Image Analysis and Computer Vision

Introduction to Computer Vision

Lecture Notes

human perception

Vision is important

- half our brain is devoted to it
- developed multiple times during evolution
- it is non-contact
- it can be implemented with high resolution
- works with ambient E-M waves
- yields colour, texture, depth, motion, shape

central take home message

- For people vision is the most important sense, for good reason
- Effective vision needs more than sheer filtering and measuring
- It is feasible now to let most things see and interprete their environment

The perception of intensity, colour, length, lines being straight, parallelism, curvatures, motion, intensity
The brain factors out illumination

Kanisza illusion

- Fill-in : averaging of perceived contrast at edges over regions possibly obtained via extrapolation of the edges… in any case such illusion seems to help peopleto detect patterns in the world.
  The role of context
- human vision is much more than a bottom-up process of subsequent signal processing steps.

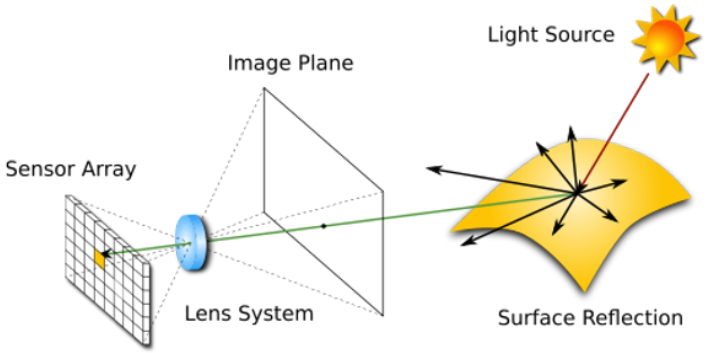
applications

The explosion of photography
The development of computer vision apps

- Most early applications where found in production environments, as these allow for **controlled conditions** and have **little uncertainty**
- currently CV is **conquering other less controllable areas** by storm
- image enhancement: mobile -> DSLR
  synthetic face generation
  autonomous vehicles(car detection, putting vision modalities together)
  image retrieval, captioning,...
  visual surveillance
  Augm. Reality, eg sports
  motion capture for movies/games
  computer-assisted surgery
  mobile mapping

light

Kickoff: the light, surface, lens & cam

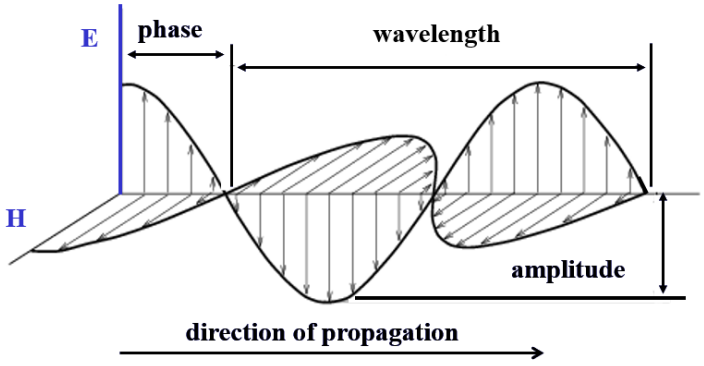


Levels of optical analysis

1. Geometrical optics
2. Physical optics, or (← **wave character**)
3. Quantum-mechanical optics

Light as electromagnetic waves

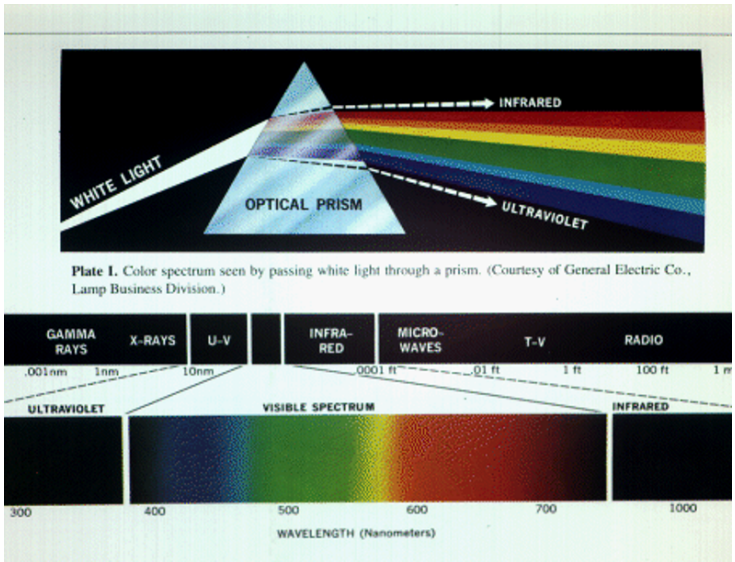
Self-sustaining exchange of electric and magnetic fields



wavelength, direction, amplitude E, phase, direction of polarisation

The spectrum

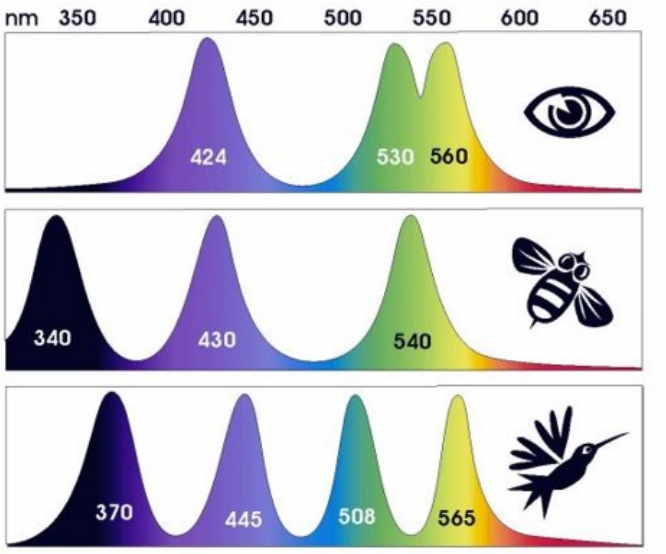
Normal ambient light is a mixture of wavelengths, polarisation directions, and phases



The visible range of wavelengths

Wavelength (in *nm*)	Colour
380 - 450	violet
450 - 490	blue
490 - 560	green
560 - 590	yellow
590 - 630	orange
630 - 760	red

- **NOTE 1**: From the observed colour you must not conclude that the light only contains wavelengths as given on the left
- **NOTE 2**: Cameras may have different spectral sensitivities (i.e. also different from human vision)



- **NOTE 3**: animals may have different spectral sensitivities (i.e. different from human vision), and may also have a different number of cone types (see lecture on colour), like 4 in most birds.

Also cams for non-visible `light', e.g. infrared

Interactions with matter

four types

phenomenon	example
absorption	blue water
scattering	blue sky, red sunset
reflection	coloured ink
refraction	dispersion by a prism

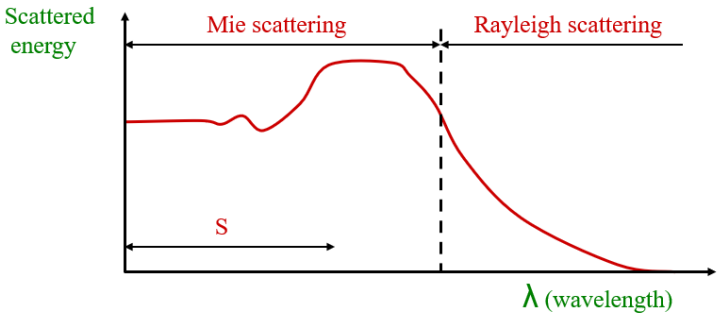
+ diffraction

Scattering

3 types depending on relative sizes of particles and wavelengths:

1. small particles: **Rayleigh** (strongly wavelength dependent)
2. comparable sizes: **Mie** (weakly wavelength dependent)
3. Large particles: **non-selective** (wavelength independent)

Wavelength dependence



- Less haze in the infrared (long wavelengths -> little scatter)
- Looking through clouds by radar (even longer wavelengths)
- NOTE: without scatter we would wander mainly in the dark

Atmospheric showcase

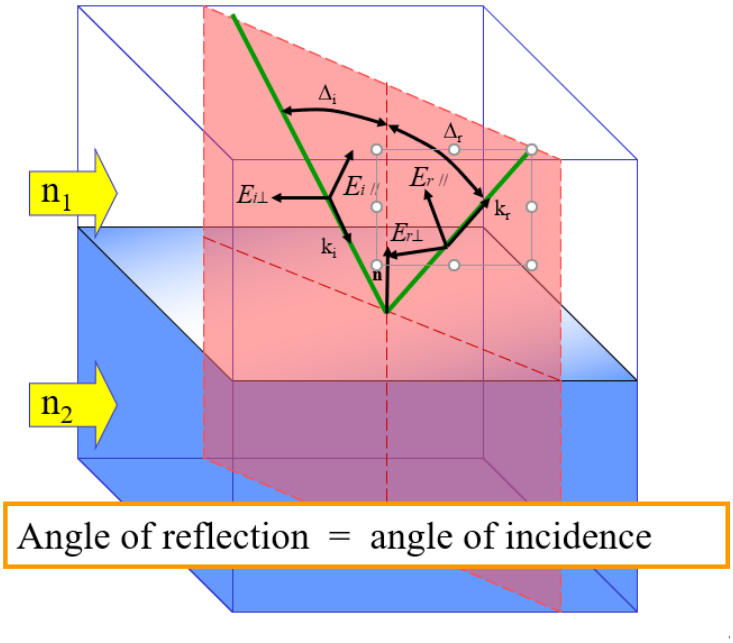
Rayleigh: Tyndall effect (blue sky) Red, setting sun

Non-selective: Grey clouds

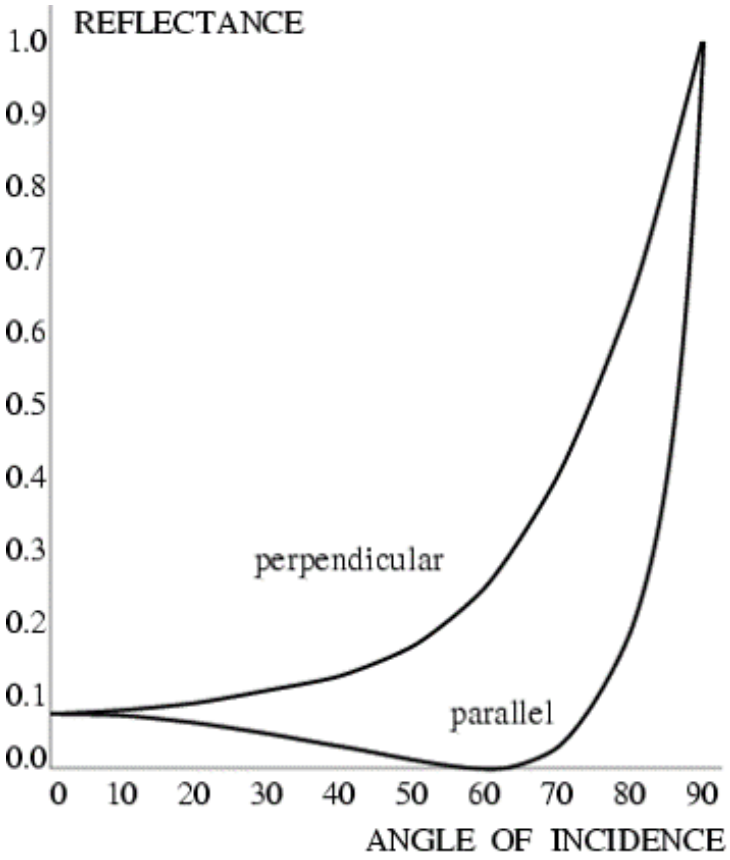
Mie: Coloured cloud from volcanic eruption

Reflection

Mirror reflection



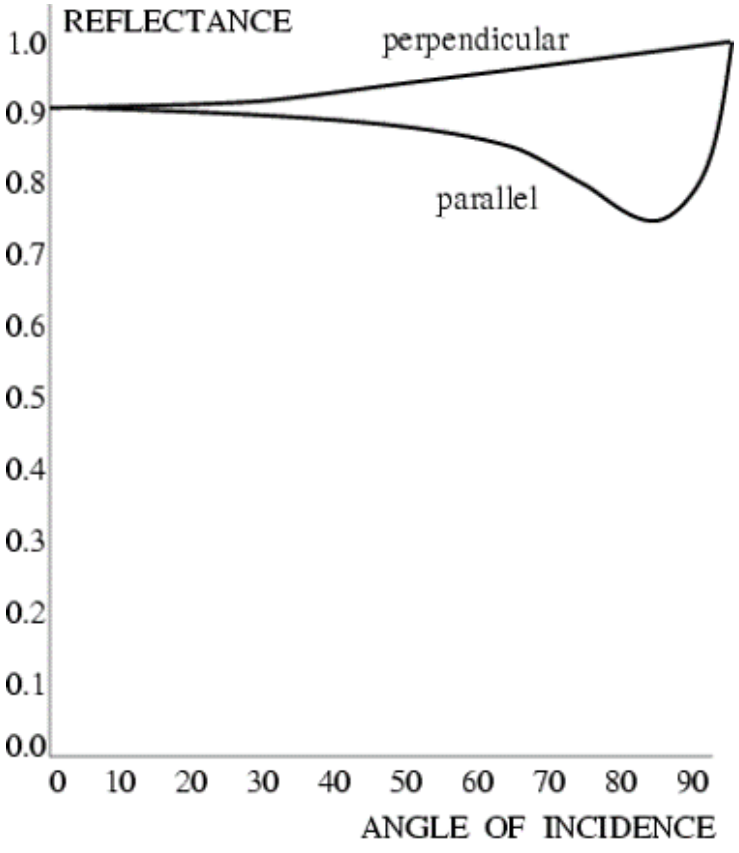
Mirror reflection : dielectric 电介质



Polarizer at **Brewster angle**

Full reflection at grazing angles

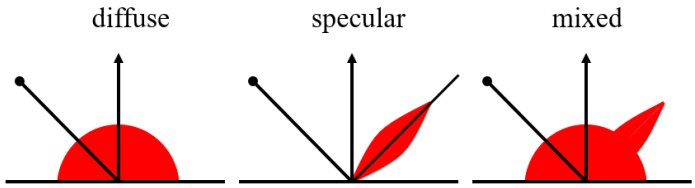
Mirror reflection : conductor 導体



- strong reflectors (under all angles)
- more or less preserve polarization

Roughness of surfaces leads to `diffuse' reflection
… and to mixed reflection for most real surfaces

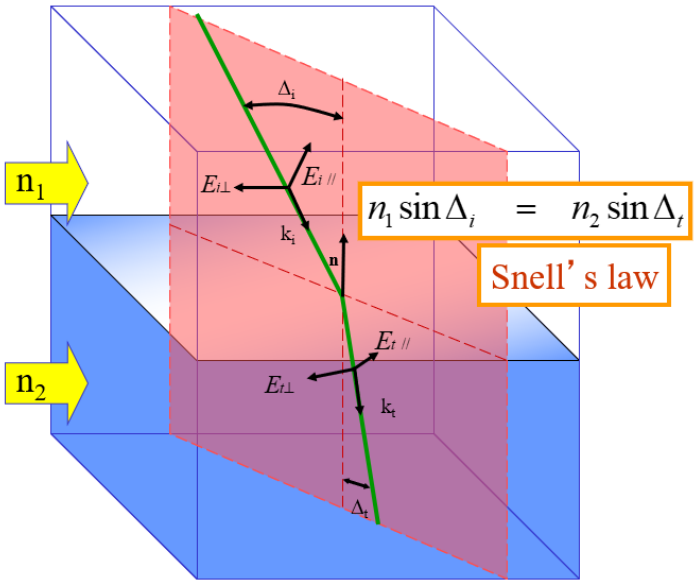
- three types of reflection :



- Note : Lambertian example of diffuse reflection.
- Under Lambertian reflection the surface looks equally bright when viewed from any direction
- Lambertian reflection: 理想散射

Spectral reflectance e.g. vegetation

Refraction 折射

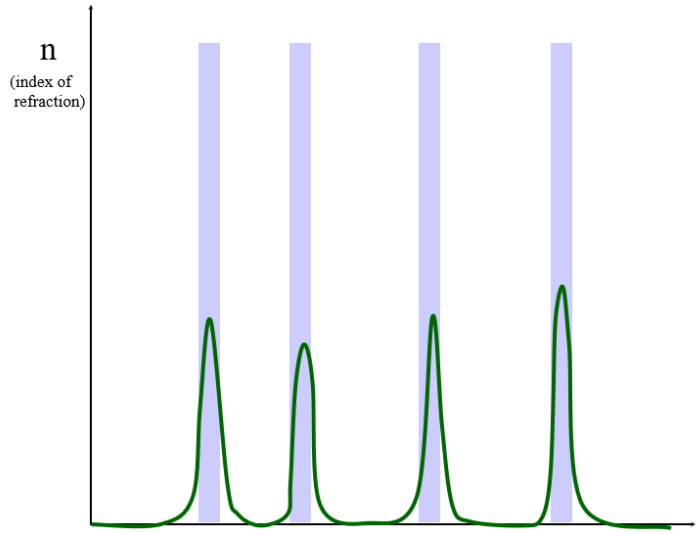


Dispersion 色散

- Refraction is more complicated than mirror reflection: the path orientation of light rays is changed depending on material AND wavelength !!!

Absorption

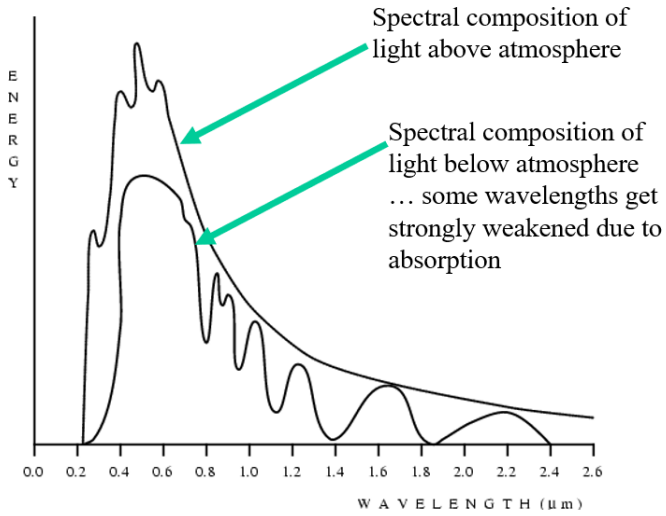
Dissipation of wavelengths specific for the medium



- Based on resonance frequencies of molecules -> peaks (where lights are absorbed)
- Holes in sky light spectrum observed by Fraunhofer

The solar spectrum

- Peaks around 500nm, hence human sensitivity for that part of the spectrum



Acquisition of Images

Lecture Notes

Illumination

Well-designed illumination often is key in visual inspection

Illumination techniques

Simplify the image processing by controlling the environment

1. back-lighting

- lamps placed behind a transmitting diffuser plate, light source behind the object
- generates high-contrast silhouette images, easy to handle with **binary vision**
- often used in inspection

2. directional-lighting

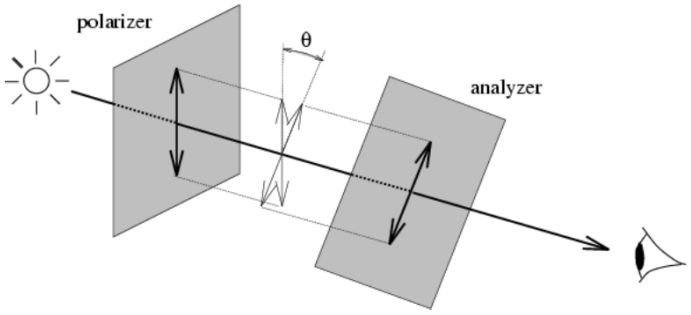
- generate sharp shadows
- generation of specular reflection (e.g. crack detection)
- shadows and shading yield information about shape

3. diffuse-lighting

- illuminates uniformly from all directions prevents sharp shadows and large intensity
- variations over glossy surfaces: all directions contribute extra diffuse reflection, but contributions to the specular peak arise from directions close to the mirror one only

4. polarized-lighting

- polarizer/analyzer configurations



- **Law of Malus**:
  $I(\theta) = I(0)cos^2\theta$
- to improve contrast between Lambertian and specular reflections
  - **specular** reflection keeps polarisation; **diffuse** reflection depolarises
  - suppression of specular reflection: polarizer/analyzer **crossed**, prevents the large dynamic range caused by glare
- to improve contrasts between dielectrics and metals
  - reflection on dielectric: Polarizer at **Brewster angle**
  - conductors are strong reflectors, more or less preserve polarization
  - distinction between specular reflection from dielectrics and metals;
    works under the Brewster angle for the dielectric, so that dielectric has no parallel component, metal does
    suppression of specular reflection from dielectrics: **polarizer/analyzer aligned**
  - **distinguished metals and dielectrics**

5. coloured-lighting

- **highlight** regions of a similar colour
- with **band-pass filter**: only light from projected pattern (e.g. monochromatic light from a laser)
- differentiation between specular and diffuse reflection
- comparing colours $\Rightarrow$ same spectral composition of sources!
- spectral sensitivity function of the sensors!

