Deep Learning for Visual Computing (COMP0169)

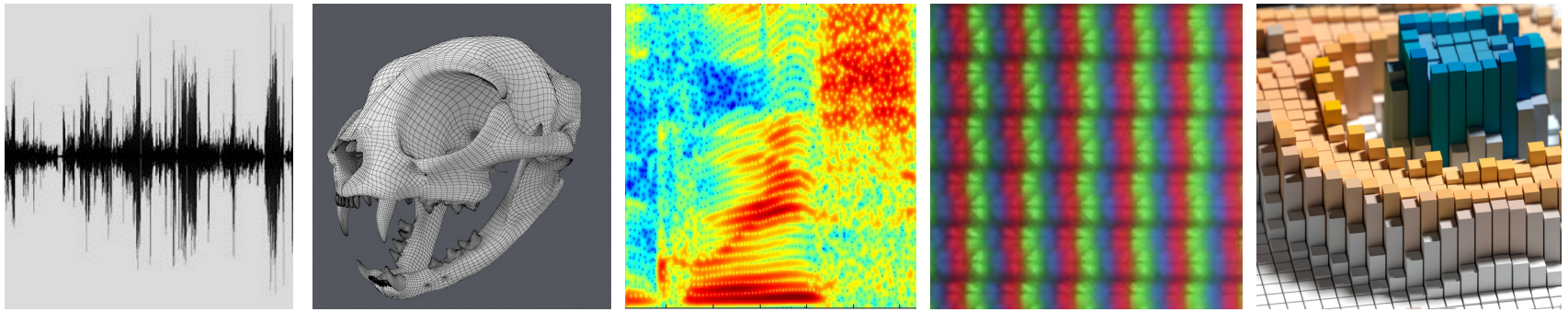
Visual Representation and Processing

Niloy Mitra **Tobias Ritschel**

UCL

Representation & Processing

- Often, the key to **processing** *(Part 2)* using a machine, also in ML, is choosing the right **representation** *(Part 1)*



Challenge

- Media
  - Images
  - Videos
  - Sounds
  - 3D Surfaces
  - .. and beyond (stereo images, light field, light field video, etc)
- We want to
  - Analyse,
  - Change and
  - Generate them using ML techniques.

Images/Video/Audio/X Commonalities

- What are they? Physically?
- Setting scope (what do we ignore)
- How are they captured?
- How are they stored?
- How are they reproduced?
- How are they processed

- Particular useful to see this symmetry when doing ML

Part A: Representation

Part B: Processing

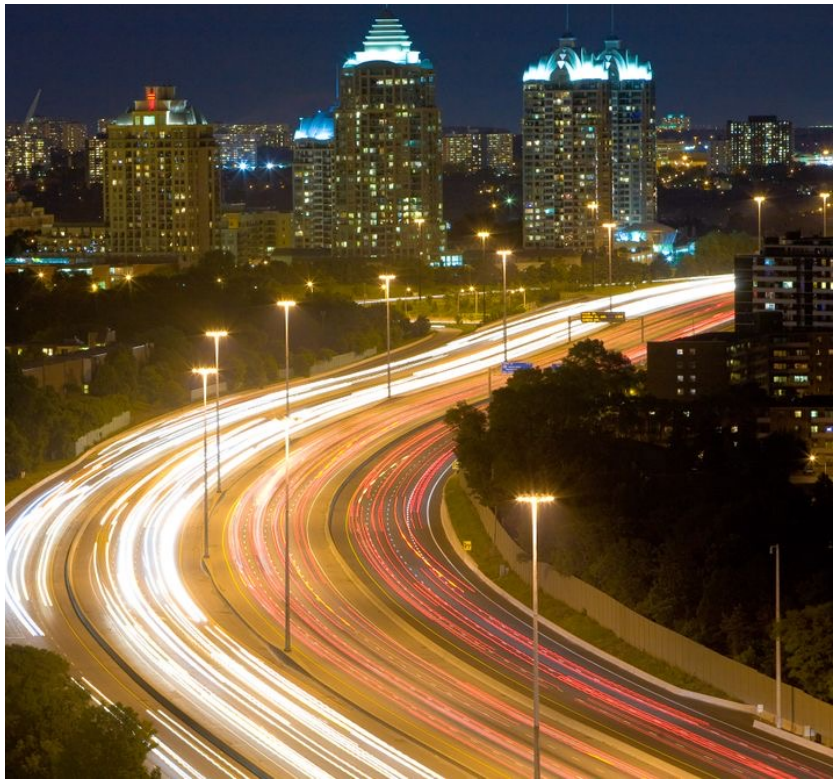
Image representation

Images = Light

- What we mean is actually **radiance**
- Radiance is to say
  - This unit piece of area
  - at that position
  - and in this unit solid angle,
  - for this unit amount of time,
  - and for this unit part of the spectrum, was
  - sending so-and-so-many photons.

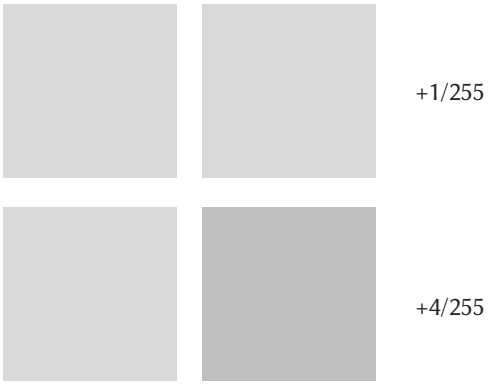
How do humans see light?

- We need to look at sensitivity to
  - Intensity
  - Chroma(Color)
  - Spatial frequency
- This informs how we process it



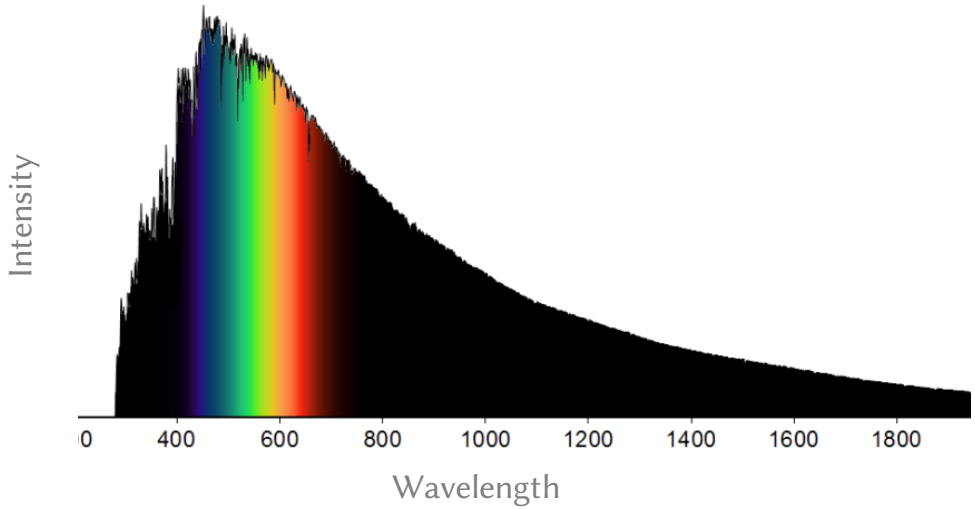
Intensity sensitivity

- There is only so many levels of grey you can tell apart
- Likely 100+
- So in images we play safe and say
  - 256 is twice that, and
  - 256 is 8 bit
- So we use 8 bit to store intensity



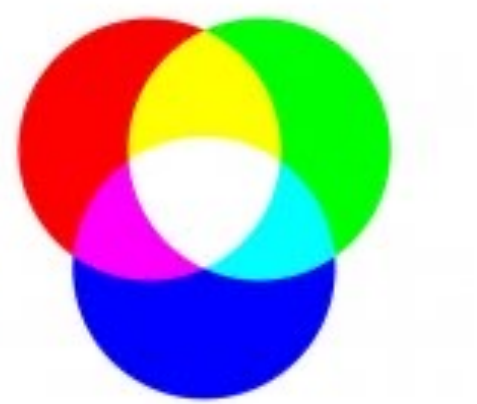
Chromatic sensitivity

- Light has an arbitrary spectrum
- We only see a part of that



Color spaces: RGB

- For processing
- There are many others
- Only one sRGB
- Additive
- Imagine three lamps of spectrum R, G and B

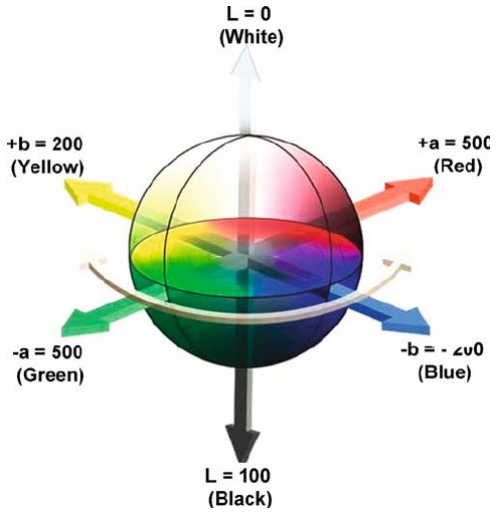


Color spaces: CMYK

- For reproduction: print, film etc
- Multiplicative
- Imagine three glass filters reducing light

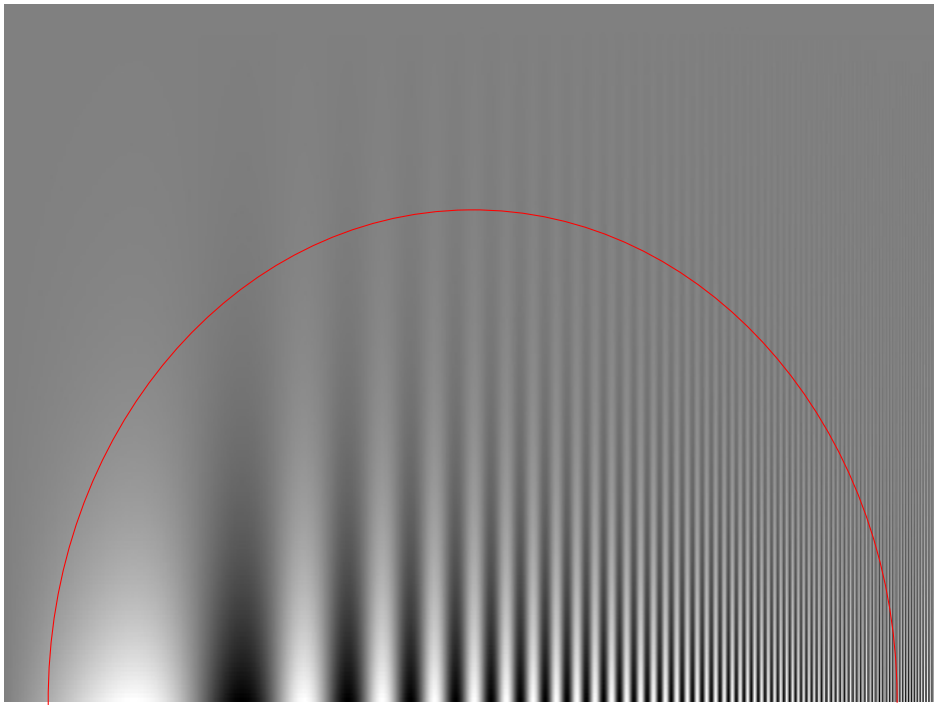
Color spaces: LMS, Lab, YCrCb

- For editing and intuitive control
- Change brightness, saturation, hue, such semantic things
- Formally: decorrelation



Spatial sensitivity 空间敏感度

- Small details cannot be seen
- Depends on contrast
- Spatial contrast sensitivity function



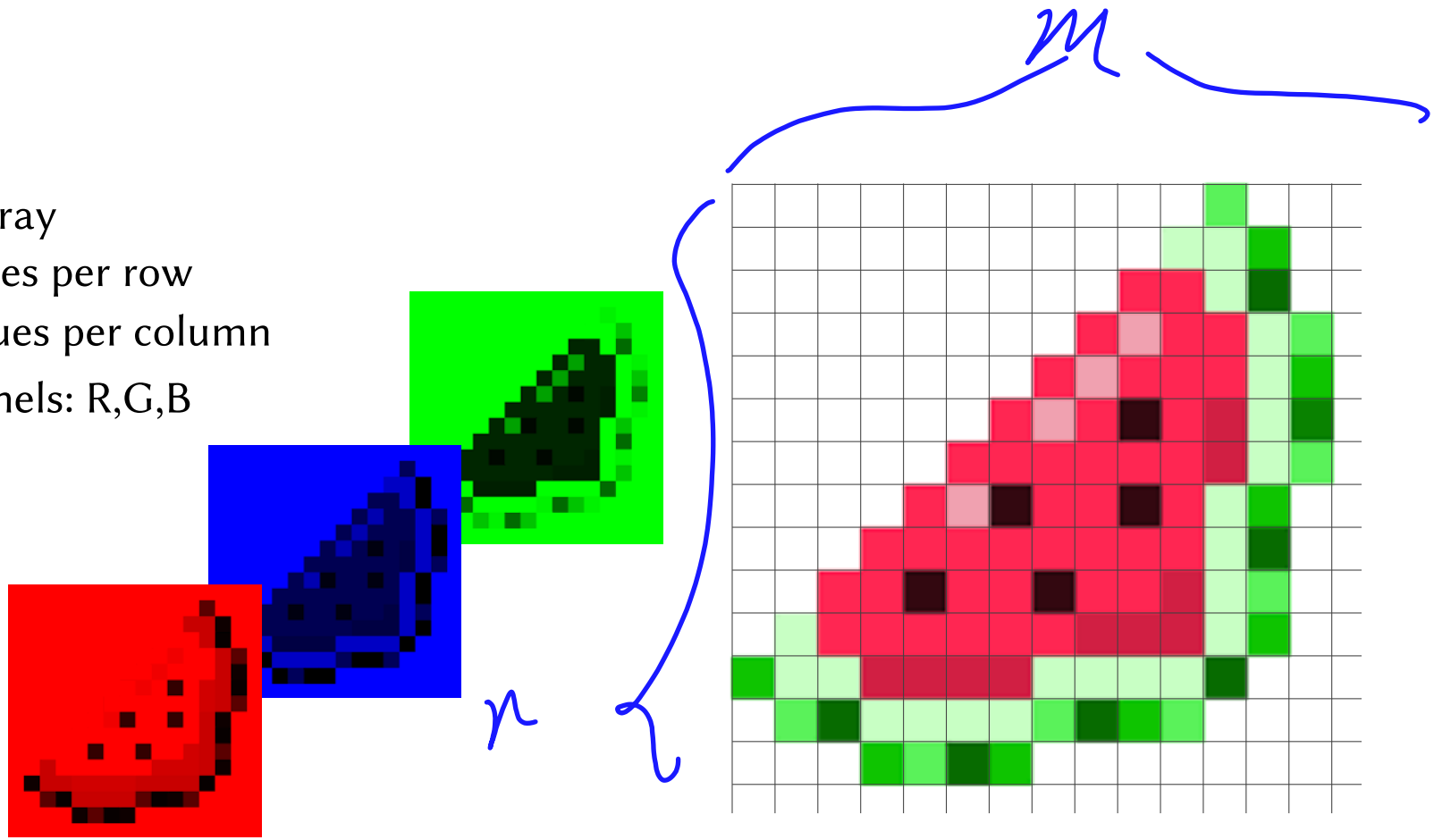
Digital images

- Not all wavelengths, just three
- Not all spatial details, but some minimum, say a 1/5000 of your visual field
- Not all intensity levels, just some, say 256

- So: Three arrays of 5000x5000 values with 8 bit values

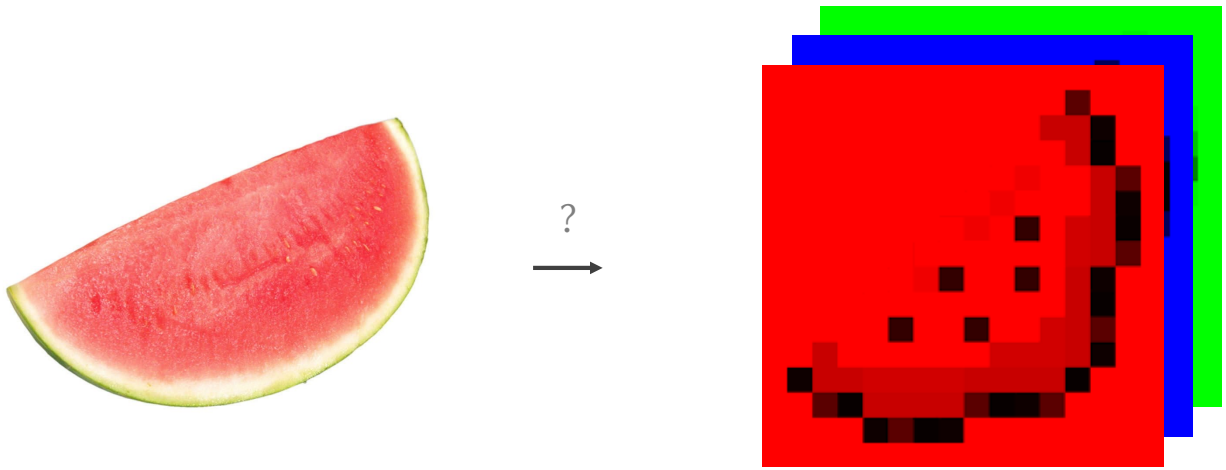
Pixels

- Just an array
  - n values per row
  - m values per column
- In 3 channels: R,G,B

