Introduction to Image Processing

Lecturer:
Dr. mat. nat. Christian Kehl

Week 1.1 – Introduction and Basics

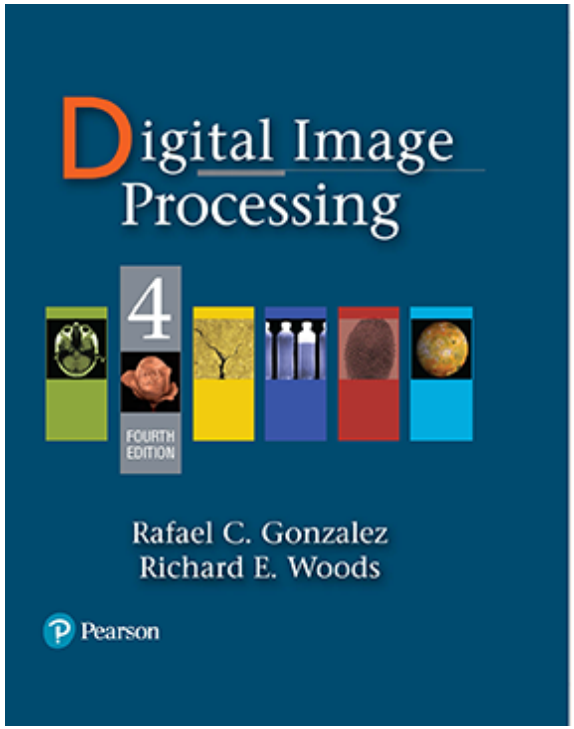
November 11, 2024

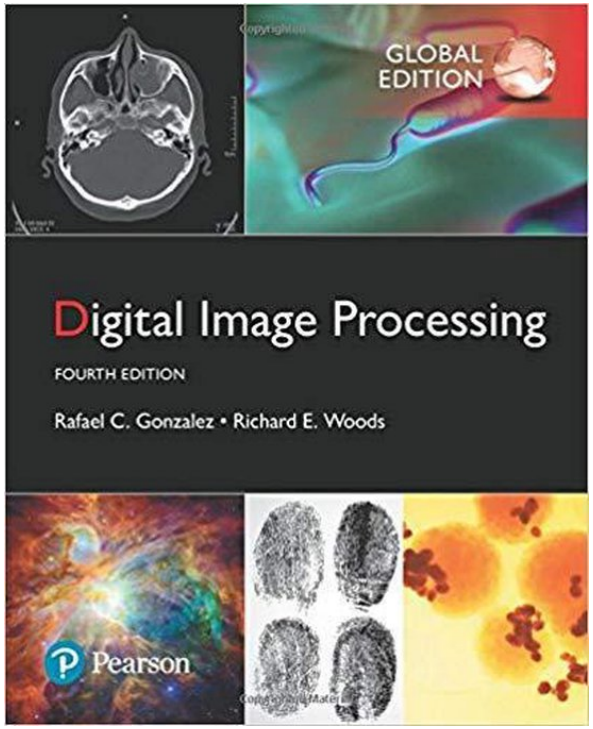
Staff

How to reach us?

- (Dr. mat. nat.) Christian Kehl (coordinator) e-mail: c.kehl@rug.nl, room: 5161.0456
- teaching assistants (TA's):
  - **Jeroen** de Baat
  - **Anton** Bredenbals
  - collective e-mail: introip@rug.nl

Course Book





Gonzalez & Woods, 4 / E. - Global Edition, 2018.

`www.ImageProcessingPlace.com`

Entry requirements

- **Mathematics**:
  - Linear algebra
  - Calculus/Analysis (integration, differentiation, Taylor series, etc.)
  - Discrete structures (set theory, lattices)
  - Elementary statistics

Entry requirements

- **Mathematics**:
  - Linear algebra
  - Calculus/Analysis (integration, differentiation, Taylor series, etc.)
  - Discrete structures (set theory, lattices)
  - Elementary statistics
- **Programming skills (ideally: Java)**

Lab assignments

- Group assignments, in groups of 2 – 4 students.

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.
- Assignment submission & (self-)correction via *Themis* (`https://themis.housing.rug.nl/course/2024-2025/improc`)

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.
- Assignment submission & (self-)correction via *Themis* (`https://themis.housing.rug.nl/course/2024-2025/improc`)
- Submit signed *work attribution form* with the last code submission to Themis

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.
- Assignment submission & (self-)correction via *Themis* (`https://themis.housing.rug.nl/course/2024-2025/improc`)
- Submit signed *work attribution form* with the last code submission to Themis
- Penalty for delayed lab submission past deadline: 0.5 pt per 12h delay.

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.
- Assignment submission & (self-)correction via *Themis* (`https://themis.housing.rug.nl/course/2024-2025/improc`)
- Submit signed *work attribution form* with the last code submission to Themis
- Penalty for delayed lab submission past deadline: 0.5 pt per 12h delay.
- Lab sessions start in the 2nd **week** of the course.

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.
- Assignment submission & (self-)correction via *Themis* (`https://themis.housing.rug.nl/course/2024-2025/improc`)
- Submit signed *work attribution form* with the last code submission to Themis
- Penalty for delayed lab submission past deadline: 0.5 pt per 12h delay.
- Lab sessions start in the 2nd **week** of the course.
- Total of 4 assignments, each covering a 2 week period

Lab assignments

- Group assignments, in groups of 2 – 4 students.
- Assignment submission is done *as a group*.
- Please use your own laptop, if possible (*BYOD* scheme).
- See Brightspace for lab assignments and **submission deadlines**.
- Implementation done exclusively in **Java**.
- Assignment submission & (self-)correction via *Themis* (`https://themis.housing.rug.nl/course/2024-2025/improc`)
- Submit signed *work attribution form* with the last code submission to Themis
- Penalty for delayed lab submission past deadline: 0.5 pt per 12h delay.
- Lab sessions start in the 2nd **week** of the course.
- Total of 4 assignments, each covering a 2 week period
- Assignment 4 – Surprise ...

Support

- Questions about the theory?

Support

- Questions about the theory? First check section 2.6 of the course book: *Basic Mathematical Tools used in Image Processing*. If that does not help, ask the lecturer(s).

Support

- Questions about the theory? First check section 2.6 of the course book: *Basic Mathematical Tools used in Image Processing*. If that does not help, ask the lecturer(s).
- Questions about the lab sessions?

Support

- Questions about the theory? First check section 2.6 of the course book: *Basic Mathematical Tools used in Image Processing*. If that does not help, ask the lecturer(s).
- Questions about the lab sessions?
  1. Review info on Brightspace & study the assignment sheet.
  2. Ask your TA.
  3. Upon TA-referral, ask the lecturer(s) (via mail).

Exam

- The exams are **digital, closed-question** exams (i.e. no text-writing involved).

Exam

- The exams are **digital, closed-question** exams (i.e. no text-writing involved).
- There will be one final exam in the exam week of 1b.

Exam

- The exams are **digital, closed-question** exams (i.e. no text-writing involved).
- There will be one final exam in the exam week of 1b.
- The exam is *closed book*. You are allowed to use a single-page A5 – note sheet (self-prepared & hand-written; can differ between exam 1 & 2; readable *without extra tools*).

Exam

- The exams are **digital, closed-question** exams (i.e. no text-writing involved).
- There will be one final exam in the exam week of 1b.
- The exam is *closed book*. You are allowed to use a single-page A5 – note sheet (self-prepared & hand-written; can differ between exam 1 & 2; readable *without extra tools*).
- Last lecture slot: exam-preparation session + Q&A.

Exam

- The exams are **digital, closed-question** exams (i.e. no text-writing involved).
- There will be one final exam in the exam week of 1b.
- The exam is *closed book*. You are allowed to use a single-page A5 – note sheet (self-prepared & hand-written; can differ between exam 1 & 2; readable *without extra tools*).
- Last lecture slot: exam-preparation session + Q&A.
- Check the information on Brightspace about the material (scope) of the exam.

Lab assignment grades

- Lab assignments are mandatory. In case of no submission, you get the grade 0 for that lab assignment.

Lab assignment grades

- Lab assignments are mandatory. In case of no submission, you get the grade 0 for that lab assignment.
- **One** resit lab is given at the end of the block for the labs missed because of a *force majeure* or for the labs with a *non-passing grade* (< 5.5).

Lab assignment grades

- Lab assignments are mandatory. In case of no submission, you get the grade 0 for that lab assignment.
- **One** resit lab is given at the end of the block for the labs missed because of a *force majeure* or for the labs with a *non-passing grade* (< 5.5).
- You cannot take resit to improve your already passing grade (common FSE policy).

Grading

- The final grade is computed as follows:

$$F = \textbf{if } (E, A) < 5.5 \textbf{ then } F = 5.0(\text{fail}) \textbf{ else } F = 0.7 \times E + 0.3 \times A$$

, with:

  - F – final mark
  - E – (cumulative) exam mark
  - A – (cumulative) assignment mark

Grading

- The final grade is computed as follows:

  $$F = \textbf{if } (E, A) < 5.5 \textbf{ then } F = 5.0(\text{fail}) \textbf{ else } F = 0.7 \times E + 0.3 \times A$$

  , with:
  - F – final mark
  - E – (cumulative) exam mark
  - A – (cumulative) assignment mark
- If $5.5 \leqslant F \leqslant 6.0$ then $F = 6$.
- F is rounded to the nearest half-integer.
- Resit grade replaces the original grades for (E, A) (whichever applicable)

Grading

- Exams E:
  - Number of exam(s): 1
  - Weight of the exam grade E: 0.7

Grading

- Exams E:
  - Number of exam(s): 1
  - Weight of the exam grade E: 0.7
- Assignments A:
  - Number of (graded) assignments: 3
  - Weight per assignment to grade of A: 0.33
  - Contribution of an individual assignment to final grade F: 0.1

Policy on AI language models (e.g., ChatGPT)

- You *may* use AI (language) models only as a grammar- and spelling writing aid.
- You *must not* use AI models to bypass target learning outcomes of the course (see Ocasys).
- You *must not* use AI as functional aid (see Ocasys[1]).

[1] RUG AI policy: `https://www.rug.nl/about-ug/organization/quality-assurance/education/artificial-intelligence-ai/`

Policy on AI language models (e.g., ChatGPT)

- You *may* use AI (language) models only as a grammar- and spelling writing aid.
- You *must not* use AI models to bypass target learning outcomes of the course (see Ocasys).
- You *must not* use AI as functional aid (see Ocasys[1]).
- This is a **zero-tolerance** policy. Rules, regulations and consequences are in-place.

[1] RUG AI policy: `https://www.rug.nl/about-ug/organization/quality-assurance/education/artificial-intelligence-ai/`

Further information

Check Brightspace page of the course. If the information that you are looking for is not there contact the TA's (for labs) or lecturer(s).

Course Overview

- *Introduction & Motivation*

Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.

Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.

Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.
- *Spatial filter banks*: filter composition, image convolution, analytical filters.

Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.
- *Spatial filter banks*: filter composition, image convolution, analytical filters.
- *Morphological processing & filters*: Erosion, dilation, openings and closings; thinning, rank-order filters.

Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.
- *Spatial filter banks*: filter composition, image convolution, analytical filters.
- *Morphological processing & filters*: Erosion, dilation, openings and closings; thinning, rank-order filters.
- *Frequency filters*: Pixel-, local- & global functions; (discrete) Fourier transform, (discrete) cosine transform.

Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.
- *Spatial filter banks*: filter composition, image convolution, analytical filters.
- *Morphological processing & filters*: Erosion, dilation, openings and closings; thinning, rank-order filters.
- *Frequency filters*: Pixel-, local- & global functions; (discrete) Fourier transform, (discrete) cosine transform.
- *Image compression*: Lossy- and lossless image compression.

Course Overview

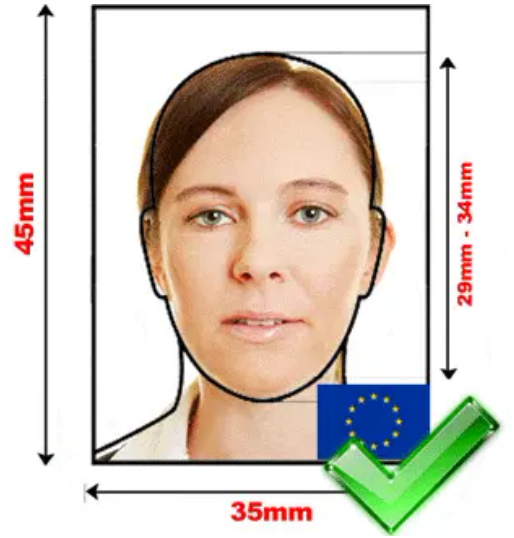
- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.
- *Spatial filter banks*: filter composition, image convolution, analytical filters.
- *Morphological processing & filters*: Erosion, dilation, openings and closings; thinning, rank-order filters.
- *Frequency filters*: Pixel-, local- & global functions; (discrete) Fourier transform, (discrete) cosine transform.
- *Image compression*: Lossy- and lossless image compression.
- *Image segmentation*: Thresholding, region-growing.

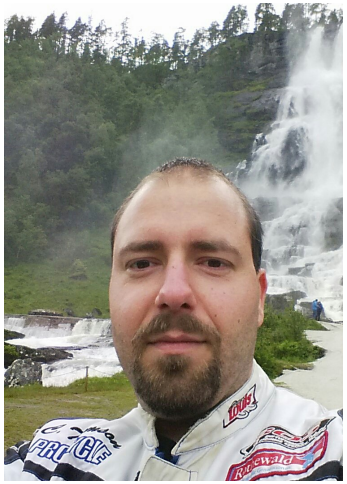
Course Overview

- *Introduction & Motivation*
- *Fundamentals*: Information theory, measuring images, image enhancement, multi-channel images.
- *Geometric transformations*: Projection, pixel interpolation, rigid- affine transformations, basic corner detection.
- *Spatial filter banks*: filter composition, image convolution, analytical filters.
- *Morphological processing & filters*: Erosion, dilation, openings and closings; thinning, rank-order filters.
- *Frequency filters*: Pixel-, local- & global functions; (discrete) Fourier transform, (discrete) cosine transform.
- *Image compression*: Lossy- and lossless image compression.
- *Image segmentation*: Thresholding, region-growing.
- **Special-Interest Topic**: surprise.

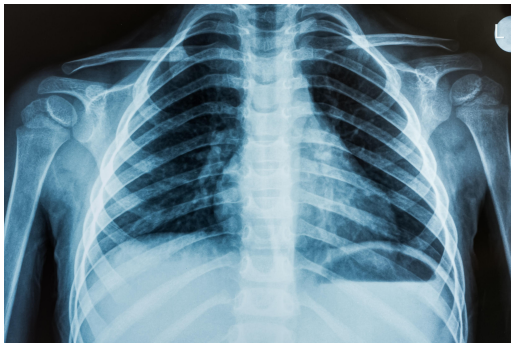
Introduction to Images

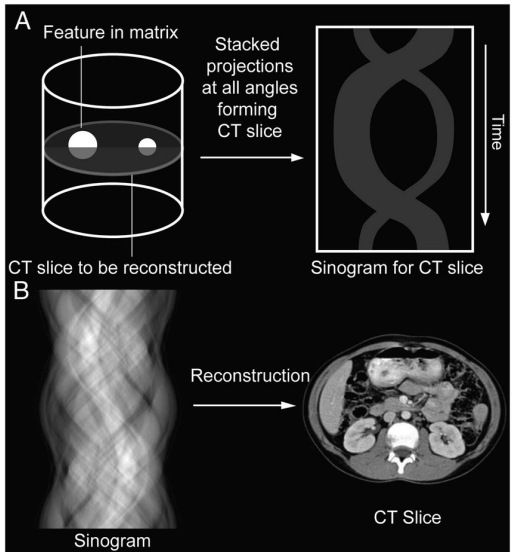
(Digital) Images ...

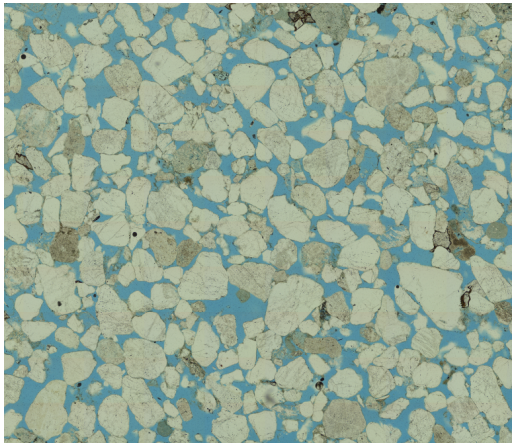


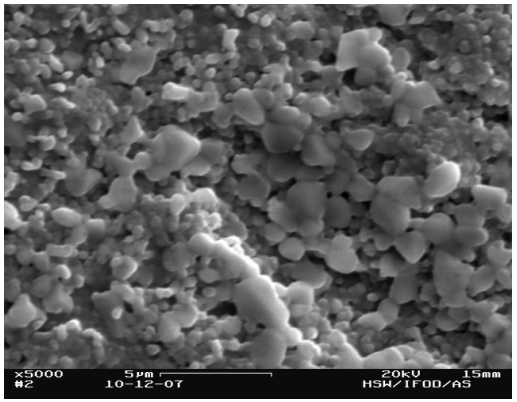


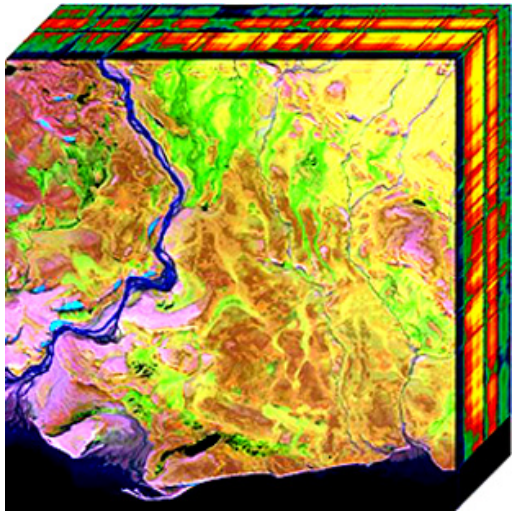
... also: Digital Images











Definition(s)

Digital Image Processing is the processing of information contained in a digital image.[2]

[2] [Oxford Dictionary of CS, 2016]
[3] [Gonzalez & Woods, 2018]
[4] [Geeks4Geeks, 2022]
[5] [Medium]

Definition(s)

Digital Image Processing is the processing of information contained in a digital image.[2]
Digital Image Processing is the use of a digital computer to process digital images through an algorithm.[3]

[2] [Oxford Dictionary of CS, 2016]
[3] [Gonzalez & Woods, 2018]
[4] [Geeks4Geeks, 2022]
[5] [Medium]

Definition(s)

Digital Image Processing is the processing of information contained in a digital image.[2]
Digital Image Processing is the use of a digital computer to process digital images through an algorithm.[3]
Digital Image Processing is the use of algorithms and mathematical models to process and analyze digital images.