6. structured-lighting

- spatially modulated light pattern
- e.g. : 3D shape : objects distort the projected pattern (more on this later)

7. stroboscopic lighting

- temporally modulated light pattern
- high intensity light flash to eliminate motion blur

Use of specular reflection – e.g. crack detection

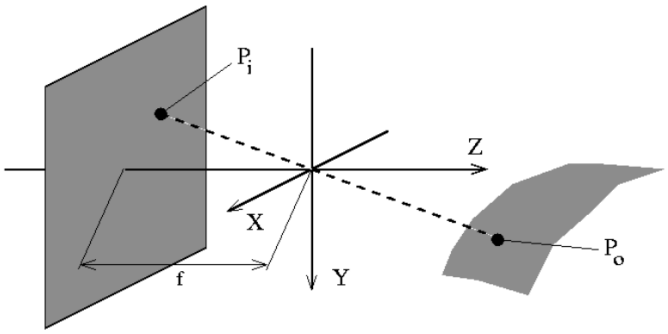
- 'Dark' and 'bright' field
  In the 'dark' field, the camera is placed out of the area of specular reflection for the normal surface, and only abnormally oriented parts of the surface will lighten up (showing specular reflection) – flaws
  In the 'bright' field, the camera is placed so to capture the specular reflection for normally oriented parts of the surface. Parts with an abnormal orientation – flaws - will appear dark.

App: vegetable inspection (colored light + polarization)

Cameras

Optics for image formation

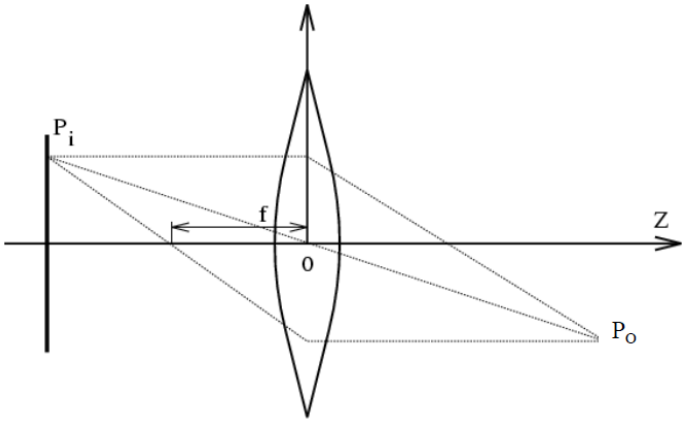
- the pinhole model:



$\frac{X_i}{X_o} = \frac{Y_i}{Y_o} = \frac{Z_i}{Z_o} = -m$
where m is linear magnification

The thin-lens equation

- lens to capture enough light:

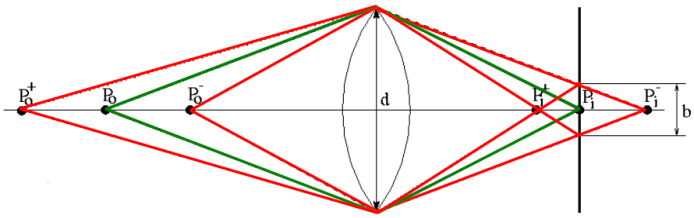


$\frac{1}{Z_o} - \frac{1}{Z_i} = \frac{1}{f}$

- assuming
  - spherical lens surfaces
  - incoming light ± parallel to axis
  - thickness << radii
  - same refractive index on both sides

The depth-of-field

Only reasonable sharpness in Z-interval



- $\Delta Z_o^- = Z_o - Z_o^- = \frac{Z_o(Z_o - f)}{Z_o}$
- decreases with d, increases with Z_o
- Similar expression for $Z_o^+ - Z_o$
  strike a balance between incoming light (d) and large depth-of-field (usable depth range)
- **with a smaller d, the brightness of image decrease, but we get a larger depth-of-field**
- Ex 1: microscopes -> small DoF
  Ex 2: special effects -> flood miniature scene with light

Deviations from the lens model

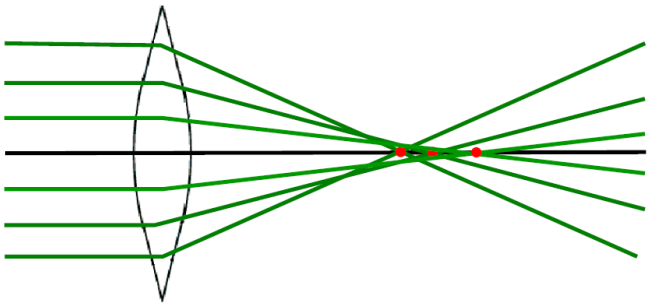
3 assumptions:

1. all rays from a point are focused onto 1 image point
2. all image points in a single plane
3. magnification is constant
   deviations from this ideal are **aberrations** 像差

Aberrations

2 types:

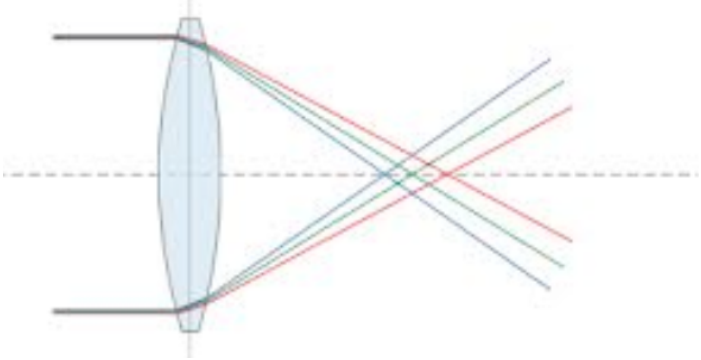
1. geometrical
   small for paraxial rays(近轴近似)
   - spherical aberration(球差)
     rays parallel to the axis do not converge
     outer portions of the lens yield smaller focal lenghts

     
   - astigmatism(散光)
   - **radial distortion**(径向变形)
     - magnification different for different angles of inclination

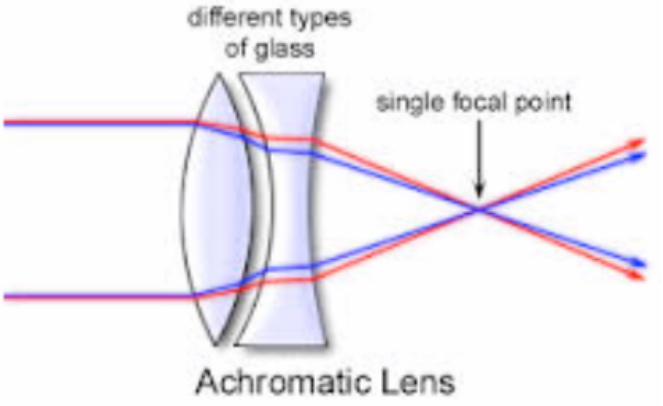
       

       barrel *none* *pincushion*
     - The result is pixels moving along lines through the center of the distortion - typically close to the image center - over a distance d, depending on the pixels' distance r to the center
       $d = (1 + \kappa_1 r^2 + \kappa_2 r^4 + ...)$
     - This aberration type can be corrected by software if the parameters $(\kappa_1, \kappa_2, ...)$ are known
       Some methods do this by looking how straight lines curve instead of being straight
   - coma(彗形像差)
2. chromatic(色差)
   refractive index function of wavelength (Snell's law !!)

- rays of different wavelengths focused in different planes

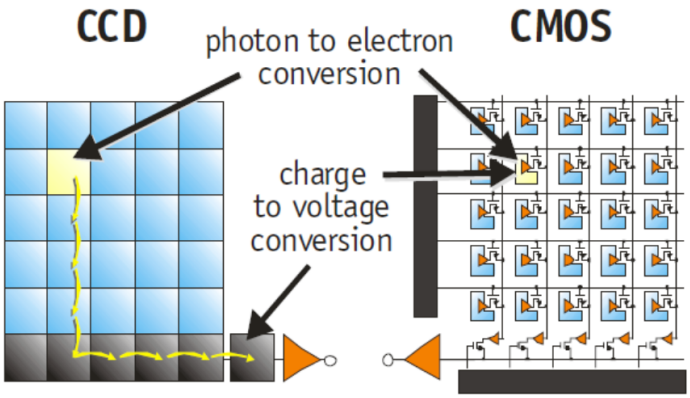


- cannot be removed completely but **achromatization** can be achieved at some well chosen wavelength pair, by combining lenses made of different glasses



sometimes **achromatization** is achieved for more than 2 wavelengths

Cameras



1. CCD = Charge-coupled device
2. CMOS = Complementary Metal Oxide Semiconductor

- Same sensor elements as CCD
- Each photo sensor has its own amplifier
  - More noise (reduced by subtracting ‘black’ image)
  - Lower sensitivity (lower fill rate)

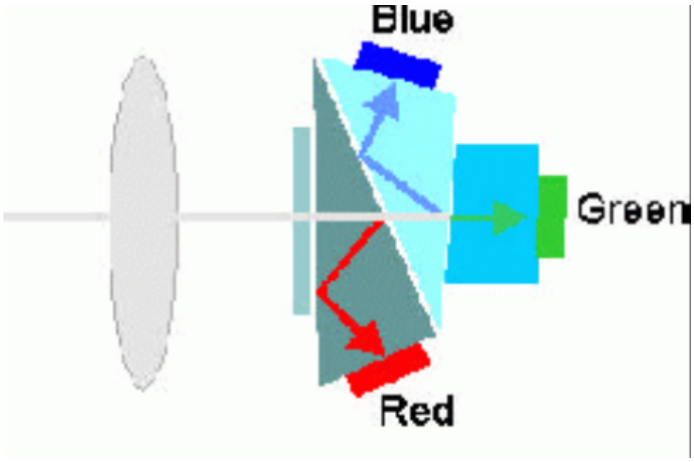
- Uses standard CMOS technology
  - Allows to put other components on chip
  - ‘Smart’ pixels

CCD vs. CMOS

- CCD: Niche applications, Specific technology, High production cost, High power consumption, Higher fill rate, Blooming, Sequential readout
- CMOS: Consumer cameras, Standard IC technology, Cheap, Low power, Less sensitive, Per pixel amplification, Random pixel access, Smart pixels, On chip integration with other components
- 2006 was year of sales cross-over
  In 2015 Sony said to stop CCD chip production

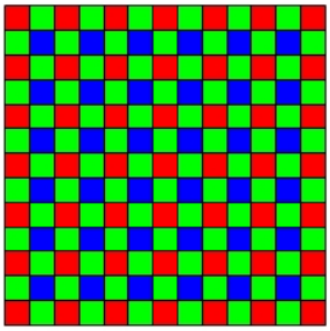
Colour cameras

1. Prism (with 3 sensors)
   Separate light in 3 beams using dichroic prism
   Requires 3 sensors & precise alignment
   Good color separation

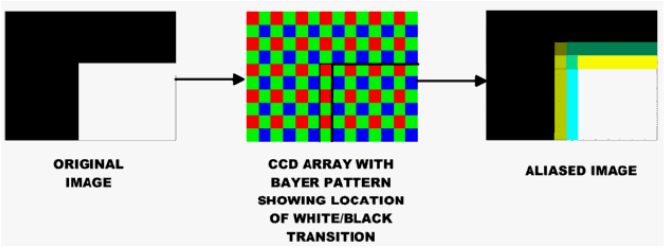


2. Filter mosaic

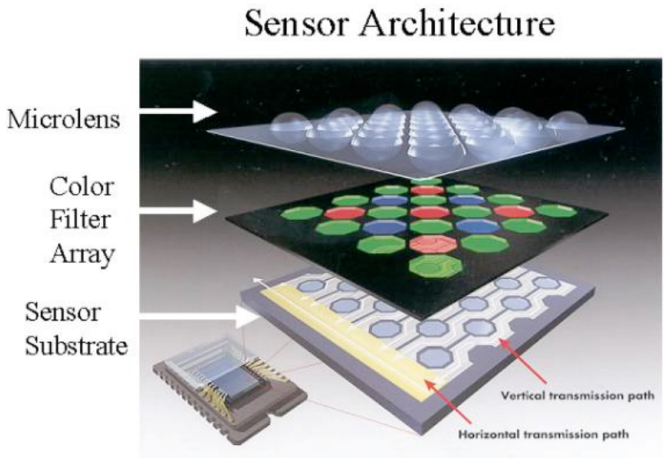
- Coat filter directly on sensor


Bayer filter

- Demosaicing (obtain full colour & full resolution image)

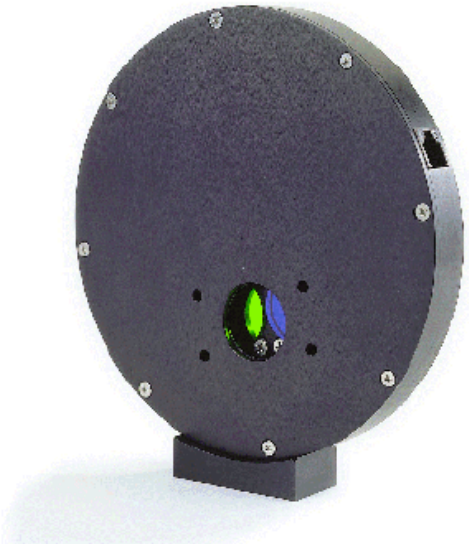


- Color filters lower the effective resolution, hence **microlenses** often added to gain more light on the small pixels



3. Filter wheel

- Rotate multiple filters in front of lens
  Allows more than 3 colour bands
- Only suitable for **static** scenes



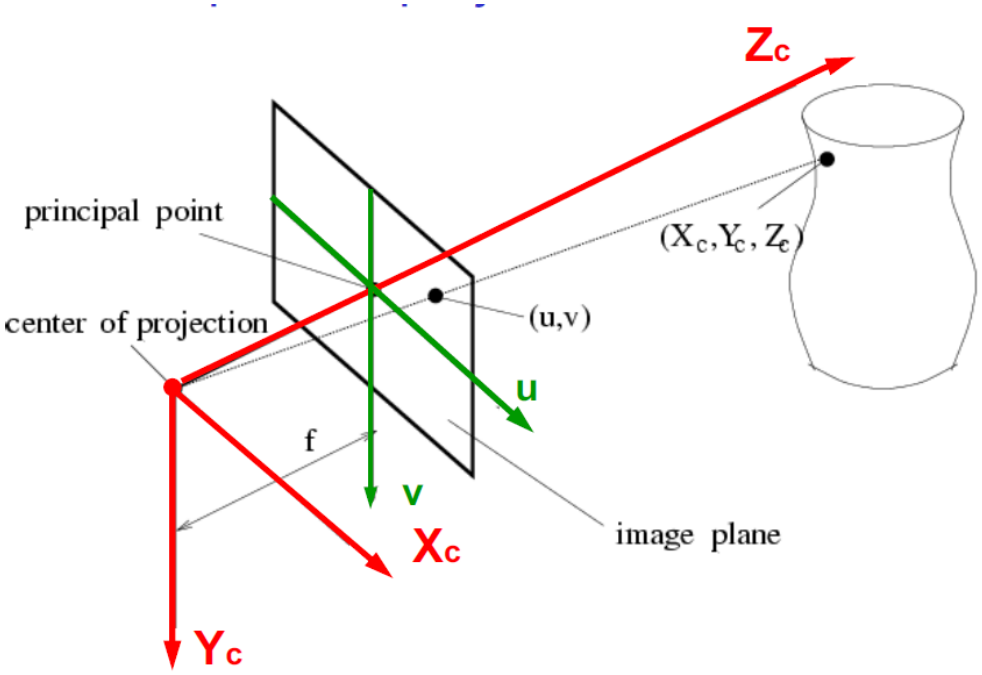
Prism vs. mosaic vs. wheel

approach	Prism	Mosaic	Wheel
# sensors	3	1	1
Resolution	High	Average	Good
Cost	High	Low	Average
Framerate	High	High	Low
Artefacts	Low	Aliasing	Motion
Bands	3	3	3 or more
	High-end cameras	Low-end cameras	Scientific applications

Models for camera projection

- the pinhole model revisited:
  center of the lens = center of projection
  notice the virtual image plane
  this is called **perspective** projection

Perspective projection



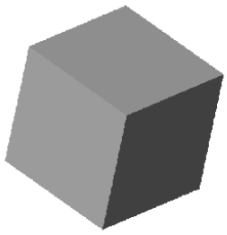
- origin lies at the center of projection
- the Z_c axis coincides with the optical axis
- X_c-axis $\|$ to image rows, Y_c-axis $\|$ to columns
- $u = f\frac{X}{Z}, v = f\frac{Y}{Z}$

Pseudo-orthographic projection

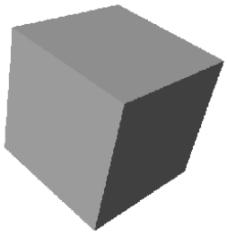
- If Z is constant $\Rightarrow x = kX$ and $y = kY$, where $k = f/Z$
  i.e. **orthographic projection + a scaling**

- Good approximation if $f/Z \pm constant$, i.e. if objects are **small compared to their distance** from the camera
- Pictoral comparison

Pseudo - orthographic



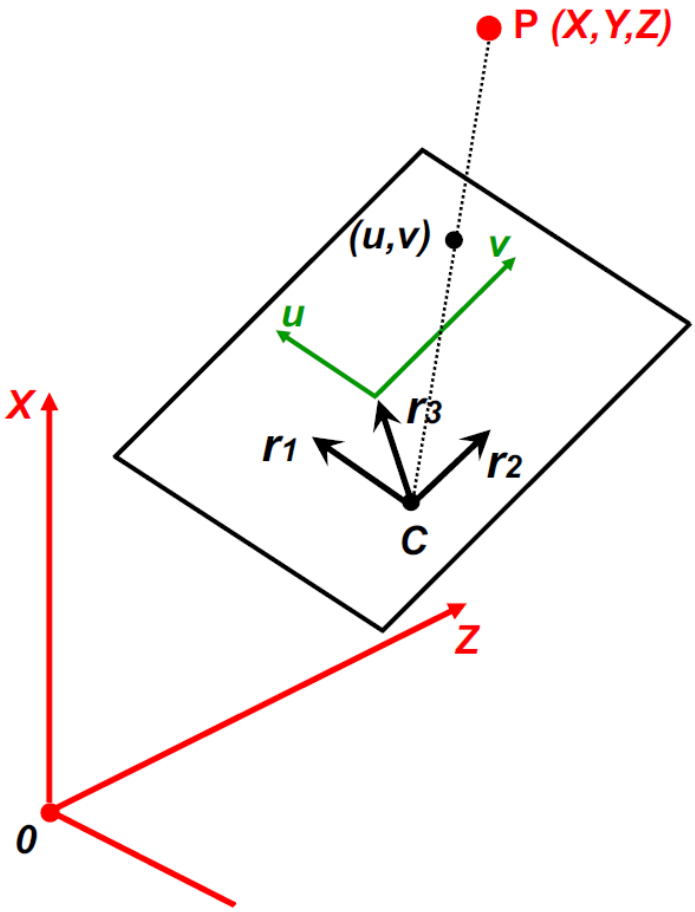
Perspective



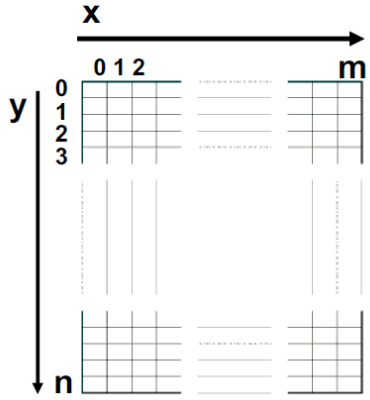
Projection matrices

- the perspective projection model is incomplete :
  what if :
  1. 3D coordinates are specified in a **world coordinate frame**
  2. Image coordinates are expressed as **row and column numbers**
- We will not consider additional refinements, such as radial distortions,...