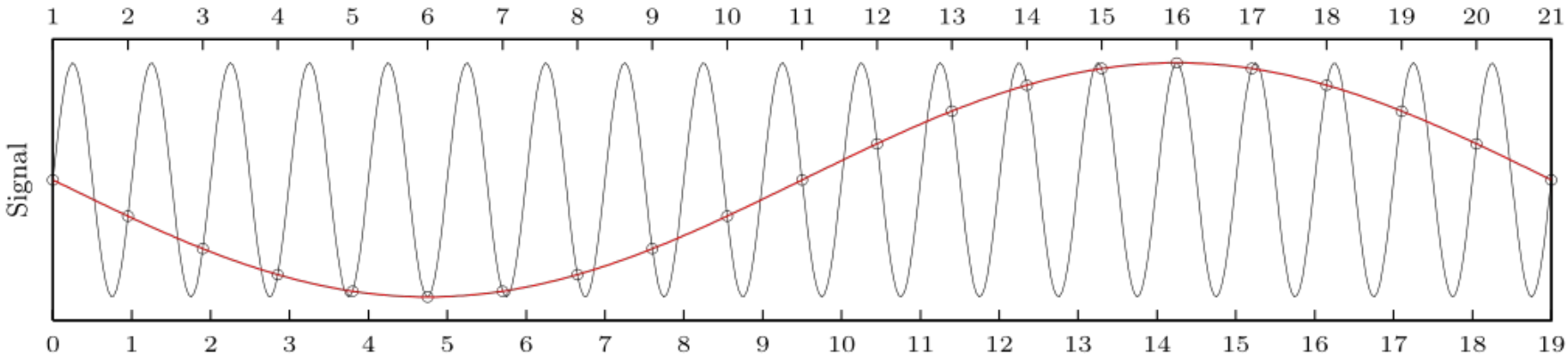


Sampling

- Sampling: Turn image into the pixel array
- CMOS/CCD etc do the photon counting, but what next?

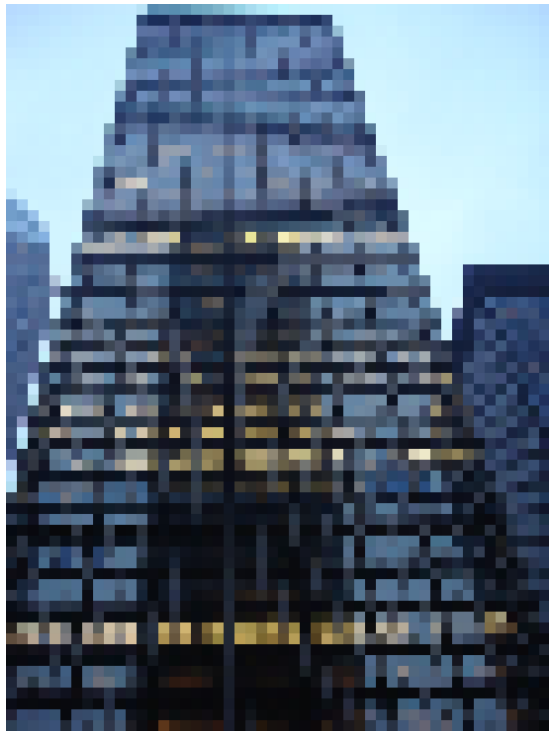


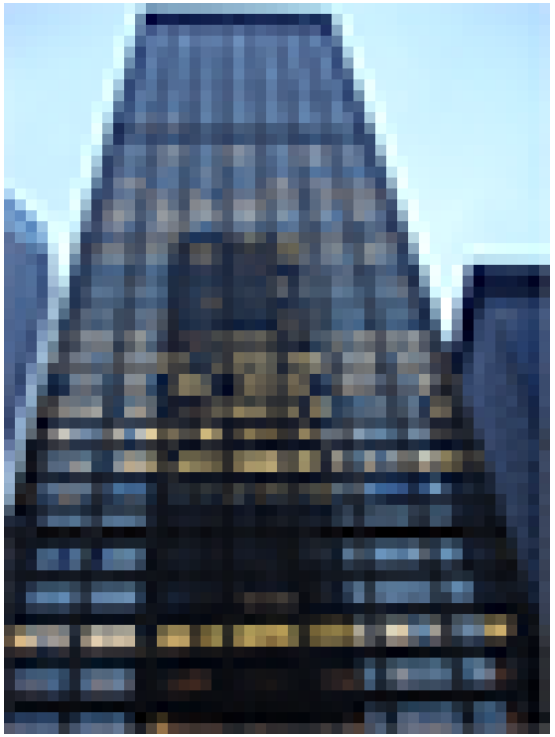
Sampling and Aliasing



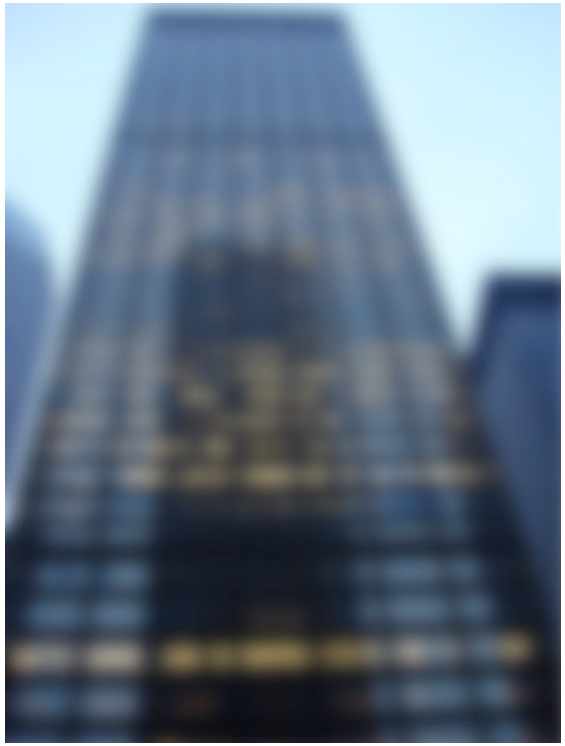
Source: Wikipedia

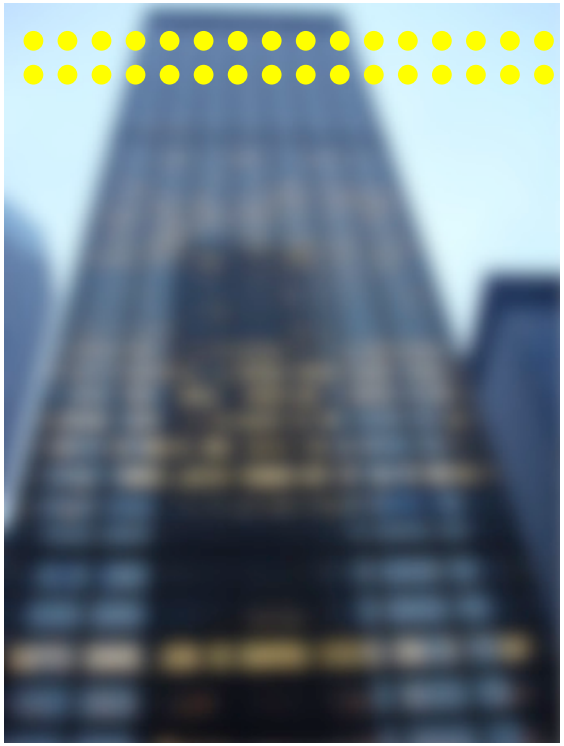
Aliasing in a real image

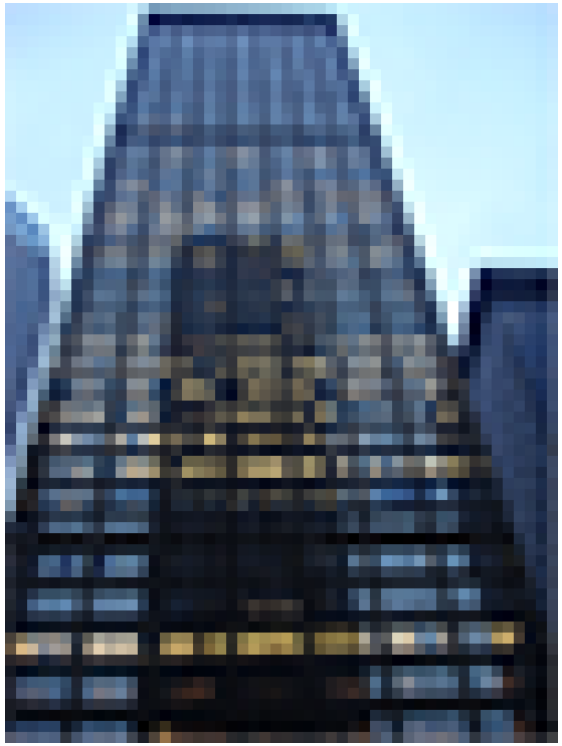




Pre-filter





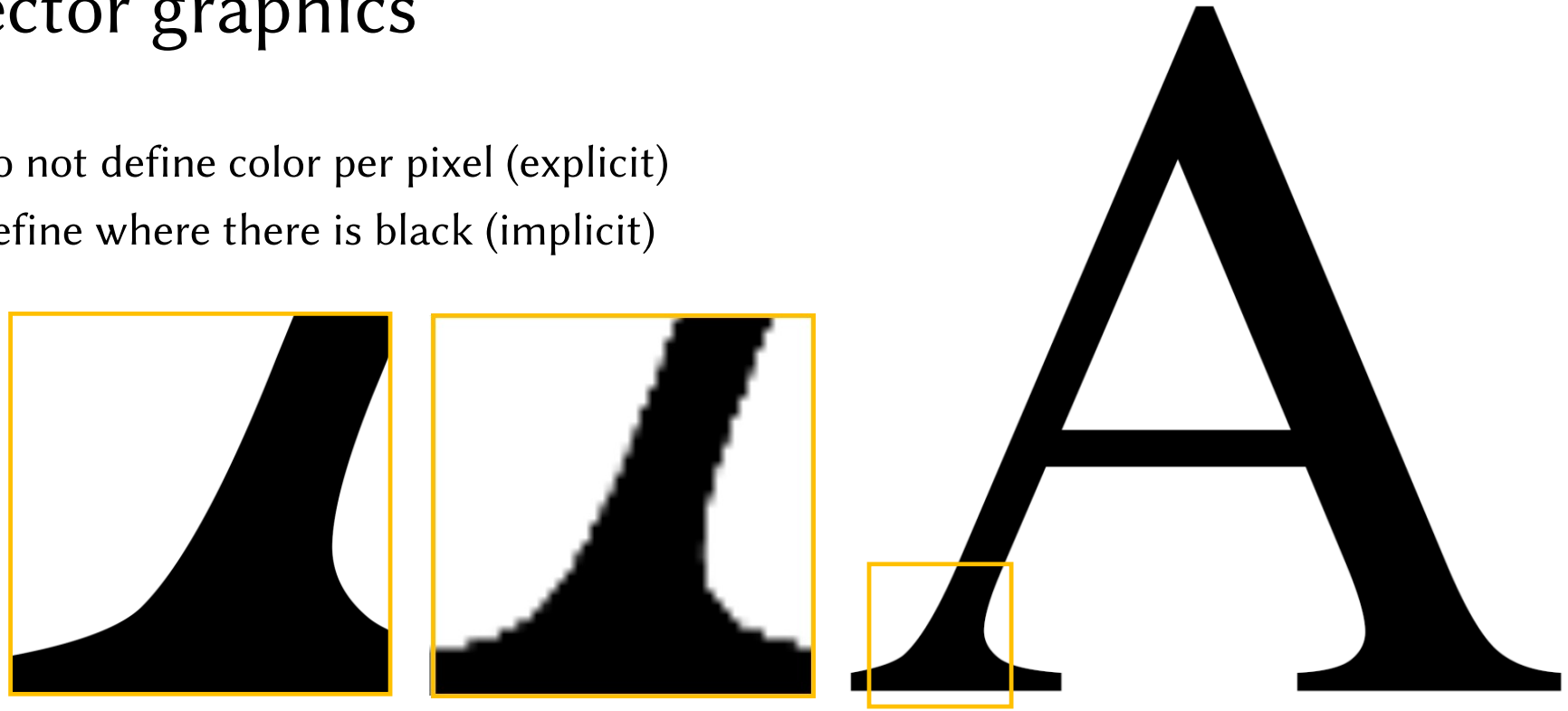


Solution

- **First**: Remove all frequencies higher than half the sampling frequency
- **Second**: Sample the function

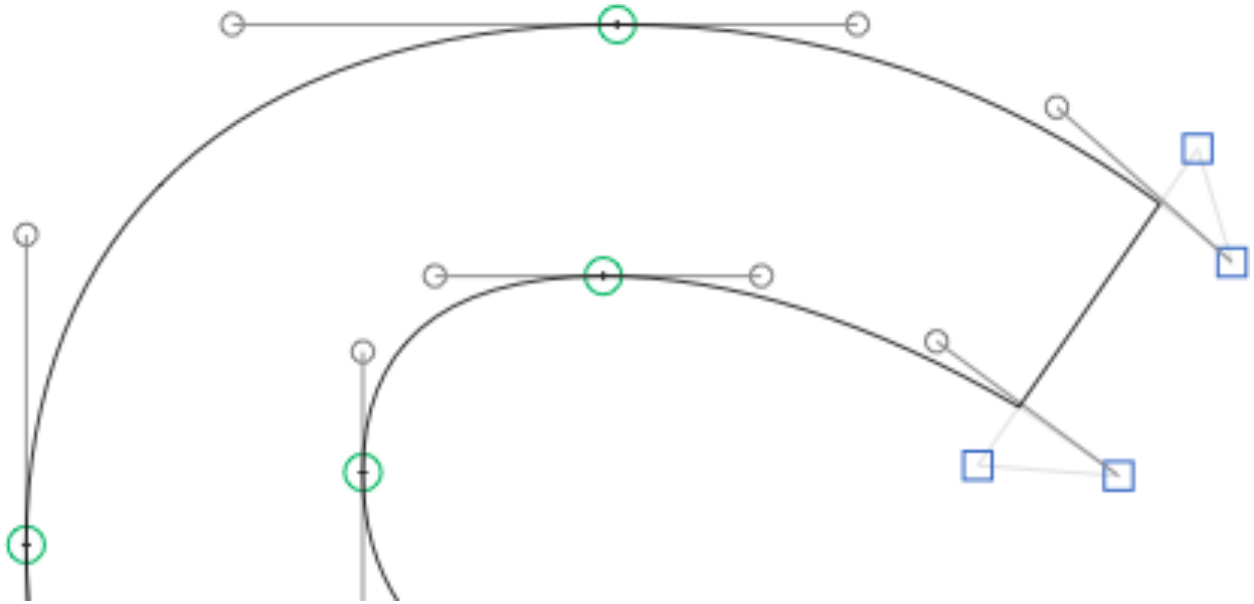
Vector graphics

- Do not define color per pixel (explicit)
- Define where there is black (implicit)



Vector graphics

- Defined via control points

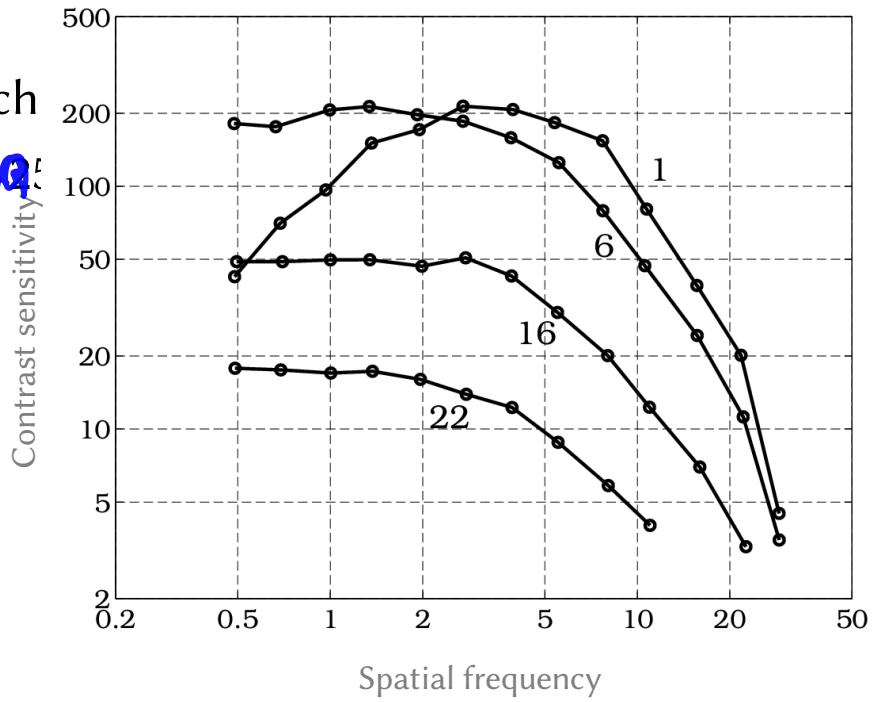


Video representation

How do humans see motion?

