1 Introduction to Vision

The - relatively vaguely defined - term *computer vision* is used a lot nowadays to describe a field dealing with the capture, analysis and interpretation of images to infer useful information in a sense of *understanding* the scene.

The choice of the term *vision*, as well as the ultimate goal of *understanding*, both borrow from the human biology, and indeed, mimicking the capabilities of human visual perception has been a constant inspiration in the history of this field. While we will see later of the recent successes of modern machine learning applied to imaging problems, for certain specific problems even reaching *super human performance*, the human visual system is still superior to the machine in other problems.

Consequently, we will in this chapter try to develop a basic understanding of the visual perception in humans, get an idea of the concept of an *image* and how *digital images* are the basis of the field of computer vision.

1.1 Relevance and Physiology of the Human Visual System

Among the different senses of the human body, the visual sense is often arguably considered the most important.

There are very valid social arguments *against* this statement, for example considering the auditory sense more important, as hearing to a large extend enables human social interaction and thus it might create more severe social consequences for a human when hearing is impaired than when vision is impaired. However, we have a strong technical / engineering argument towards considering vision as the most important sense: **among all human senses vision uses by far the most bandwidth.**

Try to derive a lower and an upper bound on the bandwidth of the human visual sense and compare with the literature.

1.1.1 Relevance of the visual sense

- Images can contain a very large amount of information (*"a picture says more than a thousand words"*)

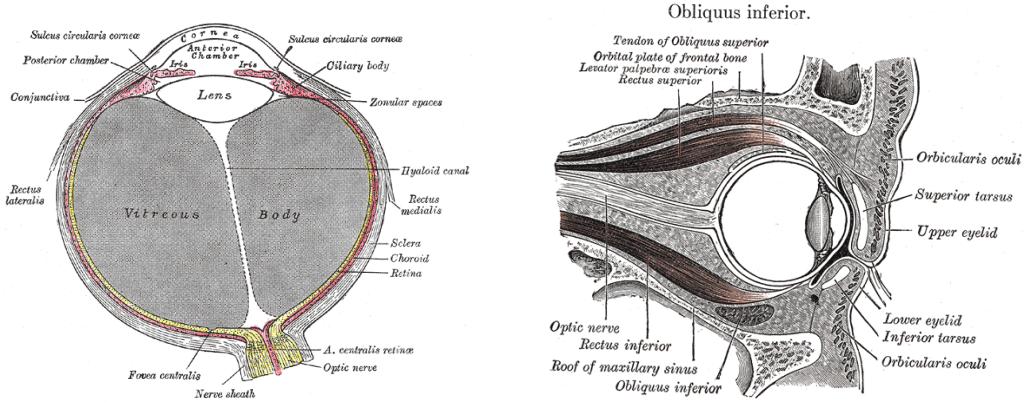
- Humans can grasp very complex information very quickly from a visual representation (see for example George Polya's famous four principles how to solve a mathematical problem, where the first principle includes the suggestion *"Can you think of a picture or a diagram that might help you understand the problem?"*)
- However, how to computationally interpretate an image is not directly clear, we have to transform the image in some sense to make clues out of it, may it be in an "engineered" or a "learned" fashion

Performance of biological vision systems

- Detection of simple objects in less than 1 second
- Interpretation of very complex scenes in a few seconds
- Ability to correctly derive information from incomplete or distorted visual inputs

Basic Parameters of the human visual system

- **Sensor:** approx. $130 \cdot 10^6$ receptors in the eye (for comparison: typical image resolution of a 4k screen: $3 \cdot 3840 \cdot 2160 = 3 \cdot 8.3 \cdot 10^6$ subpixels)
- **Datalink:** approx. 10^6 nerve fibers in the optical nerve
- **Processing:** approx. 10^{11} Neurons where each is coupled to approx. 10^4 other neurons



(a) (b)

Figure 1.1: The human eye from Gray's anatomy. (a) Closeup of the eyeball, axial cut, (b) Saggital overview.