Projection matrices



- $u = f\frac{\langle r_1, P-C\rangle}{\langle r_3, P-C\rangle} \quad v = f\frac{\langle r_2, P-C\rangle}{\langle r_3, P-C\rangle}$
- $u = f\frac{r_{11}(X-C_1)+r_{12}(Y-C_2)+r_{13}(Z-C_3)}{r_{31}(X-C_1)+r_{32}(Y-C_2)+r_{33}(Z-C_3)} \quad v = f\frac{r_{21}(X-C_1)+r_{22}(Y-C_2)+r_{23}(Z-C_3)}{r_{31}(X-C_1)+r_{32}(Y-C_2)+r_{33}(Z-C_3)}$
- Image coordinates are to be expressed as pixel **coordinates**



  - $\begin{cases} x = k_x u + sv + x_0 \\ y = k_y v + y_0 \end{cases}$
  - (x_0, y_0) the pixel coordinates of the principal point
  - k_x the number of pixels per unit length horizontally
  - k_y the number of pixels per unit length vertically
  - s indicates the **skew**; typically $s = 0$
- **NB1**: often only integer pixel coordinates matter
- **NB2**: k_y / k_x is called the **aspect ratio**

- **NB3**: k_x, k_y, s, x_0, y_0 are called **internal camera parameters**
- **NB4**: when they are known, the camera is **internally calibrated**
- **NB5**: vector C and matrix $R \in SO(3)$ are the **external camera parameters**
- **NB6**: when these are known, the camera is **externally calibrated**
- **NB7**: **fully calibrated** means internally and externally calibrated

Homogeneous coordinates

Often used to linearize non-linear relations

- 2D: $\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x/z \\ y/z \end{pmatrix}$
- 3D: $\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} \rightarrow \begin{pmatrix} X/W \\ Y/W \\ Z/W \end{pmatrix}$
- **Homogeneous coordinates are only defined up to a factor**

Projection matrices

- Exploiting homogeneous coordinates:
$$\tau \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fr_{11} & fr_{12} & fr_{13} \\ fr_{21} & fr_{22} & fr_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X - C_1 \\ Y - C_2 \\ Z - C_3 \end{pmatrix}$$
$$\tau \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$
- Thus, concatenating the results:
$$\tau \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} fr_{11} & fr_{12} & fr_{13} \\ fr_{21} & fr_{22} & fr_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X - C_1 \\ Y - C_2 \\ Z - C_3 \end{pmatrix}$$
- Or, equivalently:
$$\tau \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X - C_1 \\ Y - C_2 \\ Z - C_3 \end{pmatrix}$$
- Re-combining matrices in the concatenation yields the calibration matrix K:
$$K = \begin{pmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} fk_x & fs & x_0 \\ 0 & fk_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

- We define $p = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$; $P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$; $\tilde{P} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$
  yielding $\rho p = KR^t(P - C)$ for some non-zero $\rho \in \mathbb{R}$
  or, $\rho p = K(R^t| - R^tC)\tilde{P}$
  or, $\rho p = K(M|t)\tilde{P}$ with $rank(M) = 3$

From object radiance to pixel grey levels

- a **photometric** camera model

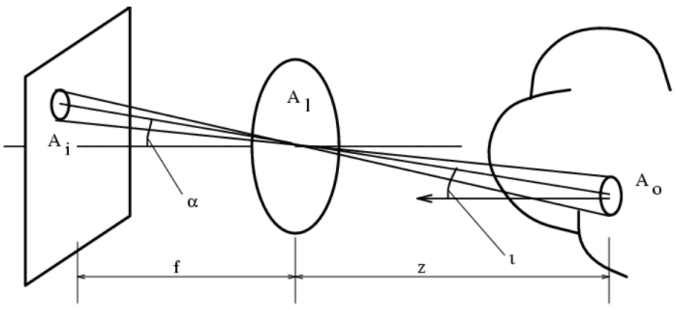
1. from object radiance to image irradiance
2. from image irradiance to pixel grey level

Image irradiance(辐照度) and object radiance

- we look at the irradiance that an object patch will cause in the image
- assumptions:
  radiance R assumed known
  object at large distance compared to the focal length
- Is image irradiance directly related to the radiance of the image patch?

The viewing conditions

- the cos^4 law
  $I = R\frac{A_l}{f^2}cos^4\alpha$



  Especially strong effects for wide-angle and fisheye lenses

From irradiance to gray levels

- $f = gI^\gamma + d$
  where g is gain, set w. size diaphragm,

γ is "gamma", close to 1 nowadays
d is Dark reference, signal w. cam cap on

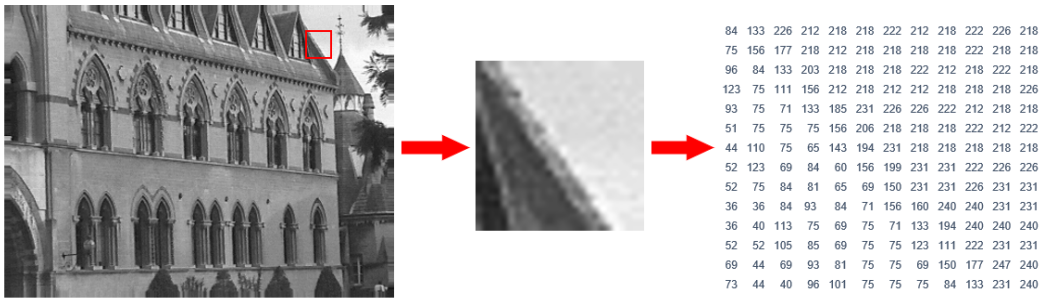
Sampling, Quantization and Image Enhancement

Lecture Notes

Sampling & quantization

Discretization / Digitization

- Necessary computer to process an image
- Includes two parts
  1. Sampling – spatial discretization, creates “pixels”
  2. Quantization – intensity discretization, creates “grey levels”



Creating finite number of points in space in a grid, i.e. pixels, and intensity value in each pixel is represented with finite number of bits in the computer.
The original scene is continuous in space and intensity value.

Example of quantization(#pixels,#levels)
Image distortion through sampling
Remarks

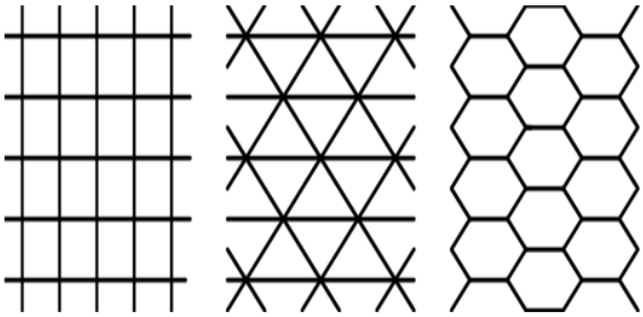
1. Binary images – 1-bit quantization – are useful in industrial applications. They usually have control over imaging conditions
   e.g. background color, lighting conditions,...
2. Non-uniform sampling and/or quantization is sometimes used for specialized applications

- Fine sampling to capture details
- Fine quantization for homogeneous regions

3. Different sampling strategies than square grids exist

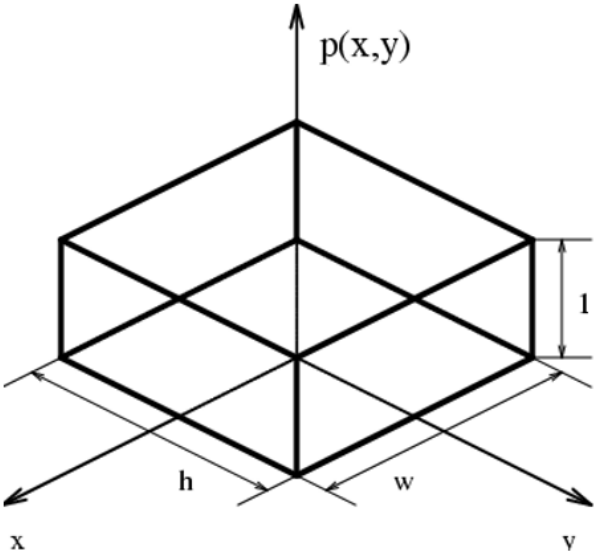
Different sampling schemes

- You need regular, image covering tessellation
- There are 11 polygons to achieve this. If you want to use the same polygon across the image then only 3, shown on the right.
- Rectangular (square) is the most popular
- Hexagonal has advantages (more isotropic, no connectivity ambiguities). Similar structure is seen in the retina of various species.



A model for sampling

- There are two essential steps
  1. Integrate brightness over a cell window
     Leads to blurring type degradation



     $o(x', y') = \int \int i(x, y) p(x' - x, y' - y) dx dy$ (modified)
     This is a **convolution**: $i(x, y) * p(-x, -y)$
  2. Read out values only at the pixel centers
     Leads to aliasing and leakage, frequency domain issues

Convolution

- While the previous convolution was in continuous domain, we'll look at discrete convolution to get an intuition.
  Image: $x(i,j)$
  Convolutional kernel: $w(i,j) \rightarrow w * x$
  $a_{ij} = \sum_p \sum_q x_{(i-p)(j-q)} w_{(p)(q)}$
- Consider the continuous case as the limit where pixels are very small as well as the convolutional kernel is formed to correspond to that with many very small elements.
  The kernel for this case is a rectangular box.

Properties of convolution

- Commutative
  $f * g = g * f$
- Associative
  $k = h * f$
  $= (h_1 * h_2) * f$
  $= h_1 * (h_2 * f)$

The Fourier Transform

- An important tool we should remind ourselves is the Fourier Transform (FT).
- This is crucial to understand the effects of STEPI as well as STEPII taken in sampling.
- Particularly, it is difficult to understand what type of information we lose when we convolve an image with a kernel with a box shape.
- Using FT, this becomes much easier!

Characterization of functions in the frequency domain

- Represent any signal as a linear combination of orthonormal basis functions
  $e^{i2\pi(ux+vy)} = cos2\pi(ux+vy) + isin2\pi(ux+vy)$
- Waves with wavelength orthogonal to the stripes of
  $\lambda = \frac{1}{\sqrt{u^2+v^2}}$

The Fourier Transform: definition

- Linear decomposition of functions in the new basis
  Scaling factor for basis function (u,v)
- The Fourier Transform
  $\mathcal{F}[f(x,y)] = F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-i2\pi(ux+vy)} dxdy$
- Reconstruction of the original function in the spatial domain:
  weighted sum of the basis functions

- The Inverse Fourier Transform
  $\mathcal{F}^{-1}[F(u,v)] = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{i2\pi(ux+vy)}dudv$ (modified)

Fourier Coefficients

- Complex function
  $F(u,v) = \underbrace{F_R(u,v)}_{\text{real part}} + \underbrace{iF_I(u,v)}_{\text{imaginary part}}$
- Magnitude
  $|F(u,v)| = \sqrt{F_R(u,v)^2 + F_I(u,v)^2}$
- Phase-angle
  $\phi(u,v(= arg(F_R(u,v) + iF_I(u,v)) = arctan\frac{F_I(u,v)}{F_R(u,v)}$

Decomposition visually

$f(x,y) =$ 

$= F(u,v)$ $+ F(u',v')$ $+ F(u'',v'') \cdots$

$\times$ $\times$ $\times$

Example of FT

center, where u=0,v=0, value is the largest, and in the corners where u and v are large, the magnitude of Fourier Coefficients are small

so that fewer number of very high frequency components are needed to represent this image.

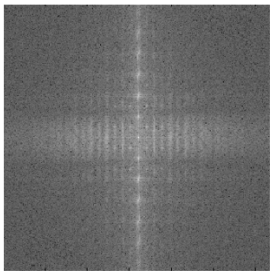
Effect of additional components

higher frequency$\rightarrow$ more details

Importance of the magnitude in FT

- Image with periodic structure

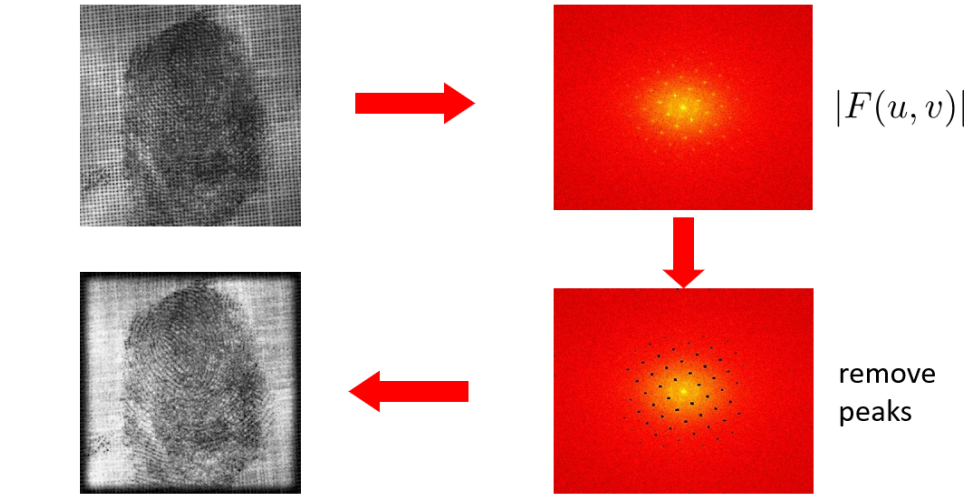
$f(x,y)$



$|F(u,v)|$

- **FT has peaks at spatial frequencies of repeated texture**

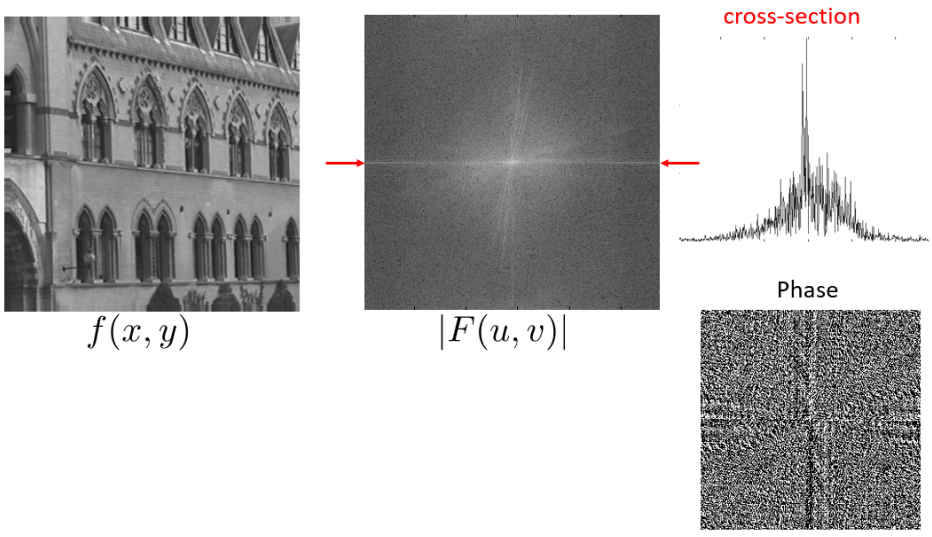
Importance of the magnitude in FT



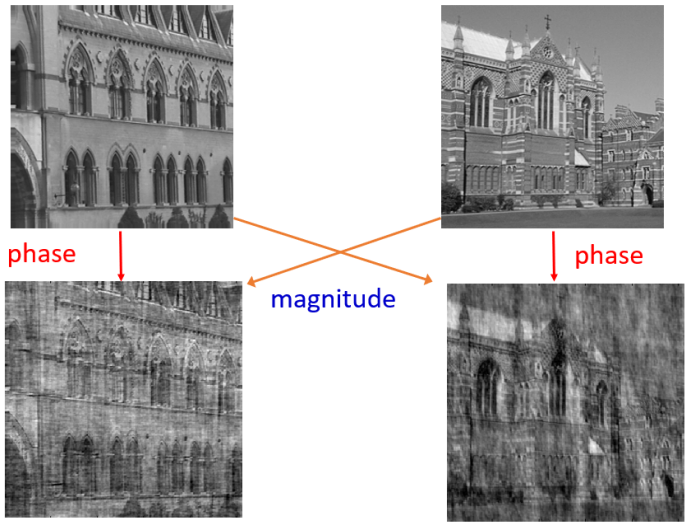
Periodic background removed

General structure of the magnitude

- Magnitude generally decreases with higher spatial frequencies
- phase appears less informative



- The bright certain lines in the middle image shows certain periodic structures
  The cross-section indicates that higher frequency components have smaller magnitude.



The convolution theorem

- $c(x,y) = a(x,y) * b(x,y)$
  What is the FT of a convolution?

$$
\begin{aligned}
C(u,v) &= \int\int [a(x,y) * b(x,y)] e^{-i2\pi(ux+vy)}\, dxdy \\
&= \int\int [\int\int a(x-\alpha, y-\beta) b(\alpha,\beta) d\alpha d\beta] e^{-i2\pi(ux+vy)}\, dxdy \\
&= \int\int [\int\int a(x-\alpha, y-\beta) e^{-i2\pi(ux+vy)}\, dxdy] b(\alpha,\beta) d\alpha d\beta \\
&= \int\int [\int\int a(x',y') e^{-i2\pi(ux'+vy')}\, dx'dy'] b(\alpha,\beta) e^{-i2\pi(u\alpha+v\beta)}\, d\alpha d\beta \\
&= \int\int A(u,v) b(\alpha,\beta) e^{-i2\pi(u\alpha+v\beta)}\, d\alpha d\beta \\
&= A(u,v)B(u,v)
\end{aligned}
$$

- **Space convolution = frequency multiplication**

Reciprocity in convolution theorem

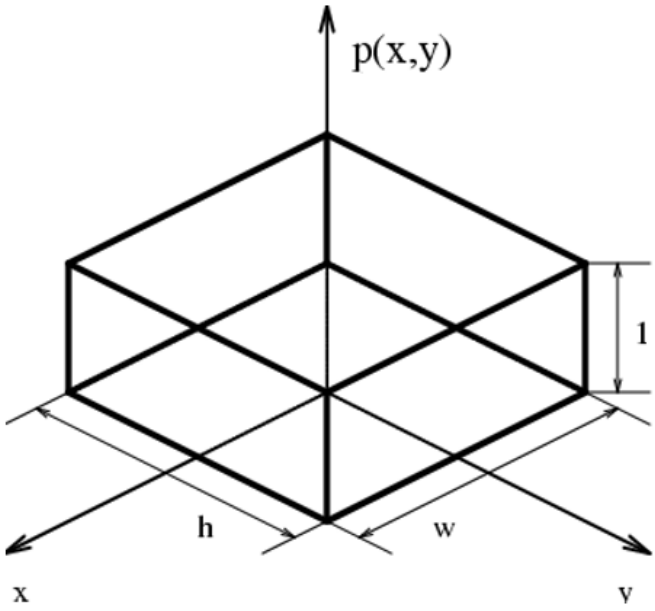
- $C(u,v) = A(u,v)B(u,v) \Leftrightarrow c(x,y) = a(x,y) * b(x,y)$
  $C(u,v) = A(u,v) * B(u,v) \Leftrightarrow c(x,y) = a(x,y)b(x,y)$
- **Space multiplication = frequency convolution**

Point spread function and Modulation transfer function

- When we talk about an imaging system where there is an image $i(x,y)$ and a kernel $r(x,y)$ that convolves the image, it is common to call the kernel the point spread function
- The convolution spreads the intensities to adjacent pixels based on $r(x,y)$
  Widely used terminology in microscopic imaging

- $O(u,v) = \mathcal{F}\{o(x,y)\} = \mathcal{F}\{i(x,y) * r(x,y)\} = I(u,v)R(u,v)$
  $R(u,v) = \mathcal{F}\{r(x,y)\} = \mathcal{F}\{\text{point spread function}\} = \text{modulation transfer function}$

Recall: Integrating over a cell window



$o(x',y') = \int\int i(x,y)p(x'-x,y'-y)dxdy$ (modified)

- Assuming p(x,y) is symmetric around the origin
- From convolution theorem, $O(u,v) = I(u,v)P(u,v)$

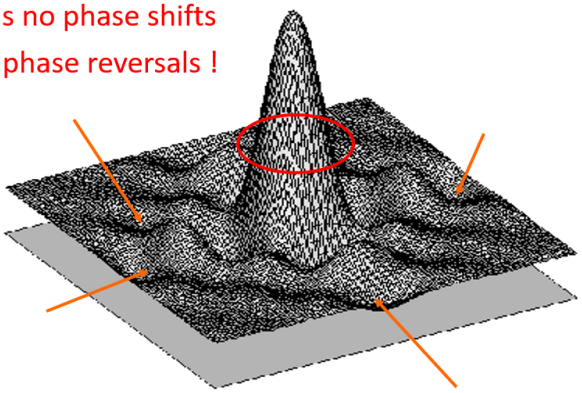
Modulation transfer function of the window function

- Fourier transform of window:

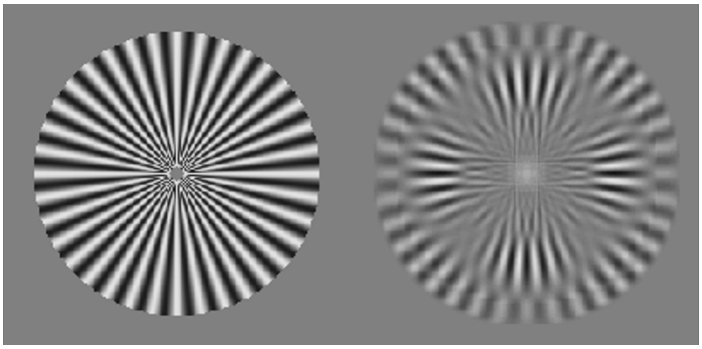
$$\begin{aligned} P(u,v) &= \int\int e^{-i2\pi(ux+vy)}p(x,y)dxdy \\ &= \int_{-w/2}^{w/2} e^{-i2\pi ux}dx \int_{-h/2}^{h/2} e^{-i2\pi vy}dy \\ &= wh(\frac{sin(\pi wu)}{\pi wu})(\frac{sin(\pi hv)}{\pi hv}) \leftarrow \text{2D sinc function} \end{aligned}$$

Modulation transfer function – 2D sinc

predominantly low-pass
real means no phase shifts
however, phase reversals !