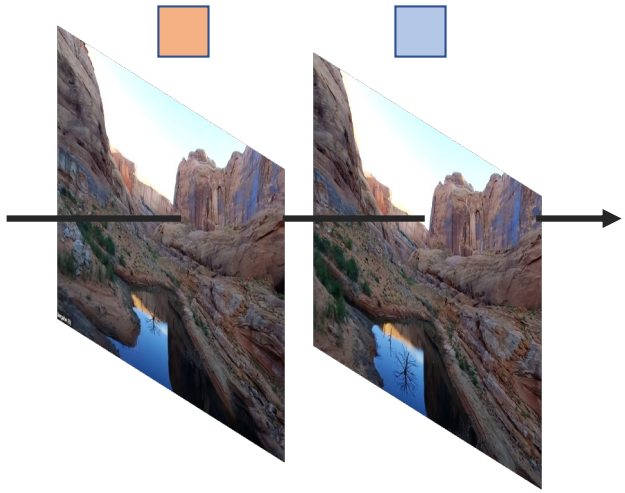
- Temporal contrast sensitivity
- Humans cannot see arbitrary fast ch
- Limit temporal sample rate, e.g. to

25 Hz



Optical flow

- While it is easy to just make 2D+time a 3D space it is not making sense
- Trouble is the alignment?




dergabe (k)


dergabe (k)

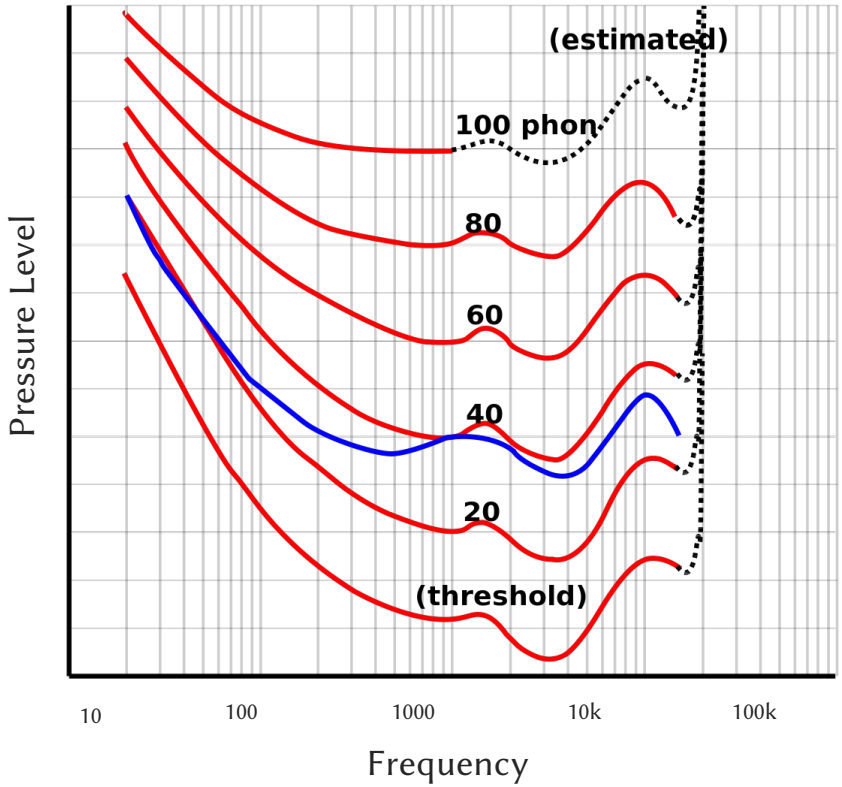
Video compression & file format

- MPEG
  - Same as JPEG
  - Do only store some frames
  - Store optical flow between them

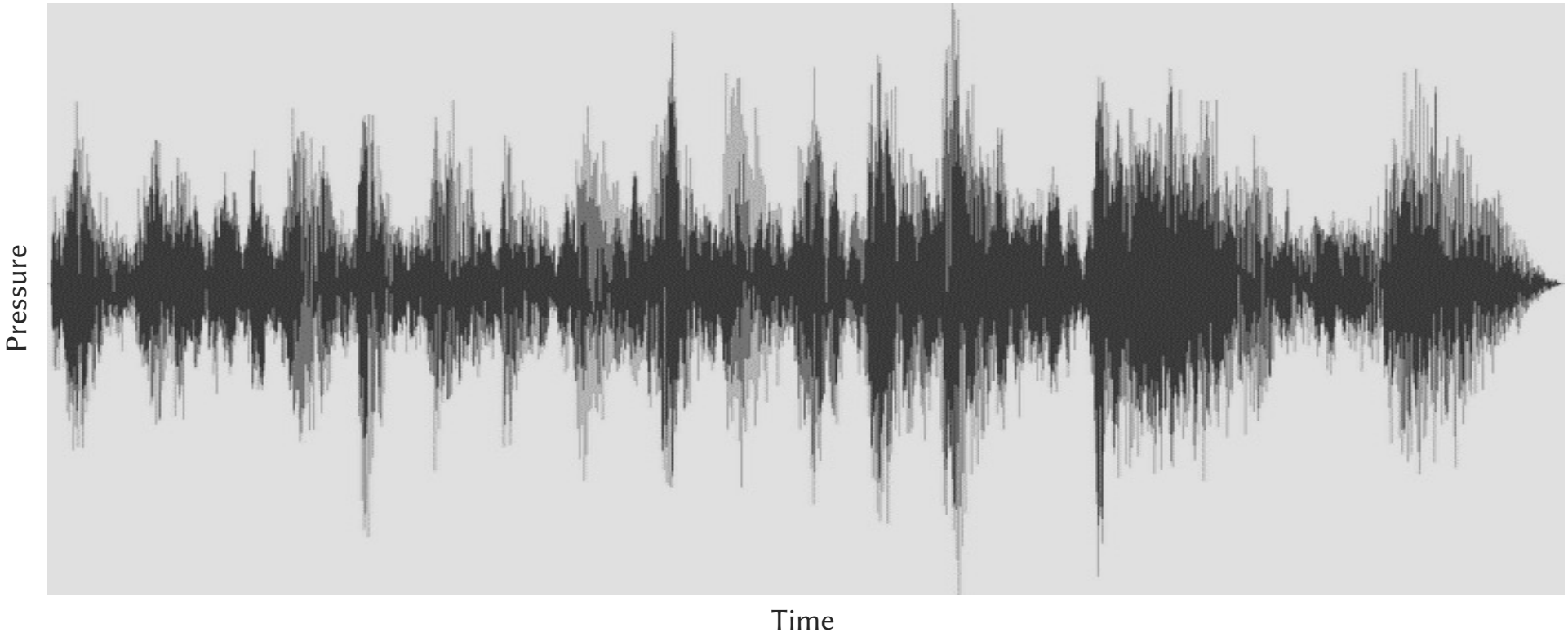
Audio representation

What is audio?

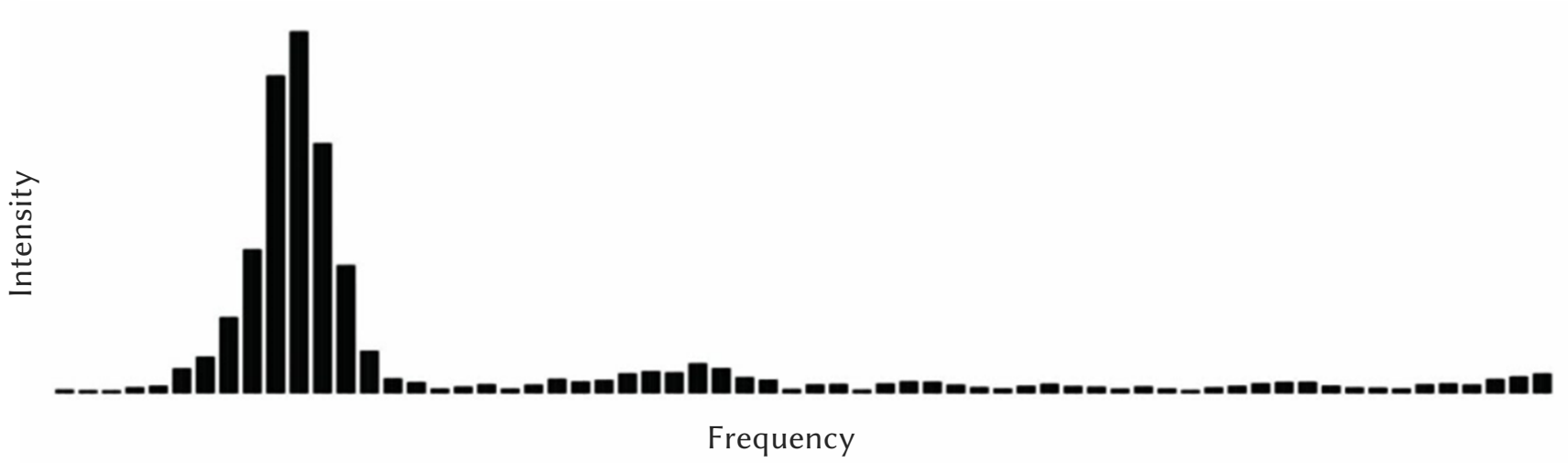
- Rapid changes in air pressure
- Dezibel: Log of a ratio
- Phon: Equal loudness



