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.

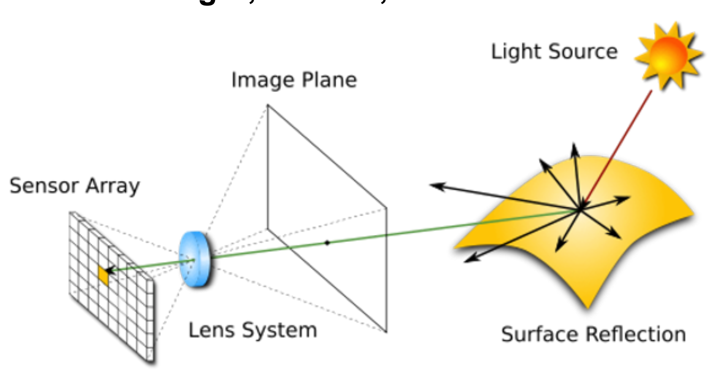
applications

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

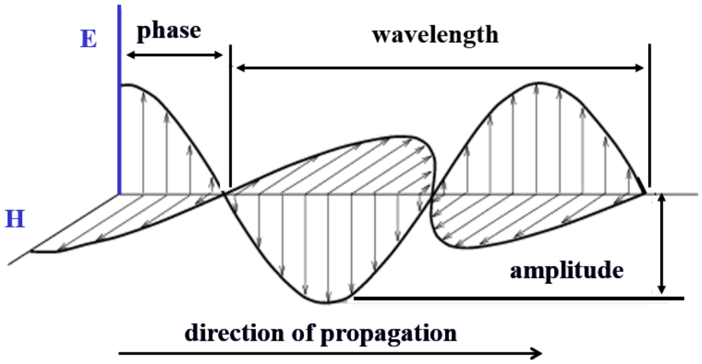


Levels of optical analysis

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

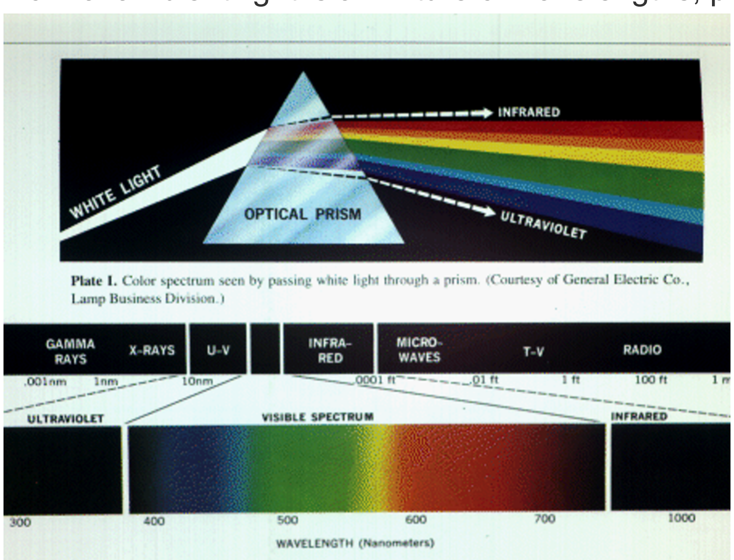
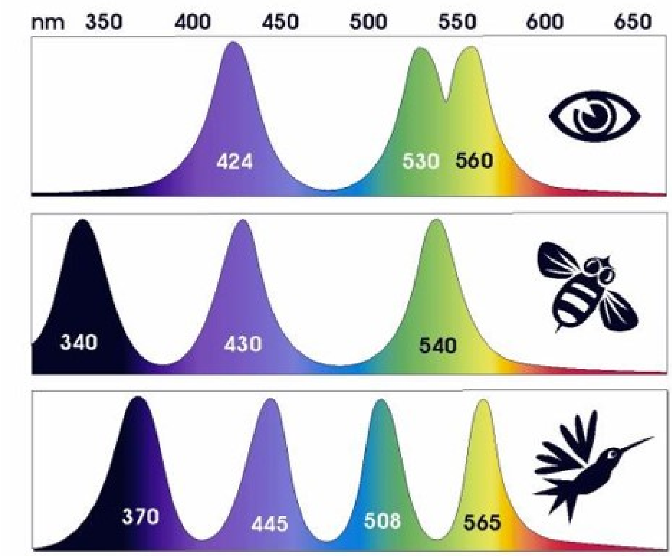


Plate I. Color spectrum seen by passing white light through a prism. (Courtesy of General Electric Co., Lamp Business Division.)

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

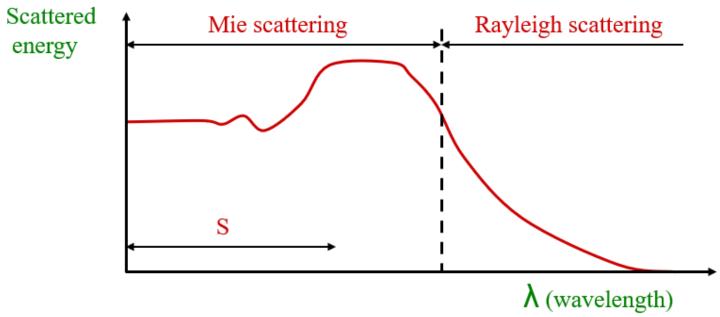
+ diffraction

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

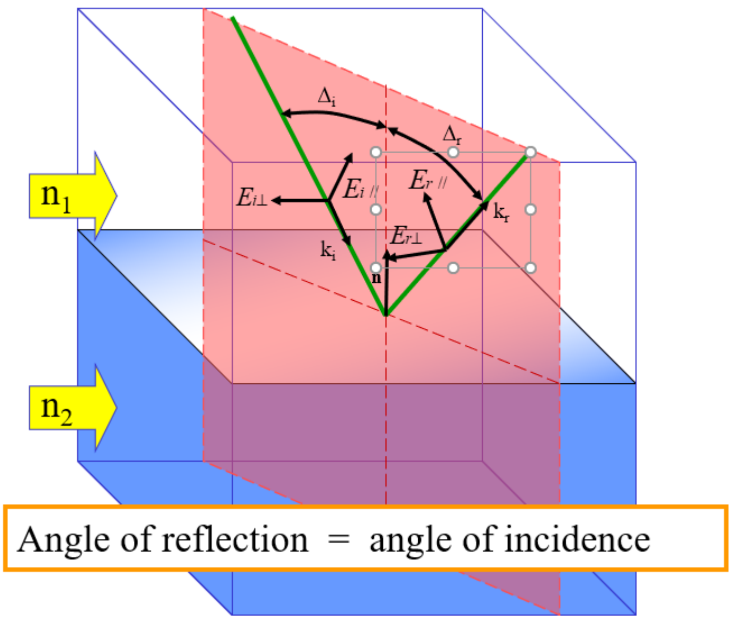
Rayleigh: Tyndall effect (blue sky) Red, setting sun

Non-selective: Grey clouds

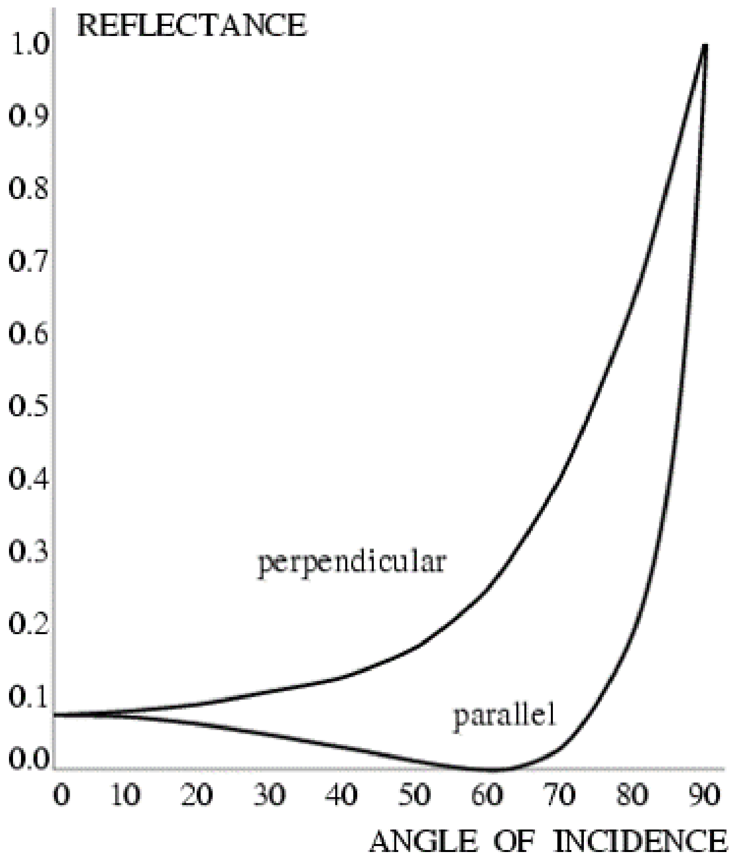
Mie: Coloured cloud from volcanic eruption

