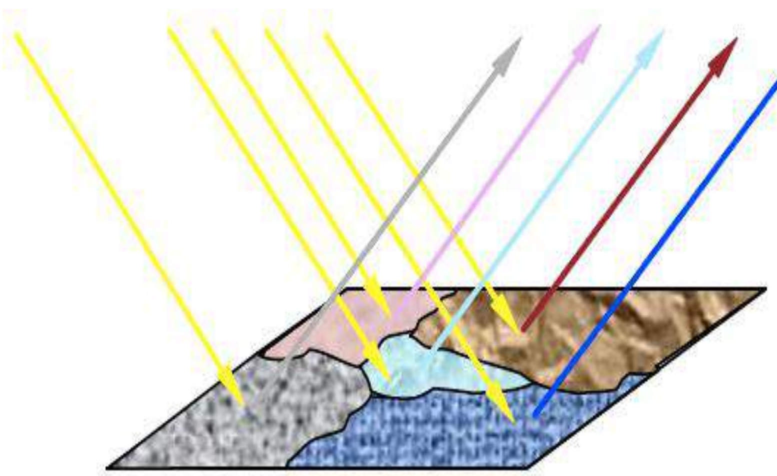
- In *linear* spectral unmixing, the macroscopically pure components are assumed to be homogeneously distributed in separate patches within the field of view.
- In *nonlinear* spectral unmixing, the microscopically pure components are intimately mixed inside the pixel. A challenge is how to derive the nonlinear function.
- Nonlinear spectral unmixing requires detailed *a priori* knowledge about the materials.



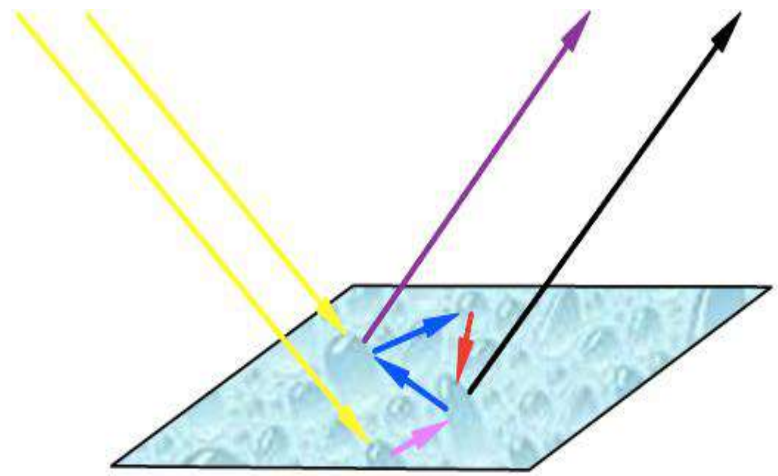


Introduction to spectral unmixing

- In *linear* spectral unmixing, the macroscopically pure components are assumed to be homogeneously distributed in separate patches within the field of view.
- In *nonlinear* spectral unmixing, the microscopically pure components are intimately mixed inside the pixel. A challenge is how to derive the nonlinear function.
- Nonlinear spectral unmixing requires detailed *a priori* knowledge about the materials.



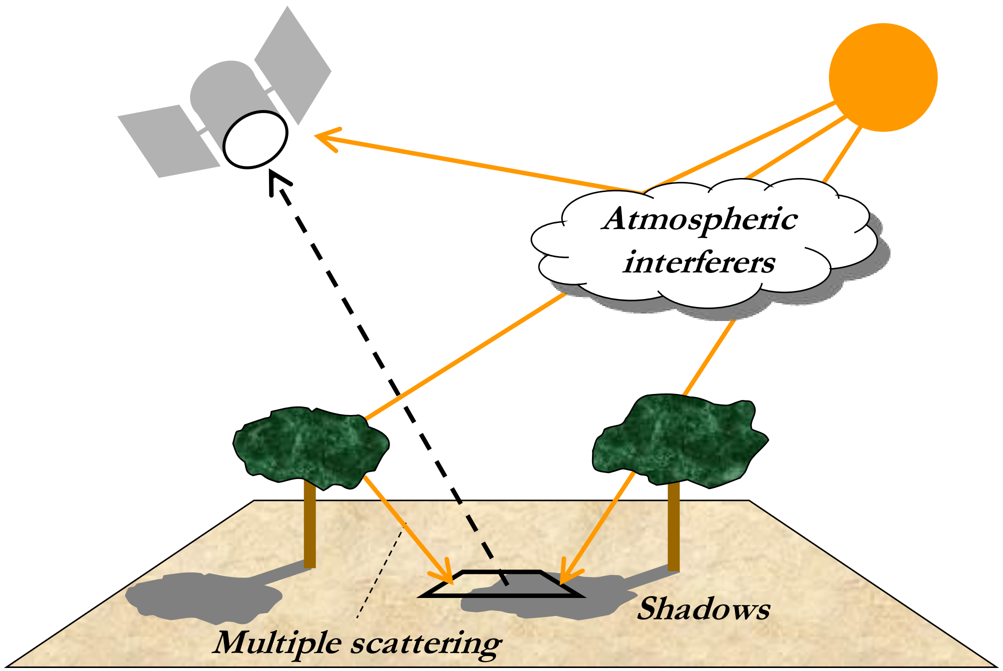
Linear interaction



Nonlinear interaction

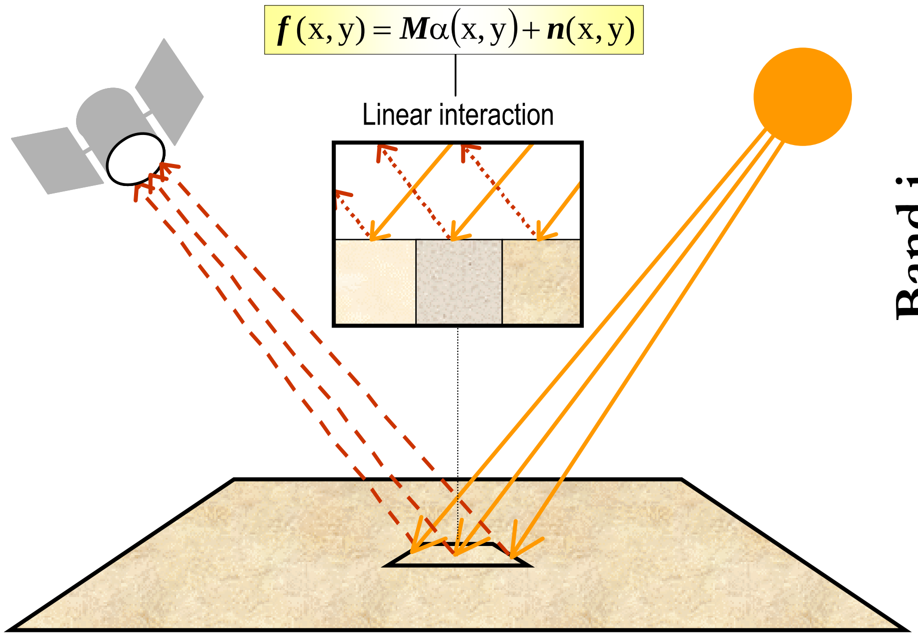
Introduction to spectral unmixing

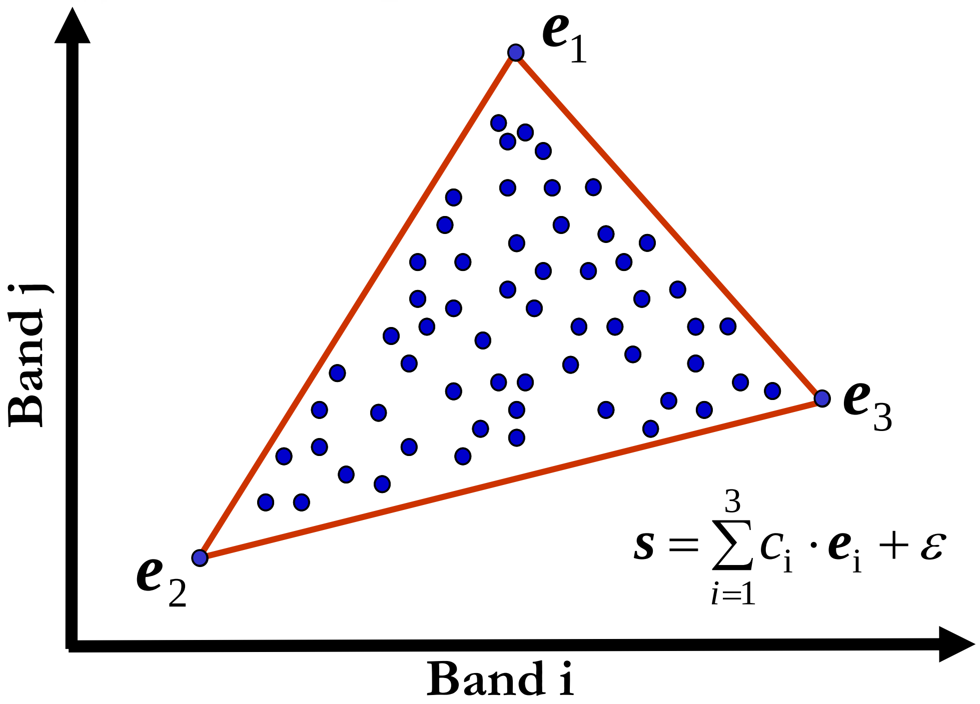
- In addition to spectral mixing effects, there are many other *interferers* that can significantly affect the process of analyzing the remotely sensed hyperspectral data.
- For instance, *atmospheric interferers* are a potential source of errors in spectral unmixing.
- On the other hand, *multiple scattering* effects can also lead to model inaccuracies.
- Finally, shadows and variable *illumination* conditions should also be considered.



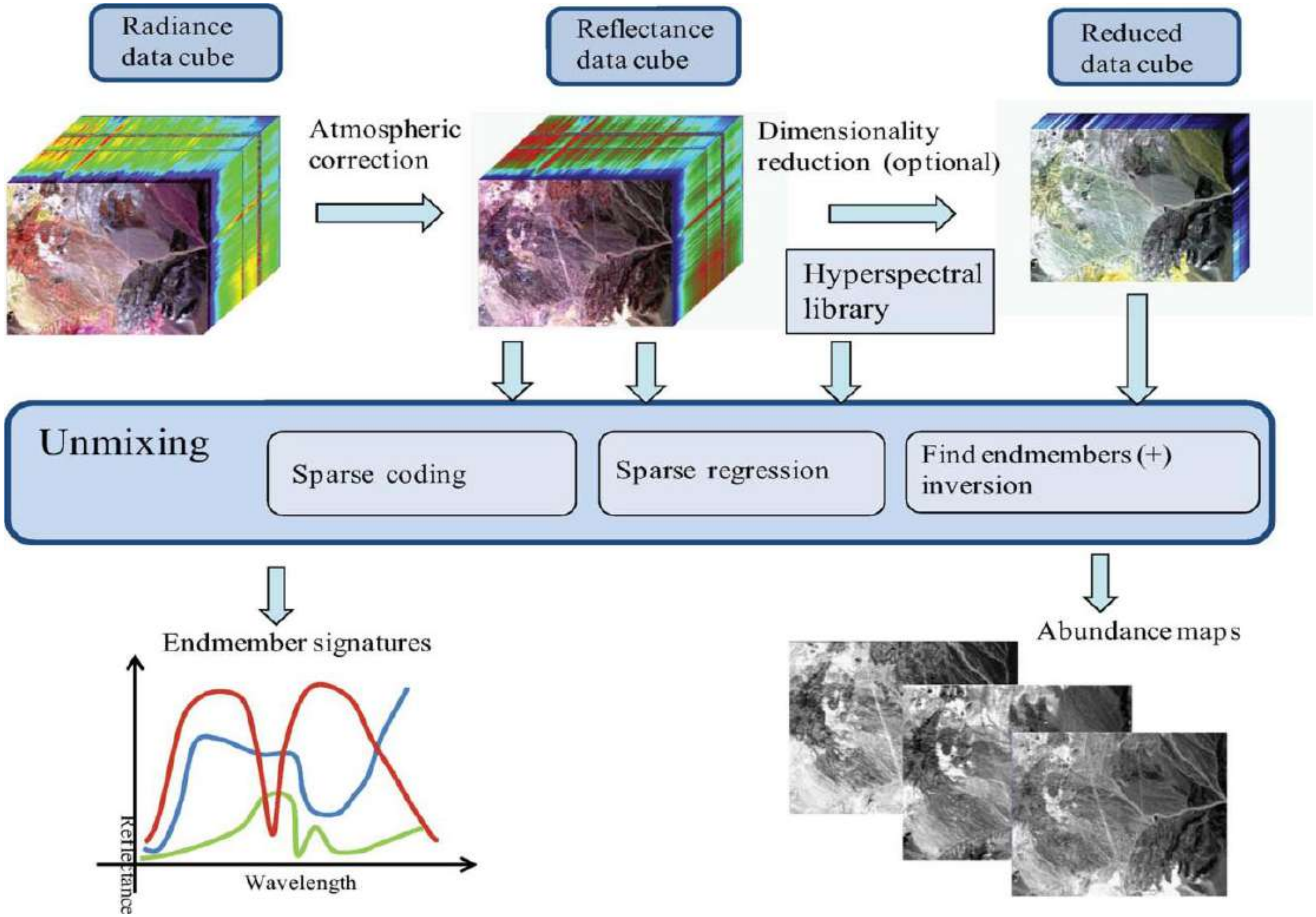
Introduction to spectral unmixing

- In *linear* spectral unmixing, the goal is to find a set of macroscopically pure spectral components (called *endmembers*) that can be used to unmix all other pixels in the data.
- Unmixing amounts at finding the fractional coverage (*abundance*) of each endmember in each pixel of the scene, which can be approached as a *geometrical* problem:





Introduction to spectral unmixing

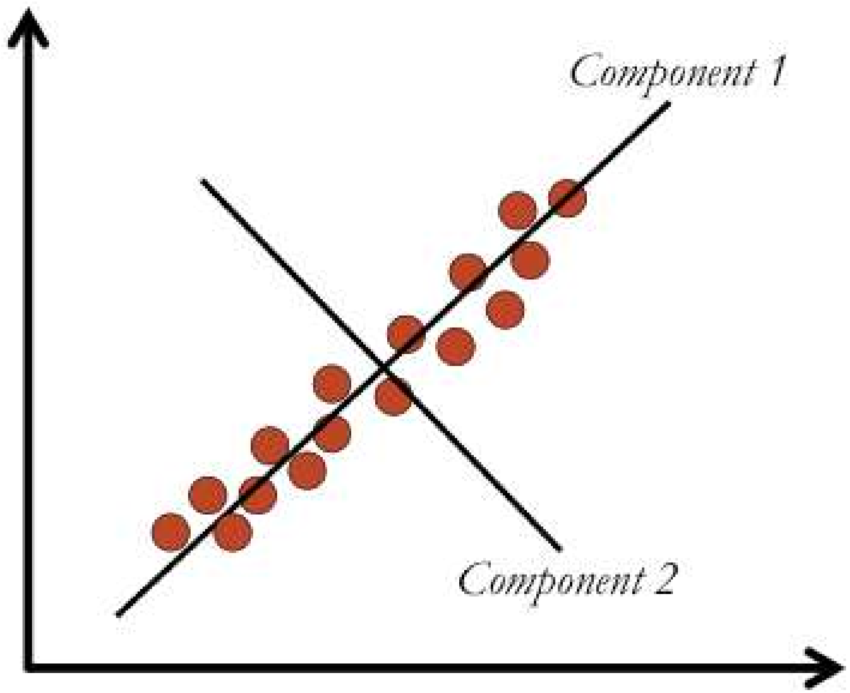


J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader and J. Chanussot, "Hyperspectral unmixing overview: geometrical, statistical and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354-379, April 2012.

Contents

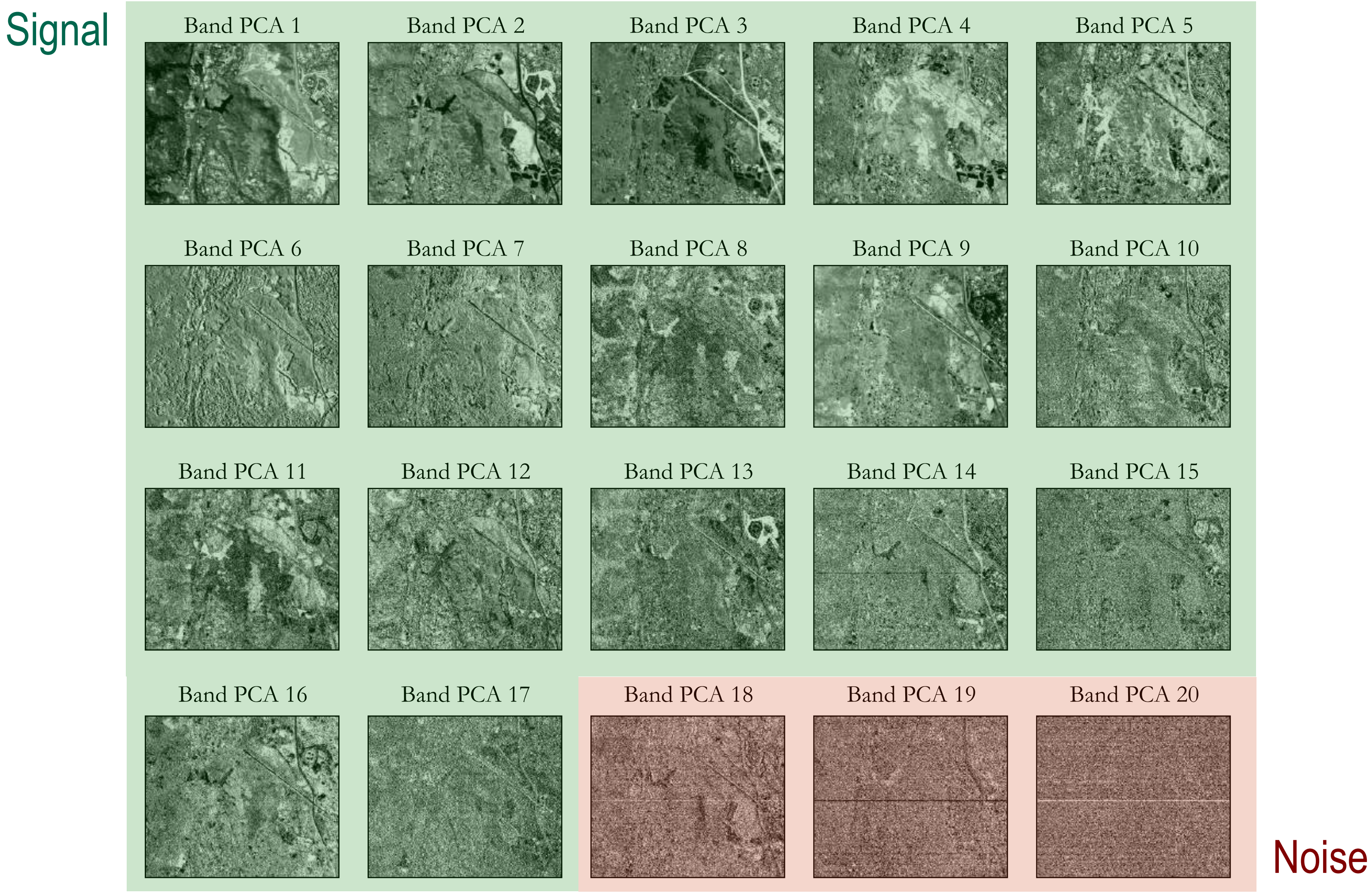
Classic methods for subspace estimation

- Determining the dimensionality of remotely sensed imagery is a challenging problem.
- The intrinsic dimensionality is defined as the minimum number of parameters needed to account for the observed properties of the data.
- Principal component analysis (PCA) transforms the data in a new coordinate system so that the number of significant components can be used as an estimate.



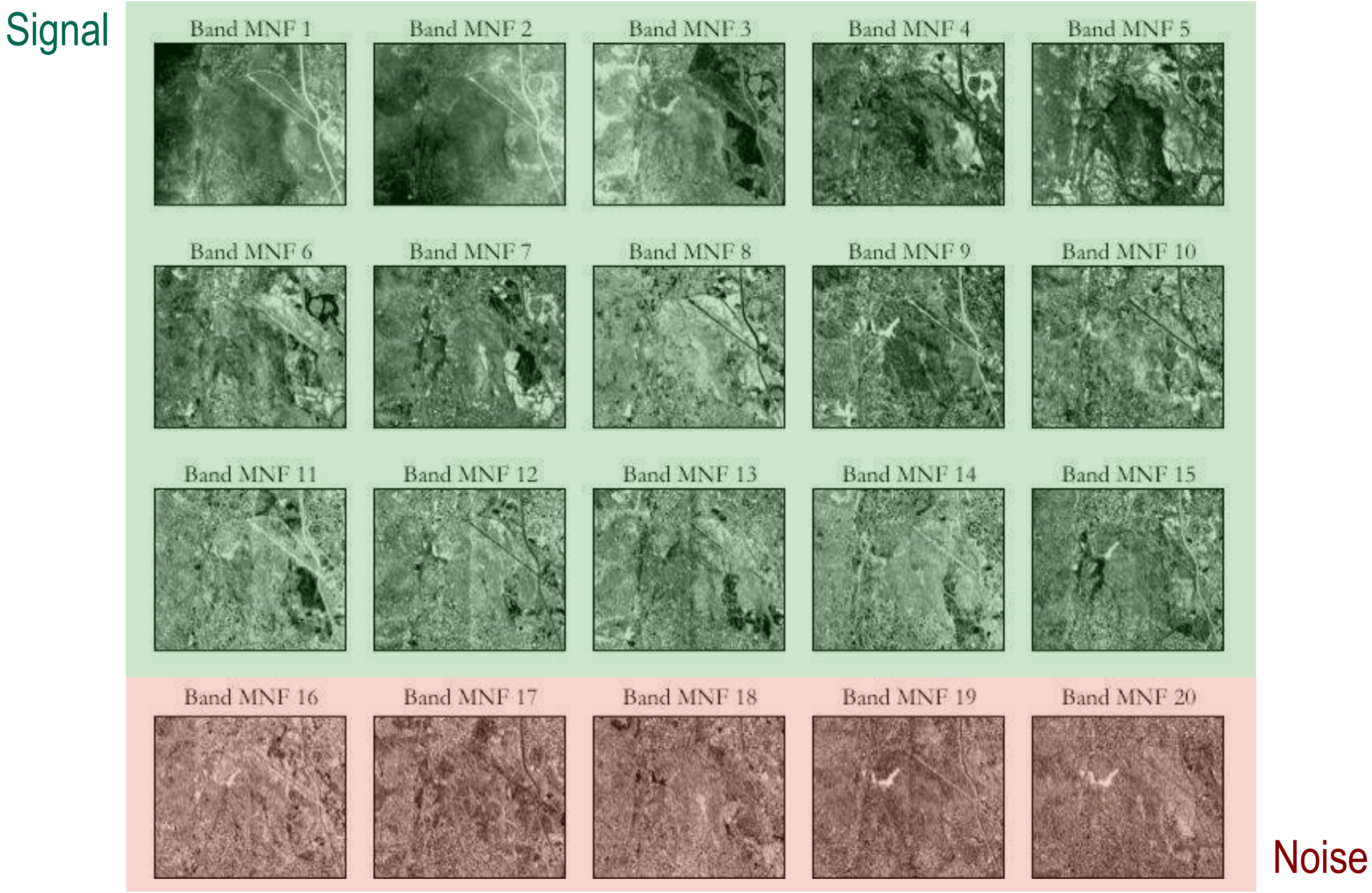
Classic methods for subspace estimation

- The resulting PCA components are ordered in descending order of data variance:



Classic methods for subspace estimation

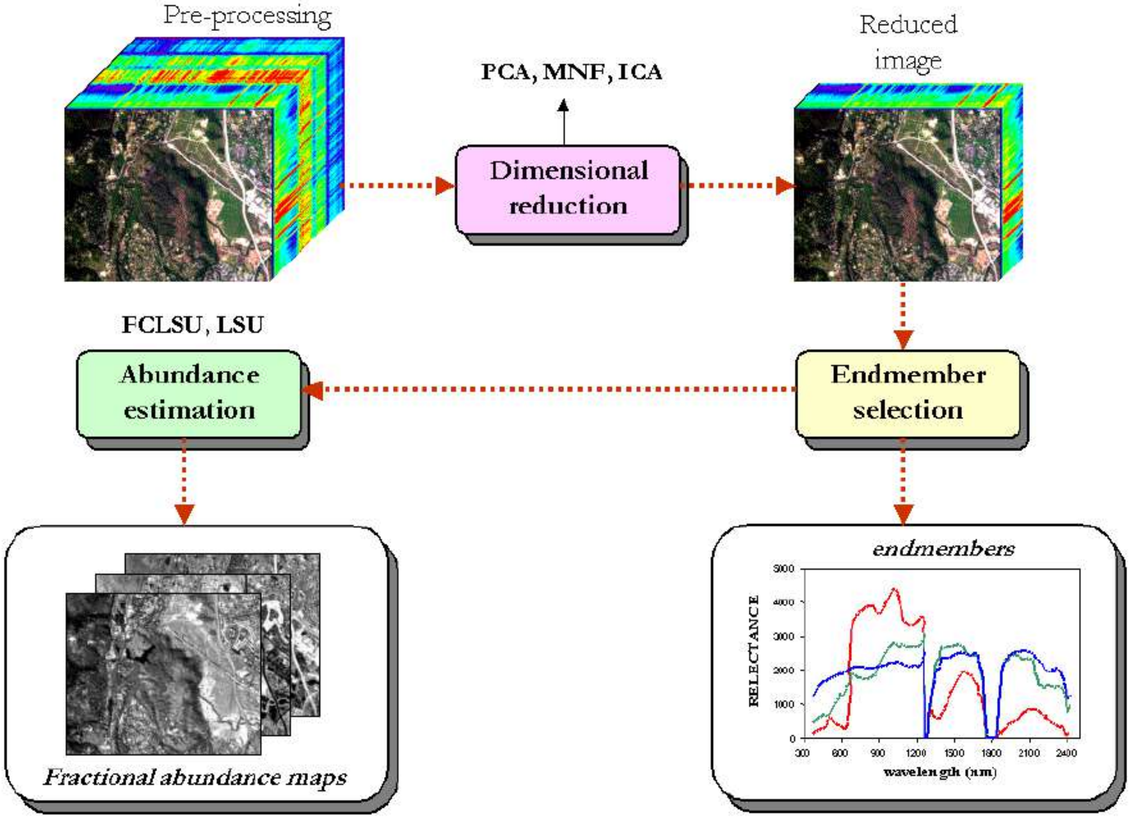
- Minimum noise fraction (MNF) orders the components in terms of signal to noise:



Contents

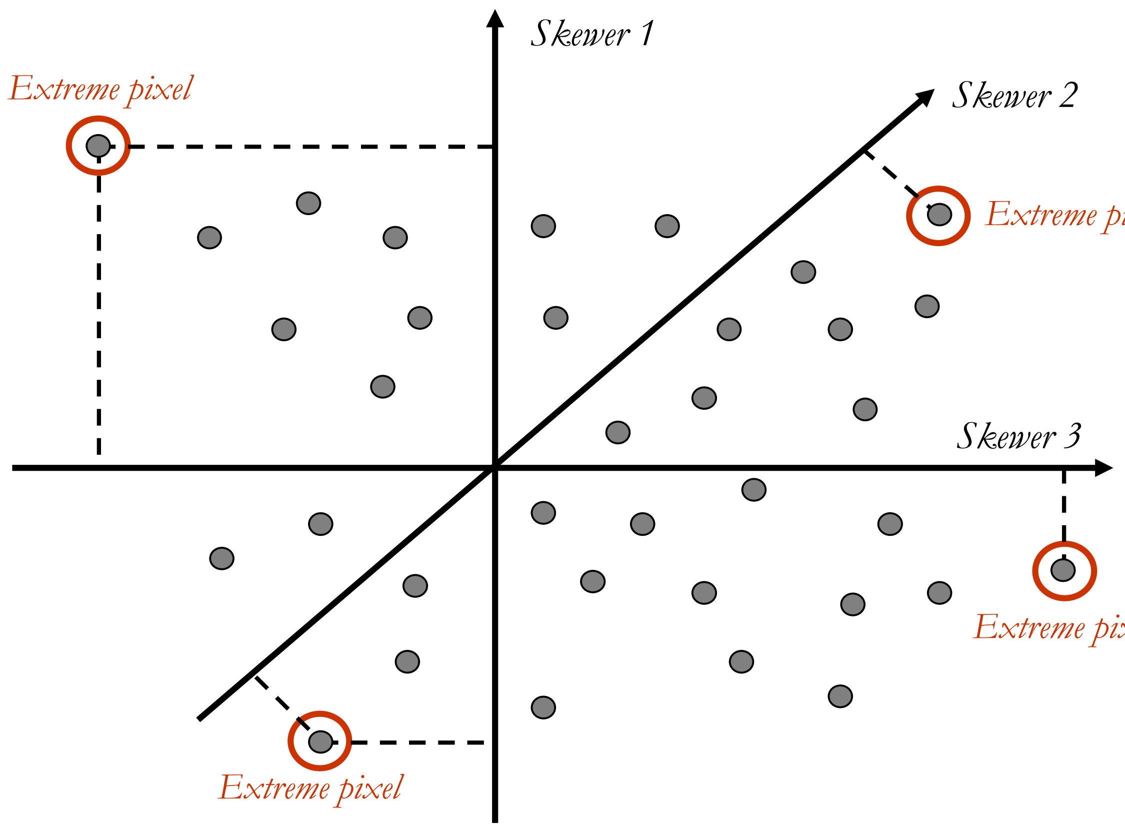
Classic methods for endmember extraction

- These methods assume a classic spectral unmixing chain made up of three stages: dimensional reduction, endmember selection and abundance estimation.
- Here, the endmembers are directly derived from the original hyperspectral scene.



Classic methods for endmember extraction

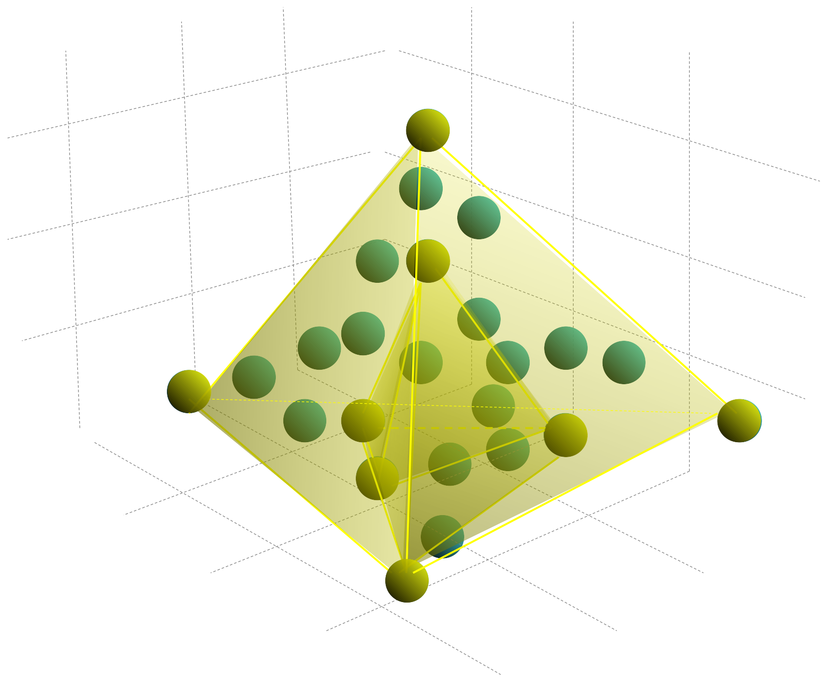
- The pixel purity index (PPI) is perhaps the most popular endmember extraction algorithm due to its availability in commercial software packages such as ENVI.



J. W. Boardman, F. A. Kruse and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," Proceedings of the Fifth JPL Airborne Earth Science Workshop, vol. 95, pp. 23-26, 1995.

Classic methods for endmember extraction

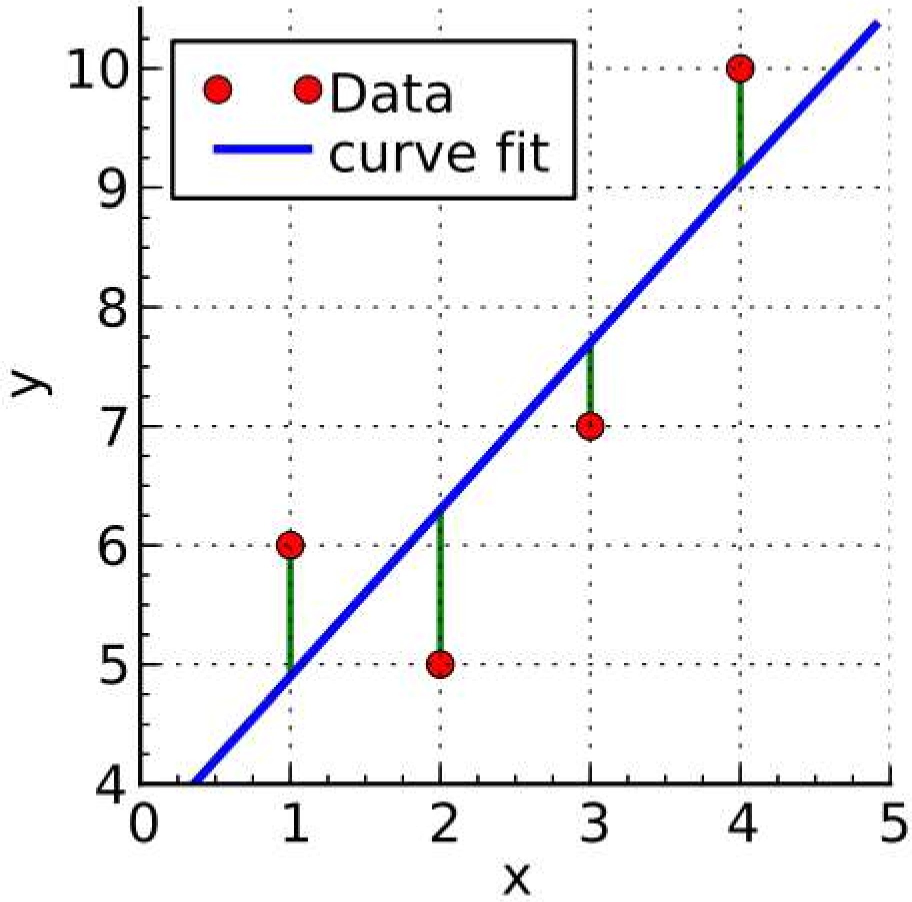
- The N-FINDR algorithm is also a very popular approach for endmember extraction.
- It assumes the presence of pure pixels in the original hyperspectral scene and further maximizes the volume that can be formed with pixel vectors in the data cube.



M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," Proceedings of SPIE, vol. 3753, pp. 266–270, Oct. 1999.

Contents

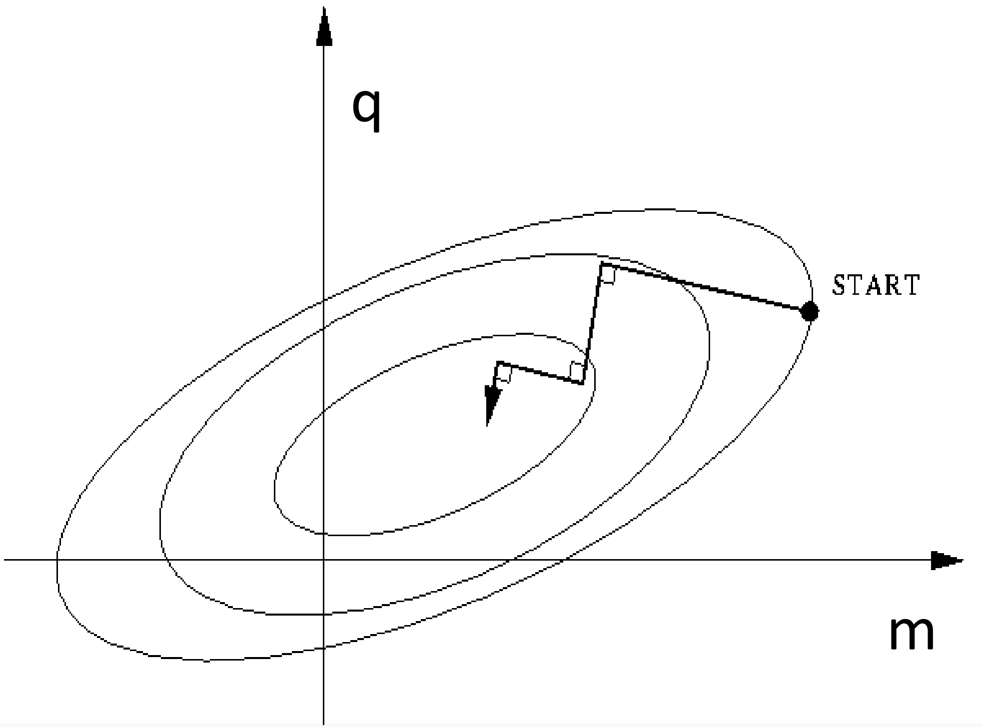
The idea of Least Squares



Find a line y=mx+q
in order to minimize the sum
of the square distances of the
data points from the line

Min(sum(dist(line, points)^2))

How do we get m & q in this example (in an intuitive but not effective way)?