Waveform

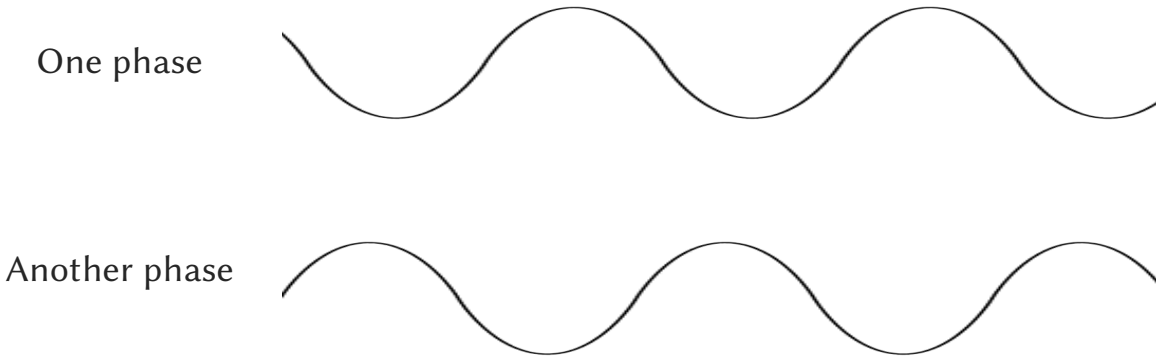


Spectrum

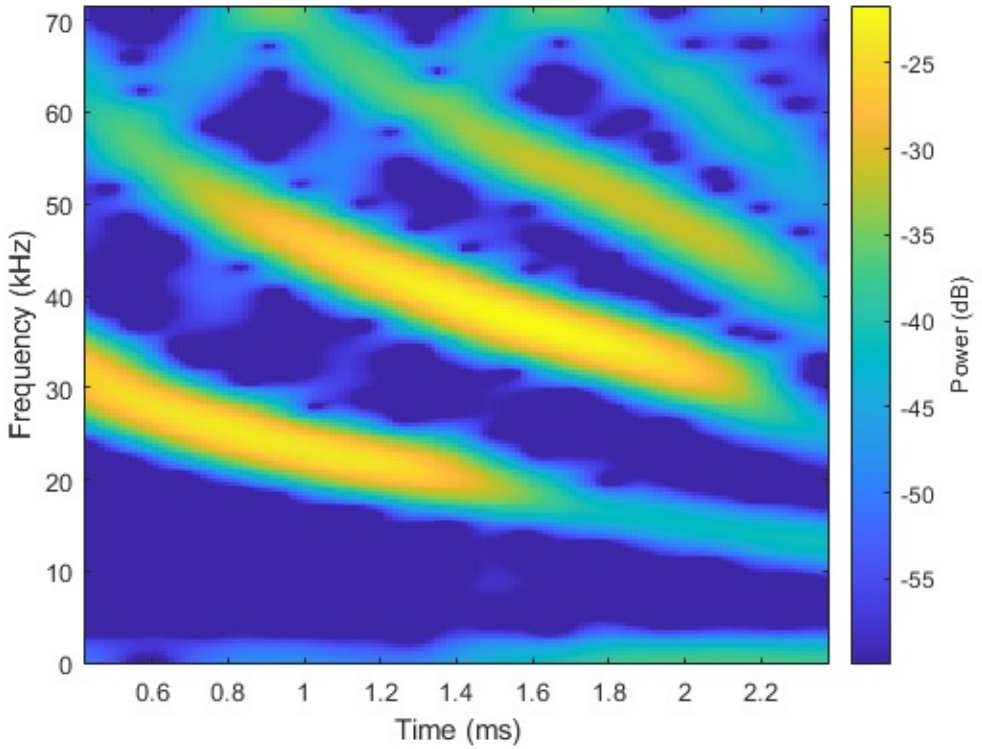


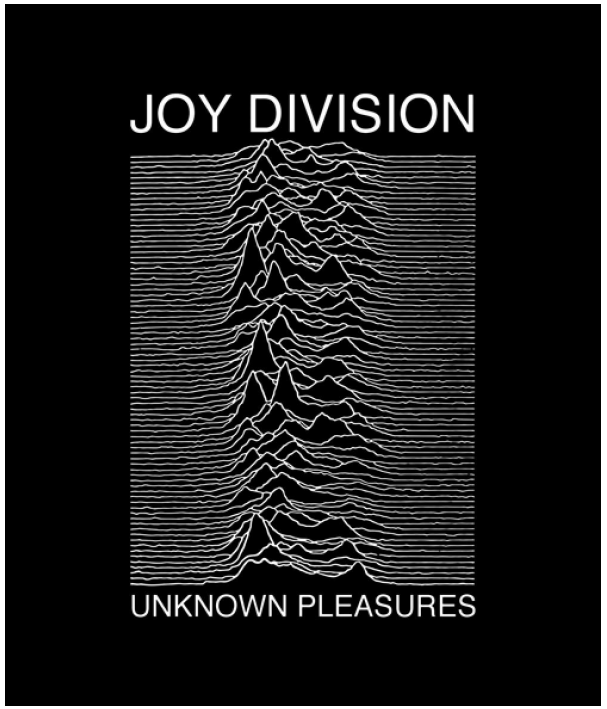
Phase matters

- Signals do not only differ by amplitude, also by phase

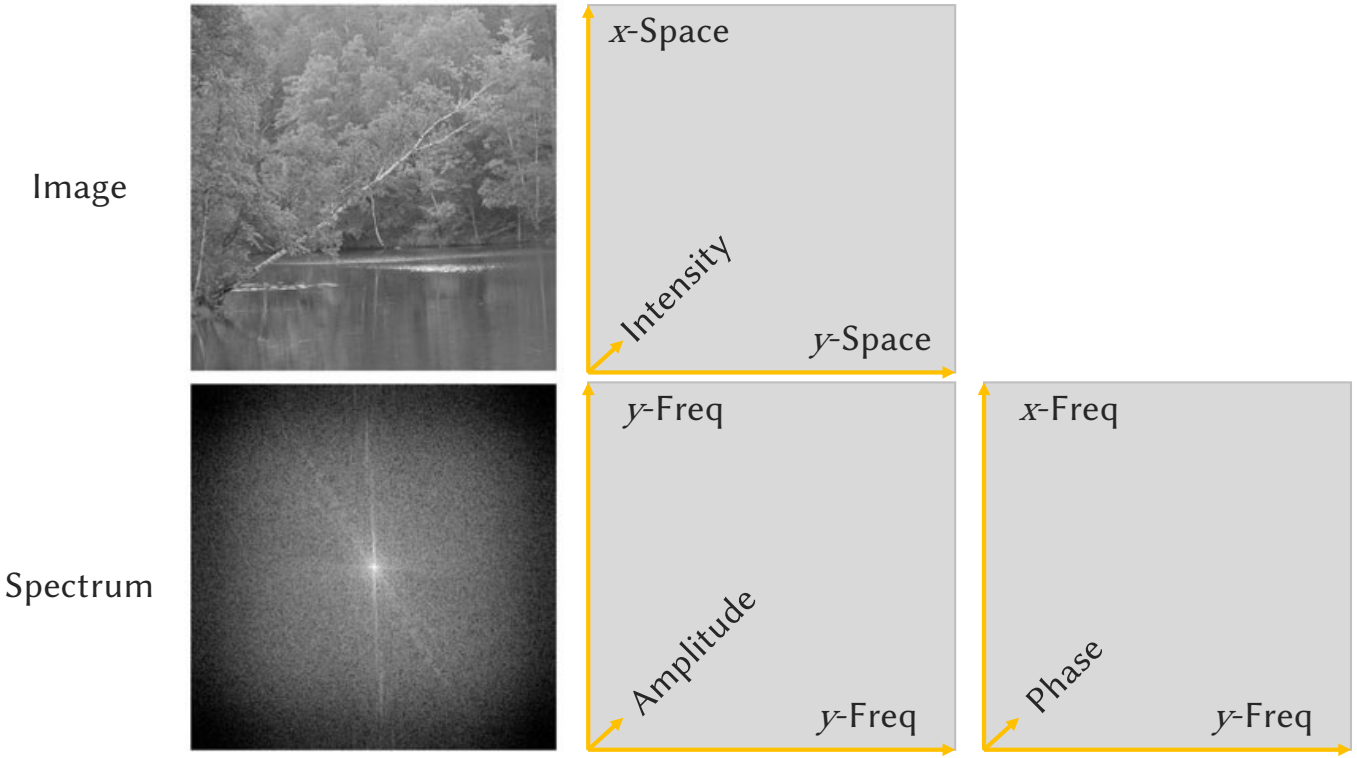


Time-Frequency



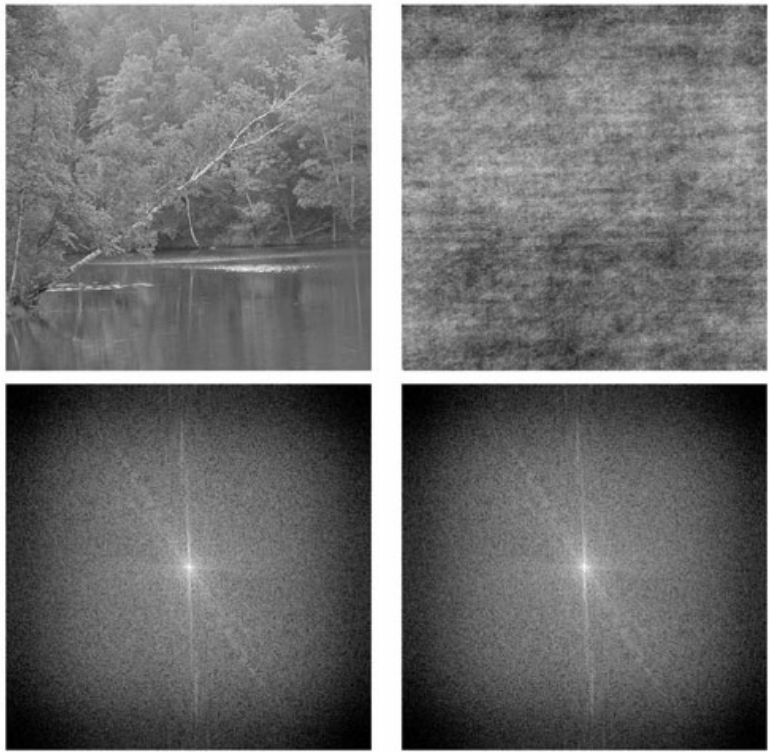


Works on images, too



Works on images, too

Correct phase



Incorrect phase

Spatial audio

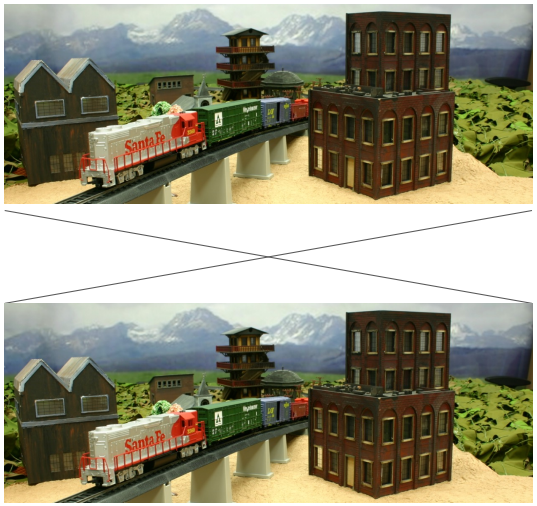
- **Stereo**: Typically store two channels
- **Spatial audio**: Store audio signal for n different positions

risd.edu

.. and representing the beyond

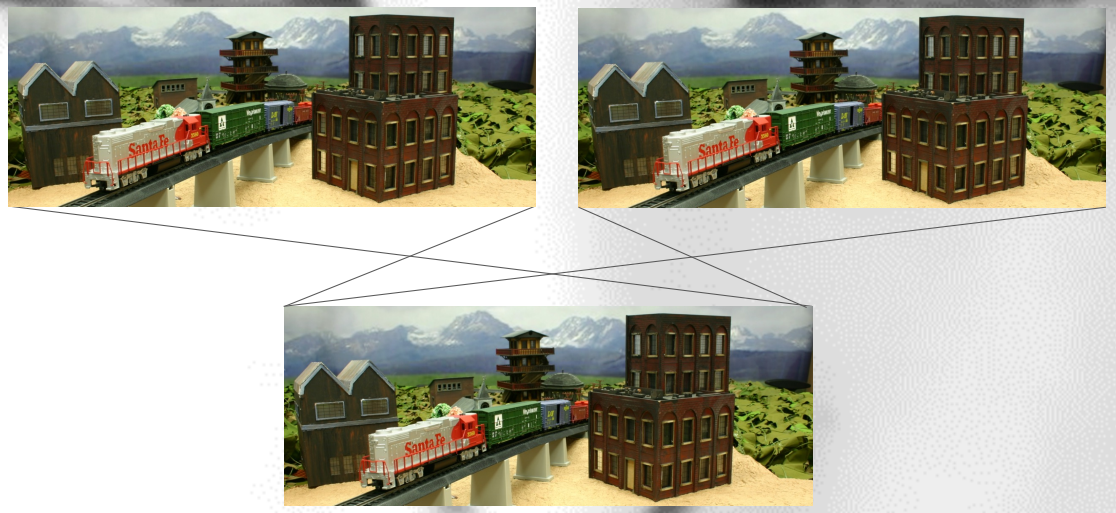
Stereo

- Mono image: One image of one scene



Stereo

- Stereo image: Two images of one scene



Stereo images

Stereo images

Stereo Disparity

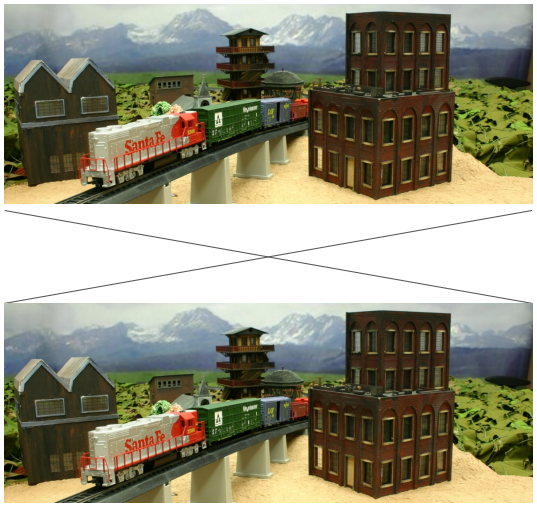
- Note how this is not just a shift
- Different parts move differently
- This difference in motion is **disparity**
- It this gives us the idea about depth