Reflection

Mirror reflection



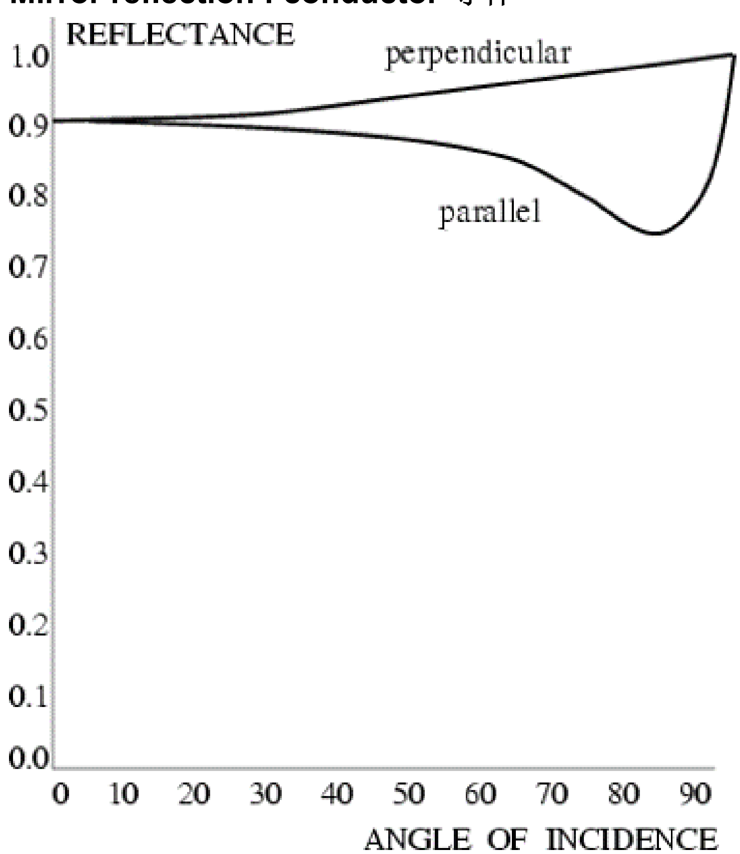
Mirror reflection : dielectric 电介质



Polarizer at **Brewster angle**

Full reflection at grazing angles

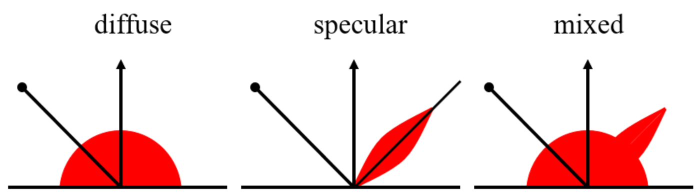
Mirror reflection : conductor 导体



- 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

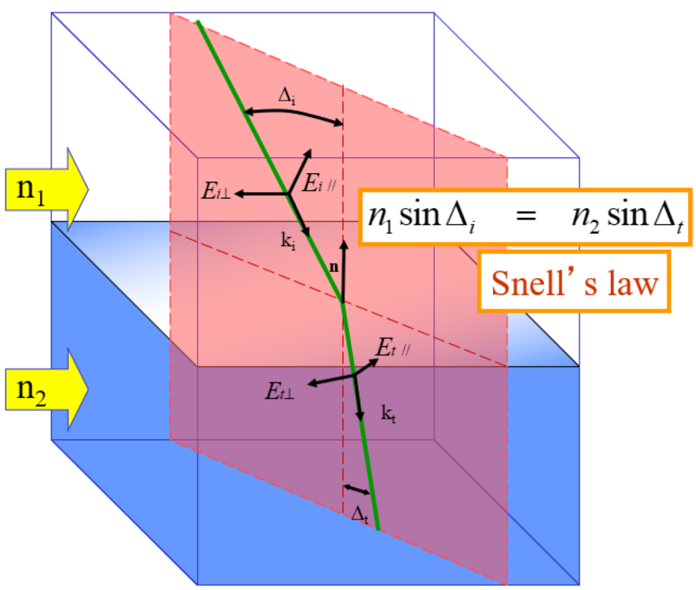
- three types of reflection :



- 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

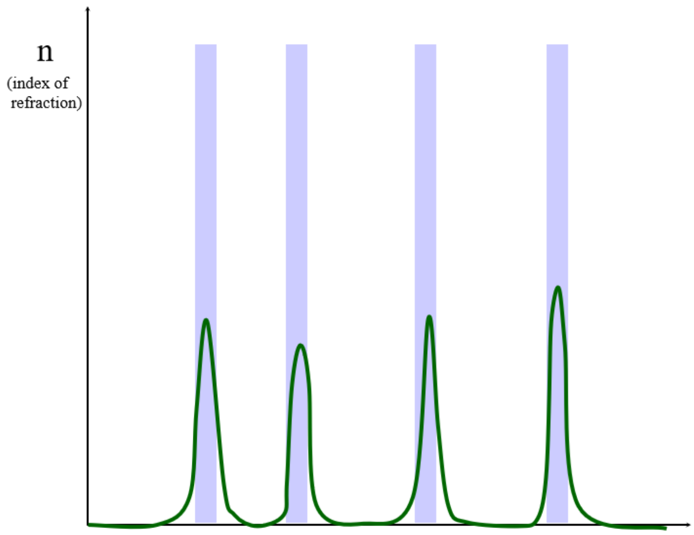
Refraction 折射



Dispersion 色散

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

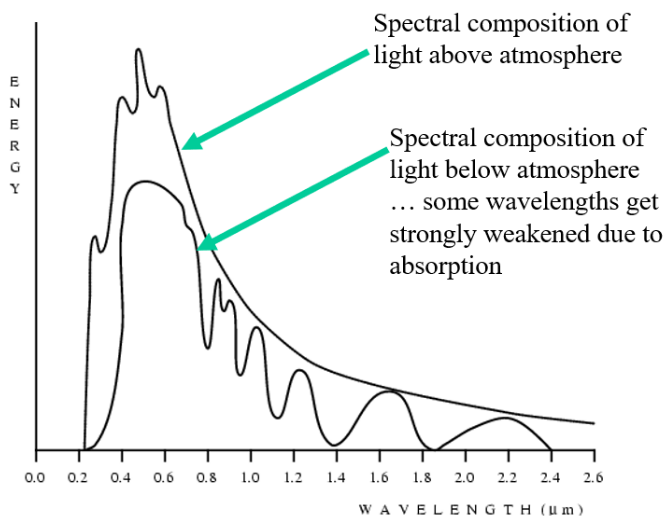
Absorption
Dissipation of wavelengths specific for the medium



- 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

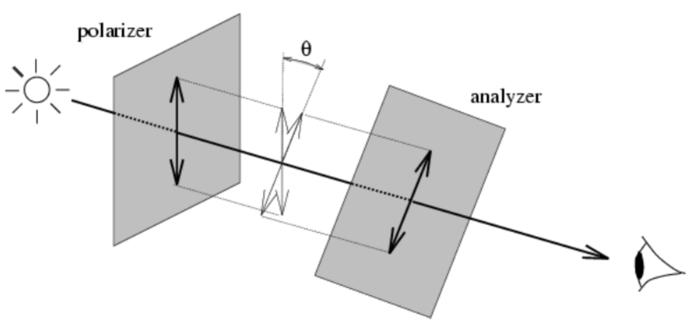
- generate sharp shadows
- generation of specular reflection (e.g. crack detection)
- shadows and shading yield information about shape

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

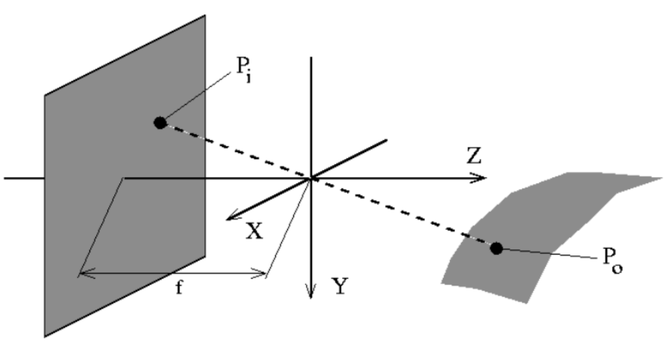
- 'Dark' and 'bright' field
  In the 'dark' field, the camera is placed out of the area of specular reflection for the normal surface, and only abnormally oriented parts of the surface will lighten up (showing specular reflection) – flaws
  In the 'bright' field, the camera is placed so to capture the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

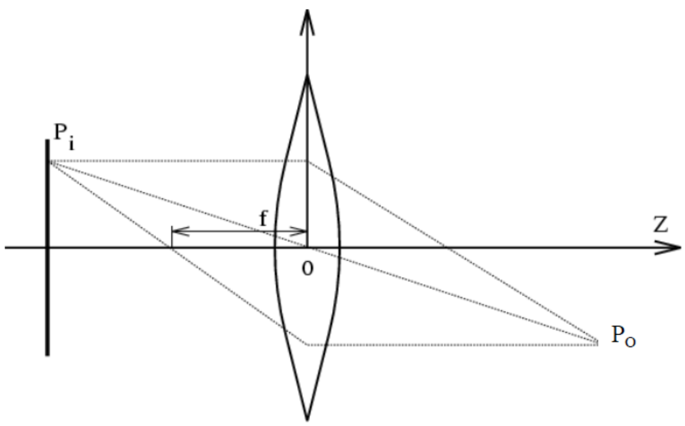
- the pinhole model:



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

The thin-lens equation

- lens to capture enough light:

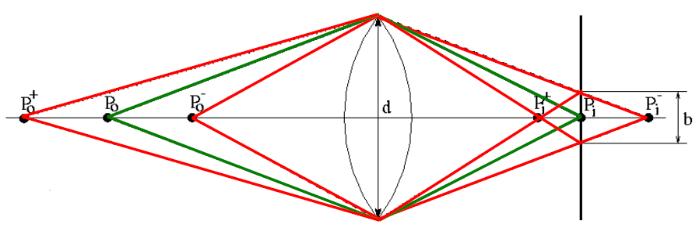


$$\frac{1}{Z_o} - \frac{1}{Z_i} = \frac{1}{f}$$

- 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

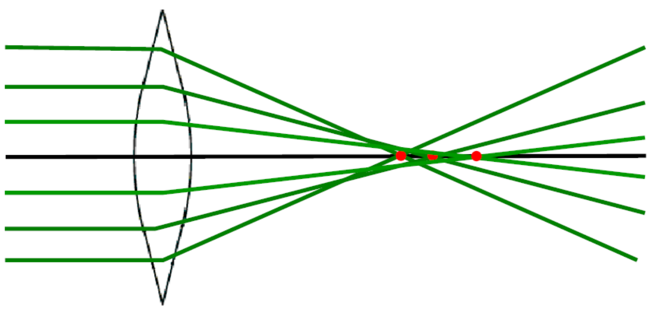
Deviations from the lens model

3 assumptions:

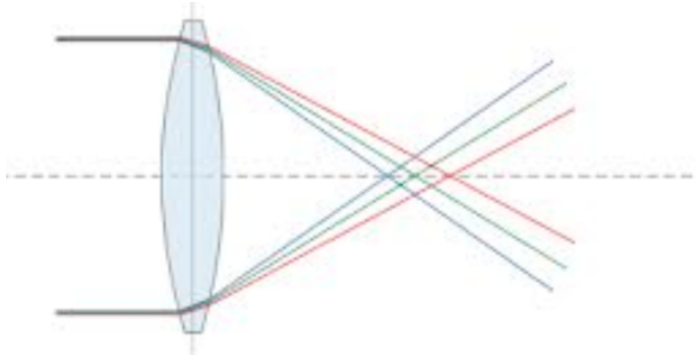
1. all rays from a point are focused onto 1 image point
2. all image points in a single plane
3. magnification is constant
   deviations from this ideal are **aberrations 像差**

Aberrations

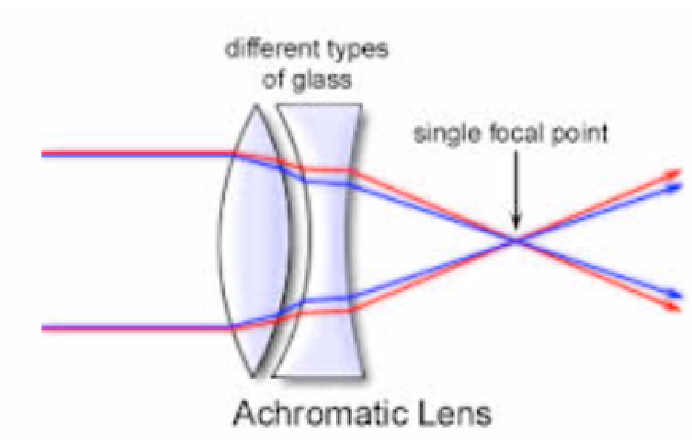
2 types:

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

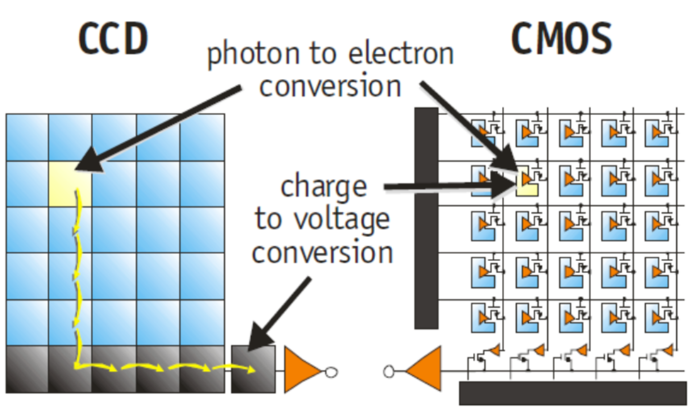


- cannot be removed completely but **achromatization** can be achieved at some well chosen wavelength pair, by combining lenses made of different glasses



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

Cameras



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

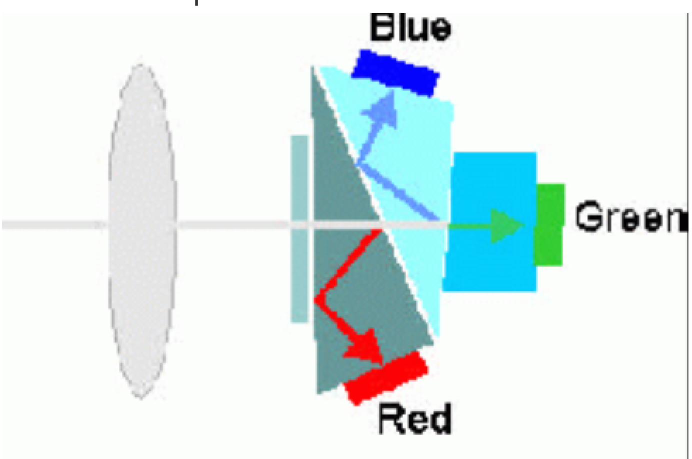
- Same sensor elements as CCD
- Each photo sensor has its own amplifier
  - More noise (reduced by subtracting 'black' image)
  - Lower sensitivity (lower fill rate)

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

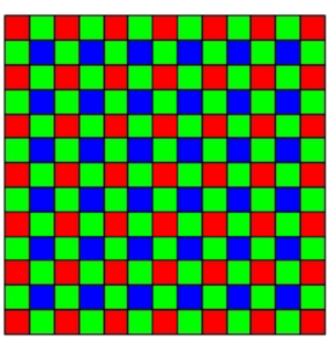
CCD vs. CMOS

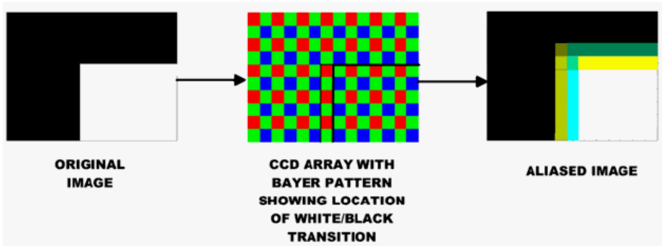
- CCD: Niche applications, Specific technology, High production cost, High power consumption, Higher fill rate, Blooming, Sequential readout
- CMOS: Consumer cameras, Standard IC technology, Cheap, Low power, Less sensitive, Per pixel amplification, Random pixel access, Smart pixels, On chip integration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

Colour cameras

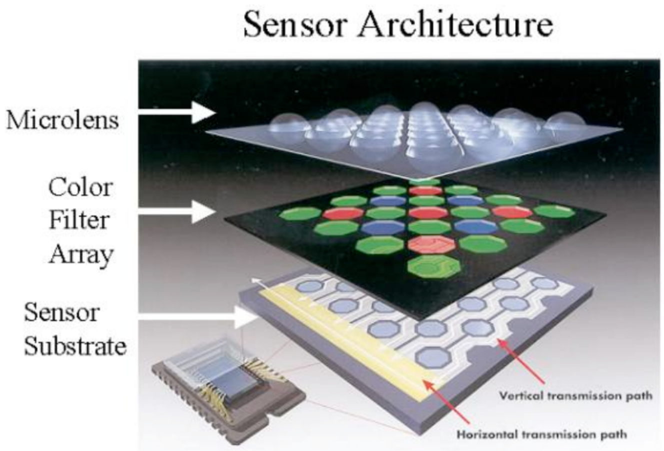
1. Prism (with 3 sensors)
   Separate light in 3 beams using dichroic prism
   Requires 3 sensors & precise alignment
   Good color separation
   

2. Filter mosaic

- Coat filter directly on sensor
  
  Bayer filter
- Demosaicing (obtain full colour & full resolution image)

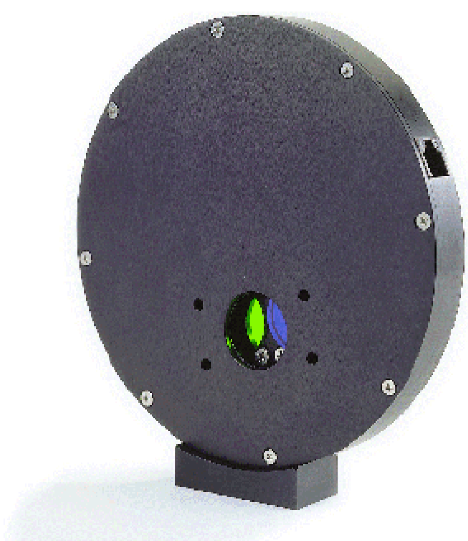


- Color filters lower the effective resolution, hence **microlenses** often added to gain more light on the small pixels



3. Filter wheel

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



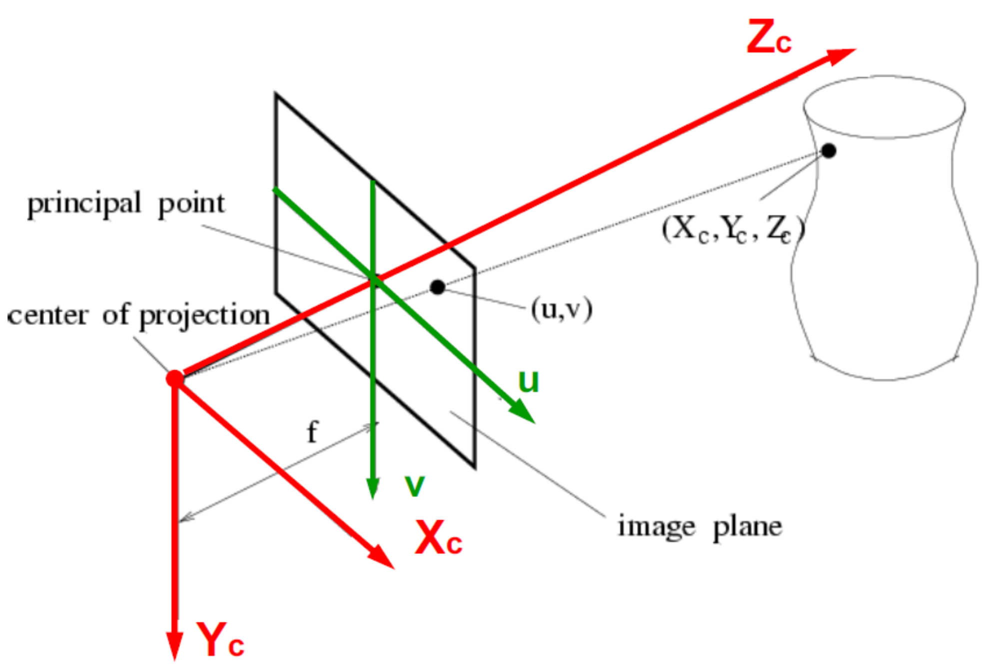
Prism vs. mosaic vs. wheel

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

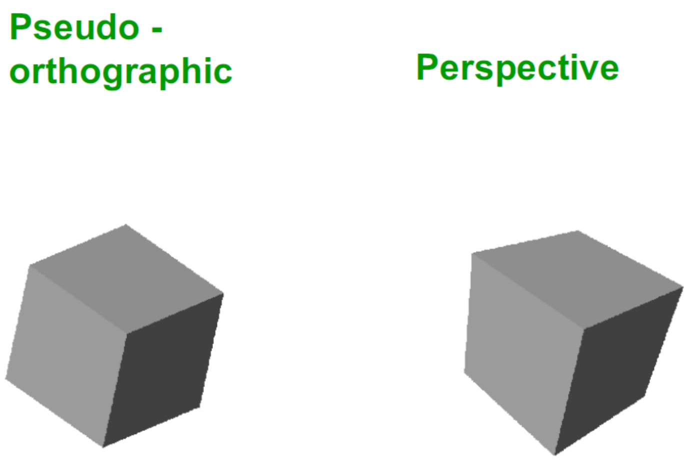


- origin lies at the center of projection
- the Z_c axis coincides with the optical axis
- X_c-axis $||$ to image rows, Y_c-axis $||$ to columns
- $u = f\frac{X}{Z}, v = f\frac{Y}{Z}$

Pseudo-orthographic projection

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

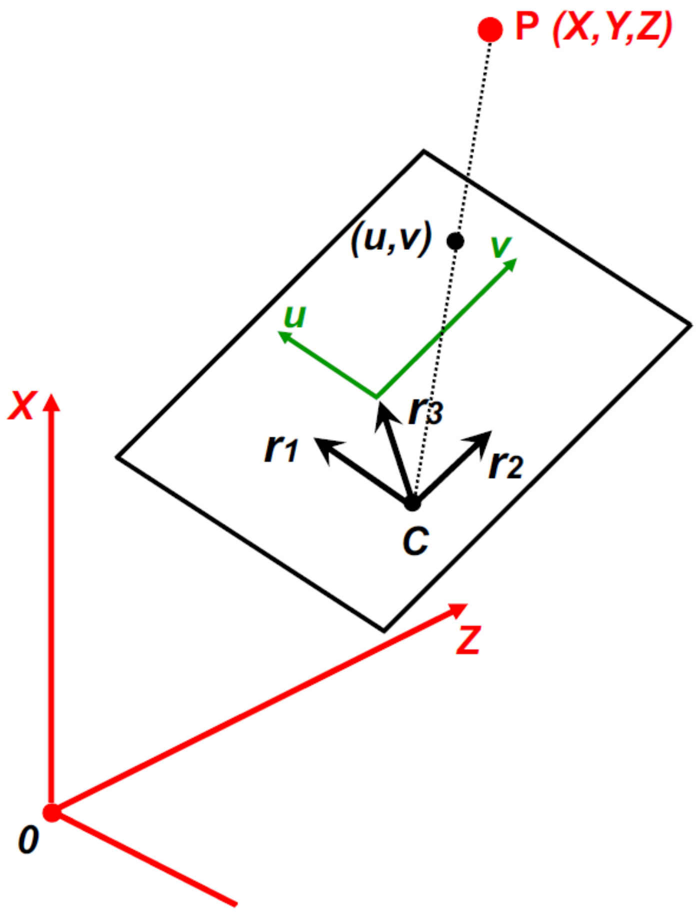
- Good approximation if $f/Z \pm constant$, i.e. if objects are **small compared to their distance** from the camera
- Pictoral comparison