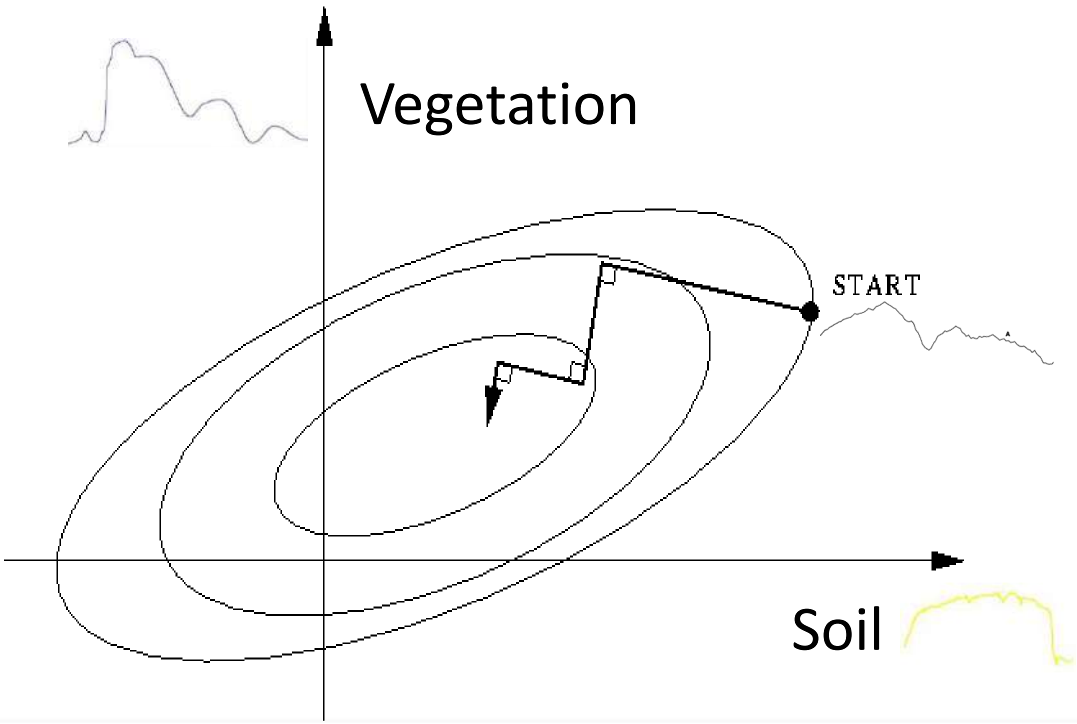
We have a surface in this space (think of it in 3d) which is given by the cost function for given values of m and q:
sum(dist(line, points)^2)

We select a random point to start and compute its first derivative

We move towards the direction of the derivative and compute again the cost function

We go on until we stop in the optimum value!

What if we have 2 endmembers?



We try to express our spectrum *S* as a linear combination of two sample spectra V and S, for example:

S‘ = 0.4 V + 0.5 S.

Our cost function is the difference (*S* - *S‘*)^2 between the original spectrum *S* and the reconstructed one *S‘*.

We take another step and compute again the cost function = distortion, until convergence!

Unconstrained least squares (UCLS)

- When all the endmember information (i.e., the number of endmembers and their spectral signatures) are known, abundances can be estimated by least squares.
- The idea is to find the abundances that minimize the reconstruction error obtained after approximating the original hyperspectral scene using a linear mixture model:

$$e = \left\| \mathbf{r} - \mathbf{M}\hat{\boldsymbol{\alpha}} \right\|^2$$

- Here, the least squares solution is given by the following simple term:

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{M}^T \mathbf{M}\right)^{-1} \mathbf{M}^T \mathbf{r}$$

- However, this is an unconstrained solution which does not satisfy the abundance non-negativity (ANC) and the abundance sum-to-one constraints (ASC).

A. Plaza, G. Martin, J. Plaza, M. Zortea and S. Sanchez,."Recent developments in spectral unmixing and endmember extraction, in: *Optical Remote Sensing - Advances in Signal Processing and Exploitation Techniques*. Edited by S. Prasad, L. Bruce and J. Chanussot, Springer, 2011, ISBN: 978-3-642-1241-6, pp. 235-268.

Contents

Non-negative constrained least squares

- If the ANC constraint needs to be satisfied, the problem of abundance estimation becomes a constrained optimization problem:

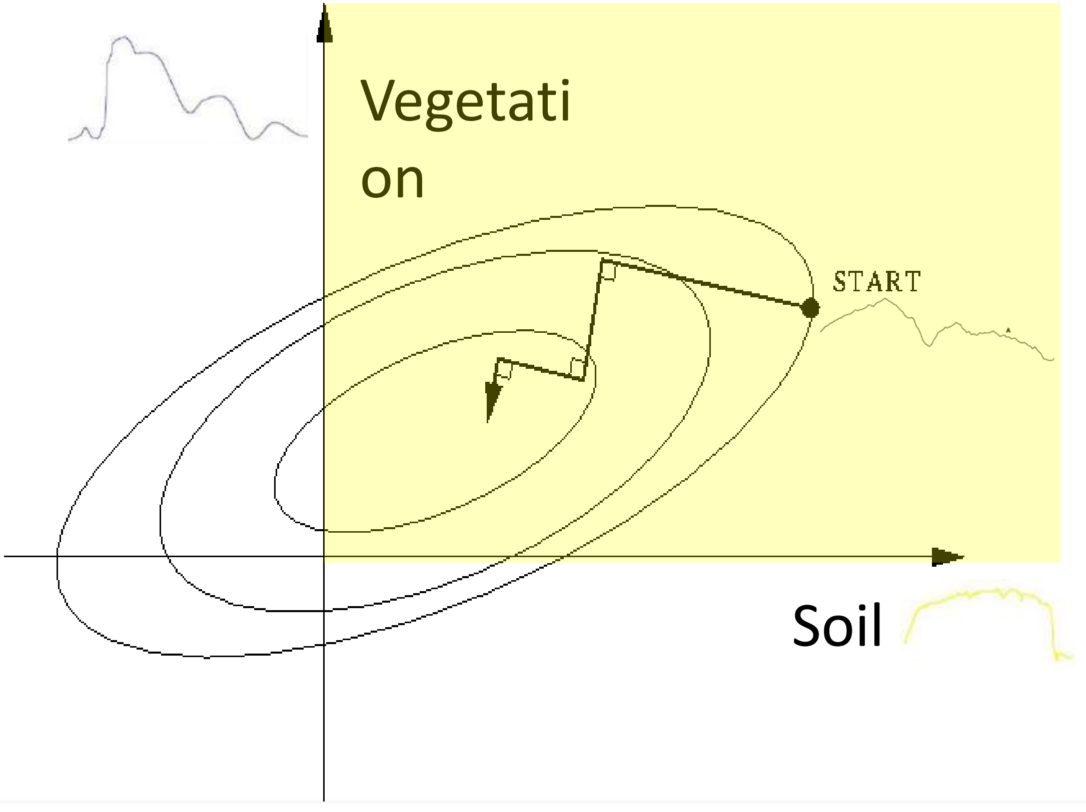
$$\min e = \min\ f(\boldsymbol{\alpha}) = \mathbf{r}^T\mathbf{r} - 2\mathbf{r}^T\mathbf{M}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{M}^T\mathbf{M}\boldsymbol{\alpha}$$

$$\text{Subject to}: 0 \le \alpha_i \le 1, \text{ for } 1 \le i \le p$$

- This optimization problem with inequality constraints can be solved effectively by means of quadratic programming since the objective function is a quadratic function.
- However, imposing the ANC constraint can significantly increase the computational complexity of the abundance estimation problem.
- Normally the ASC constraint alone is not imposed, but in conjunction with the ANC.
- When both ASC and ANC constraints need to be imposed in the abundance estimation model, we have a fully constrained problem (more difficult to solve).

C.-I Chang and D. Heinz, "Constrained subpixel detection for remotely sensed images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1144-1159, May 2000.

What happens with Non-negative Least Squares?



If we enforce the non-negativity constraint, we search for a solution only in the area where all parameters are positive….

In this case we have no problem, but if the optimum combination had a negative value for the abundance of the soil spectrum, we should have found the best solution (usually with soil = 0).

Contents

Fully constrained least squares unmixing

- If both the ANC and the ASC constraints need to be satisfied, the problem of abundance estimation becomes an even more complicated one:

$$\min e = \min f(\boldsymbol{\alpha}) = \mathbf{r}^T\mathbf{r} - 2\mathbf{r}^T\mathbf{M}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{M}^T\mathbf{M}\boldsymbol{\alpha}$$

$$\text{Subject to}: \alpha_1 + \alpha_2 + \cdots + \alpha_p = 1$$

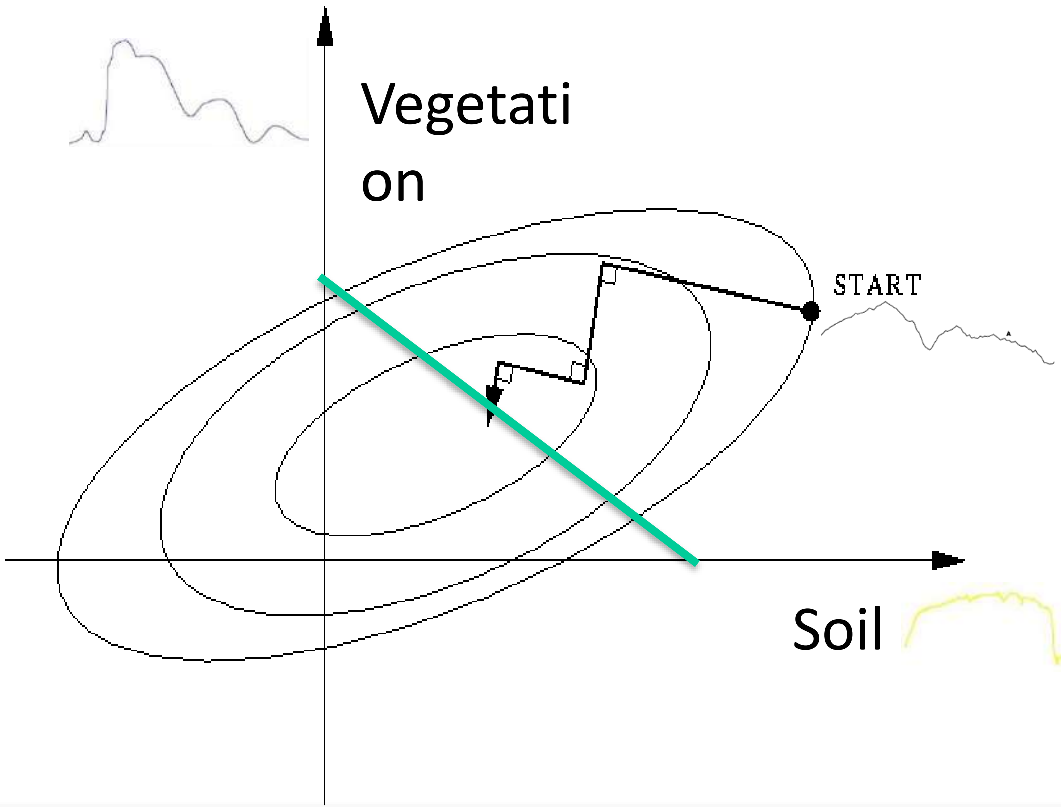
$$0 \le \alpha_i \le 1, \text{ for } 1 \le i \le p$$

- Fortunately, the ASC can be easily included in the ANC-constrained formulation by simply adding a row vector with all elements set to one to the endmember matrix, adding an element one to the pixel vector, and solving the resulting least squares problem as follows:

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M} \\ \mathbf{1} \end{bmatrix} \qquad \tilde{\mathbf{r}} = \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix} \qquad \tilde{\mathbf{r}} = \tilde{\mathbf{M}}\boldsymbol{\alpha} + \mathbf{n}$$

D. Heinz and C.-I Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529-545, 2001.

What happens with Fully Constrained Least Squares?



If we enforce the sum-to-one constraint, we search for a solution only in the area where the sum of all abundances is one….

It can lead to unrealistic results as it restricts too much the search space

Daniele Cerra

German Aerospace Center (DLR)

Remote Sensing Technology Institute

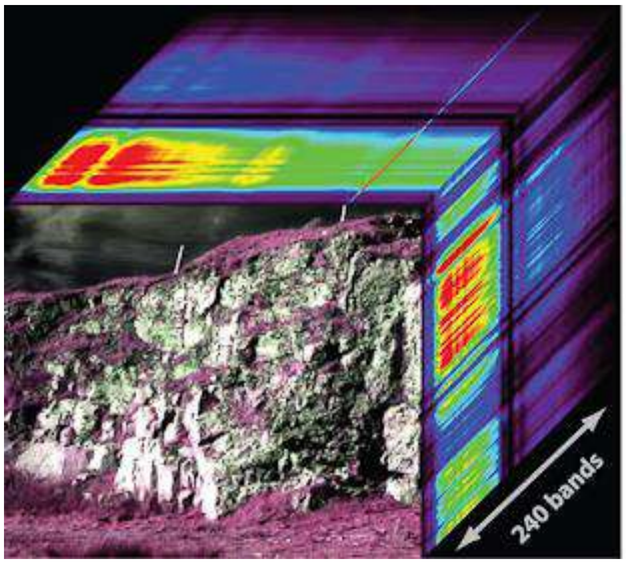
Analysis of Hyperspectral Images

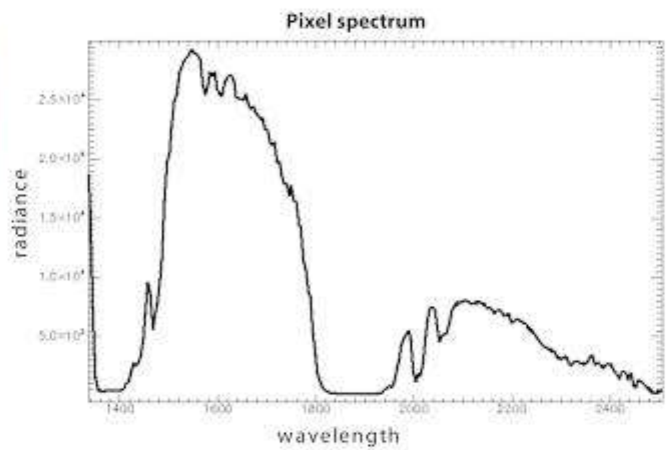
Band Selection

DLR

Problem

– We have a hyperspectral image…



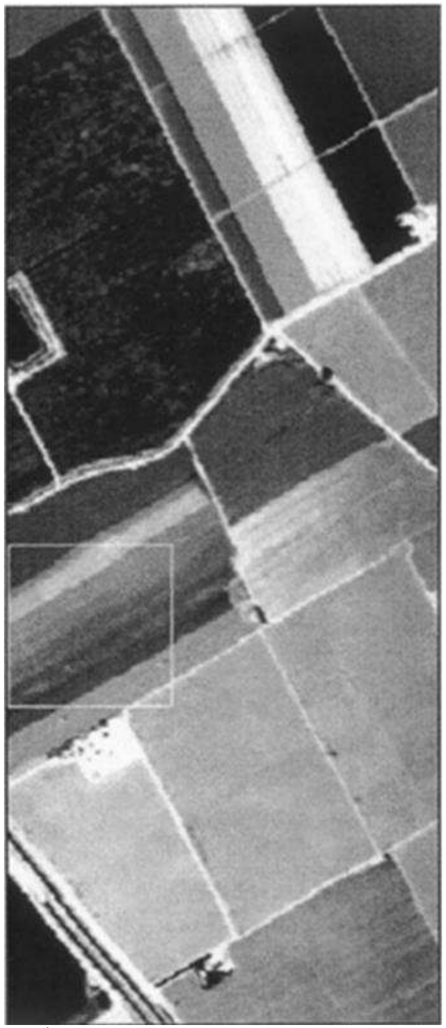


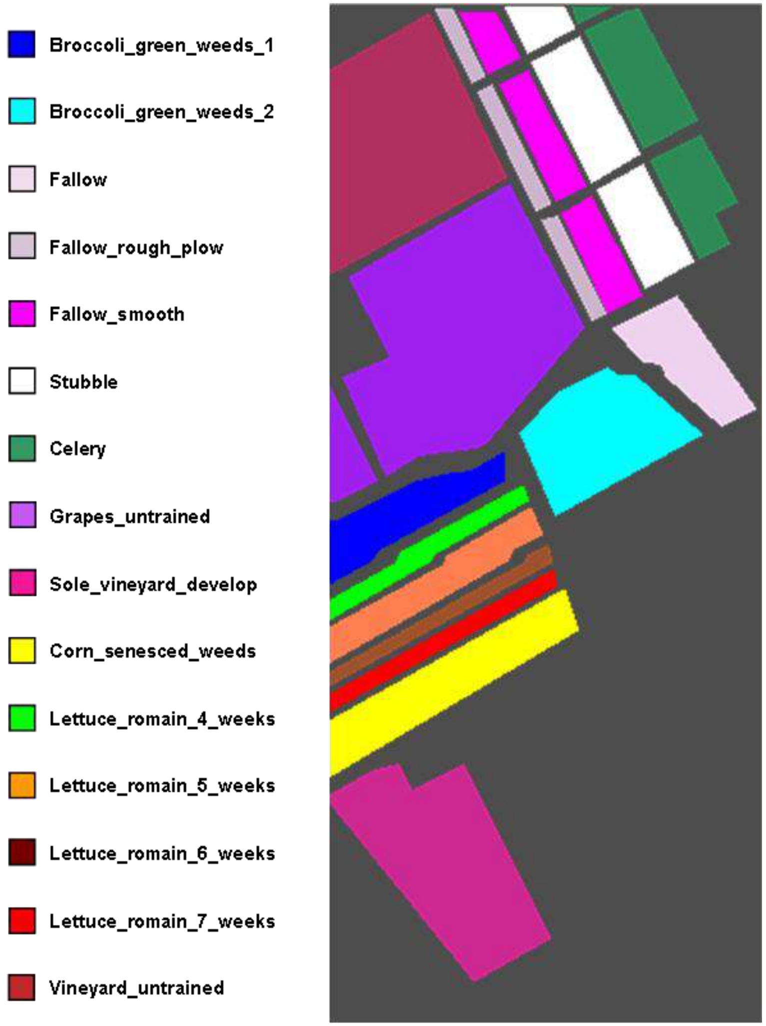
– ..and we want to classify it using a reduced number of dimensions

  – We want to avoid overfitting – curse of dimensionality

  – We do not have „almighty“ computers ☺

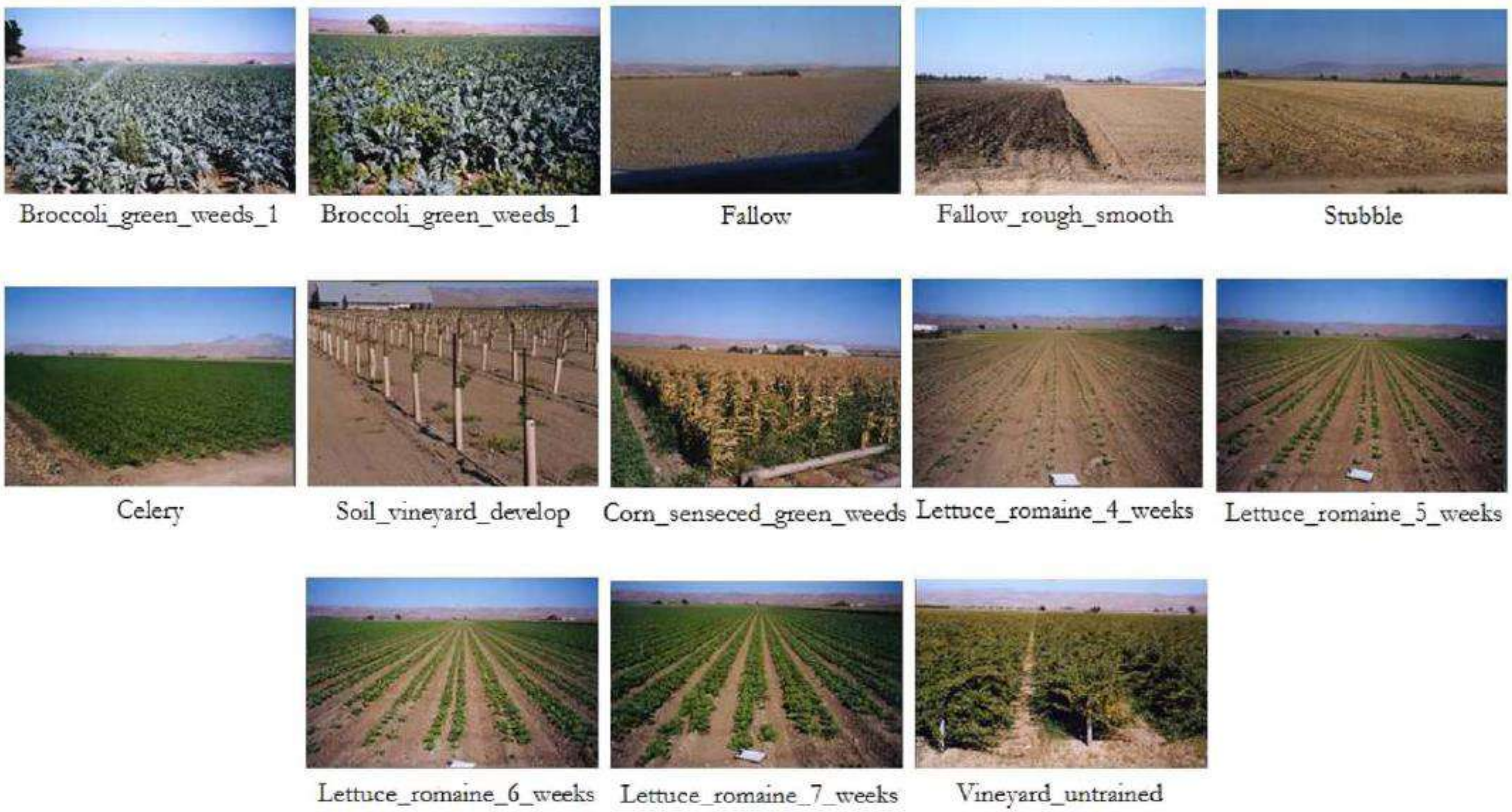
Sample image: Salinas AVIRIS Dataset

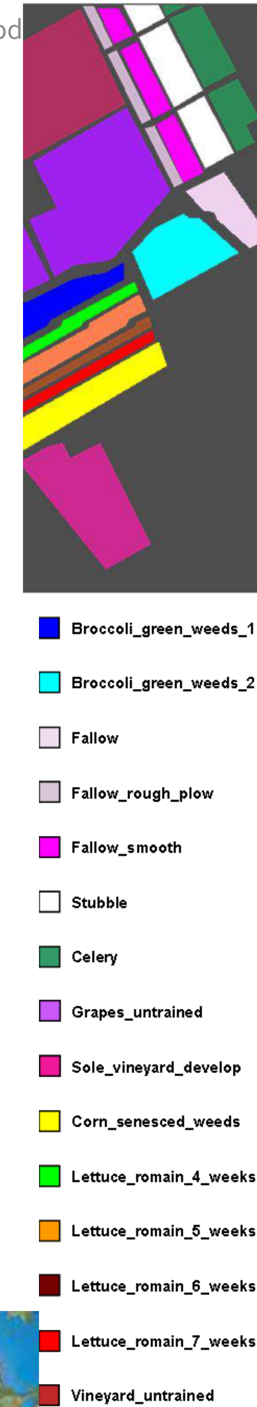




- Widely used as benchmark dataset
- 512 x 217 pixels
- 224 bands
- 4 m resolution
- 15 classes
- Several crops
- Some classes very similar
  - Broccoli 1 & 2
  - Grapes & Vineyard
  - Lettuces

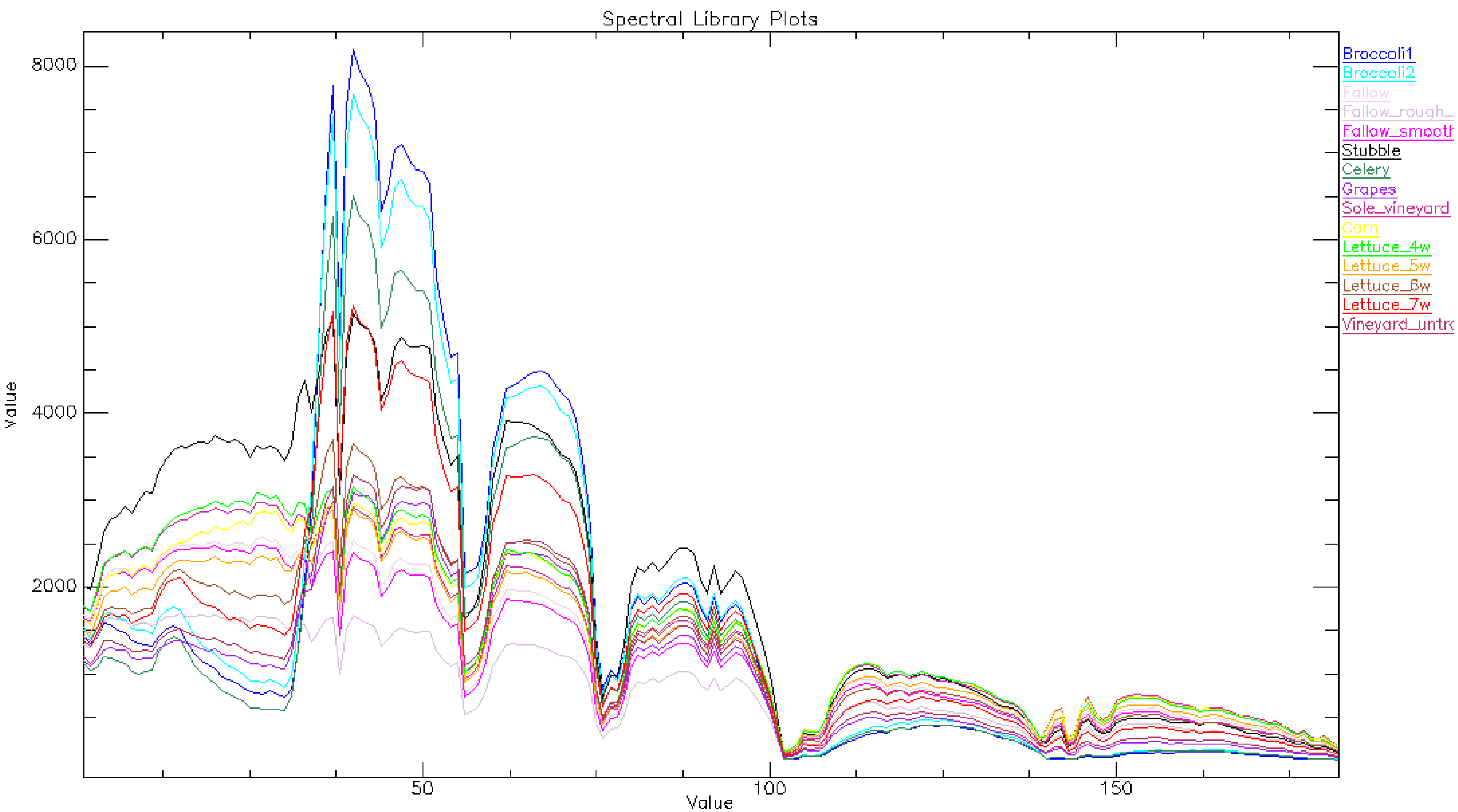
Salinas Dataset

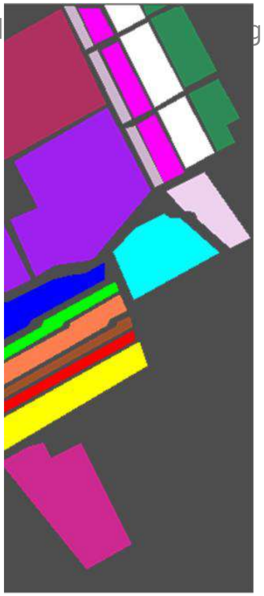




DLR

Sample Spectra





Broccoli_green_weeds_1
Broccoli_green_weeds_2
Fallow
Fallow_rough_plow
Fallow_smooth
Stubble
Celery
Grapes_untrained
Sole_vineyard_develop
Corn_senesced_weeds
Lettuce_romain_4_weeks
Lettuce_romain_5_weeks
Lettuce_romain_6_weeks
Lettuce_romain_7_weeks
Vineyard_untrained

DLR

Problem

– How to select the “best“ bands?

– For example, we want to select 10 bands

– Let‘s see how…

How to select these 10 bands?

- Several methods of band selection
- Let‘s do a small „journey“ into statistics up to the concept of mutual information
- What is the relationship between pixel values in a band and the amount of information they contain?

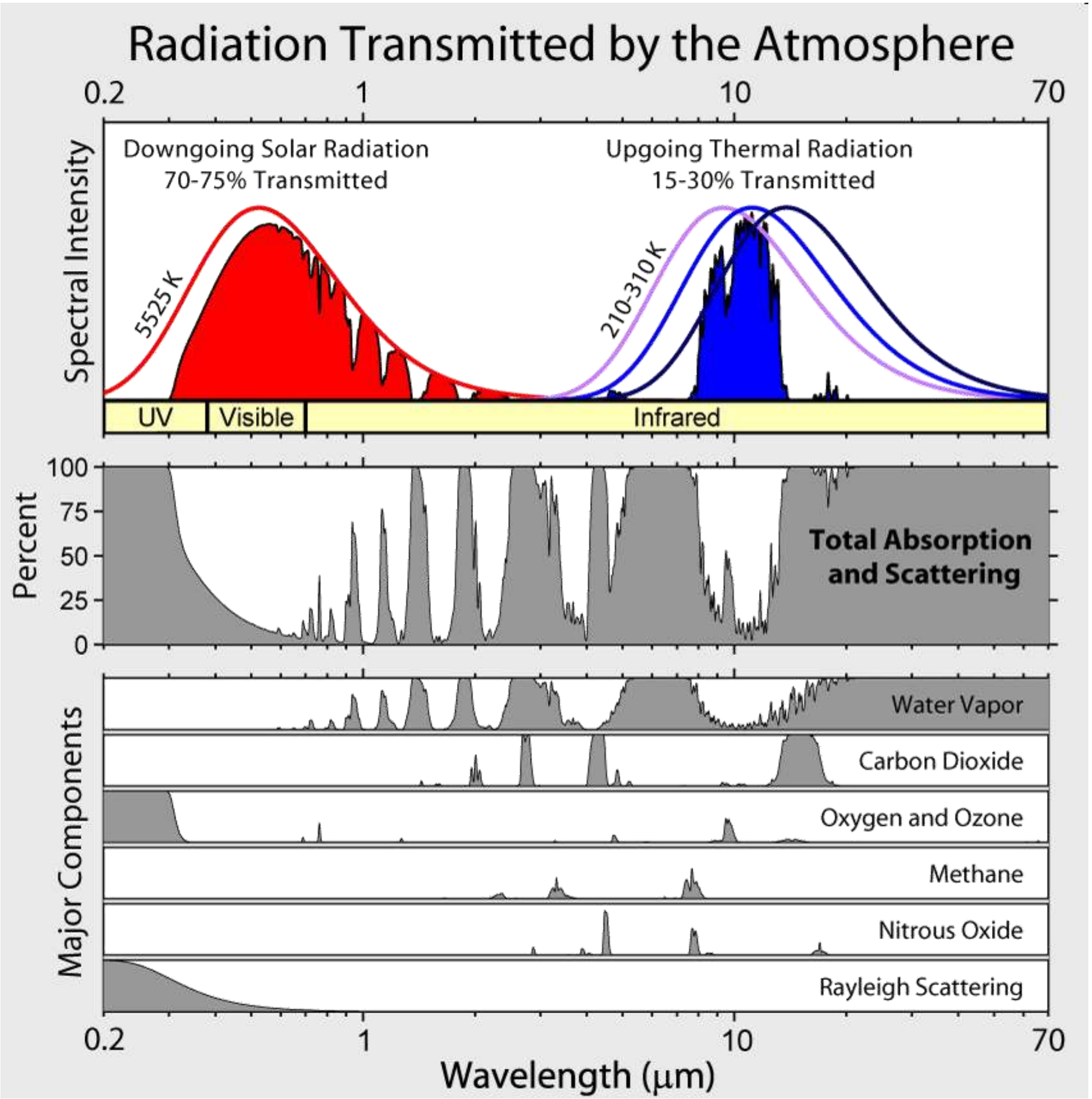
Solution 1

– Bands 1-10

Really?

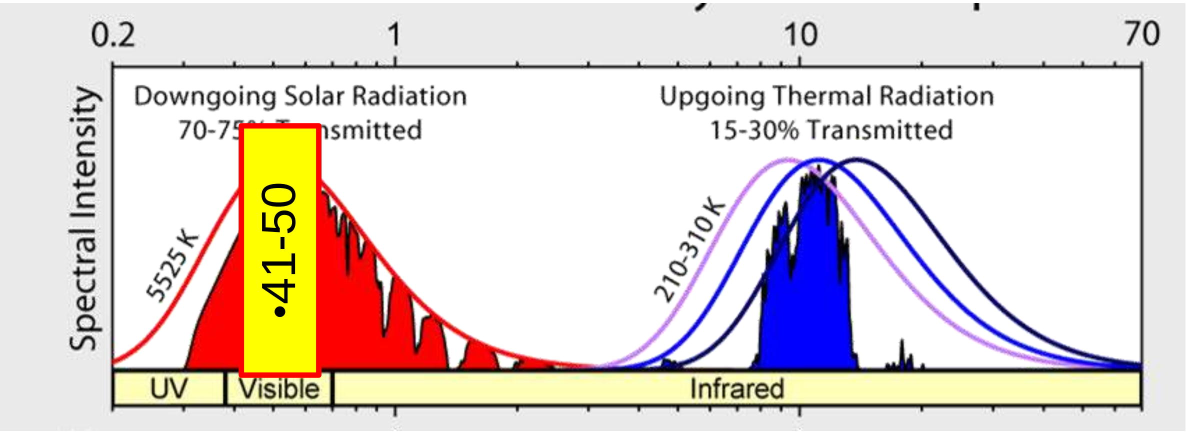
Band 1

- Noisy bands are not used in the analysis
- Why are there noisy bands?

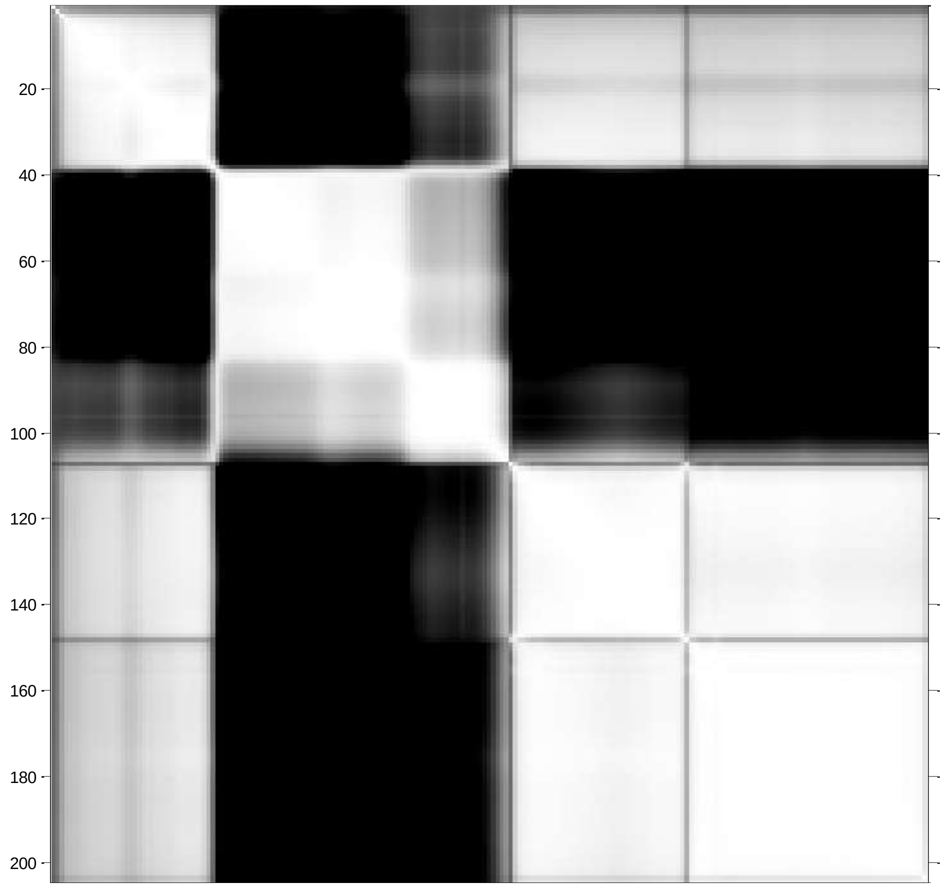

Radiation Transmitted by the Atmosphere
0.2
1
10
70
Spectral Intensity
Downgoing Solar Radiation
70-75% Transmitted
Upgoing Thermal Radiation
15-30% Transmitted
5525 K
210-310 K
UV
Visible
Infrared
Percent
100
75
50
25
0
Total Absorption
and Scattering
Major Components
Water Vapor
Carbon Dioxide
Oxygen and Ozone
Methane
Nitrous Oxide
Rayleigh Scattering
Wavelength (μm)

Solution 2

– Bands 41-50
  – In the range with the best Signal-to-Noise Ratio (SNR)



Correlation between bands



Matriz de correlacion entre bands para la imagen Salinas

Mean

	Score X	$X - \bar{X}$	$(X - \bar{X})^2$
1	3		
2	5		
3	7		
4	10		
5	10		
Totals	35		

- The mean is 35/5=7.