Stereo representation

- Either pare of RGB images
- Or one RGB+Depth image

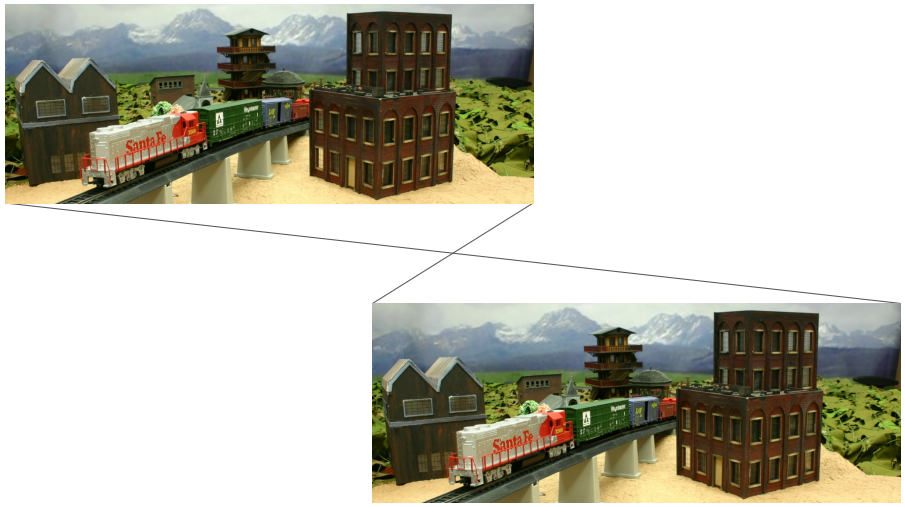
Light field

- Image, but not with one view point
- All rays that pass a scene



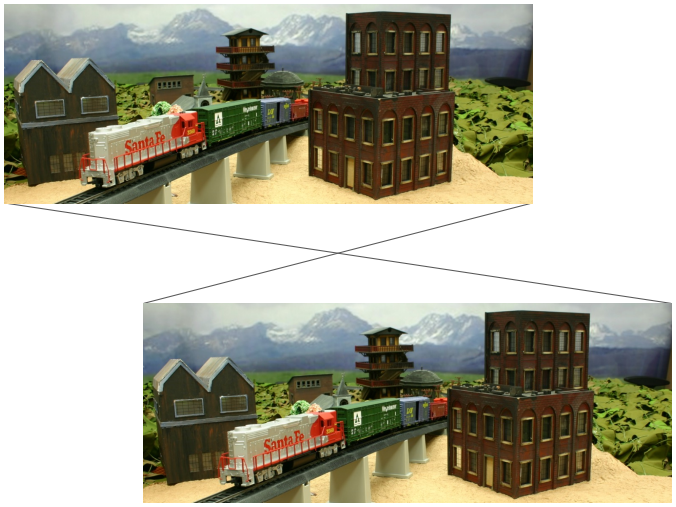
Light field

- Image, but not with one view point
- All rays that pass a scene



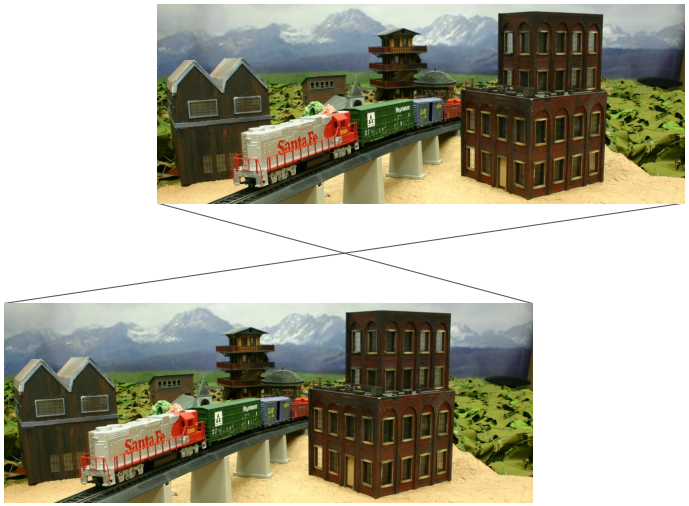
Light field

- Image, but not with one viewpoint
- All rays that pass a scene



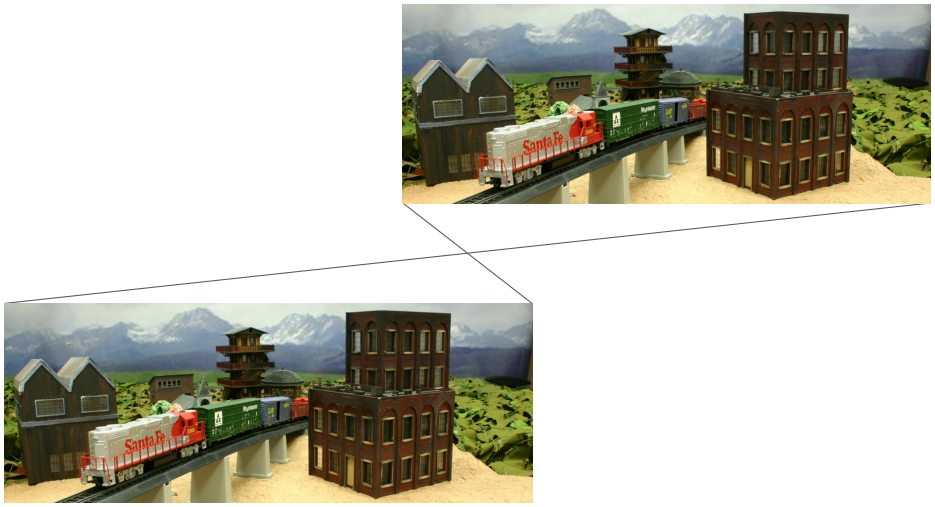
Light field

- Image, but not with one viewpoint
- All rays that pass a scene

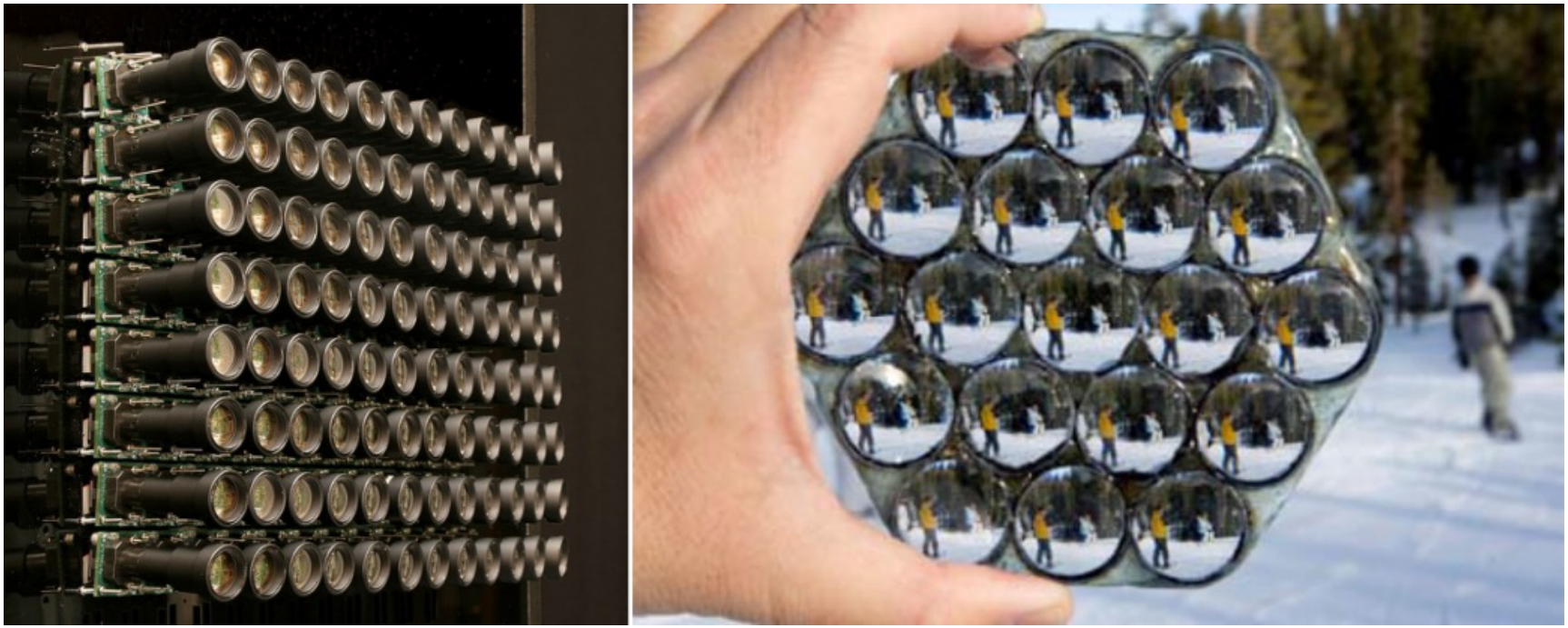


Light field

- Image, but not with one viewpoint
- All rays that pass a scene

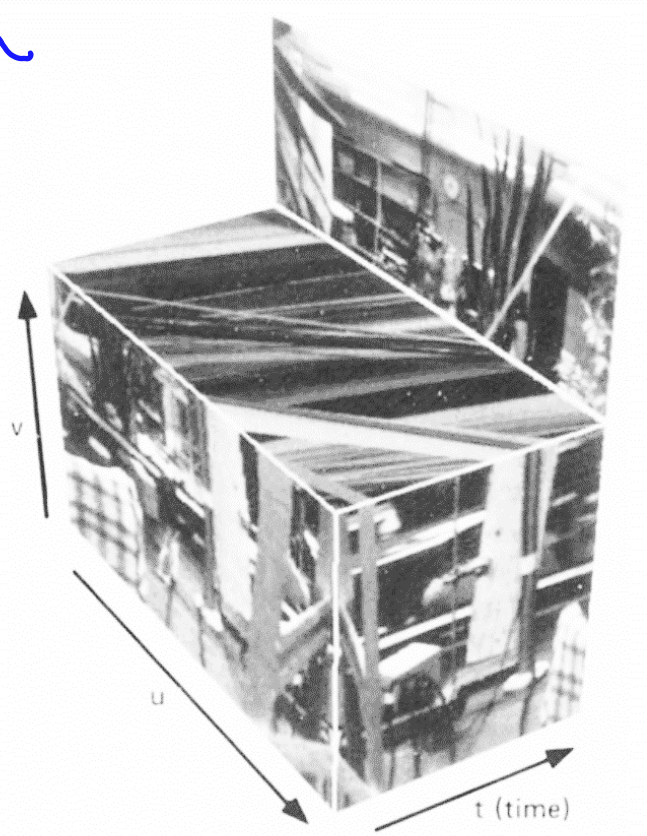


Light field capture



Epipolar image

- Bolles et al. 1986
- Third axis is not space, but time or angle



Light field representation

- Stack of images
- Stereo if the special case for n=2

Stereo and Light field Video

- The same thing, just with time

Main observation for “beyond”:

- Increasing redundancy
- Great opportunity for ML!

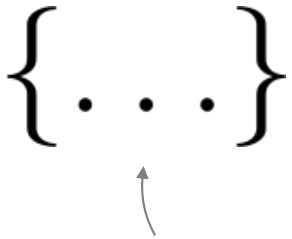
3D Surfaces

UCL

3D surfaces options

- 3D volumes
- 3D parametric surfaces
- 3D Point clouds
- 3D Triangle meshes

- All of them are sets of points, not neccesarily discrete



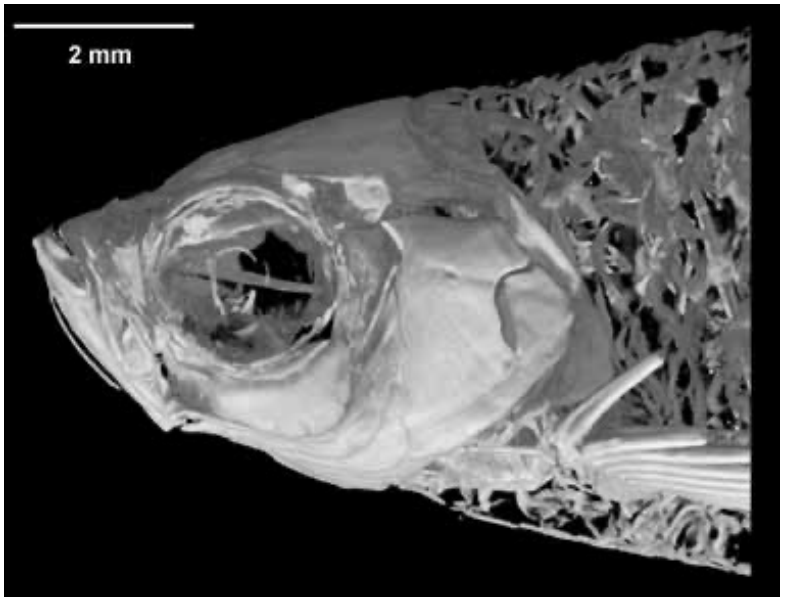
3D volumes

- Mapping from a 3D coordinate to a scalar value
- That is not really a “surface”, but is 3D

$$\mathbb{N}^{n \times m} \rightarrow \mathbb{R}$$

3D volumes: Example

- Zebra fish from Digimorph

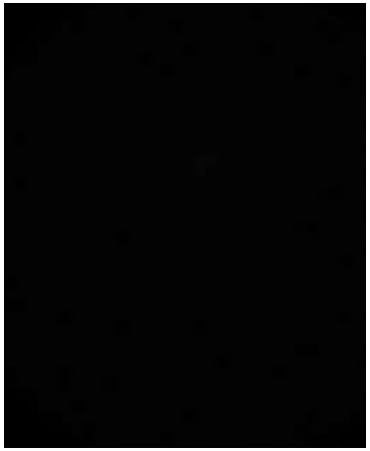


3D volumes: Example

- Zebra fish from Digimorph

Front to back

Top to bottom



Left to right

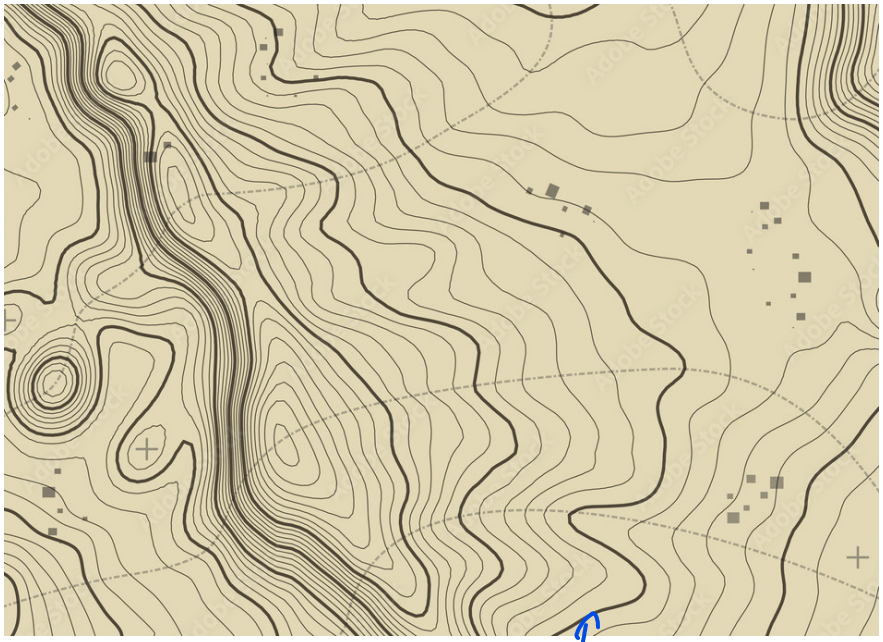
Iso-surfaces

- The set of all 3d points for which the scalar field takes the value c

$$\{\mathbf{x}, f(\mathbf{x}) = c, f \in \mathbb{R}^3 \rightarrow \mathbb{R}\}$$

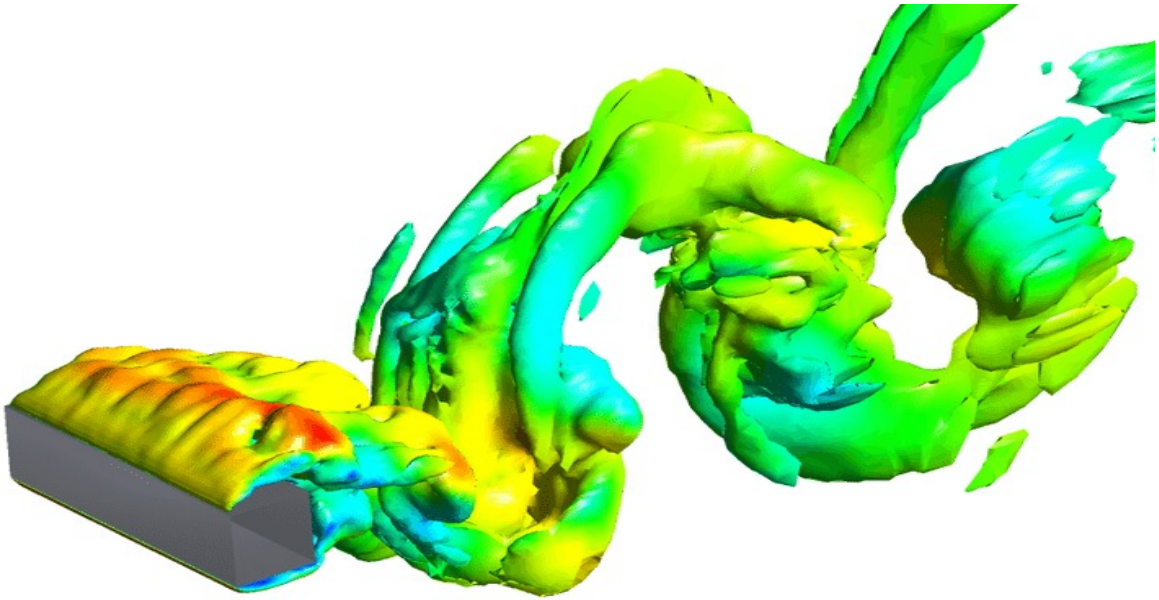
Iso-surfaces: Example in 2D

$$\{\mathbf{x}, f(\mathbf{x}) = c, f \in \mathbb{R}^3 \to \mathbb{R}\}$$



Iso-surfaces: Example in 3D

$$\{\mathbf{x}, f(\mathbf{x}) = c, f \in \mathbb{R}^3 \to \mathbb{R}\}$$



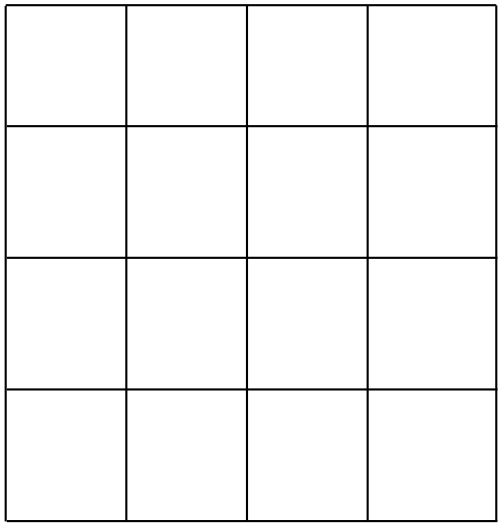
Parametric surfaces

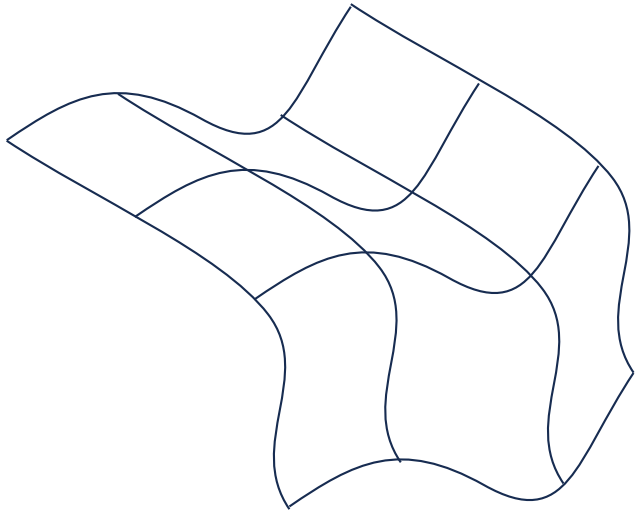
- Mapping from a parameter vector $\mathbf{x}$ to a coordinate $f(\mathbf{x})$

$$\{f(x), f \in \mathbb{R}^2 \rightarrow \mathbb{R}^3\}$$

Parametric surfaces

$$\{f(x), f \in \mathbb{R}^2 \rightarrow \mathbb{R}^3\}$$





Parametric surfaces

- Best suited for technical objects

3D point clouds

- Simply a list if 3d points

$$\{\mathbf{x}_i \in \mathbb{R}^3\}$$

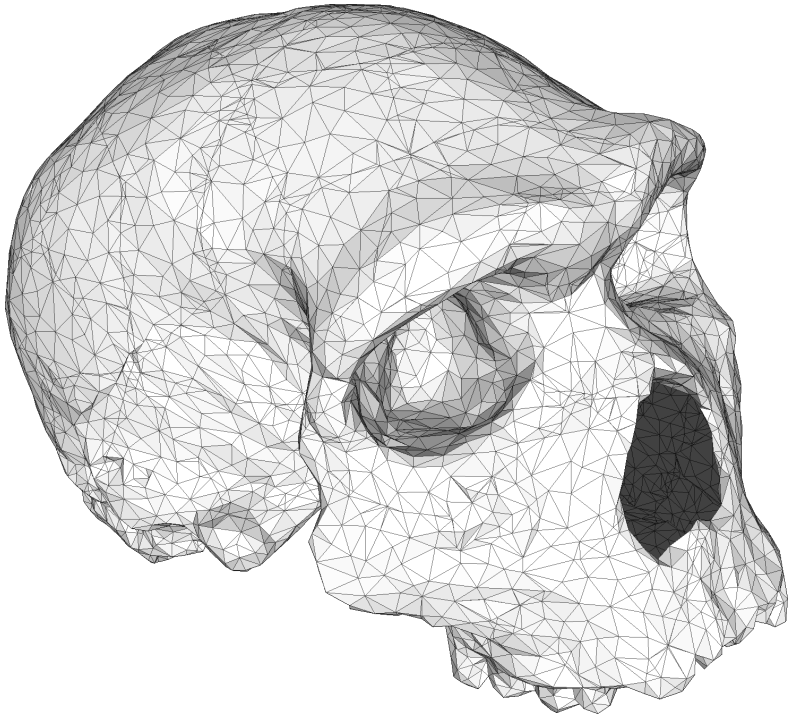
Comes out of 3D scanners

3D triangle meshes

- All 3D points that are part of one of many triangles

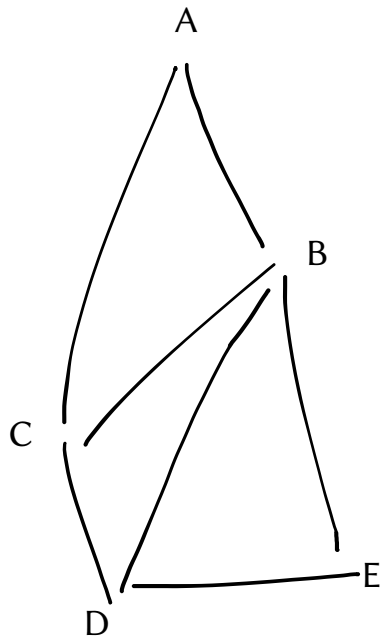
$$\{\mathbf{x}, \exists i, \mathbf{x} \in \triangle_i\}$$

