

Summary of the lecture
'Image Processing'
by Prof. Christian Bauckhage
(WiSe 2021/2022)

Fabrice Beaumont
Matrikel-Nr: 2747609
Rheinische Friedrich-Wilhelms-Universität Bonn

November 11, 2021

Chapter 2

Digital Photography

2.1 Image acquisition and digital photography

(Digital) photography is the practice of visualizing, recording and storing light.

Definition 1.1: Light

Light can be described as both

- as electromagnetic radiation where energy propagates in form of electromagnetic waves and
- as a particle or quantum phenomenon where photons carry the electromagnetic force.

Light has a constant *speed* of $c := 299\,792\,458\text{ m/s}$. Light can have different *frequency* ν ($[\text{s}^{-1}]$) and *wavelength* λ ($[\text{m}]$), but these properties are reciprocal:

$$c = \nu\lambda \quad \left[\frac{\text{m}}{\text{s}} \right] \quad (\text{Eq. 1})$$

The *energy* E of light is defined as

$$E = h\nu = h\frac{c}{\lambda} \quad [\text{J}] \quad (\text{Eq. 2})$$

where $h = 6.6260689633 \times 10^{-34} [\text{J s}]$ is the **Planck's constant**.

From these definitions we can derive the relation that light with a high frequency and small wavelength carries more energy.

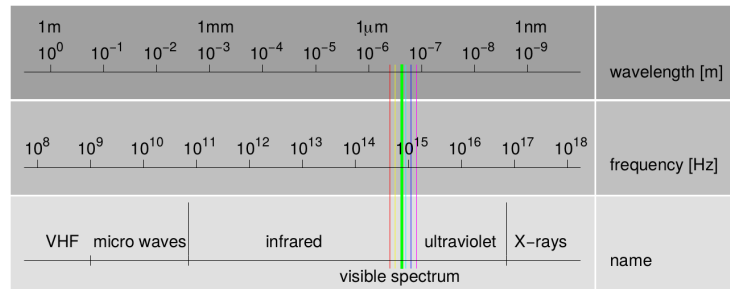


Figure 2.1 – Overview over the electromagnetic spectrum.

There are three ways to visualize light. It can be:

- **reflected** (e.g. photographic images or scanned images),
- **absorbed** (e.g. X-ray images) or
- **emitted** (infrared images).

There are several ways to record light. Digital (color-)photography relies on CCD cameras. CCD stands for **charged-coupled device** and describes a semiconductor device consisting of photosensitive elements. Modern consumer cameras have at least 1600×1200 such elements.

Rays of light incident on the camera lens hit the CCD matrix. These photons generate a charge on the CCD elements by freeing electrons. The more light there is, the more electrons are freed. Depending on the camera quality, a single CCD element stores 100,000 to 350,000 electrons before saturating. An electric field gathers electrons into packets which are read and quantized into intensity levels. Thus every CCD element records a value of light intensity.

Since the amount of represented intensity levels relates to the needed memory bits per CCD, and since the perception of the human visual system is limited in this regard, typical images typically only store $2^8 = 256$ intensity levels (one byte) per color channel (intensity quantization).

Definition 1.2: Intensity image

Intensity images (grayscale images) do not contain color information and only display recorded light intensity (luminescence) per pixel in terms of shades of gray.

Definition 1.3: Color image

Color images do contain color information and split the recorded light into basic colors (commonly red, green and blue) from which the appropriate colors can be reconstructed on a display device later. To split the light one can use either a **trichroic prism** and record each basic color in a separate CCD array, or superimpose a mosaic of color filters (**Bayer filter**) over a single CCD array. Since the latter technique uses three times less CCD elements it is much cheaper (but also less accurate).

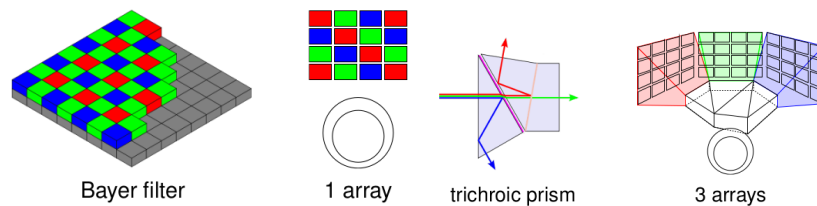


Figure 2.2 – Two techniques of CCD cameras to split recorded light into three basic colors.

There are several problems with the CCD technology:

- **Geometric distortions.** CCD elements are typically not perfectly square. This may necessitate computational correction to compensate for geometric distortions.
- **Blooming.** Overexposure may free too many electrons, which may overflow to neighboring elements and cause highlight effects.
- **Dark noise.** CCD sensors are sensitive to parts of the invisible electromagnetic spectrum (UV or IR radiation) and may thus record invisible apparitions.