- Illustration of the effect of 2D sinc

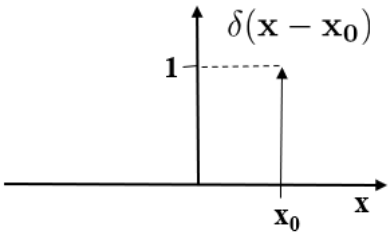


Summary for STEP I

- Convolve with a window function – rectangular box
- Blurs the image
- May cause phase reversals in certain frequencies – modify the image content

Local probing of functions

- To understand the effect of Step II, we need the probing function: Dirac pulse
  $\delta(\mathbf{x}-\mathbf{x}_0)=0 \quad \mathbf{x}\neq\mathbf{x}_0$
  $\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\delta(\mathbf{x}-\mathbf{x}_0)d\mathbf{x}=1$

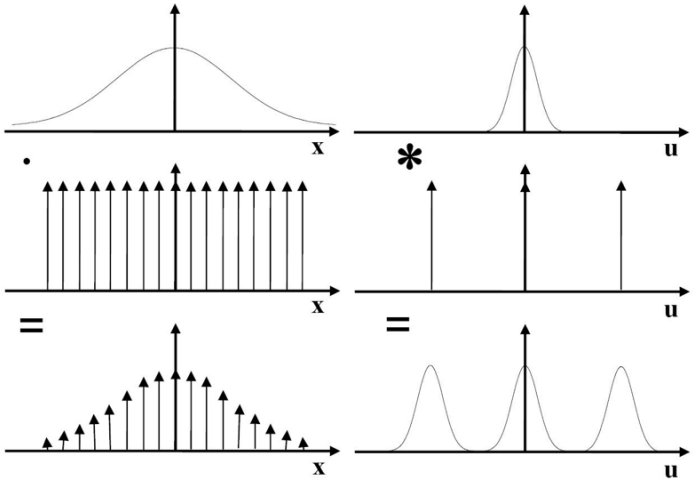


- Function probing (in 1D)
  $\int_{-\infty}^{\infty}\delta(x)f(x)dx=f(0)$
  $\int_{-\infty}^{\infty}\delta(x-x_0)f(x)dx=f(x_0)$

Discretization in the spatial domain is multiplication with a Dirac train

- 2D Dirac train / Dirac comb: $\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(x - kw, y - lh)$
- Fourier transform is also a Dirac train / Dirac comb:
  $\frac{1}{wh} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(u - k\frac{1}{w}, v - l\frac{1}{h})$
- Convolution with a Dirac train: periodic repetition
  Yet another duality: discrete vs. periodic

Effect on the frequency domain



- After sampling you may not get back the original signal
- It depends on the frequency domain representation, only band limited signals can be sampled and retrieved back
- Even then you need to sample at a certain rate

The sampling theorem

- If the Fourier transform of a function $f(x, y)$ is zero for all frequencies beyond u_b and v_b, i.e. if the Fourier transform is **band-limited**, then the continuous periodic function $f(x, y)$ can be completely reconstructed from its samples as long as the sampling distances w and h along the x and y directions are such that $w \leq \frac{1}{2u_b}$ and $h \leq \frac{1}{2v_b}$

Summary for STEP II

- When we read off one value per pixel area, we are losing information on the image indefinitely, if the image is not band-limited, which is almost always the case.
- The information we lose is on the higher frequencies, meaning very fine details on edges, corners and texture patterns.

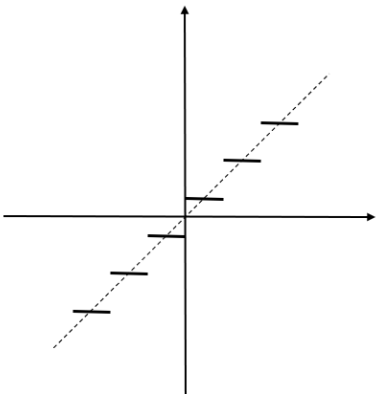
Quantization

- Create K intervals in the range of possible intensities and each interval with only one value
- Measured in bits: $log2(K)$

- Design choices:
  - Decision levels / boundaries of intervals $z_1, z_2, ..., z_{K-1}$
  - Representative values for each interval $[z_i, z_{i+1}] \rightarrow q_i$
- Simplest selection
  - Equal intervals between min and max
  - Use mean in the interval as the representative value
  - Uniform quantizer
  - $K = 256$ is used very often in practice

The uniform quantizer

- Simple interpretation
- Fine quantization is needed for perceptual quality (7-8 bits)
- It can be better designed if we know what intensities we expect
  $min \sum_{k=1}^{K} \int_{z_k}^{z_{k+1}} (z - q_k)^2 p(z) dz$
- $p(z)$ is the probability density function of intensities – constant for uniform quantizer



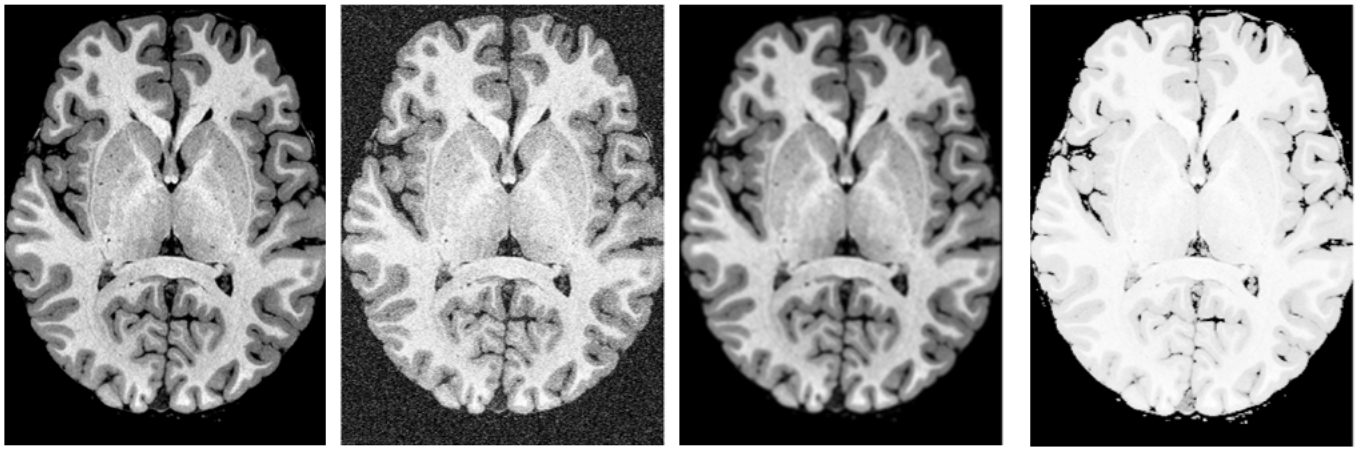
Underquantization examples(different gray level)
Small remarks on quantization

- 8 bits is often used in monochrome images
- 24 bits (8 x 3) used for RGB images per pixel
- Medical imaging may require finer quantization. 12 bits (4096 levels) and 16 bits (65536) are often used.
- Satellite imaging also use 12 or 16 bits regularly.

Image Enhancement

Three types of image enhancement

1. Noise suppression
2. Image de-blurring
3. Contrast enhancement

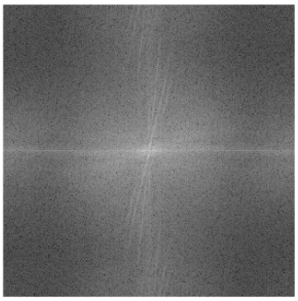


More on Fourier transform

Signal and noise

Fourier power spectra of images

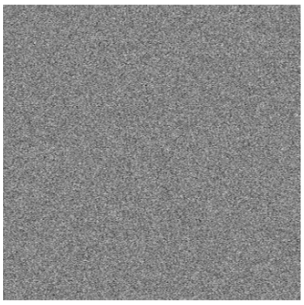
$i(x,y)$



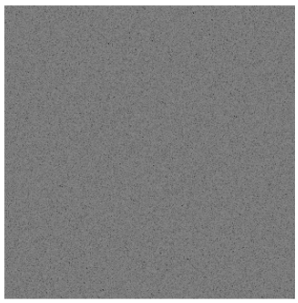
$\phi_{ii} = |I(u,v)|^2$

- Amount of signal at each frequency pair
- Images are mostly composed of **homogeneous** areas
- Most **nearby** object pixels have **similar** intensity
- **Most** of the signal lies in **low** frequencies!
- **High** frequency contains the **edge** information!

Fourier power spectra of noise



$n(x,y)$



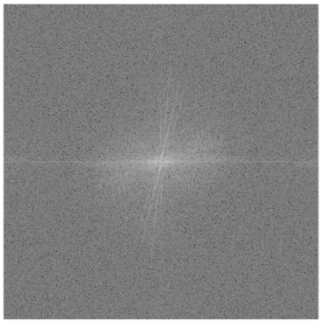
$\phi_{nn} = |N(u,v)|^2$

- Pure noise has a uniform power spectra

- Similar components in high and low frequencies.

Fourier power spectra of noisy image

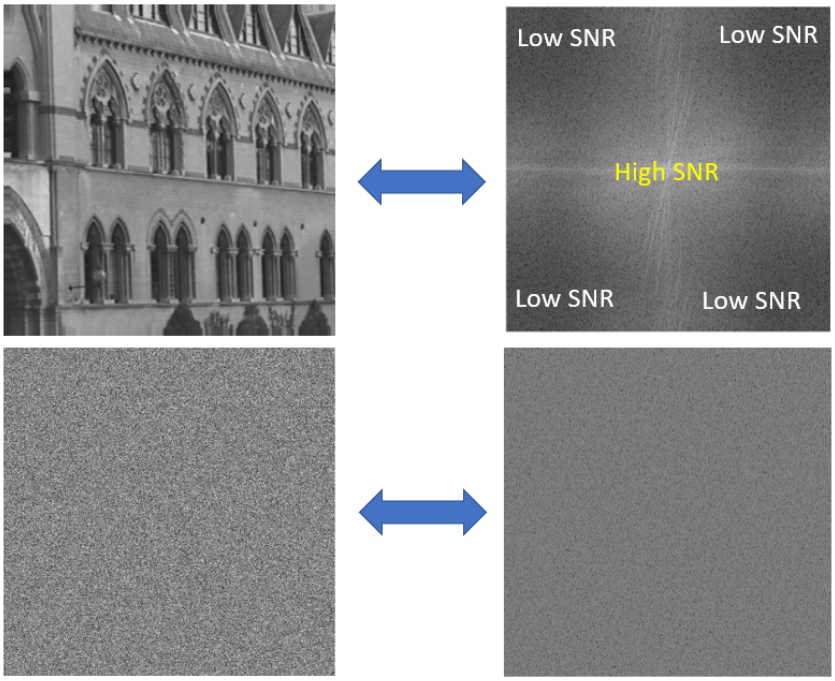
$f(x,y)$



$\phi_{ff} = |F(u,v)|^2$

- Power spectra is a combination of image and noise

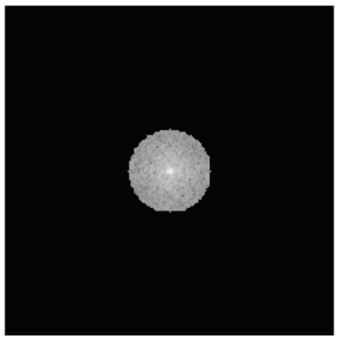
Signal to noise ratio (SNR)



$\phi_{ii}(u,v) \,/\, \phi_{nn}(u,v)$

Only retaining the low frequencies

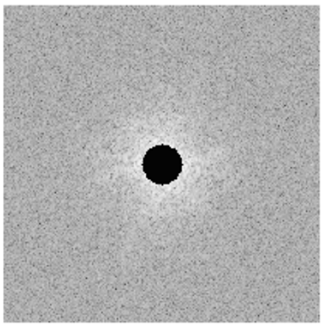
- Low signal/noise ratio at high frequencies #\Rightarrow$ eliminate these



- **Smoother** image but we **lost details**!

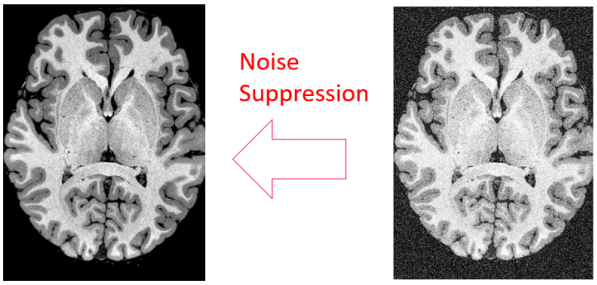
High frequencies contains noise and edge information

- We cannot simply discard the higher frequencies
- They are also introduced by edges

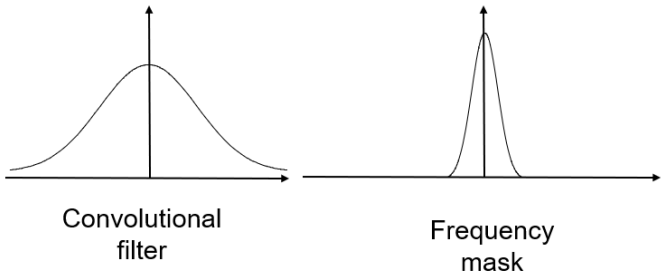
Noise suppresion

- In general specific methods for specific types of noise
- We only consider 2 general options here:
  - Convolutional linear filters – low-pass convolutional filters
  - Non-linear filters - edge-preserving filters
    - Median
    - Anisotropic diffusion



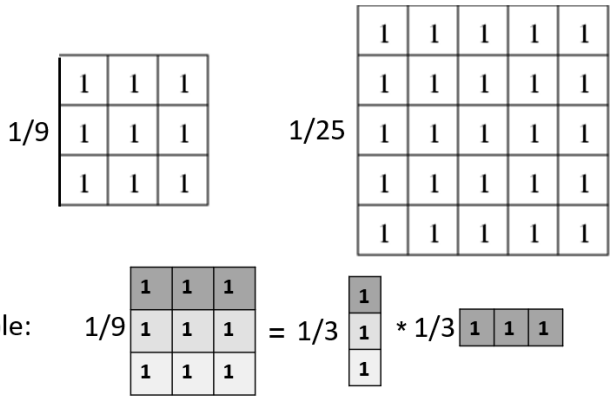
Low-pass filtering - principle

- Goal: remove low-signal/noise part of the spectrum
- Approach 1: Multiply the Fourier domain by a mask
  Such spectrum filters yield “rippling” due to ripples of the spatial filter and convolution
- Approach 2: Low-pass convolution filters
  generate low-pass filters that do not cause rippling
  Idea: Model convolutional filters in the spatial domain to approximate low-pass filtering in the frequency domain



Average filtering – Box filtering

One of the most straight forward convolution filters: averaging filters



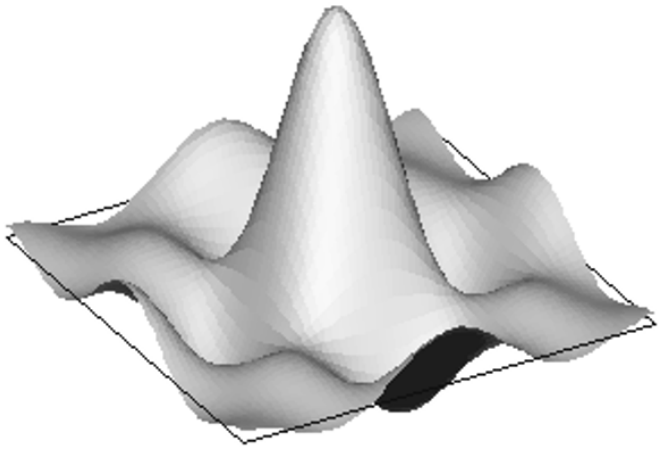
- $o(x,y) = f(x,y) * i(x,y) = f_1(x,y) * (f_2(x,y) * i(x,y))$

Example for box/average filtering

Noise is gone. Result is blurred!

MTF for box / average filtering

- 5×5(separable)
  $(1 + 2cos(2\pi u) + 2cos(4\pi u))(1 + 2cos(2\pi v) + 2cos(4\pi v))$
- not even low-pass!



So far

- Masking frequency domain with window type low-pass filter yields sinc-type of spatial filter and ripples -> disturbing effect
- box filters are not exactly low-pass, ripples in the frequency domain at higher freq. remember phase reversals?
- no ripples in either domain required!

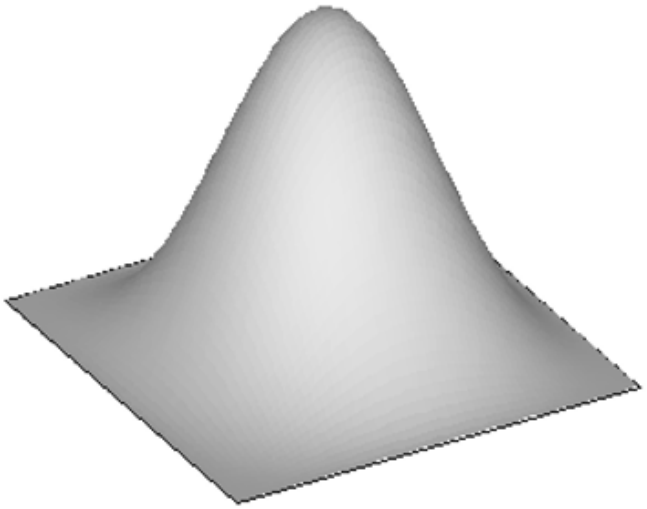
Solution: Binomial filtering

- iterative convolutions of $(1, 1)$
- only odd filters: $(1, 2, 1), (1, 4, 6, 4, 1)$
- also **separable**

2D :

1	2	1
2	4	2
1	2	1

- MTF: $(2 + 2cos(2\pi u))(2 + 2cos(2\pi v))$



Results of binomial filtering
Limit of binomial filtering

- $f(x, y) * f(x, y) * \cdots * f(x, y) = f^n(x, y)$
  $f^n(x, y) \rightarrow aexp(\frac{\|(x,y)\|^2}{b})$, as $n \rightarrow \infty$
- **Gaussian with b controlling the amount of smoothing**

Gaussian smoothing

Gaussian is limit case of binomial filters

- noise gone, no ripples, but still blurred...
- Actually linear filters **cannot** solve this problem

Some notes on implementation

- separable filters can be implemented efficiently
- large filters through multiplication in the frequency domain
- integer mask coefficients increase efficiency powers of 2 can be generated using shift operations
- In Gaussian filter increasing b (the standard deviation) leads to more smoothing and blurring

Questions

- Can convolutional filters do a perfect job?
  Can they separate edge information from noise in the higher frequency components?
  Why?