Standard Deviation

	Score X	$X - \overline{X}$	$(X - \overline{X})^2$
1	3	3-7=-4	
2	5	5-7=-2	
3	7	7-7=0	
4	10	10-7=3	
5	10	10-7=3	
Totals	35	12	

- The (population) SD is the square root of the squared mean value of the difference from the mean:

- Sdev(X) = $\sqrt{\frac{4^2+2^2+0^2+3^2+3^2}{5}} = 2.76$

Variance

	Score X	$X - \overline{X}$	$(X - \overline{X})^2$
1	3	3-7=-4	16
2	5	5-7=-2	4
3	7	7-7=0	0
4	10	10-7=3	9
5	10	10-7=3	9
Totals	35		38

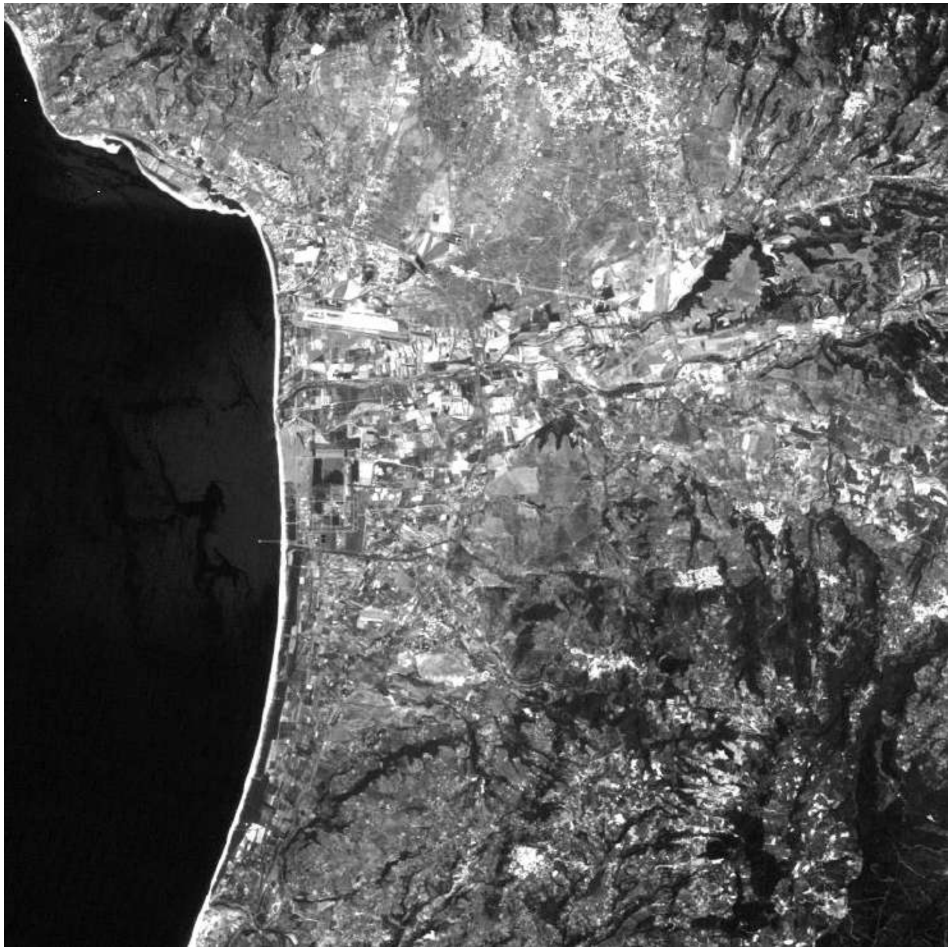
Variance

	Score X	$X - \bar{X}$	$(X - \bar{X})^2$
1	3	3-7=-4	16
2	5	5-7=-2	4
3	7	7-7=0	0
4	10	10-7=3	9
5	10	10-7=3	9
Totals	35		38

$$s^2 = \frac{\sum \left(x - \bar{X}\right)^2}{n} = \frac{38}{5} = 7.6$$

DLR

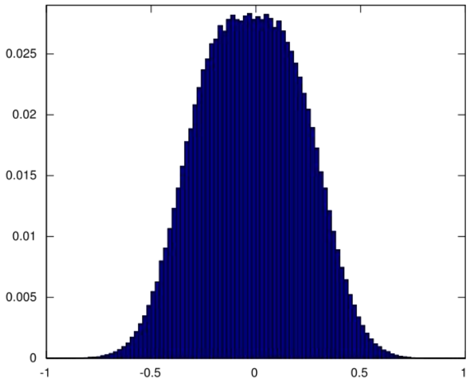
Example



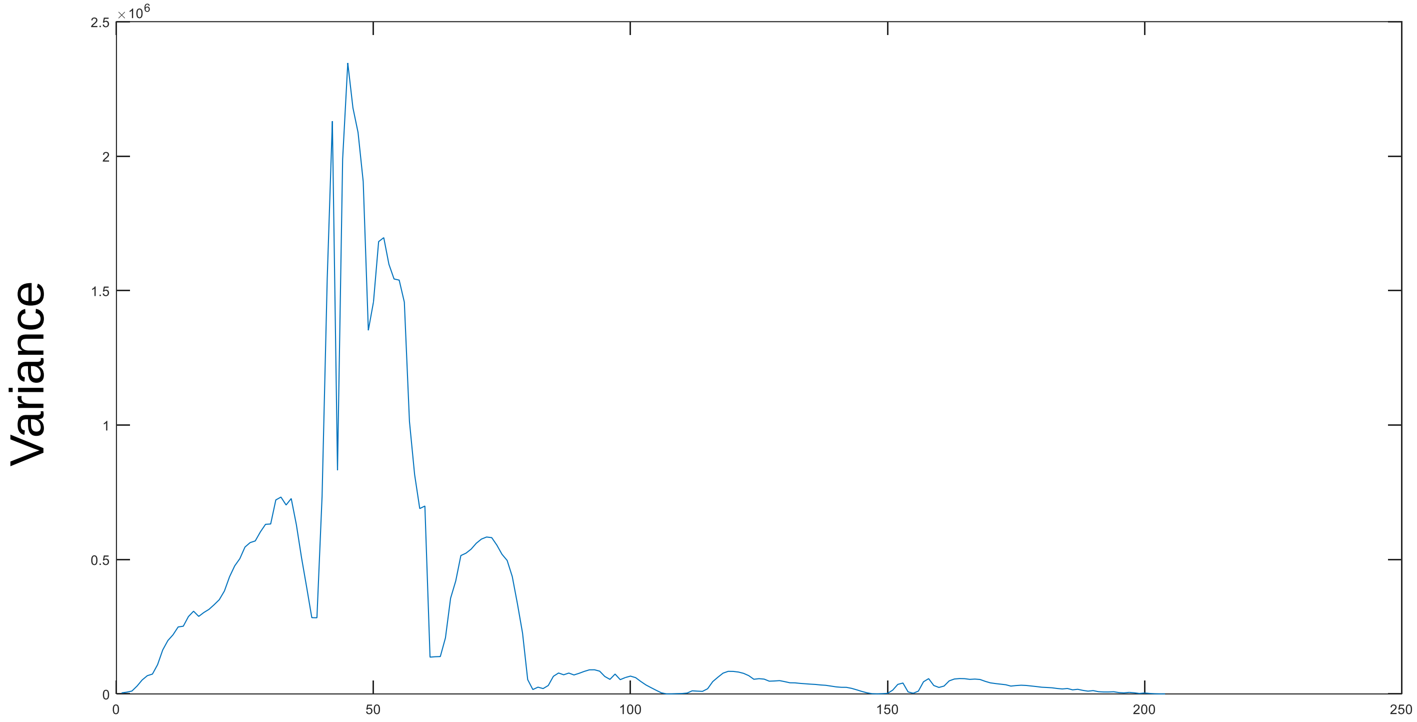
Local Variance in a 7x7 Sliding Window

What about hyperspectral data?

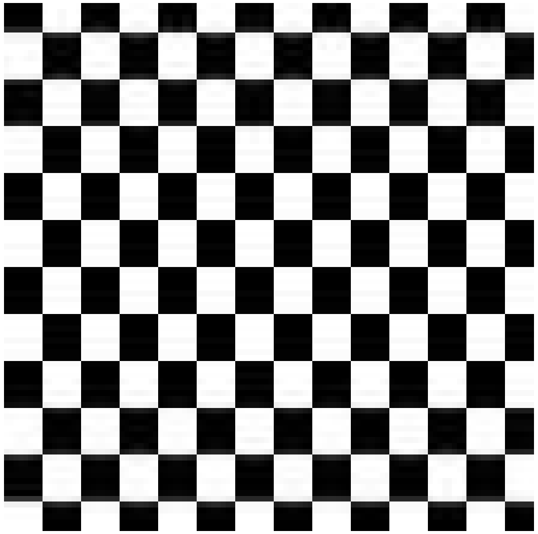
- Each image has hundreds of bands
- Each band has a histogram
- We can compute the variance of each histogram!
- Higher variance -> higher information
  - Neglecting Noise Influences

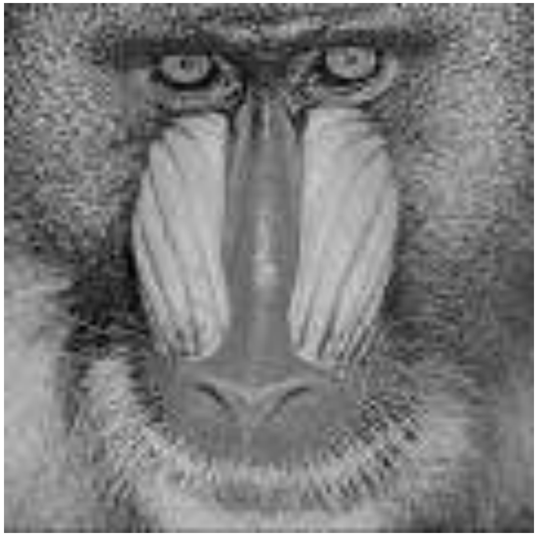


Really + Variance → + information?

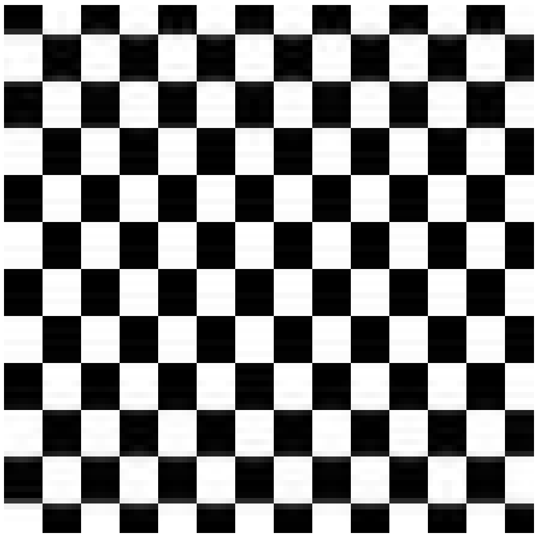


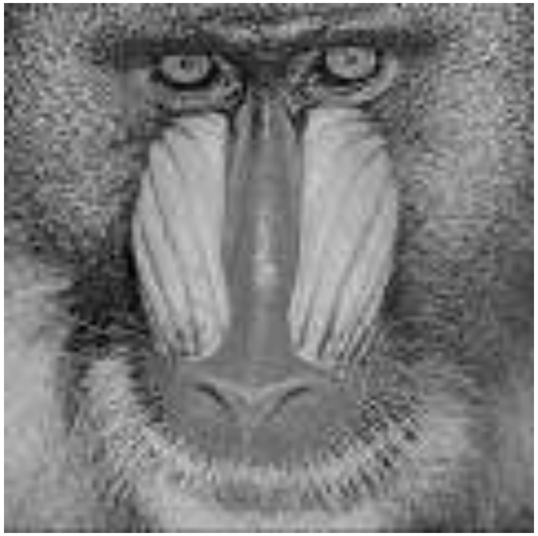
Which image has a higher variance?





And which image contains more information?





Entropy in a nut-shell

Low Entropy:

location of soup

High Entropy: location of soup

Entropy in a nut-shell

Low Entropy

..the values (locations of soup) sampled entirely from within the soup bowl

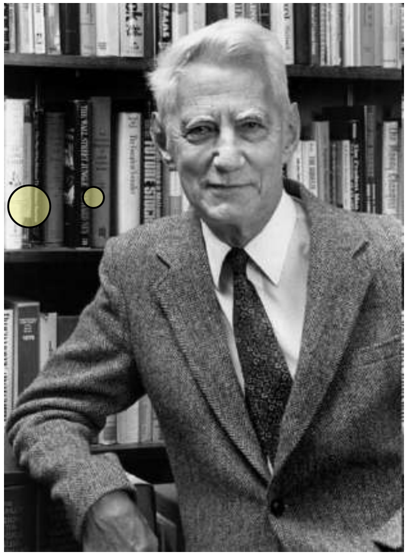
High Entropy

..the values (locations of soup) unpredictable... almost uniformly sampled throughout our dining room

DLR

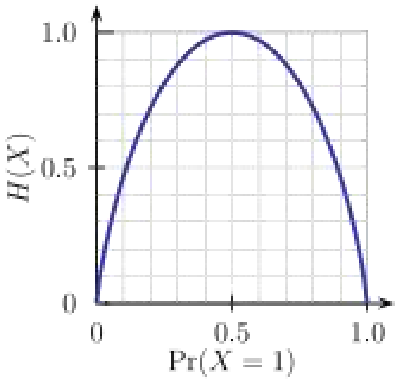
Shannon Entropy

$$H(X) = -\sum_{x} p(x) \log p(x)$$



- Information content of the output of a random variable *X*

- Example: Entropy of the outcomes of the toss of a biased/unbiased coin
  - Max H(X) -> Coin not biased
    - Every toss carries a full bit of information!
  - Note: H(X) can be (much) greater than 1 if the values that X can take are more than two!



Two bits of Entropy (source:wikipedia)

$$H(X) = -\sum_{x} p(x) \log p(x)$$

Two bits of Entropy (source:wikipedia)

$$H(X) = -\sum_{x} p(x)\log p(x) \rightarrow \sum_{1:4} \frac{1}{4} 2$$

Bits

You are watching a set of independent random samples of X

You see that X has four possible values

P(X=A) = 1/4	P(X=B) = 1/4	P(X=C) = 1/4	P(X=D) = 1/4

So you might see: BAACBADCDADDDA…

You transmit data over a binary serial link. You can encode each reading with two bits (e.g. A = 00, B = 01, C = 10, D = 11)

0100001001001110110011111100…

Fewer Bits

Someone tells you that the probabilities are not equal

P(X=A) = 1/2	P(X=B) = 1/4	P(X=C) = 1/8	P(X=D) = 1/8

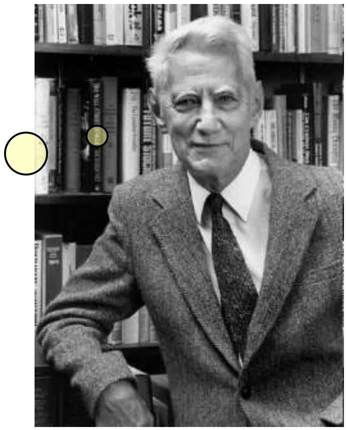
It's possible…

…to invent a coding for your transmission that only uses 1.75 bits on average per symbol. How?

A	0
B	10
C	110
D	111

(This is just one of several ways)

$$H(X) = -\sum_{x} p(x) \log p(x)$$



- X={A,B,C,D,E}
- We should use 3 bits per symbol to encode the outcomes of *X*

x	A	B	C	D	E
P(x)	**2/5**	**1/5**	**1/5**	**1/10**	**1/10**

⇔

Symbol	A	B	C	D	E
Code	**00**	**01**	**10**	**110**	**111**

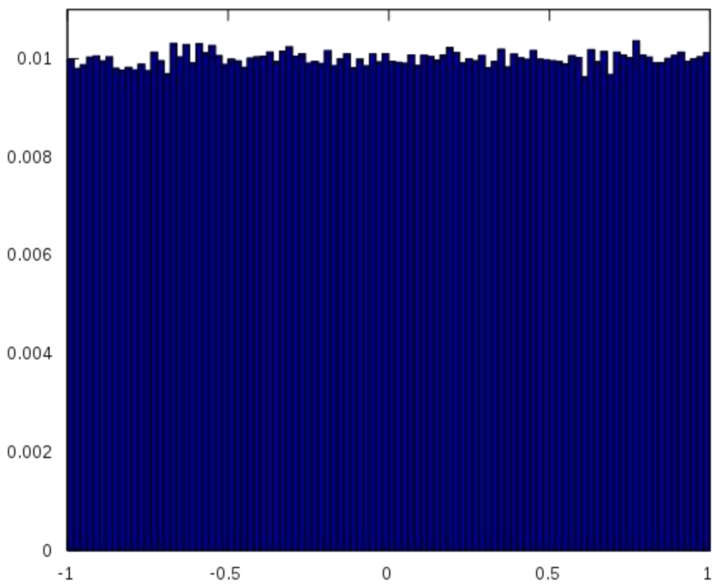
$$2\left(\frac{2}{5}+\frac{1}{5}+\frac{1}{5}\right)+3\left(\frac{1}{10}+\frac{1}{10}\right)=2.2$$

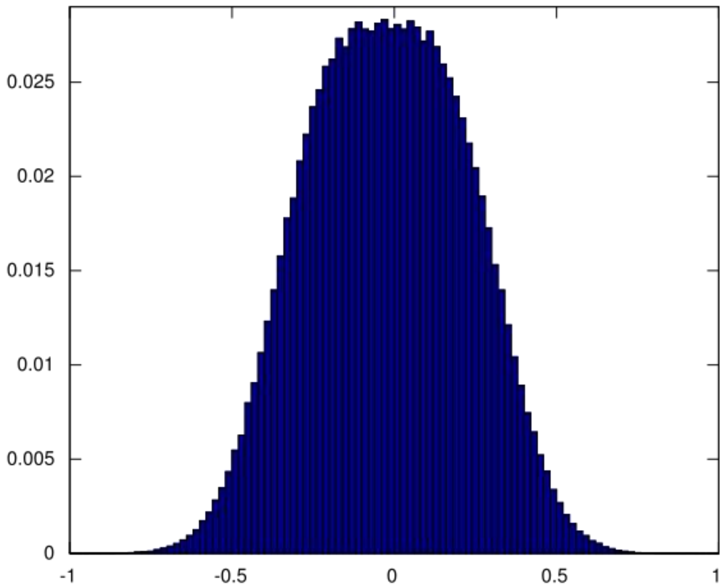
- Bits per Symbol in average

Compression is achieved!

DLR

Test: Which distribution has higher entropy?





Entropy in a nut-shell

- Low Entropy

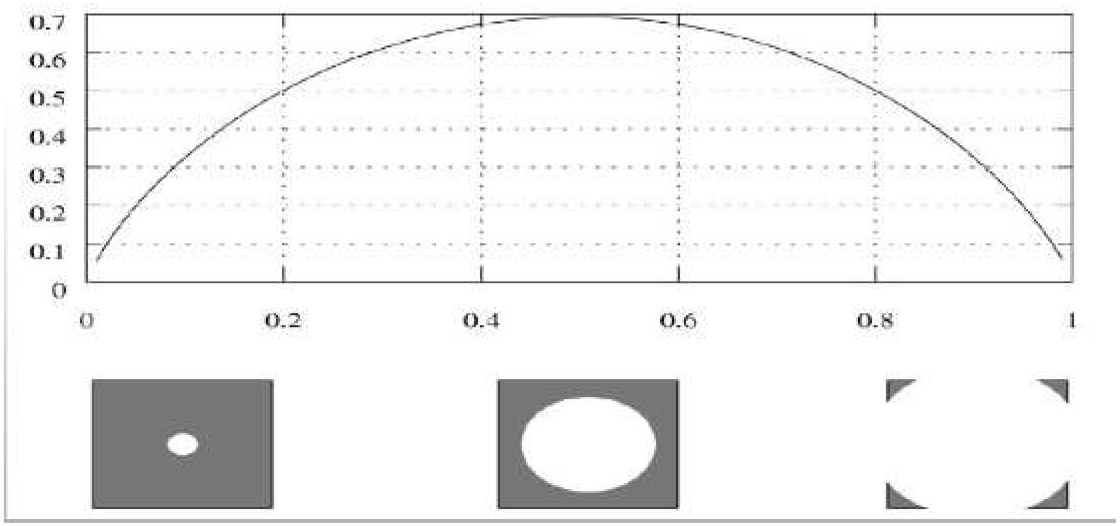
- ..the values (locations of soup) sampled entirely from within the soup bowl

- High Entropy

- ..the values (locations of soup) unpredictable... almost uniformly sampled throughout our dining room

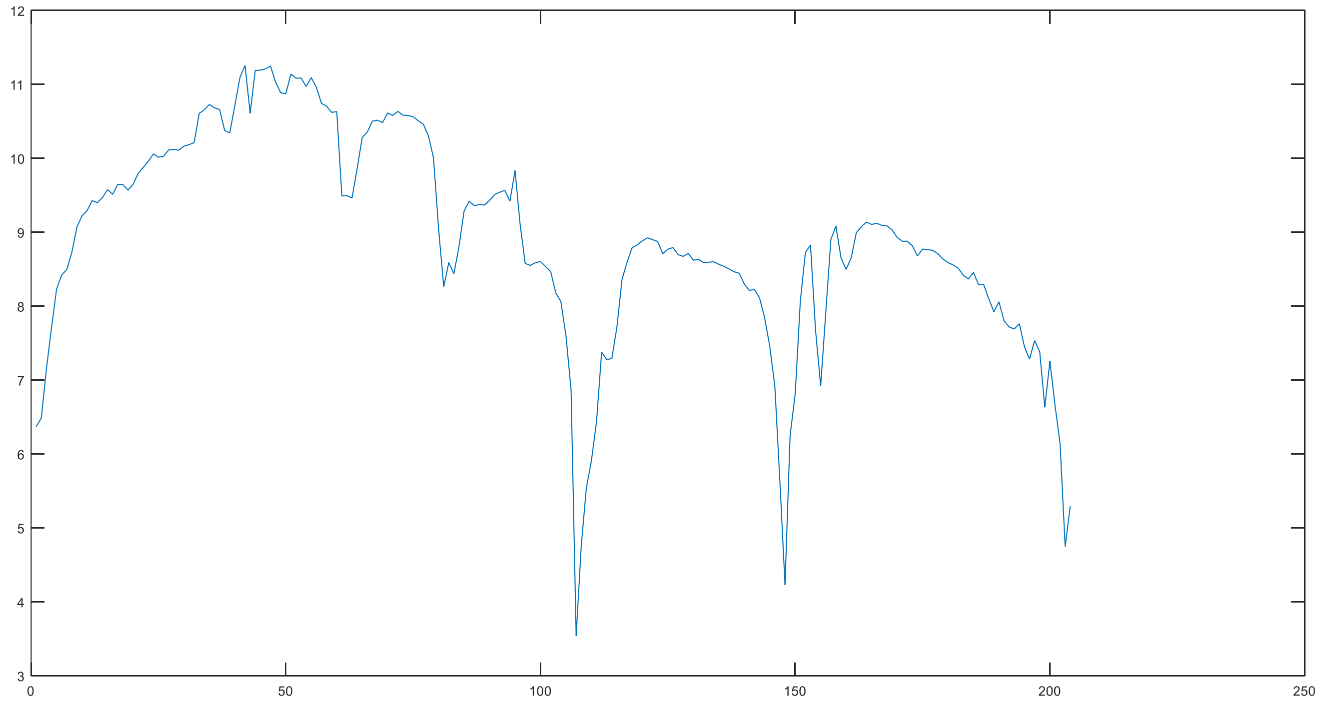
DLR

Entropy: A Binary Image Example

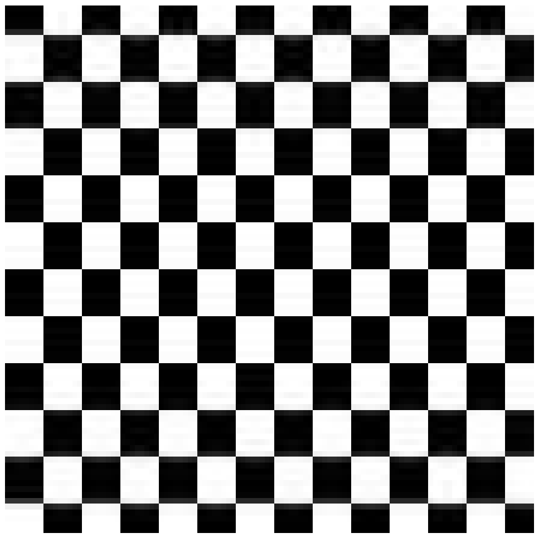


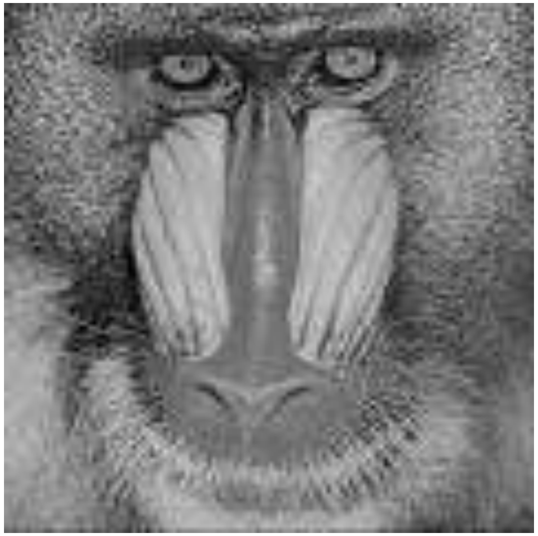
•How many bits of information is conveyed by each pixel?

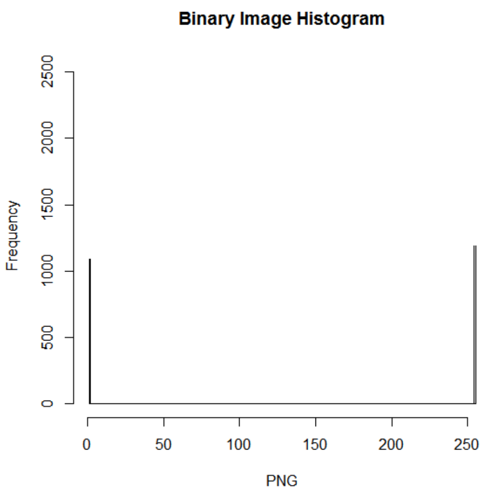
Entropy in the bands of the Salinas dataset

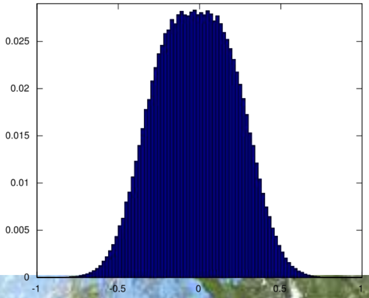


Which image has higher entropy?

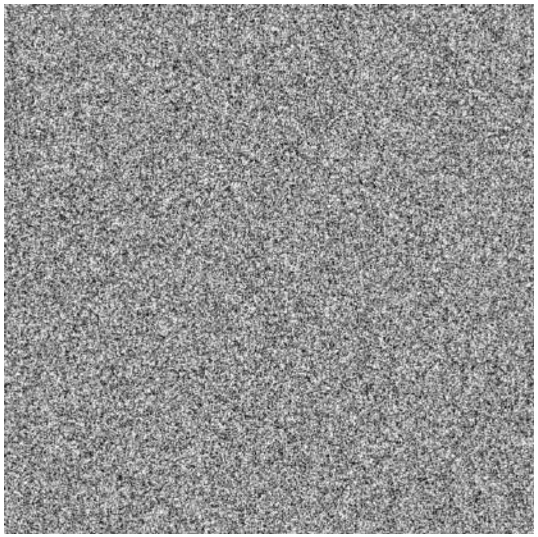


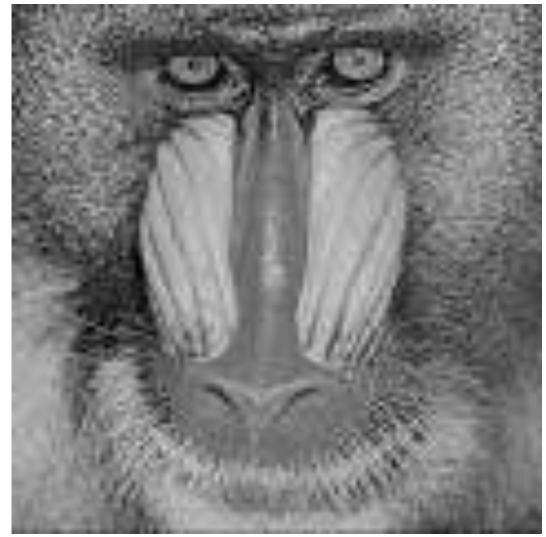


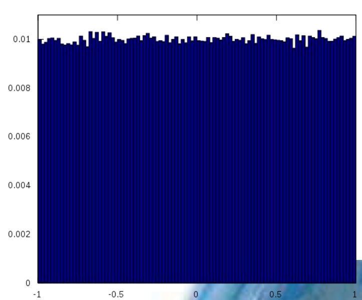


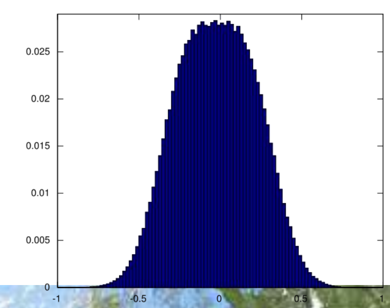


..and now?

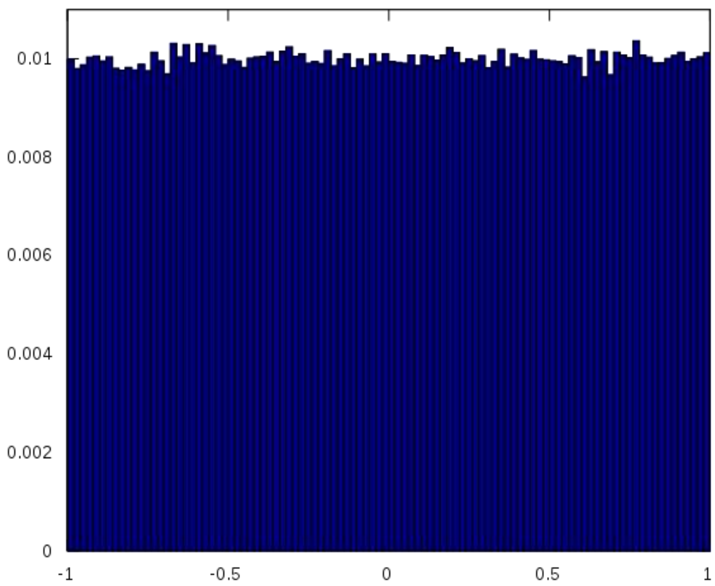


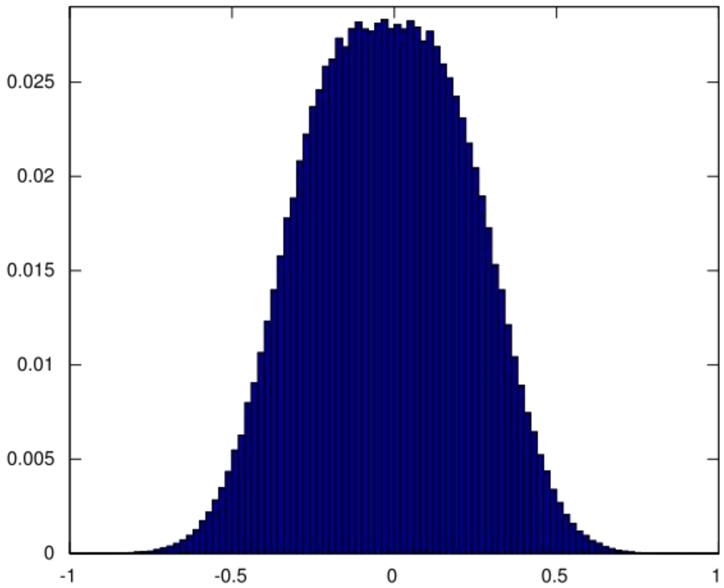



0.01
0.008
0.006
0.004
0.002
0
-1
-0.5
0
0.5
1


0.025
0.02
0.015
0.01
0.005
0
-1
-0.5
0
0.5
1

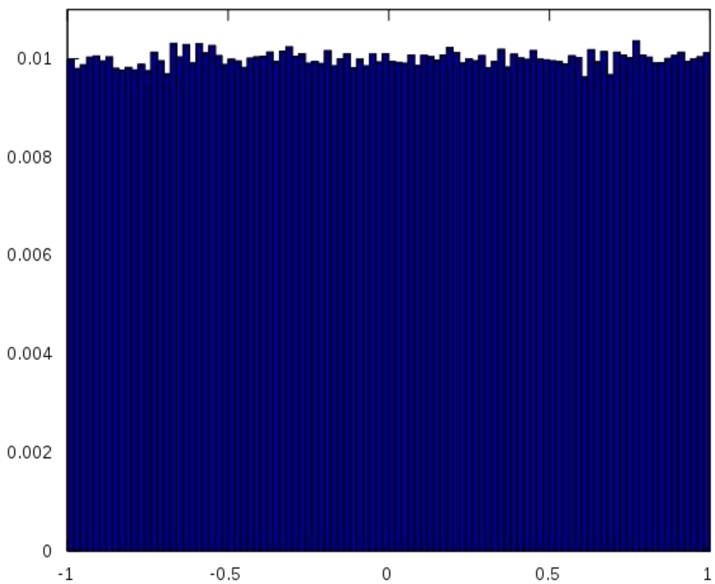
Test: Which distribution has higher entropy?

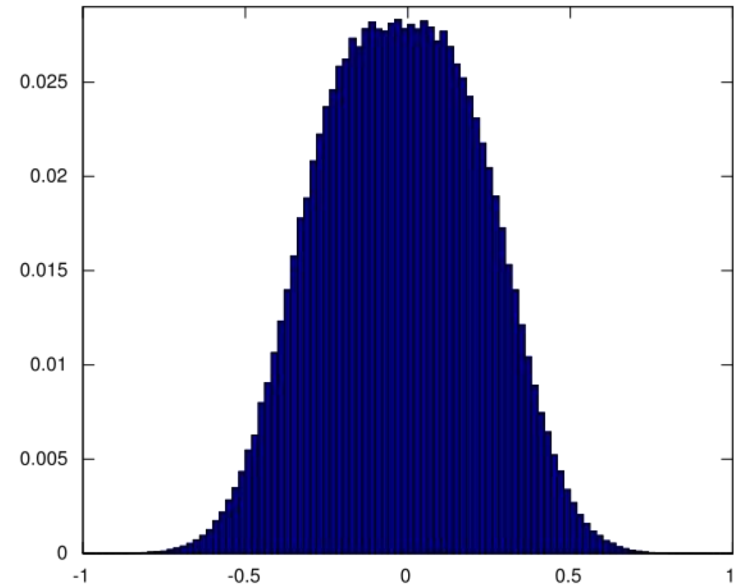




DLR

Remember: Histogram of noise is flat! Noise has maximum entropy/information!



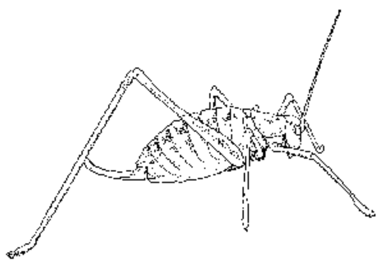


•To use this concept in the best way, we must relate it to the objective of our application, Let's see what happens when we use it to select the best parameters for a classification procedure!

DLR

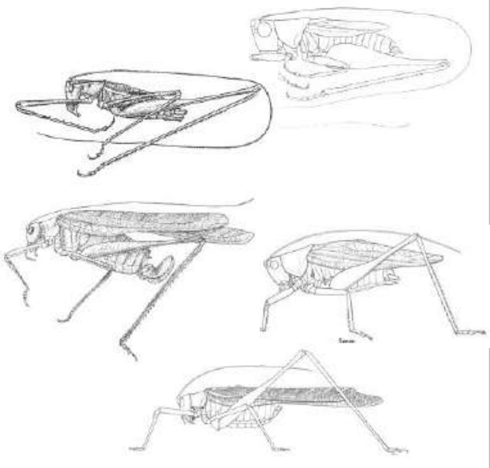
The Classification Problem
(informal definition)

Given a collection of annotated data. In this case 5 instances **Katydids** of and five of **Grasshoppers**, decide what type of insect the unlabeled example is.

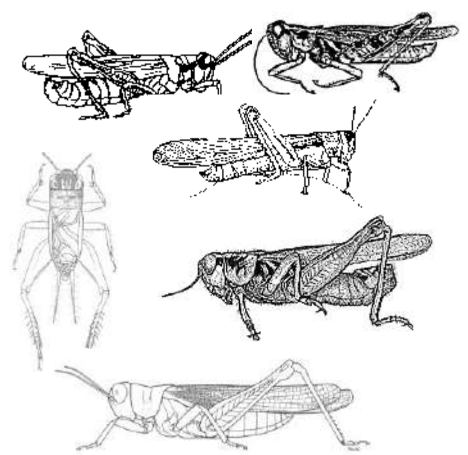


Katydid or **Grasshopper**?