There are numerous file formats for storing digital images. But these can be separated into two main types of image file formats:

- vector graphics formats (e.g. SVG, FIG) and
- pixmaps (bitmap, raster graphics; e.g. JPG, GIF, PNG, TIFF).

Definition 1.4: PPM format

The **PPM (portable pixmap) format** stores image data in the following way:

The first four lines in the file format contain:

1. a magic number (indicating ASCII [P1, P2, P3] or binary [P4, P5, P6] format),
2. comments,
3. image width and image height and
4. the maximum color value (which is the same for all color channels).

After these lines, the image data is stored in red-green-blue-triples of hexadecimal values.

```
P6
# CREATOR: XV Version 3.10a  Rev: 12/29/94  (PNG patch 1.2)
74 82
255
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff fe fe fe fe fe ff ff ff ff ff ff ff ff ff
ff ff fd c7 c6 c4 19 15 ...
```

Figure 2.3 – Example header of a PPM file.

Keep in mind, that there are many different ways of image compression. In general these annihilate information and thus reduce memory requirements.

2.2 Representations of digital images

As explained, a digital image represents digitized and quantized information. The basic element of digital images are called **pixels** (picture elements), which are (commonly) arranged on a rectangular grid called a **pixmap**. The **resolution** of a digital image is expressed in terms of width times height of its pixmap. Digital images often have an aspect ratio of 4:3 (e.g. 1024×768) or 16:9 (e.g. 1920×1080).

In memory, intensity images are stored in a single pixmap. Color images are stored in three to four pixmaps. One per color channel and a possible α -layer. Accordingly, these images can be stored in two-, three- or four-dimensional arrays. Rows and columns of a pixel array are counted from zero and the pixel $p_{0,0}$ is located in the *upper left corner* of the pixel array.

Definition 2.1: 4-Neighborhood

The **4-neighborhood** of a pixel is the set of the two vertically and the two horizontally adjacent pixels.

Definition 2.2: 8-Neighborhood

The **8-neighborhood** of a pixel is the set of the 4-neighborhood and the four diagonally adjacent pixels.

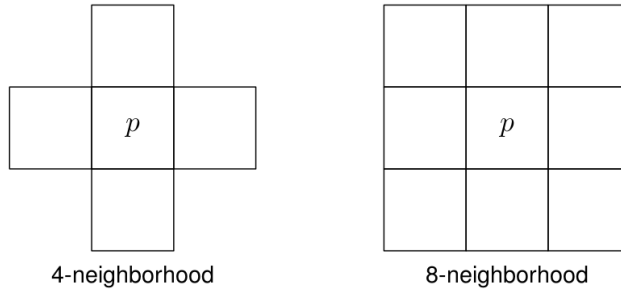


Figure 2.4

Note that the Euclidean distance yields different distances between the pixels in the 8-neighborhood. It is possible to use another metric or another arrangement of the pixmap (e.g. a hexagonal-grid), but the technically simplest approach with respect to computer memory is the default rectangular grid.

2.2.1 Images as functions

In a digital intensity image, array coordinates $[i, j] \in \{0, \dots, M-1\} \times \{0, \dots, N-1\}$ are mapped to intensity values $p_{i,j} \in \{0, \dots, L-1\}$. Thus one may think of such an image as a **bi-variate discrete function with finite domain** $f : \mathbb{R}_+^2 \rightarrow \mathbb{R}_+$.

Since the pixels are enumerated from the upper left corner, mapping the planar points $[x_j, y_i]$ to the images array coordinates $[i, j]$ is done as

$$[x_i, y_j] \leftrightarrow [j, M - 1 - i] \quad (\text{Eq. 3})$$

Also, we assume that the two-dimensional point at which an image function is defined are equally spaced. More precise, the horizontal and vertical distances between neighboring point amounts to one:

$$\begin{aligned} \delta_x &= |x_{j\pm 1} - x_j| = |x \pm 1 - x| = 1 \\ \delta_y &= |y_{i\pm 1} - y_i| = |y \pm 1 - y| = 1 \end{aligned} \quad (\text{Eq. 4})$$