Median filtering: principle

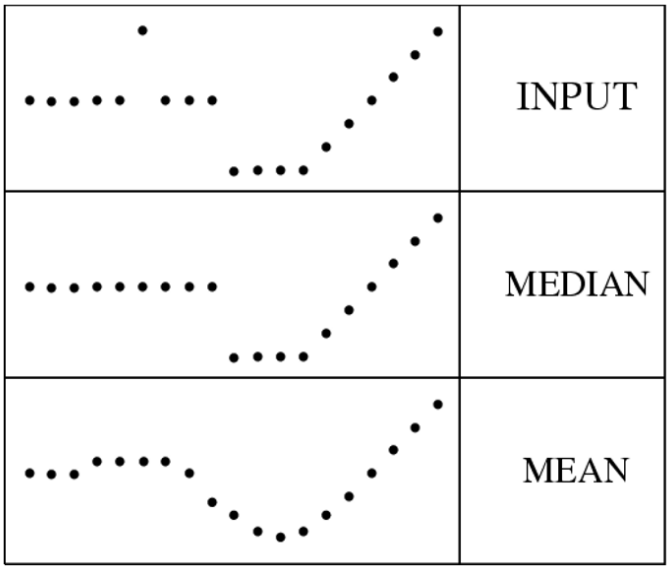
- Non-linear filter
- Simple method:
  - Rank-order neighborhood intensities in a patch of the image
  - Take middle value and assign it to the patch center
  - Go over all the image in a sliding window
- No new grey levels will emerge.

Median filtering – main advantage "odd-man-out'

- advantage of this type of filter is its “odd-man-out” effect
  e.g. $1, 1, 1, 7, 1, 1, 1, 1 \quad \rightarrow \quad ?, 1, 1, 1, 1, 1, 1, ?$

Example showing the advantage

- **Notice that the outlier is gone and sharp transitions (edge) are preserved**
- patch/box width 5



Median filtering – is it the solution to our blurring problem?

- median completely discards the spike, linear filter always responds to all aspects. Great for robustness to outliers and salt-and-pepper type noise
- median filter preserves discontinuities, linear filter produces rounding-off effects. Great for preserving sharp transitions, high frequency components and, essentially, edges and corners.
- **DON'T** become all too optimistic

Median filtering results

- sharpens edges, destroys edge cusps and protrusions

3 x 3 median filter :

Comparison with Gaussian :

Further results

- 10 times 3 X 3 median
- patchy effect: important details lost (e.g. ear-ring)


Question

- For what types of noise would you clearly prefer median filtering over Gaussian filtering?
  - Gaussian noise, i.e. noise distributed by independent normal distribution
  - Salt and pepper noise
  - Uniform noise, i.e. distributed by uniform distribution
  - Exponential noise model
  - Rayleigh noise

Anistropic diffusion: principle

- Non-linear filter
- More complicated method:
  - Gaussian smoothing across homogeneous intensity areas
  - No smoothing across edges

Gaussian filter revisited

- The diffusion equation
  $\frac{\partial f(\overrightarrow{x},t)}{\partial t} = \nabla \cdot (c(\overrightarrow{x},t)\nabla f(\overrightarrow{x},t))$
  Initial/Boundary conditions
  $f(\overrightarrow{x},0) = i(x,y)$, for $\overrightarrow{x} \in \Omega$
  $f(\overrightarrow{x},t) = 0$, for $\overrightarrow{x} \in \delta(\Omega)$
  If $c(\overrightarrow{x},t) = c$
  $\frac{\partial f(\overrightarrow{x},t)}{\partial t} = c\Delta f(\overrightarrow{x},t)$, in 1D: $\frac{\partial f(x,t)}{\partial t} = c\frac{\partial^2 f(x,t)}{\partial x^2}$

Solution is a convolution!
$f(\overrightarrow{x},t) = f(\overrightarrow{x},0) * g(\overrightarrow{x},t) = i(\overrightarrow{x}) * g(\overrightarrow{x},t)$

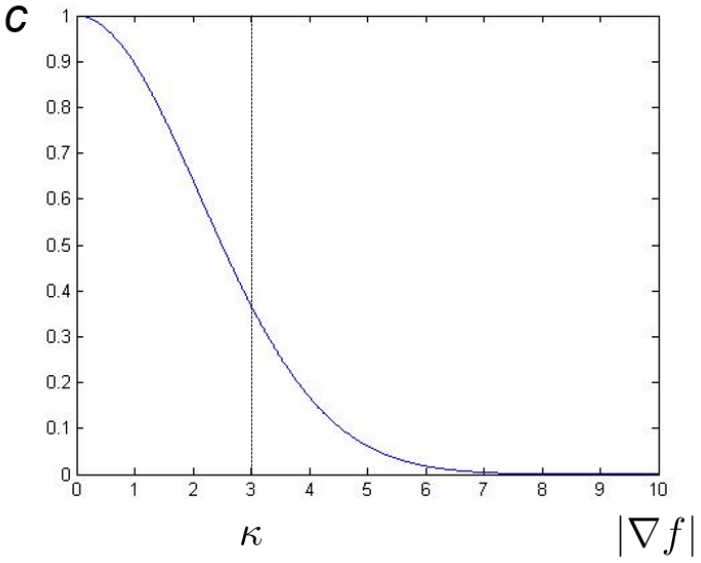
Diffusion as Gaussian low-pass filter

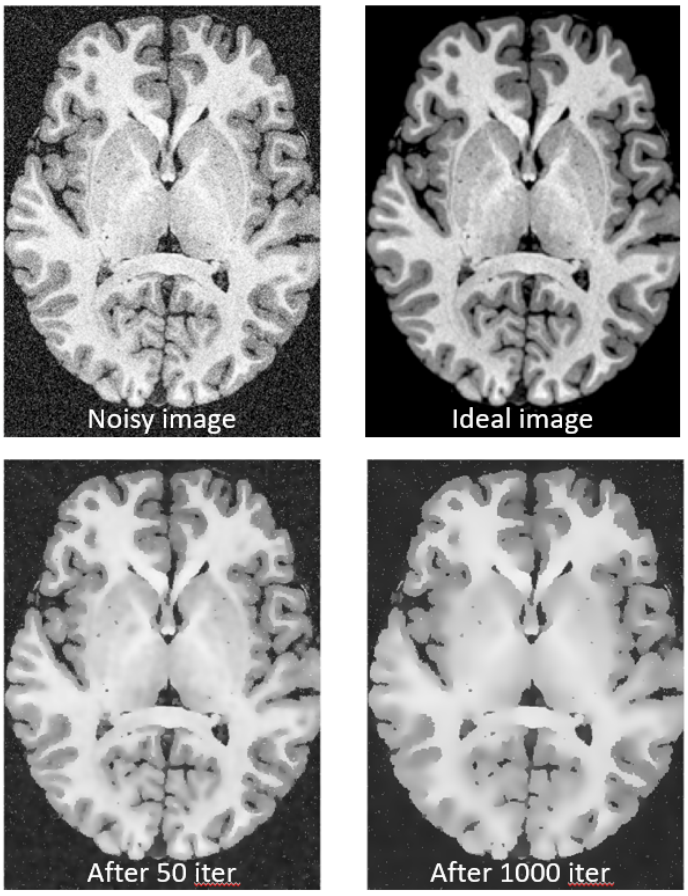
- $f(\overrightarrow{x},t) = i(\overrightarrow{x}) * \frac{1}{(2\pi)^{d/2}\sqrt{ct}} exp(-\frac{\overrightarrow{x}\cdot\overrightarrow{x}}{4ct})$
- Gaussian filter with time dependent standard deviation: $\sqrt{2ct}$
- Nonlinear version can change the width of the filter locally
  $c(\overrightarrow{x},t) = c(f(\overrightarrow{x},t))$
- Specifically dependening on the edge information through gradients
  $c(\overrightarrow{x},t) = c(|\nabla f(\overrightarrow{x},t)|)$

Selection of diffusion coefficient

- $c(|\nabla f(\overrightarrow{x},t)|) = exp(-\frac{|\nabla f|^2}{2\kappa^2})$
  or $c(|\nabla f(\overrightarrow{x},t)|) = \frac{1}{1+(\frac{|\nabla f|}{\kappa})^2}$
- κ controls the contrast to be preserved by smooting actually edge sharpening happens

Dependence on contrast

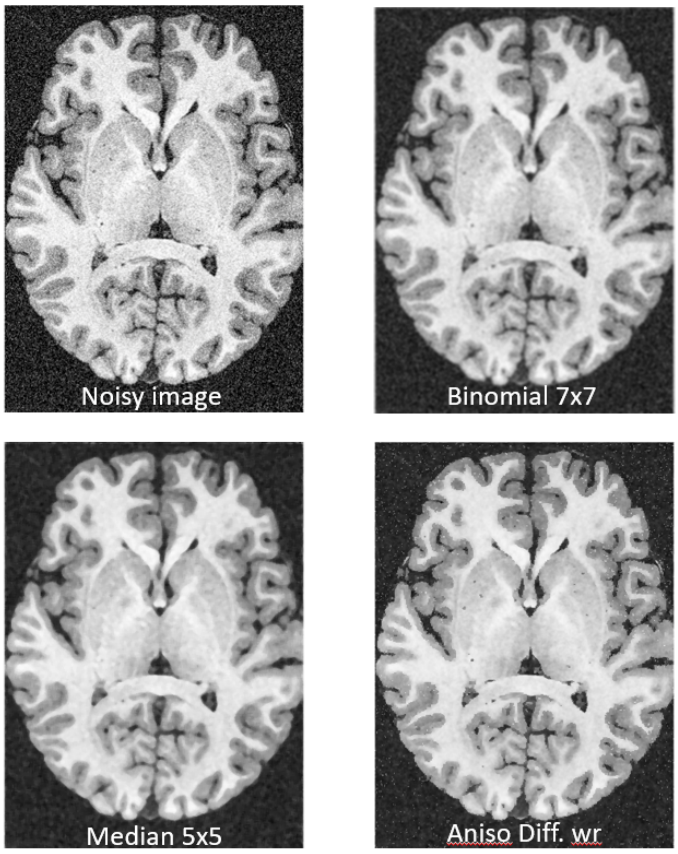




Unrestrained anisotropic diffusion

- End state is homogeneous

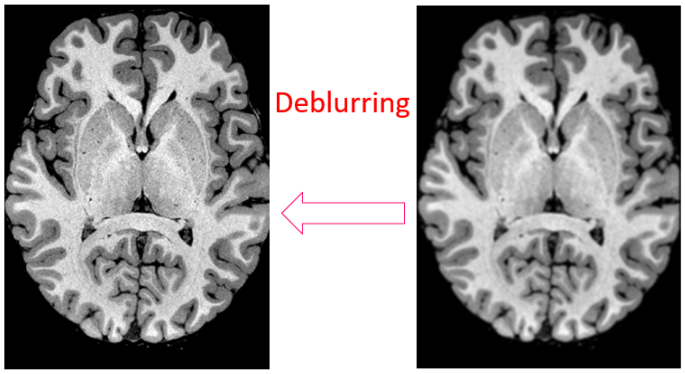
- adding restraining force:
  $\frac{\partial f}{\partial t} = \Delta \cdot (c(|\nabla f|)\nabla f) - \frac{1}{\sigma^2}(f - i)$



Anisotropic diffusion – numerical solutions

- When c is not a constant solution is found through solving the equation
  $\frac{\partial f(\vec{x},t)}{\partial t} = \nabla \cdot (c(\vec{x},t)\nabla f(\vec{x},t))$
- Partial differential equation
  - Numerical solutions through discretizing the differential operators and integrating
  - Finite differences in space and integration in time

Deblurring



Original Image
What we want

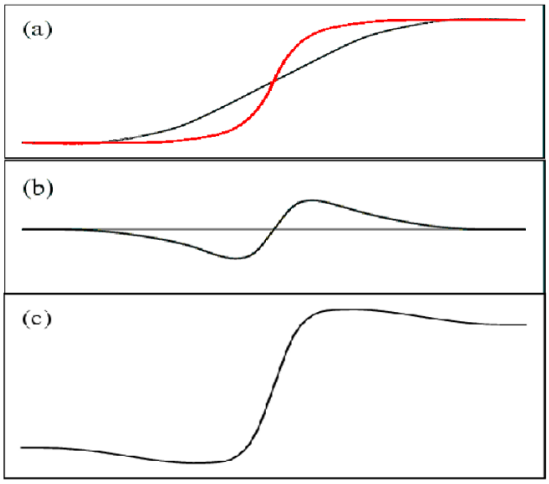
Blurred image
What we observe

Approach I: Unsharp masking

- simple but effective method

- image independent
- linear
- used e.g. in photocopiers and scanners

Unsharp masking - sketch



red = original
black = smoothed

orginal – smooth

original + difference

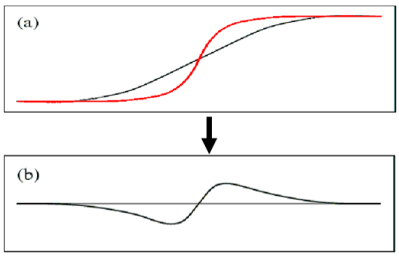
Unsharp masking - principle

- Interpret blurred image as snapshot of diffusion process
  $\frac{\partial f}{\partial t} = c(\nabla^2 f)$
  In a first order approximation, we can write
  $f(x, y, t) \approx f(x, y, 0) + \frac{\partial f}{\partial t} t$
  Hence,
  $f(x, y, 0) \approx f(x, y, t) - \frac{\partial f}{\partial t} t = f(x, y, t) - ct\nabla^2 f$
  Unsharp masking produces o from i
  $o = i - k\nabla^2 i$
  with k a well-chosen constant

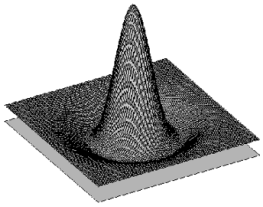
Need to estimate $\nabla^2 i(x, y)$

- DOG (Difference-of-Gaussians) approximation for Laplacian :

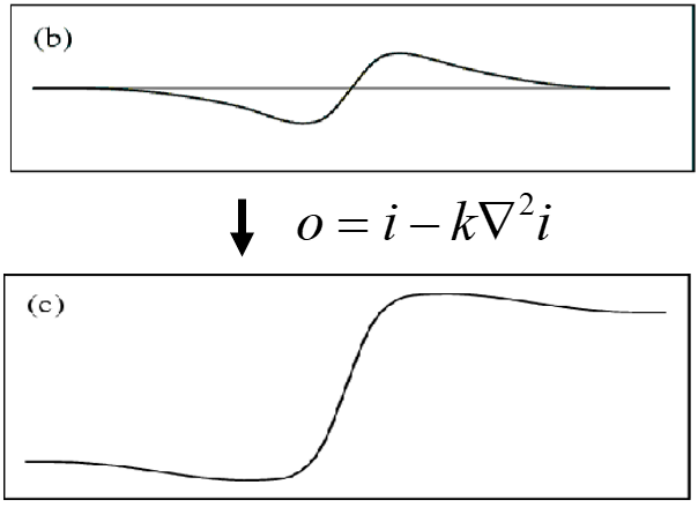
Our 1D example:



Convolution mask in 2D:



Unsharp masking analysis



- The edge profile becomes steeper, giving a sharper impression
- Under-and overshoots flanking the edge further increase the impression of image sharpness

Unsharp masking results

Approach II: Inverse filtering

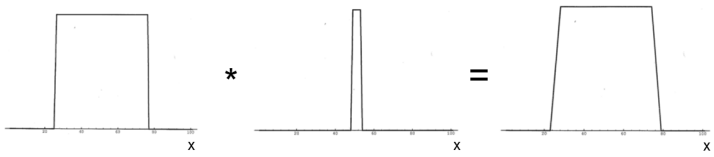
- Relies on system view of image processing
- Frequency domain technique
- Defined through Modulation Transfer Function
- Links to theoretically optimal approaches

Inverse filtering principle

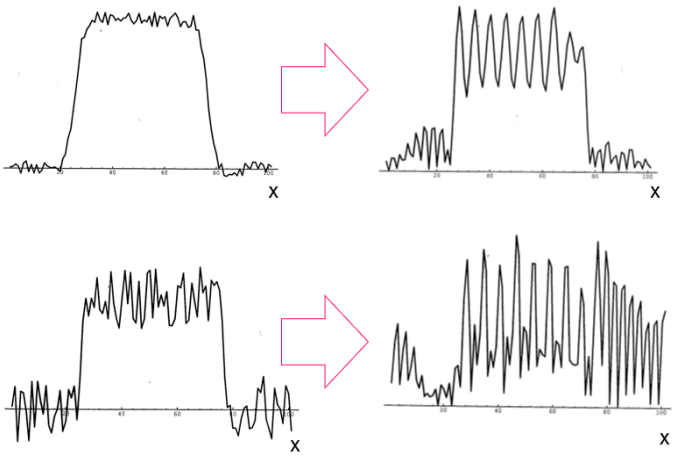
- Frequency domain technique
- suppose you know the MTF $B(u, v)$ of the blurring filter
  $f(x, y) = b(x, y) * i(x, y)$
  $F(u, v) = B(u, v)I(u, v)$
- to undo its effect new filter with MTF $B'(u, v)$ such that
  $B'(u, v)B(u, v) = 1$
  $I(u, v) = B'(u, v)F(u, v)$
  For additive noise after filtering
  $F(u, v) = B(u, v)I(u, v) + N(u, v)$
  Result of inverse filter
  $F(u, v)B'(u, v) = I(u, v) + N(u, v)/B(u, v)$
- Frequencies with $B(u, v) = 0$
  Information fully lost during filtering
  Cannot be recovered
  Inverse filter is ill-defined

- Also problem with noise added after filtering, $B(u,v)$ is low $1/B(u,v)$ is high, VERY strong **noise amplification**

1D example



Deblurring the noisy version



Inverse filtering example on an image

- we will apply the method to a Gaussian smoothed example (σ = 16 pixels)

- noise leads to spurious high frequencies

Wiener filter

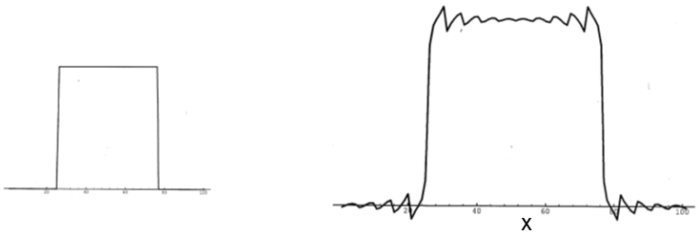
- Looking for the optimal filter to do the deblurring

- Consider the noise to avoid amplification
- A much better version of inverse filtering
- Optimization formulation
- Filter is given analytically in the Fourier Domain

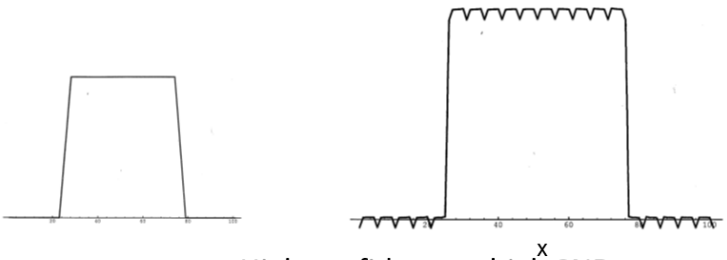
Wiener filter and its behavior

- $Wf(H) = H'(u,v) = \frac{H(u,v)}{H^*(u,v)H(u,v)+1/SNR}$
  where $SNR = \frac{\Phi_{ii}}{\Phi_{nn}}$
- $H(u,v) = 0 \Rightarrow Wf(H) = 0$
- $SNR \to \infty \Rightarrow Wf(H) \to \frac{1}{H}$
- $SNR \to 0 \Rightarrow Wf(H) \to 0$