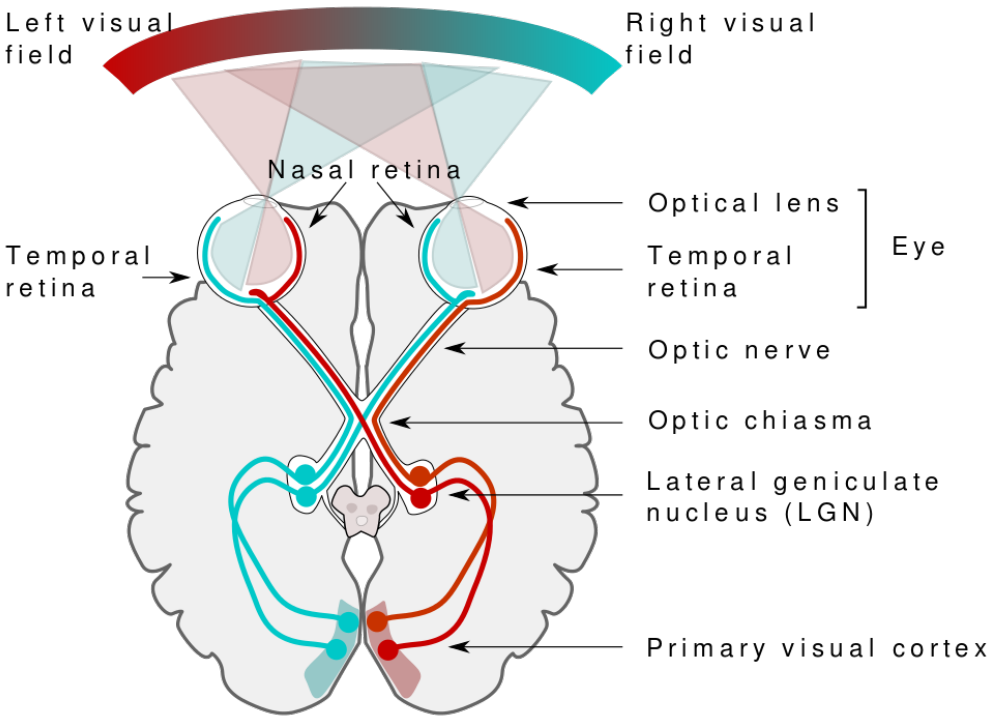


Figure 1.2: The human visual pathway. Information about the light that is passing through the optical system of the eyes is converted to electrical signal of the nerves. Information from both eyes is processed at both hemispheres of the brain, by the crossing of the information about half of the visual field at the optic chiasm. After synaptic relaying at the LGN the image information finally is projected by the optic raditation to the primary visual correctly at the very posterior part of the brain.

