





# Liste der noch zu erledigenden Punkte

 Layout! . . . . .	2
 so richtig? . . . . .	2
 Wort? . . . . .	2
 Skizze . . . . .	2
 skipped: Very fast intro: Matlab and images . . . . .	2
 motions?P.4 . . . . .	3
 Matlab stuff . . . . .	4

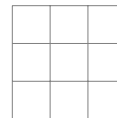
# Kapitel 1

## 1. Overview

- „image society“ (webpages: 1995 text-based, 2005 image based, 2015 video based ...)
  - data transfer rates  $\uparrow$ , compression rates  $\uparrow$
  - critical shift: reading  $\rightarrow$  watching
- „Photoshop“-ing (remove wrinkles, bumps, ...)
- Images in medicine („medical image processing“), x-ray, CT, MRI, ultrasound, ... („modalities“).  
different questions:

### 1.) Layout!

align bottom    measurments  $\stackrel{?}{\Rightarrow}$  image  
                  expl: tomography  
                   $\Rightarrow$  difficult mathematical problems



### 2.) Image enhancements

- denoising
  - simple pixels/lines: „sandpaper“ interpolation
  - global noise: smoothing
- grayscale
  - histogramm balancing (spreading)
- distortion
  - makes straight lines (in real world) straight (in the images)
- edge detection
  - contour enhancement
- segmentation
  - detect and separate parts of the image
- registration
  - sequence* of images of the same object  $\Rightarrow$  Wort?
  - , compare Skizze ↗ object following in a movie

so richtig?

### Our Focus:

- mathematical models/methods/ideas
- (algorithms)
- ((implementation))

skipped: Very fast intro: Matlab and images

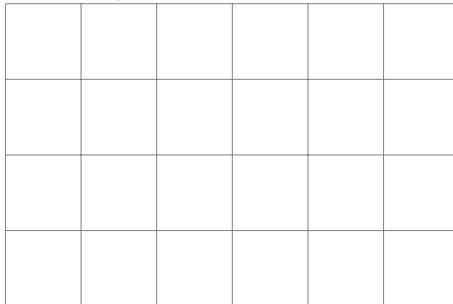
# Kapitel 2

## 2. What is an image?

### 2.1 Discrete and continuous images

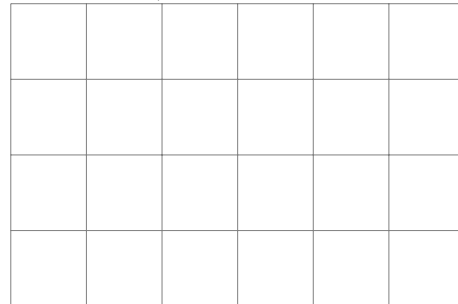
There are (at least) two different points of view:

• discrete/digital



**object:** matrix  
**tools:** linear algebra (SVD, ...)  
**pros:** (finite storage) storage, complexity  
**cons:** limitations: zooming, rotations, ...

• continuous/analogue



**function**  
 analysis (differentiation, integrate, ...)  
 freedom, tools, **motions? P.4**  
 (e.g. edge discontinuity)  
 storage (infinite amount of data)

arguably, one has:

- real life  $\Rightarrow$  continuous „images“ (objects)
- digital cameras  $\Rightarrow$  discrete images

In general we will say:

**Definition 2.1** ((mathematical) image). A (mathematical) *image* is a function

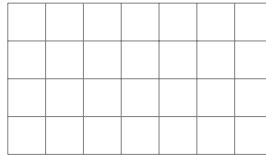
$$u : \Omega \rightarrow F,$$

where:  $\Omega \subset \mathbb{Z}^d$  (discrete) or  $\Omega \subset \mathbb{R}^d$  (continuous) ... *domain*  
 $d = 2$  (typical case 2D),  $d = 3$  („3D image“=body or  $\underbrace{2D+time}_{\text{movie}}$ )  
 $d = 4$  (3D + time)

$F$  ... *range of colours*

$F = \mathbb{R}$  or  $[0, \infty]$  or  $[0, 1]$  or  $\{0, \dots, 255\}$ , ... grayscale (light intensity)  
 $F \subset \mathbb{R}^3$  ... RGB image (colored)

$F = \{0, 1\}$  ... black/white



3 Layers  
 $\Rightarrow$  colored images:w

### Matlab stuff

Large parts of the course: analytical approach (i.e. continuous domain  $\Omega$ )  
 Since we want to differentiate, ... the image  $u$ .

Still: need to assume that also  $F$  is continuous (not as  $\{0, 1\}$ ,  $\{0, 1, \dots, 255\}$  or  $\mathbb{N}$ )  
 since otherwise the only differentiable (actually, the only continuous) functions  $u : \Omega \rightarrow F$  are  
*constant* functions  $\Leftrightarrow$  single-colour images

Also: We usually take  $F$  one-dimensional ( $F \subset \mathbb{R}$ ). Think of it as either

- gray scaled image, or
- treating R,G & B layer separately

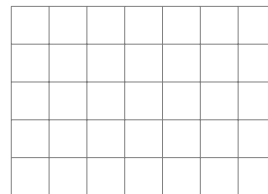
## 2.2 Switching between discrete and continuous images

**continuous  $\rightarrow$  discrete:**

- divide the continuous image in small squared pieces (boxes) (superimpose grid)
- now: represent each box by *one* value

- strategy 1: take function value  $u(x_i)$   
 for  $x_i = \text{midpoint of box } B_i$
- strategy 2: use mean value

$$\frac{1}{|B_i|} \int_{B_i} u(x) dx$$



$\Rightarrow$  discrete image

strategy 1: simple (and quick) but problematic ( $u(x_i)$  might represent  $u|_{B_i}$  badly; for  $u \in L^p$ , single point evaluation not even defined)

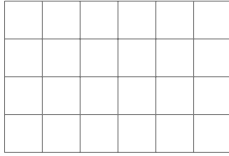
strategy 2: more complex but also more „democratic“ (actually closer to the way how CCD Sensors in digital cameras work)

often the image value of the box  $B_i$  gets also digitized, i.e. fitted (by scaling & rounding) into range  $\{0, 1, \dots, 255\}$

**discrete  $\rightarrow$  continuous**

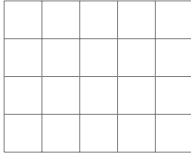
This is of course more tricky ...

- Again: each pixel of the discrete image corresponds to a „box“ of the continuous image (that is still to be constructed)
- Usually: pixel value  $\mapsto$  function value at the *midpoint* of the box
- Question: How to get the other function values (in the box)?



idea 1: just take the function value of the nearest midpoint („nearest neighbour interpolation“)

For each  $x \in B_i : u(x) := u(x_j)$  where  $|x - x_j| = \min_k |x - x_k|$



- $\Rightarrow u(x) = u(x_i)$  for all  $x \in B_i$
- $\Rightarrow$  each box is uni-color
- $\Rightarrow$  the continuous image is essentially still discrete

idea 2: (bi-) linear interpolation