Deblurring noise-free signal

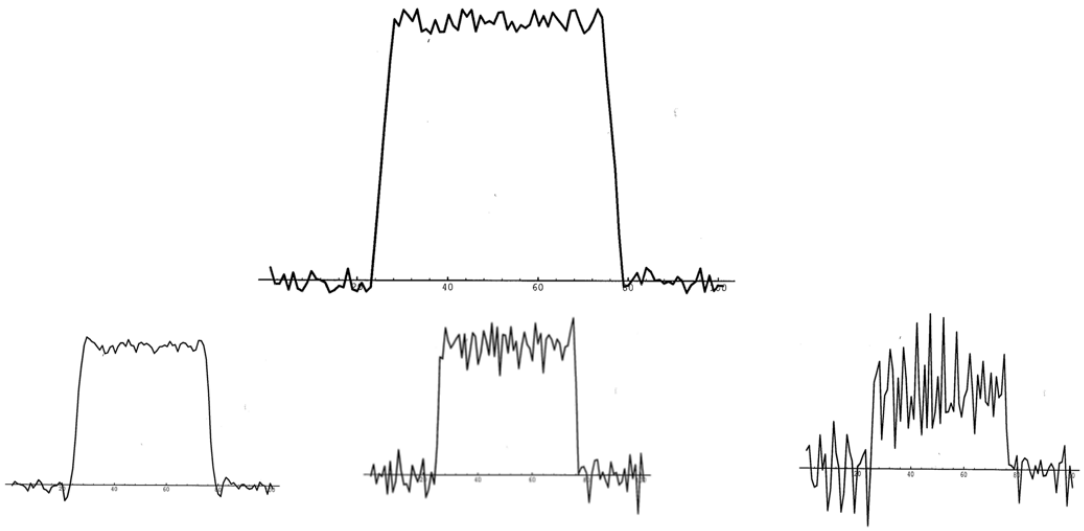


Medium confidence – medium SNR assumption



High confidence – high SNR assumption

Deblurring noisy signal



Low confidence | Medium confidence Correct SNR | High confidence

Wiener filtering example

- spurious high freq. eliminated, conservative

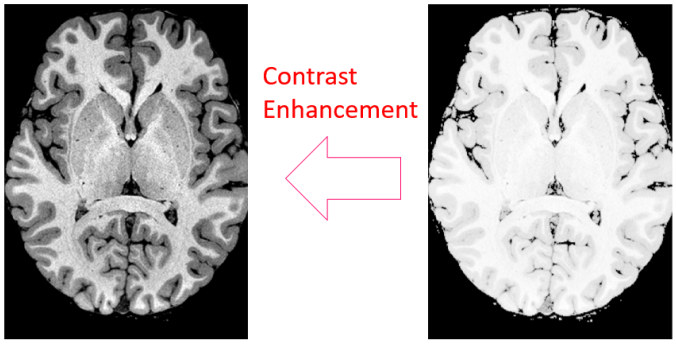
Problems in applying Wiener filtering

- $O(u,v) = Wf(H)(H(u,v)I(u,v)) = (Wf(H)H(u,v))I(u,v)$
  $Ef = Wf(H)$, H is the effective filter(should be 1)
- Conservative if SNR is low tends to become low-pass blurring instead of sharpening
- $SNR = \frac{\Phi_{ii}}{\Phi_{nn}}$ depends on $I(u,v)$ strictly speaking is unknown
- $H(u,v)$ must be known very precisely

Contrast Enhancement

Two use cases:

1. Compensating under-, over-exposure
2. Spending intensity range on interesting part of the image

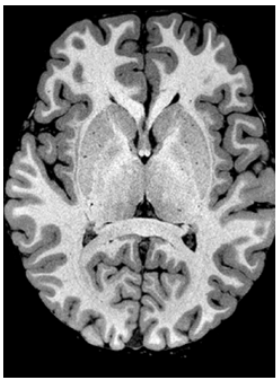


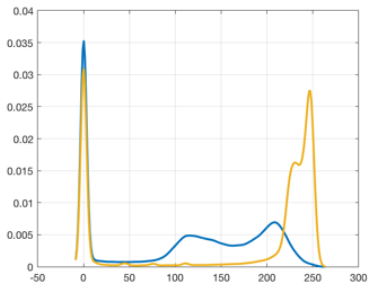
Original Image

Observation
with
Bad Contrast

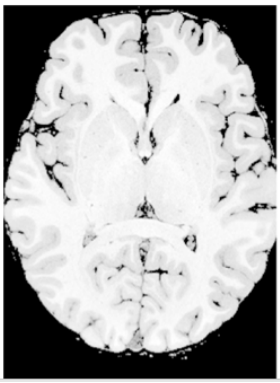
We will study histogram equalization

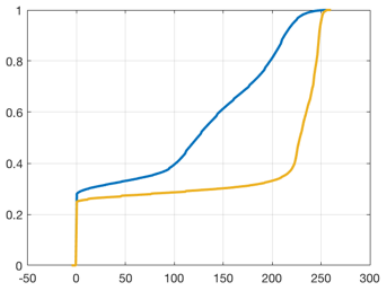
Intensity distributions - histogram





Histogram

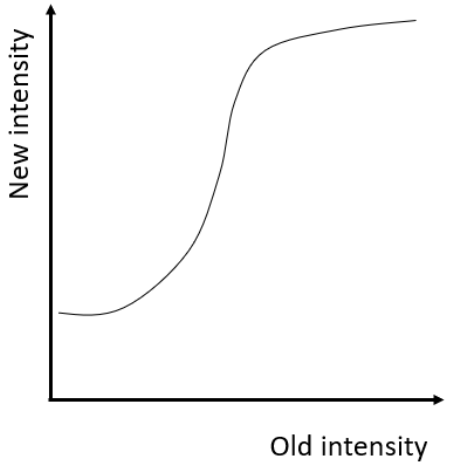




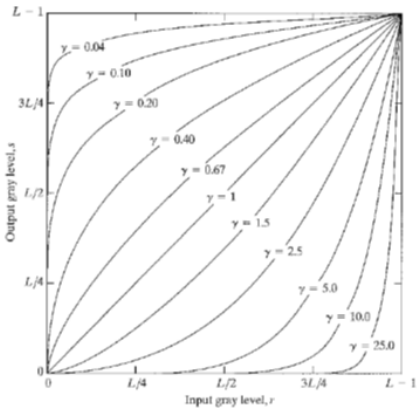
Cumulative histogram

Intensity mappings

- Usually monotonic mappings required



Generic transformation function

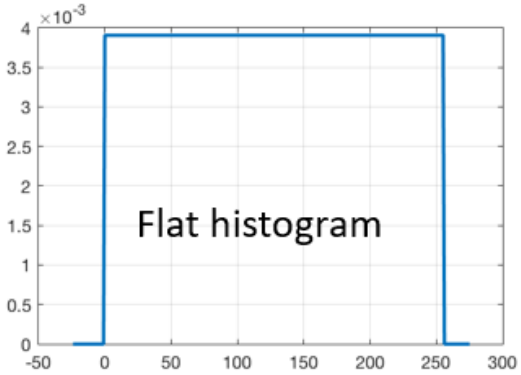


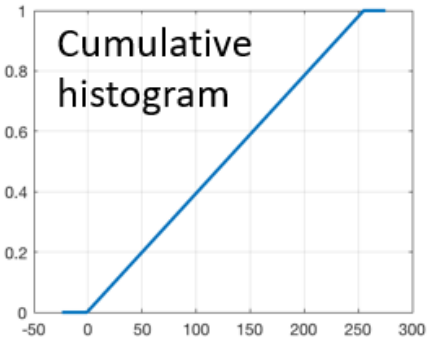
Power law transformation

$$I_{new} = I_{old}^{\gamma}$$

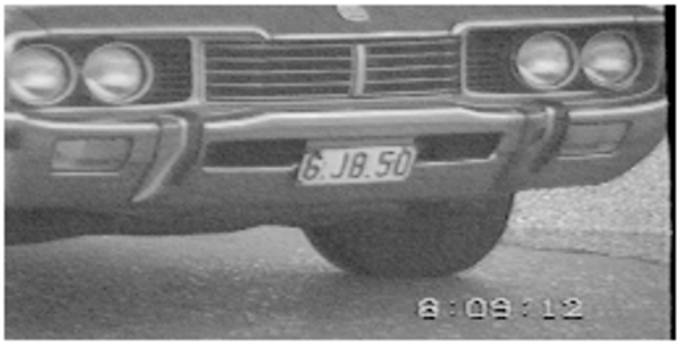
Histogram equalization

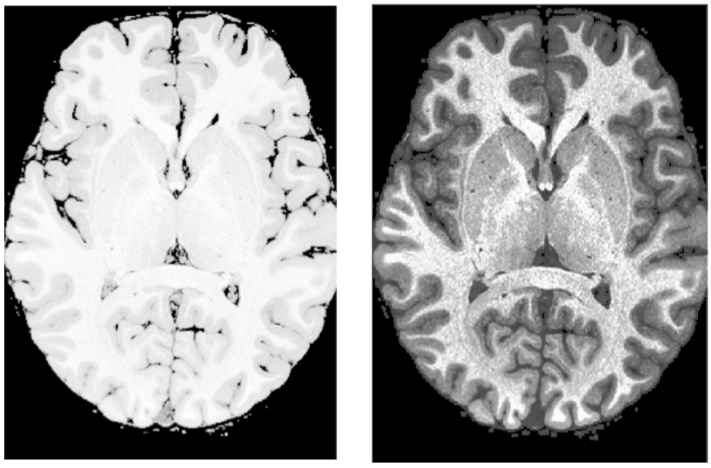
- Goal: create a flat histogram
- How: apply an appropriate intensity map depending on the image content method will be generally applicable





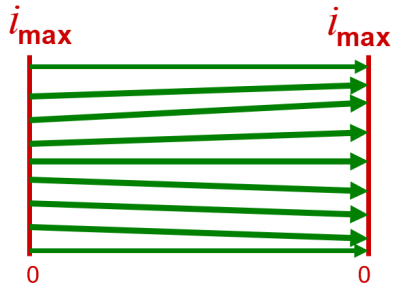
Histogram equalization example





Histogram equalization - principle

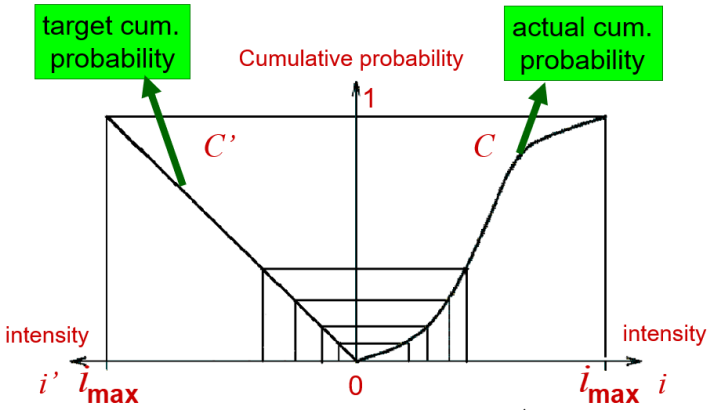
- Redistribute the intensities, 1-to-several (1-to-1 in the continuous case) and keeping their relative order, as to use them more evenly
- Ideally, obtain a constant, flat histogram



Histogram equalization - algorithm

- This mapping is easy to find:
  It corresponds to the **cumulative intensity probability** or cumulative histogram

Algorithm sketch



- $i' = T(i) = i_{max} C(i) = i_{max} \int_0^i p(j) dj$

Mathematical justification in continuous case

- suppose continuous probability density of original intensities i: $p(i)$
- Our mapping: $i' = T(i) = i_{max} \int_0^i p(j) dj$
- Probability density of the transformed intensities are given as
  $p(i') = p(i) \frac{di}{di'} = p(i) \frac{1}{p(i)} \frac{1}{i_{max}} = \frac{1}{i_{max}}$
- Indeed a flat distribution!

Feature Extraction

Lecture Notes

Features and matching

- Feature: description of a (part of) a pattern / object in the image, e.g. shape, texture, emitted heat if infrared…
- Mathematically: Describe the pattern with a vector of values
  $\mathbf{f} = [f_1, ..., f_n]$
- Goal : efficient matching for
  registration, correspondences for 3D, tracking, recognition,...

Specific object recognition

- Identifying the same object in multiple scenes
- Tracking objects in videos

- Finding correspondences between scenes
- Multiple applications in research and industry

Object category recognition

- Recognize different instances of the same object category
- For example, identify cars in this scene
- In addition to the challenges in specific object recognition, now intrinsic variation in the category also plays a role

Challenges
Variations in

- View point
- Illumination
- Background
- Occlusion
- Intra-class variation

Considerations when selecting features

- Complete (describing a pattern unambiguously) or **not**
- Robustness of extraction
- Ease and speed of extraction
- Global vs. **local** representation

Strategy in two parts

1. Identifying points of interest
2. Extract local features (vector of numbers) around the interest points

- Many features per image
- Representing different parts of many objects

Part I: Identifying “basic” points of interest

- A feature should capture something **discriminative** about a well localizable patch of a pattern
- We start with the well localizable bit
- Patch should be distnct with respect to its surrounding
- Shifting the patch a bit should make a big difference in terms of the underlying pattern
- We should be able to get the same point of interest under pose/lighting variations as much as possible.

Outline for points of interest

1. Edge detection

- Gradient operators
- Zero-crossings of Laplacians
- Canny edge detector

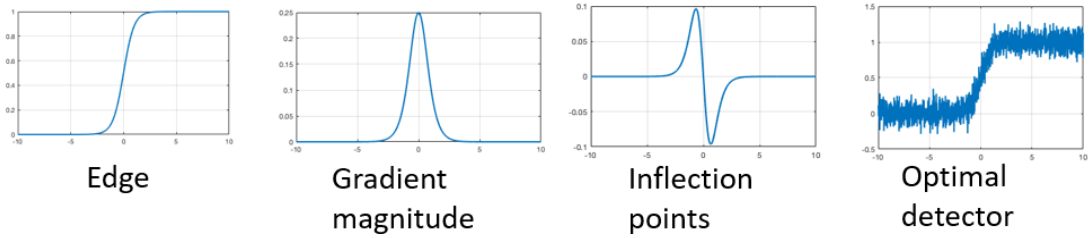
2. Corner detection
3. Blob detection

Edge detection

- Edges arise from changes in
  - Reflectance
  - Orientation
  - Illumination (e.g. shadows)
- Thus, not all edges are necessarily relevant to an object
- Methods introduced here form only the first step, linking edges to extract for instance boundaries of an object is a harder question

Edge detection methods

- We investigate three approaches
  1. Locating high intensity gradient magnitudes
  2. Locating inflection points in the intensit profile
  3. Signal processing view (optimal detectors) – Canny edge detector
- We will only consider isotropic operators – similar sensitivity in all directions



Gradient operators: principle

- Let us assume a continuous image $f(x, y)$
- Locate $f(x, y)$'s steepest slopes using gradient magnitude $\sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$
- Gradient magnitude measures the slope of the steepest direction at a given point.

Gradient magnitude

- Change of $f(x, y)$ in one v direction is given as $v = [cos\theta, sin\theta]$, $\frac{\partial f}{\partial v} = \frac{\partial f}{\partial x} cos\theta + \frac{\partial f}{\partial y} sin\theta$
- Differentiate with respect θ and set to 0 to find the max
  Angle of the maximum change: $\theta_{xtr} = tan^{-1}(\frac{\partial f / \partial y}{\partial f / \partial x})$
  Magnitude of the maximum change: $\sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$

Gradient operator: implementation

- Gradient magnitude is a non-linear operator
- But individual derivatives $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are linear
- They are actually linear and shift invariant operators
- Discrete approximations, finite differences can be estimated via convolution

$\frac{\partial}{\partial x} \approx$ | -1 | 1 | $\frac{\partial}{\partial y} \approx$ | -1 | 1 |

- Once these are computed, we can approximate the gradient magnitude and direction of maximum change using the results at each pixel.
- While easy to interpret and implement these operators are prone to noise. Noise will create high spurious derivatives
- We instead want a tool that will not be prone to noise.
- It should smooth in one direction and take the derivative in the other direction.

Gradient operators: Sobel

- Another discrete approximation

$\frac{\partial}{\partial x} \approx$ | -1 | 0 | 1 | / | -2 | 0 | 2 | / | -1 | 0 | 1 | $\frac{\partial}{\partial y} \approx$ | -1 | -2 | -1 | / | 0 | 0 | 0 | / | 1 | 2 | 1 |

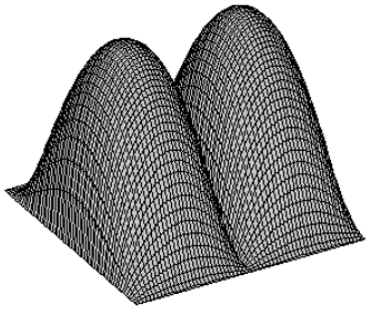
- These are called the **Sobel masks**.
- One mask primarily for compute horizontal and one for vertical pixel-wise derivatives
- Combine the results
  - Take the square root of the sum of their squares to compute the magnitude – approximates edge strength
  - Take arctan of their proportion to obtain the direction of maximal change – approximates edge direction
- Separable masks that are integer based – efficient and easy to implement

$\frac{\partial}{\partial x} \approx$ | -1 | 0 | 1 | / | -2 | 0 | 2 | / | -1 | 0 | 1 | $=$ | -1 | 0 | 1 | $*$ | 1 | / | 2 | / | 1 |

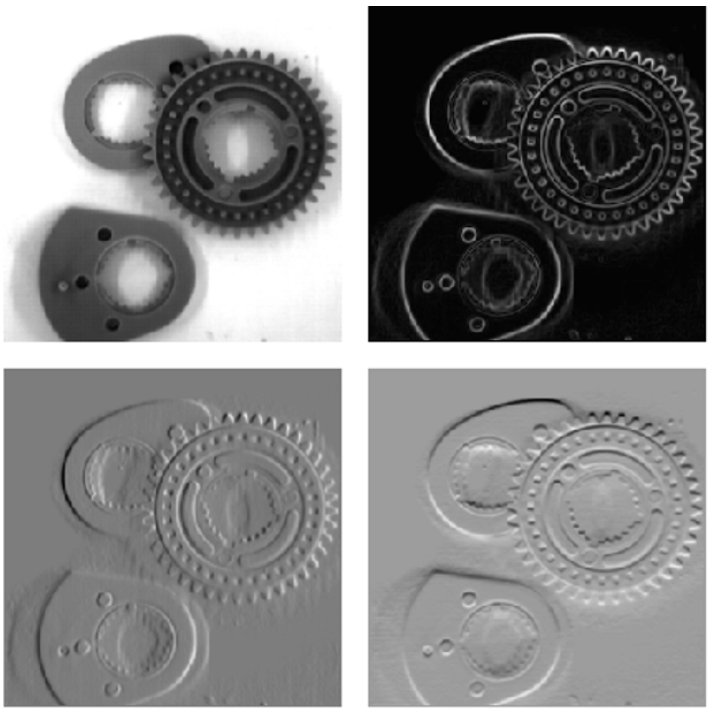
Notice something?

$$\frac{\partial}{\partial x} \approx \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} = \underbrace{\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline \end{array}}_{\text{Derivative operator}} * \underbrace{\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array}}_{\text{Binomial filter}}$$

- Power spectrum of the modulation transfer function $(2isin2\pi u)(2cos2\pi v + 2)$
- Low pass in one direction and band pass in the other



Gradient operators: example



Gradient operators: analysis

- Results are far from following the boundaries of object with a single line
  - Gaps
  - Several pixel thick at places
  - Some edges are weak whereas others are salient
- Sobel masks are the optimal 3 x 3 convolution filters with integer coefficients for step edge detection

Zero-crossings: principle

- Consider edges to lie at intensity inflection points
- Can be found at the zero-crossings of the Laplacian:
  $\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$
- linear and shift-invariant operator $\rightarrow$ convolution
- also isotropic operator

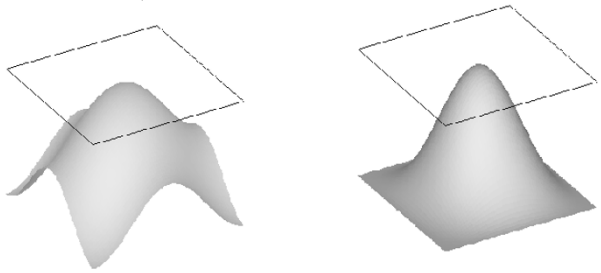
Discrete approximations of the Laplacian