[2] [Oxford Dictionary of CS, 2016]
[3] [Gonzalez & Woods, 2018]
[4] [Geeks4Geeks, 2022]
[5] [Medium]

Definition(s)

Digital Image Processing is the processing of information contained in a digital image.[2]
Digital Image Processing is the use of a digital computer to process digital images through an algorithm.[3]
Digital Image Processing is the use of algorithms and mathematical models to process and analyze digital images. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image-based tasks. [4]

[2] [Oxford Dictionary of CS, 2016]
[3] [Gonzalez & Woods, 2018]
[4] [Geeks4Geeks, 2022]
[5] [Medium]

Definition(s)

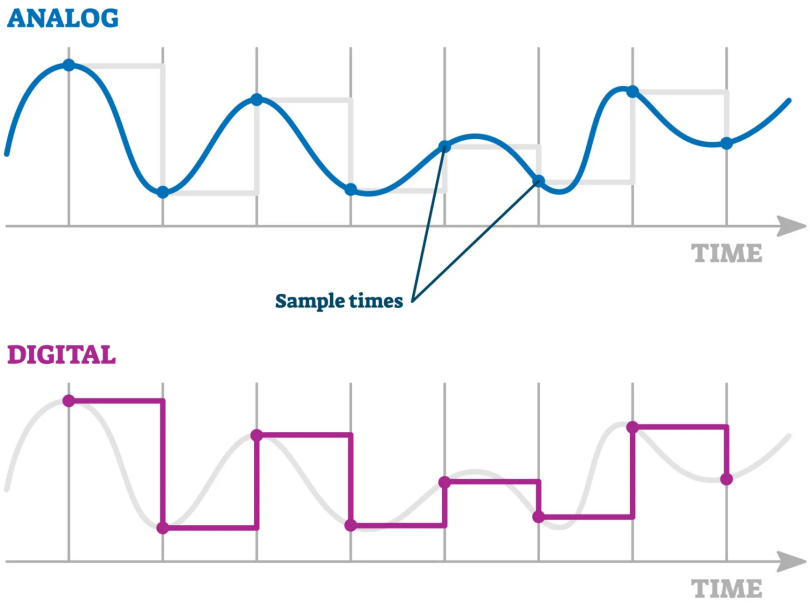
Digital Image Processing is the processing of information contained in a digital image.[2]
Digital Image Processing is the use of a digital computer to process digital images through an algorithm.[3]
Digital Image Processing is the use of algorithms and mathematical models to process and analyze digital images. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image-based tasks. [4]
Digital image processing is the use of computer algorithms to perform operations on digital images. [5]

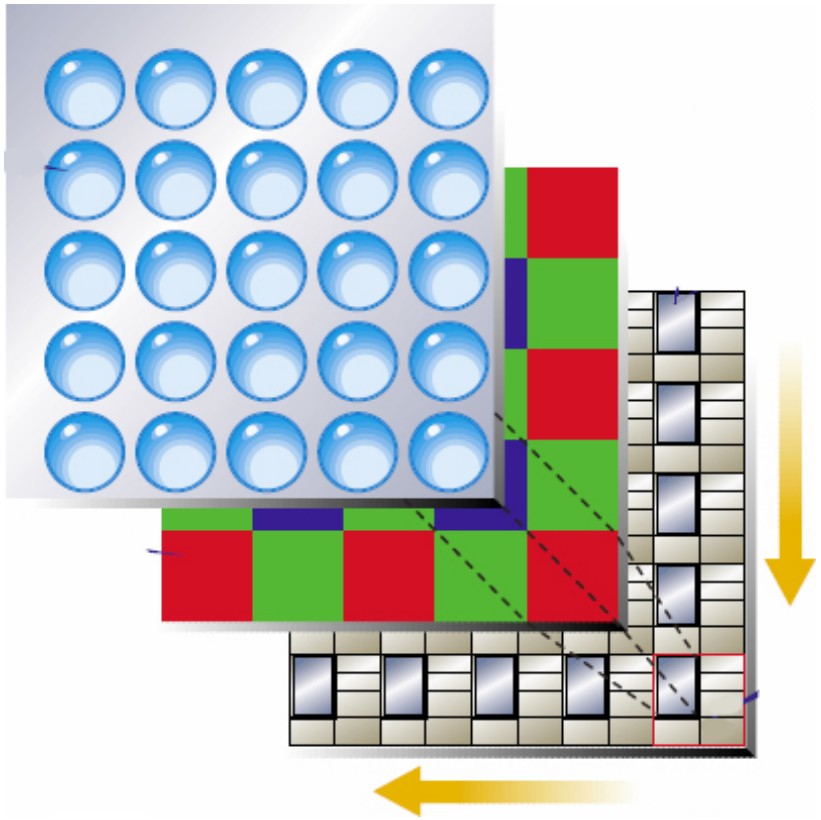
[2] [Oxford Dictionary of CS, 2016]
[3] [Gonzalez & Woods, 2018]
[4] [Geeks4Geeks, 2022]
[5] [Medium]

Definition(s)

Digital Image Processing is the processing of information contained in a **digital image**.
Digital Image Processing is the use of a digital computer to process **digital image** through an algorithm.
Digital Image Processing is the use of algorithms and mathematical models to process and analyze **digital images**. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image-based tasks.
Digital image processing is the use of computer algorithms to perform operations on **digital images**.

Definition(s)





Definition(s)

Digital Image Processing is the processing of information contained in a digital image.
Digital Image Processing is the use of a digital computer to process digital images through an **algorithm**.
Digital Image Processing is the use of **algorithms and mathematical models** to process and analyze digital images. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image-based tasks.
Digital image processing is the use of **computer algorithms** to perform operations on digital images.

→ course pre-**requirements**

Definition(s)

Digital Image Processing is the **processing** of information contained in a digital image.
Digital Image Processing is the use of a digital computer to **process** digital images through an algorithm.
Digital Image Processing is the use of algorithms and mathematical models to **process and analyze** digital images. The goal of digital image processing is to **enhance the quality of images, extract meaningful information** from images, and automate image-based tasks.
Digital image processing is the use of computer algorithms to perform **operations** on digital images.

Image processing pipeline - components

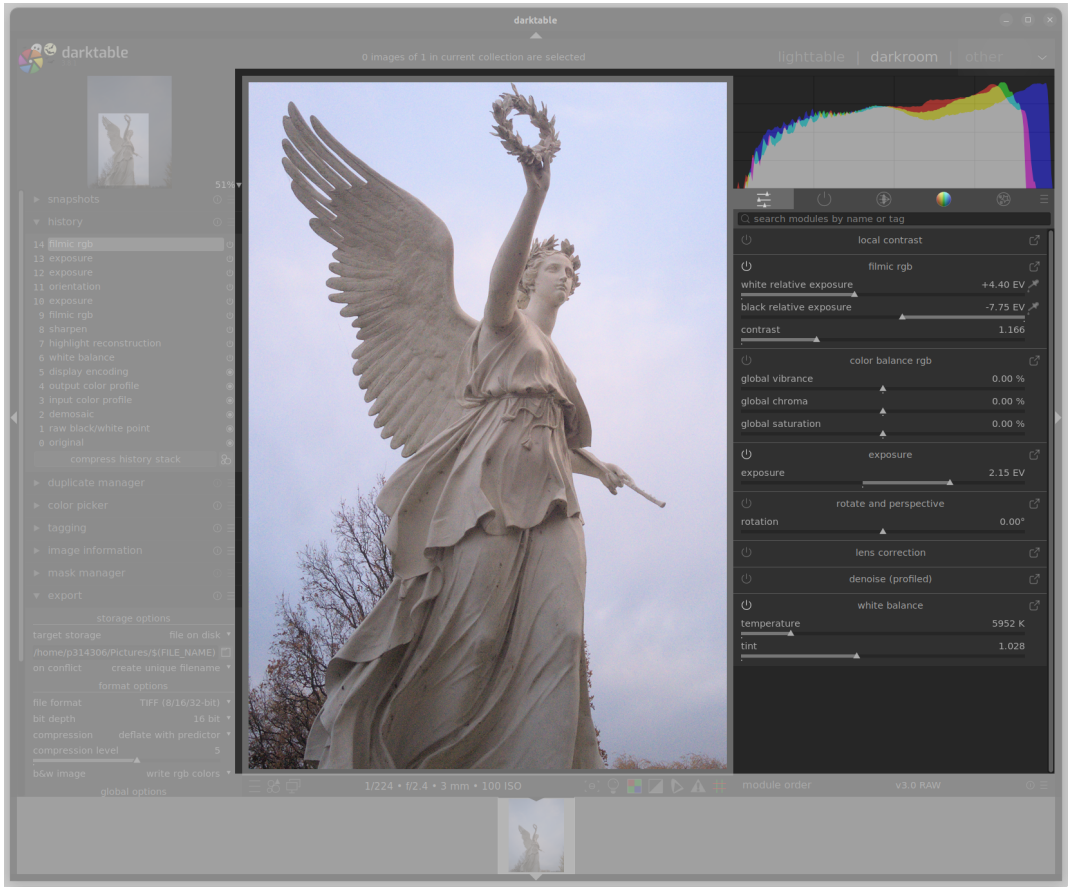
How do we process images ?

Image processing pipeline - components

How do we process images ?

1. *image formation / acquisition*

Image processing pipeline - components



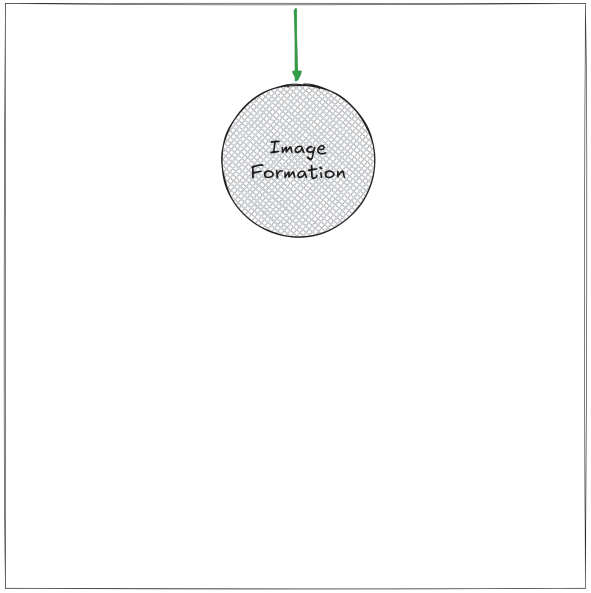


Image processing pipeline - components

How do we process images ?

1. *image formation / acquisition*
2. *(appearance) filtering*: image enhancement & noise suppression

Image processing pipeline - components

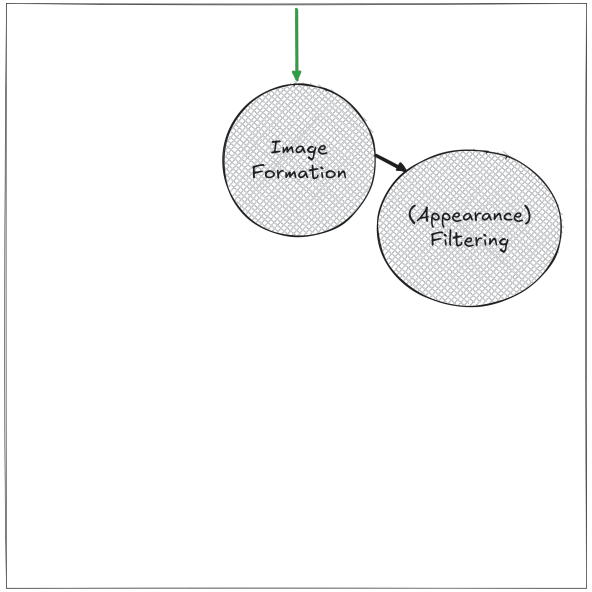
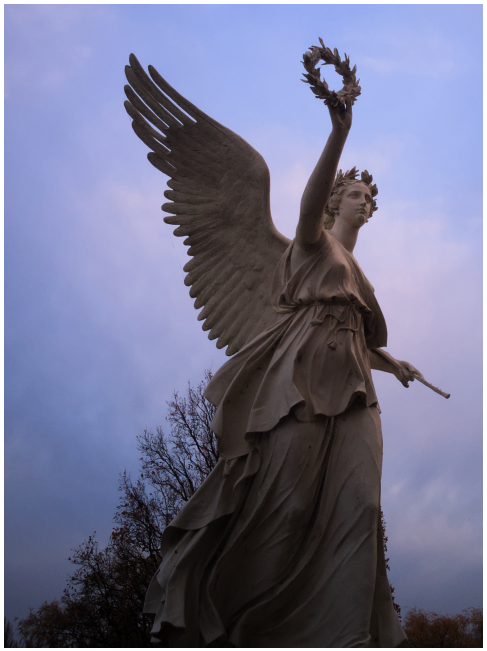


Image processing pipeline - components

How do we process images ?

1. *image formation / acquisition*
2. *(appearance) filtering*: image enhancement & noise suppression
3. *structure filtering & labelling*: filtering *spatial structures*, e.g. edges, corners, connected components

Image processing pipeline - components



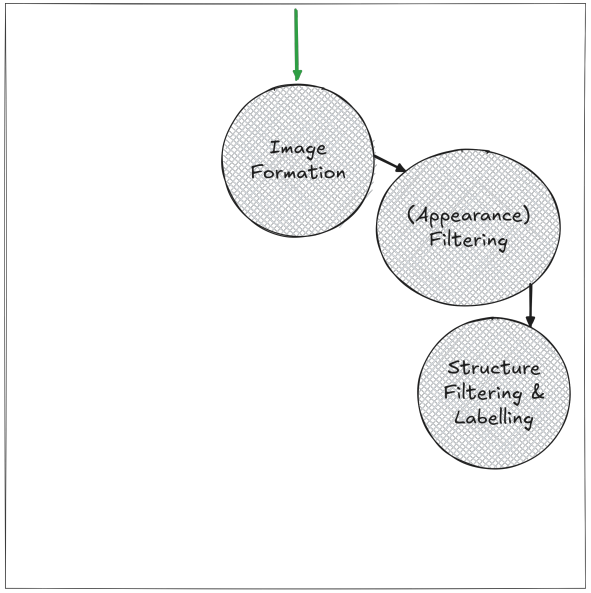
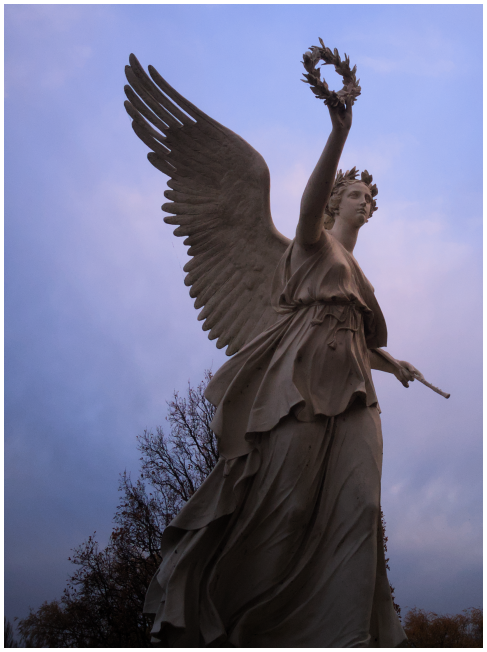


Image processing pipeline - components

How do we process images ?

1. *image formation / acquisition*
2. *(appearance) filtering*: image enhancement & noise suppression
3. *structure filtering & labelling*: filtering *spatial structures*, e.g. edges, corners, connected components
4. *segmentation*: linking similar pixels into pixel *sets*

Image processing pipeline - components



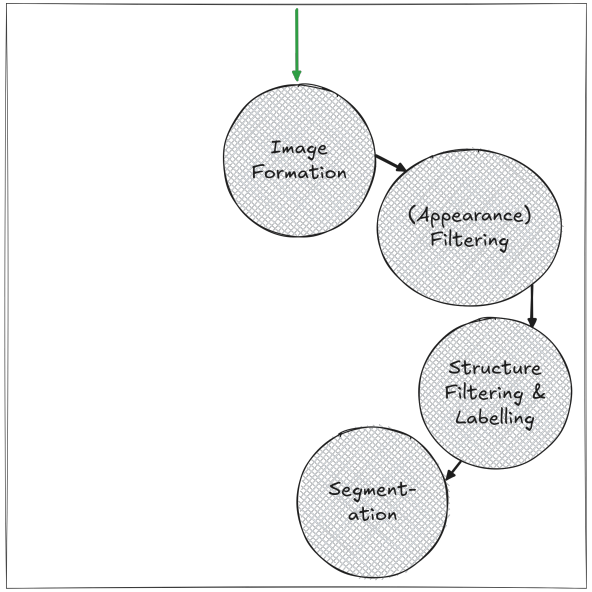
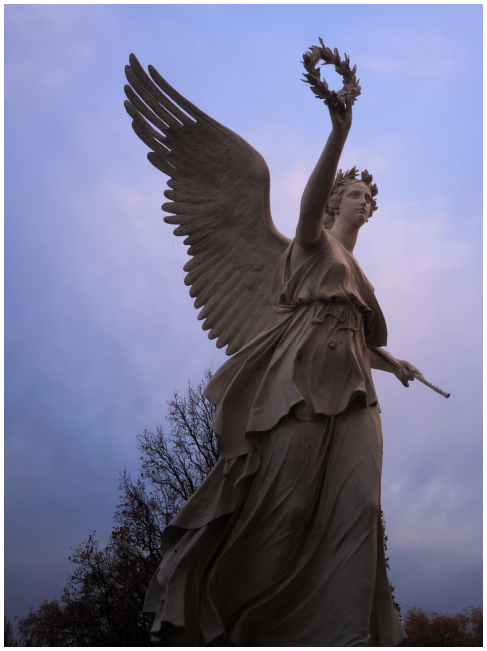


Image processing pipeline - components

How do we process images ?

1. *image formation / acquisition*
2. *(appearance) filtering*: image enhancement & noise suppression
3. *structure filtering & labelling*: filtering *spatial structures*, e.g. edges, corners, connected components
4. *segmentation*: linking similar pixels into pixel *sets*
5. *feature extraction*: area, center of gravity, number of holes, curvature, relation to other groups, etc.