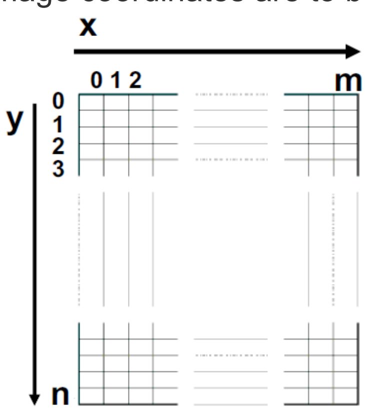
Projection matrices

- 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 $p = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$; $\quad P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$; $\quad \tilde{P} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$
  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

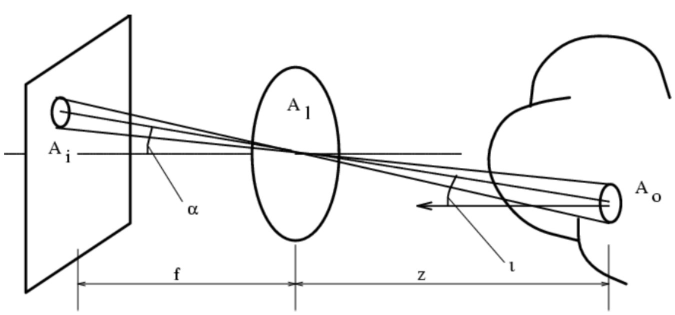
1. from object radiance to image irradiance
2. from image irradiance to pixel grey level

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

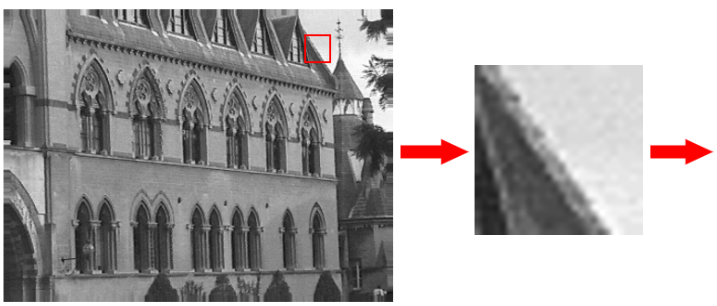
Sampling, Quantization and Image Enhancement

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”



84	133	226	212	218	218	222	212	218	222	226	218
75	156	177	218	212	218	218	218	218	222	218	218
96	84	133	203	218	218	218	222	212	218	222	218
123	75	111	156	212	218	212	212	218	218	218	226
93	75	71	133	185	231	226	226	222	212	218	218
51	75	75	75	156	206	218	218	218	222	212	222
44	110	75	65	143	194	231	218	218	218	218	218
52	123	69	84	60	156	199	231	231	222	226	226
52	75	84	81	65	69	150	231	231	226	231	231
36	36	84	93	84	71	156	160	240	240	231	231
36	40	113	75	69	75	71	133	194	240	240	240
52	52	105	85	69	75	75	123	111	222	231	231
69	44	69	93	81	75	75	69	150	177	247	240
73	44	40	96	101	75	75	75	84	133	231	240

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

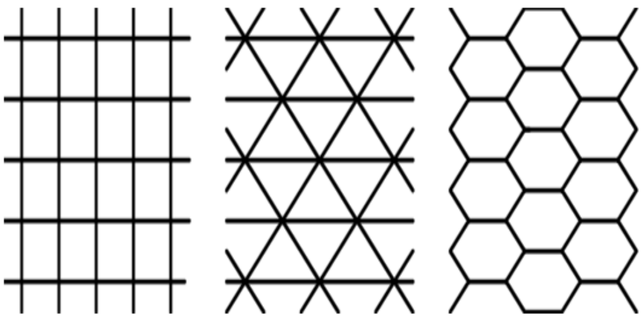
1. Binary images – 1-bit quantization – are useful in industrial applications. They usually have control over imaging conditions
   e.g. background color, lighting conditions,...
2. Non-uniform sampling and/or quantization is sometimes used for specialized applications

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

3. Different sampling strategies than square grids exist

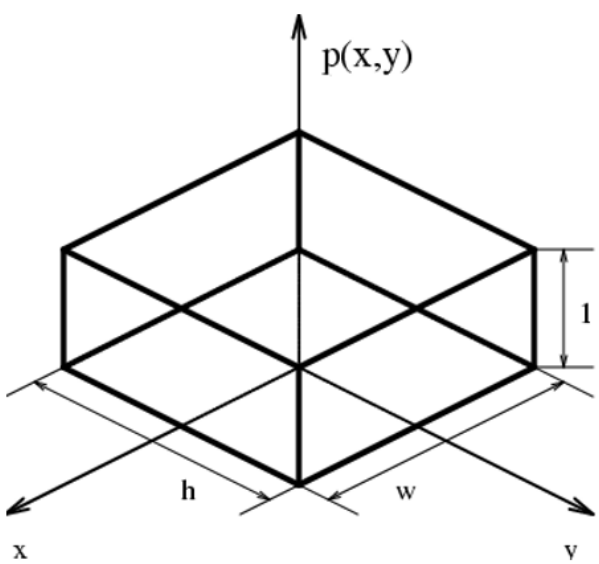
Different sampling schemes

- 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
  $$\begin{aligned} k &= h * f \\ &= (h_1 * h_2) * f \\ &= h_1 * (h_2 * f) \end{aligned}$$

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

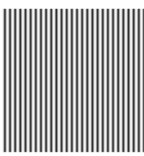
- The Inverse Fourier Transform
  $\mathcal{F}^{-1}[F(u,v)] = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{i2\pi(ux+vy)}dudv$ (modified)

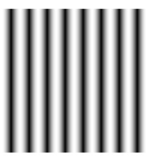
Fourier Coefficients

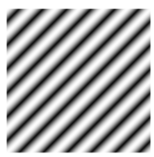
- Complex function
  $F(u,v) = \underbrace{F_R(u,v)}_{\text{real part}} + \underbrace{iF_I(u,v)}_{\text{imaginary part}}$
- Magnitude
  $|F(u,v)| = \sqrt{F_R(u,v)^2 + F_I(u,v)^2}$
- Phase-angle
  $\phi(u,v(= arg(F_R(u,v) + iF_I(u,v)) = arctan\frac{F_I(u,v)}{F_R(u,v)}$

Decomposition visually

$f(x,y) =$ 

$= F(u,v)$ $\times$ 

$+ F(u',v')$ $\times$ 

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

Example of FT

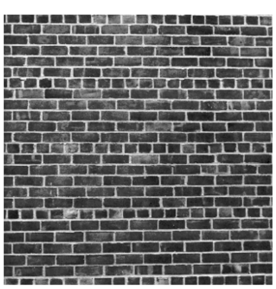
center, where u=0,v=0, value is the largest, and in the corners where u and v are large, the magnitude of Fourier Coefficients are small
so that fewer number of very high frequency components are needed to represent this image.

Effect of additional components

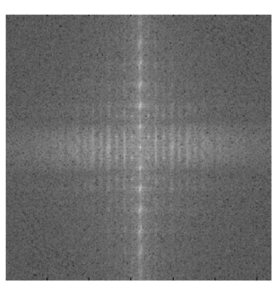
higher frequency$\rightarrow$ more details

Importance of the magnitude in FT

- Image with periodic structure



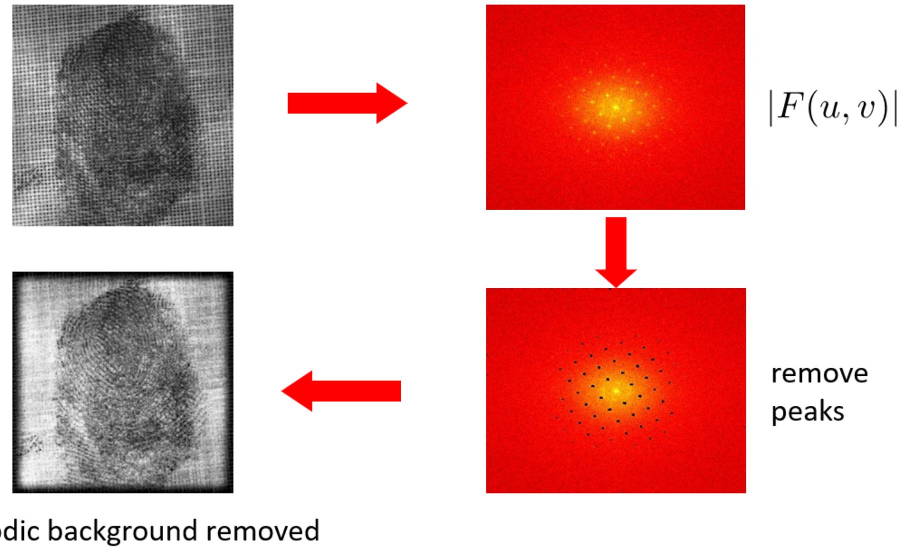
$f(x,y)$



$|F(u,v)|$

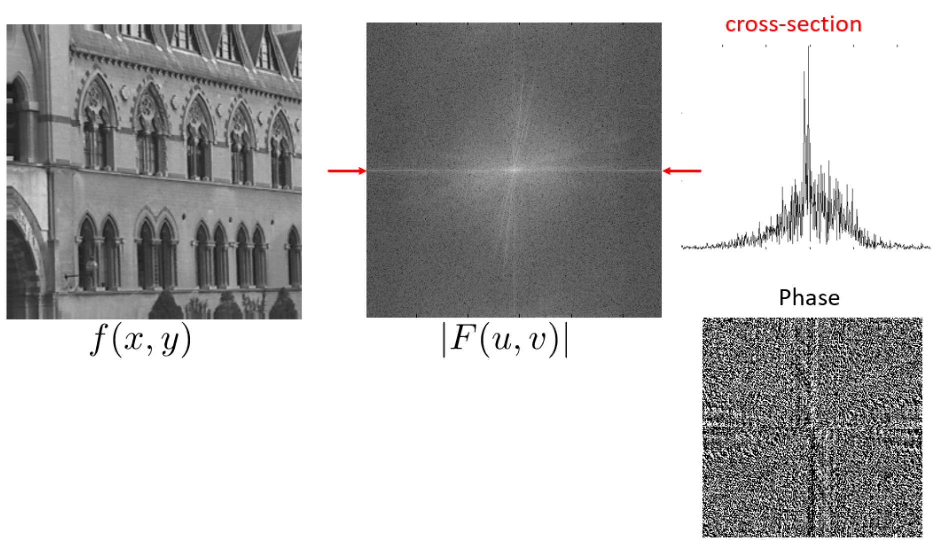
- **FT has peaks at spatial frequencies of repeated texture**