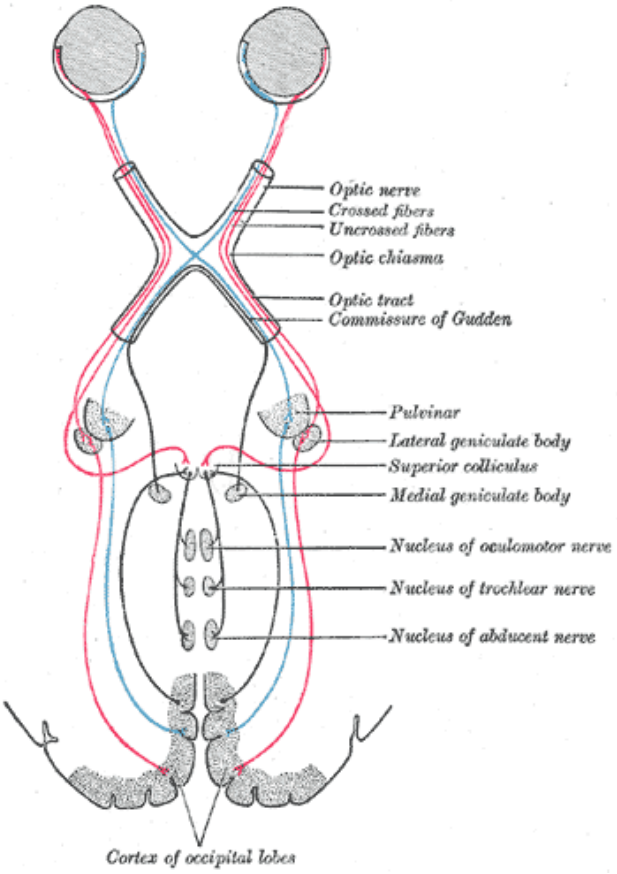


Figure 1.3: The human visual pathway as depicted on *Figure 722 of Gray's Anatomy*, first published in 1855 **[Gray1955]** Compare to the simplified modern Wikipedia figure presented before.

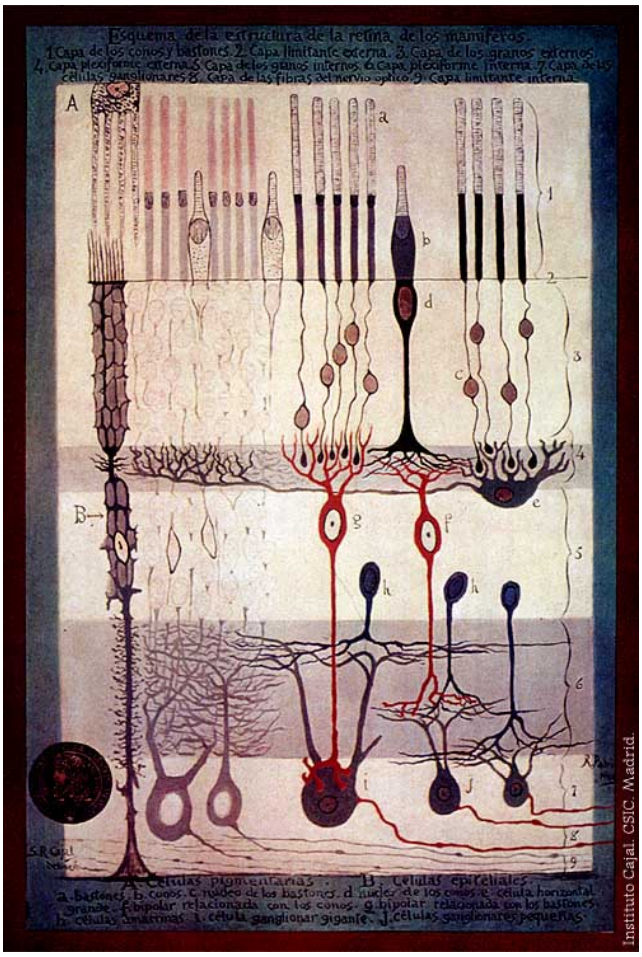


Figure 1.4: Drawing of the human retina by Ramón y Cajal, 1884 [**Cajal1884**], who shared the nobel price with Golgi in 1906 for their independent studies on the nervous system. The drawing shows the sensory receptors on the top and the subsequent "wiring" of the different retinal neuron layers - the retina is itself already forming a multi-layer neuronal network.

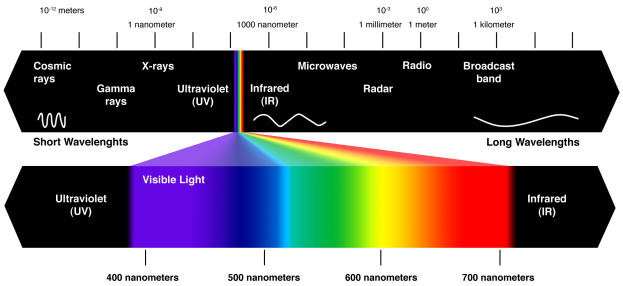


Figure 1.5: The human eye can see only a small part of the spectrum, referred to as the *visible spectrum*. When visualizing other parts of the spectrum we have to use techniques to map the information to the visual spectrum, for example on a screen.