3D triangle mesh example



3D triangle meshes are graphs

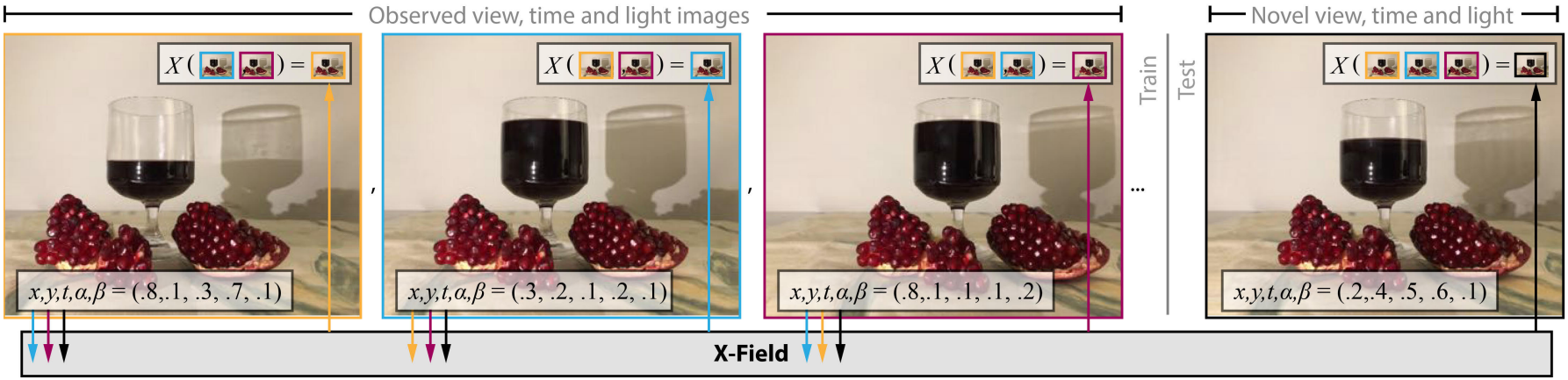
- Graphs are made from vertices and simplices
  - The vertices are 3d point
  - The simplices are tuples of indices into the list of vertices
- Example
  - ACB, BDC, DEB



Summary

	Continuous	Discrete
Image	$\mathbb{R}^2 \rightarrow \mathbb{R}^3$	$\mathbb{N}^{n \times m} \rightarrow \mathbb{R}^3$
Video	$\mathbb{R}^2 \rightarrow \mathbb{R}^3$	$\mathbb{N}^{n \times m \times t} \rightarrow \mathbb{R}^3$
Audio	$\mathbb{R}^1 \rightarrow \mathbb{R}$	$\mathbb{N}^n \rightarrow \mathbb{R}$
Light field	$\mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}^3$	
Light field video	$\mathbb{R}^{2 \times 2+1} \rightarrow \mathbb{R}^3$	
3D surfaces		

X-Fields



- A bit of shameless ad now: https://xfields.mpi-inf.mpg.de/
- Very related to questions of representation
- Choosing them right + NN = great results

Part A: Representation

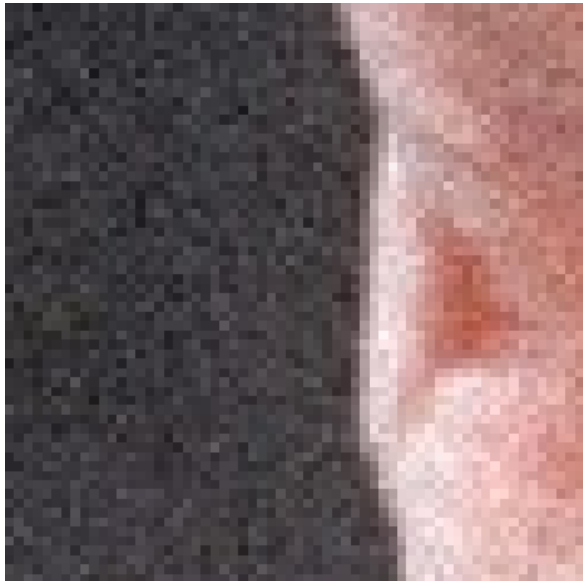
Part A: Processing

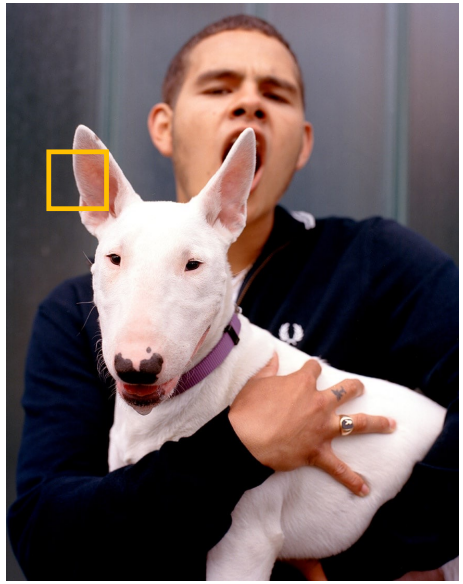
Processing overview

- Denoising as a model task
- Finding edges
- Working in multiple resolutions

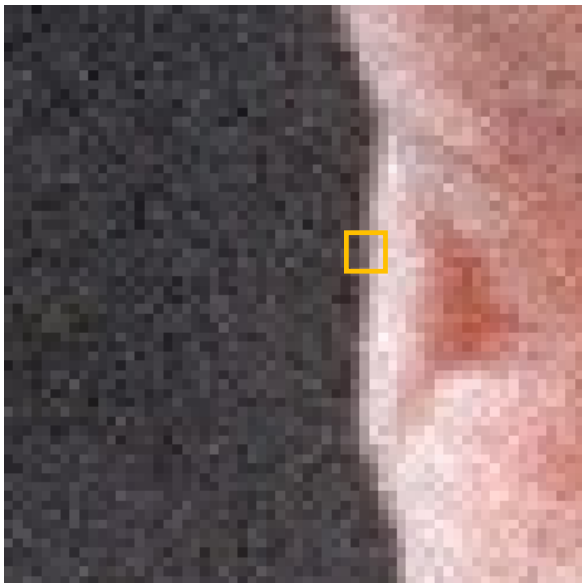
Denoising

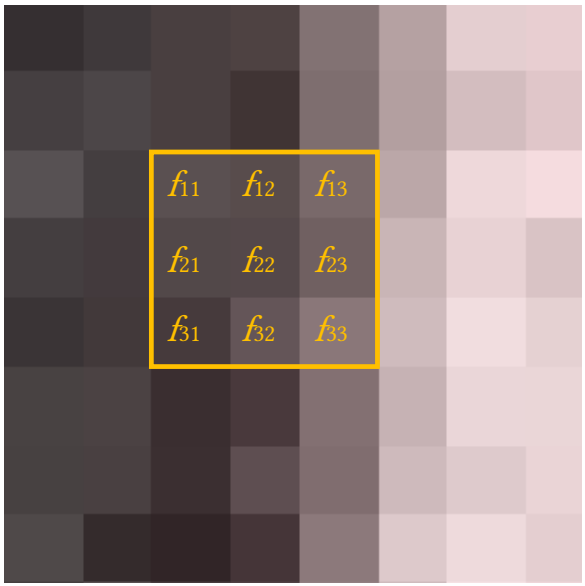
Denoising





Mean filter





Three neighbours

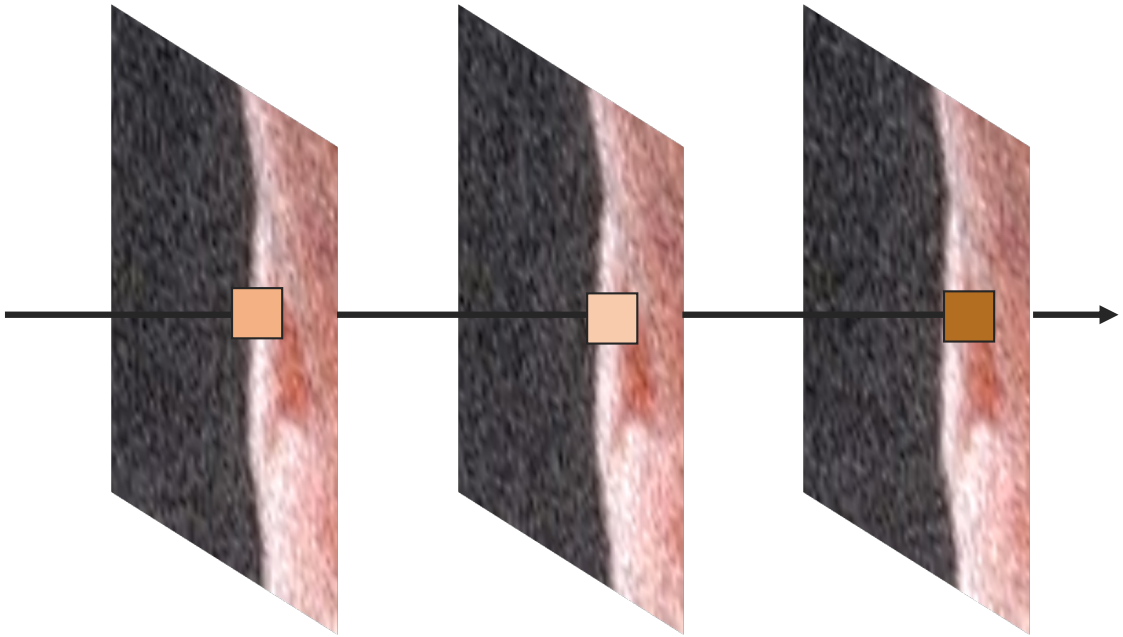
$$\frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3} f_{ij}$$

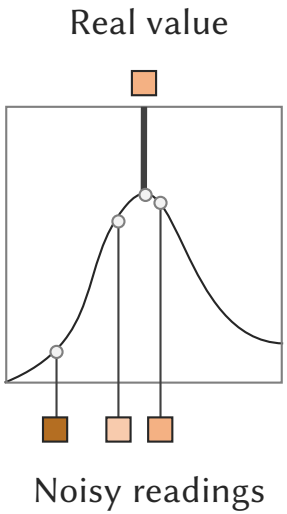
Image

Divide by number

Why should this be better?

- Consider taking n images of the same scene: More stable reconstruct

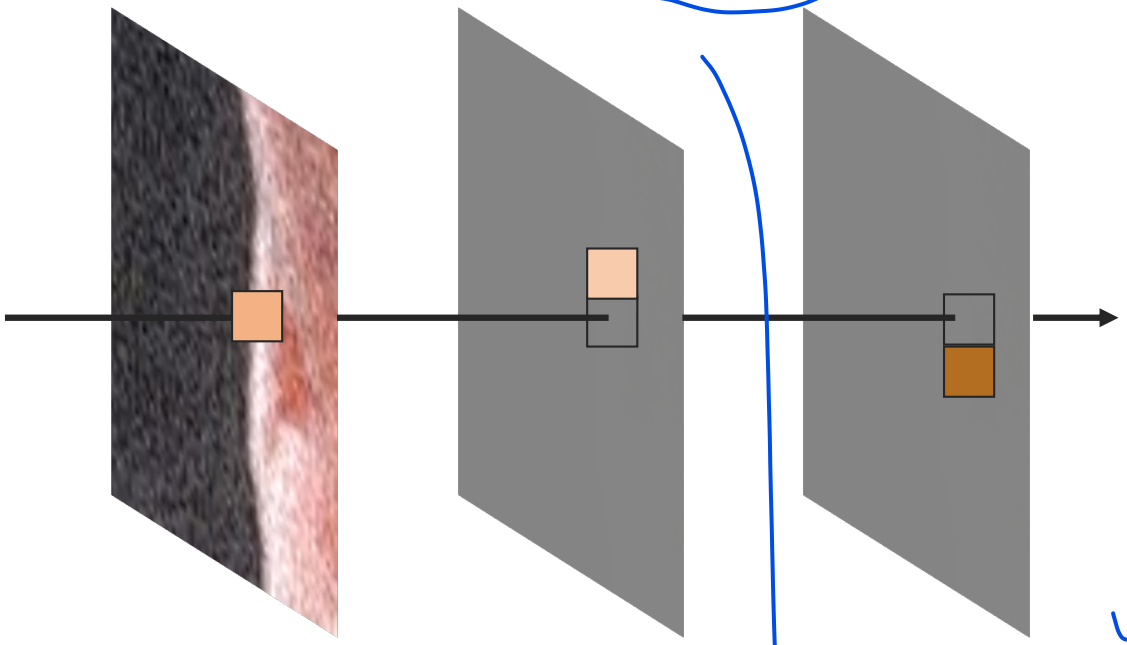


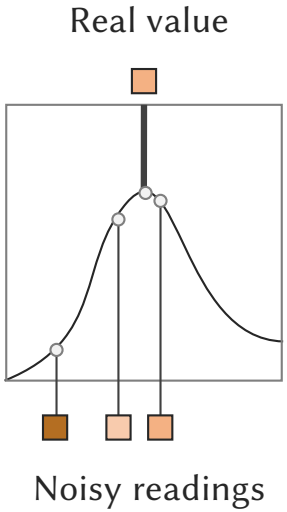


Why should this be better?

because

- As we only have one, we hope neighbours correlated

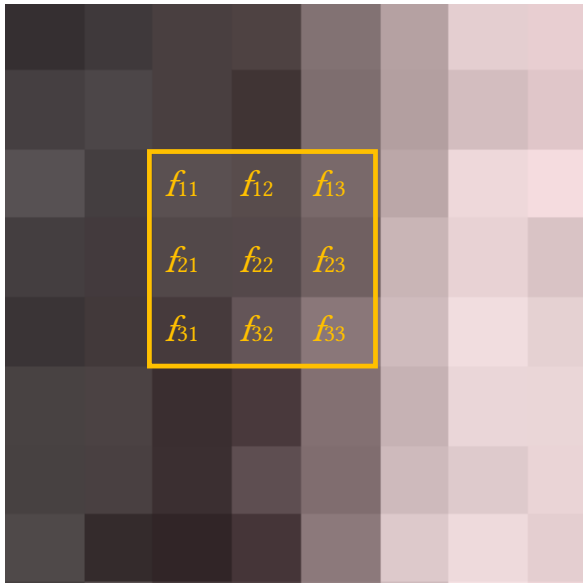




假设 neighbours 是上一张 slide 不同时间上的像素

Denoising

- So lets do this for every pixel, we get a new, de-noised image



Count plus-to-minus

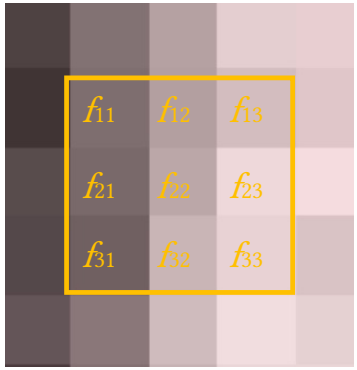
Output pixel value

$$f'_{x,y} = \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} f_{x+j,y+i}$$

Pixel, shifted

Convolution of signal f with kernel g

- Maybe not all neighbours are equally important
- Weight them with another matrix g



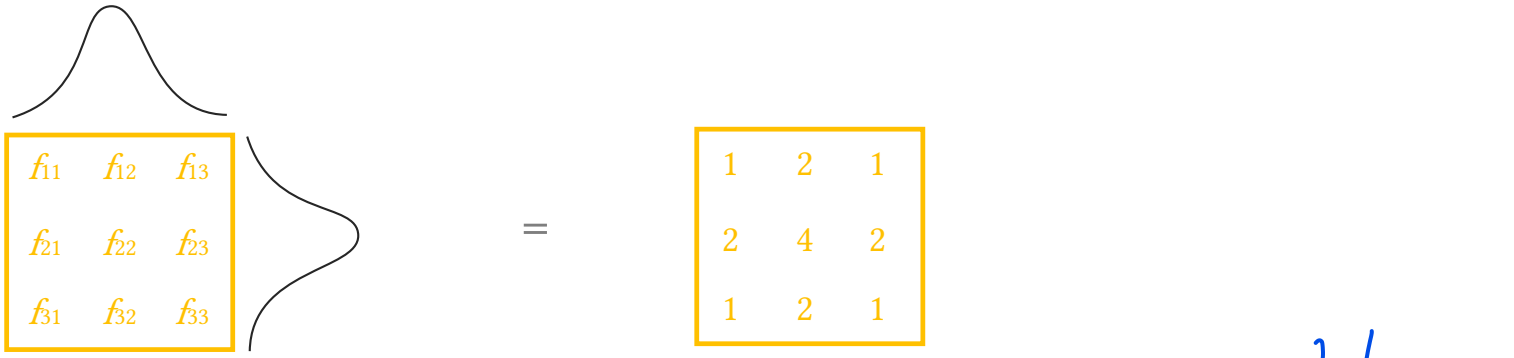
Convolution operator

$$f'_{x,y} = (f \odot g)_{x,y} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} g_{j,i} \times f_{x+j,y+i}$$

Kernel

Gaussian filter

- Choose weights from a **Gaussian**
- In my opinion no deeper meaning, just that center has more weight