Importance of the magnitude in FT

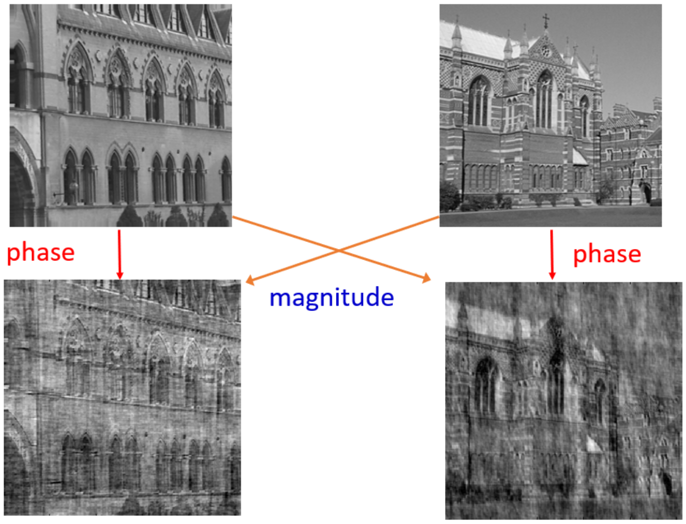


General structure of the magnitude

- 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

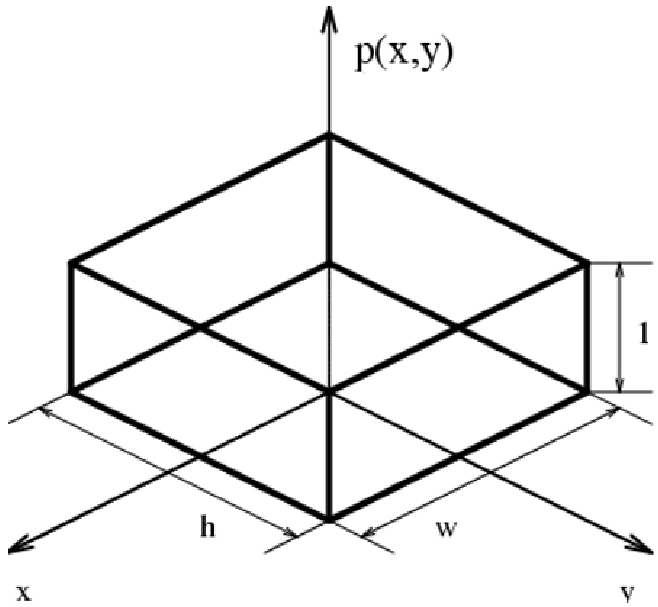
- $C(u,v) = A(u,v)B(u,v) \Leftrightarrow c(x,y) = a(x,y) * b(x,y)$
  $C(u,v) = A(u,v) * B(u,v) \Leftrightarrow c(x,y) = a(x,y)b(x,y)$
- **Space multiplication = frequency convolution**

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



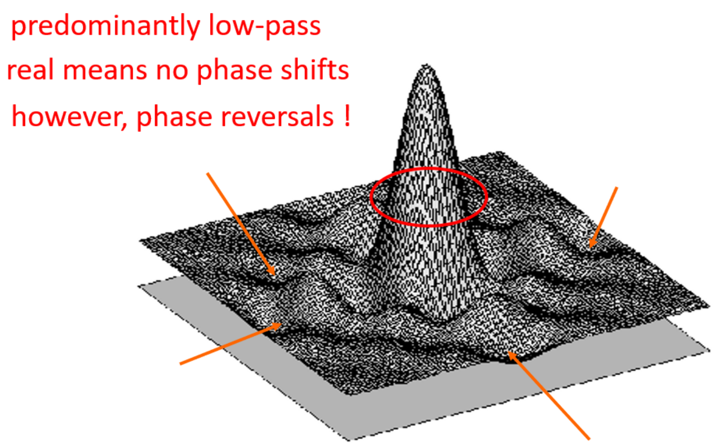
$o(x',y') = \int\int i(x,y)p(x'-x,y'-y)dxdy$ (modified)

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

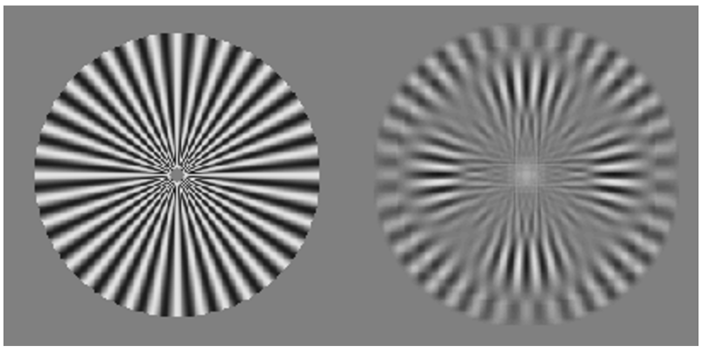
Modulation transfer function of the window function

- 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



- Illustration of the effect of 2D sinc

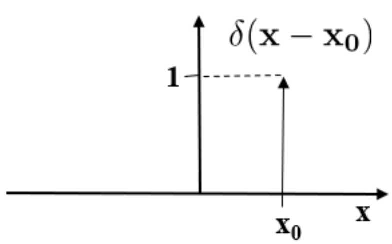


Summary for STEP I

- 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
  $\delta(\mathbf{x}-\mathbf{x}_0)=0 \quad \mathbf{x}\neq\mathbf{x}_0$
  $\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\delta(\mathbf{x}-\mathbf{x}_0)d\mathbf{x}=1$

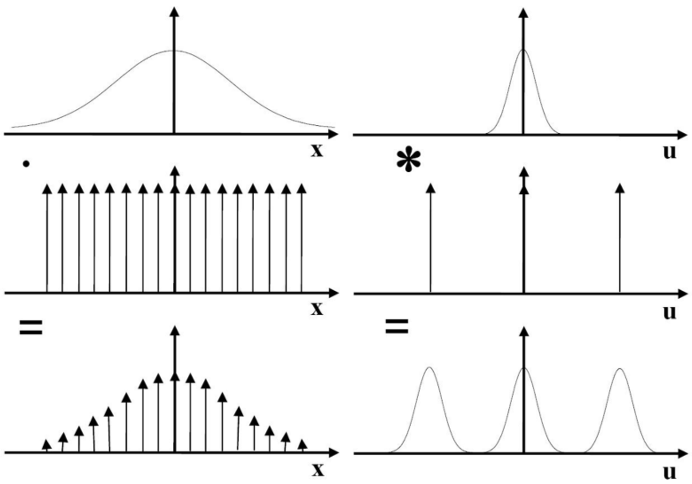


- Function probing (in 1D)
  $\int_{-\infty}^{\infty}\delta(x)f(x)dx=f(0)$
  $\int_{-\infty}^{\infty}\delta(x-x_0)f(x)dx=f(x_0)$

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.

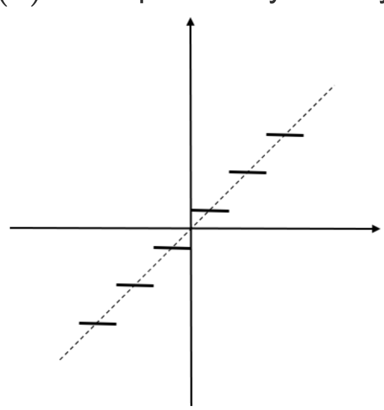
Quantization

- 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



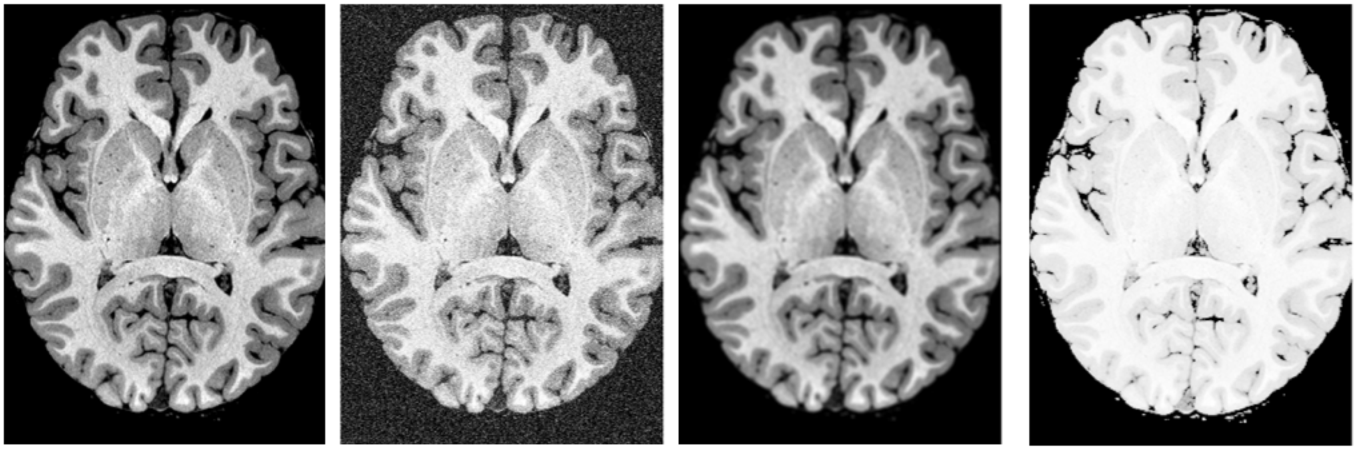
Underquantization examples(different gray level)
Small remarks on quantization