2 Digital Images

In this chapter we discuss how to describe images as mathematical object that we can treat in a computer. We look into basic statistics on images and into basic imaging operations,in particular morphological operators and filter.

2.1 Images as Functions

We can describe a 2D image as a function

$$f : \mathbb{N} \times \mathbb{N} \to \mathbb{R} \tag{2.1}$$

for example

$$f : i, j \mapsto z \text{ where } i, j \in \mathbb{N} \text{ and } z \in \mathbb{R} \tag{2.2}$$

f(i,j) = b

in expression, tuples of *pixel indices* (i, j) over the natural numbers are mapped to a real-valued *intensity value* z. Such an image is therefore also referred to as an intensity image, or as a grayscale image, as typically the lowest intensity is mapped to black, and the highest intensity to white. Intermediate intensities thus attain values of gray.

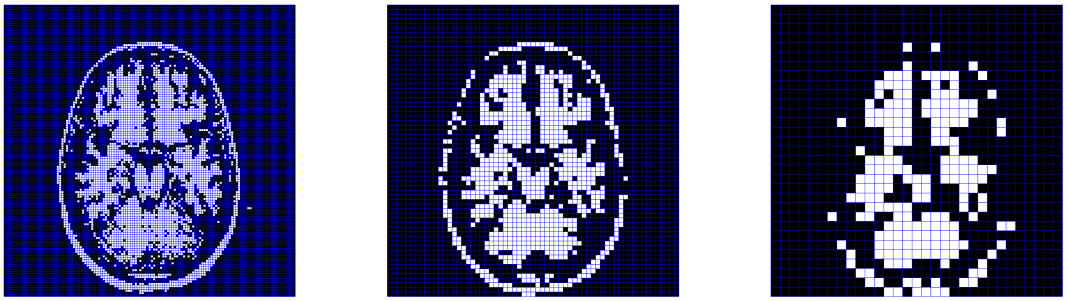
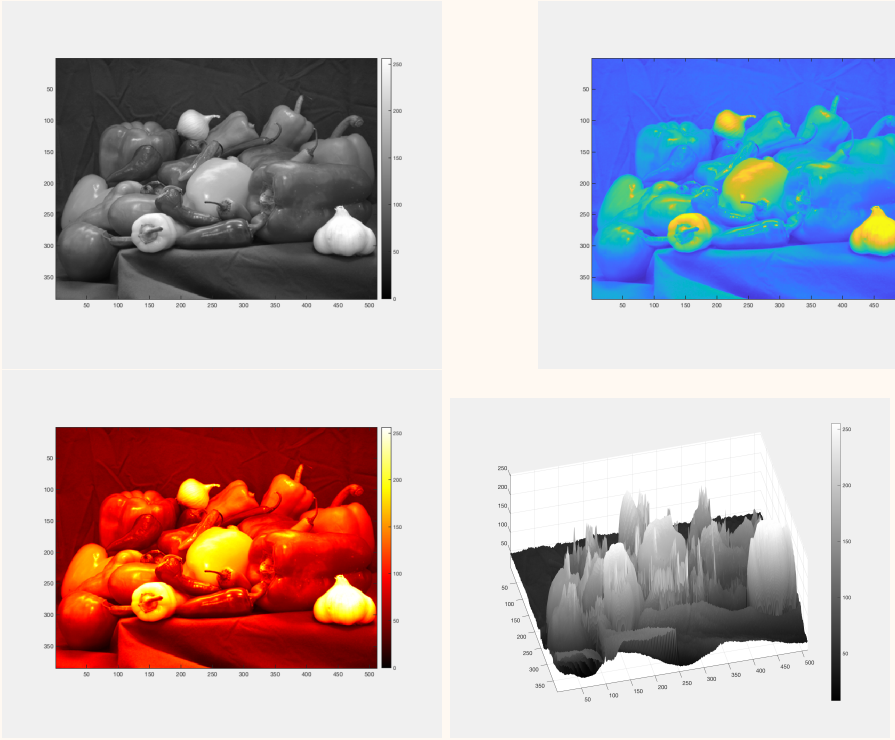


Figure 2.1: A binarized slice of a T1 weighted MRI scan of a human brain using different resolutions. Left: 128x128 pixels, middle: 64x64 pixels, right: 32x32 pixels. Pixel grid overlayed in blue.