Katydids



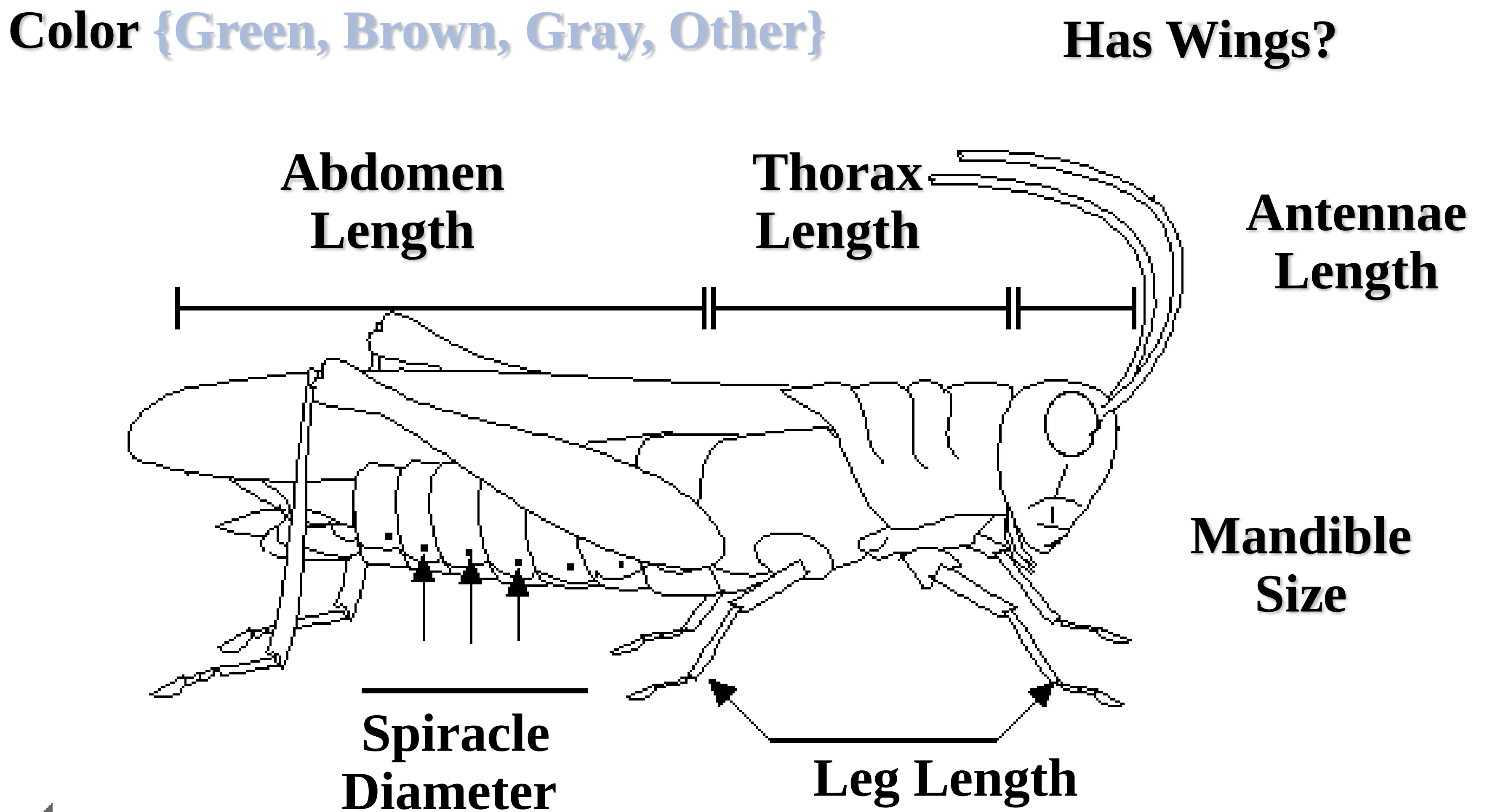
Grasshoppers



Keogh

DLR

For any domain of interest, we can measure *features*



Keogh

We can store features in a database.

The classification problem for images can now be expressed as:

Given a training database (**My_Collection**), predict the **class** label of a previously unseen pixel

Pixel ID	Band 1	Band 2	Pixel Class
1	27	55	**Water**
2	80	91	**Vegetation**
3	9	47	**Water**
4	11	31	**Water**
5	54	85	**Vegetation**
6	29	19	**Water**
7	61	66	**Vegetation**
8	5	10	**Water**
9	83	66	**Vegetation**
10	81	47	**Vegetation**

My_Collection

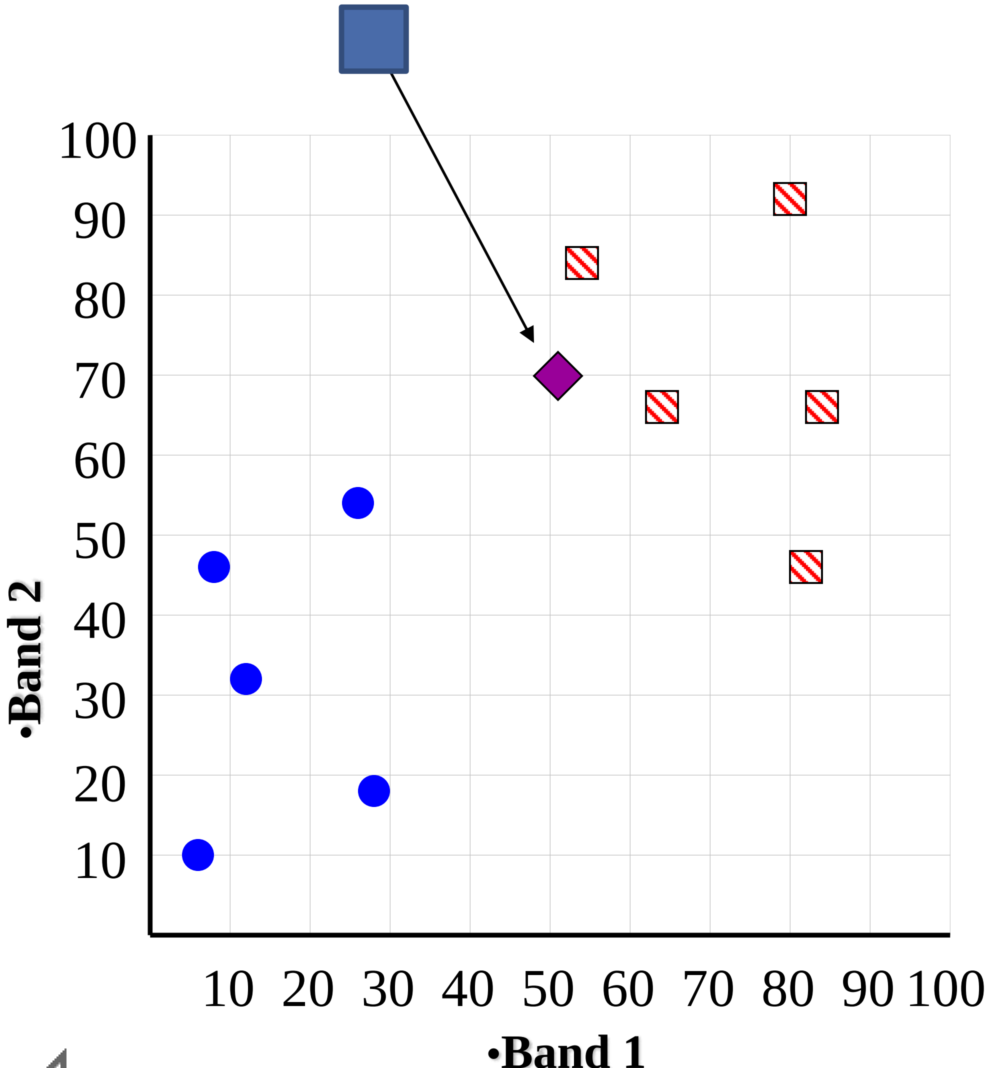
previously unseen pixel =

11	51	70	???????

Keogh

previously unseen pixel =

11	**51**	**70**	???????

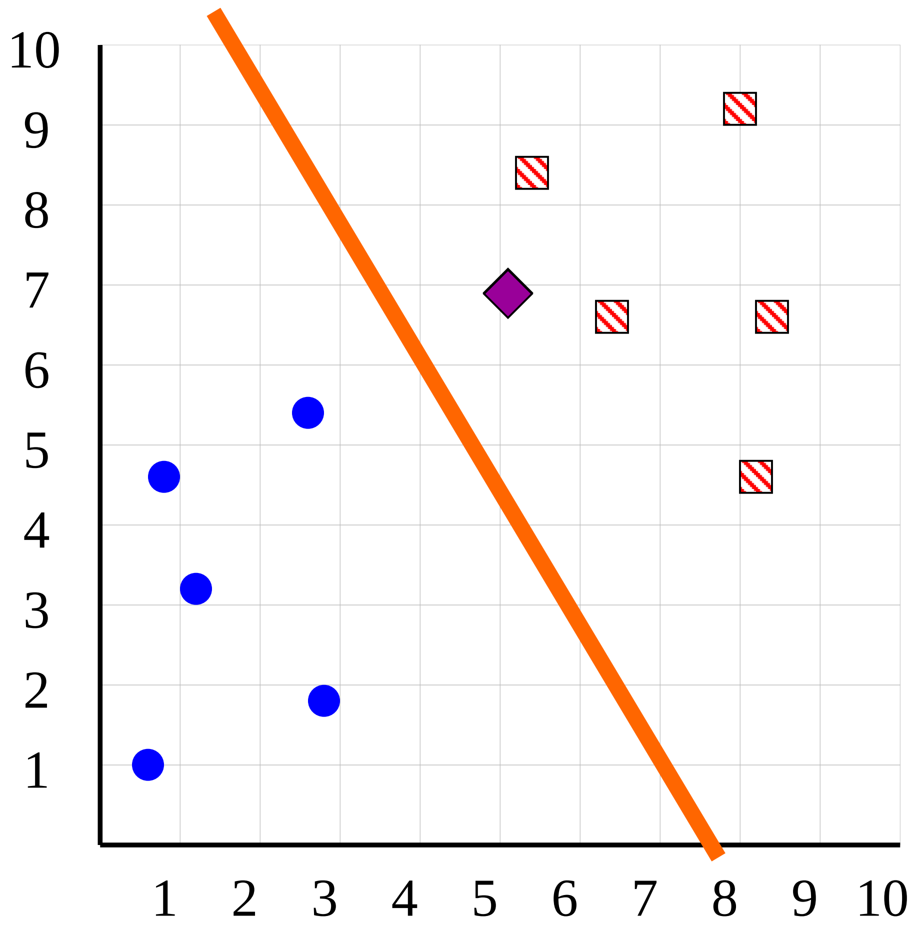


We can "project" the previously unseen pixel into the same space as the database.

We have now abstracted away the details of our particular problem. It will be much easier to talk about points in space.

Vegetation
Water

Simple Linear Classifier



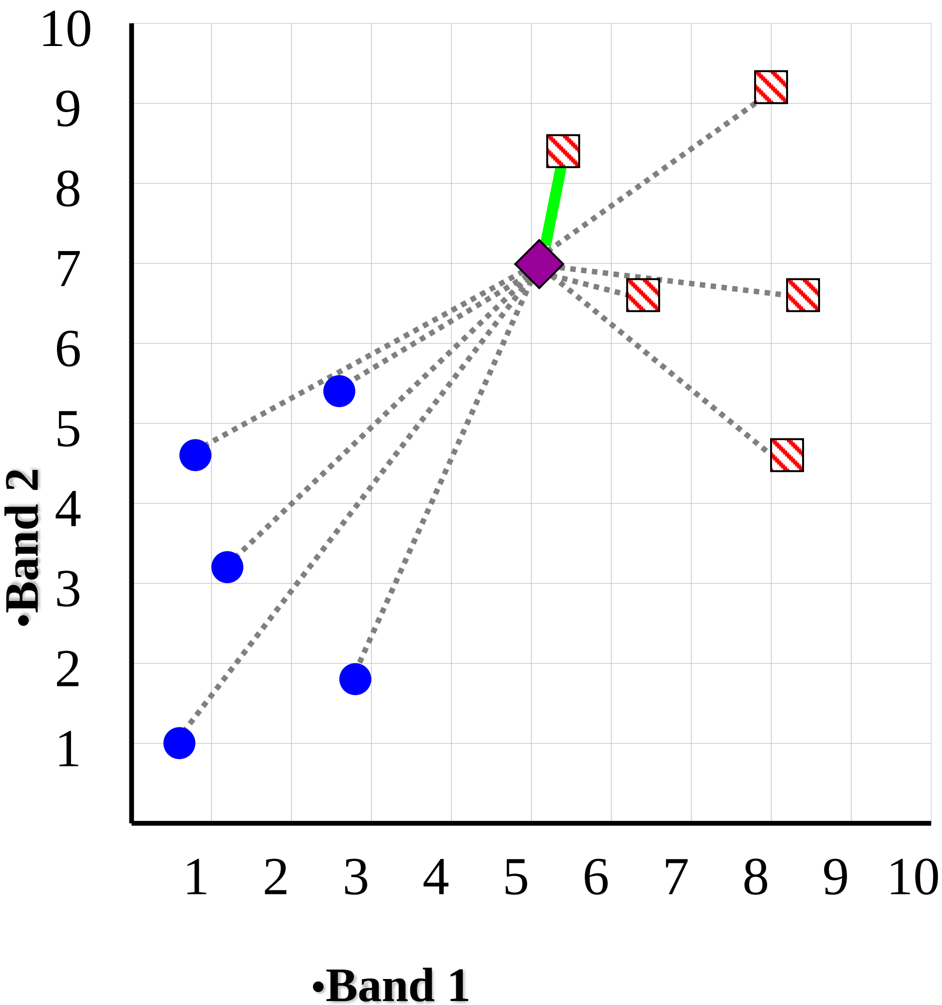
R.A. Fisher
1890-1962

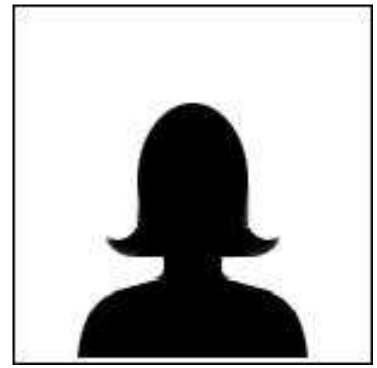
If previously unseen pixel **above** the line
then
class is **Vegetation**
else
class is **Water**

Vegetation
Water

Keogh

Nearest Neighbor Classifier





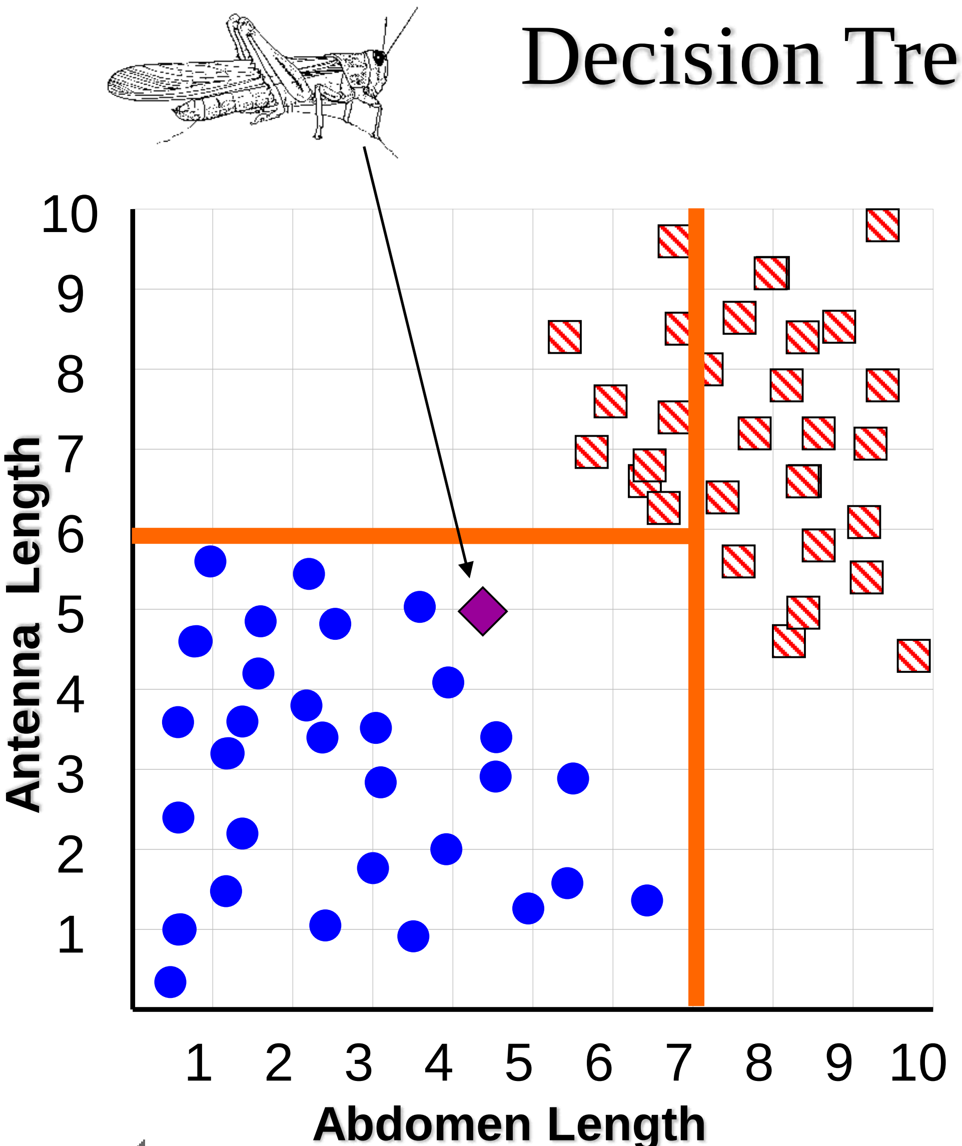
Evelyn Fix
1904-1965

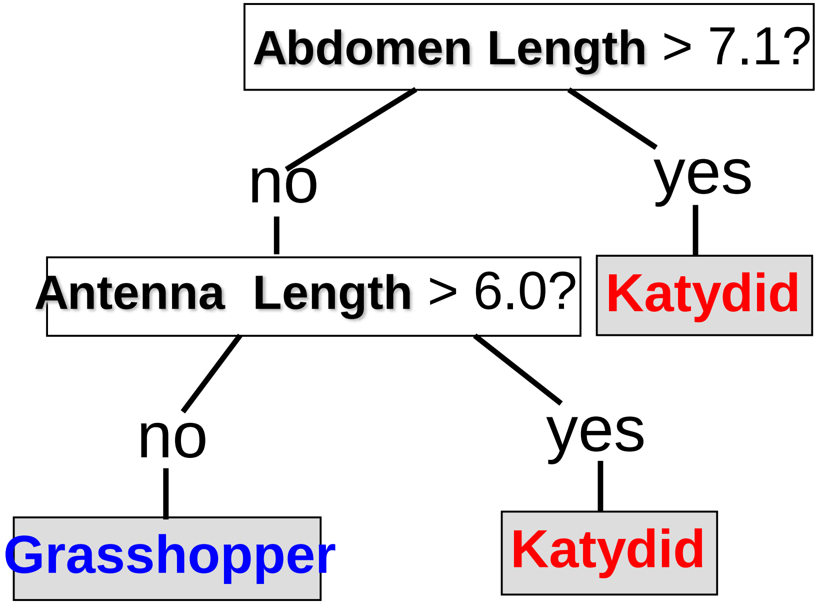
Joe Hodges
1922-2000

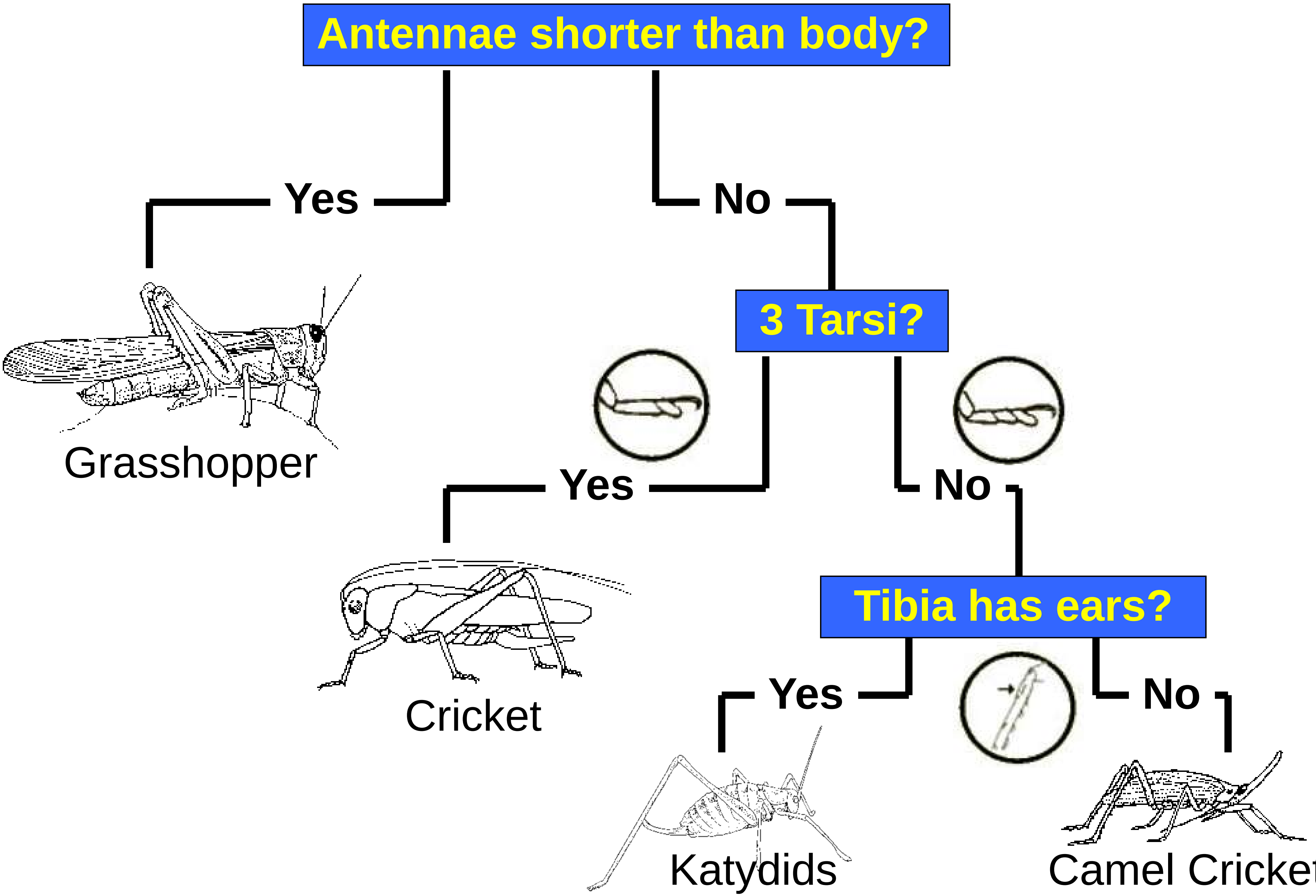
If the **nearest** instance to the previously unseen pixel is Vegetation
  class is Vegetation
else
  class is Water

Vegetation
Water

Decision Tree Classifier






Antennae shorter than body?
Yes
No
Grasshopper
3 Tarsi?
Yes
No
Cricket
Tibia has ears?
Yes
No
Katydids
Camel Cricket

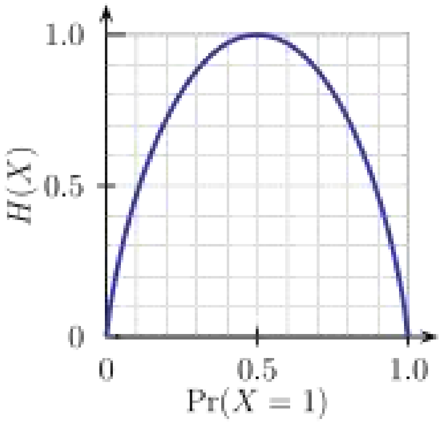
Information Gain as A Splitting Criteria

Select the attribute with the highest information gain (information gain is the expected reduction in entropy).

Assume there are two classes, P and N

- Let the set of examples $\boldsymbol{S}$ contain p elements of class P and n elements of class N
- The amount of information needed to decide if an arbitrary example in S belongs to P or N is defined as

$$E(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$



0 log(0) is defined as 0

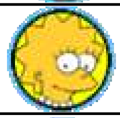
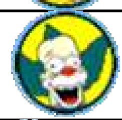
Information Gain in Decision Tree Induction

Assume that using attribute A, a current set will be partitioned into some number of child sets

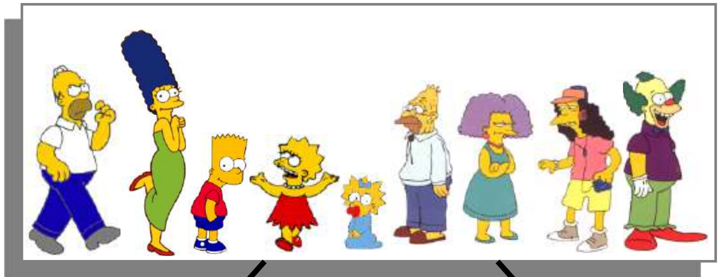
The encoding information that would be gained by branching on *A*

$$Gain(A) = H(Current\ set) - \sum H(all\ child\ sets)$$

Note: entropy is at its minimum if the collection of objects is completely uniform

Person		Hair Length	Weight	Age	Class
	Homer	0”	250	36	M
	Marge	10”	150	34	F
	Bart	2”	90	10	M
	Lisa	6”	78	8	F
	Maggie	4”	20	1	F
	Abe	1”	170	70	M
	Selma	8”	160	41	F
	Otto	10”	180	38	M
	Krusty	6”	200	45	M

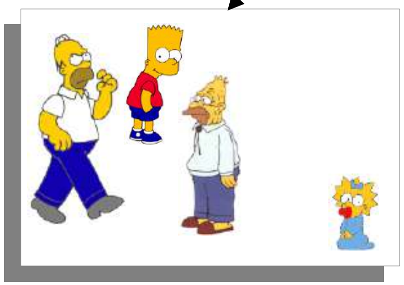
	Comic Guy	8”	290	38	?



$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

Entropy(4**F**,5**M**) = -(4/9)$\log_2$(4/9) - (5/9)$\log_2$(5/9)
= **0.9911**

yes — Hair Length <= 5? — no



Let us try splitting on *Hair length*

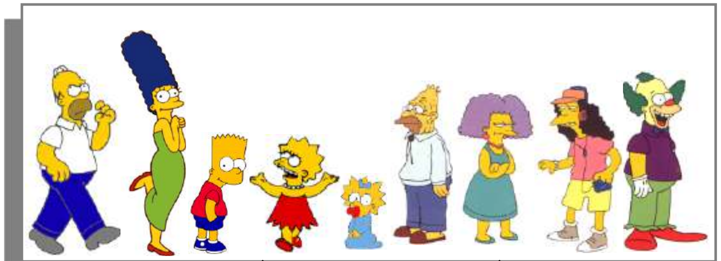
Entropy(1**F**,3**M**) = -(1/4)$\log_2$(1/4) - (3/4)$\log_2$(3/4)
= ***0.8113***

Entropy(3**F**,2**M**) = -(3/5)$\log_2$(3/5) - (2/5)$\log_2$(2/5)
= ***0.9710***

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Gain(Hair Length <= 5) = **0.9911** – (4/9 * **0.8113** + 5/9 * **0.9710**) = **0.091**

Adapted from Eamonn Keogh



$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

Entropy(4**F**,5**M**) = $-(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$
= **0.9911**

yes   no

Weight <= 160?

Let us try splitting on *Weight*

Entropy(4**F**,1**M**) = $-(4/5)\log_2(4/5) - (1/5)\log_2(1/5)$
= ***0.7219***

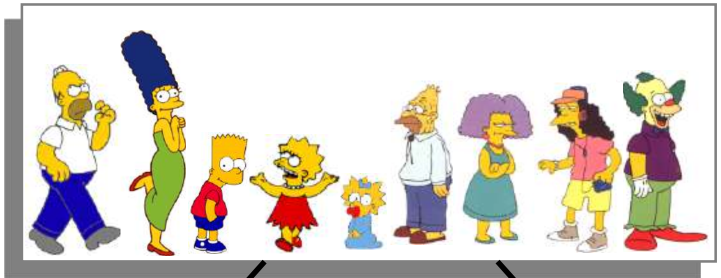
Entropy(0**F**,4**M**) = $-(0/4)\log_2(0/4) - (4/4)\log_2(4/4)$
= ***0***

$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

Gain(Weight <= 160) = **0.9911** – (5/9 * **0.7219** + 4/9 * **0**) = **0.5900**

Adapted from Eamonn Keogh

$$Entropy(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

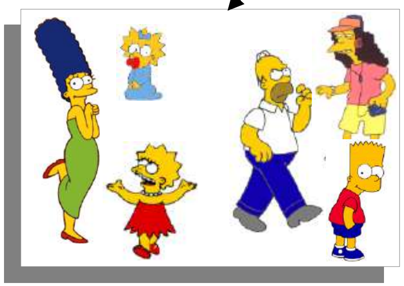


$$Entropy(4F,5M) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9) = 0.9911$$

yes

age <= 40?

no



Let us try splitting on *Age*

$$Entropy(3F,3M) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$$

$$Entropy(1F,2M) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.9183$$

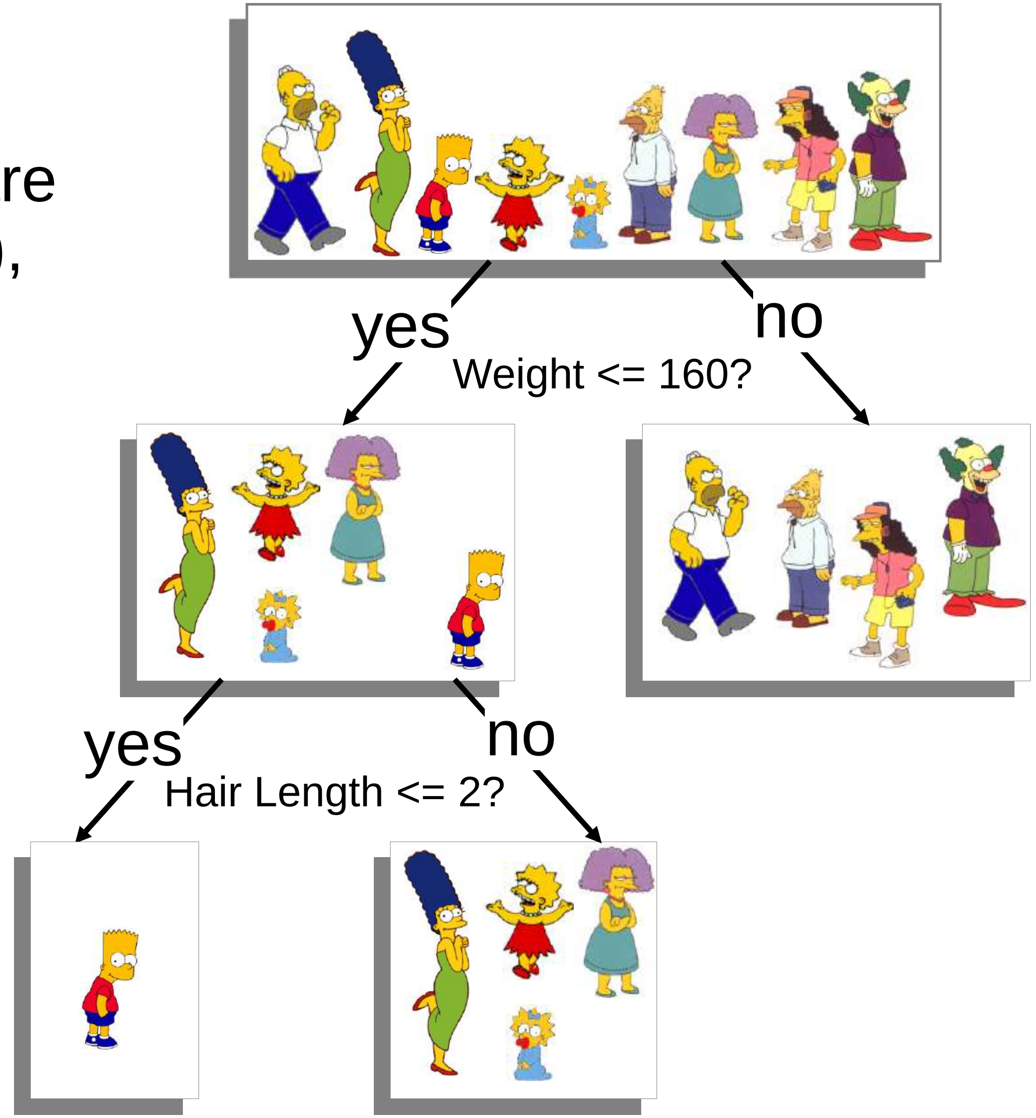
$$Gain(A) = E(Current\ set) - \sum E(all\ child\ sets)$$

$$Gain(Age <= 40) = 0.9911 - (6/9 * 1 + 3/9 * 0.9183) = \mathbf{0.0183}$$

Adapted from Eamonn Keogh

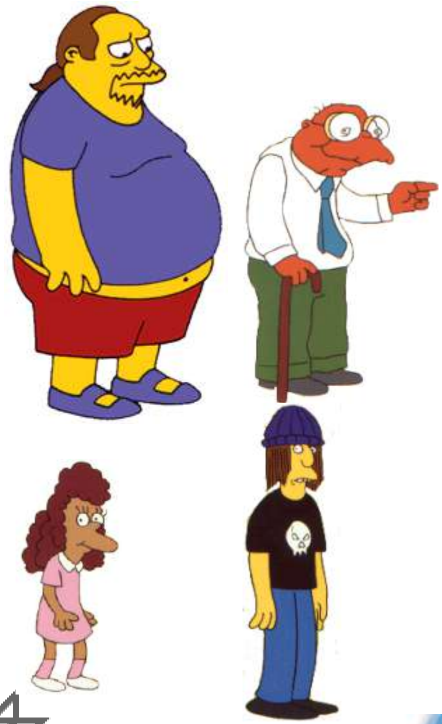
Of the 3 features we had, *Weight* was best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified… So we simply recurse!

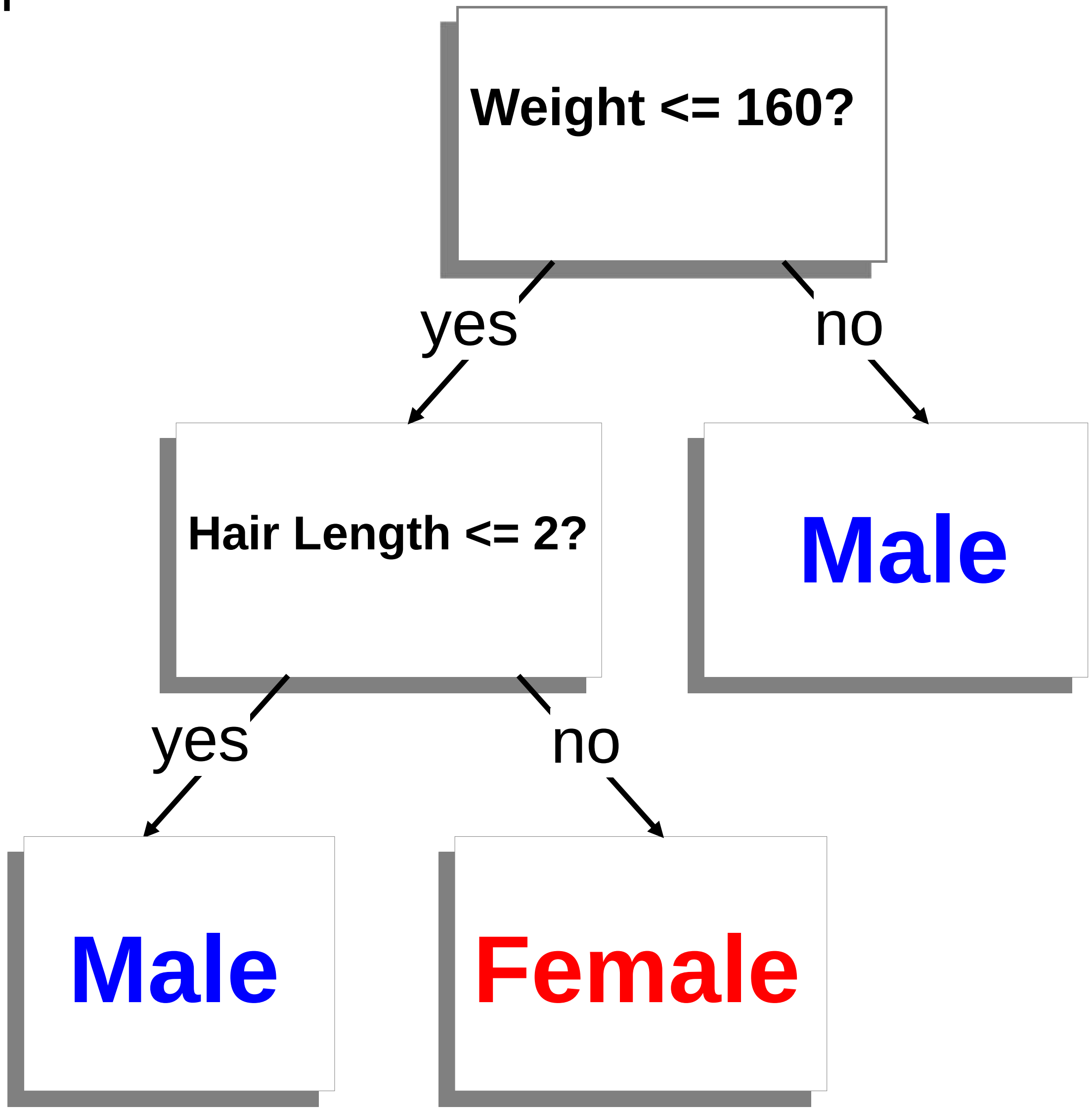
This time we find that we can split on *Hair length,* and we are done!



We need don't need to keep the data around, just the test conditions.