Image processing pipeline - components



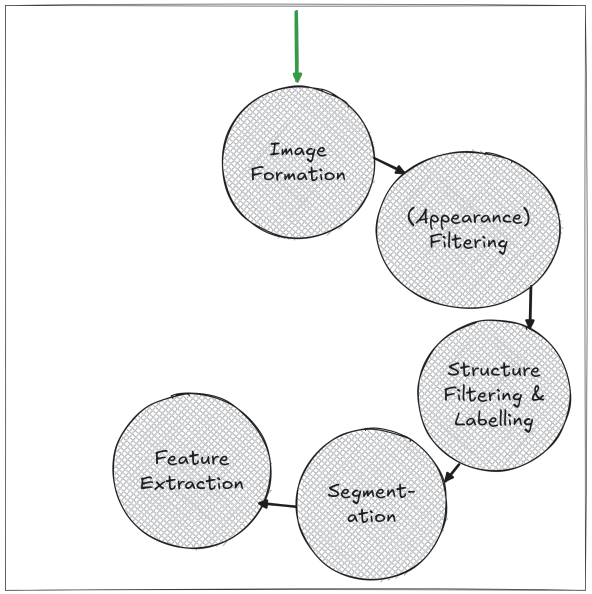
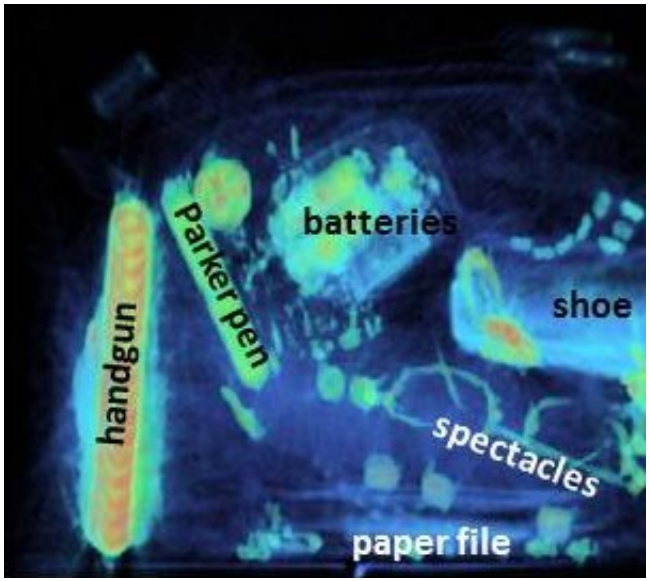


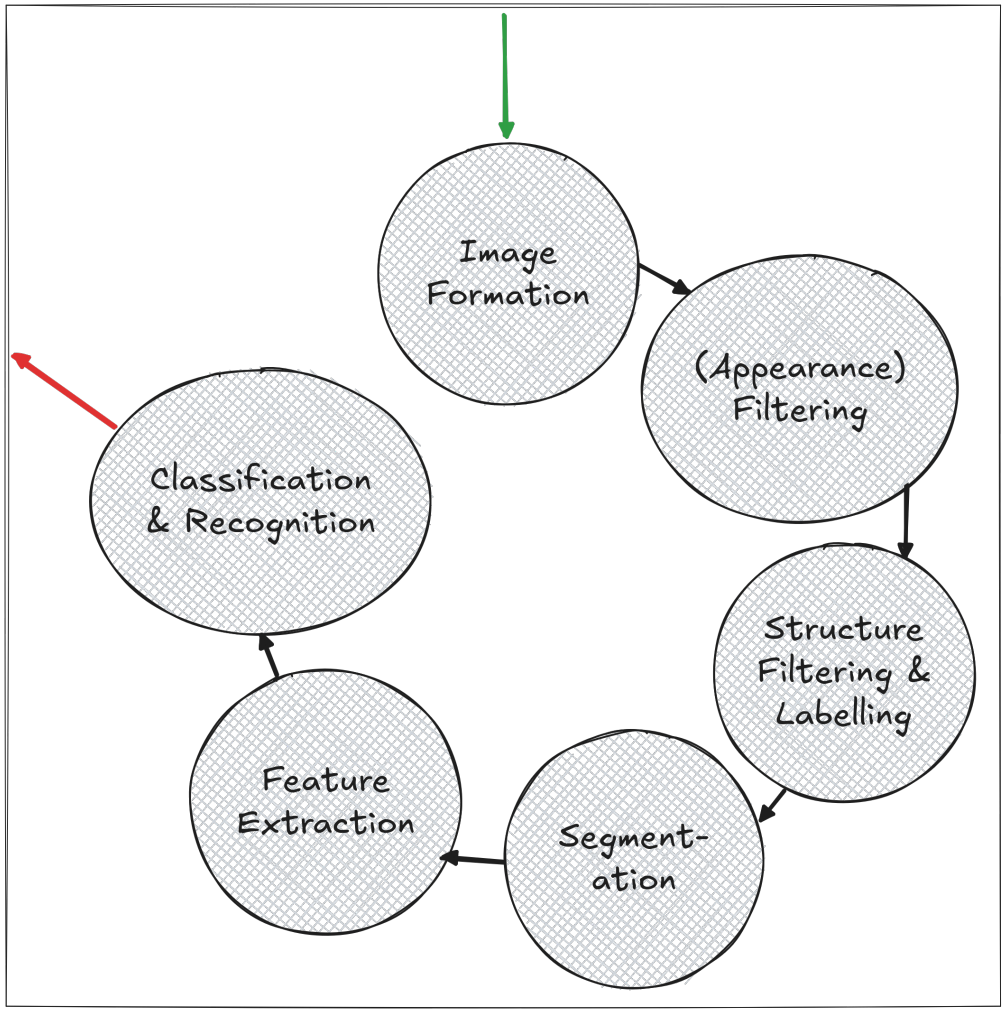
Image processing pipeline - components

How do we process images ?

1. *image formation / acquisition*
2. *(appearance) filtering*: image enhancement & noise suppression
3. *structure filtering & labelling*: filtering *spatial structures*, e.g. edges, corners, connected components
4. *segmentation*: linking similar pixels into pixel *sets*
5. *feature extraction*: area, center of gravity, number of holes, curvature, relation to other groups, etc.
6. *classification & recognition*: attach (physical) meaning to extracted objects

Image processing pipeline - components





Subfields in visual sciences

	Image OUT	Data OUT
Image IN	image processing	image analysis / computer vision
Data IN	computer graphics, computational imaging	pattern recognition, data science, signal processing

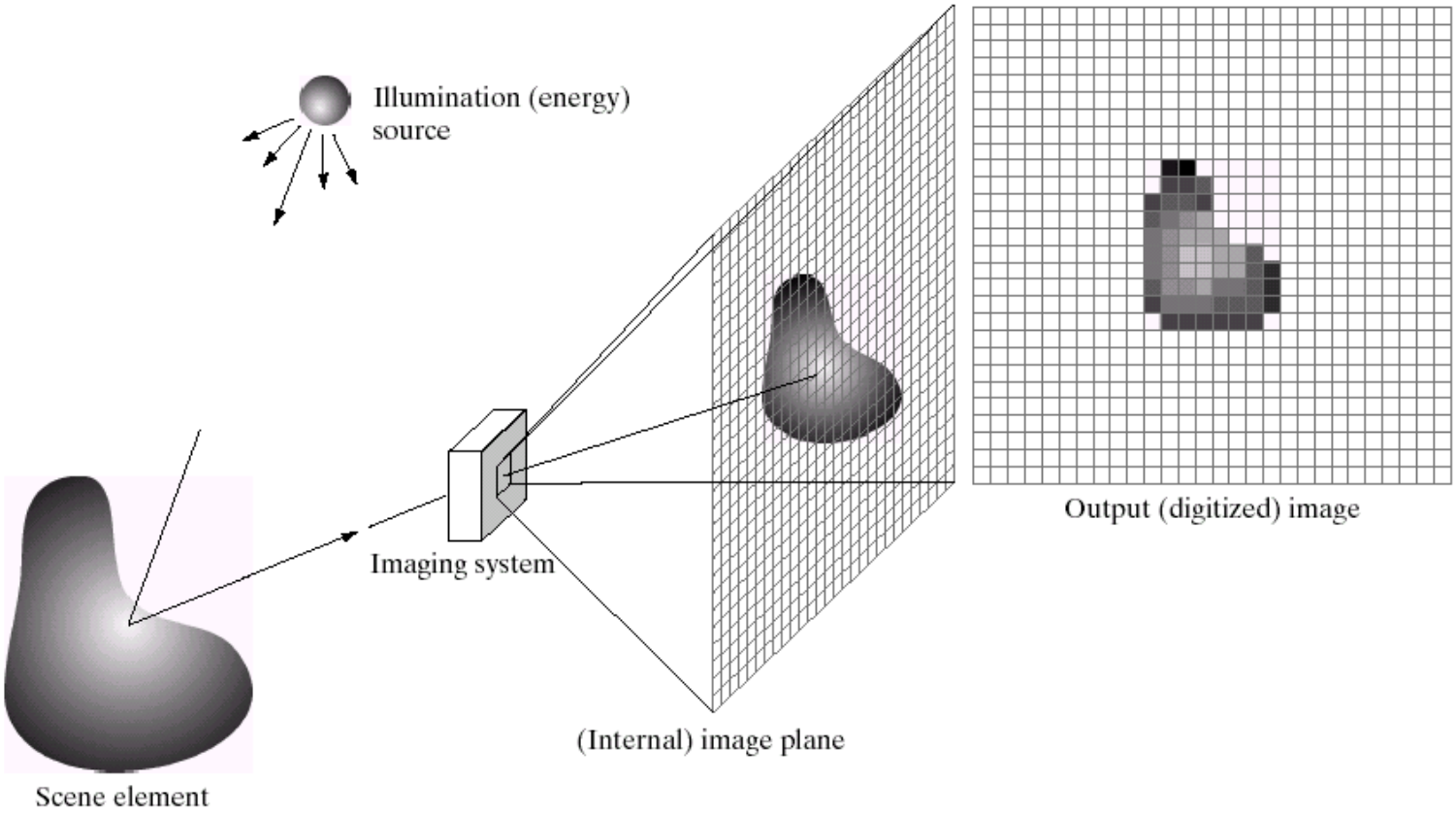
Subfields in visual sciences

	Image OUT	Data OUT
Image IN	image processing	image analysis / computer vision
Data IN	computer graphics, computational imaging	pattern recognition, data science, signal processing

- COMPUTATIONAL IMAGING - Acquisition & Reconstruction of Images from Signals
- IMAGE PROCESSING - Image Transformation & Image Signal Processing
- IMAGE ANALYSIS - (Numeric) Image Information Extraction
- COMPUTER VISION - 3D Scene Analysis, Description & Reconstruction
- COMPUTER GRAPHICS - Image Synthesis / CGI
- (SCIENTIFIC) VISUALIZATION - visual representation of (scientific) data

Fundamentals of Images

Image Acquisition



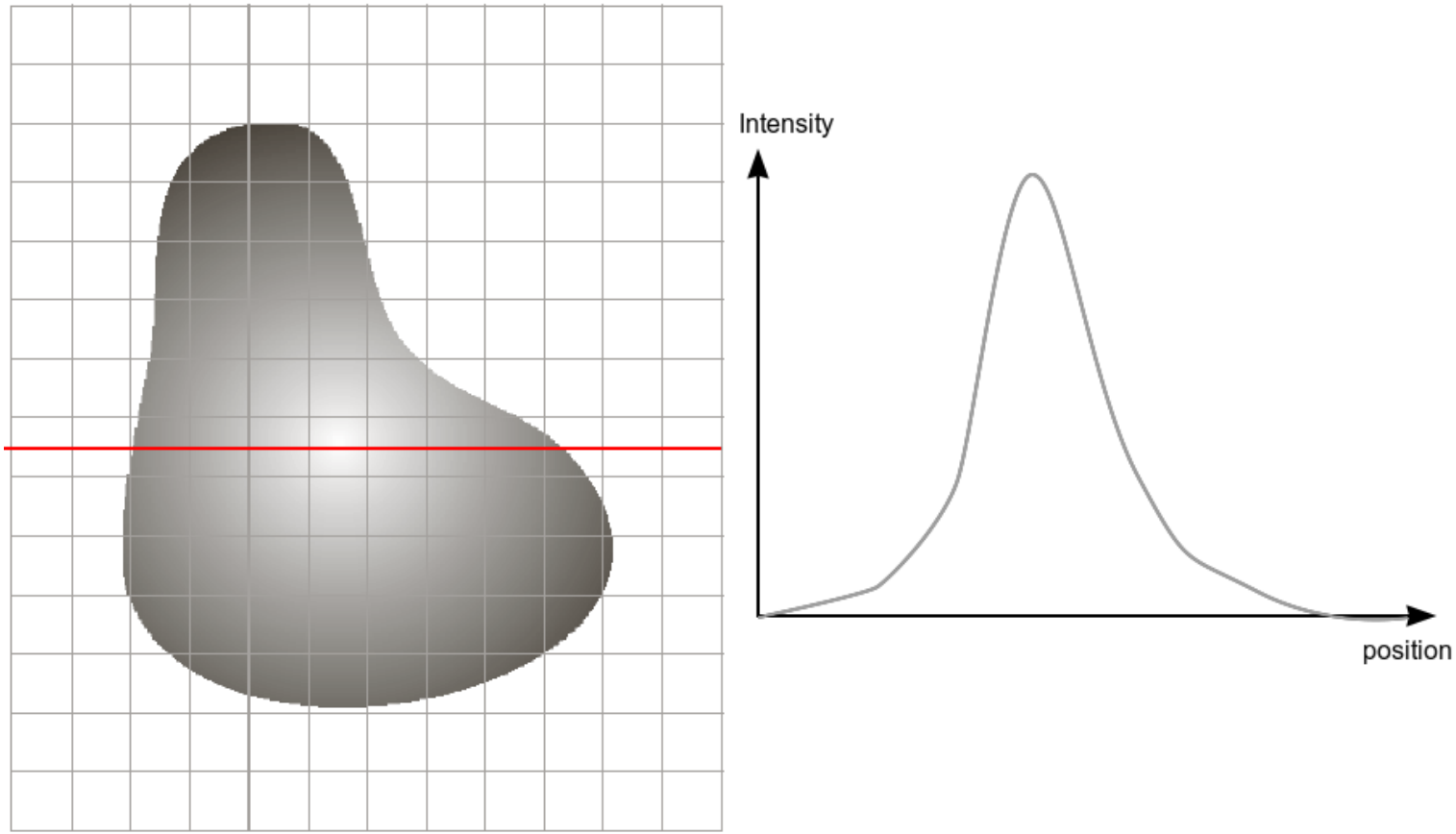
(a) Illumination source. (b) Element of scene. (c) Imaging system. (d) projection on image plane. (d) Digitized image.

Signal- & Information Theory – Fundamentals

Major subtopics:

1. Digitization / Analog-Digital conversion (ADC)
2. Sampling & grid resolution
3. Quantization & value resolution
4. Mathmematical representation

Coninuous analog signal



Continuous image & position-intensity plot (along red line).

Image sampling (spatial discretization)