Remark 2.1.1: Different possibilities to visualize intensity maps



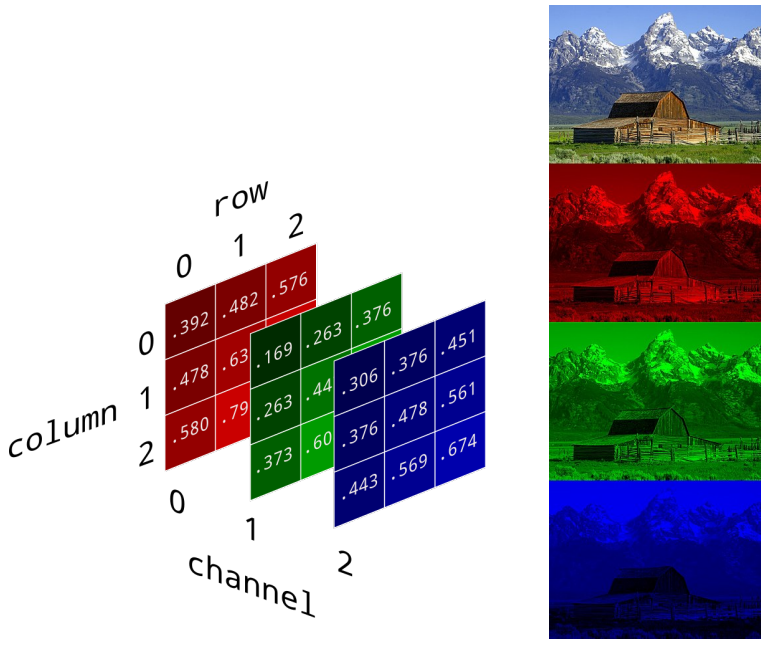
An 2D intensity map (intensity image) is frequently displayed as grayscale image where the lowest intensity is mapped to black, and the highest intensity to white, intermediate intensities attaining values of gray. However this is not the only way to visualize intensity maps. We can use arbitray colormaps, as shown here with the frequently used *colormaps* `parula` and `hot`, or even non-image based representations, for example surfaces where the intensity values encodes the height.

Q.: Whats special about `parula` or `viridis` compared to other colormaps?

N-Dimensional Image In the general, N-dimensional case, we can consider an n-dimensional image as a map

$$f : \mathbb{N}^N \to \mathbb{R}. \tag{2.3}$$

A typical example for $N = 3$ is a 3D-MRI scan. Note that "true" 3D scans exists as well as stacks of 2D acquisition slices *represented* as a 3-dimensional array in the computer (the latter case sometimes reffed to as *2.5D imaging*).



(a) (b)

Figure 2.2: RGB Color image split in its R, G and B components. (source (b): Wikimedia commons, public domain, (a): Brandon Rohrer)

Color as an additional (non-spatial) dimension Color can be encoded in different ways, where the RGB encoding, using red, green and blue channels, is very common. RGB encoding is used in the subpixels in nowadays displays as well as in the color receptors of the human eye. In general, in a 2D-color image, the 2D (i, j) tuple is augmented with a color dimension $c \in 1, 2, 3$

$$f : \mathbb{N}^2 \times \{1, 2, 3\} \to \mathbb{R}. \tag{2.4}$$

Remark 2.1.2: N+1 dimensions in the computer

Note that by the augmentation of the color dimension a 2D image is now represented as a 3D array ("i x j x c") in the computer. An N-D image is represented as N+1 dimensional array ("j x ... x k x c").