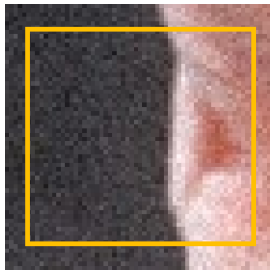


Convolution of signal f with kernel g

- Kernel can have any size n by m

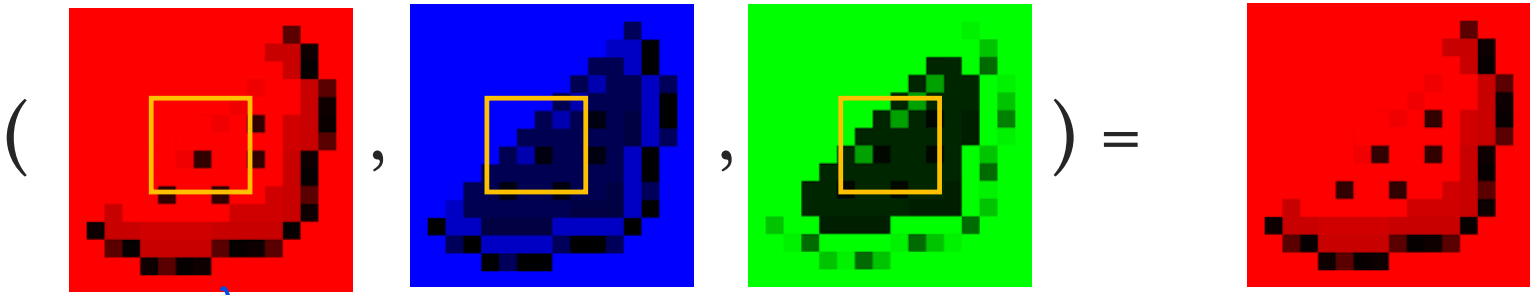
Assuring sums-to-one

$$f'_{x,y} = \Big(\sum_{i=\lfloor -n/2 \rfloor}^{\lceil n/2 \rceil} \sum_{j=\lfloor -m/2 \rfloor}^{\lceil m/2 \rceil} g_{j,i} \Big)^{-1} \sum_{i=\lfloor -n/2 \rfloor}^{\lceil n/2 \rceil} \sum_{j=\lfloor -m/2 \rfloor}^{\lceil m/2 \rceil} g_{j,i} \times f_{x+j,y+i}$$



3D convolution

- So far only for scalar (mono) images
- Can just run on channels independently
- Can also do it jointly (imperfect name: 3D convolution)



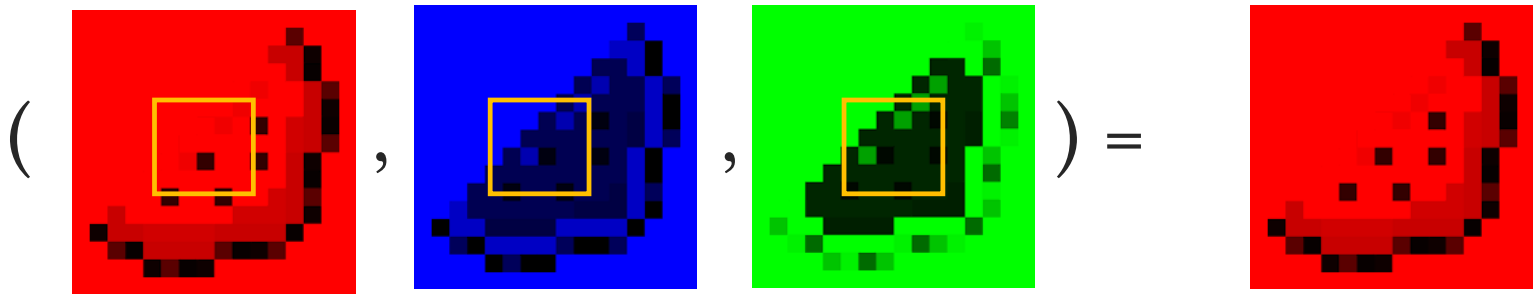
3D convolution

Magic sum-to-1

Loop over chans

$$f'_{x,y} = \mathrm{Z}_{i,j,c} \sum_i \sum_j \sum_c g_{j,i,c} \times f_{x+j,y+i,c}$$

Indexing chans



Works in any dimension

- 1-dimensional (audio)

$$f'_x = \mathrm{Z} \sum_i g_{i,c} \times f_{x+i,c}$$

Works in any dimension

- *n*-dimensional

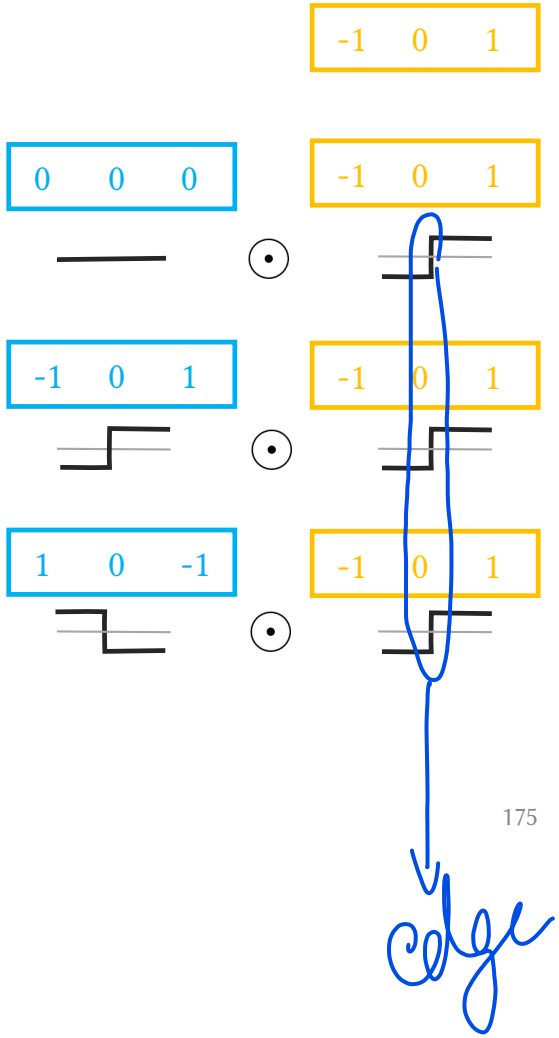
$$f'_{\mathbf{x}} = \mathrm{Z} \sum_{i_0} \cdots \sum_{i_n} g_{i_0,\ldots,i_n,c} \times g_{x_1+i_0,\ldots,x_n+i_n,c}$$

Other filters

- Denoising
- Sharpening
- Detecting edges
- Detecting points
- Detecting whatever

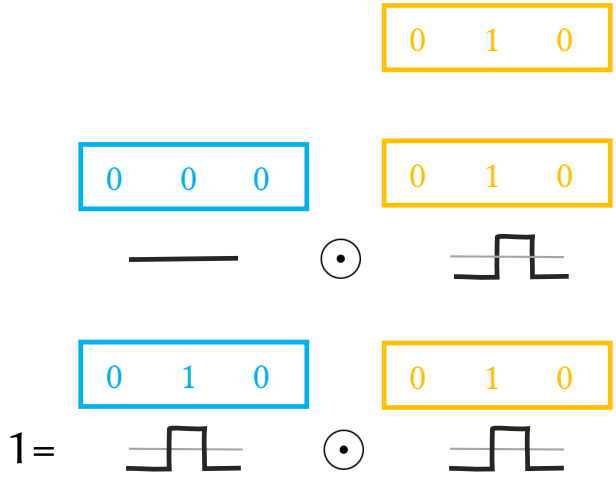
Edge detection

- Lets think 1D for now
- We look for kernel values so that
  - On a flat region there is no response 0=
  - On a rising edge there is positive response 2=
  - On falling one negative response -2=



Point detection

- We look for kernel values so that
  - On a flat region there is no response
  - On dot it is



Grandmother cells

- J Lettvin (Read up on this man …)
- We look for a value
  - That is high if it's the grandmother 1

  - That is low if it is not 0

- This is not how it works, not complete, but also not fully wrong

General image filters

- Image sharpening
  - Mix of edge detection and blur
- Morphological filters
  - Booleans, mins, maxs
- Non-linear filters
  - Filters, ranking
- Bi-lateral filters
  - Do not blur across edges

Not relevant anymore

Not relevant anymore

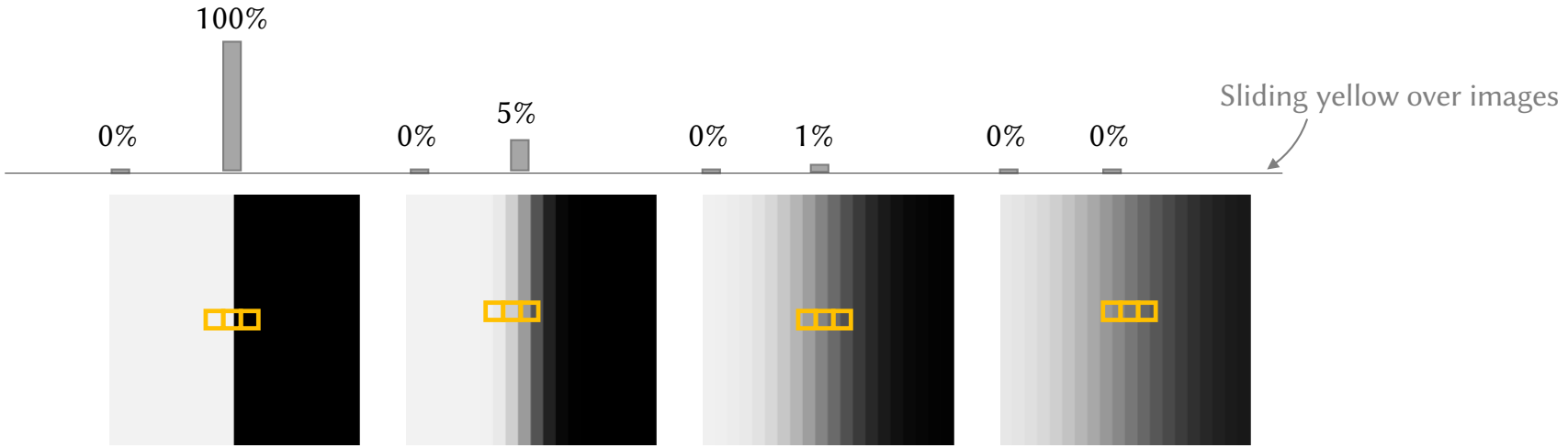
Maybe not

To some level, maybe

Multi-resolution

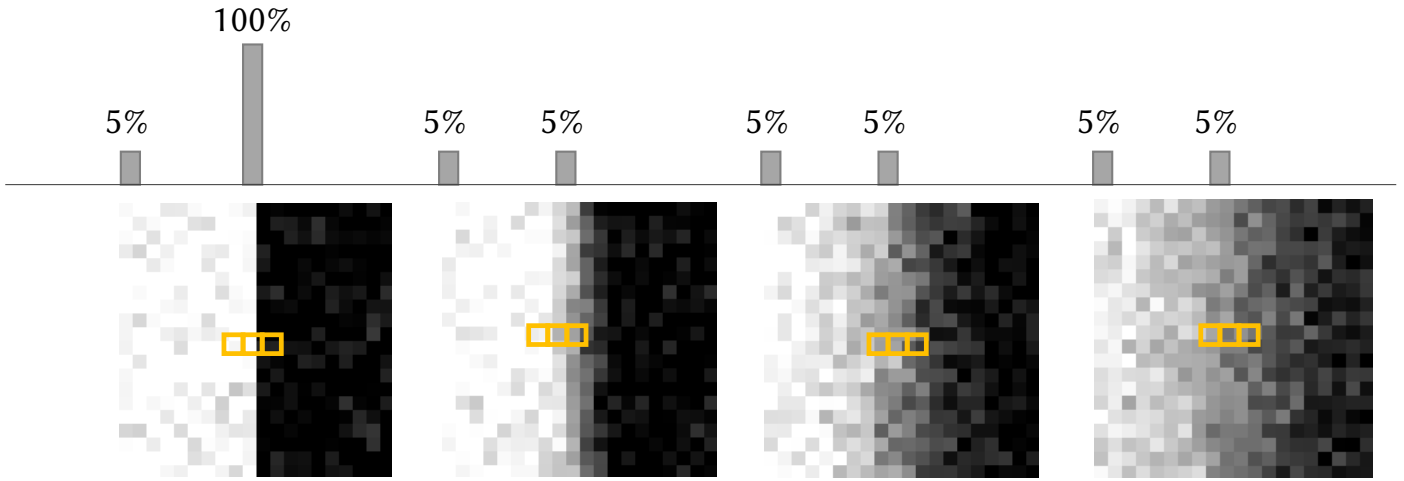
Motivation: Edges on all scales

- Lets try to find edges in those images:



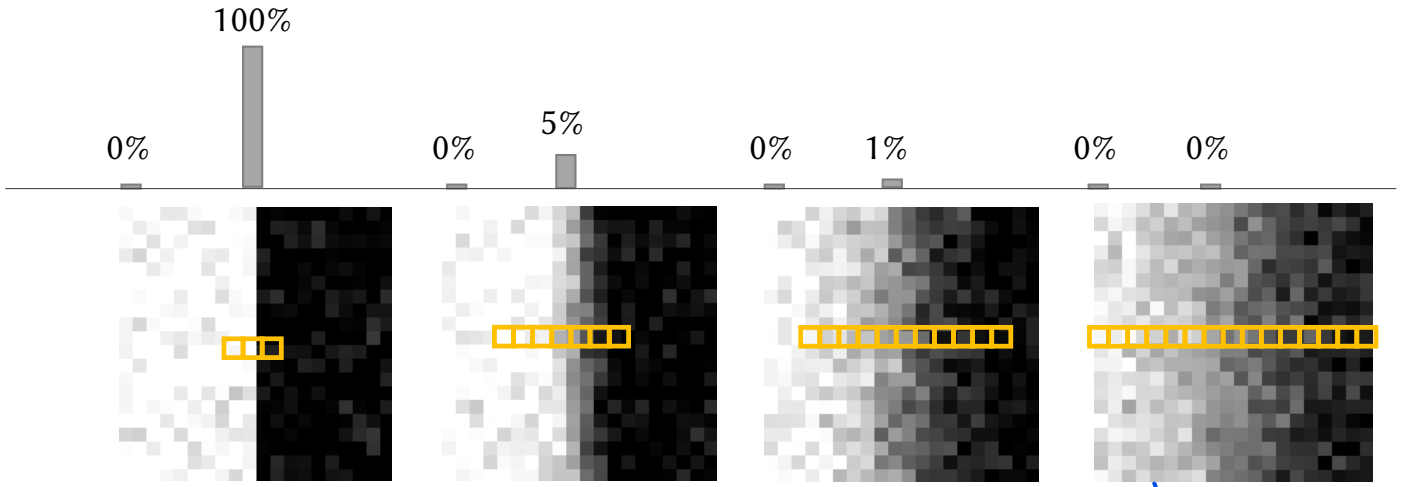
Motivation: One kernel not enough

- If we add some noise: no useful signal anymore



Motivation: Larger kernels?

- Solution: Larger filter? All good? No. Slow! And large kernels = complex.



Motivation: Smaller images!

- Key insight: instead of making **kernel larger**, we make **image smaller**

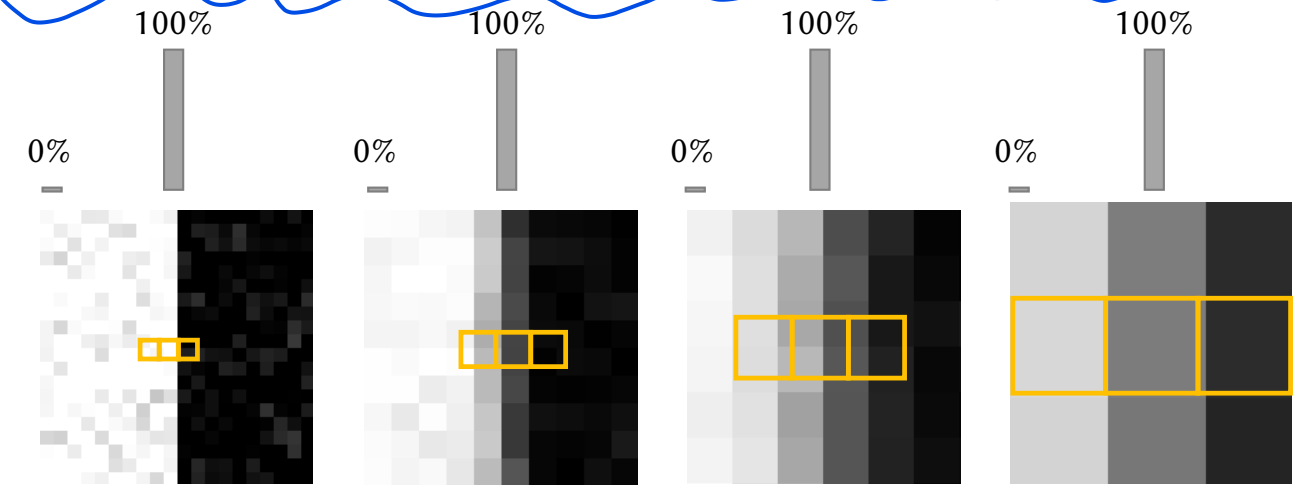


Image pyramids

- But what resolution is right? Simply do all!