0	1	0
1	-4	1
0	1	0

1	2	1
2	-12	2
1	2	1

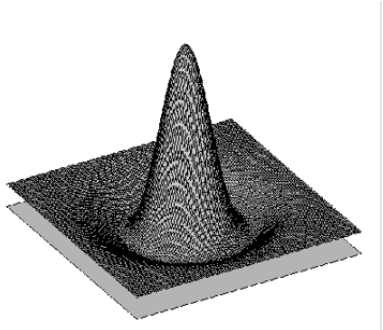
MTF: $2\cos 2\pi u + 2\cos 2\pi v - 4$

MTFs

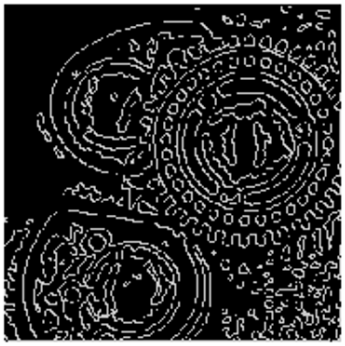


Zero-crossings: implementation

- Sensitive to noise (2nd order derivatives)
- It should be combined with smoothing, e.g. with a Gaussian
  $L * (G * f(x,y)) = (L * G) * f(x,y)$
- This $L * G$ is a "**Mexican hat**" filter used frequently
- Also called Laplacian of Gaussian (LoG)
- Also implemented as difference of Gaussians **DoG**



Zero-crossings: example

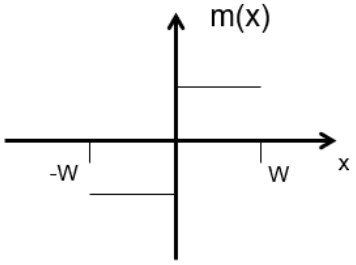
- One pixel-thick edges
- Closed contours
- Yet not very convincing results

The Canny edge detector

- A 1D signal processing approach
- Looking for “optimal” filters
- Optimality criteria:
  - Good SNR
    Strong response to edges and no response to noise
  - Good localization
    edges should be detected at the right position
  - Uniqueness
    edges should be detected only once

Characterization of SNR

- Response of system $\mathbf{h}$ to signal deterministic signal model $m(x)$ step edge at the origin
  $\mathbf{h}[m(x)] = \int_{-W}^{W} h(x - \hat{x})m(\hat{x})d\hat{x}$
- at the edge position
  $\mathbf{h}[m(0)] = \int_{-W}^{W} h(-\hat{x})m(\hat{x})d\hat{x}$



- Response to noise $n(x)$ noise is stochastic (Gaussian, white and uncorrelated)
  can be characterised by the expected value

$$\sqrt{\mathbb{E}[(\int_{-W}^{W} h(\hat{x})n(x-\hat{x})d\hat{x})^2]}|_{x=0} = \sigma\sqrt{\int_{-W}^{W} h^2(\hat{x})d\hat{x}}$$

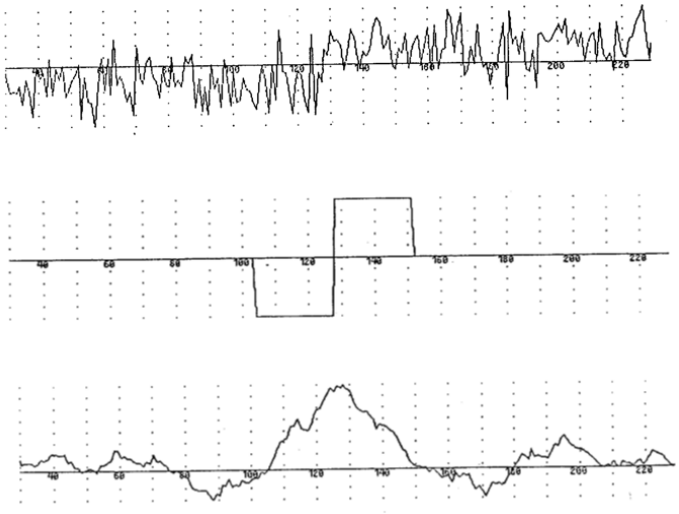
$$SNR = \frac{\int_{-W}^{W} h(-\hat{x})m(\hat{x})d\hat{x}}{\sigma\sqrt{\int_{-W}^{W} h^2(\hat{x})d\hat{x}}}$$

Characterization of localization

- Edge location: maximum of the system response extremum of $\mathbf{h}(m(x)+n(x))$ at x_0
  again stochastic (depending on the noise)
  will deviate from the ideal edge position at 0
- Quantification through expected value of the deviation from the real edge location
  $\sqrt{\mathbb{E}[x_0^2]}$
- Localization measure
  $LOC = \frac{1}{\sqrt{\mathbb{E}[x_0^2]}} = \frac{|\int_{-W}^{W} h'(\hat{x})m'(-\hat{x})d\hat{x}|}{\sigma\sqrt{\int_{-W}^{W}[h'(\hat{x})]^2 d\hat{x}}}$

The matched filter

- Optimal filter $\mathbf{h}(x)$ for which
  $max(SNR \times LOC)$
- This is given by $h(x) = \lambda m(x) \quad x \in [-W, W]$
- Essentially identical with the signal to be detected.
- For the edge model used: difference of boxes filter (DoB)



Uniqueness

- Filtering the DoB generates many local maxima due to noise
- Hinders unique detection
- Remedy: minimize the number of maxima within the filter support
- Caused by noise (stochastic)
  Characterized by the average distance between subsequent zero-crossings of the noise response

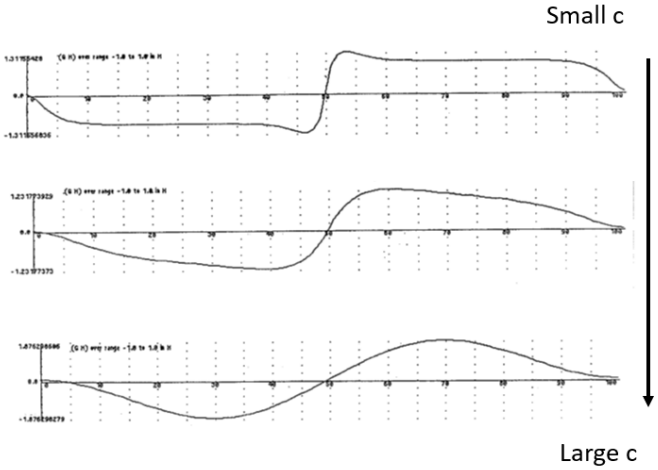
$(f = \mathbf{h}'(n))$

Rice theorem:

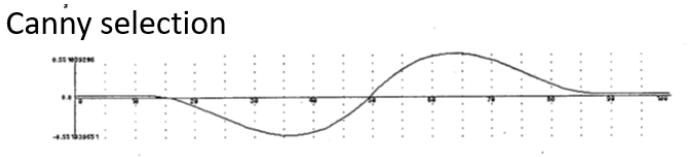
$x_{ave} = \pi \sqrt{\frac{-\Phi_{ff}(0)}{\Phi''_{ff}(0)}}$

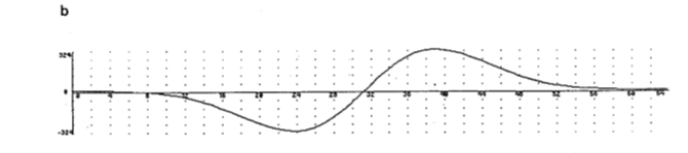
1D optimal filter

- Average number of maxima within filter support
  $N_{max} = \frac{2W}{x_max} = \frac{W}{x_{ave}}$
  should be minimized.
- Overall goal function is a linear combination of the two criteria
  $max(SNR \times LOC + c\frac{1}{N_{max}})$
  the solution depends on c, which is empirically selected

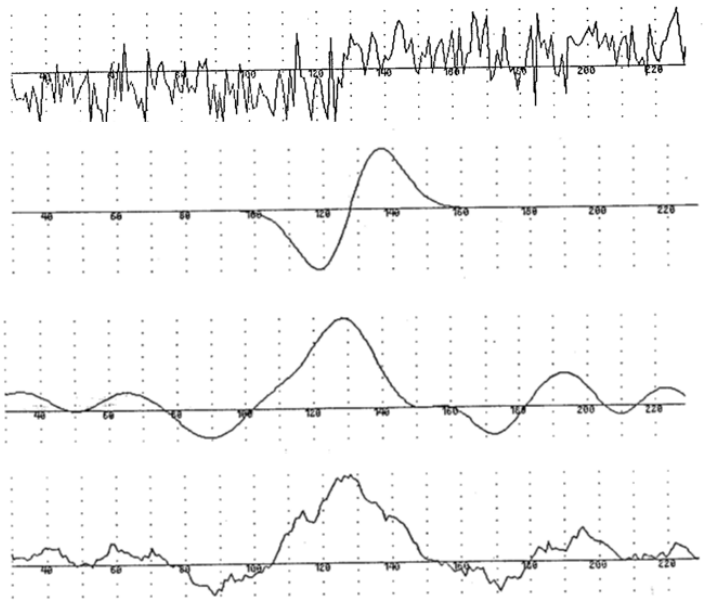


- Resembles the first derivative of the Gaussian





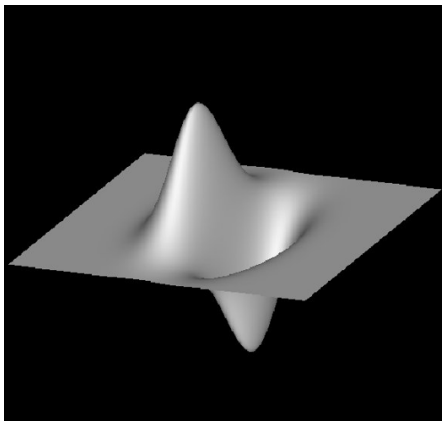
-



Canny filter in nD

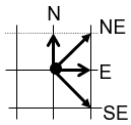
- Canny filter is essentially 1D
- Extensions to higher dimensions requires assumptions:
  - Simplified edge model
  - Intensity variation only orthogonal to the edge
  - No intensity change along the edge
- Combination of two filtering principles
  - 1D Canny filter across the edge
  - $(n-1)D$ smoothing filter along the edge
    Gaussian smoothing is used commonly
- The effective filter is a directional derivation of Gaussian

The 2D Canny filter



2D implementation on the discrete image raster

- Faithful implementation by selecting gradient direction: does not respect discretization
- Estimation of directional derivatives instead considering neighbours on the image raster

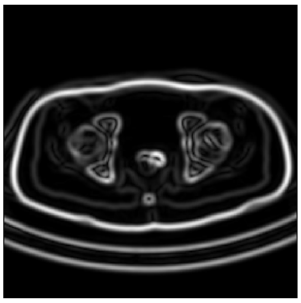


- Start with 2D Gaussian smoothing $f = G * I$
- Directional derivatives from discrete differences
  $f'_N = f(i, j+1) - f(i,j); f'_{NE}(i,j) = (f(i+1, j+1) - f(i,j))/\sqrt{2}$
  $f'_E = f(i+1, j) - f(i,j); f'_{SE}(i,j) = (f(i+1, j-1) - f(i,j))/\sqrt{2}$
- Selecting the maximum as gradient approximation, edge direction

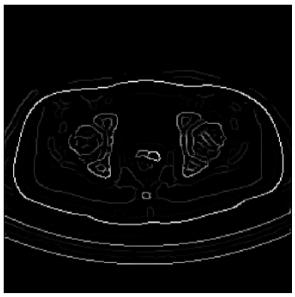
Canny 2D Results

Post-processing steps to get one line thick edges

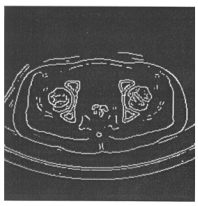
- Non-maximum suppression
  - Compare derivative magnitude at the two neighbours along the selected direction
  - Keeping only values which are not smaller than any of them
- Hysteresis thresholding
  - Using two threshold values t_{low} and t_{high}
  - Keep class 1 edge pixels for which $|f'(i,j)| \geq t_{high}$
  - Discard class 2 edge pixels $|f'(i,j)| < t_{low}$
  - For class 3 edge pixels $t_{high} > |f'(i,j)| \geq t_{low}$
    keep them only if connected to class 1 pixels possibly through other class 3 pixels



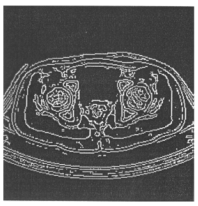
Gradient approximation



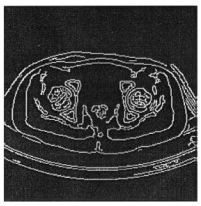
After non-maximum suppression



threshold t_{high}



threshold t_{low}



hysteresis thresholding

Remarks to Canny edge detection

- State-of-the-art non-learning-based edge detector
- Very efficient implementation – no interpolation is needed as respecting the raster
- Post-processing is a major contribution

- It can be applied to any gradient-based edge detection
- Fails where the simplified edge model is wrong
  - Crossings and corners
  - Gaps can be created
  - Mainly due to the non-maximum suppression step
- Hysteresis thresholding is effective in 2D where connectivity is easily defined.

Last remark: Match filter idea

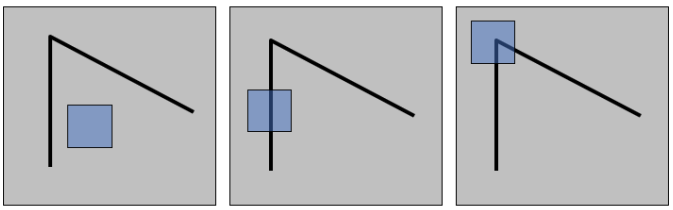
- This idea is very generic and can be used not only with edges but with any type of pattern
- For example the match filter for a line detector would be

-1	-1	-1
2	2	2
-1	-1	-1

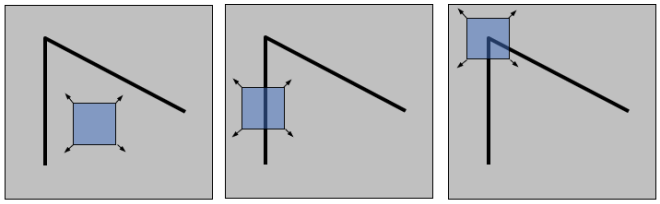
- This filter would need to be rotated to detect lines in different orientations.

Uniqueness of a patch

- Consider the pixel pattern within the blue patch



- Think of the wedge as being darker than the background, i.e. not as drawn in the figure…
- How do the patterns change upon a shift?



- Consider shifting the patch or 'window' W by (u, v)
- How do the pixels in W change?
- Compare each pixel before and after by summing up the squared differences (SSD)
- High change means distinct patch from its surrounding
- Low change means not distinct at all
- The SSD distance is defined as
  $E(u, v) = \sum_{(u,v)\in W}[I(x + u, y + v) - I(x, y)]^2$
  Taylor Series expansion of this term:

$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$ + higher order terms

If the motion (u,v) is small, then 1st order appr. is good

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \approx I(x,y) + [I_x\ I_y]\begin{bmatrix}u\\v\end{bmatrix}$$

Plugging this into the formula at the top…

- Then using the notation $I_x = \frac{\partial I}{\partial x}$

$$\begin{aligned}E(u,v) &= \sum_{(u,v)\in W}[I(x+u,y+v) - I(x,y)]^2 \\ &\approx \sum_{(u,v)\in W}[I(x,y) + [I_x\ I_y]\begin{bmatrix}u\\v\end{bmatrix} - I(x,y)]^2 \\ &\approx \sum_{(u,v)\in W}[[I_x\ I_y]\begin{bmatrix}u\\v\end{bmatrix}]^2\end{aligned}$$

- This can be rewritten further as

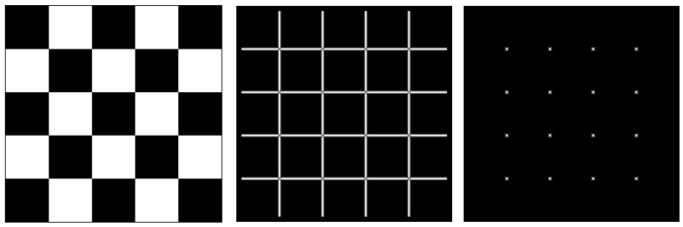
$$E(u,v) = \sum_{(u,v)\in W}[u\ v]\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}$$

$$E(u,v) = [u\ v]\sum_{(u,v)\in W}\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}$$

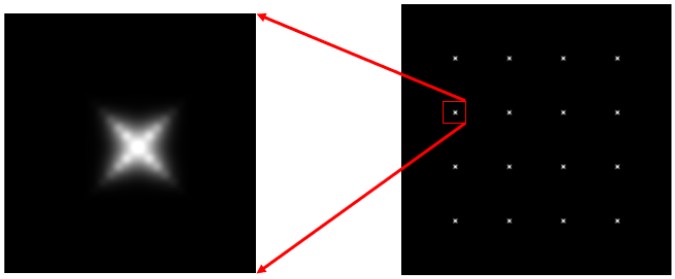
$$E(u,v) = [u\ v]H\begin{bmatrix}u\\v\end{bmatrix}$$ where H is structure tensor

- Which directions (u,v) will result in the largest and smallest E changes?
- $H = \sum_{(u,v)\in W}\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}$
- Eigenvalues and eigenvectors of H
  - Define shifts with the smallest and largest change in E value
  - x_+ = direction of largest increase in E
  - λ_+ = amount of increase in direction x_+
  - x_- = direction of smallest increase in E
  - λ_- = amount of increase in direction x_-
- We would like $E(u,v)$ to be large for small shifts in ALL directions

  This means the minimum λ_- of H should be large



Example image λ_+ λ_-



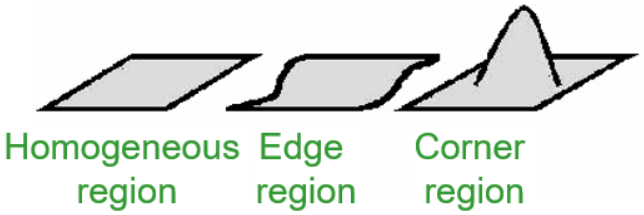
λ_-

Relation to interest points

- Corners are the most prominent example of interest points
- They can be well localized in different views of a scene – a corner is a corner, it has to be visible though
- Blobs are another as we will see
- Very similar to corners, a blob is a region with intensity changes in multiple directions, similar to corners

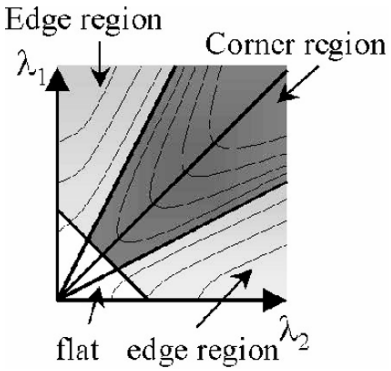
Harris corner detector

- Goal: one approach that distinguishes
  - Homogeneous areas
  - Edges
  - Corners

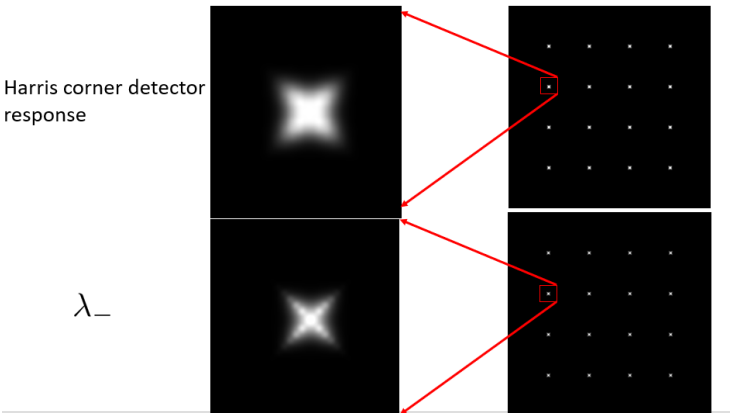


- Key: use the same idea with eigenvalues of the structure tensor, look for intensity variations in different directions
  - Small in all directions
  - Large in one direction, small in the other
  - Large in all directions
- The classification can be made as
  - Two small eigenvalues – homogeneous area
  - One large and one small eigenvalue – edge
  - Two large eigenvalues – corner
- We can actually look at two maps to decide
- Two eigenvalues would create two maps that we would need to combine
- To create one score we can look at the determinant of the structure tensor: product of the eigenvalues

- That does not work very well.
- One very large eigenvalue can still trigger a corner response
- Refined strategy: Use iso-lines for $Determinant - k(Trace)^2$.



Harris corner detector's response



Are corners stable under viewpoint and illumination changes?