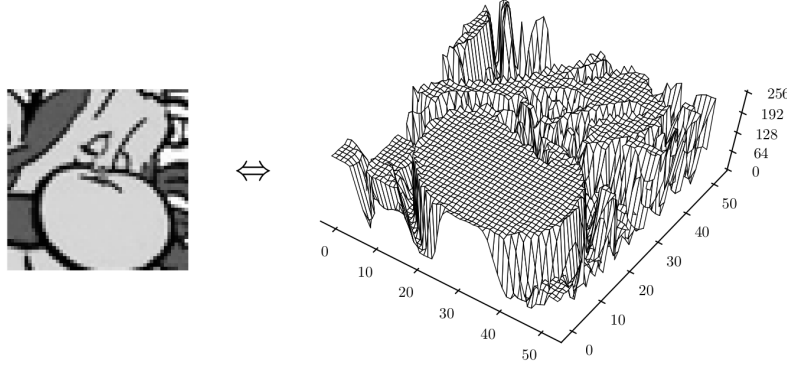


Figure 2.5

Notation: To signify that f is continuous, write $f(x)$. To signify that f is discrete (partial), write $f[x]$ (with $x \in \{\mathbb{R}, \mathbb{R}^2\}$).

Definition 2.3: Derivatives

Let f be a continuous, uni-variate function $f : \mathbb{R} \rightarrow \mathbb{R}$ where $x \mapsto f(x)$ and $x_0 \in \mathbb{R}$.

At x_0 , the **derivative of f from above** is

$$f'_+(x_0) = \lim_{\epsilon \rightarrow 0} \frac{f(x_0 + \epsilon) - f(x_0)}{\epsilon} \quad (\text{Eq. 5})$$

At x_0 , the **derivative of f from below** is

$$f'_-(x_0) = \lim_{\epsilon \rightarrow 0} \frac{f(x_0) - f(x_0 - \epsilon)}{\epsilon} \quad (\text{Eq. 6})$$

f is **differentiable** at x_0 , if

$$f'_+(x_0) = f'_-(x_0) = f'(x_0) = \frac{\partial}{\partial x} f(x_0) \quad (\text{Eq. 7})$$

In this case, $f'(x_0)$ indicates the *rate of change* of f at x_0 . That is the *slope of the tangent* of f at x_0 .

f is differentiable, if it is differentiable at any $x \in \mathbb{R}$.

Note that the derivation is a **linear operator**^a:

$$\frac{\partial}{\partial x} (af(x) + bg(x)) = a \frac{\partial}{\partial x} f(x) + b \frac{\partial}{\partial x} g(x) \quad (\text{Eq. 8})$$

^aAn **operator** is a function that maps functions to functions.

For the discrete case set for simplicity $x_{i+1} - x_i = 1 = \epsilon$. Then, the definitions are analogue:

- **Backward difference:**

$$f'_-[x] = \frac{f[x] - f[x-1]}{x - (x-1)} = f[x] - f[x-1] \quad (\text{Eq. 9})$$

- **Forward difference:**

$$f'_+[x] = \frac{f[x+1] - f[x]}{(x+1) - x} = f[x+1] - f[x] \quad (\text{Eq. 10})$$

- **Central difference:**

$$f'[x] = \frac{f'_+[x] - f'_-[x]}{2} = \frac{f[x+1] - f[x-1]}{2} \quad (\text{Eq. 11})$$

Note that if $f[x]$ has been sampled from $f(x)$, then the central difference of $f[x]$ provides a provably better approximation of the derivative of $f(x)$ than the backward- or forward differences.

Using an operator notation, we will also write the central difference as $f'[x] = d_x f[x]$.

2.2.2 Image gradient

By considering images as functions, we can compute their the gradient. For intensity images $f[x, y]$ it is defined as

$$\nabla f[x, y] = \begin{bmatrix} d_x f[x, y] \\ d_y f[x, y] \end{bmatrix} = \frac{1}{2} \begin{bmatrix} f[x+1, y] - f[x-1, y] \\ f[x, y+1] - f[x, y-1] \end{bmatrix} \quad (\text{Eq. 12})$$

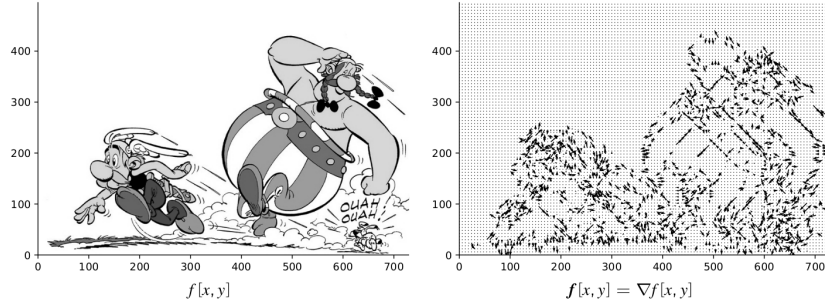


Figure 2.6 – Example intensity image and its gradient image.