- 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



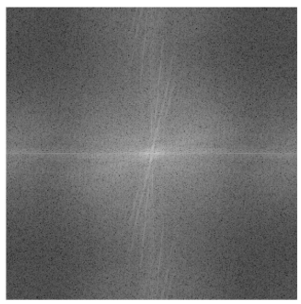
Original Image Noise Blur Bad Contrast

More on Fourier transform

Signal and noise

Fourier power spectra of images

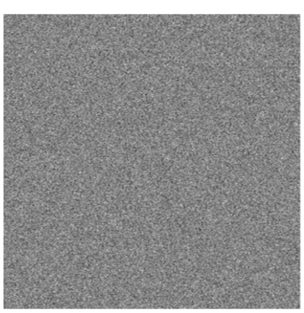
$i(x,y)$



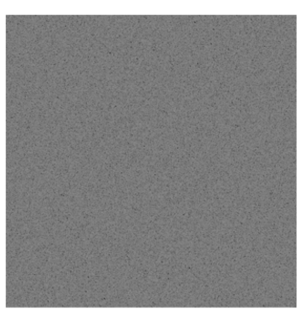
$\phi_{ii} = |I(u,v)|^2$

- 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

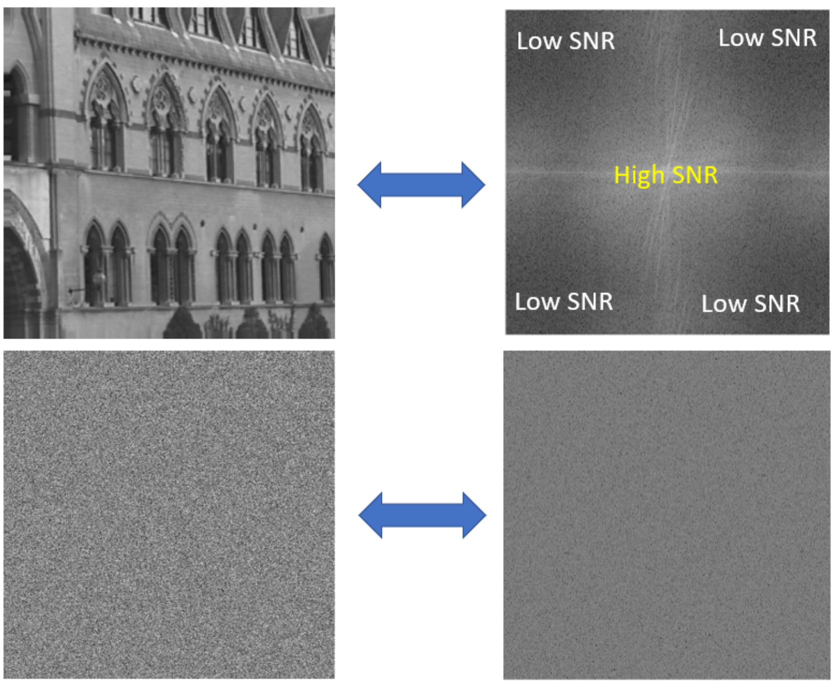
- Similar components in high and low frequencies.

Fourier power spectra of noisy image

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

- Power spectra is a combination of image and noise

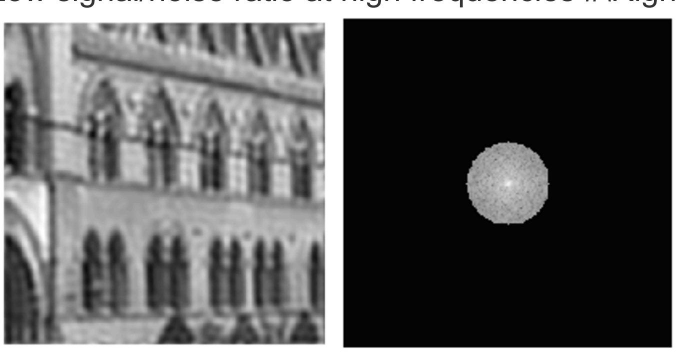
Signal to noise ratio (SNR)



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

Only retaining the low frequencies

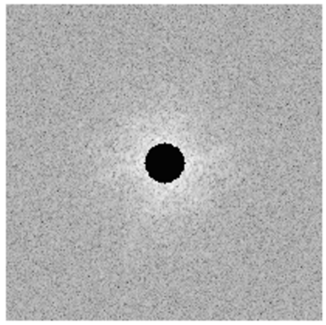
- Low signal/noise ratio at high frequencies #\Rightarrow$ eliminate these



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

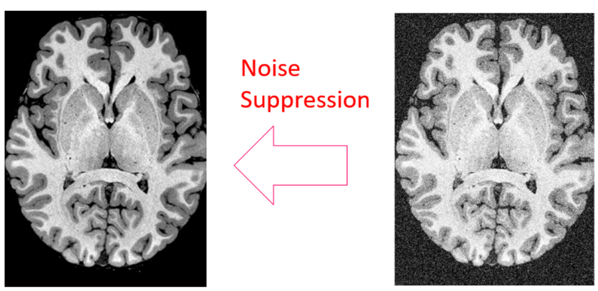
High frequencies contains noise and edge information

- 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

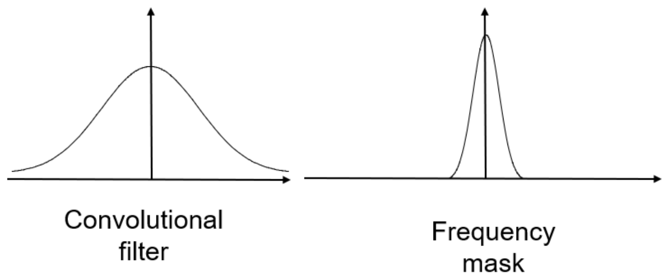


Original Image

Noisy Observation

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

$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \qquad \frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

Separable: $\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} * \frac{1}{3}\begin{bmatrix}1&1&1\end{bmatrix}$

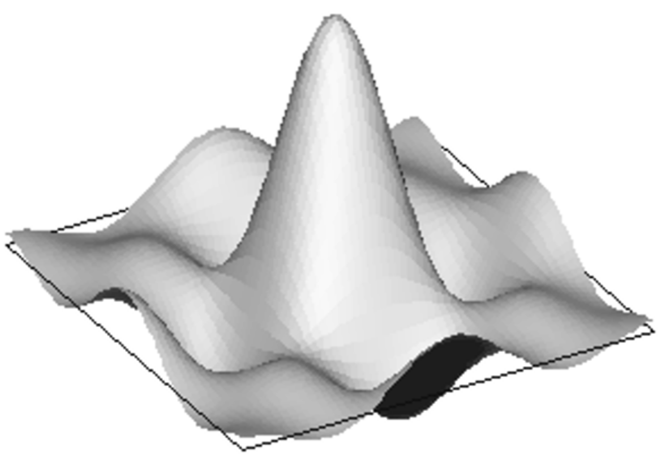
- $o(x, y) = f(x, y) * i(x, y) = f_1(x, y) * (f_2(x, y) * i(x, y))$

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!

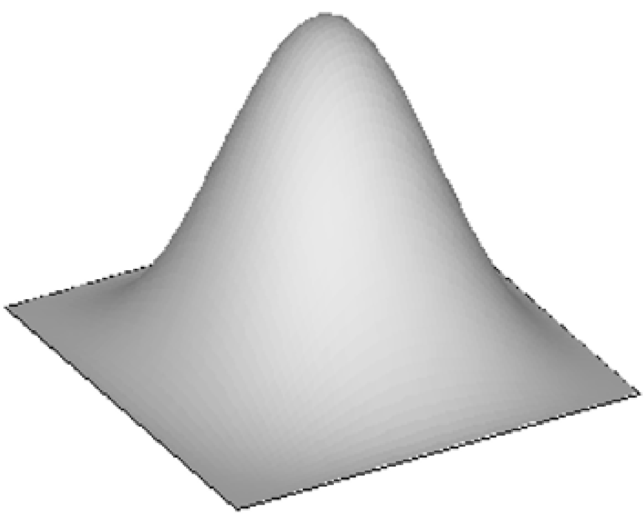
Solution: Binomial filtering

- 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

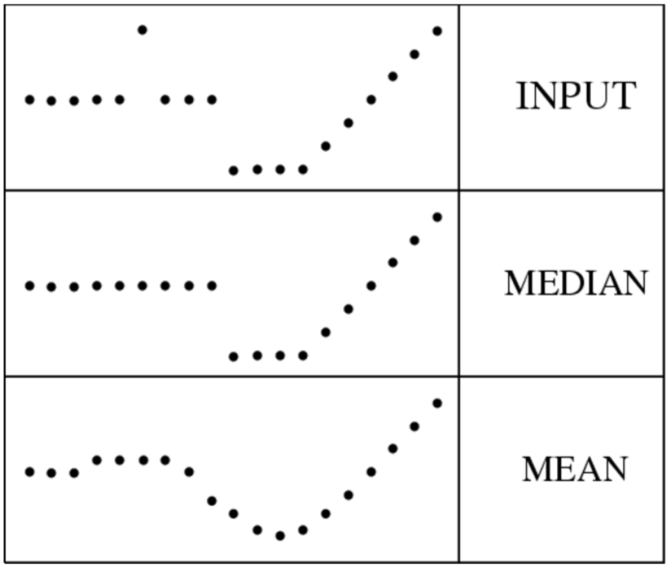
- Non-linear filter
- Simple method:
  - Rank-order neighborhood intensities in a patch of the image
  - Take middle value and assign it to the patch center
  - Go over all the image in a sliding window
- No new grey levels will emerge.

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

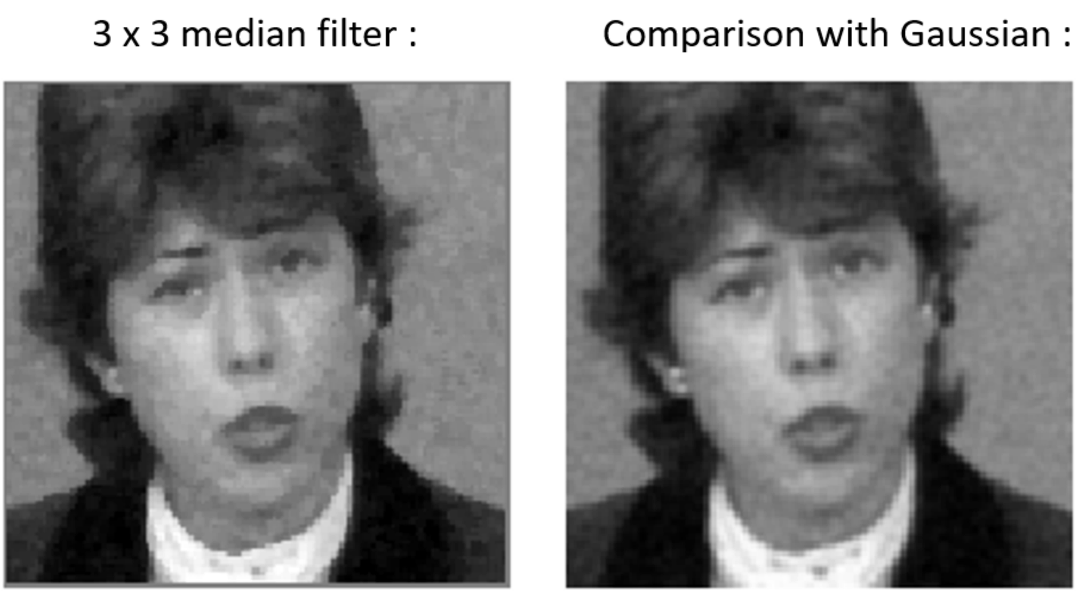


Median filtering – is it the solution to our blurring problem?

- 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



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!