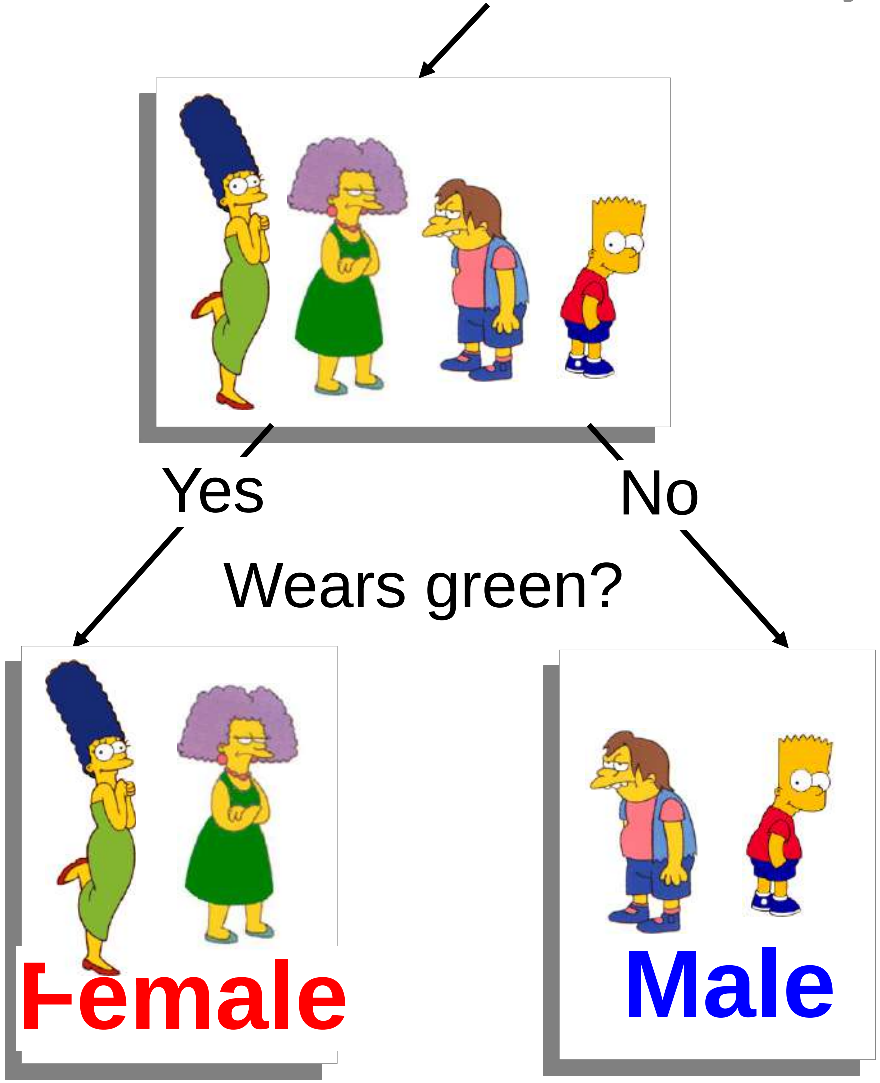
How would these people be classified?





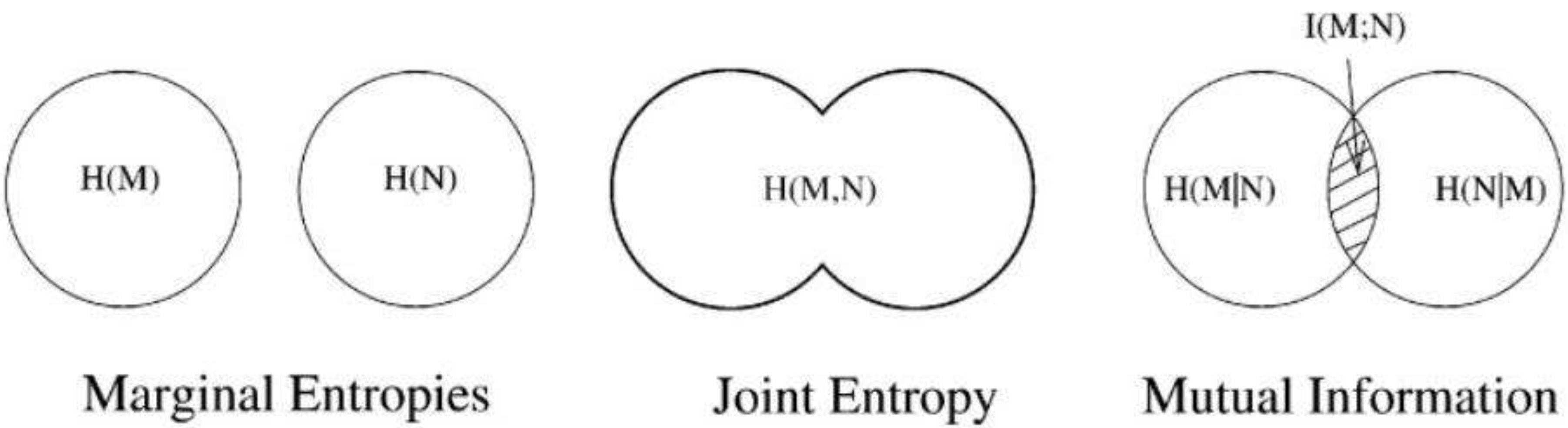
The worked examples we have seen were performed on small datasets. However with small datasets there is a great danger of overfitting the data…

When you have few datapoints, there are many possible splitting rules that perfectly classify the data, but will not generalize to future datasets.



For example, the rule "Wears green?" perfectly classifies the data, so does "Mothers name is Jacqueline?", so does "Has blue shoes"…

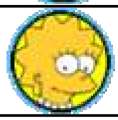
Mutual Information (in terms of Entropy)



Think of this quantity as directly related to the information gain we saw in the previous example!

$$I(M,N) = H(M) + H(N) - H(M,N)$$

Person		Hair Length	Weight	Age	Class
	Homer	0”	250	36	**M**
	Marge	10”	150	34	**F**
	Bart	2”	90	10	**M**
	Lisa	6”	78	8	**F**
	Maggie	4”	20	1	**F**
	Abe	1”	170	70	**M**
	Selma	8”	160	41	**F**
	Otto	10”	180	38	**M**
	Krusty	6”	200	45	**M**
	Comic Guy	8”	290	38	**?**

Pixel	Value in band 1	Value in band 2	Value in band 3	Class
 X:100, Y:120	10	30	50	**Broccoli**
 X:50, Y:100	25	130	50	**Fallow**
 X:16, Y:12	13	12	48	**Grapes**
 X:200, Y:420	5	70	49	**Corn**

Which band is better to separate these classes? Which one will give me the maximum information gain? And which one would only make things more difficult?

Mutual Information

(can be expressed in terms of probability)

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right),$$

Small Test

– Suppose we have the following random variables

 – x = "is the temperature below 0 degrees?" → (0 = no, 1 = yes)
 – y = "do I have ice or water?" → (0 = ice, 1 = water)
 – z = "is it snowing outside?"→ (0 = no, 1 = yes)
 – w = „Are the Simpsons today on Pro7?"→ (0 = no, 1 = yes)

– How do you expect the mutual information to be between:

 – X and Y
 – X and Z
 – Y and Z
 – X and W

Mutual Information for HS data analysis

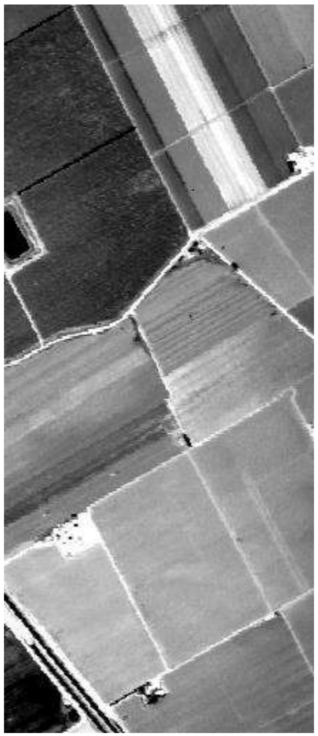
– What we REALLY want is how to select bands which are good to classify a specific dataset.

– The MI is great at finding correspondences between variables, even if their values are very different!

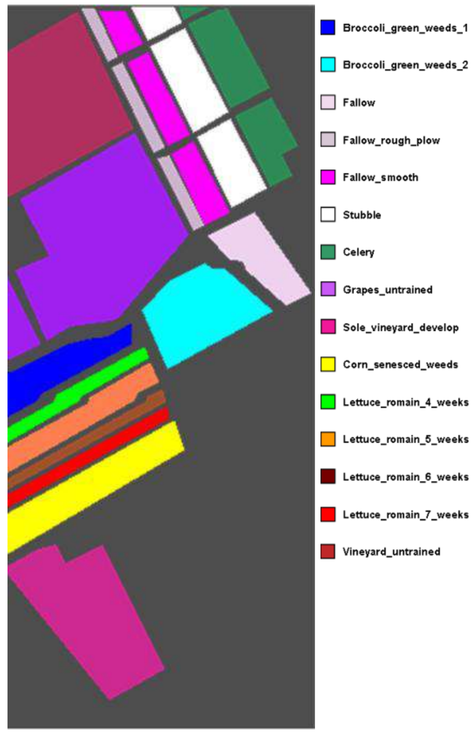
  – For example it has been used in our department to improve coregistration between radar and optical data, which are completely different!!

The mutual information between any two bands in the Salinas dataset and the ground truth are based on these joint distributions…

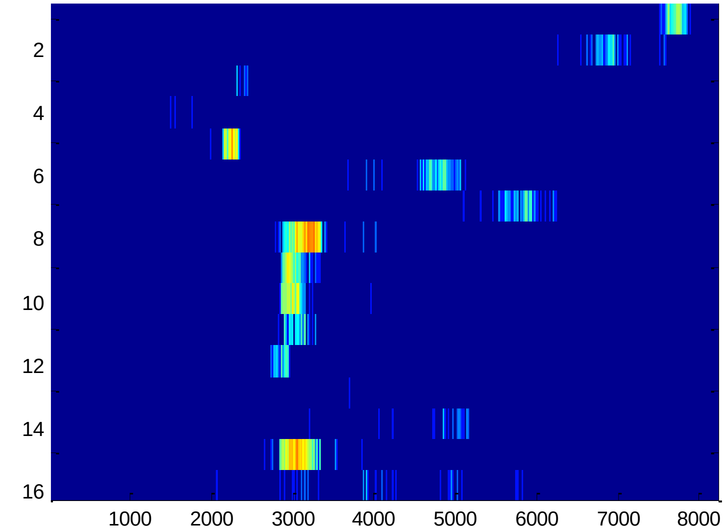
Banda 1



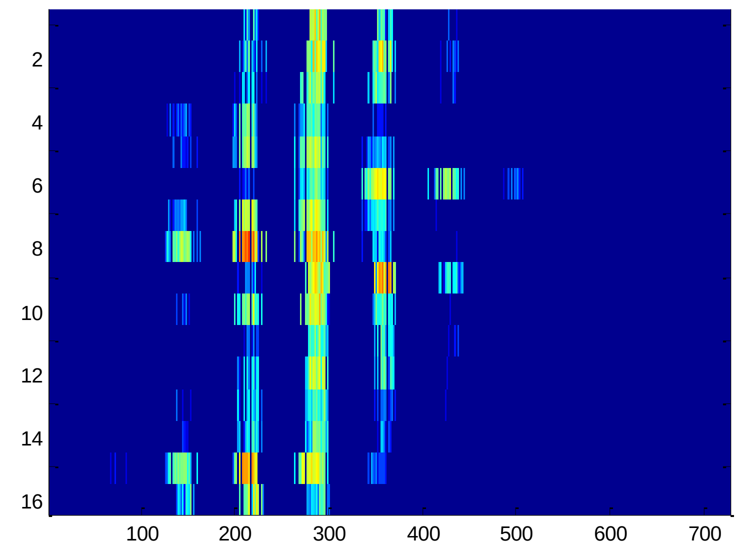
Banda 42



Ground Truth



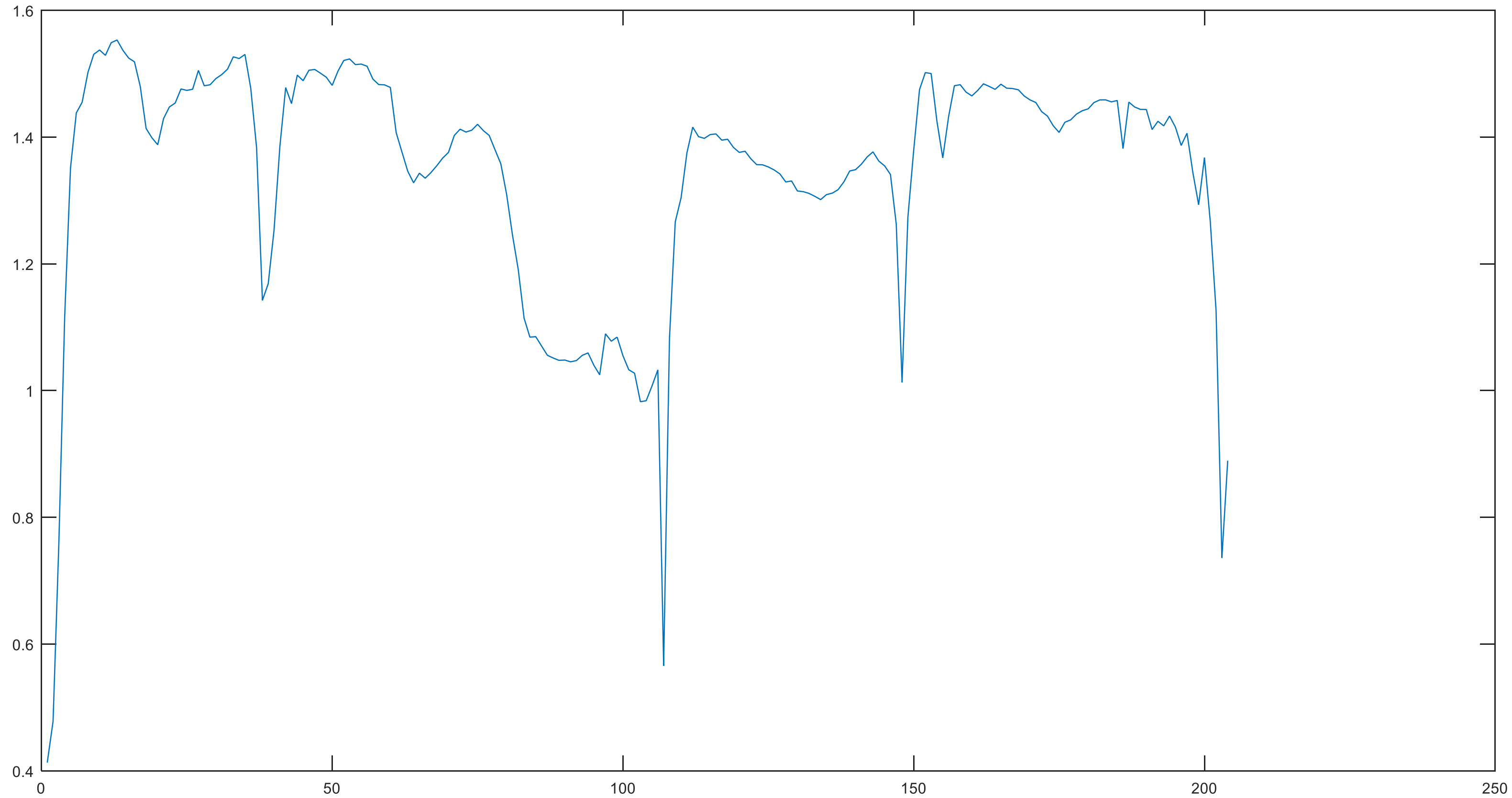
Mutual info band 42 / ground truth



Mutual info band 1 / ground truth

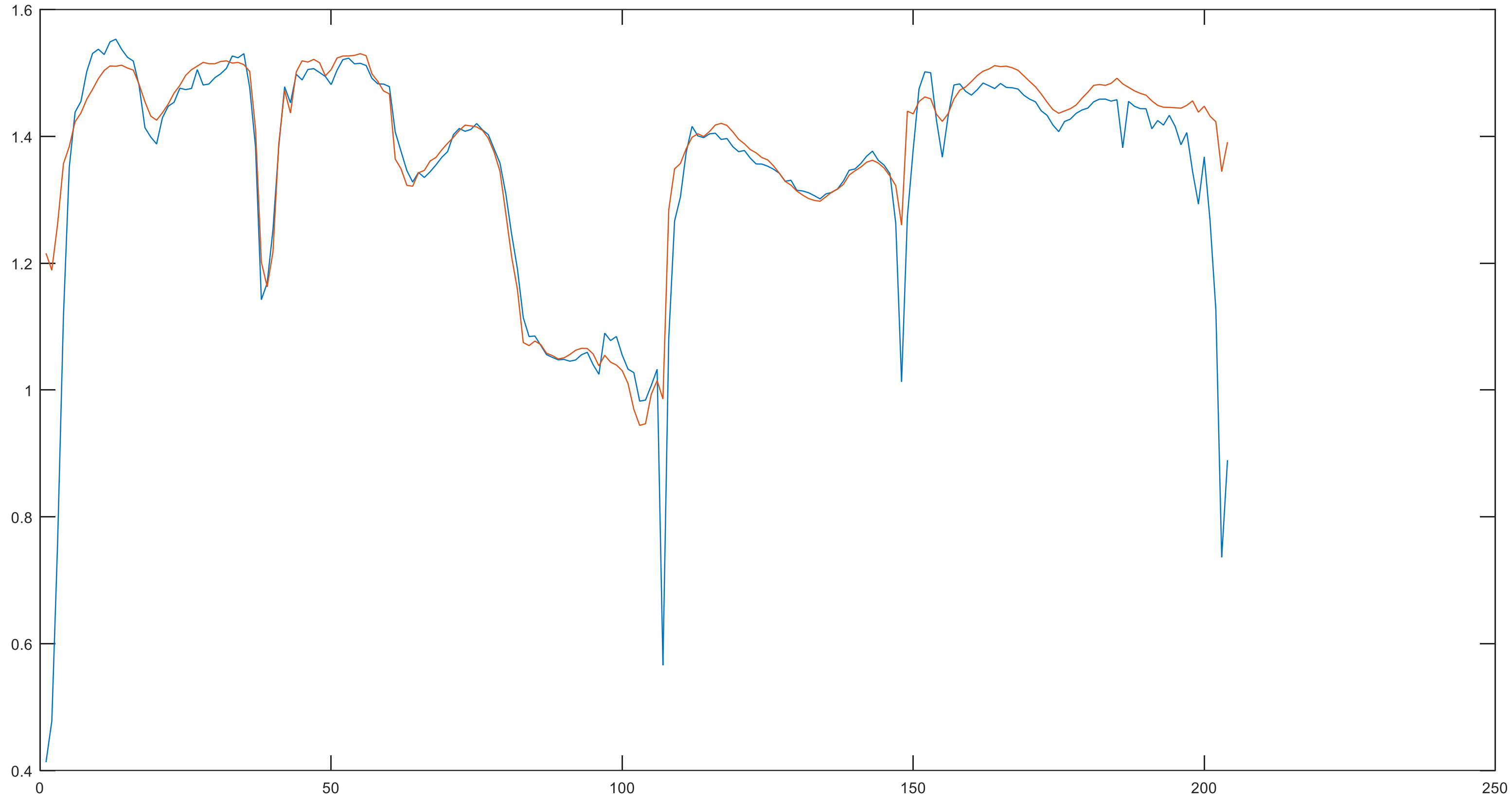
DLR

Mutual information Salinas / ground truth



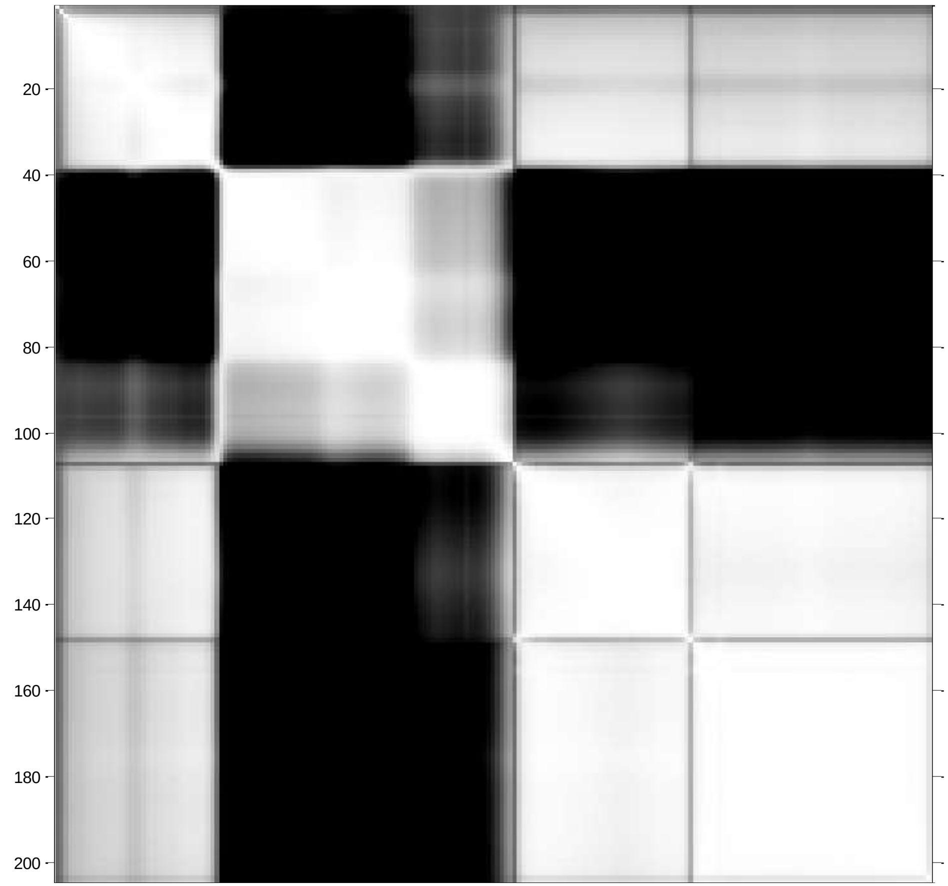
Mutual information Salinas / ground truth

Before (blue) and after (red) noise removal



DLR

Reminder: Correlation between bands



Intraband correlation matrix for the Salinas dataset

Summary

– How to select our 10 bands now?

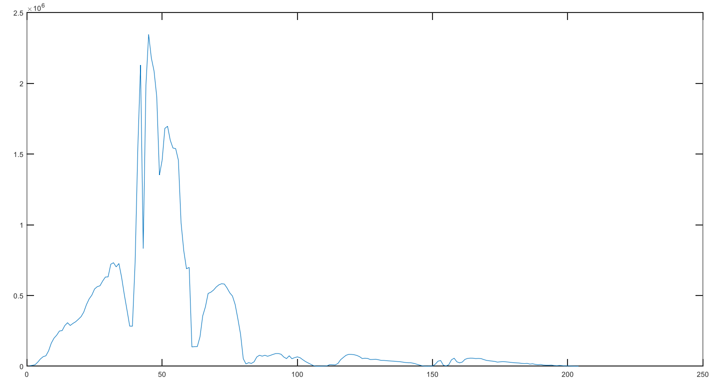
– Usye the information derived so far and the intraband correlation to select them

– For example:

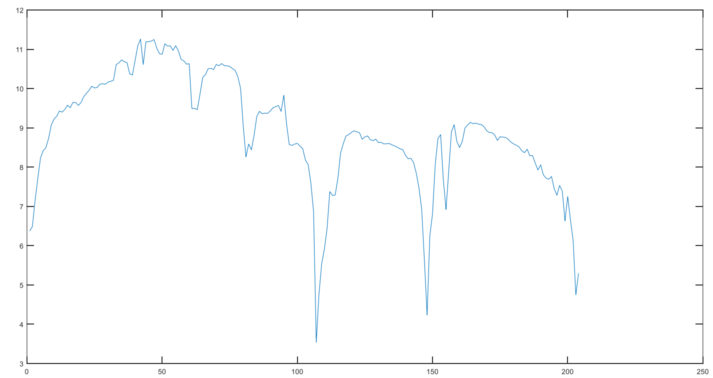
  – Cluster bands

  – Select the best band in each cluster

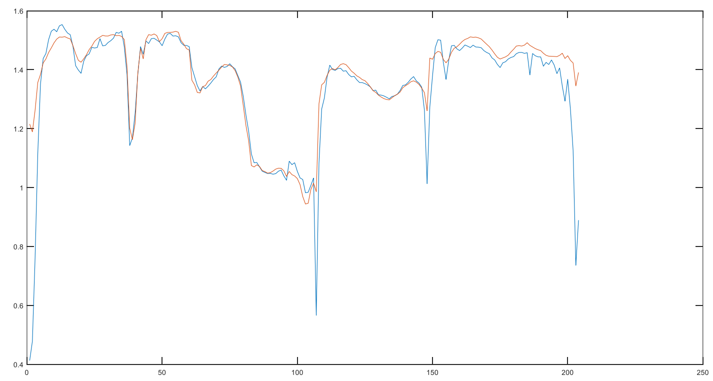
Variance



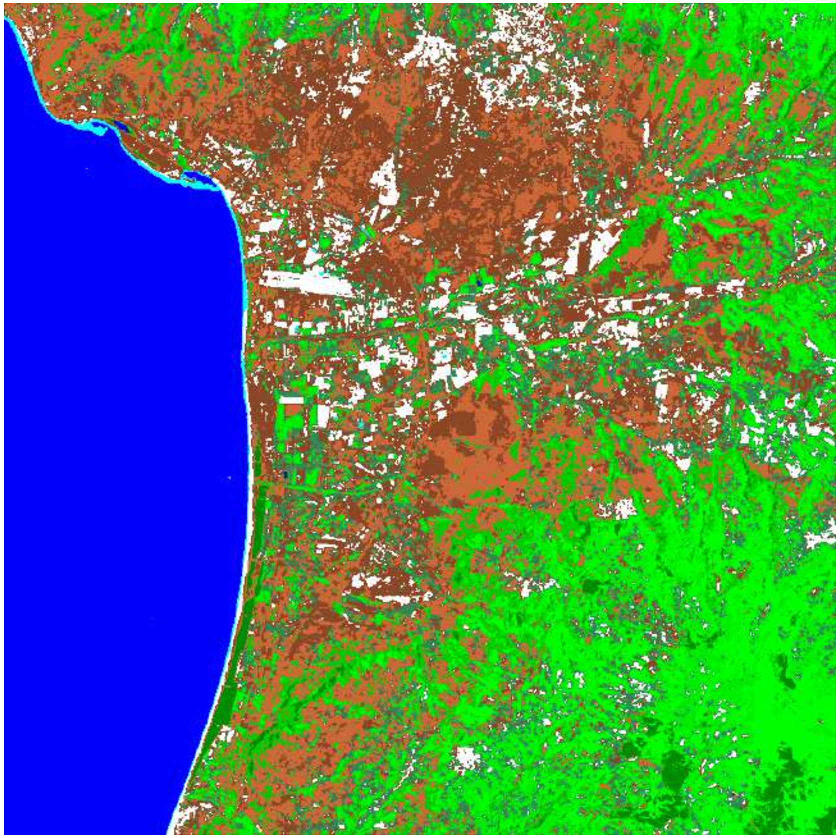
Entropy



Mutual Information



K-means Clustering (k = 7)



With pixels it is clear..

– How to do it with spectral bands?

1. We convert the bands (each originally in 2D) in pixel vectors (1D)

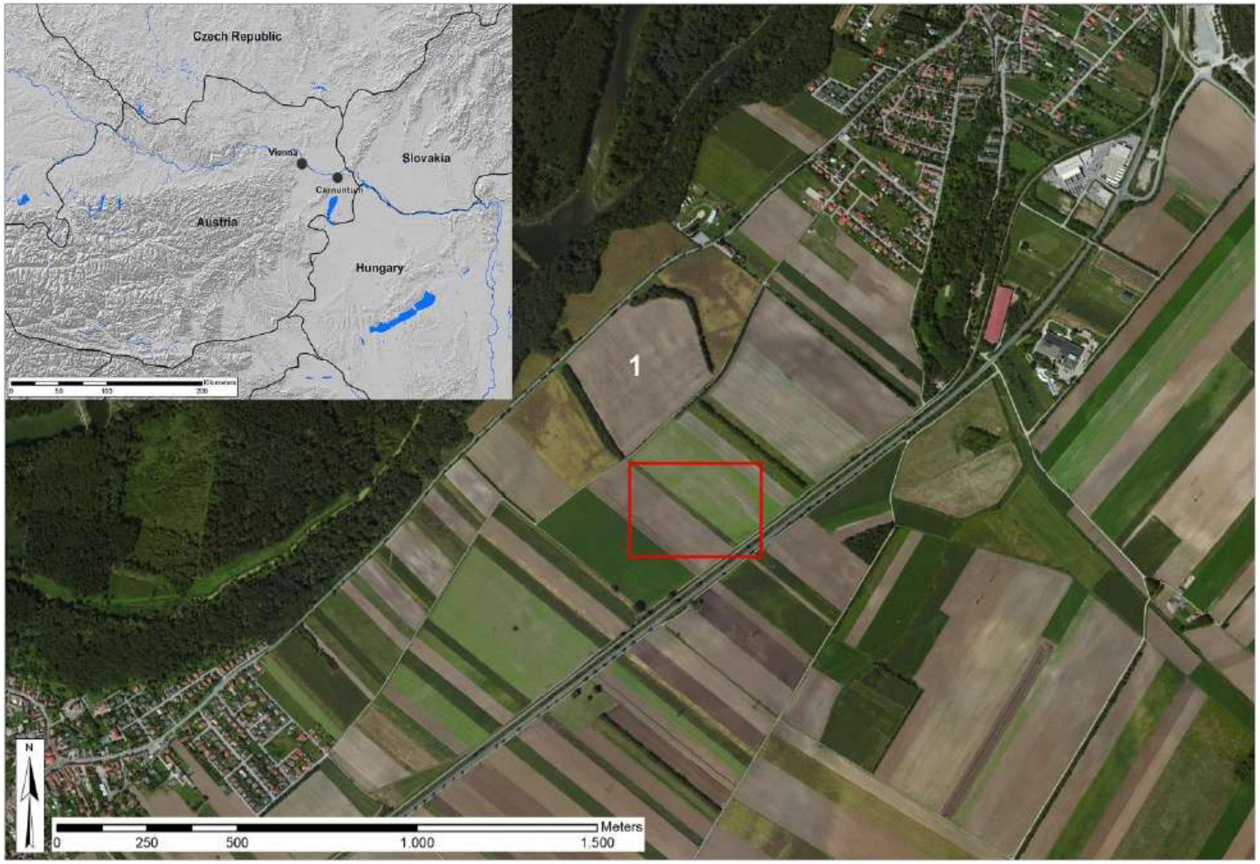
2. Our space in kmeans now has the same dimensionality as number of pixels in the image

3. EXAMPLE:

   1. In which space are we doing the clustering with an image of 100 x 100 pixels and 200 bands?

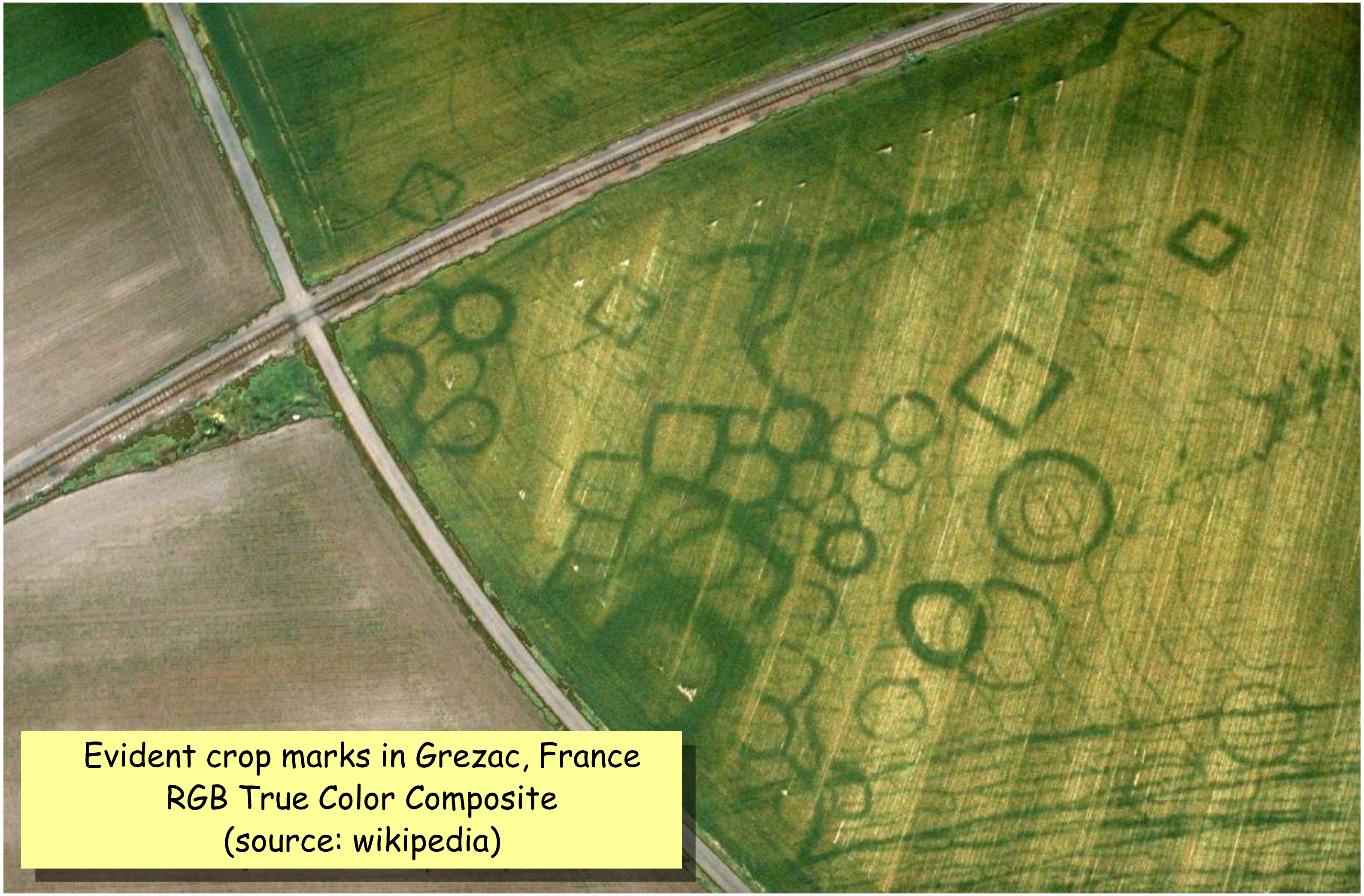
An interesting application

- Dataset: Carnuntum
  - Capital of the former Roman province *Pannonia superior*
  - *Centuries IV BC – I AD*
- Airborne HS campaign
- AisaEAGLE
  - 65 bands
  - 400-1000 nm
  - 0.4 m GSD
  - Courtesy of prof. Michael Donus



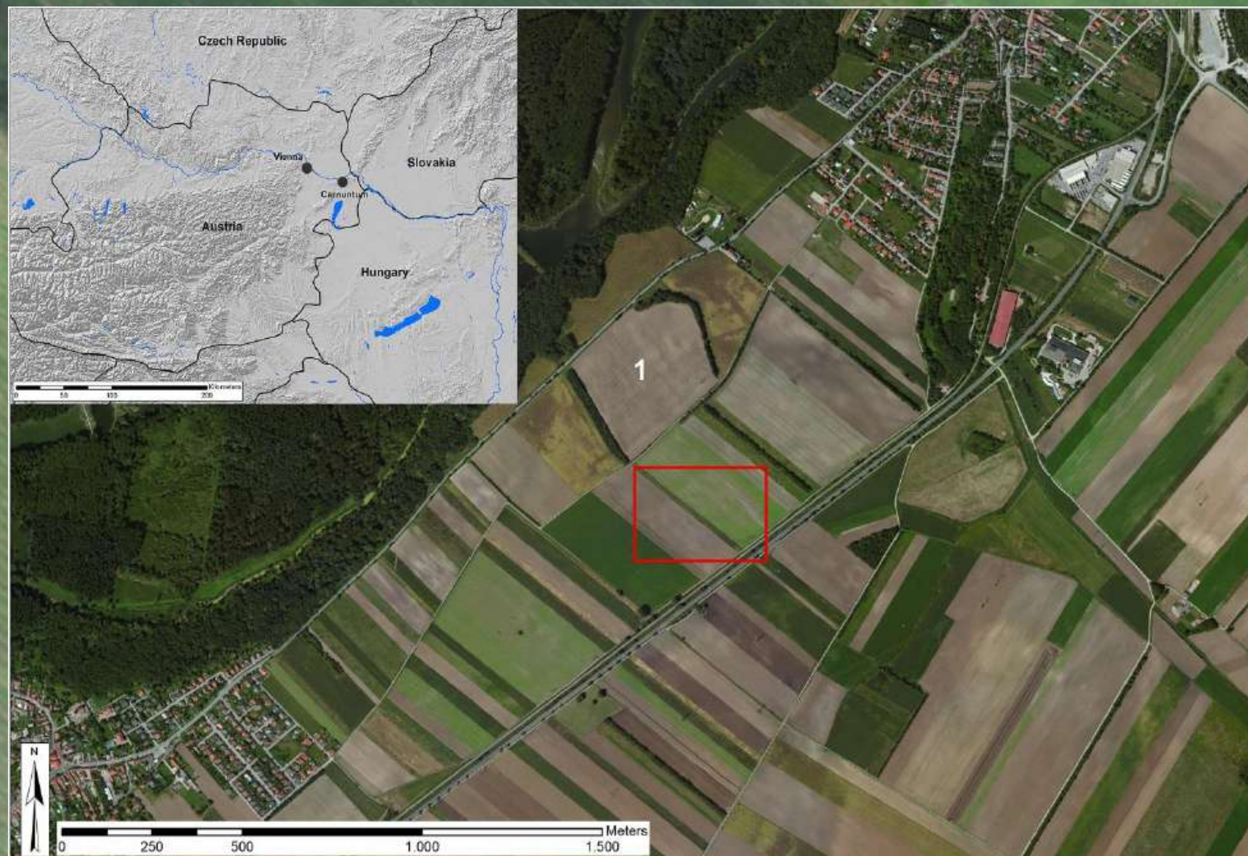
Michael Doneus et al., „New ways to extract archaeological information from hyperspectral pixels“, Journal of Archaeological Science, Volume 52, December 2014

How to highlight crop marks?