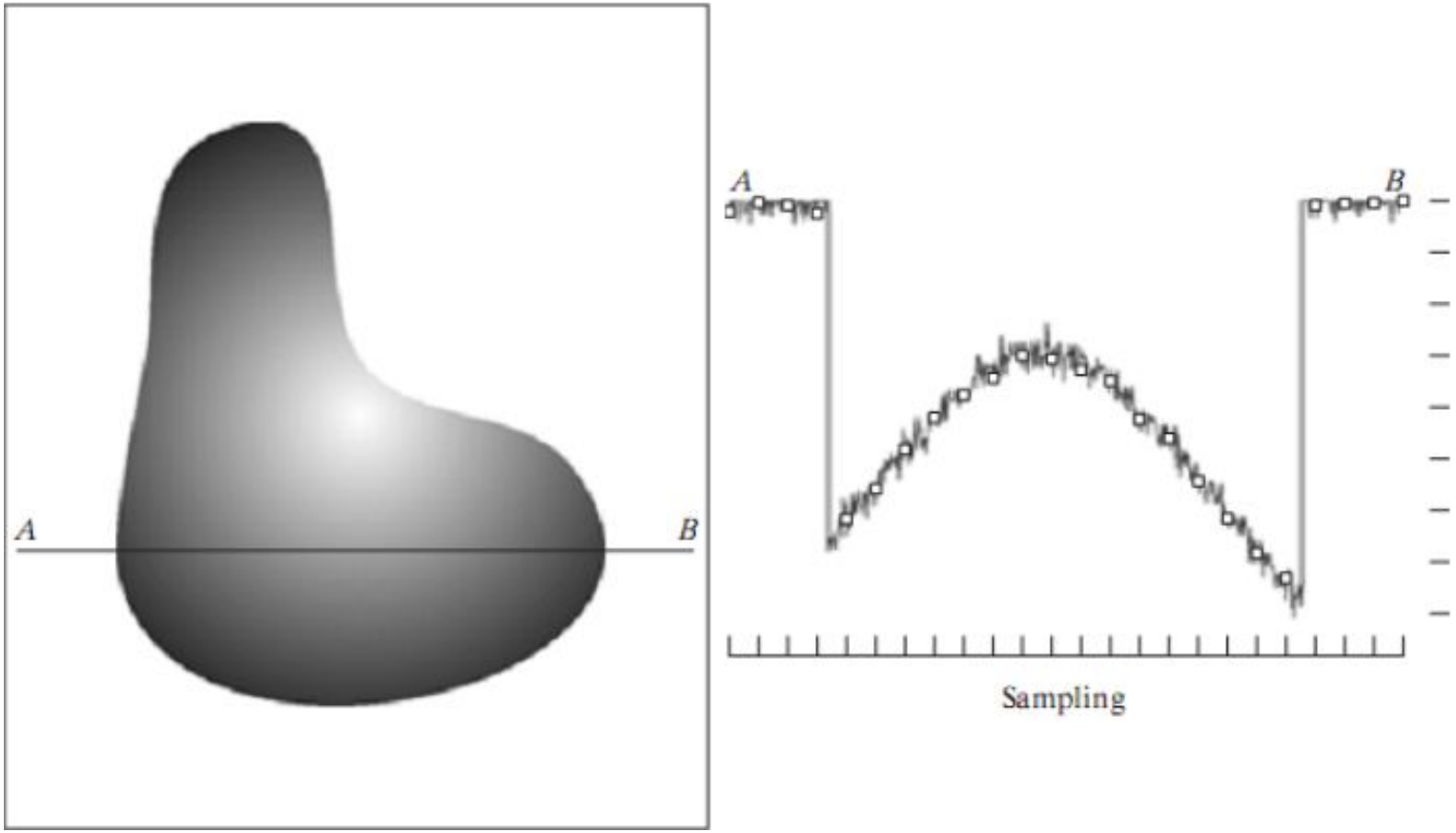
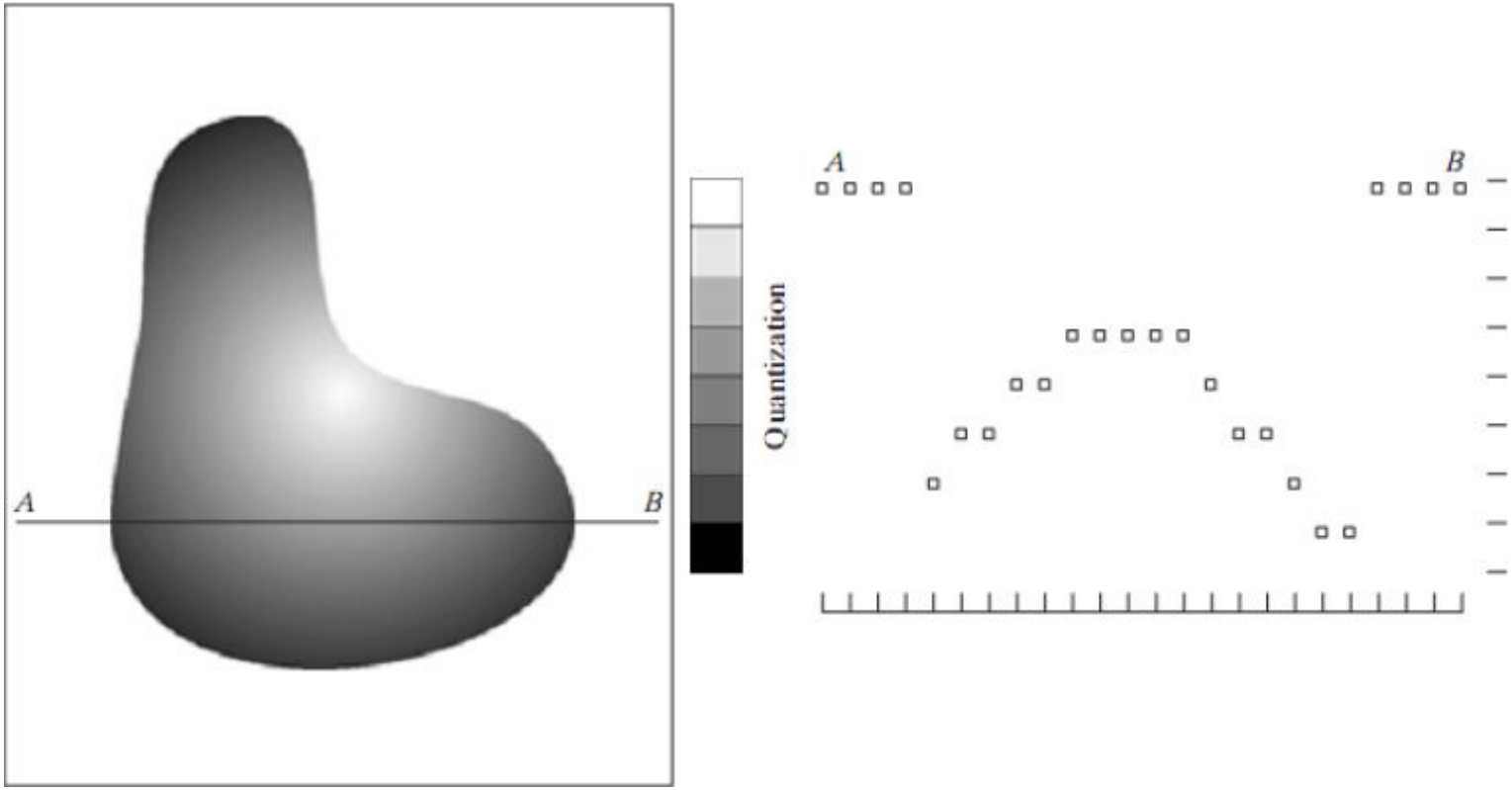


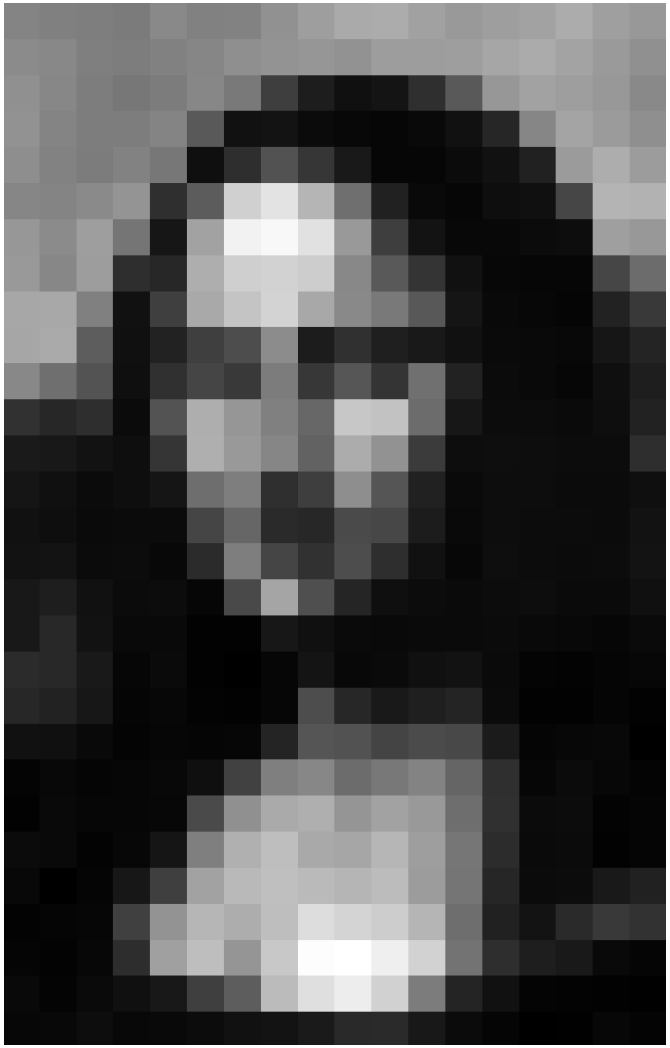
Image sampled on sensor array & position-intensity plot (along red line).

Quantization (value discretization)

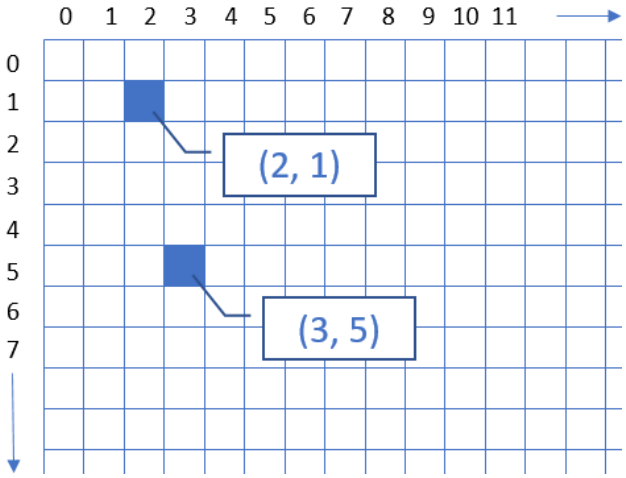


(a) Pixel quantization of sampled image with 4-bit.
Sampled & quantized image: discrete domain, discrete range.

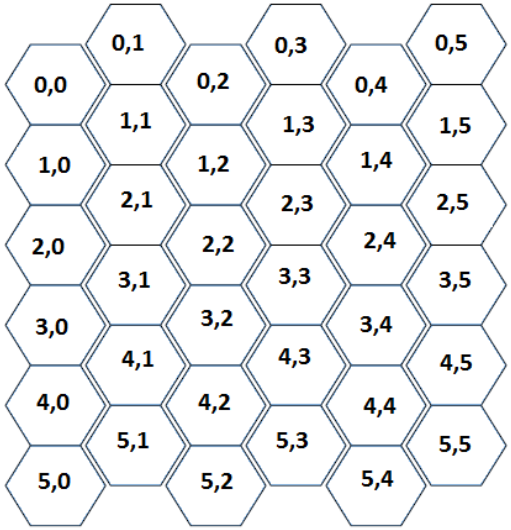
Digital image: discrete domain and range



Discrete domain, discrete range.



Standard rectangular grid



Hexagonal grid

Image representation

- Formal representation: images as a *function* $(x, y) \rightarrow f(x, y)$, where
  - (x, y) - 2D image coordinates (i.e. *domain*)
  - $f(x, y)$ - image values

Image representation

- Formal representation: images as a *function* $(x, y) \rightarrow f(x, y)$, where
  - (x, y) - 2D image coordinates (i.e. *domain*)
  - $f(x, y)$ - image values
- (x, y):
  - usually *discrete* ($(x, y) \in \mathbb{Z}$)
  - can be *continuous* ($(x, y) \in \mathbb{R}$)
  - if (x, y) is discrete: coordinates named *pixels*
- $f(x, y)$

Image representation

- Formal representation: images as a *function* $(x, y) \rightarrow f(x, y)$, where
  - (x, y) - 2D image coordinates (i.e. *domain*)
  - $f(x, y)$ - image values
- (x, y)
- $f(x, y)$:
  - $f(x, y) \in \mathbb{Z}^1$ (i.e. *scalar* or *monochromatic*) $\rightarrow$ discrete
  - $f(x, y) \in \mathbb{Z}^3$ (i.e. *coloured* or *trichromatic*) $\rightarrow$ discrete

Image representation

- Formal representation: images as a *function* $(x, y) \rightarrow f(x, y)$, where
  - (x, y) - 2D image coordinates (i.e. *domain*)
  - $f(x, y)$ - image values
- (x, y)
- $f(x, y)$:
  - $f(x, y) \in \mathbb{Z}^1$ (i.e. *scalar* or *monochromatic*) $\rightarrow$ discrete
  - $f(x, y) \in \mathbb{Z}^3$ (i.e. *coloured* or *trichromatic*) $\rightarrow$ discrete
  - general: $f(x, y) \in \mathbb{Z}^n$, with n = no. *channels*
  - in professional applications: $f(x, y) \in \mathbb{R}^n$ (i.e. continuous value range or *bandwidth*)

Image representation

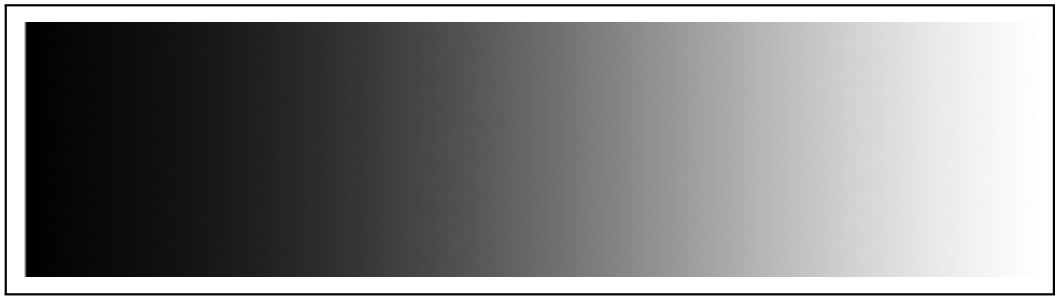
- Formal representation: images as a *function* $(x, y) \rightarrow f(x, y)$, where
  - (x, y) - 2D image coordinates (i.e. *domain*)
  - $f(x, y)$ - image values
- (x, y)
- $f(x, y)$:
  - $f(x, y) \in \mathbb{Z}^1$ (i.e. *scalar* or *monochromatic*) $\rightarrow$ discrete
  - $f(x, y) \in \mathbb{Z}^3$ (i.e. *coloured* or *trichromatic*) $\rightarrow$ discrete
  - general: $f(x, y) \in \mathbb{Z}^n$, with n = no. *channels*
  - in professional applications: $f(x, y) \in \mathbb{R}^n$ (i.e. continuous value range or *bandwidth*)
  - quantization $\rightarrow$ limited range: $f(x, y) \in \{a..b\}$
  - common: $a = 0; b = 2^B - 1$ with B - bits per channel
  - value range = no. values for $f(x, y)$ = *gray levels*

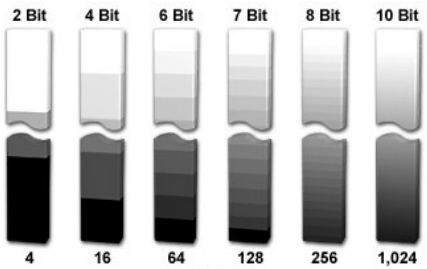
Image representation

- Formal representation: images as a *function* $(x, y) \rightarrow f(x, y)$, where
  - (x, y) - 2D image coordinates (i.e. *domain*)
  - $f(x, y)$ - image values
- (x, y)
- $f(x, y)$:
  - $f(x, y) \in \mathbb{Z}^1$ (i.e. *scalar* or *monochromatic*) $\rightarrow$ discrete
  - $f(x, y) \in \mathbb{Z}^3$ (i.e. *coloured* or *trichromatic*) $\rightarrow$ discrete
  - general: $f(x, y) \in \mathbb{Z}^n$, with n = no. *channels*
  - in professional applications: $f(x, y) \in \mathbb{R}^n$ (i.e. continuous value range or *bandwidth*)
  - quantization $\rightarrow$ limited range: $f(x, y) \in \{a..b\}$
  - common: $a = 0; b = 2^B - 1$ with B - bits per channel
  - value range = no. values for $f(x, y)$ = *gray levels*
  - semantic meaning / *interpretation*: application-dependent

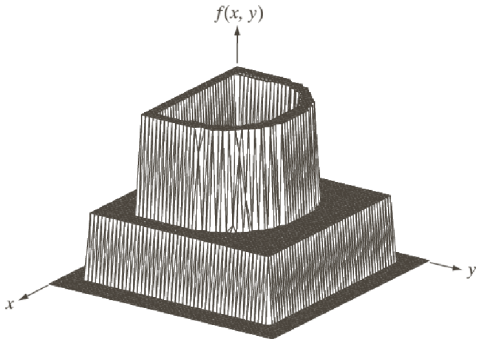
Image representation

- 8-bit image → gray levels {0..255}

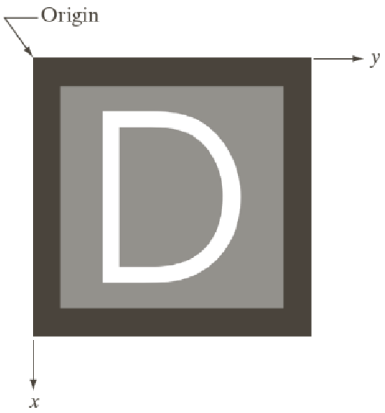




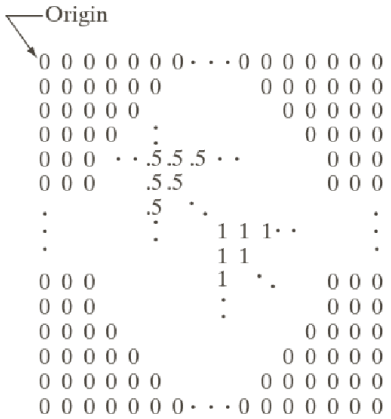
- Depictions



(a) Surface



(b) Intensity figure.



(c) Numerical array.

Image representations

Image with M rows and N columns:

$$\mathbf{f} = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix}$$

Element $f(i,j)$ is value of pixel on row i and column j.

Image representations

1	4	7
2	5	8
3	6	9

3×3 image

$$\mathbf{f} = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

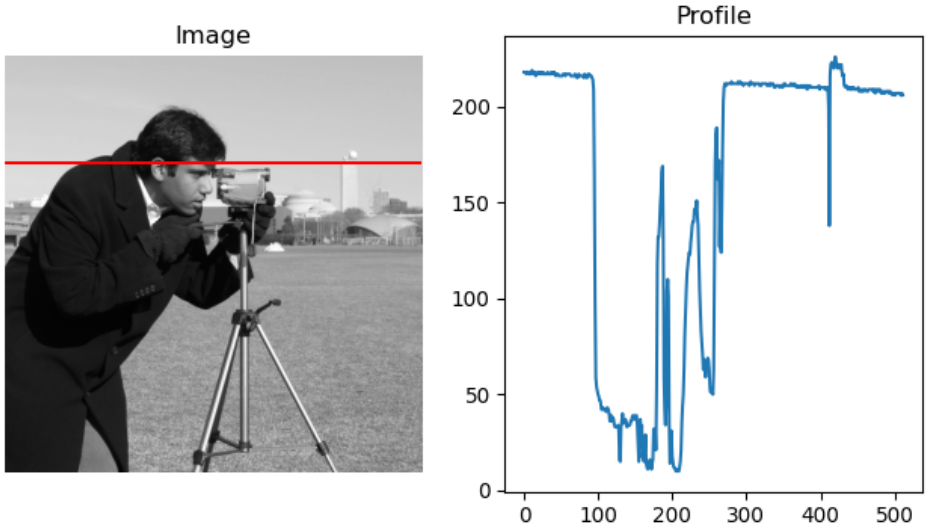
matrix repr.