Q.: How is an RGB image represented in the MATLAB workspace?

Remark 2.1.3: Slicing the color dimension to return RGB triplets

An alternative way to access RGB data is to *slice* an image map f at the color dimension in the sense of `p=f(i,j,:)`. By this means, we effectively treat f as a vector valued function, returning a RGB-triplet vector p for each tuple (i, j) . Depending on the application, this functionally equivalent view on the data is more handy than the view of returning three independent R, G and B intensity maps.

Multi-Spectral and Multi-Modal-Imaging Images providing information of different sensors in an aligned coordinate space are referred to as multi-spectral-images. An example are multi-spectral-cameras, as typically used in space exploration, where several sensors capture different wavelength ranges of the optical spectrum, for example spectral intervals in the visible range plus intervals in the near infrared plus intervals in the far infrared. RGB color images fullfill this definition, i.e. a RGB color image is a multi-spectral-image in the strict sense. However, the term multi-spectral-image is frequently also used to make clear that the image is *not* "just an ordinary" RGB image but considers more / different spectral intervals.

A conceptual very similar example in medical imaging, mostly referred to as *multi-modal-imaging*, is the acquisitions of different MRI sequences of a patients anatomy (for example a T1-weighted MRI scan and a T2-weighted-MRI) acquired in the same scanning session (neglecting patient movement).

Formally we can describe both approaches, consider multi-spectral-imaging and multi-modal-imaging exactly as the color imaging before:

$$f : \mathbb{N}^N \times \mathbb{N} \to \mathbb{R} \tag{2.5}$$

i.e. the image dimension is "j x ... x k x s" where the dimension $j, ..., k$ are geometric dimensions and s is an additional dimension encoding the spectrum / the color-channel.

Remark 2.1.4: Fixed channel dimension in deep networks

Note that many pre-trained deep learning architectures for imaging are trained on color images, i.e. they have a dimensionality of $(i, j, 3)$ where i,j are traditionally 224 and nowadays (high resolution networks) up to around 2000 pixels. While the fixed i,j resolution constitutes own problems, the fixed channel dimension c=3 needs special treatment when applying such networks two non RGB data (as it is very typical in medical imaging).

However, in practical image processing applications multi-modal images are often provided as sets of independent image files (yielding independent image arrays in the computer).

For example in the case of a multi-modal 2D MRI acquisition with two modes (T1 and T2) and a geometric image size of 512×512 pixels the data could be represented as one array of dimension $(512 \times 512 \times 2)$ where the triple $(15, 7, 1)$ would map to the intensity value of the T1 MRI at coordinate i=15, j=7 and the triple $(15, 7, 2)$ would map to the intensity value of the T2 MRI at this image coordinate. However in the clinical practice often two *separate image files* each with a size of size of 512×512 pixels are used. One file for the T1 acquisition, one for the T2. When we want to to deep learning on such images we might have to combine the separate files into a ($512 \times 512 \times 2$) representation to feed them into the network for multi-channel learning

Example

Image Sequence in Time Sequences of images in time (like for example video) are normally treated by adding another dimension in the sense of $(i, ..., k, c, t)$. For an uncompressed 2D-Color-Video Sequence with Full HD resolution we would for example yield a dimensionality of $(1920, 1080, 3, t)$

where t is the number of stored frames depending on the frame rate and the length of the sequence in seconds.

Remark 2.1.5: Framework dependent ordering

Note that the temporal dimension t in an image $f(i, ..., k, c, t)$ could also be put as the first parameter, i.e. $g(t, i, ..., k, c)$, depending on the used software framework. The same holds for the color dimension c or in principle any other dimension. Always consult the documentation of the respective framework used for loading the image data which order is valid and confirm correct usage by visualizing test images.
Note that different libraries in the same programming language might use different conventions. For example Python's often used imaging library `PIL` uses (i,j,c) ("row x column x channel)" while the famous `pytorch` deep learning library expects (c,i,j) ("channel x row x column").