$f(\overrightarrow{x}, t) = f(\overrightarrow{x}, 0) * g(\overrightarrow{x}, t) = i(\overrightarrow{x}) * g(\overrightarrow{x}, t)$

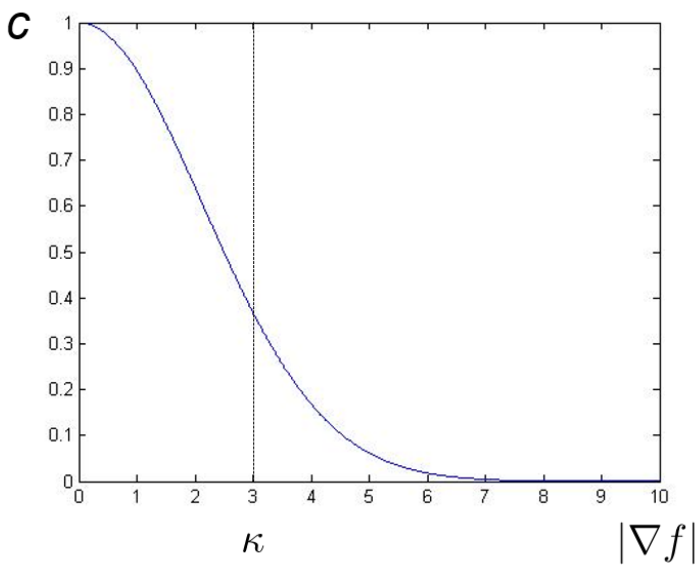
Diffusion as Gaussian low-pass filter

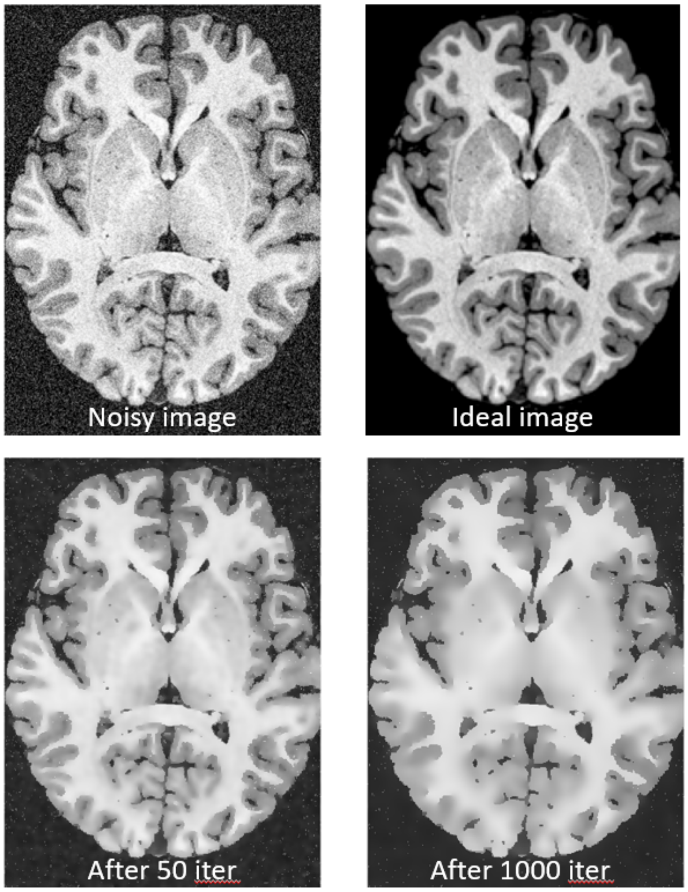
- $f(\overrightarrow{x}, t) = i(\overrightarrow{x}) * \frac{1}{(2\pi)^{d/2}\sqrt{ct}} exp(-\frac{\overrightarrow{x} \cdot \overrightarrow{x}}{4ct})$
- Gaussian filter with time dependent standard deviation: $\sqrt{2ct}$
- Nonlinear version can change the width of the filter locally
  $c(\overrightarrow{x},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

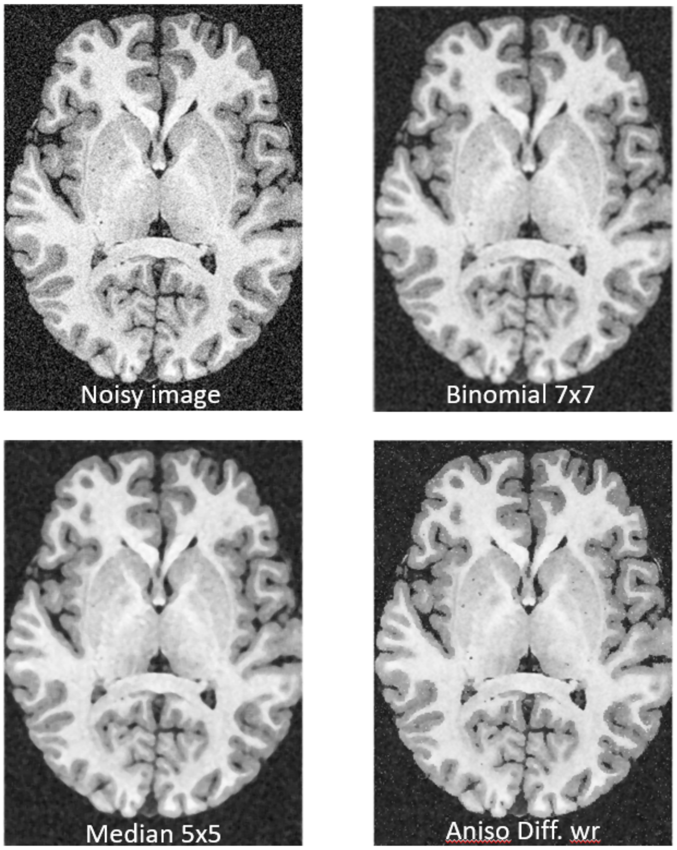




Unrestrained anisotropic diffusion

- End state is homogeneous

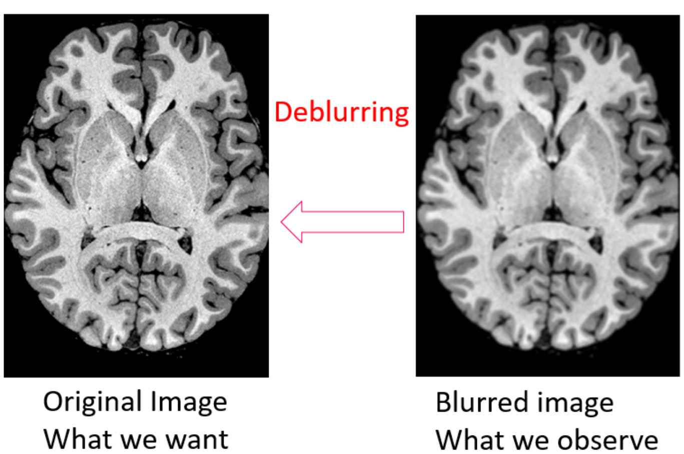
- adding restraining force:
  $\frac{\partial f}{\partial t} = \Delta \cdot (c(|\nabla f|)\nabla f) - \frac{1}{\sigma^2}(f - i)$



Anisotropic diffusion – numerical solutions

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

Deblurring

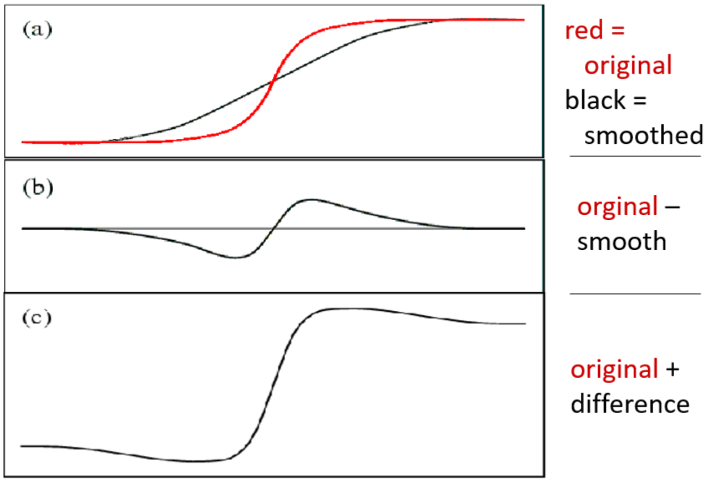


Approach I: Unsharp masking

- simple but effective method

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

Unsharp masking - sketch

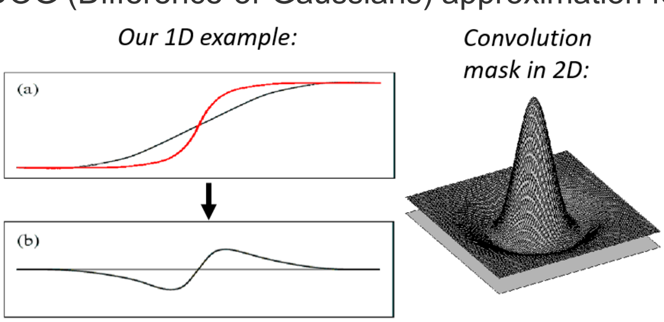


Unsharp masking - principle

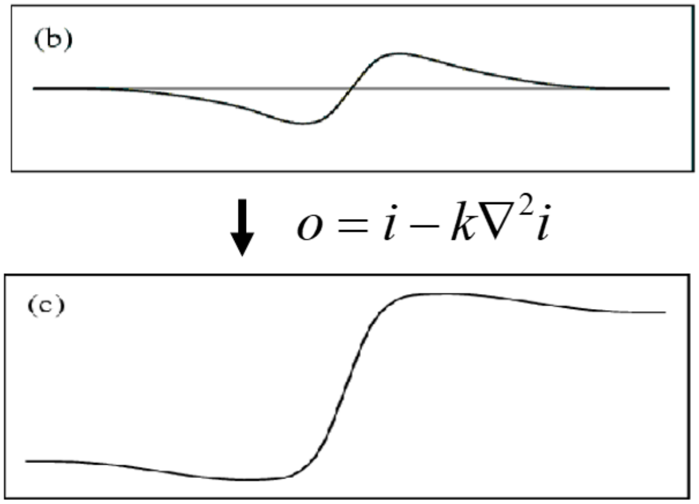
- Interpret blurred image as snapshot of diffusion process
  $\frac{\partial f}{\partial t} = c(\nabla^2 f)$
  In a first order approximation, we can write
  $f(x,y,t) \approx f(x,y,0) + \frac{\partial f}{\partial t} t$
  Hence,
  $f(x,y,0) \approx f(x,y,t) - \frac{\partial f}{\partial t} t = f(x,y,t) - ct\nabla^2 f$
  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 :



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

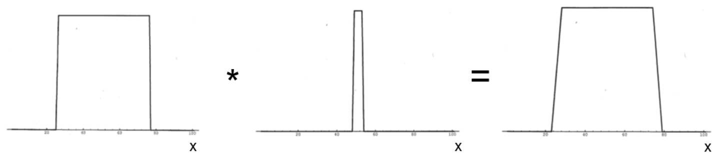
- Relies on system view of image processing
- Frequency domain technique
- Defined through Modulation Transfer Function
- Links to theoretically optimal approaches