Resolution

- *Spatial resolution*: no. pixels per unit distance (i.e. *dpi*: dots per inch)
- *Value resolution*: no. bits used for quantization

Fundamentals: Image Characterization & Measuring Images

Profile

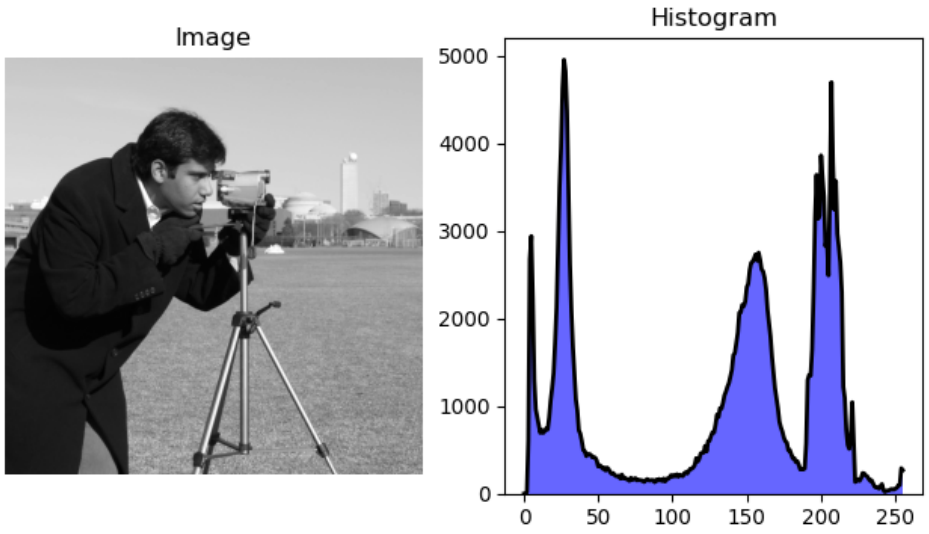


(Line) Profile (i.e. scanline)

→ distribution of values, measured along (horizontal) line

use: illustration of 1D processing concepts

Histogram

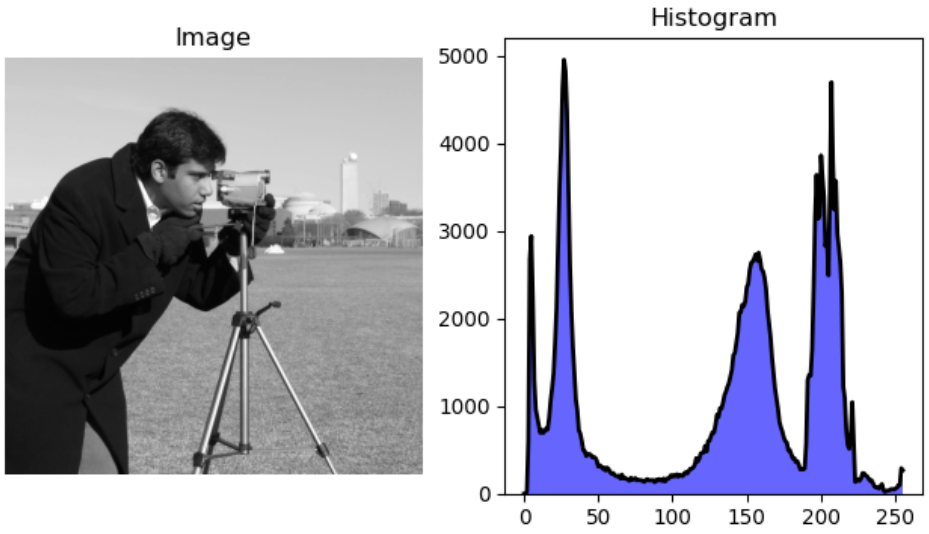


sometimes needed: statistical, global intensity distribution

Histogram

→ measures occurence of values within image

Histogram



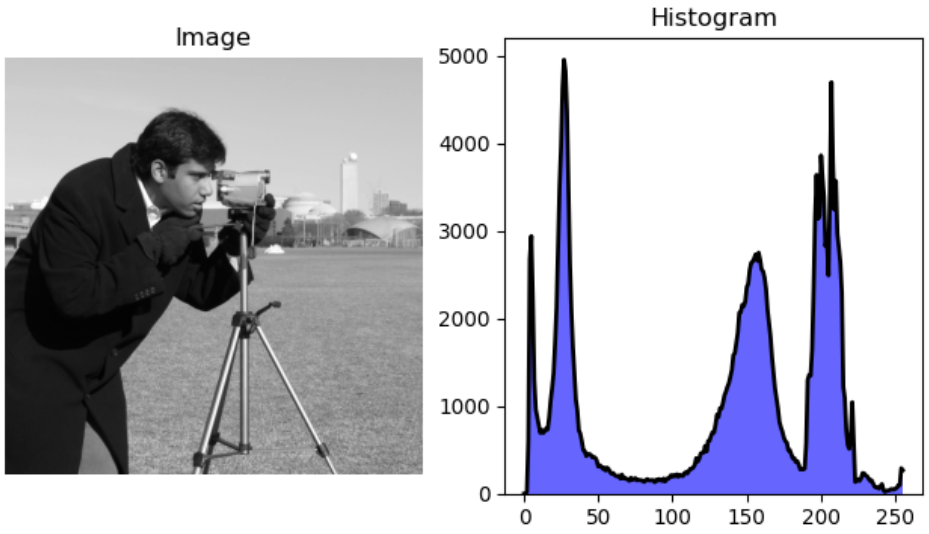
sometimes needed: statistical, global intensity distribution

Histogram

→ measures occurence of values within image

basis to derive *probability distribution function (pdf)*

Histogram



sometimes needed: statistical, global intensity distribution

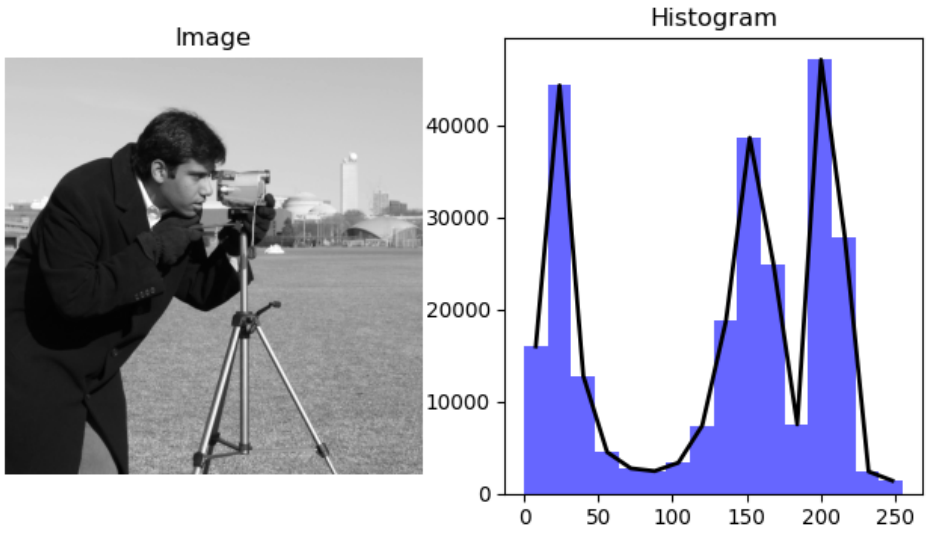
Histogram

→ measures occurence of values within image

basis to derive *probability distribution function (pdf)*

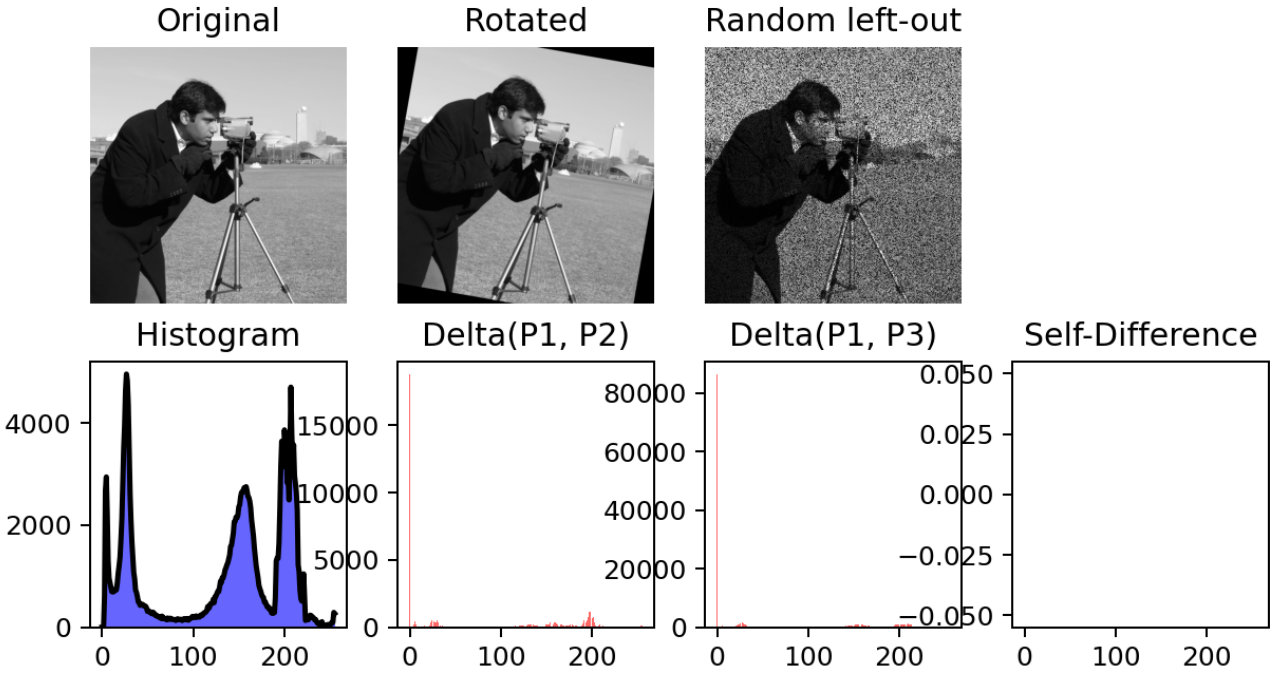
→ quantifies *how likely any given pixel* $x \in X$ *assumes value* y *in image* I

Histogram



Histogram $P(I, b)$ has binning parameter
binning: *in how many value categories to you want to measure the distribution ?*

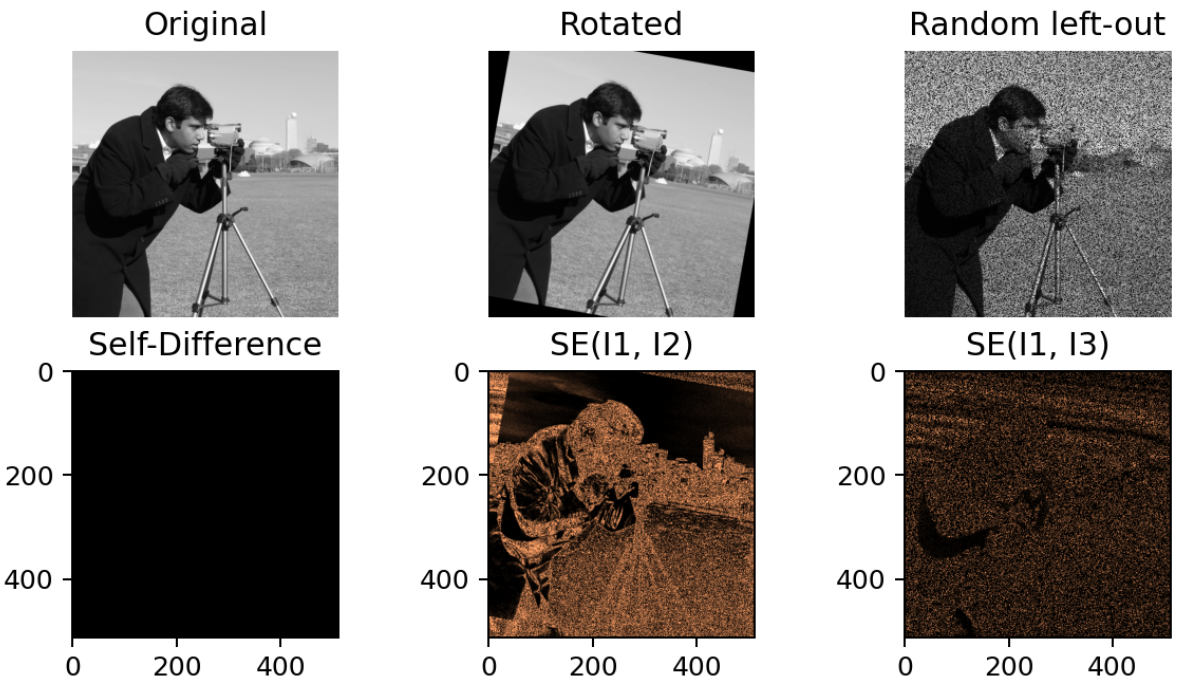
Comparing images - histogram difference



simple comparison approch: *histogram difference*

$\Delta(\mathrm{I}_1, \mathrm{I}_2) = |\Delta(P(\mathrm{I}_1), P(\mathrm{I}_2))|$

Comparing images - (root) mean square error



still simple: *(root) mean square error*
$RSME(\mathrm{I}_1, \mathrm{I}_2) = \sqrt{(\mathrm{I}_1 - \mathrm{I}_2)^2}$ here, $RSME(\mathrm{I}_1, \mathrm{I}_2) = 8.22$ and $RSME(\mathrm{I}_1, \mathrm{I}_3) = 5.99$
Advantage: global metric + per-pixel Δ

Comparing images - cross correlation

Original

Rotated

Random left-out

instead of comparing difference, focus on *similarity* and *correlation*
$\rightarrow$ use *correlation* $(f \star g)(x, y)$:
$(f \star g)(x, y) = \int_{k=0}^{h} \int_{l=0}^{w} \overline{f(k, l)} g(k + x, l + y) dk\ dl$, with (h, w) denoting image *height* and *width*
Note: High values are ideal, low values indicate differences
$corr(\mathrm{I}_1, \mathrm{I}_2) = 0.66$ and $corr(\mathrm{I}_1, \mathrm{I}_3) = 0.58$

Comparing images - structural similarity

Idea behind cross-correlation: pixel-pixel/region-region relationship

[6] detail info: Wang et al. 2004. https://doi.org/10.1109/TIP.2003.819861, section III.C

[7] see eq. 5, 12 &13 in Wang et al. 2004

Comparing images - structural similarity

Idea behind cross-correlation: pixel-pixel/region-region relationship

Structural similarity (SSIM)[6]: luminance l, contrast c & structure s

$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$[7] for $x = \mathrm{I}_1; y = \mathrm{I}_2$

[6] detail info: Wang et al. 2004. `https://doi.org/10.1109/TIP.2003.819861`, section III.C

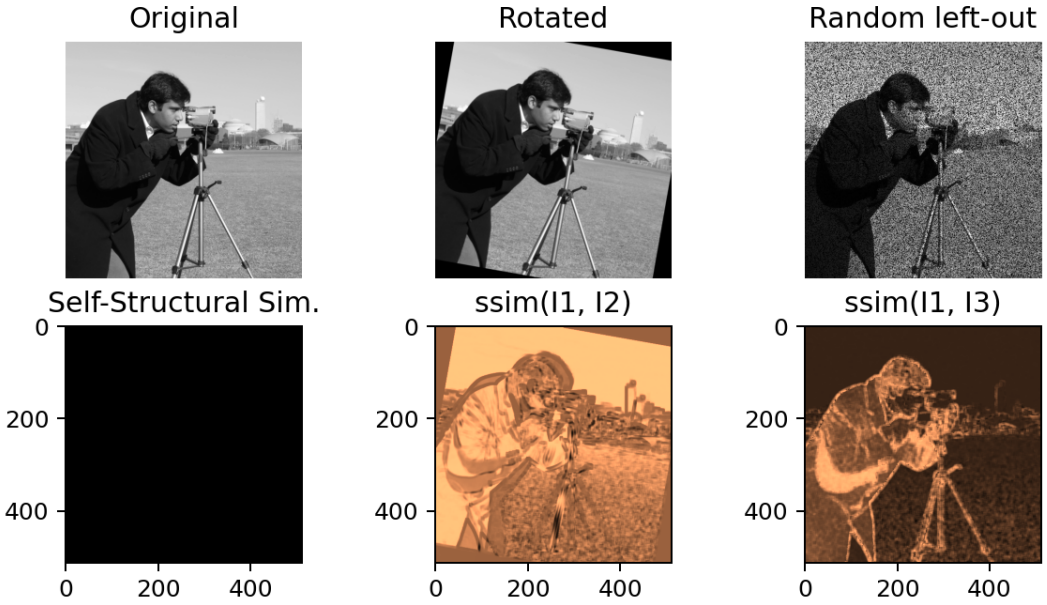
[7] see eq. 5, 12 &13 in Wang et al. 2004

Comparing images - structural similarity

Idea behind cross-correlation: pixel-pixel/region-region relationship

Structural similarity (SSIM)[6]: luminance l, contrast c & structure s

$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$[7] for $x = \mathrm{I}_1; y = \mathrm{I}_2$



$SSIM(\mathrm{I}_1, \mathrm{I}_2) = 0.42$; $SSIM(\mathrm{I}_1, \mathrm{I}_3) = 0.17$

[6] detail info: Wang et al. 2004. `https://doi.org/10.1109/TIP.2003.819861`, section III.C

[7] see eq. 5, 12 &13 in Wang et al. 2004

Comparing images - Joint histogram

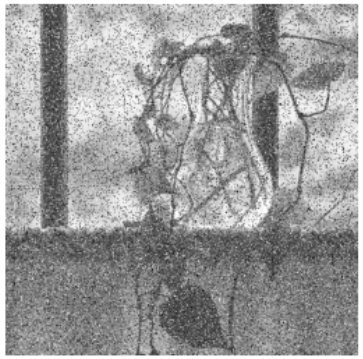
Joint (2D) histogram: information shared between two images

$P(\mathrm{I}_1, \mathrm{I}_2) = P(\mathrm{I}_1 = k | \mathrm{I}_2 = l) \qquad \forall \{k, l\} \in \{0..255\}$

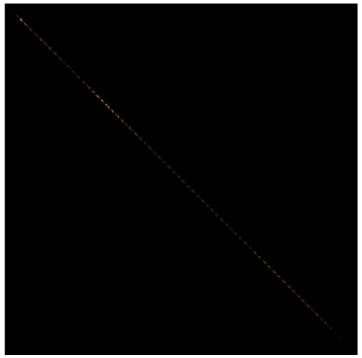
Original

Rotated

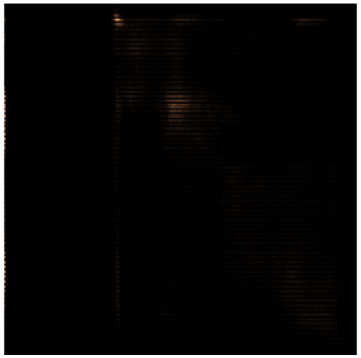
Noise-modulated



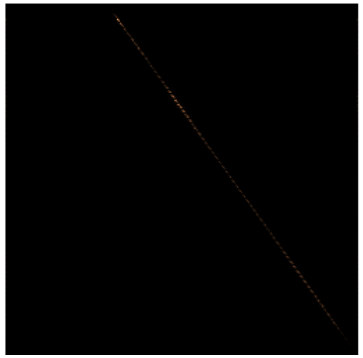
2D Histogram(I1, I1)