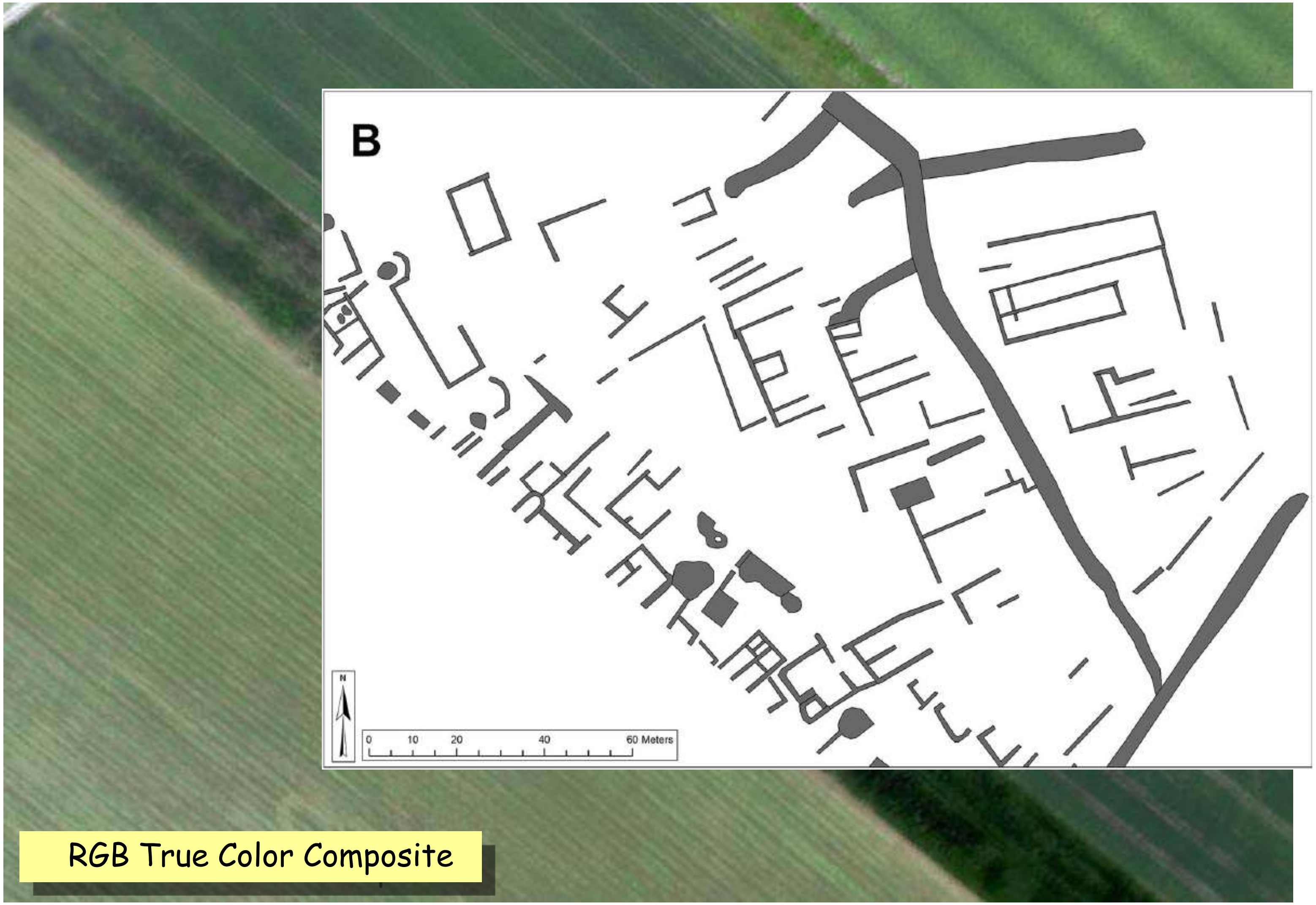


Evident crop marks in Grezac, France
RGB True Color Composite
(source: wikipedia)

Not always that easy…

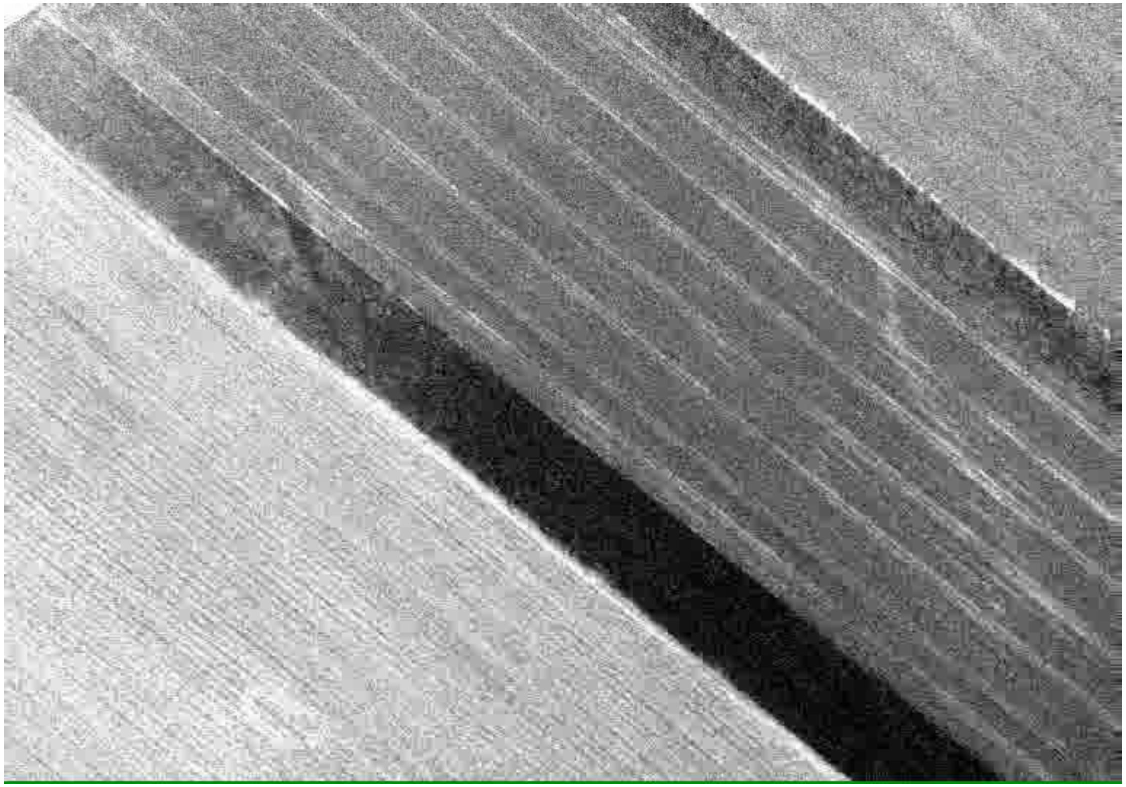


Not always that easy…



Which band is better?

– Let‘s have a look at all available bands…



Which band is better?

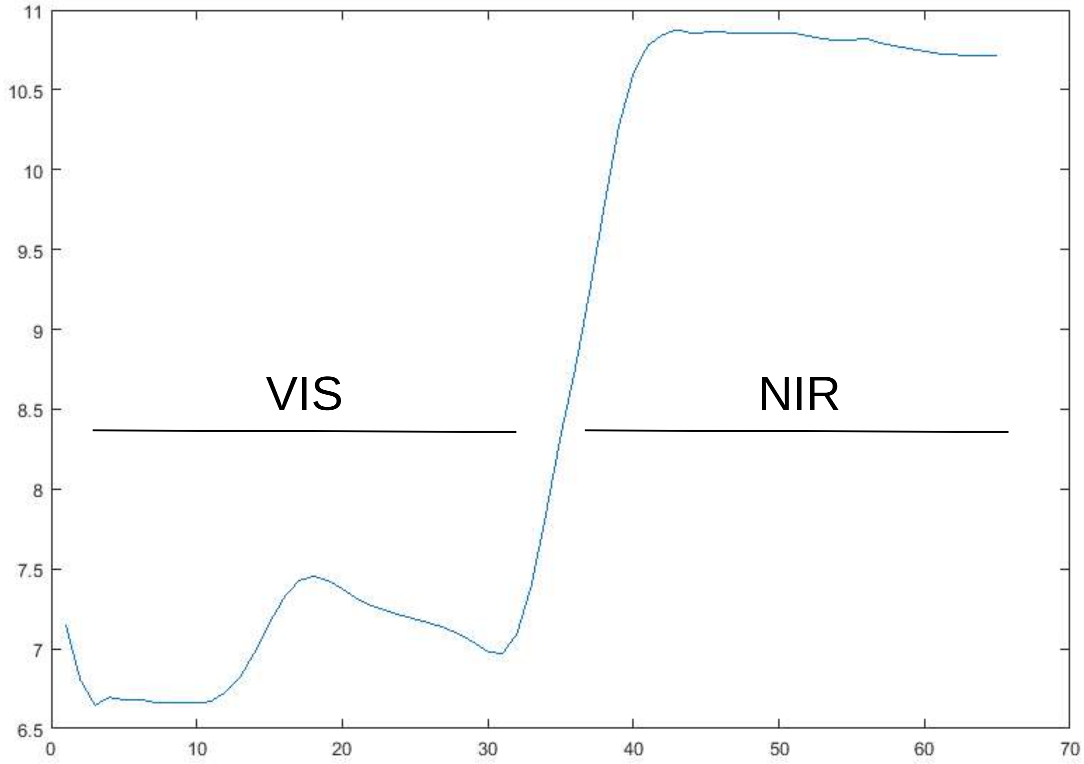
- The transition between red and NIR and the whole NIR spectral range looks good..
- If we find which band is best, we can apply it to other images to look for crop marks
- How to quantify the performance of each band?

First Step: Entropy

- We can compute it directly for each band
  - We get a score for each band
- How much „information“ do we have in each portion of the spectrum?
- It works better if we select only the area of interest
  - We are not interested in variations throughout the whole image

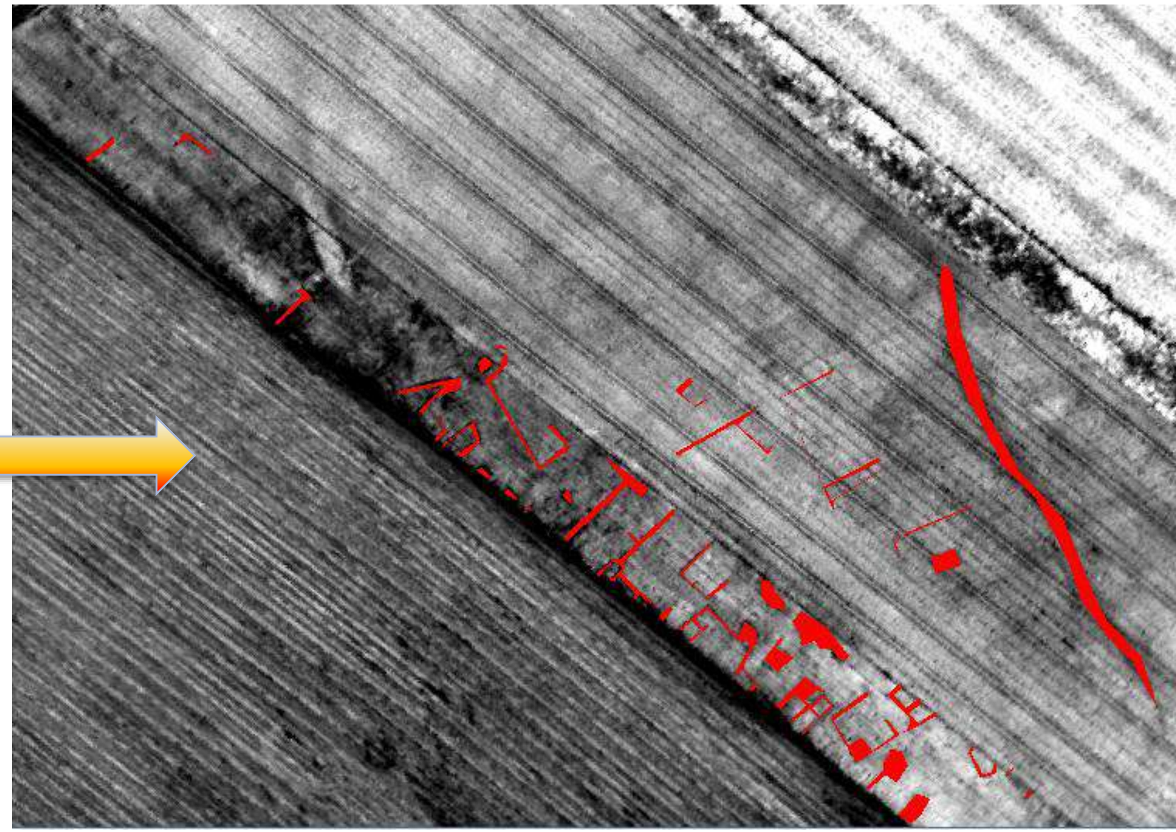
First Step: Entropy

Entropies for each band



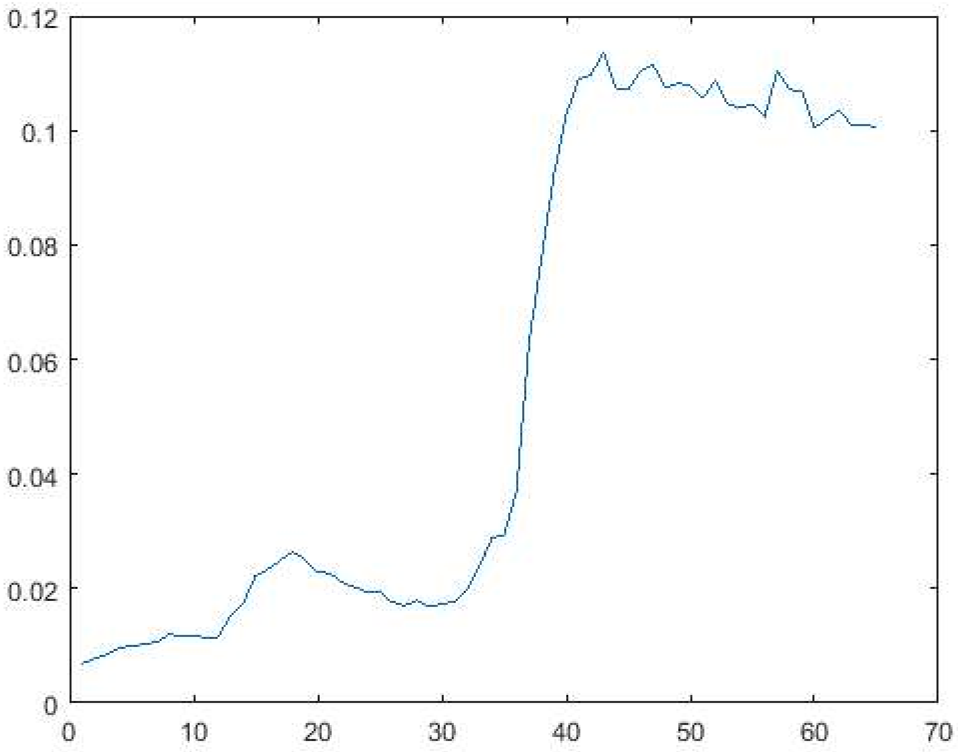
Let‘s take a step forward and compute Mutual Information

– Let‘s derive a reference image (manually)

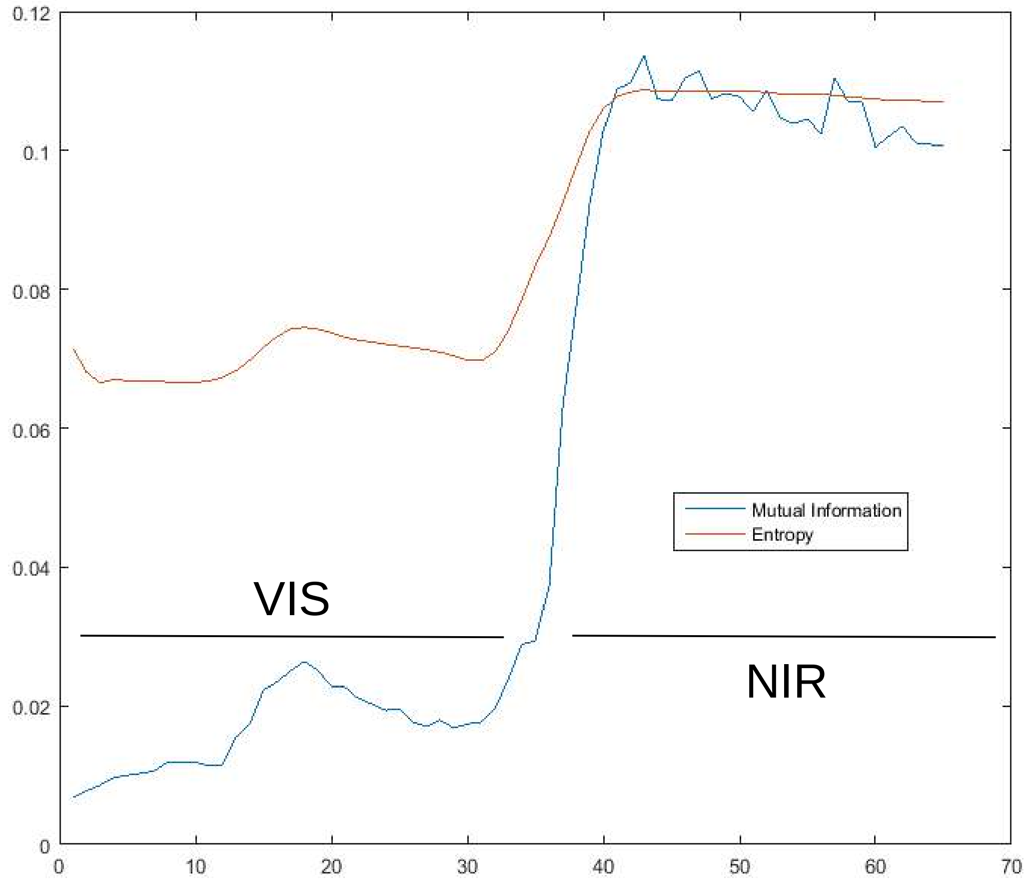


Mutual Information

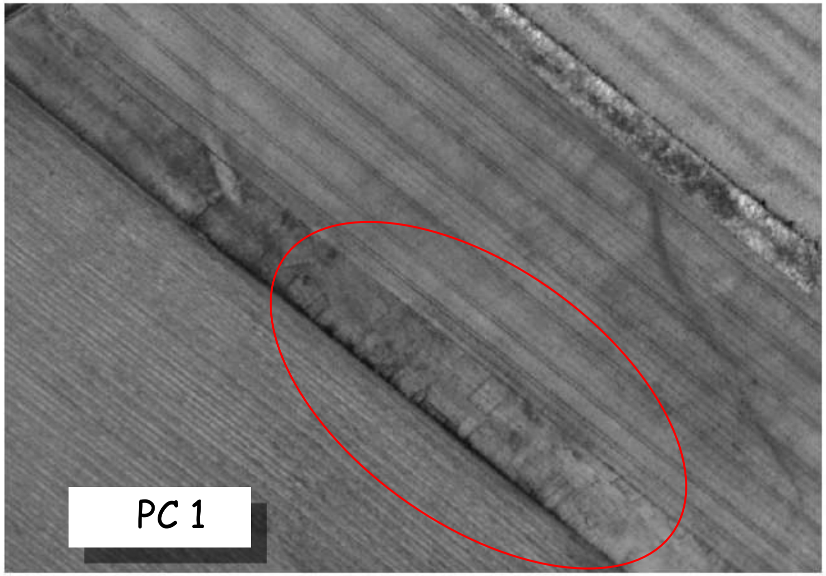
– Between each band and the reference data in the area of interest

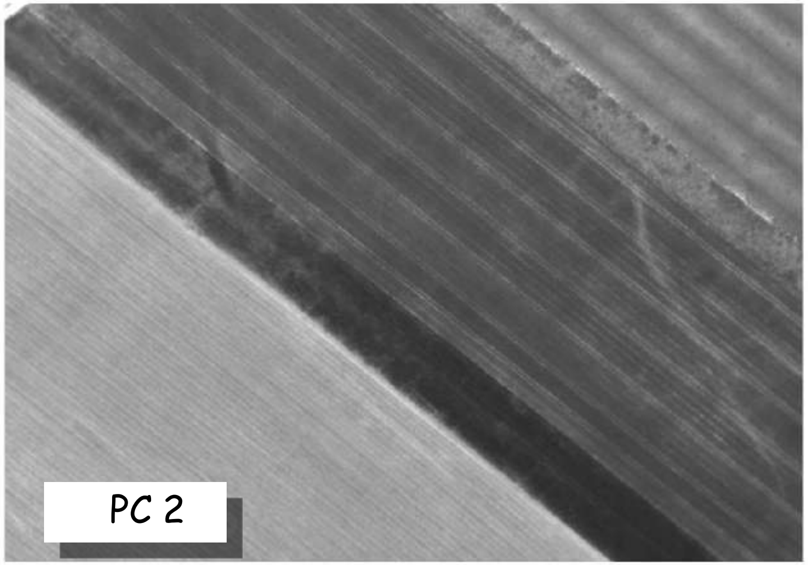


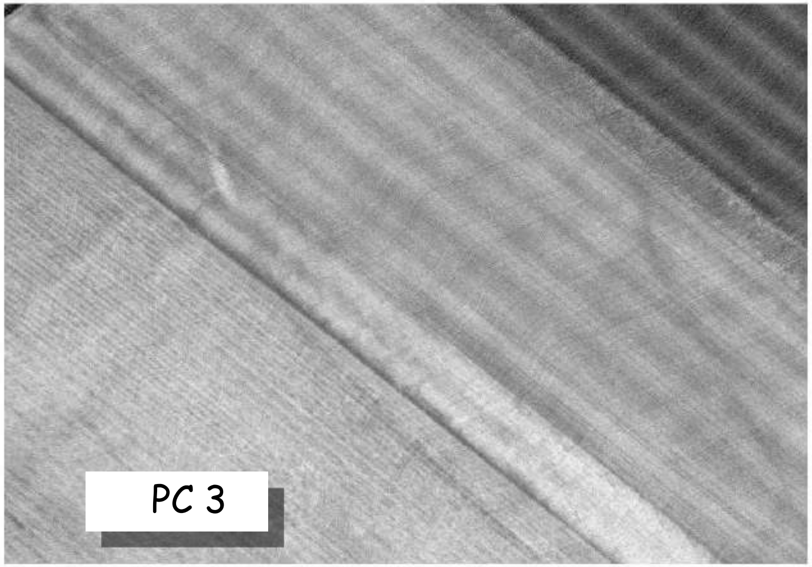
What does the mutual information tell us respect to entropy?

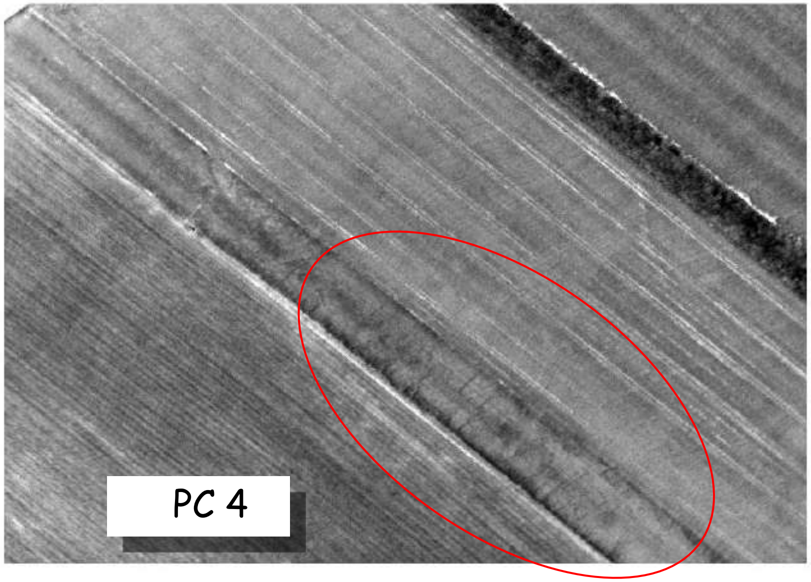


Next step: let's analyse the Principal Components

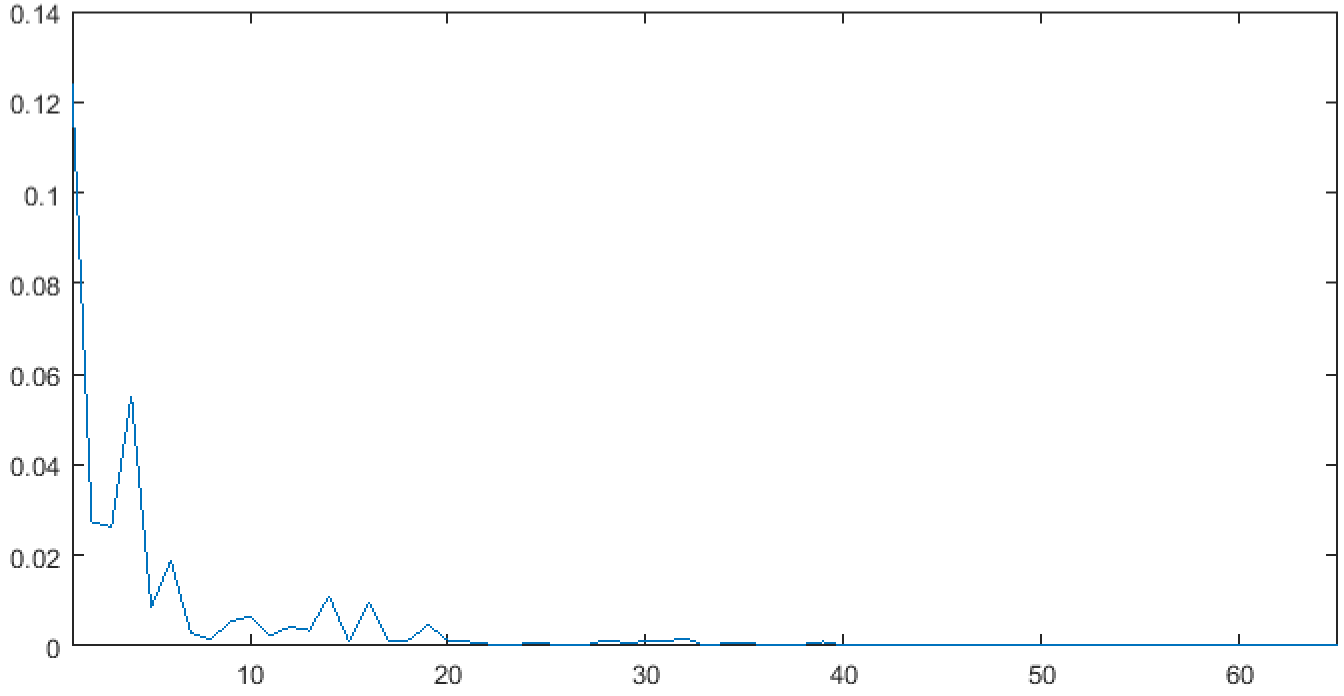






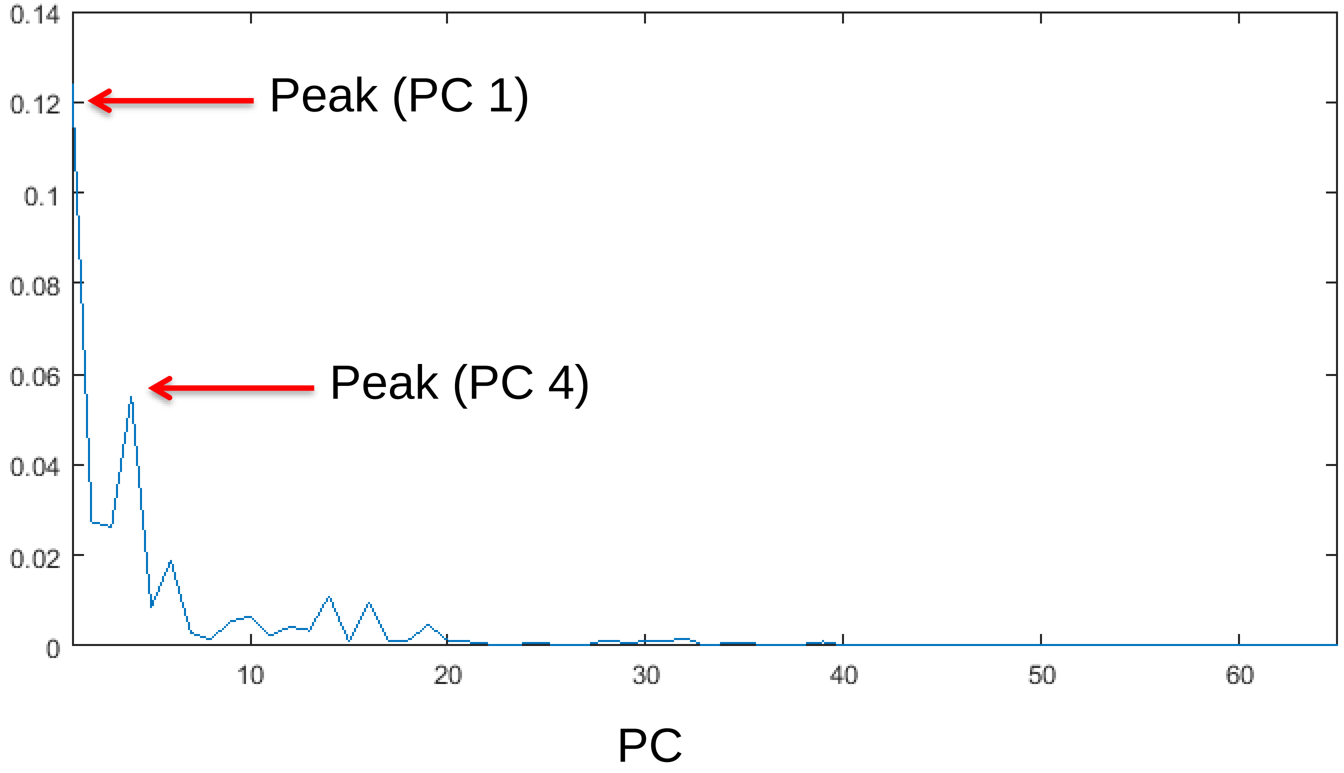


Principal Components: Mutual Information



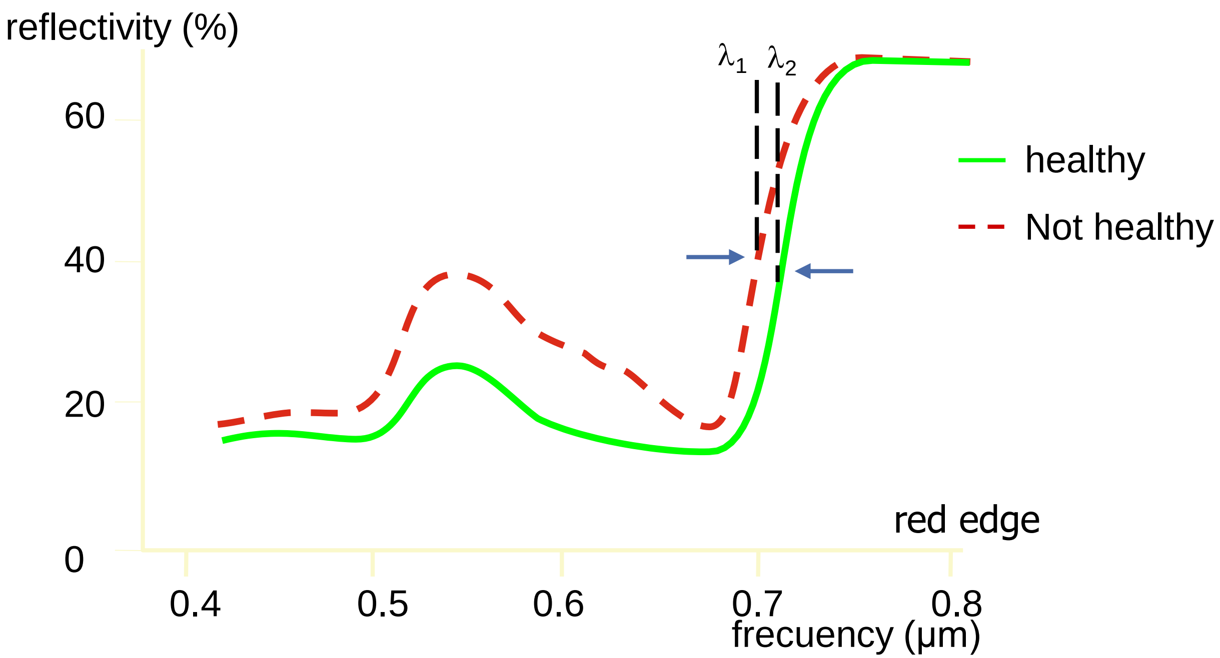
DLR

Principal Components: Mutual Information



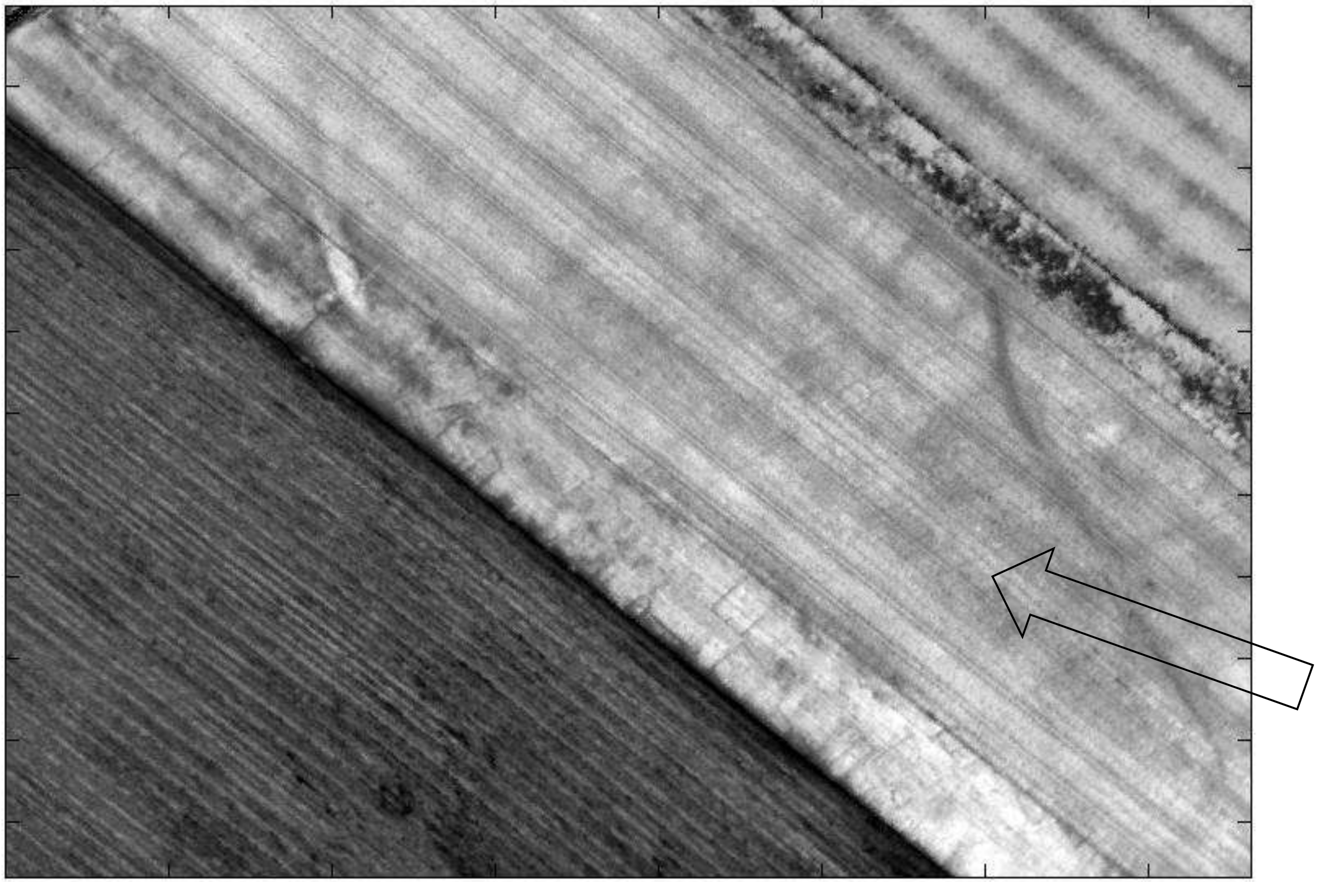
Hyper- vs. Multispectral: Vegetation Analysis