2.1.1 Sampling, Resolution and Quantization

The **Shannon-Nyquist** theorem for signals applies to images (which are signals) as well. Images can also be analyzed in *frequency domain* (Fourier-Transformation), however this is not covered in this course and also not very typical for deep learning applications, where working in the *spatial domain* is more common.

Sampling describes the spatial discretisation of the image to a certain image resolution. **Quantization** describes the discretisation of the image intensity range.

No further lecture notes on this subsection.

2.1.2 Derived color images (e.g. pseudo-color and false-color)

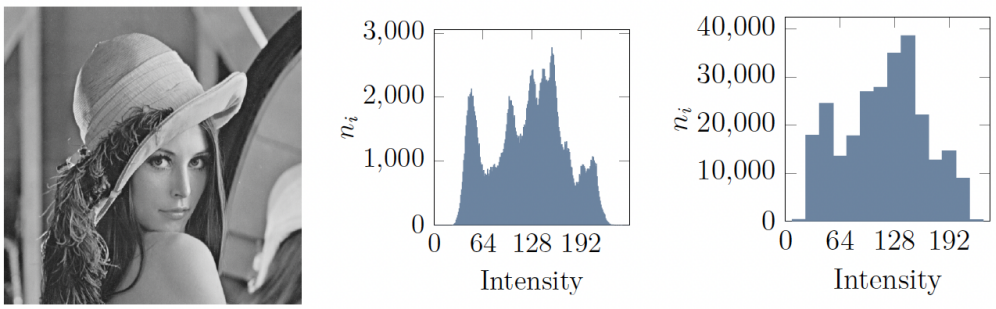
No lecture notes on this subsection. Refer to this excellent documentation on dealing with different image types, including indexed-images (summarizing both pseudo-color and false-color):
`https://de.mathworks.com/help/images/image-types-in-the-toolbox.html`

2.2 Image Enhancement

We will discuss subjective and objective enhancement techniques.

2.2.1 Histograms

Histograms describe the intensity value distribution of an image by means of its discrete probability distribution. They can be normalized or not. For



(a) Example grayscale image.

(b) Histogram of the example image with 256 bins.

(c) Histogram of the example image with 16 bins.

Figure 2.3: Example of an image and it's histogram (Figure from Maier et al., 2018)

a discrete 2D-image $B = b(x, y)$with $b \in \mathcal{D}$ the absolute histogram is given as the PDF

$$H_g(i) = |\{(x,y) \in B | b(x,y) = i\}| \forall i \in \mathcal{D} \tag{2.6}$$

Where $\|\cdot\|$ denotes the cardinality of the set and $\mathcal{D}$ the domain of the image intensities. An example for $\mathcal{D}$ is the set $\{0,...,255\}$ for typical 8-bit grayscale images.

Remark 2.2.1: CDF

Note the discrete cumulative distribution function (CDF) is given as the cumulative sum over the PDF. In expression, the CDF $C_g(j)$ of an absolute grayscale histogram $H_g(i)$ is given as

$$C_g(j) = \sum_{i=1}^{j} H_g(i) \tag{2.7}$$

(See also exercises)

The absolute histogram $H_g(i)$ could be normalized to a relative histogram $h_g(i)$ by dividing by the number of pixels NM.

Instead of computing the histogram for every possible intensity value of the image, histograms are often using a *binning* approach. Therefore, the domain $\mathcal{D}$ is partitioned into intervals of a fixed length. See figure part (c).

Q.: Why is binning strictly necessary for image with real valued domain?

2.2.2 Window and Level

How many gray levels the human eye can distinguish (without re-accommodating the iris to subparts of the image) is an important questions. However, definite quantitative answer is difficult (and would involve things like screen calibration, lighting condition definition and more), but a simple **quantization** experiment can give us an idea, see Figure 2.4.