2D Histogram(I1, I2)



2D Histogram(I1, I3)



Noise

Different types of noise:

- *(additive white) Gaussian noise (AWGN)*
- Salt-and-Pepper noise
- Grain noise
- Frequency noise

Noise

Different types of noise:

- *(additive white) Gaussian noise (AWGN)*
  - Normal (i.e. Gaussian) distribution over power spectrum
  - is spatially uniformally distributed
- Salt-and-Pepper noise
- Grain noise

Noise

Different types of noise:

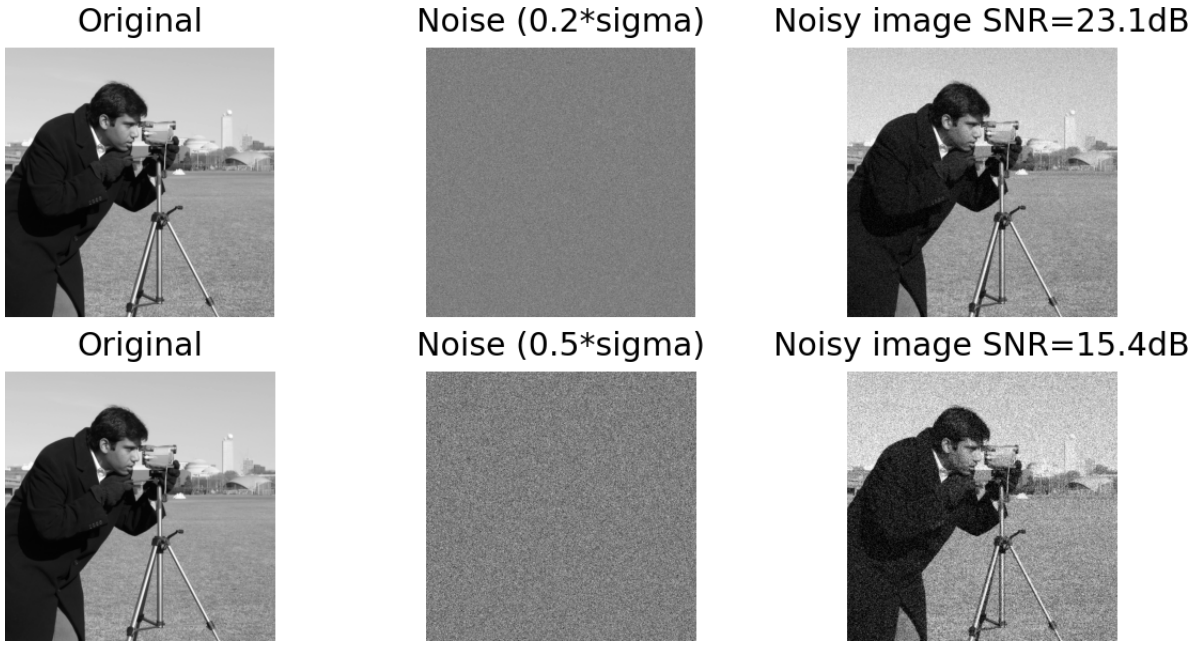
- *(additive white) Gaussian noise (AWGN)*
- Salt-and-Pepper noise
  - Random fully dark-/bright pixels
  - Power spectrum affected by Dirac impulse (i.e. "spike")
  - is spatially uniformally distributed
- Grain noise

Noise

Different types of noise:

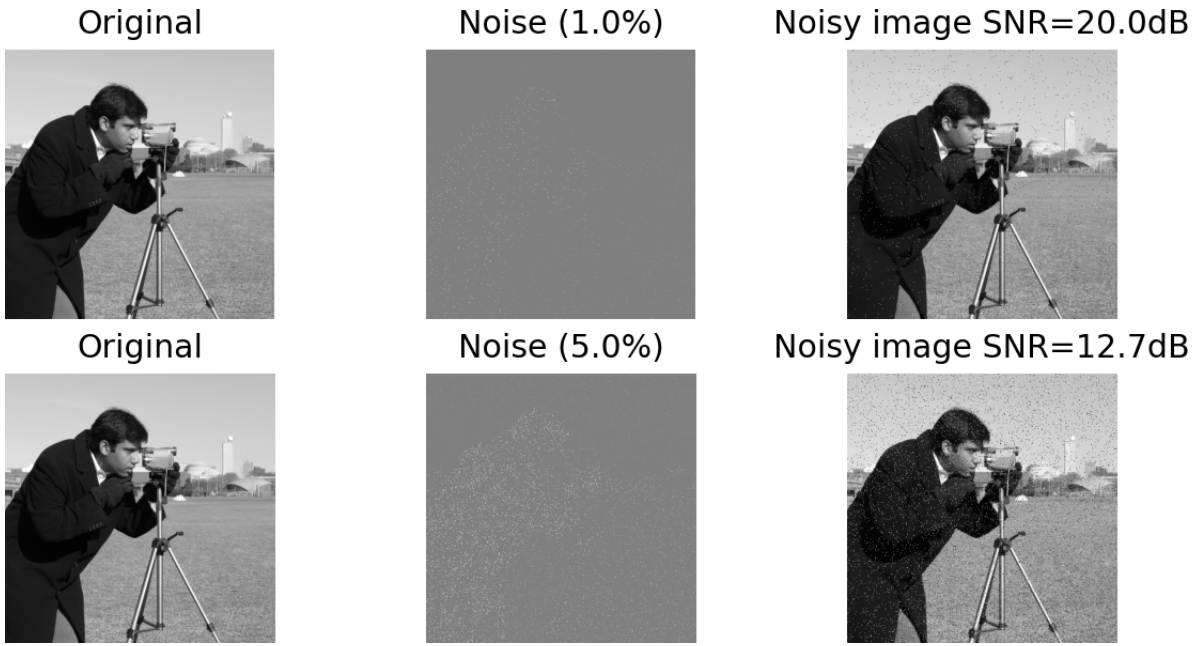
- *(additive white) Gaussian noise (AWGN)*
- Salt-and-Pepper noise
- Grain noise
  - Poisson distribution over power spectrum
  - is spatially binomially distributed

Gaussian noise



- normal-distributed modulation of image values
- distribution parameters: μ and σ
- noise level measured in:
  - *Signal-to-noise ratio (SNR)* [dB]
    $\mathrm{SNR} = 20log_{10}(\frac{P_\mathrm{I}}{P_\mathrm{N}}) = 20log_{10}(\frac{\sigma_\mathrm{I}^2}{\sigma_\mathrm{N}^2})$, with N = noise residual
  - percentage of max. value distortion
  - $k\sigma$, with k = distortion parameter

Salt-and-pepper noise

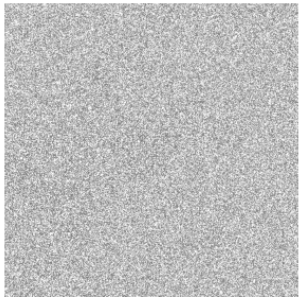


- uniformally-distributed "dead pixels" for x pixels
- noise level measured in:
  - *Signal-to-noise ratio (SNR)* [dB]
    $\mathrm{SNR} = 20log_{10}(\frac{P_\mathrm{I}}{P_\mathrm{N}}) = 20log_{10}(\frac{\sigma_\mathrm{I}^2}{\sigma_\mathrm{N}^2})$, with N = noise residual
  - percentage of dead pixels

Poisson noise

Original

Poisson noise



Noisy image SNR=16.4dB

- Poisson pixel modulation of original image
- *grain noise* effect from old CRT TVs
- noise level measured in:
  - *Signal-to-noise ratio (SNR)* [dB]
    $\mathrm{SNR} = 20log_{10}(\frac{P_\mathrm{I}}{P_\mathrm{N}}) = 20log_{10}(\frac{\sigma_\mathrm{I}^2}{\sigma_\mathrm{N}^2})$, with N = noise residual

Fundamentals: Signal Processing & Sampling

Adapt spatial resolution

(a) 1250 dpi (b) 300 dpi (c) 150 dpi (d) 72 dpi

Adapt spatial discretization

(a) 1250 dpi (b) 300 dpi (c) 150 dpi (d) 72 dpi

Q: *Why is this distinct wave-pattern emerging here ?*

Adapt spatial discretization

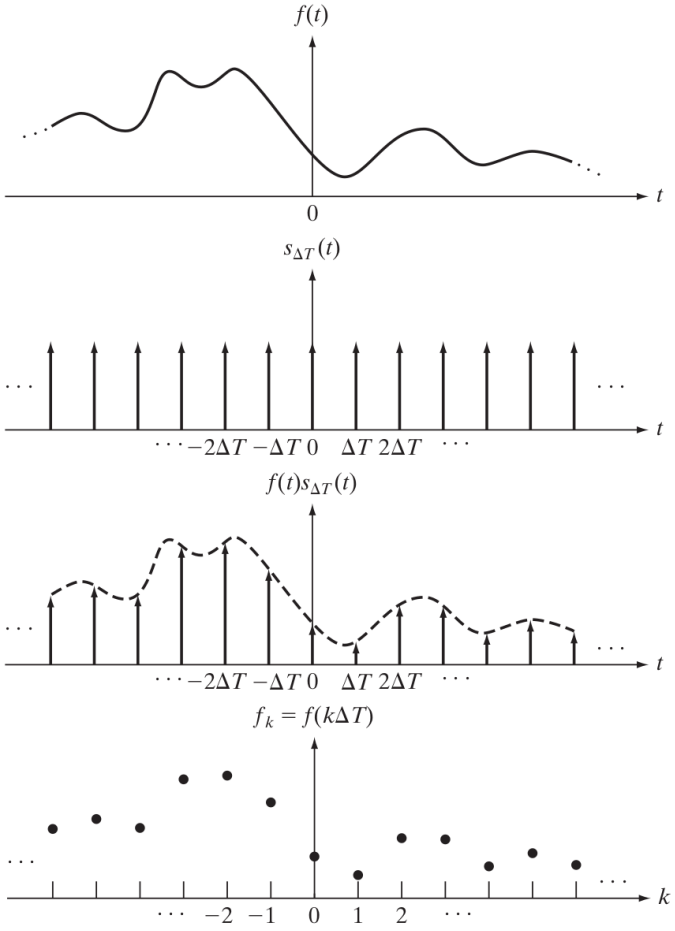
(a) 1250 dpi (b) 300 dpi (c) 150 dpi (d) 72 dpi

Q: *Why is this distinct wave-pattern emerging here ?*

→ **Sampling**

Spatial discretization - Sampling

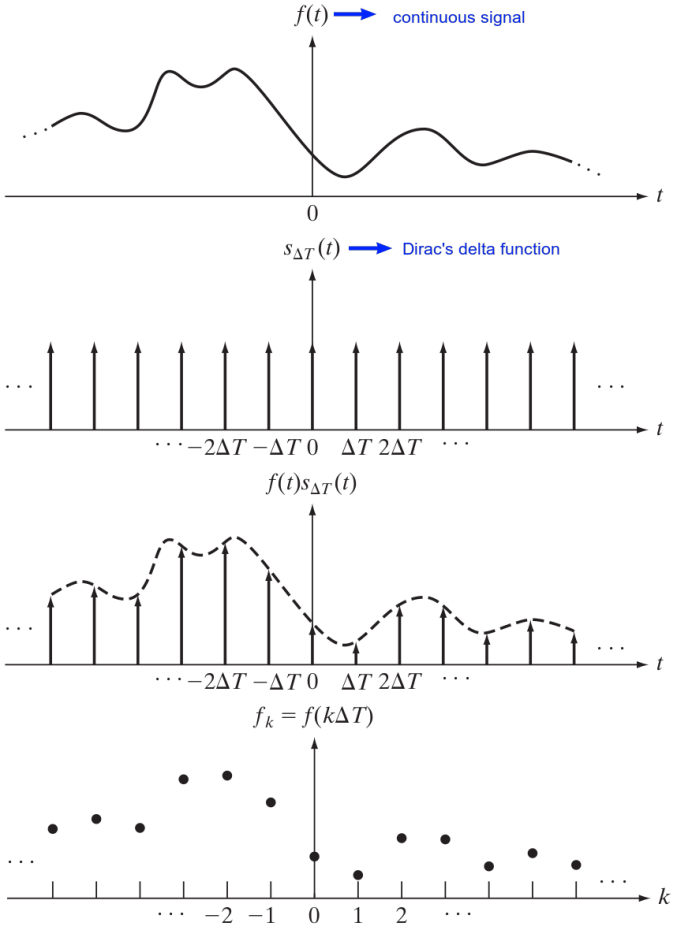
"Sampling converts a *continuous* signal into a *discrete* sequence."[8]



[8] paraphrased from: Shannon, C.E. "Communication in the presence of noise". Proceedings of the Institute of Radio Engineers. 37 (1): 10–21. 1949.

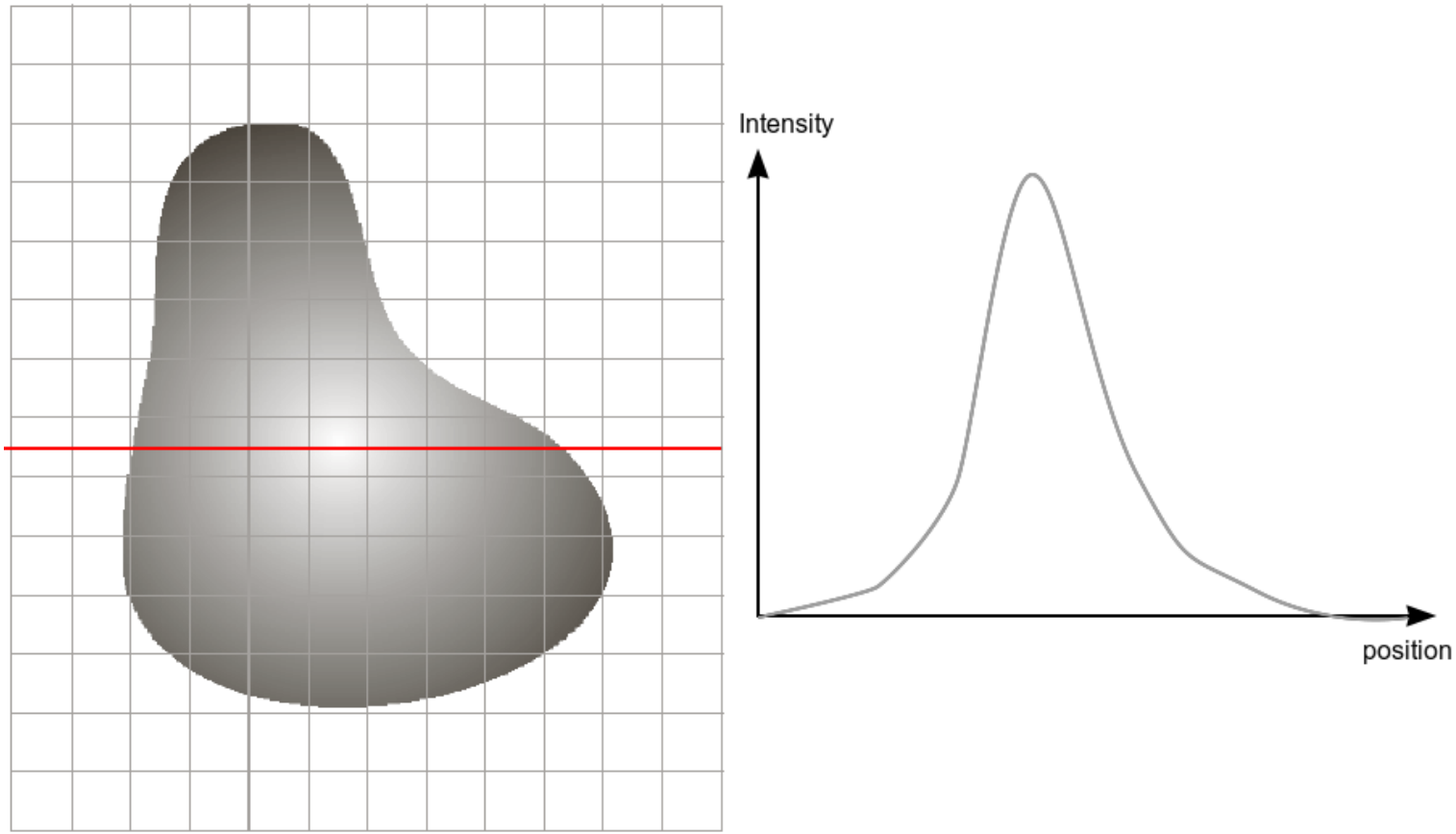
Spatial discretization - Sampling

"Sampling converts a *continuous* signal into a *discrete* sequence."[8]



[8] paraphrased from: Shannon, C.E. "Communication in the presence of noise". Proceedings of the Institute of Radio Engineers. 37 (1): 10–21. 1949.

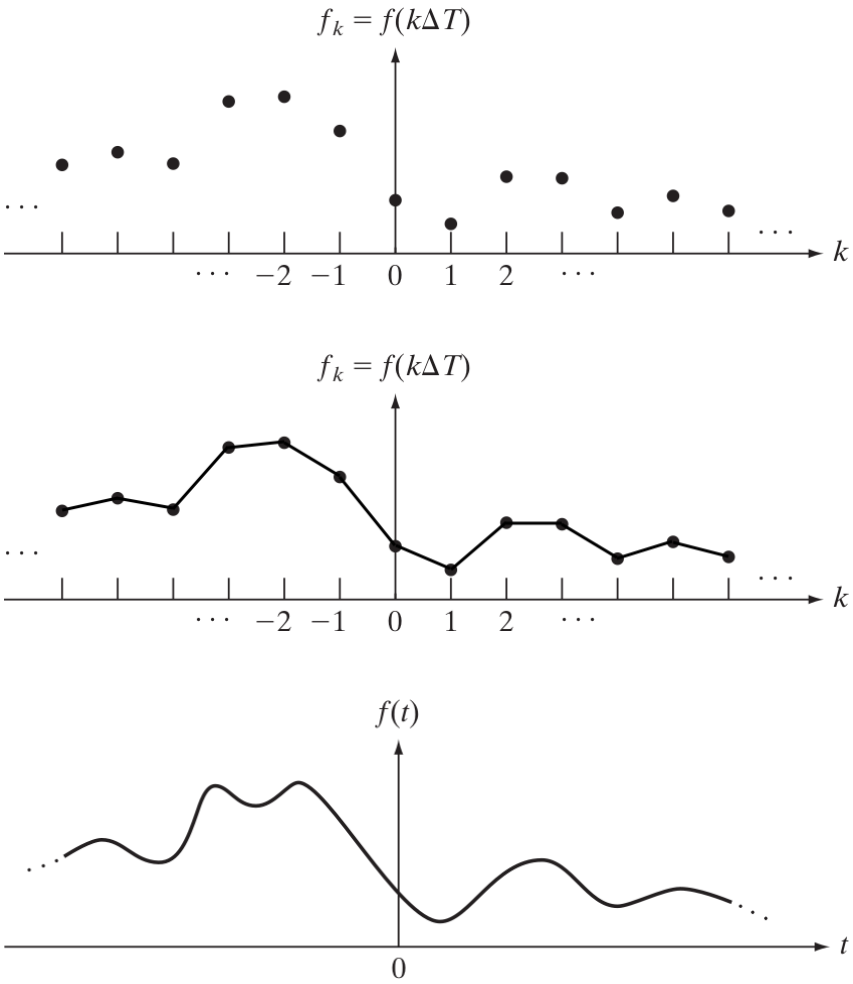
Continuous analog signal



Continuous image & position-intensity plot (along red line).