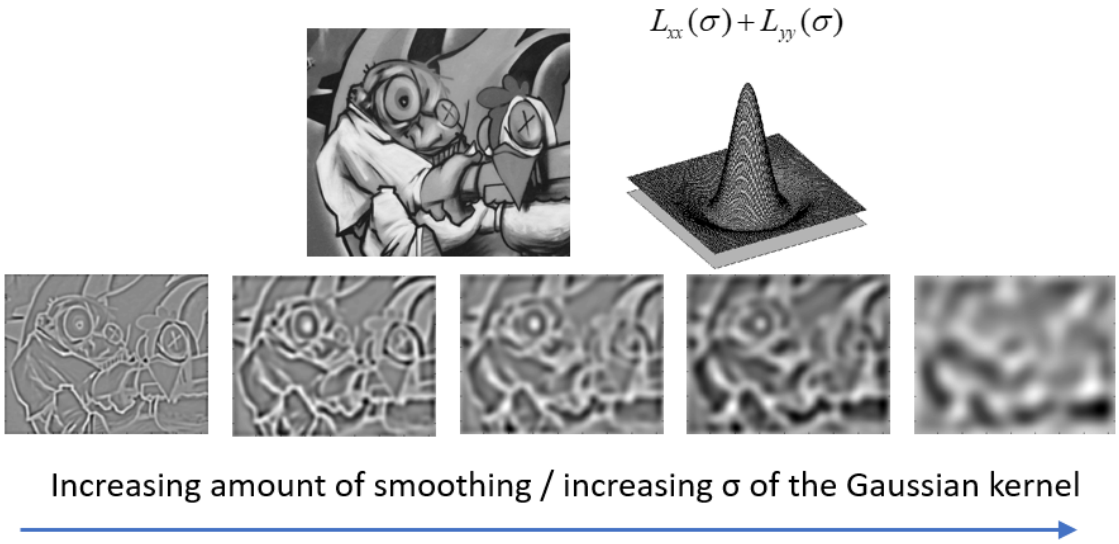
Blob

- A blob is an area of an image where some properties are roughly costant and differ from their surrounding.
- Similar to corners – it is different than its neighborhood in all directions.
- Its definition is less formal
- Property can change based on the aspect you look at:
  - Gray level
  - A color channel
  - Gradient magnitude
  - …

Blob detection through Laplacian of Gaussian (LoG)

- We have already seen Laplacian operator – 2nd order derivatives.
- Gaussian smoothing first is necessary leading to the Laplacian of Gaussian (LoG) or the Mexican hat filter.

Dependence of the smoothing amount



Part II: Extracting descriptors around interest points
Need for a descriptor

- A feature should capture something **discriminative** about a well **localizable** patch of pattern
- There are many corners or blobs coming out of our detectors but they still cannot be told apart
- We need to describe their surrounding patch in a way that we can discriminate betwee them, i.e. we need to build a feature vector for the patch, a so-called **DESCRIPTOR**
- During matching or recognition, the descriptors will be compared

Two components of a descriptor
This will be extracted for each point of interest

1. Define an area – local patch – in the image around each interest point.
2. Extract features from the local patch to represent each interest point with a vector of numbers

Additional requirements on the descriptors

- **Invariance** under geometric / photometric changes – if impossible then insensitivity
- Small local patches – can consider how they deform with geometric changes
- Intensity-based descriptors that do not change with illumination changes
- Invariance theory is rich and fun

Example I: Euclidean invariant features(Schmid and Mohr ‘97)

- Harris corner detector to identify corners as interest points
- Extract circular patches around each interest point.
- For each interest point circular areas of different radii are chosen – to account for scale * changes to some degree

- Extract invariant features under planar rotation from each area to form the final descriptors
- Very successful model

- Rotation invariant gradient magnitude $G_x G_x + G_y G_y$
  where G_x and G_y represent horizontal and vertical derivatives of intensity weighted by a Gaussian profile in the patch (“Gaussian derivatives”)
- Rotation invariant 2nd derivatives $G_{xx} + G_{yy}$
  where G_{xx} and G_{yy} represent 2nd order Gaussian derivatives

Invariance properties

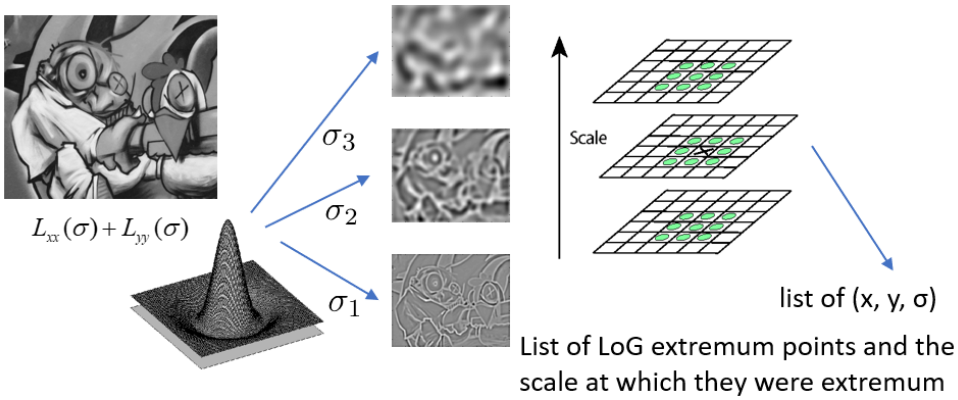
- Areas are rotationally invariant because they are circular
- Taking circles with multiple radii at each point achieve some level of robustness to scale changes – zoom
- Gradient magnitude is rotationally invariant $\rightarrow$ Can you show this?
- 2nd order derivatives is also rotationally invariant $\rightarrow$ Can you show this?
- Are the features invariant to scale changes?

Example 2: SIFT

- Scale-invariant Feature Transform
- Developed by David Lowe
- Carefully crafted interet point detection and descriptor based on intensity gradients
- Invariance to similarity transformations

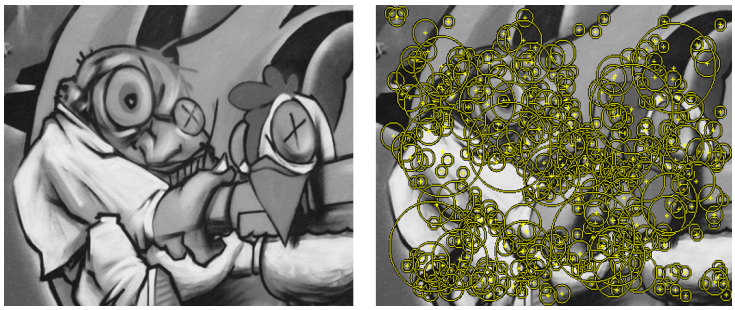
SIFT: Blob detection

- Interest points and patches around interest points are jointly determined through blob detection at multiple scales
- Determine local extrema of the LoG at each scale

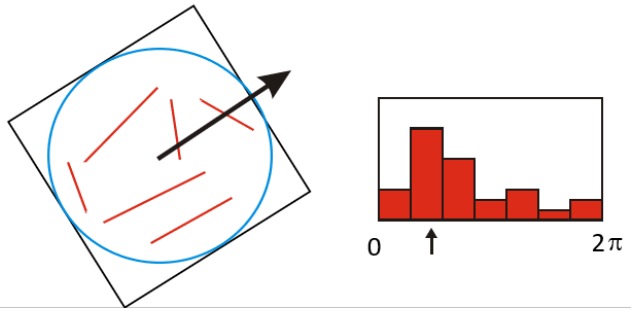


List of LoG extremum points and the scale at which they were extremum

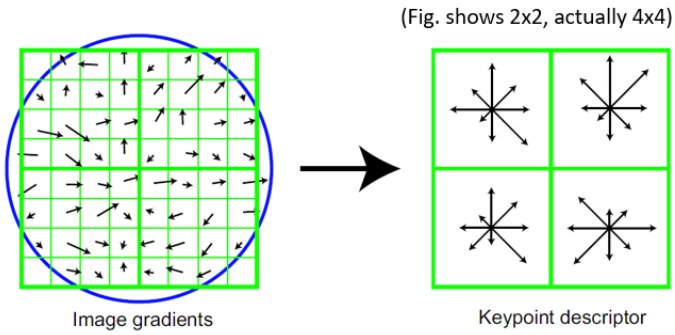
SIFT: Interet points and local patches



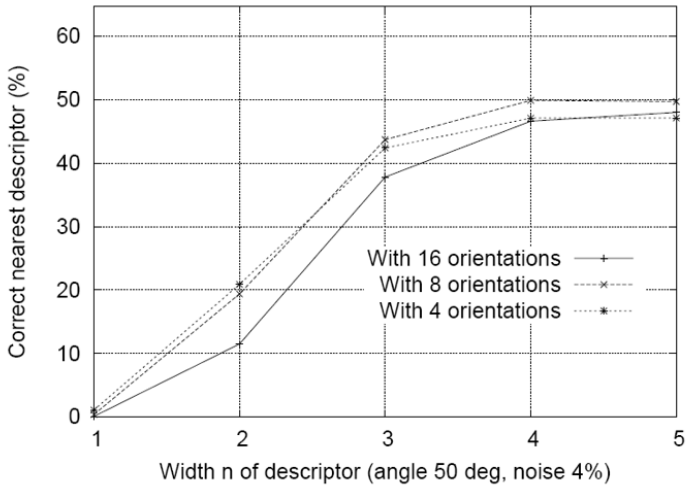
- Same patch can be rotated in any direction
- To gain invariance to such rotations SIFT implements **dominant orientation selection**
  - Compute image gradients
  - Build orientation histograms
  - Find maximum



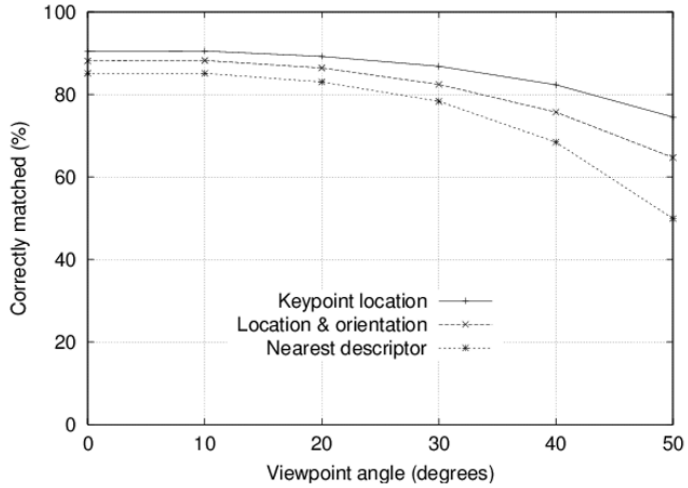
- Image gradients are sampled over a grid
- Create array of orientation histograms within blocks
- 8 orientations $\times$ 4x4 histogram array = 128 dimensions
- Apply weighting with a Gaussian located at the patch center
- Normalized to unit vector



SIFT is carefully crafted why 4x4x8?



Sensitivity to affine changes are good



Invariance properties

- Local patches are rotationally invariant because they are circular
- Scale invariance in patch selection
- Dominant direction selection leads to rotation invariance
- Gradient orientation histograms are invariant to scale changes

- Huge improvements over SIFT in computational efficiency, accuracy and robustness to different transformations:
  SURF: Speeded up robust features
  Bay, Ess, Tuytelaars, Van Gool

Principal Component Analysis

Remember the Fourier transform?

- It converted an image into another domain – frequency domain
- This can be applied to the entire image but also to parts of it
- New representation of the image or its parts
- Much like extracting features!
- Fourier transform is an all applicable unitary transform and it may not extract best possible features.
- Can we have transformations specific to certain image types?

Principal component analysis: Motivation

- Image independent transforms are suboptimal for extracting “useful” descriptors or representations
- Image dependent transformations may yield more useful representations of images
- PCA also known as Karhunen-Loève Transform (KLT)
- Extracts statistics from images for a **customized** orthogonal basis set with **uncorrelated weights**
- Customization refers to having a basis set specific to a given set of images instead of generic basis such as those in FT
  $e^{i2\pi(ux+vy)}$

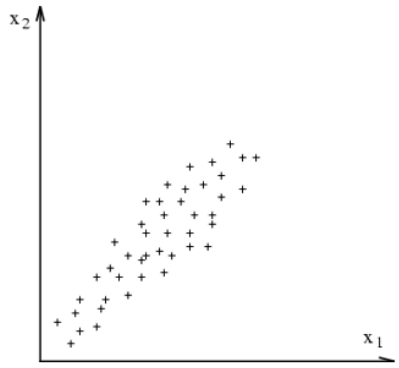
PCA: Central idea

- Reduce the dimensionality of data consisting of many interrelated variables, while retaining as much as possible of the variation
- Achieved by transforming to new, **uncorrelated** variables, the **principal components**, which are ordered so that the first few retain most of the variation
- Representation of the image under the new variables will be useful as features for recognition, classification, inspection,...

Correlated variables

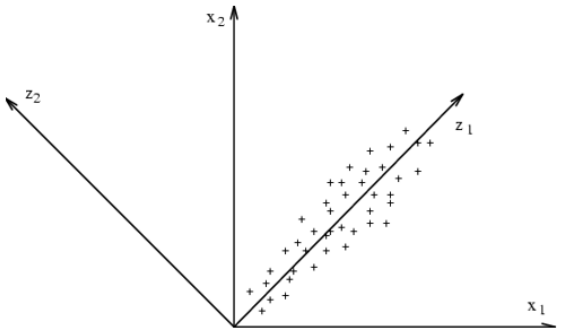
- Observations with two highly-correlated variables, e.g.
  - grey-level at neighbouring pixels

    - Length & weight of growing children
- Highly correlated values: x_1 has info on x_2
- We can represent both variables with only 1
- This new variable would be a more compact representation that has information on both of the previous ones.



Basic idea in PCA: Decorrelation through rotation

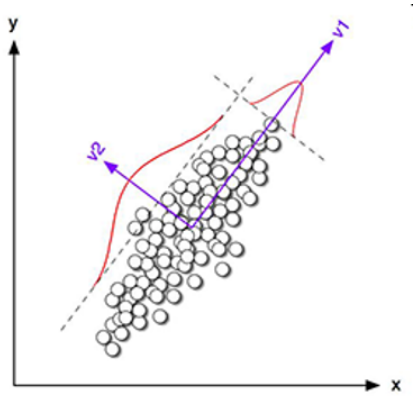
- Using correlation, rotate coordinates so to maximise variation in the 1st component and minimise in the 2nd component.
- Then you can potentially drop the second component now
- This also applies to higher dimensions
- The principle behind is the same as **unitary transforms: rotation in high-dimensional spaces**



Decorrelation through rotation

We will work **around the mean** of the data points

- Thus, we are applying a rotation about the mean of the samples, or mean of the distribution if you like
- This is in essence fitting an ellipse to the data
- Analogy extends to hyperellipsoids in higher dimensions, where visual inspection is not possible



Rotation does not change sum of variances

- Sum of variances does not change with rotations
  $\sum_{i=1}^{d} \sigma_i^2 = \sum_{i=1}^{d} \tilde{\sigma}_i^2$
  with σ_i^2 variance in x_i direction and $\tilde{\sigma}_i^2$ variance in z_i direction
- Stems from invariance of center of gravity and distance under rotation transformation
- Parseval's equation
- **Redistribution of energy to dimensions**
- **We would like as much variance in as few coordinates**

PCA: mathematical treatment for high dimensions

- In high dimensional spaces an optimal rotation is no longer clear upon visual inspection. We need some abstraction
- The required statistics is the **covariance matrix**
  there is an underlying Gaussianity assumption
- Intuition: fit hyperellipsoid to the cluster of data
  subsequent principal components (PCs) correspond to axes from the longest to the shortest

PCA: method

- Suppose x is a vector of d random variables
  if x is an image then d can be the number of pixels
- **First step**: look for a linear combination $c_1^T x$ which has maximum variance – same as fitting a line in $\mathbb{R}^d$
- **Second step**: look for a linear combination $c_2^T x$ with c_1 and c_2 being orthogonal ($c_1^T c_2 = 0$) and with maximum variance
- **Third step**: repeat finding a c_3 that is orthogonal to the previous two directions
- …

Algorithm: Find PCA basis formally

- Assume all x samples are de-meaned $\sum_{n=1}^{N} x_n = 0$

$$c_1 = argmax_{c_1} \sum_{n=1}^{N} c_1^T x_n (c_1^T x_n)^T \; s.t. \; c_1^T c_1 = 1$$

$$= argmax_{c_1} \sum_{n=1}^{N} c_1^T x_n x_n^T c_1$$

$$= argmax_{c_1} c_1^T \sum_{n=1}^{N} x_n x_n^T c_1$$

$$= argmax_{c_1} c_1^T C c_1 \; s.t. \; c_1^T c_1 = 1$$

- Using Lagrange multipliers we maximise
  $c_1 = argmax_{c_1} c_1^T C c_1 - \lambda(c_1^T c_1 - 1)$
- Differentiation and setting to 0 yields
  $Cc_1 - \lambda c_1 = 0 \rightarrow (c - \lambda I)c_1 = 0$
- Thus λ must be an eigenvalue of C and c_1 the corresponding eigenvector!
- Which of the d eigenvectors?
  $c_1^T C c_1 = c_1^T \lambda c_1 = \lambda c_1^T c_1 = \lambda$
  Since λ should be as large as possible then it is the largest eigenvalue and c_1 is the corresponding eigenvector
- λ corresponds to the variance in the c_1 direction

How about the 2nd PC?

Maximise $c_2^T C c_2$ while being orthogonal to c_1

Let's Lagrange multipliers again

$argmax_{c_2} c_2^T C c_2 - \lambda(c_2^T c_2 - 1) - \eta c_2^T c_1$

Differentiation gives

$Cc_2 - \lambda c_2 - \eta c_1 = 0$

$c_1^T(Cc_2 - \lambda c_2 - \eta c_1) = 0 \Rightarrow \eta = 0$

$Cc_2 - \lambda c_2 = 0$

So c_2 should be an eigenvector and correspond to the 2nd largest eigenvalue based on the same argument for c_1.

How about the kth PC?

- The kth PC is the eigenvector corresponding to the □kth largest eigenvalue

Are the new variables decorrelated?

- In the 2D case, through rotation we could see decorrelation
- In higher dimensions does this hold?

- Covariance is estimated through
  $\sum_{n=1}^{N} z_i z_j^T = \sum_{n=1}^{N} c_i^T x_n (c_j^T x_n)^T$
  $c_i^T C c_j = c_i^T \lambda_j c_j = \lambda_i c_i^T c_j = 0$
- Hence covariance and therefore correlation is 0 between the variables. They are decorrelated

PCA: interpretation

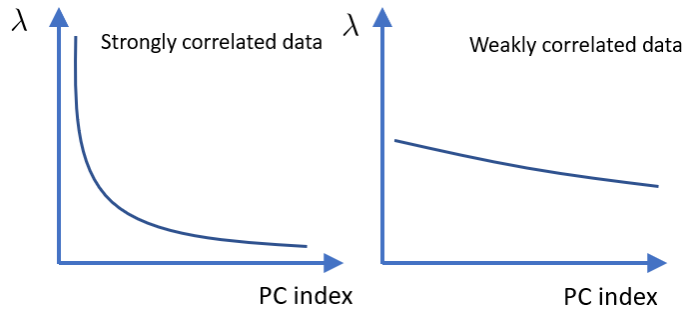
- C is the sample covariance matrix
- If number of independent samples is higher than number of dimensions, i.e. $N > d$, then C will be full rank positive semi-definite matrix and admit d real non-negative eigenvalues and corresponding eigenvectors, i.e. PCs
- We would use this notation to represent it all
  $C = U \Lambda U^T$, where U is matrix of eigenvectors, and Λ is diagonal matrix of eigenvalues
- It is still a rotation in high-dimension because $U^T U = I$

Some notes on PCA

- PCA collects maximum variance in subsequent uncorrelated components
- Any sample can be represented using PCs and corresponding variables, and vice-versa
  $z = U^T x$ and $x = Uz$
- If number of samples is less than number of dimensions Singular Value Decomposition can be used to identify principal components

Representation with fewer PCs

- Strongly correlated data will yield first PCs to contain most of the variance in the data. Representing the data only with the first PCs will cause minimal information loss



One of the first examples in computer vision

Journal of Cognitive Neuroscience 1991

- Very primitive retrospectively but has all the ingredients
- Set of images – training set
  - They used such images to compute the Principal Components
  - These images are called training images

    - The entire set is called the training set.
    - They only used 16 images
    - 128x128
    - SVD-based computation to get the PCs
- Reconstruction works quite well with only 7 PCs
  $x \in \mathbb{R}^{128\times128}$ while $z \in \mathbb{R}^7$
- Works well only for relevant face images
- Correcting partial occlusions
  Through projecting the image onto the principal components and reconstructing back
  First, $z = U^T x$ then $\hat{x} = Uz$