Problem: How to display images that *contain more gray values than the human eye can discriminate?* => Trick: Show only a certain *subset* of gray values (an interval of the histogram) at a given time using the full intensity range of the display screen to visualize this interval.

Implementation int he computer by defining of a injective partial linear map that maps a certain interval of the intensity spectrum to the grayvalue spectrum of the screen (and suppresses everything else). See figure 2.5.

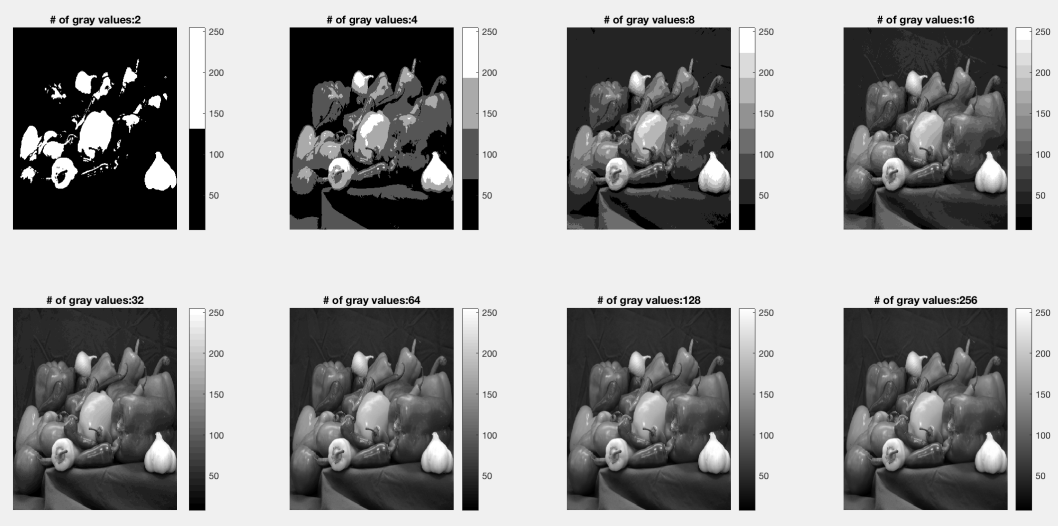


Figure 2.4: Simple experiment on how many grayvalues the human eye can approximatly distinguish by reducing the number of displayed grayvalues in power of two steps. (see also exercise example code `ex_1_gray_value_quantisation.m`

In clinical practise, the function specified over its domain by level (offset, L) and window (interval width of the partial definition domain,W). L/W are typcial parameters of clinical display systems in medical imaging. But the trick of using injective partial linear maps to visualize only subsets of the image intensits domain is used all over computer vision (see for example *tonemapping* for displaying HDR images, which is the same problem from an algorithmic perspective)

Q.: Try loading example MRI images in ITK-Snap and changing window and level.

End of Lecture 1

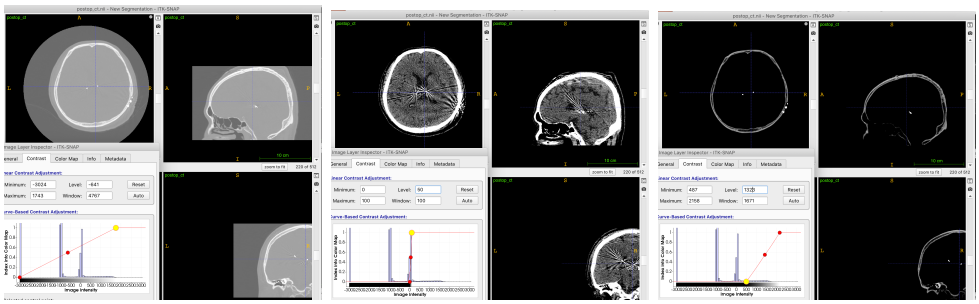


Figure 2.5: Different mapping of a CT image intensity range to the screen range by a partial linear function defined via level and window. Left: full range, middle: range useful to see the brain tissue, right: range useful to see bone.