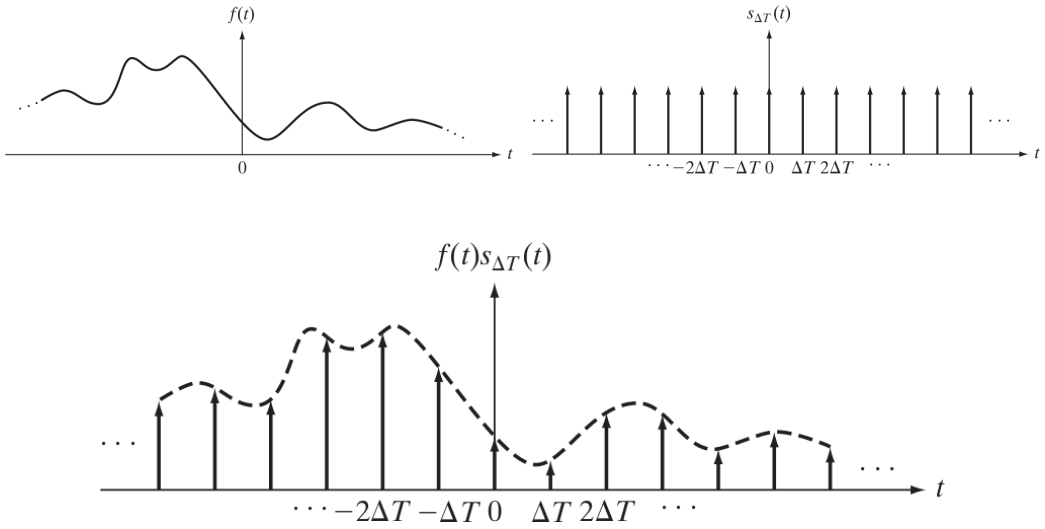
Spatial discretization - Reconstruction

Goal of the sampling: *perfectly* (i.e. lossless) reconstruction of the initial signal.



Signal sampling

How does sampling work ? [9]



$$\tilde{f}(t) = f(t)s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t)\,\delta(t - n\Delta T)$$

assuming sample index $k \in \mathbb{Z} = \{.. - 1, 0, 1, 2, ..\}$, replacing $n = k$:

$$f_k = \int_{-\infty}^{\infty} f(t)\delta(t - k\Delta T)dt = f(k\Delta T) \tag{1}$$

[9] see course book, chapter 4.3.1 "Sampling".

Signal sampling

How do we assess the quality of sampling ?

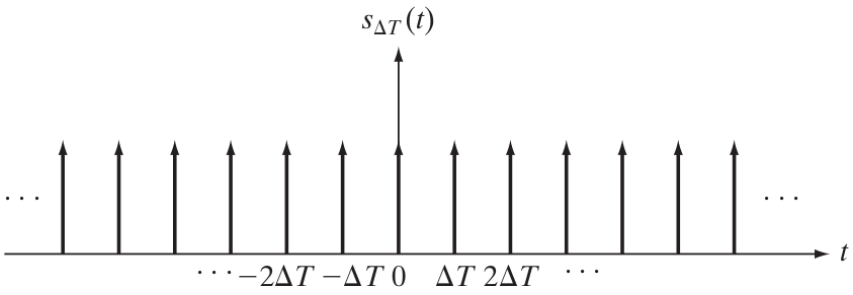


Figure: Sampling function

$\rightarrow$ $s_{\Delta T}$ looks periodic ...

Signal sampling

How do we assess the quality of sampling ?

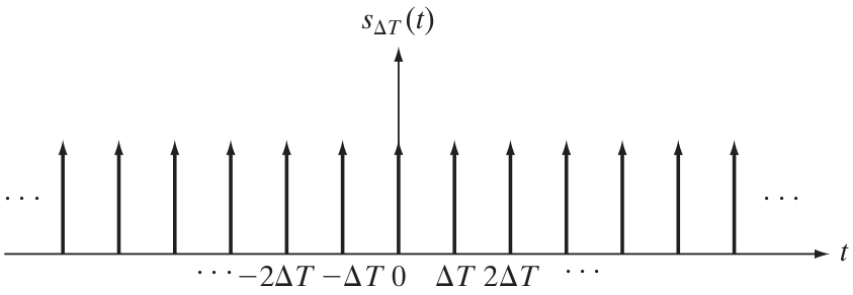


Figure: Sampling function

$\rightarrow$ $s_{\Delta T}$ looks periodic ...
every periodic event has a *frequency* ...

Signal sampling

How do we assess the quality of sampling ?

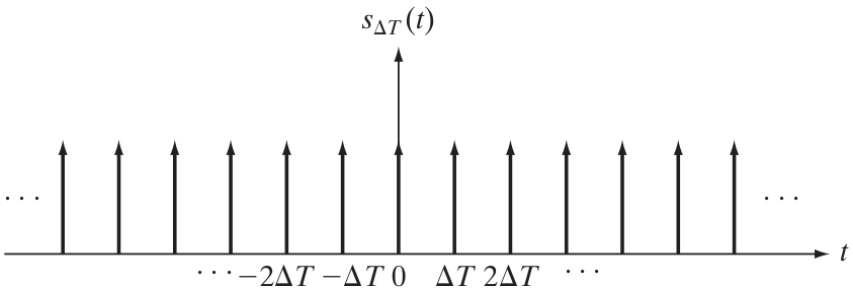


Figure: Sampling function

$\rightarrow$ $s_{\Delta T}$ looks periodic ...
every periodic event has a *frequency* ...
Q: *What is the relation of sampling frequency and signal* $f(x)$ *?*

Signal sampling

How do we assess the quality of sampling ?

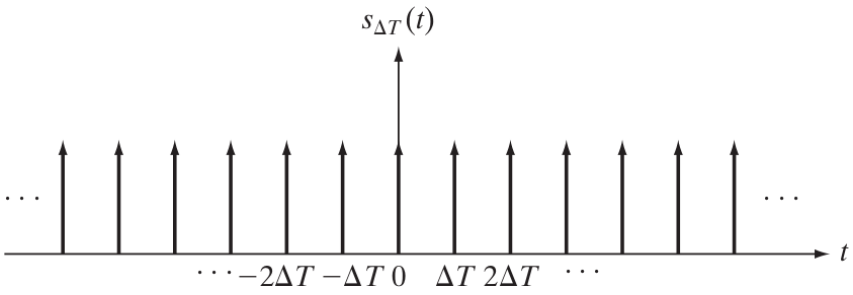


Figure: Sampling function

$\rightarrow$ $s_{\Delta T}$ looks periodic ...
every periodic event has a *frequency* ...
Q: *What is the relation of sampling frequency and signal $f(x)$?*
let's compare the transform $F(\mu)$ of function $f(t)$ in frequency space, with the transform $\tilde{F}(\mu)$ of the sampled function $\tilde{f}(t)$.

Signal sampling

Q: *How to compute the Fourier transform $F(\mu)$ and $\tilde{F}(\mu)$?*

(a) Fourier transform $F(\mu)$ of $f(t)$

$$F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi\mu t}\, dt \qquad (4.2\text{-}16)$$

(b) Fourier transform $S(\mu)$ of $s_{\Delta T}$

$$S(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(\mu - \frac{n}{\Delta T}\right) \qquad (4.3\text{-}4)$$

Fourier transform $\tilde{F}(\mu)$ of $\tilde{f}(t)$:

$$\begin{aligned} \tilde{F}(\mu) &= \Im\{\tilde{f}(t)\} \\ &= \Im\{f(t) s_{\Delta T}(t)\} \\ &= F(\mu) \star S(\mu) \end{aligned} \qquad (4.3\text{-}3)$$

$$\begin{aligned} \tilde{F}(\mu) &= F(\mu) \star S(\mu) \\ &= \int_{-\infty}^{\infty} F(\tau) S(\mu - \tau)\, d\tau \\ &= \frac{1}{\Delta T} \int_{-\infty}^{\infty} F(\tau) \sum_{n=-\infty}^{\infty} \delta\left(\mu - \tau - \frac{n}{\Delta T}\right) d\tau \\ &= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} F(\tau)\, \delta\left(\mu - \tau - \frac{n}{\Delta T}\right) d\tau \\ &= \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(\mu - \frac{n}{\Delta T}\right) \end{aligned} \qquad (4.3\text{-}5)$$

Signal sampling - result

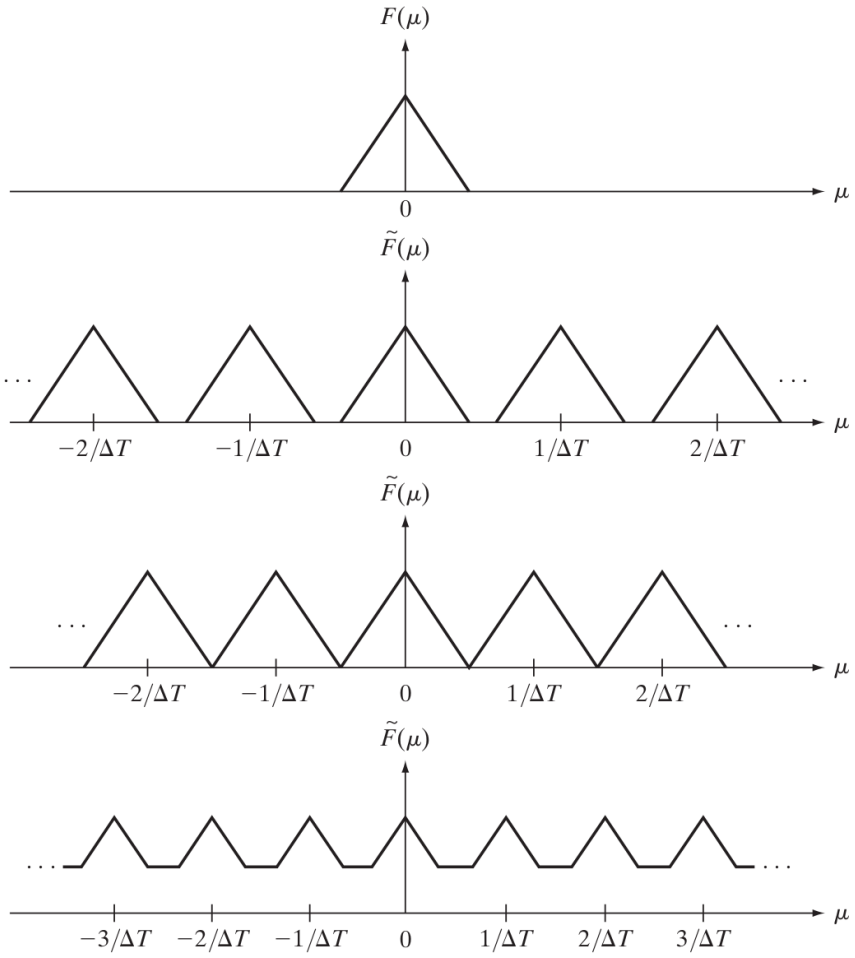
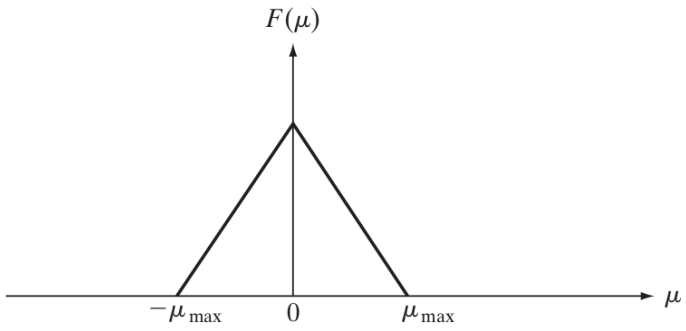
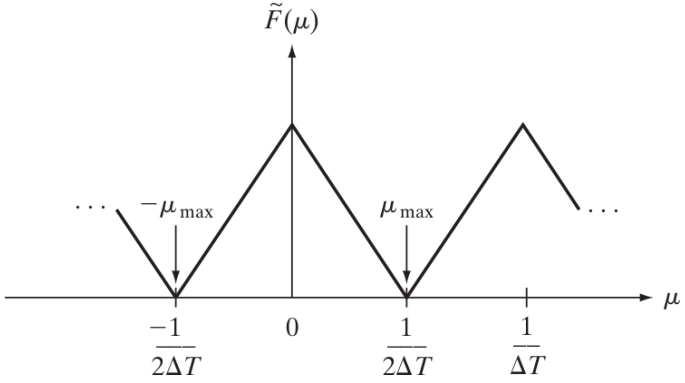


Figure: Sampling function

Nyquist-Shannon criterion

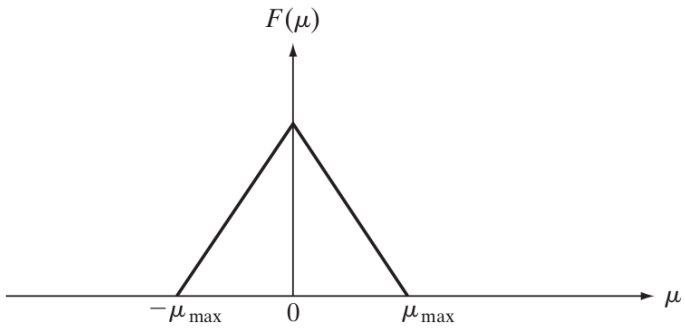
Q: *When can we guarantee now a "perfect" (i.e. lossless) signal reconstruction ?*

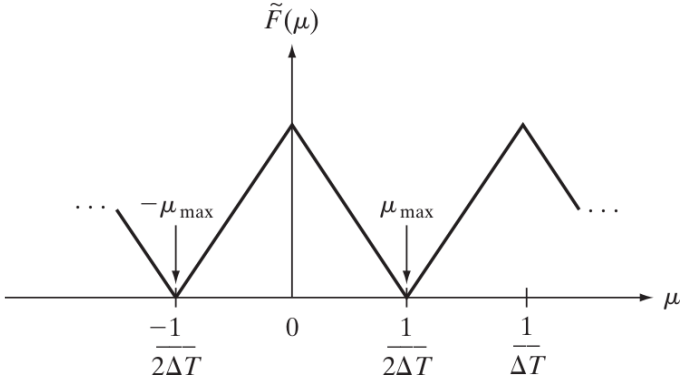




Nyquist-Shannon criterion

Q: *When can we guarantee now a "perfect" (i.e. lossless) signal reconstruction ?*





assume:

- bandwidth (i.e. signal period length) $\omega = \Delta t$
- signal frequency $\mu_{max} = \frac{1}{\omega}$
- sampling frequency $\frac{1}{\Delta T}$

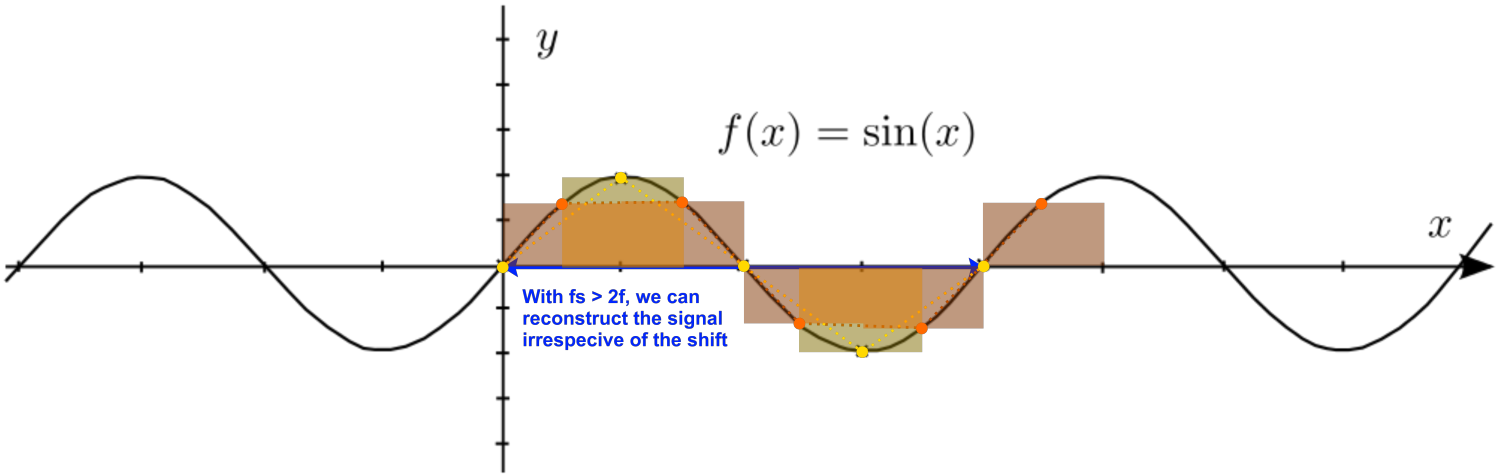
$$\frac{1}{\Delta T} > 2\mu_{\max} \qquad (4.3\text{-}6)$$

Figure: Sampling function

Nyquist-Shannon criterion on images - 1D

When connecting samples of the curve, we still see straight lines - an *imperfect* reconstruction.