Near Infrared: the Red Edge

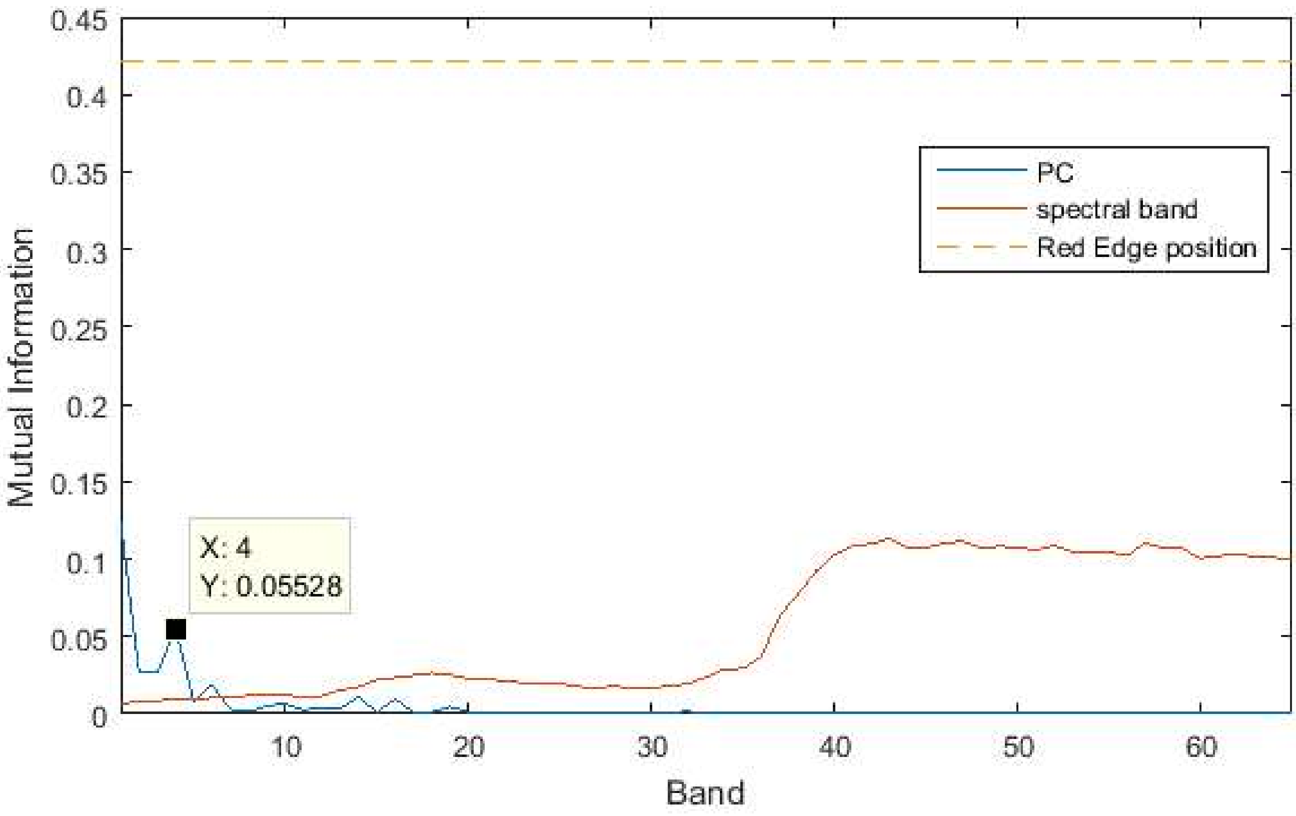


- Transition between absorption into red and high reflectance in the near infrared portions of the spectrum
- The **red edge** is the spectral range in which this change is observable (flexion point in the curve)
- It depends on the amount on clorophyll in the plant and nitrogen in the soil
- A displacemente to the left of the red edge characterizes ill vegetation
  - Scarce clorophyll in leaves
  - "Breathing" problems of the plant

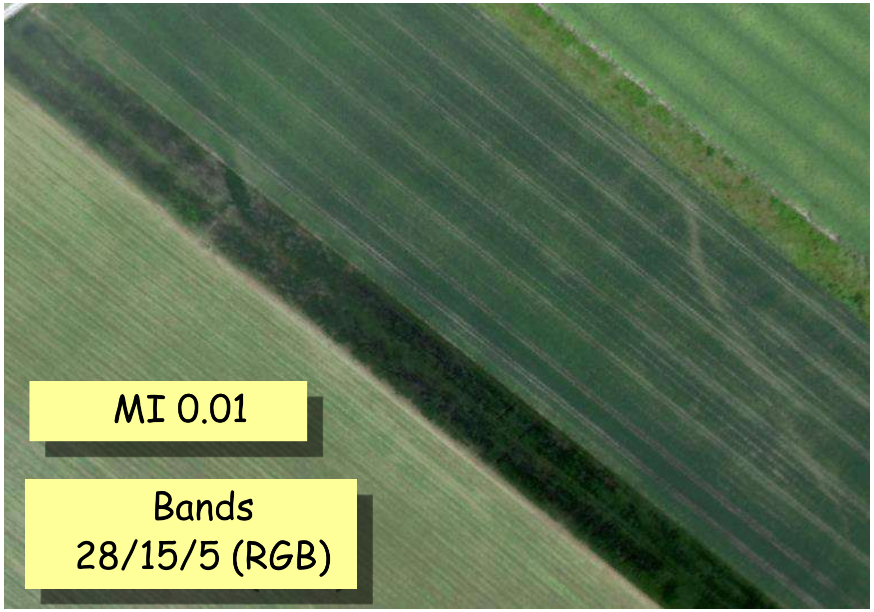
Back to our example: Red Edge image

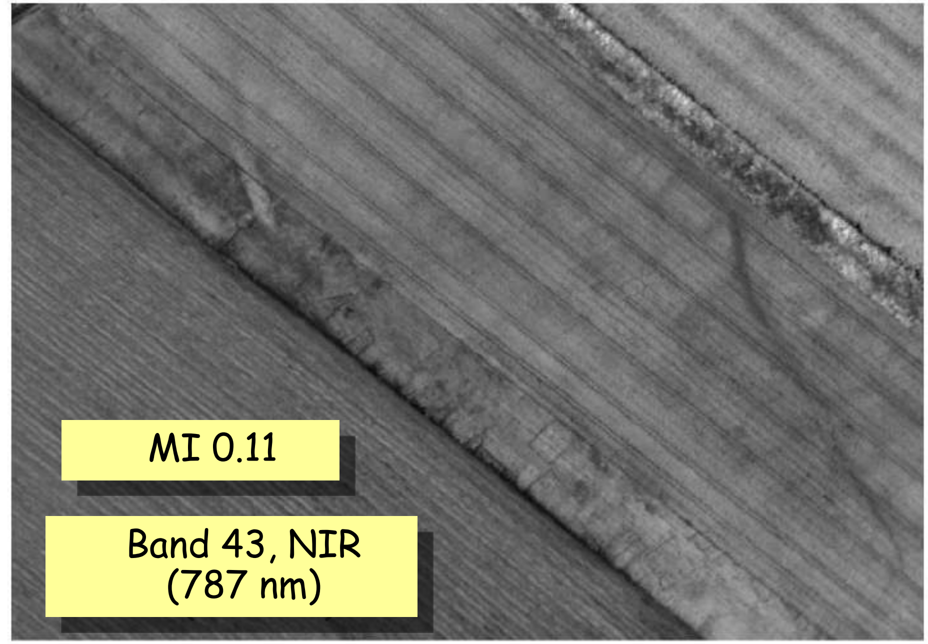


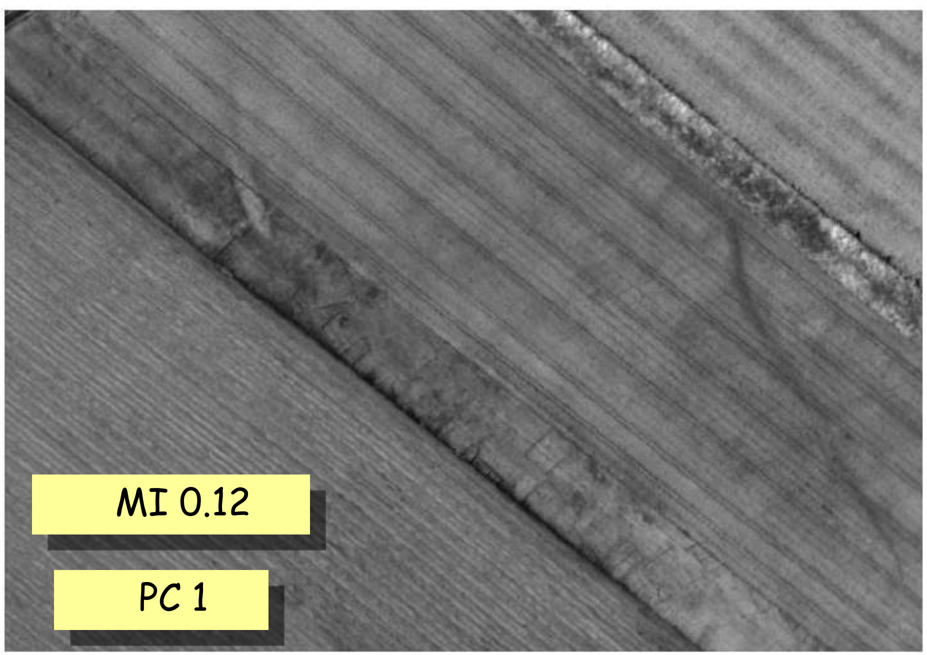
Summary (Mutual Information)

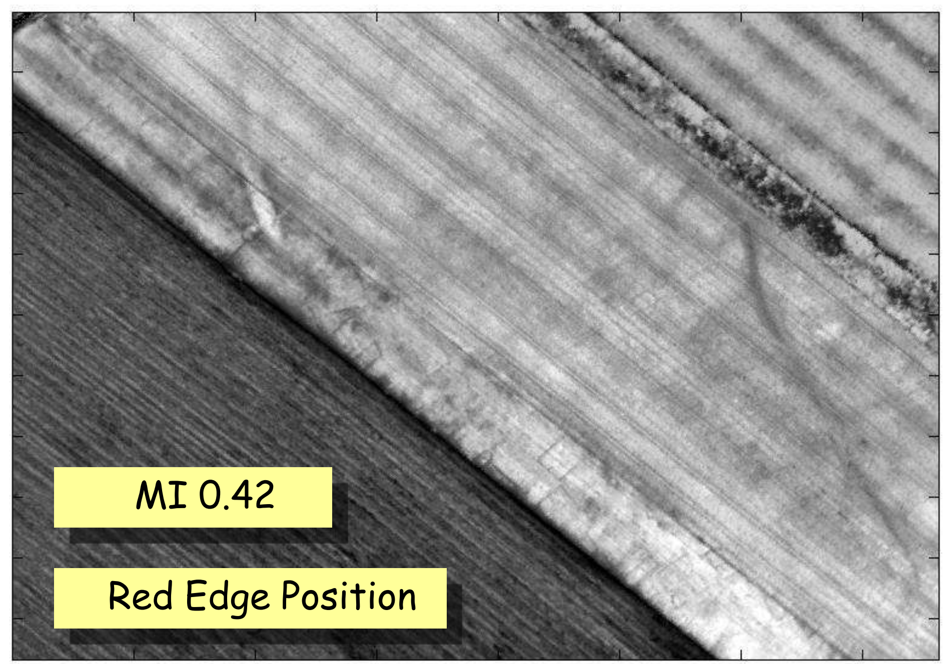


Mutual Info with what..?


MI 0.01
Bands
28/15/5 (RGB)


MI 0.11
Band 43, NIR
(787 nm)


MI 0.12
PC 1

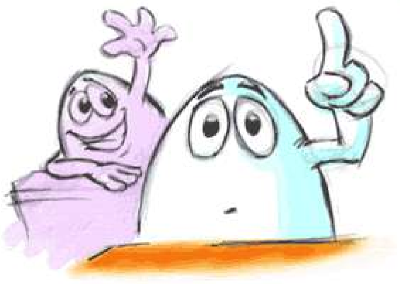

MI 0.42
Red Edge Position

Thanks a lot for your attention!

For any question / help:

Daniele.cerra@dlr.de

Questions?



CYPRUS UNIVERSITY OF TECHNOLOGY

Remote Sensing Exercises with Matlab

With a special focus on Hyperspectral Image Processing

Daniele Cerra

Remote Sensing Technology Institute
Earth Observation Center (EOC)
German Aerospace Center (DLR)

email: daniele.cerra@dlr.de

12-15 June 2017

Welcome to these Matlab exercises. At the end of this course we will have an idea of the different tools, data types and algorithms that we can use in the processing of digital satellite images. To do this we will make good use of the simplicity of programming with Matlab which is a high level programming language. Nevertheless, we will give during our lectures a deeper insight into mathematical models used in our analysis.
After an introduction to the program, we will learn how to load data and carry out simple operations on them, we will apply different filters on digital images both in time and frequency domain, we will extract edges and 'play' with histogram and spatial, spectral and radiometric resolutions.
In the second part of this course we will focus on specific algorithm for feature extraction, noise reduction and unmixing specific to hyperspectral data processing.
Have fun!

1 INTRODUCTION TO MATLAB

In this course we will use a toolbox developed independently from Matlab (the Hyperspectral Toolbox), plus the image processing toolbox which is built-in in Matlab.

1.1 FIRST STEPS

Matlab's interface is represented in Fig. 1.1.
Download the Hyperspectral toolbox from:
`http://sourceforge.net/projects/matlabhyperspec/`
Add the toolbox to the path from File > Set Path. Then copy all the provided files (with code/images) and copy them into a directory of your choice. Now you are set up for the course.
Let's try executing some operations directly from the command line, as a reminder of how Matlab works. Execute and understand the following commands:

- 3 * 4
- 7 - 3
- 11 / 7
- floor(11 / 7)
- mod(11, 7)
- sin(pi / 3)
- $a = 5 \wedge (7/2)$

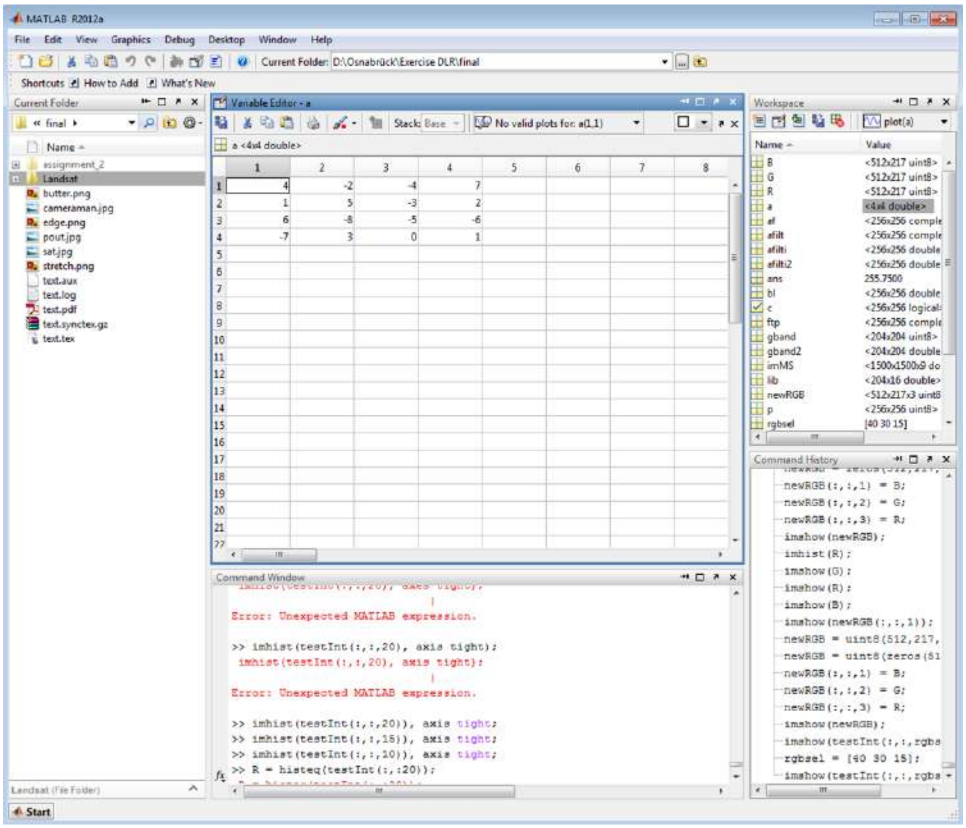


Figure 1.1: Matlab interface. The commands are executed from the command window (console). On the left the structure of the active directories. On the upper right the active variables, that can be analysed by clicking on them. On the lower right a window allows a quick reuse of the last commands given.

1.2 Operations with matrices

We will treat our images as 2- or 3-dimensional matrices, therefore before we start to use real data we need to know how to play around with matrix calculus in Matlab.
DO NOT rush through the experiments: take your time and be sure you understood what a command does before you go to the next. If you have problems just ask me.
Execute and understand the following commands (the symbol % at the end of a command denotes a comment):

- a = [4 -2 -4 7; 1 5 -3 2; 6 -8 -5 -6; 7 3 0 1]
- a(2,3)
- a(2:3,3:4)
- a(2:3,:)
- max(a)
- max(a(:)) % What is the difference with the previous command?
- inv(a) % Invert the matrix
- a * inv(a) % Is it the result you expected?
- a' % Transpose the matrix a
- size(a)
- test = reshape(a,8,2) % This command to change the dimensionality of the data is very important, and we are going to use it often in hyperspectral data processing to switch between 3D and 2D representations of the data.
- test = a + 3
- b = [2 4 -7 4; 5 6 3 -2; 1 -8 -5 -3; 0 -6 7 -1]
- c = 2*a - 3*b
- c = a * b
- c = a .* b % What is the difference with the previous command?
- test = a + b
- test = a ∧2
- a > 0 % conditional test

Figure 1.2: Cameraman - sample image.

1.3 Load and display data and images

1.3.1 1-dimensional data

Execute and understand the following commands.

- x = [0:0.1:2*pi]
- plot(x,sin(x))
- plot(x,sin(x),'.',x,cos(x),'o')

1.3.2 2D data (images)

- cameramanData=load('cameraman')
- c = cameramanData.cameraman;
- imshow(c)
- imhist(c), axis tight

The parameter "axis tight" is given into the histogram plot. Try to visualize the histogram without this additional parameter: what's the difference?

1.3.3 RGB Images

An RGB image has 3 bands: understand how to perform basic manipulation of pixels and bands in this section.

- rgbImg = imread('sat.jpg'); % ; suppresses the output
- size(rgbImg)

Figure 1.3: Sample RGB image.

- rgbImg(100,200,1:3)
- imfinfo('sat.jpg')

Exercise 1

Display each band from the RGB image as a grayscale image.

2 Basic operations with images in grayscale values

2.1 Masking / Thresholds

Exercise 2

Use the conditional test seen before, and create a binary image 'mask' containing the photographer silhouette, using as input the image 'cameraman', in which therefore all pixels with value larger than a given threshold have value 1, and all the others value 0.
You can create the mask with the command

mask = variableImage > selectedThreshold

Visualize it with imshow(mask), are you satisfied? If not, change the threshold.

You can use this mask to create a thresholded version of the image by copying the original image in a new variable 'imCopy' and then using the command
imCopy(mask) = 0;

Visualize the image with and without mask: what's the difference?

Figure 2.1: Image with low contrast.

2.2 Modify the spatial resolution (downsampling/upsampling)

The function imresize(image, newSize) or imresize(image, newSize, parameter) modifies the resolution of the image to the size

size = originalSize * newSize

with nearest neighbour interpolation. To change the resolution of an image keeping its size unaltered, you have first to downsample it and then upsample it back to its original size.

Exercise 3

Change the resolution of the image to 1/4 of the original. The new image must have the same size as the original image. To use nearest neighbour interpolation in the upsampling you must use the command imresize with three input parameters, and use 'nearest' as third parameter.
To find more information about any matlab command, type it in the command line and hover your mouse above it to visualize a short description in a pop-up window. You can also select the text of the name of the routine and press F1 to open a help file on that routine.

2.3 Operations with histograms

Let's load an image with low contrast.

- p = imread('pout.jpg');
- imshow(p)
- imhist(p), axis tight

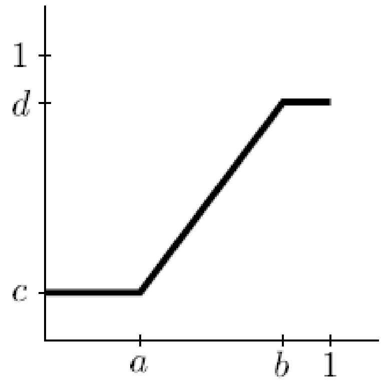


Figure 2.2: The stretch function used by imadjust.

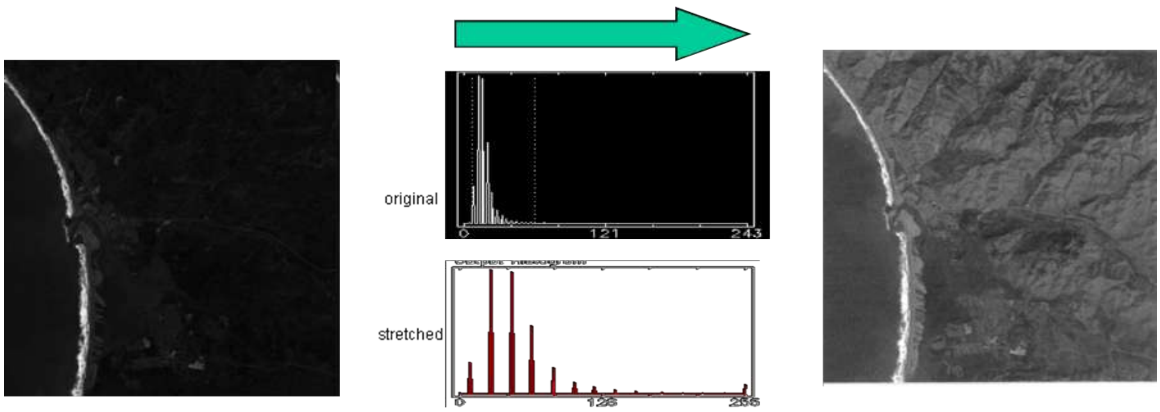


Figure 2.3: Histogram stretch.

Now let's use the function 'imadjust' to improve the image with a histogram stretch. Ref. to the graphical description of how imadjust works in Fig. 2.3.

Exercise 4

Execute the command:
pstretch = imadjust(p,[a,b],[c,d]);
Which values would you choose for a, b, c and d? The values (a, b, c & d) must be between 0 and 1.
Suggestion: check the histogram!

Exercise 5

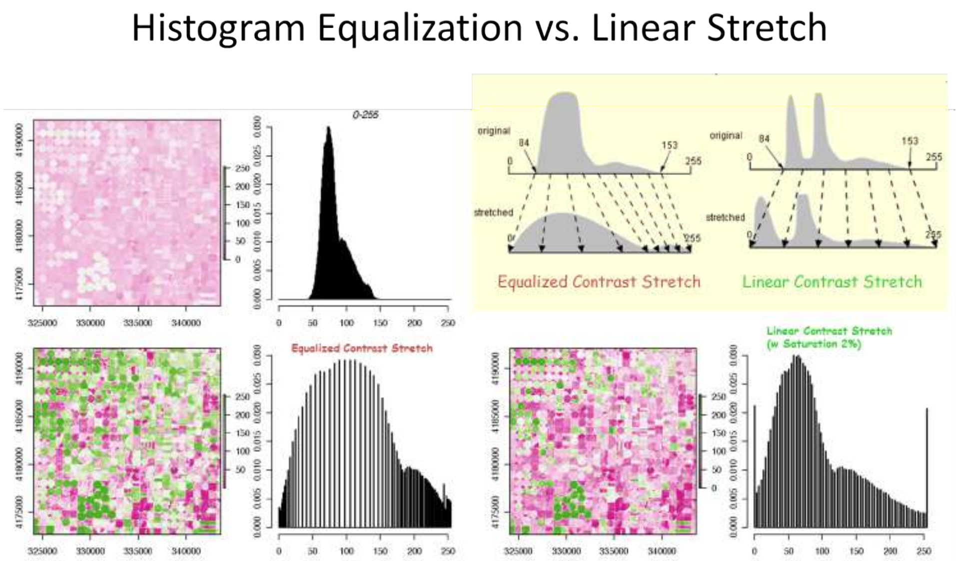


Figure 2.4: Stretch and equalization of the histogram.

Carry out a histogram equalization using the built-in Matlab function *histeq*: for differences between histogram stretch and equalization ref. fig. 2.3.

peq = histeq(p);

Compare the results!

3 Filtering of digital images

Now we will apply some basic filters to a single-band image. Filtering can be done in time domain (by direct manuipulation of the pixels) or in frequency domain (by manipulation of the different sinusoidal functions which constitute the image). Let's refresh some ideas before we move forward and then let's start with filtering in time domain.

3.1 Filtering in time domain

Exercise 6

Let's create an analysing window that we are going to use to carry out some basic filtering operations on the image 'cameraman', after checking some slides on low-pass and high-pass filters..
Let's start with a low-pass filter:

Convolution Examples: Original Images

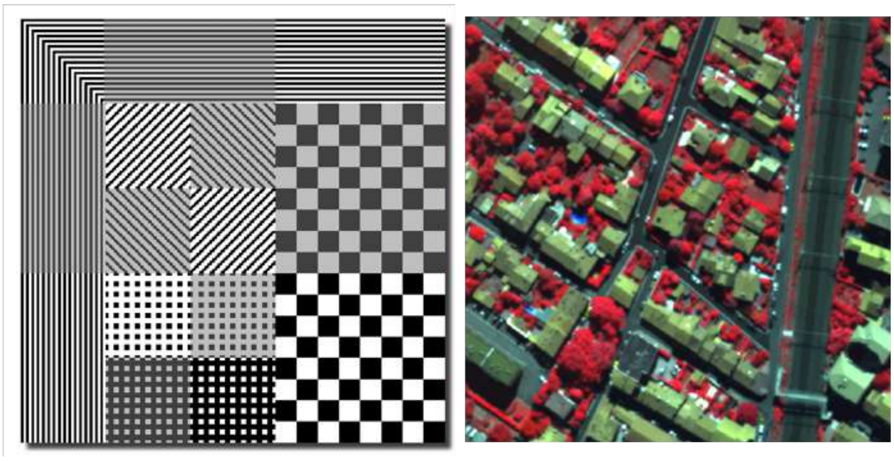


Figure 3.1: Original.

$$\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

Convolution Examples: 5×5 Blur

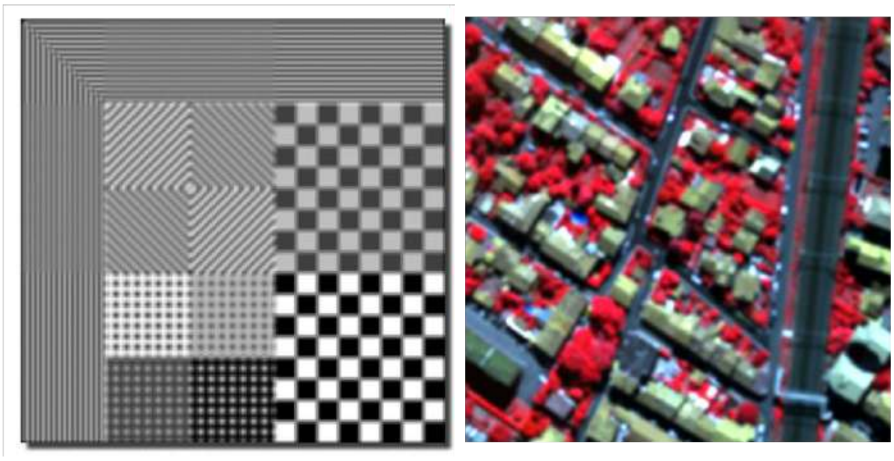


Figure 3.2: Blur / lowpass.

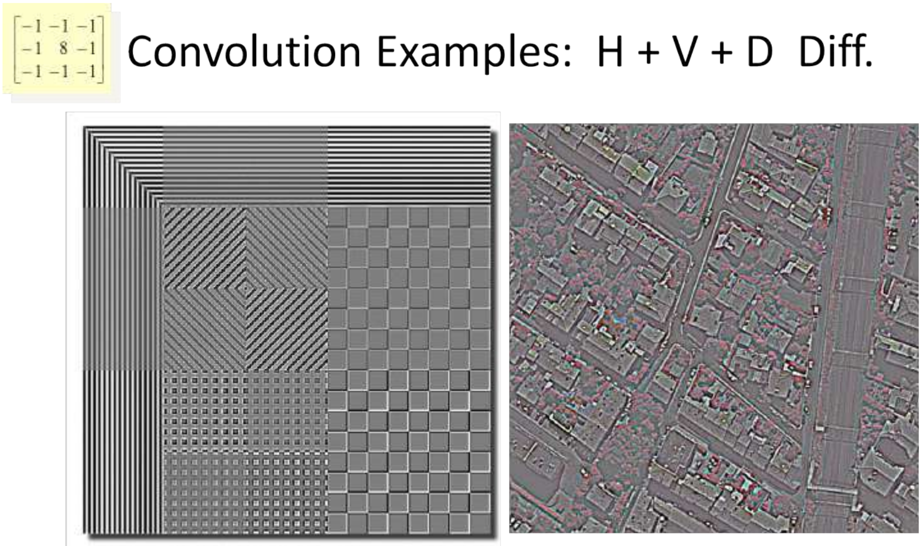


Figure 3.3: Highpass.

- Use the command "ones(n,n)" to create a square matrix of size $n \times n$ filled with 1. Consider $n = 3$ or $n = 5$, or $n = 7$ for a more extreme result.
- Divide the matrix by a constant, so that the sum of all its elements is equal to 1. Check that it is correct with the command sum(your_matrix(:)).
- Use the function filter2(f, image) to apply the filter to the image.
- Use the command result=uint8(result) to convert the image, that now has decimal values, again in byte format.
- Visualize the results: imshow(result,[])
- The brackets [ ] stretch the image between its minimum and maximum values (as we did manually until now by manipulating the histograms)

Exercise 7

Repeat the previous experiment using the built-in matlab function med2filt which applies a 3x3 filter:
medFiltered = medfilt2(image);

Now visualize the original image, the low-pass filtered one and the median-filtered one. Which differences can you see?

Now apply the low-pass and the median filters 10 times to its own output using a loop.

Figure 3.4: Original image and Fourier spectrum (the components in Fourier are computed separately for each of the three canals RGB).

For example in the case of the low-pass filter you can write:

```
im2 = filterf(f,image);
for i=1:9
im2 = filterf(f,im2);
end
```

Which one is more "stable"? Why?

Exercise 8

Repeat the previous experiment, this time use the filter:

```
f = fspecial('laplacian')
```

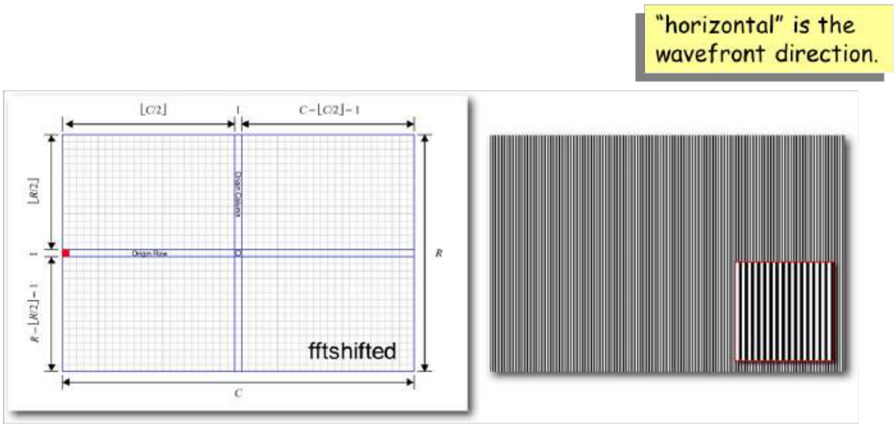
Which filter is this? What do you expect to happen when you apply it to a pixel in a homogeneous area and when you apply it on a pixel which stands on the edge of a building? And what can you see in the filtered image?

3.2 Filtering in frequency domain

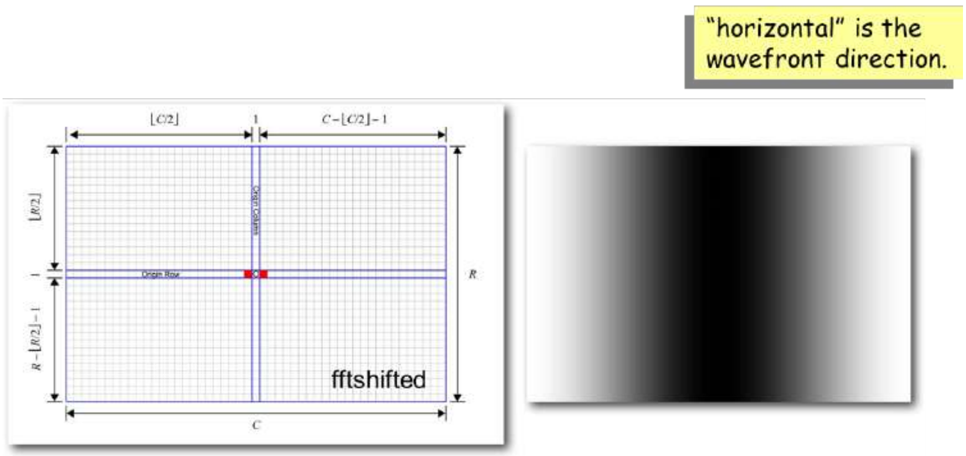
Now let's switch to the frequency (or Fourier) domain to perform similar operation in this very different representation of the data.

Exercise 9

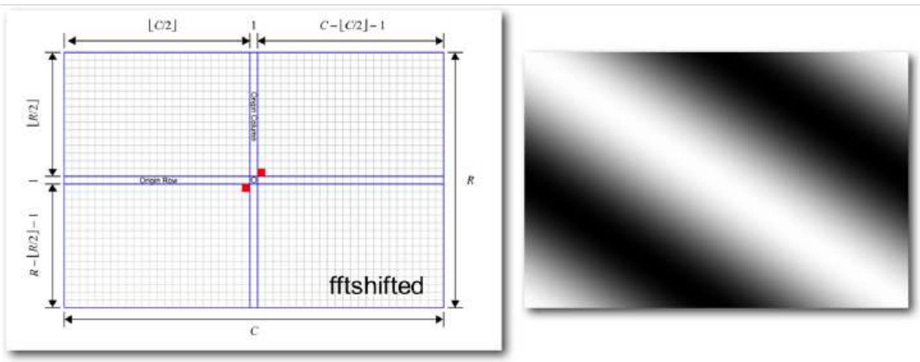
Inverse FFTs of Impulses



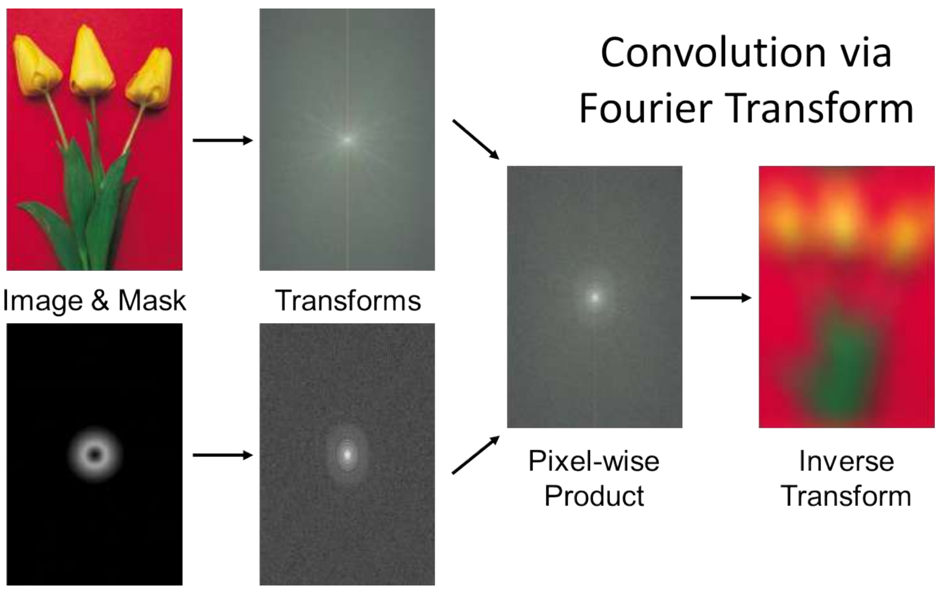
highest-possible-frequency horizontal sinusoid (*C* is even)



lowest-possible-frequency horizontal sinusoid "negative diagonal" is the wavefront direction.

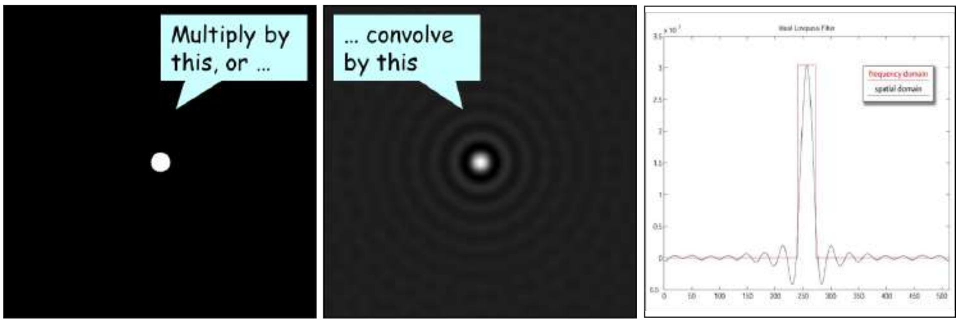


lowest-possible-frequency negative diagonal sinusoid



Ideal Lowpass Filter

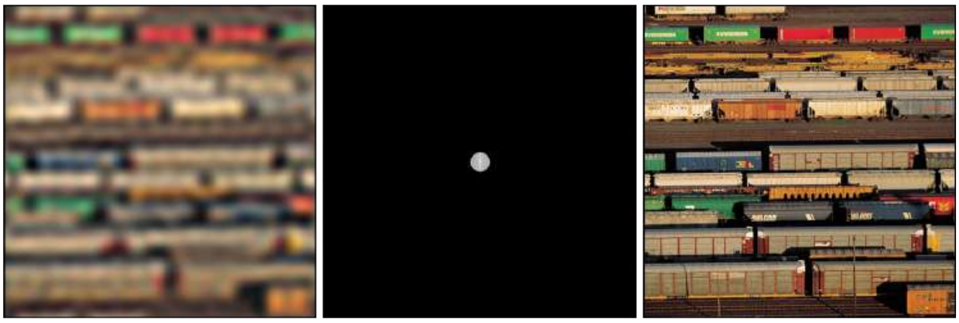
Image size: 512x512
FD filter radius: 16



Fourier Domain Rep. | Spatial Representation | Central Profile

Ideal Lowpass Filter

Image size: 512x512
FD filter radius: 16



Filtered Image | Filtered Power Spectrum | Original Image

Ideal filtering in Fourier. The Fast Fourier Transform (FFT) is an algorithm which converts an image from time domain (the one we are used to) to Fourier domain, where an image is represented as a sum of 2-D sinusoids.