Gradients of intensity images are typically visualized by only considering its magnitude

$$\|\nabla f[x, y]\| = \sqrt{\langle \nabla f[x, y], \nabla f[x, y] \rangle} = \sqrt{(\nabla f[x, y])^2} \quad (\text{Eq. 13})$$

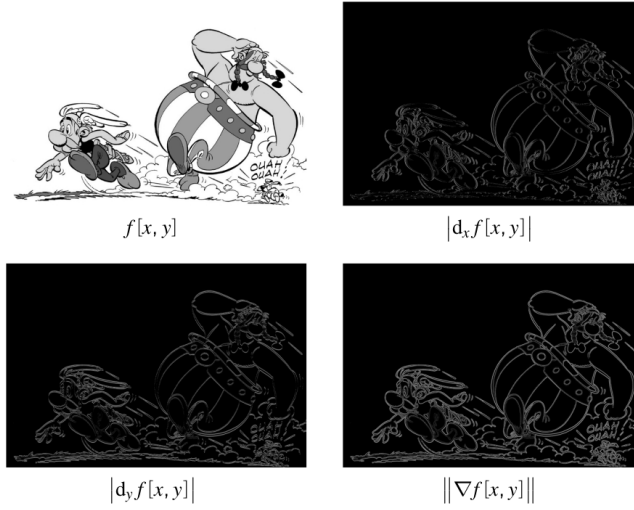


Figure 2.7 – Example of an intensity image, the norms of the directed gradients and the magnitude of the total gradient.

As one can see, the gradient (and its magnitude) can be used as an **edge detector**.

2.2.3 Emboss effect

The emboss effect transforms an image such that it resembles a copper engraving or an etching. It can be achieved by applying the following operator:

$$g[x, y] = \max \left(0, \min \left(255, 128 + f[x + 1, y - 1] - f[x - 1, y + 1] \right) \right) \quad (\text{Eq. 14})$$

Note that by definition it is $0 \leq g[x, y] \leq 255$. Furthermore, the resulting pixels are computed as 128 plus the difference in intensity between the lower left and upper right neighboring pixel (diagonal gradient).

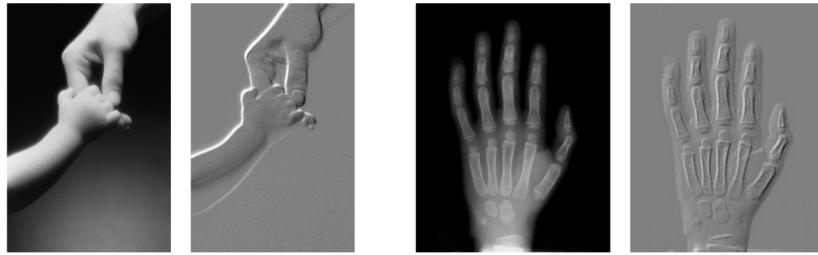


Figure 2.8 – Two examples of the emboss effect. Left the original intensity image, right an emboss effect on it.

Chapter 3

Basics

Chapter 4

Exercises

4.1 Sheet 0 - Questions in the lecture slides

4.1.1 Q1 - lecture 02

Assume you have an RGB image of resolution 1024×768 with 256 intensity levels per color channel. How many bytes of memory does such an image require when loaded into a computer?

Solution: To store $256 = 2^8$ intensity levels exactly eight bits (one byte) is required. Thus to store $1024 \times 768 = 786,432$ pixels (intensity levels), 786,432 bytes are required.

4.1.2 Q2 - lecture 03

The **divergence** of a continuous multivariate function $f(x_1, x_2, \dots, x_n)$ is defined as

$$\Delta f = \nabla^2 f = \langle \nabla, \nabla f \rangle = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2} \quad (\text{Eq. 1})$$

where Δ is called the **Laplace operator**.

Write down an expression for the divergence of a discrete function $g[x, y]$.

Solution:

4.1.3 Q3 - lecture 03

Assume a discrete uni-variate function $g[x]$. What does the following operator do?

$$\text{AM}g[x] = \frac{1}{3}(g[x-1] + g[x] + g[x+1]) \quad (\text{Eq. 2})$$

Solution:

4.1.4 Q4 - lecture 03

Assume a discrete uni-variate function $g[x]$. What does the following operator do?

$$\text{WM}g[x] = \frac{1}{4}(g[x-1] + 2g[x] + g[x+1]) \quad (\text{Eq. 3})$$

Solution:

4.2 Sheet 1

4.2.1 Assignment 1a - Bias of an estimator

...

Solution:

Bibliography

- [1] „EXAMPLE: Finite Elemente - Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie “von Dietrich Braess, 5. Auflage, Springer, 2013