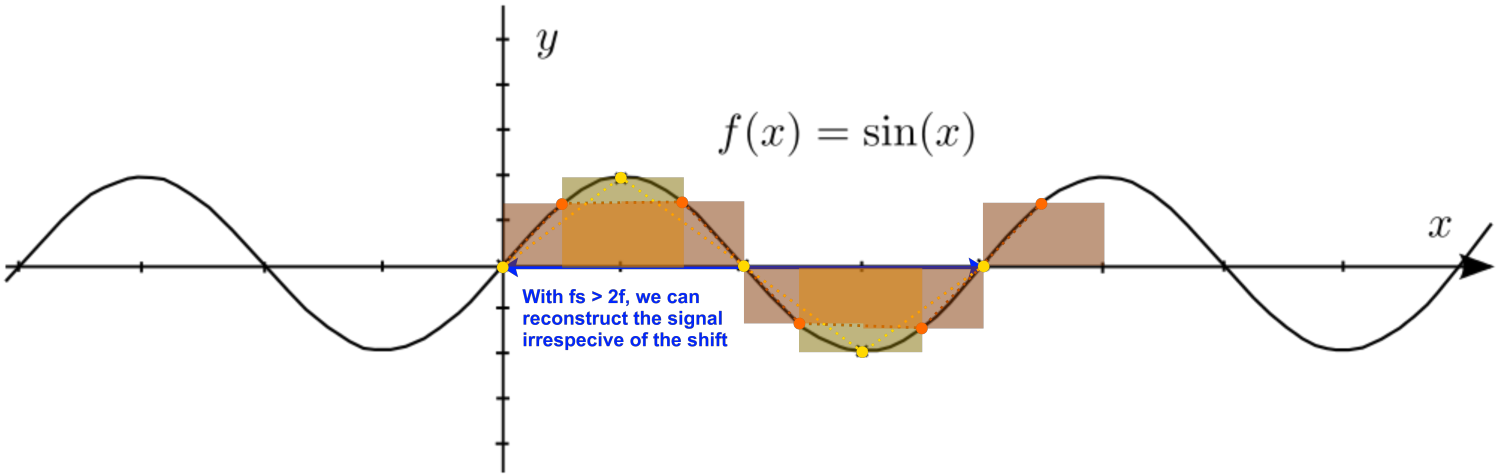
Q: *When (or: at what frequency) can the signal be perfectly reconstructed ?*



Nyquist-Shannon criterion on images - 1D

When connecting samples of the curve, we still see straight lines - an *imperfect* reconstruction.

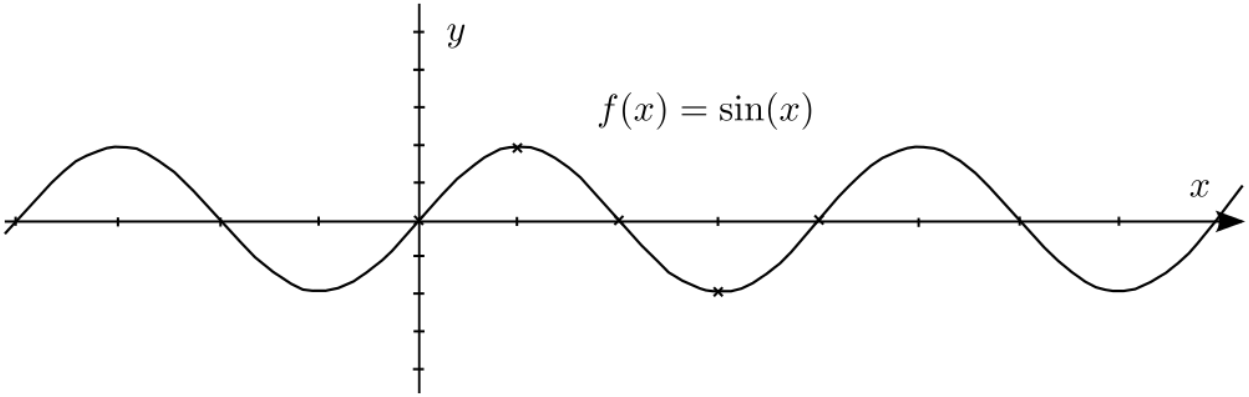
Q: *When (or: at what frequency) can the signal be perfectly reconstructed ?*



A: Here: if $\frac{1}{\Delta T} \leqslant 1px$

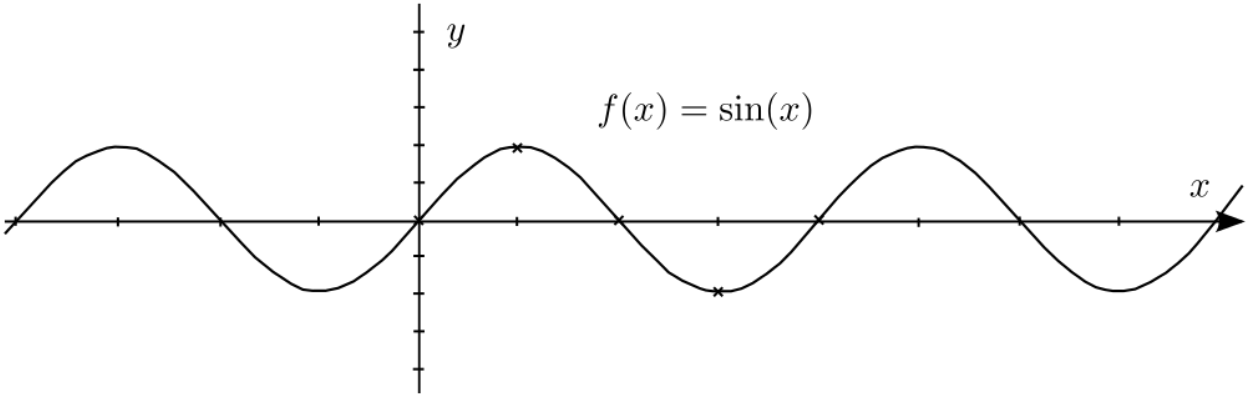
Nyquist-Shannon criterion on images - 1D

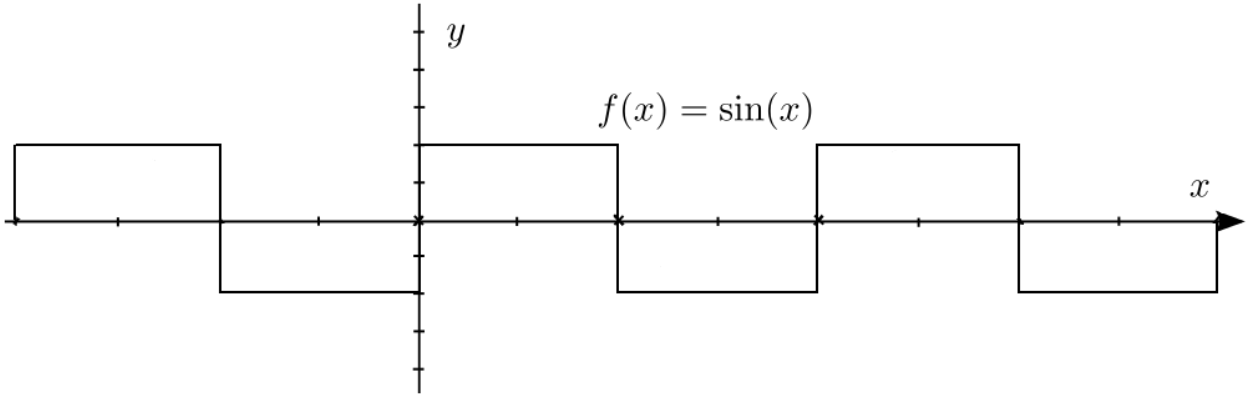
Q: *Does* $\frac{1}{\Delta T}$ *not always need to be* $\leqslant 1px$ *?*



Nyquist-Shannon criterion on images - 1D

Q: *Does* $\frac{1}{\Delta T}$ *not always need to be* $\leqslant 1px$ *?*

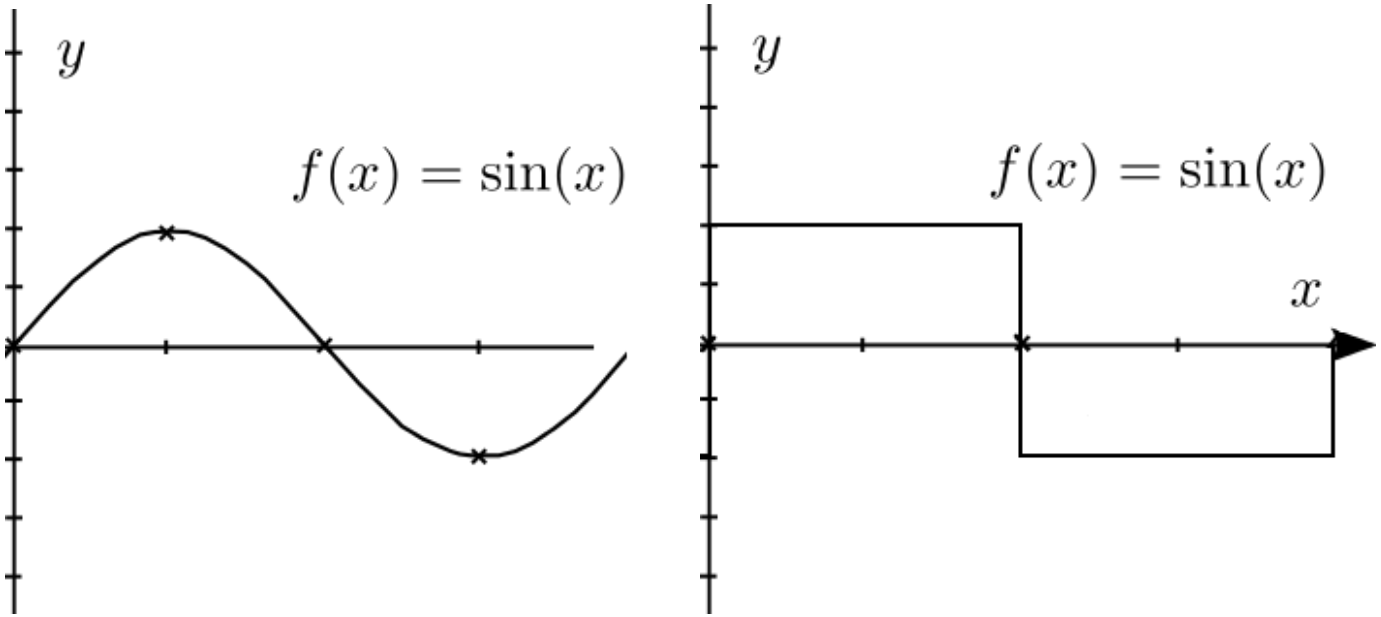




A: No - functions with a larger pixel separation can be reconstructed from smaller sampling frequencies $\frac{1}{\Delta T}$.

Nyquist-Shannon criterion

Functions $F(\omega, x)$ with a *larger bandwidth* ω (and thus a lower frequency μ_{max} in $f(x)$) can be perfectly reconstructed from a lower sampling frequency (and thus a larger sampling width).

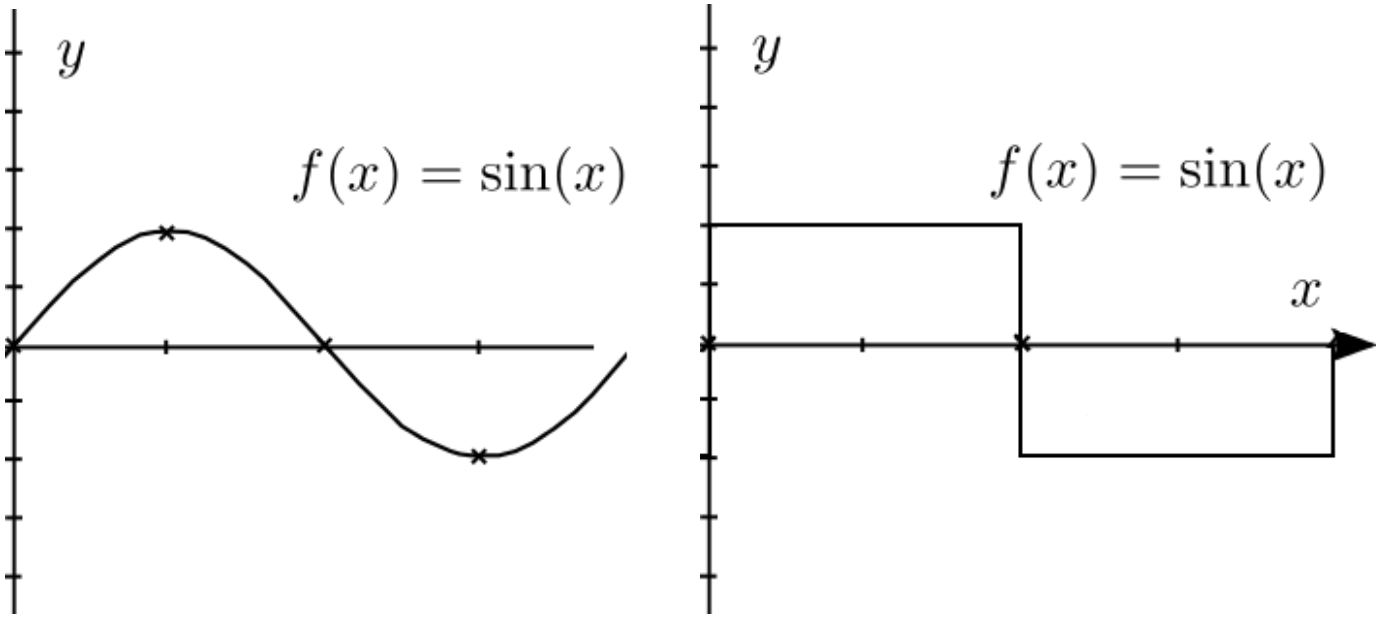


(a) Sampling sine - $\mu_{max} = 100$ samples / 100 px = 1

(b) Sampling box - $\mu_{max} = 2$ samples / 100 px = 0.02

Nyquist-Shannon criterion

Functions $F(\omega, x)$ with a *larger bandwidth* ω (and thus a lower frequency μ_{max} in $f(x)$) can be perfectly reconstructed from a lower sampling frequency (and thus a larger sampling width).



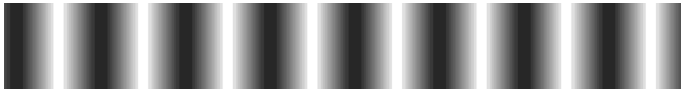
(a) Sampling sine - $\mu_{max} = 100$ samples / 100 px = 1

(b) Sampling box - $\mu_{max} = 2$ samples / 100 px = 0.02

Q: *How does this apply now to images ?*

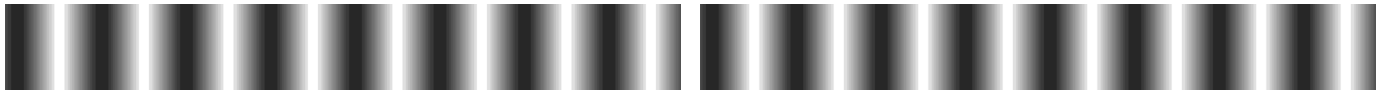
Nyquist-Shannon criterion on images - 2D

Let us replicate sine- and box as *image functions*.



Nyquist-Shannon criterion on images - 2D

Sample at half the frequency (i.e half the no. pixel samples).



(a) Sample positions (b) Resampled image

(a) Sample positions (b) Resampled image

Nyquist-Shannon criterion on images - 2D

(a) Original function $f(x)$ (b) $f_s(x) = \frac{1}{64} f(x)$

(a) Original function $f(x)$ (b) $f_s(x) = \frac{1}{64} f(x)$

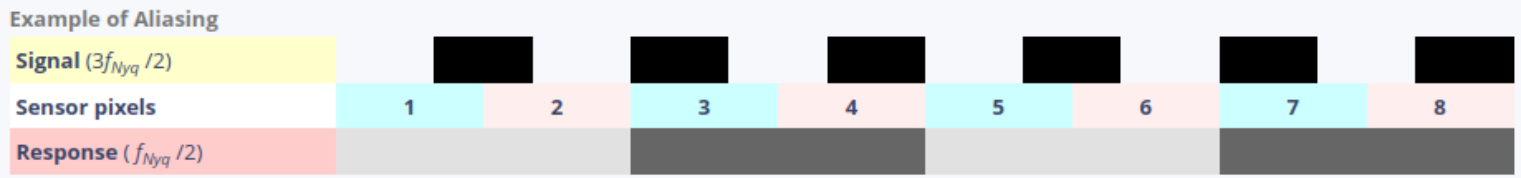
Nyquist-Shannon criterion on images - 2D

(a) Original function $f(x)$ (b) $f_s(x) = \frac{1}{64} f(x)$

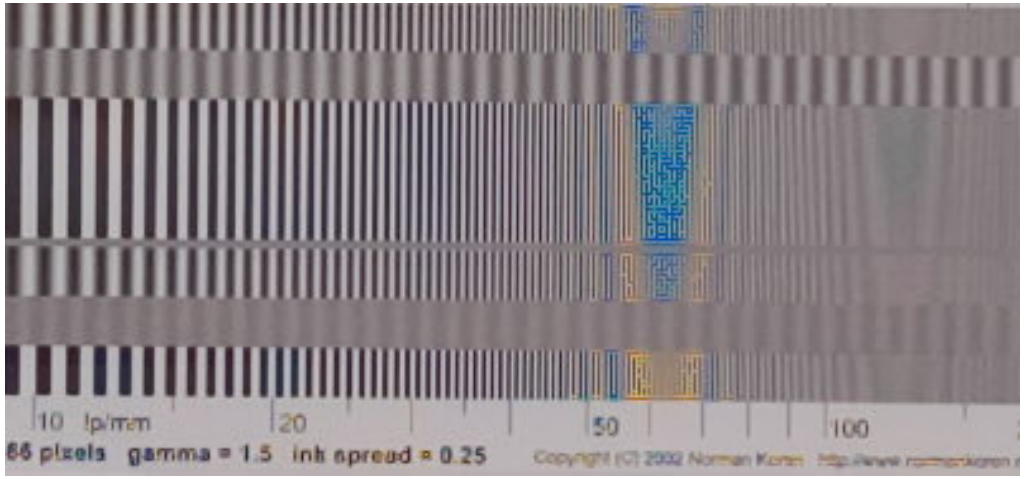
(a) Original function $f(x)$ (b) $f_s(x) = \frac{1}{64} f(x)$

Observation: the narrower the images feature ($\rightarrow$ band*width*), the higher the image resolution ($\rightarrow$ frequency) needs to be to flawlessly display the image ($\rightarrow$ perfectly reconstruct it).

Nyquist-Shannon criterion on images - 2D



(a) Pattern resampling.



(b) Sampling test on Kodac DCS 14.

Nyquist-Shannon criterion on images - wrap-up

Observation: With shrinking resolution, there are insufficient samples to reconstruct the features (here: edges) within the image, leading to **aliasing**.

That's it for this week!