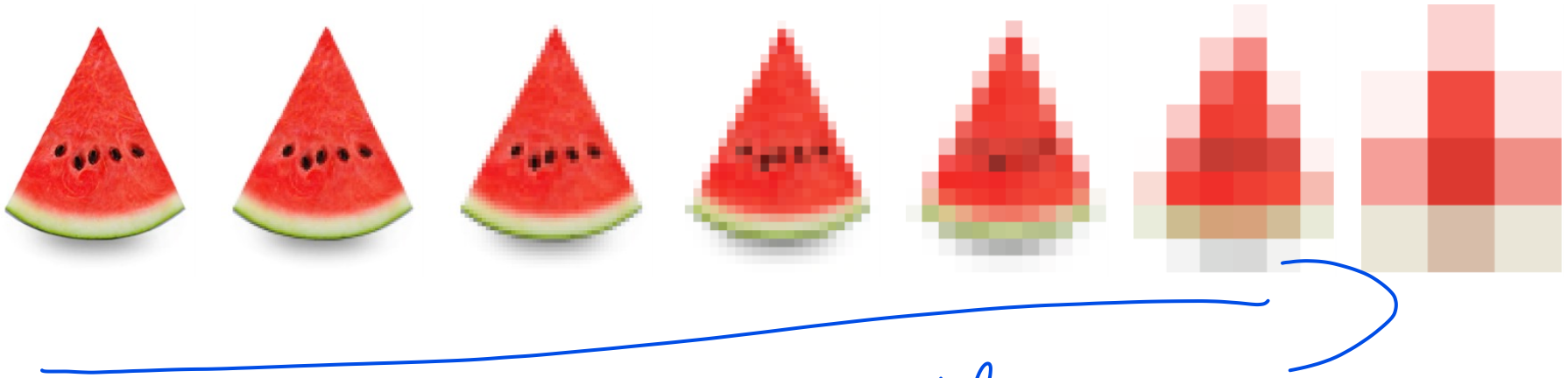
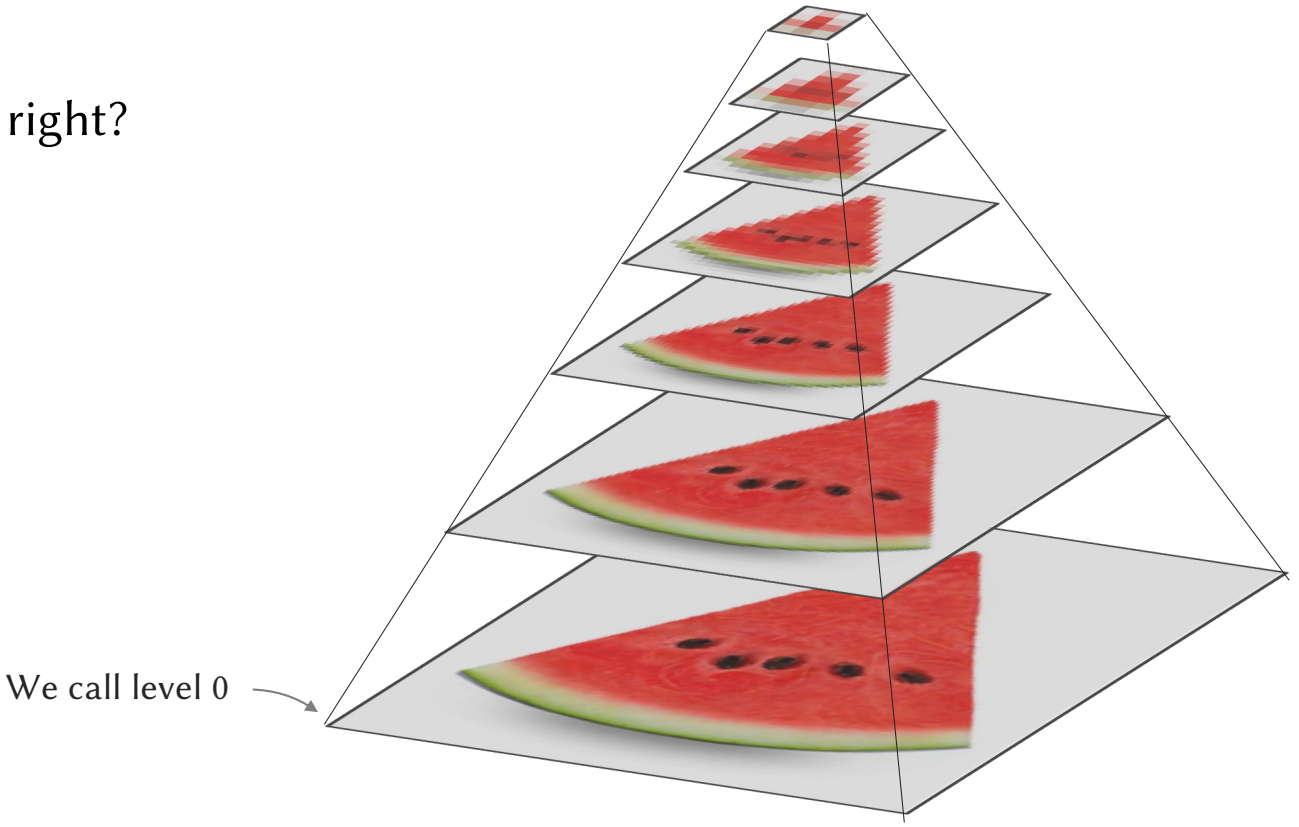


Image pyramids

- But what resolution is right?
- Simply do all!



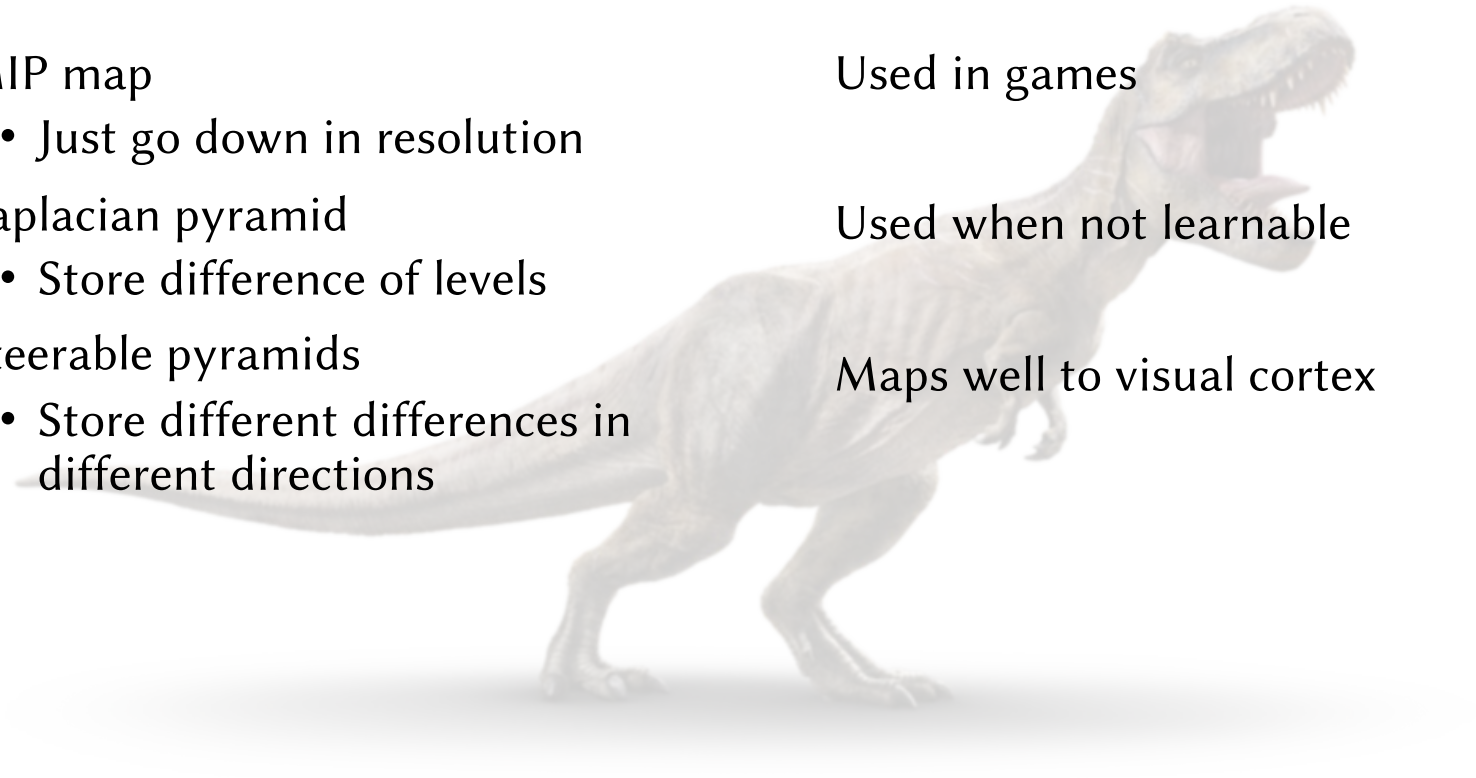
General image pyramids

- MIP map
  - Just go down in resolution
- Laplacian pyramid
  - Store difference of levels
- Steerable pyramids
  - Store different differences in different directions

Used in games

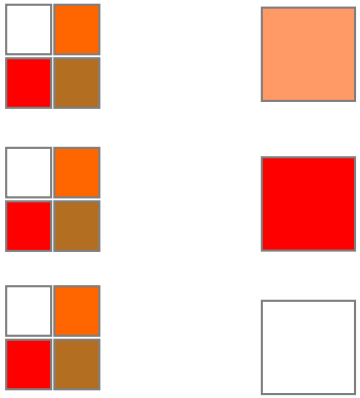
Used when not learnable

Maps well to visual cortex



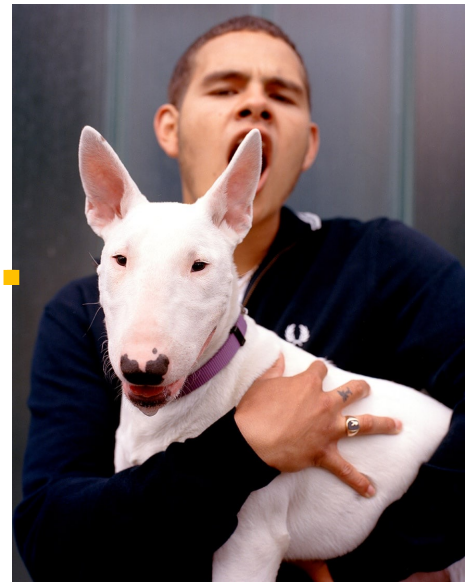
Pooling

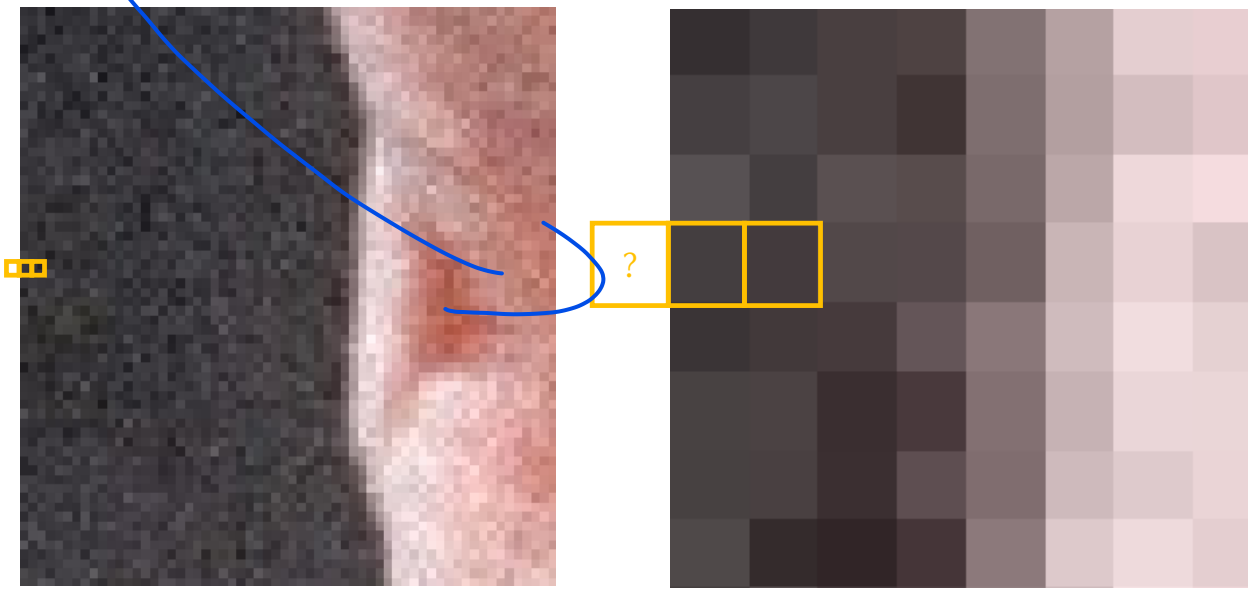
- How do we reduce resolution, actually?
- Consider 2x2 pixels being reduced to 1
- Options
  - `average`
  - `min`
  - `max`



Border handling

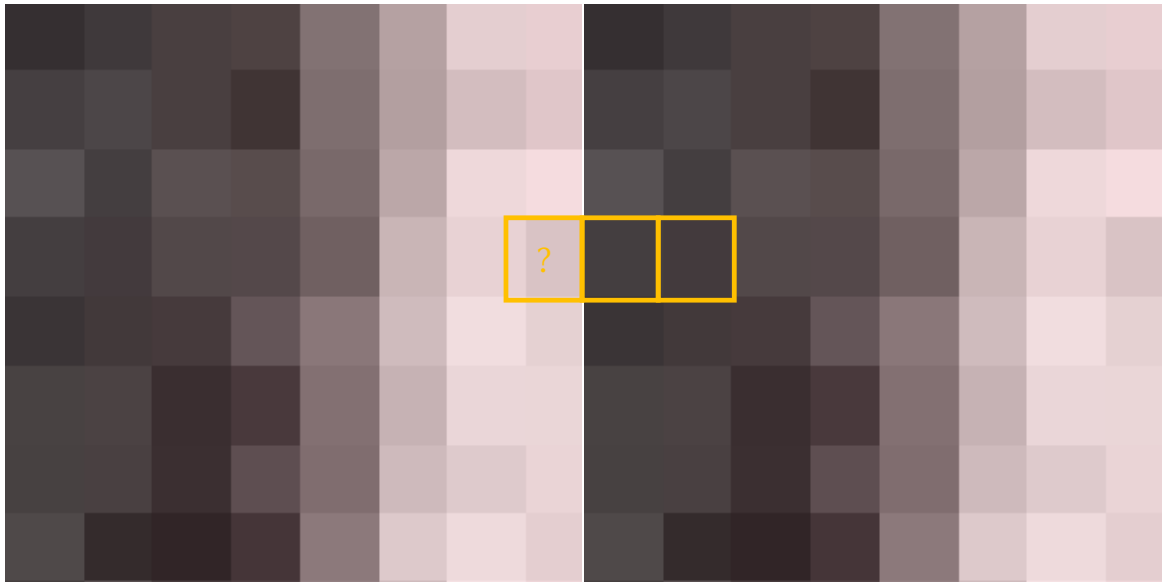
- How to deal with values the kernel needs, but not part of domain?





Border handling

- Mirror



Border handling

- Repeat

Border handling

- Constant

Conclusion

- Representation and processing go hand-in-hand
- Many similarities on many media
- Convolution is the key to processing
- Can be applied on many things
- Is there a grandmother cell?

Deep Learning for Visual Computing (COMP0169)

Convolutional Neural Networks

Niloy Mitra **Tobias Ritschel**

UCL