- Use the FFT to convert the image 'cameraman' to the frequency domain:
  ftp = fft2(image);

- For a better visualization, let's put the low frequencies, which are now in the four corners of the power spectrum, in the center of the Fourier image:
  af = fftshift(ftp);

- Visualize the Fourier spectrum. For this purpose it is better to compute the logarithm of the spectrum. We must add 1 to obtain 'safe' values (as the logarithm of 0 is $-\infty$, this could introduce some 'small' numerical problems in our computations :)).

  imshow(log(1+abs(af)),[])

  The parentheses [] are needed to perform an image stretch (as we did before manually by manipulating the histograms), and the function abs(n) computes the absolute value of n (in this case we need it to obtain real values, as in origin the points in Fourier also have an imaginary part).

- Let's create an ideal circular filter, that is 1 in the center of the image and 0 in the rest of it:

  [x,y]= meshgrid(-128:127,-128:127);
  z = sqrt(x.∧2 + y .∧ 2);
  c = (z < 15)

- Visualize z to understand what you're doing

- Visualize the ideal filter c (now you should already know which command to use...)

- Apply the filter to the spectrum. In Fourier the filtering is obtained with a simple multiplication:

  afilt = af.*c;

- Visualize the resulting spectrum (remember to compute the logarithm and add 1 to its argument)

- Let's shift the spectrum in order to have again the low frequencies in the corners: afilt = fftshift(afilt);

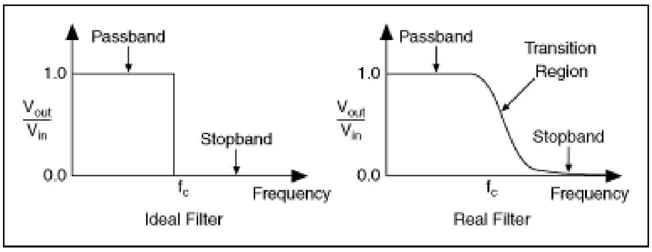


Figure 3.5: Difference between ideal filter and Butterworth filter (in one dimension).

- Let's switch back to time domain: afiltinv = ifft2(afilt);
- Visualize the results
- Can you see the ringing artefacts?
- What happens if we do the Fourier transform of the ideal filter c? Try to visualize its Fourier spectrum...

Exercise 10

Now let's filter the image with a Butterworth filter. The differences with the ideal filter can be seen in fig. 3.5, and the difference in the filtering effects will be clear at the end of this exercise.

- Let's create a Butterworth filter, which has a constant value in the center and gradually decreases to 0 on the edges of the circle:
- $bf = 1./(1 + ((x. \wedge 2 + y. \wedge 2)/60). \wedge 2);$
- Filter the image as in the previous example.
- Compare the results (to open a new image write 'figure, ' before the command imshow).
- What happens if you change the size of the Butterworth filter (the value 60 in the equation)? Try to use a larger or smaller filter.

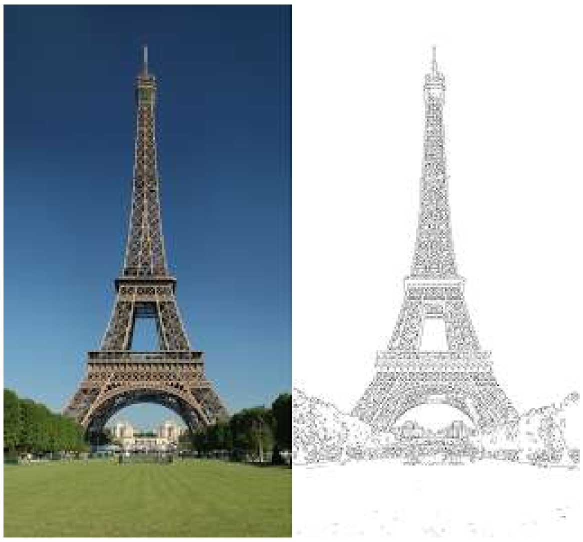


Figure 4.1: Example of edge extraction.

4 EDGE EXTRACTION

Slides - Canny

Exercise 11

Canny algorithm, defined in the 80's, keeps on being one of the favourite ways of extracting edges from an image (fig. 4). Let's apply it, and let's see what happens if we want to extract edges from a noisy image, and what we can do to solve the problem.

- Load the image "cameraman" (from the matlab file with the same name, not the jpg image)
- Extract the edge using the function edge(image, 'canny')
- Visualize the results
- Now let's add some gaussian noise to the image: imgNoise = imnoise(image,'gaussian',0,0.01); in which we are setting the mean and the variance of the noise to 0 and 0.01, respectively.
- Visualize the noisy image
- Extract the edges from the noisy image and visualize them: what happened?
- Use the window described in Exercise 6 to filter the noisy image
- Extract the edges again: could you solve the problems?

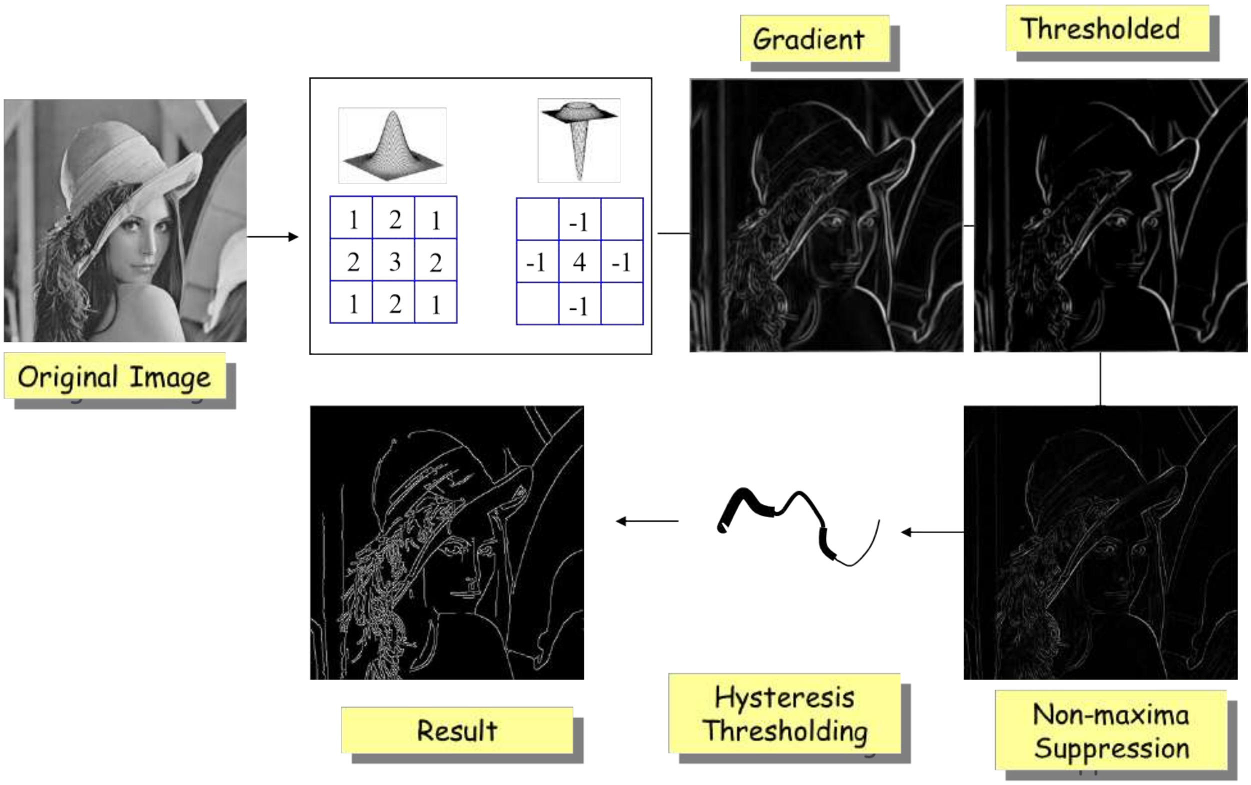


Figure 4.2: Canny's algorithm for edge extraction. After a lowpass filtering, a high-pass filter is applied to compute the gradients. A threshold selects only strong gradients. Afterwards, the thickness of each border is reduced to the size of a pixel. Finally, weak edges which are above a second lower threshold and were discarded in the first step are added to the final results, if they are connected to 'strong' edges.

5 Hyperspectral Image Processing

Slides on Introduction to Hyperspectral images

Examine a hyperspectral image.

- Open the hyperspectral data set with the command *load('test_image_new.mat')*;
- Two variables are loaded:
  - *ImgData* contains a 533 × 763 image with 65 bands acquired by the sensor AISA Eagle on the place in which the Roman city of Carnuntum once stood, somewhere in Western Austria.
  - *ImgInfo* contains ancillary information on the image.
  - You can retrieve the wavelength associated to each spectral band by accessing the variable ImgInfo.wavelenght

To visualize a single band and a single spectrum in the image: imshow(ImgData(:,:,18),[]);

This for example visualizes band 18.

Then let's select a single pixel and analyse its spectrum:
pixel = ImgData(100,100,:);
plot(pixel(:))

What does '100,100' mean?

Exercise 12

Visualize an RGB combination of the image, in which a band related to red frequencies will be loaded in the red (R) channel, a band related to green in G and a band related to blue in B.

- Let's select a band in the middle of the red frequency range (0,6-0,7 μm), a second one in the middle of the green range (0,5-0,6 μm), and a third one in the middle of the blue portion of the spectrum, (0,4-0,5 μm), and let's substitute the corresponding numbers of the bands in the following command to save them into a data structure which we name 'rgbsel':
  - rgbsel = [numbandRed numbandGreen numbandBlue]
- To visualize the image RGB use the command imshow(uint8 ((ImgData(:,:,rgbsel) - min(ImgData(:)))/10) ,[])
- in the above command, we create a byte image by converting to byte format (uint8 in Matlab) the 3 bands of the image stretched between 0 and 255 (we subtract the minimum and divide by an empirical number to rescale quickly the values).

6 Spectral Indices

Slides on NDVI

The Normalized Differential Vegetation Index (NDVI) is probably the most important spectral index, as it is can be computed also on multispectral images. It gives us an indicative quantification on the percentage of a pixel which is covered by green (alive) vegetation.
The NDVI is computed as:

$$NDVI = \frac{NIR - R}{NIR + R} \tag{6.1}$$

6.1 Applications: Archeology

Would you like to investigate if the image you visualized before contains some secret? Let's look out for crop marks, which show local anomalies in the vegetation health status due to underground structures not directly observable from the surface.

Exercise 13

- Create a new version of the image in double format, because if we divide two images in integer format the result will also be integer! Use the command imageDouble = double(ImgData)
- Take as reference again the frequencies corresponding to each band in the image.
- Select a band number in the middle of the red portion of the frequency spectrum and another one in the NIR (Near InfraRed), around 800 nm.
- Assign to the variable NDVI_index the result of the Matlab function imdivide(numerator,denominator) to compute the NDVI. Refer to the equation above to correctly express numerator and denominator!
- Visualize the NDVI with imshow(NDVI_index,[0 0.2]),colormap(jet)
- compare with the RGB image you created previously. In which image can you spot the profile of buried underground structures?

7 Dimensionality Reduction

Slides: reminder on PCA

Exercise 14

Explore a hyperspectral dataset by applying a Principal Components Analysis (PCA) rotation.

- Open the hyperspectral data set with the command *load('Salinas_Lib')*;

- Two variables are loaded:
  - *Salinas_corrected* contains a 512×217 image with 204 bands acquired by the AVIRIS sensor over the Salinas valley, California.
  - *Lib* contains 16 spectra selected from the image.
- Convert the image to 2D using the command 'reshape', where the first dimension is the number of bands and the second the total number of pixels. Therefore a hyperspectral image of size 10 × 10 and 20 bands should have two dimensions sized 20 and 100 respectively.
- Use the function hyperPCT to perform a PCA rotation. Set the maximum number of components to 50 and assign the output value to three variables. Check the help for the function hyperPCT for more information.
- Let's prepare some Principal Components in the new matrix M_pct to visualize them correctly.
  - First of all you must convert in 3D the results with the command reshape
  - Subtract the minimum value in order not to have negative values: image = image - min(image(:))
  - Divide by the maximum number in order to have all the values between 0 and 1: image = image / max (image(:))
- Now let's check the eigenvalues (lambda) of the rotated features (in the third output variable): how many components contain relevant information? What are the characteristics of the last Principal Components?
- Compare the information in the first PCs with the different classes in the image.
  - Load the ground truth for the Salinas dataset with the command the command *load('Salinas_gt')*
  - Visualize the ground truth (see legend in fig. 7), which identifies several crops in the image (broccoli, lettuce, celery, grapes...) with the command *imshow(salinas_gt, [])*
  - Let's change the color map to separate visually the different classes: *colormap(jet)*
  - Now visualize an RGB combination of the 3 first PCs using the command figure, hyperImagesc(computedPCs,[band1 band2 band3]); where in band1 band2 and band3 you can try and use several PCs among the first 10.
  - Can you more or less separate the different classes?
- Can you 'see' more information in the first 3 PCs with respect to the RGB combination of the original HS image that you created in the previous exercise? Use the command figure, hyperImagesc(salinas_corrected,[10 20 30])

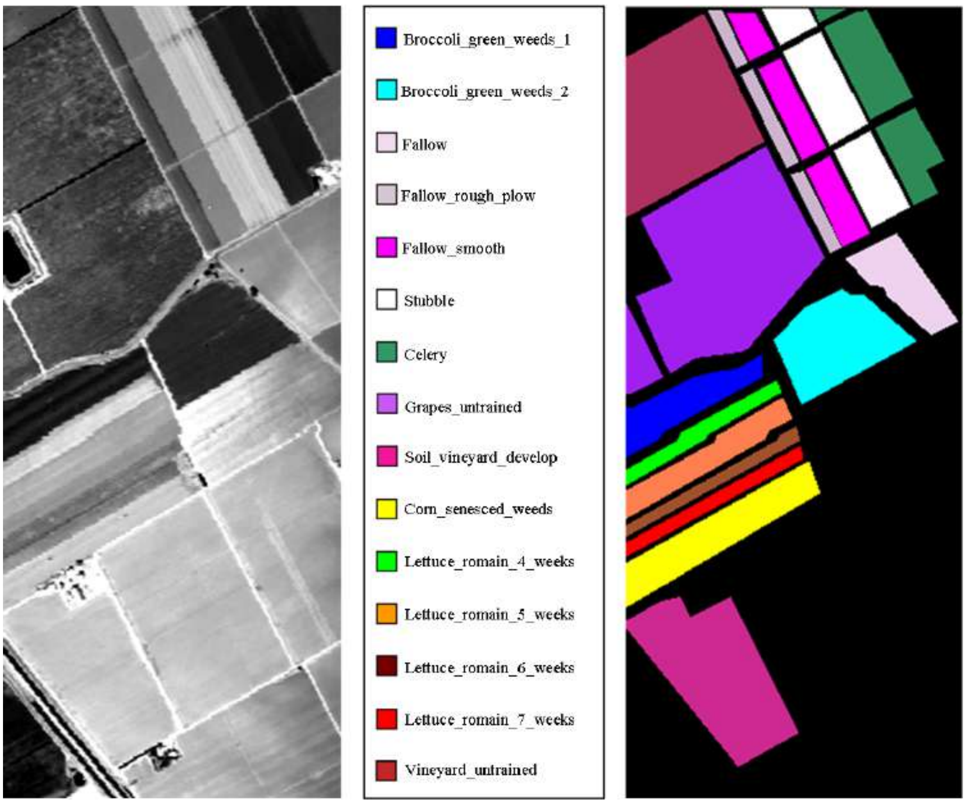


Figure 7.1: Band 42 and ground truth for the Salinas dataset.

- hyperImagesc is a much better way of visualizing a 3 band combination of a hyperspectral image: why we did not use it for the Carnuntum dataset? Try it out! What happens? Where is the problem?

8 SPECTRAL UNMIXING: WHAT IT IS?

Slides: reminder on spectral unmixing & introduction to Least Squares

Unmixing generally refers to a process which includes two steps:

- 1 - The identification of a set of pure (or purest) pixels in the scene known as endmembers, which are related to the spectra of macroscopically homogeneous materials.
- 2 - An endmember abundance quantification algorithm (inversion step) to define the percentage of different endmembers in each pixel.
- In our case step 1 has already been taken care of (results are collected in the spectral library 'lib')

```
function [M_pct, V, lambda] = hyperPct(M, q)
%HYPERPCA Performs the principal components transform (PCT)
%   hyperPct performs the principal components transform on a data matrix.
%
% Usage
%   [M_pct, V] = hyperPct(M, q)
% Inputs
%   M - 2D  matrix (p x N)
%   q - number of components to keep
% Outputs
%   M_pct - 2D matrix (q x N) which is result of transform
%   V - Transformation matrix.
%   lambda - eigenvalues
%
```

Figure 7.2: Information on the hyperPCT function.

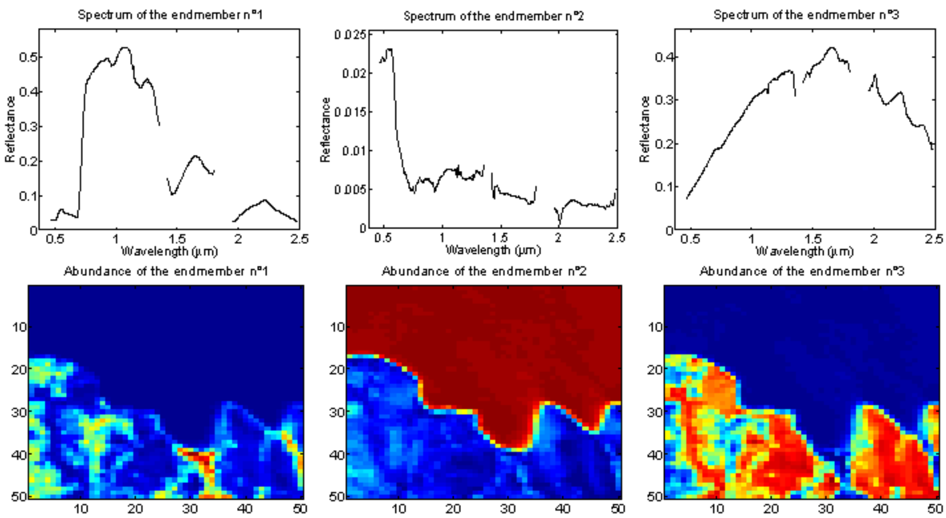


Figure 8.1: Results of Spectral Unmxing

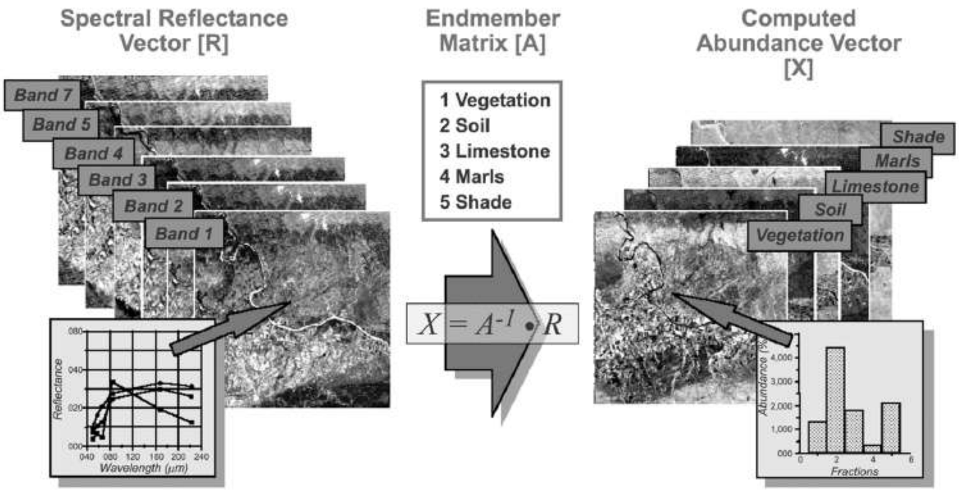
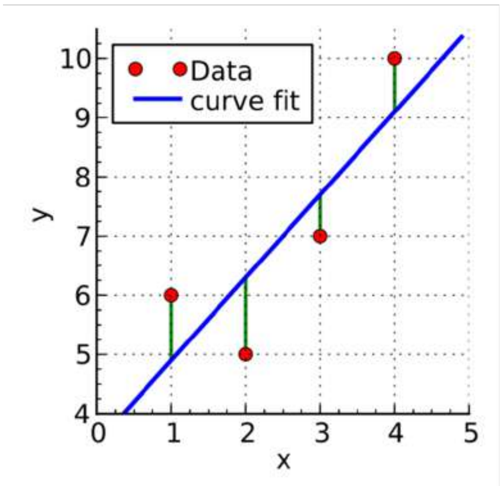


Figure 8.2: Further results of Spectral Unmxing

The idea of Least Squares



Find a line y=mx+q
in order to minimize the sum
of the square distances of the
data points from the line

Min(sum(dist(line, points)^2))

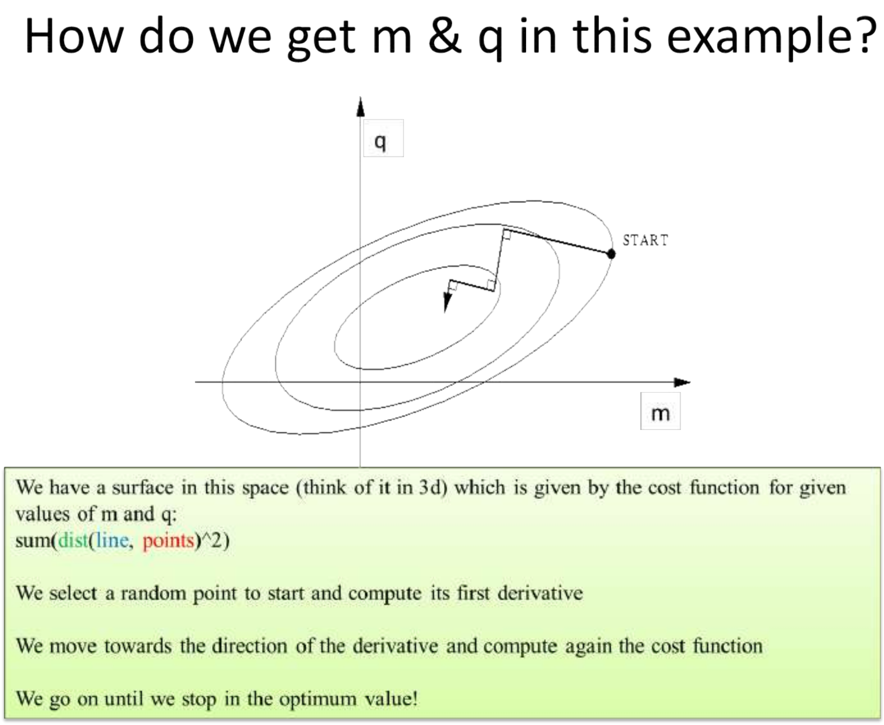
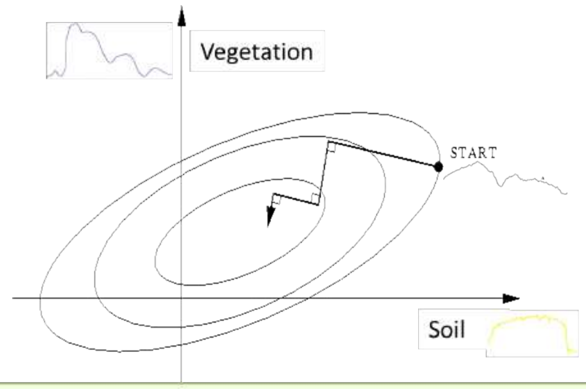


Figure 8.3: The idea behind the minimization of a cost function. The circles represent the cost function, which is to be imagined in 3rd dimensions as an upside-down cone pointing inside the page. Its center has then the lowest value.

What if we have 2 endmembers?



We try to express our spectrum *S* as a linear combination of two sample spectra V and S, for example:

S' = 0.4 V + 0.5 S.

Our cost function is the difference (*S - S'*)^2 between the original spectrum *S* and the reconstructed one *S'*.

We take another step and compute again the cost function = distortion, until convergence!

Figure 8.4: The case of linear spectral unmixing. In practice a direct solution is used, but it is easier to understand the concept through this diagram.

8.1 Unconstrained Least Squares

We will now perform a simple unmixing through Least Squares, which is a general method for the approximate solution of overdetermined systems (systems of equations which have more equations than unknown variables). We have an error function that we want to minimize, and to do it we could use the intuitive but time-consuming method presented in the images. Instead, we can solve this problem directly by multiplying each pixel by the pseudoinverse matrix of the endmembers: the pseudoinverse computation is a kind of relaxation of the inverse matrix one, and it can be applied to non-square matrices. This makes sure that we project each pixel orthogonally to the space spanned by the spectra related to the endmembers, minimizing the error: a mathematical explanation is not given here, try just to keep in mind the general idea which offers a direct (instead of iterative) solution to this problem.

Exercise 15

- Convert the HS image to 2D (you can reuse the variable you created in previous exercises), where the first dimension is equal to the number of bands and the second to the total number of pixels (use the command reshape). If the dimensions are in reverse order (first pixels and then bands), you have to transpose the 2D image as we have seen in the first pages of this exercise.
- Perform Unmixing through UCLS using the function hyperUcls(2dimage, spectralLibrary)
- Convert again the results to 3D. How many bands do you have now? What is each band representing?
- Check the abundance map for each material (or endmember). Use the jet colormap for a better visualization. Use the command 'colorbar()' to understand each abundancy map.

8.2 Non-negative Least Squares

Exercise 16

Follow the same steps as in the previous exercise, but this time use non-negative Least Squares, which forces all spectral abundances to be positive. This makes sense if we think about it, as a pixel can be composed 50% by water and 50% by vegetation, but -10% of any material would make no sense.
Use the function hyperNnls(2dImage, spectralLibrary). Is it faster or slower than its unconstrained version used in the previous exercise?

8.3 Analysis and Comparison of the Results

Exercise 17

Visualize all the results in two images. For each set of results, use the command subplot using a loop:

A sample loop in Matlab:

```
for i=1:10
i*2 % Prints i*2
end
```

Which result looks better? Could you separate the different classes on the basis of these results?

8.4 Classification and Validation

We will now perform a classification of the image, in which we will label each pixel with the class having the highest abundancy in the unmixing results.

Exercise 18

Classify the image using your unmixing results.

- Check the help for the function *max* by typing 'help max'. You will need to use the version with two outputs, one of which is the maximum abundance among all bands while the other shows which band has this maximum value: [output1, output2] = max(UnmixingResults)
- Try to build a single band image (in grayscale) which has as a value for each pixel the number of the material with highest abundancy in it (which is the number of the band with highest abundancy in the unmixing results).
- Take care: you don't want the abundancy value in the image you are creating, but only the number identifying the material (the results should range from 1 to 16).
- Now merge classes 10 and 16: assign to all pixels with class 16 the value 10, using something like: image(image_condition) = value;
  in which your condition is that the image is equal to 16.
- Mask your classification results using the ground truth: set to 0 all values for which salinas_gt is 0 (these are unclassified pixels, they are not part of the ground truth and we don't know which class they have).
- Visualize the results with the jet color map.

- Compare with the ground truth image. Does the classification results look good? Which ones look better between the results of unconstrained least squares and non-negative least squares?

Exercise 19

Validate the accuracy of your classifier. To do this, you can simply count the percentage of relevant pixels which are assigned to their correct class.

- You should have your classification results ranging from 1 to 15.
- Class 16 in salinas_gt was not included in the spectral library, therefore we have to ignore it. Create a new ground truth image containing the values of salinas_gt up to 15 (set to 0 all values equal to 16).
- Now count the total number of 'valid' pixels in the ground truth using the command sum. Inside you must write the condition salinas_gt > 0. Save this number in the variable 'TotValid'.
- Count the number of pixels in salinas_gt which are both greater than 0 and equal to the value in your classification images. You can do a test on two conditions at the same time in Matlab like this (try to execute this commands first to have an idea):

  2 > 0 & mod(4,2) == 0

  1 < 3 & 2 + 2 > 7

  Save this quantity in the variable totCorrect
- Compute the accuracy as totCorrect * 100 / totValid.
- How much is the accuracy for the two classifiers?

9 Noise reduction through spectral unmixing

Until now we saw how to represent a pixel as a linear combination of 'pure' spectra (endmembers) through spectral unmixing. You probably noticed that unmixing results have a dimensionality much lower than the original dimensionality of the data (we went from over 200 to only 16 dimensions).
What happened then with all the rest of our data? A part of them was redundant, therefore this synthetic way of representing it did not imply a loss of information. On the other hand, we have a part of the informational content of each pixel that of course couldn't be represented just as a linear combination of a restricted number of spectra in our spectral library.
Now we will see how this information we lost is mostly composed by noise. This is due to several sources: atmospheric absorption effects which decrease the signal-to-noise ratio in some bands, electronic interferences, variability in the path that a ray of light takes in coming

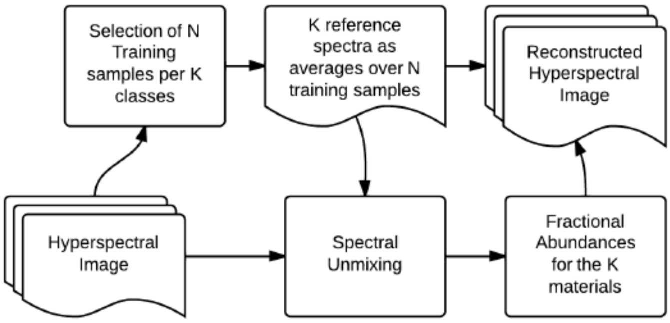


Figure 9.1: Summary of the simple algorithm for noise reduction based on spectral unmixing (UBD).

from the atmosphere, getting reflected on the ground and reaching the hyperspectral sensor. To do this, we are going to reconstruct the original data starting from our unmixing results and our spectral library.

Exercise 20

Unmixing-based Denoising (UBD).

This methodology, developed in our research group at DLR, takes as input a hyperspectral image and a spectral library (this must have some characteristics which we will describe later). The same image is given as output in which the noise is strongly reduced in the most problematic bands.

- Visualize band 1 and band 42 of the Salinas dataset. What can you say about the different level of noise affecting the two bands?
- Use the code 'UBD.m' using as input the image and the spectral library lib:
- Execute the command [reconstructed errors] = UBD (salinas_corrected, lib);
- UBD carries out a spectral unmixing step based on non-negative least squares, and reconstructs the original image ignoring any component of the signal which cannot be represented as a combination of the spectra contained in the spectral library lib.
- Visualize band 1 in the original and in the new dataset. Try improving visualization results using the command: imshow(image(:,:,1),[200 800]), which performs a histogram stretch between the values 200 and 800.

10 Band Selection

We saw how hyperspectral datasets contain redundant information as neighbouring bands are highly correlated. How can we select bands which contain the highest amounts of infor-

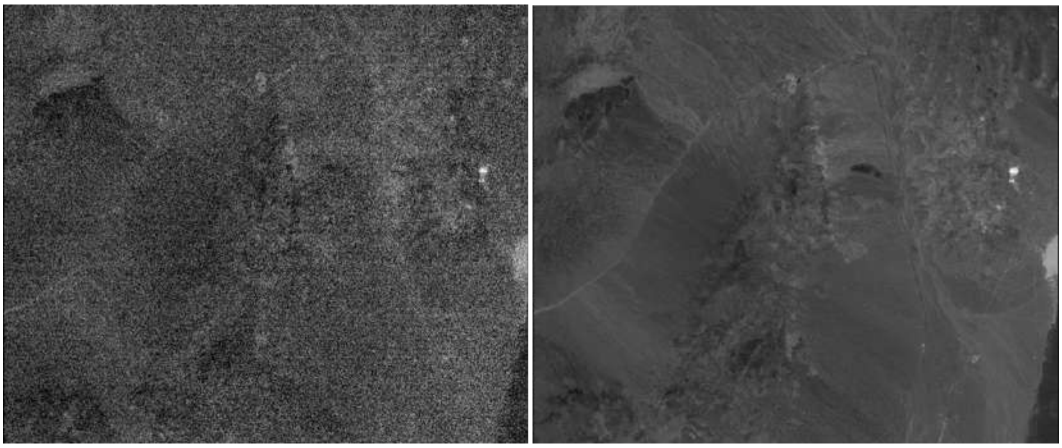


Figure 9.2: Example of noise reduction through UBD: band 224 for the AVIRIS image acquired over Cuprite, Nevada.

mation?

10.1 Inspect the data

Open band 1 and band 42 of the Salinas dataset. Which band you expect to have more information?

10.2 Variance

Slides about variance

Exercise 21

- Convert the Salinas image to 2 dimensions (pixels x bands).
- Create an array with the same number of elements as bands in the image: variances = zeros(numberOfBands, 1);
- For each value compute the variance of a band with the command var(mySingleBand)
- Plot the variances: plot(variances)

What can you see? Do you see any correlation between the variances and something else you have been creating lately?

10.3 Entropy

The variance only tells us the absolute variation in each band. Bands reflecting more energy will of course have higher variance, while "dark" bands will naturally have a low one. That's because there is no direct relation between variance and information! To have that, let's look at the concept of entropy...

Slides about entropy

Exercise 22

Compute the entropy for each band, in a similar way to what you have done before.

1. Create an array with the same number of elements as bands in the image (as you did before with the variances)
2. For each value compute the entropy of a band with the function entropy(image2D, bandNumber)
3. The entropies are computed as $H(X) = -\sum_x p(x) \log_2 p(x)$ and the histogram of each band is considered as the probability density function $p(x)$
4. Plot the entropies

Check the single bands in ENVI. How does it look like: is it getting better with respect to results from simple variance analysis? What do you have in correspondence of bands with low entropy?

10.4 Mutual Information

The point is, it is difficult to know which band is better a priori with measures which are computed for each band such as the entropy: a band with high entropy could be better for a given task and worse for another, or the entropy could be high just because the band is specially noisy. What you need is some kind of joint information measure, and what you need is to know if a band is better or worse for a given task at hand.
Let's look at the concept of mutual information, which is the base of many band selection algorithms.

Slides about mutual information

Mutual information allows us to quantify the information shared from two datasets, even if the values are completely different.

Exercise 23

First of all let's see what the differences in the joint probability function between the ground truth and different bands in the Salinas dataset represent (slide).

1. Save in a variable n the total number of pixels in Salinas, nRows × nColumns
2. Save in the variable test1 the Salinas ground truth, converted to one dimension with the command reshape: test1 = reshape(salinas,n,1);

3. Now create a new array that will contain the mutual information between each band and the ground truth: each value must be the result of applying the routine mutual-info(test1, image2D(:,bandNumber));

4. Now you can plot the mutual information. This is a very good indicator on which bands are the best to be used in our classifier.

Exercise 24

Now let's verify what is an usual effect of denoising the dataset. Do you think the mutual information will increase or decrease?

1. create a denoised version of band 1 of the salinas dataset using the Unmixing-based Denoising (UBD) you applied last time: [reconstructed errors] = UBD(salinas_corrected,lib);

2. Compute again the mutual information between band 1 of the denoised dataset and the ground truth image.

3. did the mutual information go up? This hints that this band is now much better (if taken singularly) to classify the areas of interest.

11 CLUSTERING

Slides on Clustering

Exercise 25

1. Let's open the Landsat image "subset" with the routine var = enviread('imageName','headerName')

2. Let's visualize an RGB combination with the command imshow(uint8(x(:,:,[3 2 1])),[]) or better open the image with ENVI and visualize an RGB combination (Yes, Matlab is not the best to visualize color images in a simple way).

3. Use the Matlab function var = kmeans(2D_image,nClusters) where 2D_image is the image reshaped to 2 dimensions (pixels, bands). Choose a number between 6 and 9 for the number of clusters.

4. Reshape the results in var to 2 dimensions and visualize the results. Apply the jet color map: colormap(jet)

5. Do the different clusters represent different targets in the scene?

12 MORPHOLOGICAL PROCESSING

Slides on Morphological Processing
We will now perform some basic morphological processing operations. We know already the median filter which will be not taken into account.

Exercise 26

1. Create your own structuring element: it can be a 3x3 matrix filled with ones using the command: w = ones(3,3) or a cross with the command: w = [0 1 0; 1 1 1; 0 1 0]
2. Get your own binary image by creating a mask! Select all pixels in band 7 (Short Wave Infrared) with a value smaller than 10. Visualize the mask.
3. Let's see and understand what happens when we apply the following morphological functions:
   a) Dilation -> imMorph = imdilate(image,w);
   b) Erosion -> imerode(image,w)
   c) Opening -> imopen(image,w)
   d) Verify that the result of imopen is the same as applying a dilation to an eroded image.
   e) Closing -> imclose(image,w)
   f) Verify that the result of imclose is the same as applying an erosion to a dilated image.
   g) What operator would be the best to extract this particular sea area you are interested in?
   h) Now fill the holes in the processed sea mask with the command newMask = imfill(seaMask);
   i) Click on the areas with value 0 you want to be filled with value 1. Take care not to click on anything which is connected to the large black area, if not everything will be filled by 1! When you are done, right click on the image to close the tool.
   j) Does it look nice?
4. Let's see how to use morphology to improve some bad classification results..
   a) Open one of your previous classification results, better if it is a bad one such as the one computed on the basis of the unconstrained least squares unmixing results.
   b) "open" and "close" the image a couple of times: does it improve the results?
   c) What happens if you use a very large structuring element such as w=ones(9,9)? Is it improving results more than a small one?