Inverse filtering principle

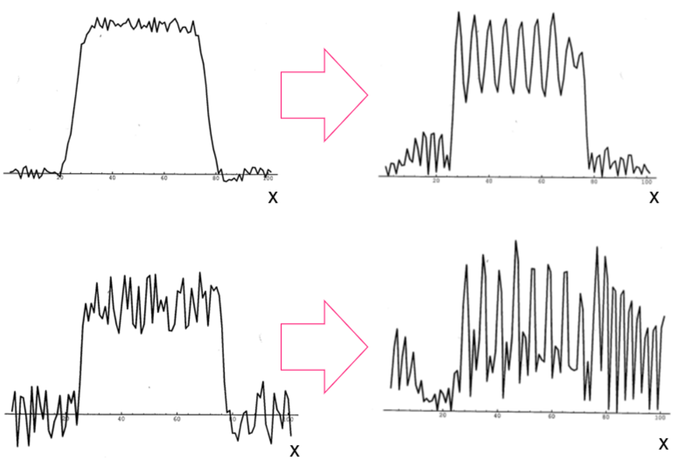
- Frequency domain technique
- suppose you know the MTF $B(u, v)$ of the blurring filter
  $f(x, y) = b(x, y) * i(x, y)$
  $F(u, v) = B(u, v)I(u, v)$
- to undo its effect new filter with MTF $B'(u, v)$ such that
  $B'(u, v)B(u, v) = 1$
  $I(u, v) = B'(u, v)F(u, v)$
  For additive noise after filtering
  $F(u, v) = B(u, v)I(u, v) + N(u, v)$
  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

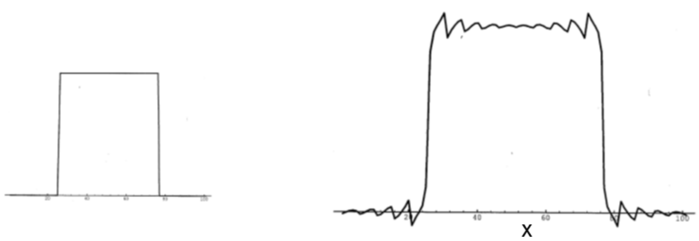
- Looking for the optimal filter to do the deblurring

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

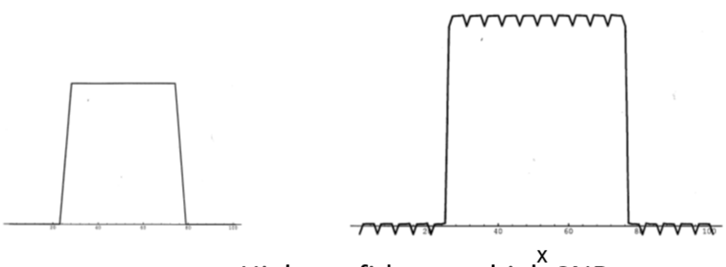
Wiener filter and its behavior

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

Deblurring noise-free signal

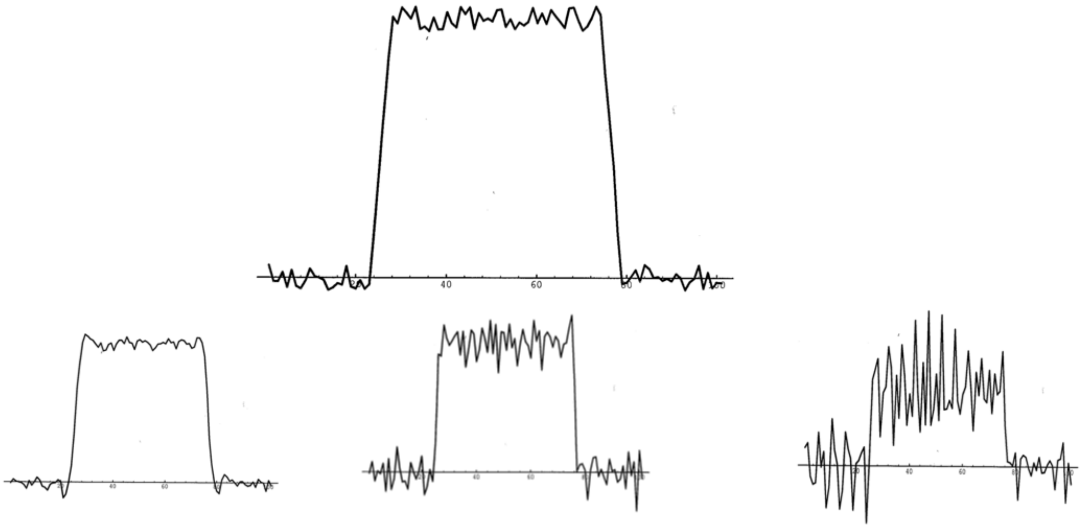


Medium confidence – medium SNR assumption



High confidence – high SNR assumption

Deblurring noisy signal



Low confidence

Medium confidence
Correct SNR

High confidence

Wiener filtering example

- spurious high freq. eliminated, conservative

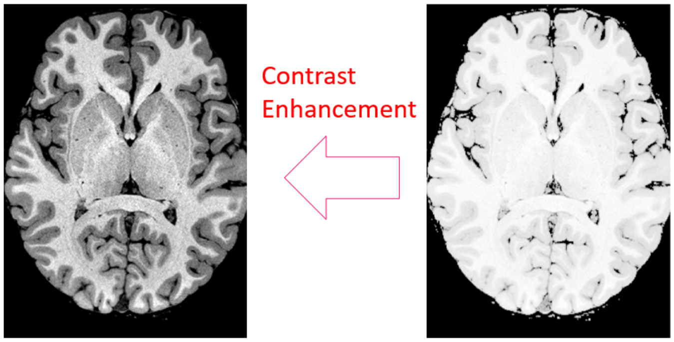
Problems in applying Wiener filtering

- $O(u,v) = Wf(H)(H(u,v)I(u,v)) = (Wf(H)H(u,v))I(u,v)$
  $Ef = Wf(H)$, 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

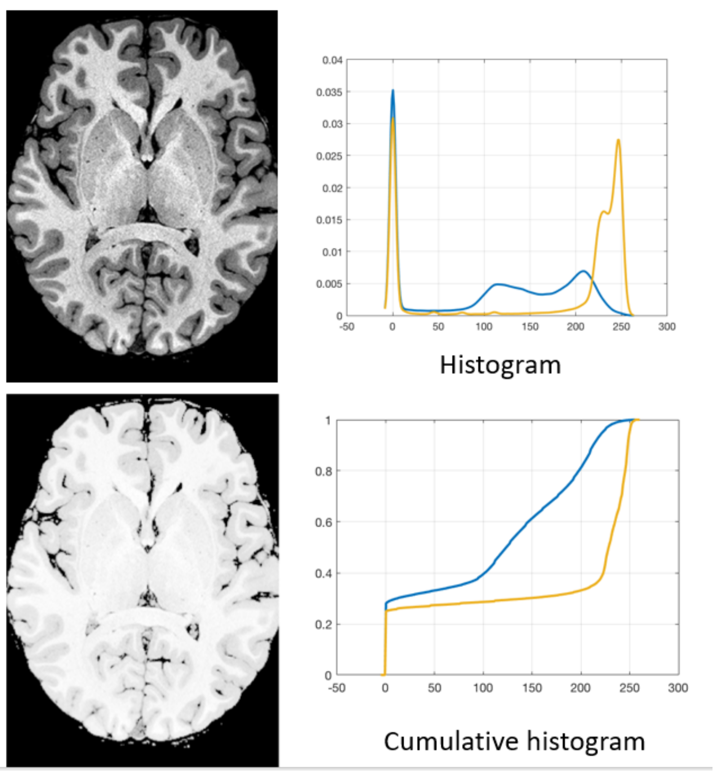


Original Image

Observation with Bad Contrast

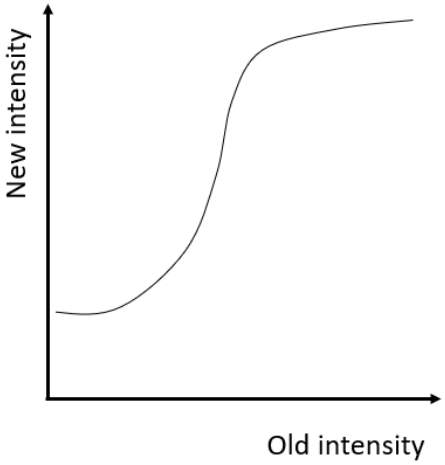
We will study histogram equalization

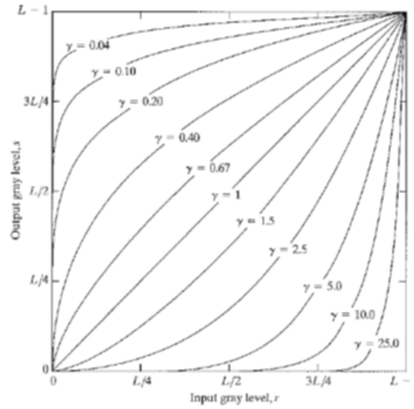
Intensity distributions - histogram



Intensity mappings

- Usually monotonic mappings required





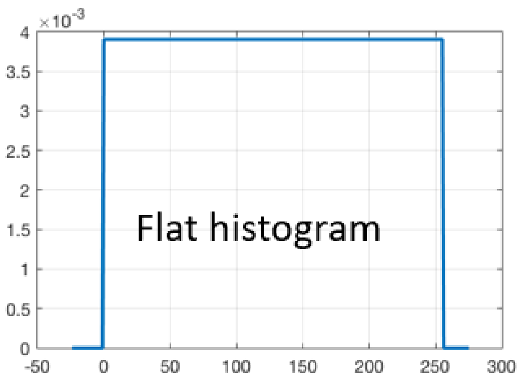
Generic transformation function

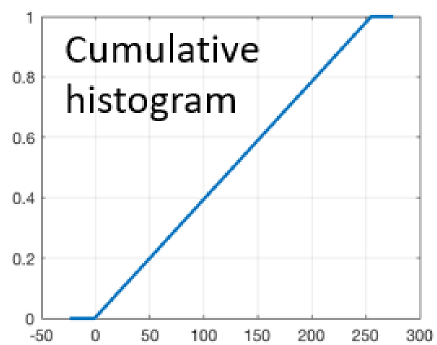
Power law transformation

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

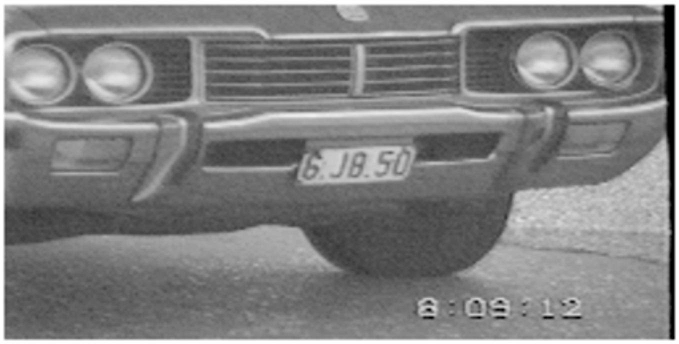
Histogram equalization

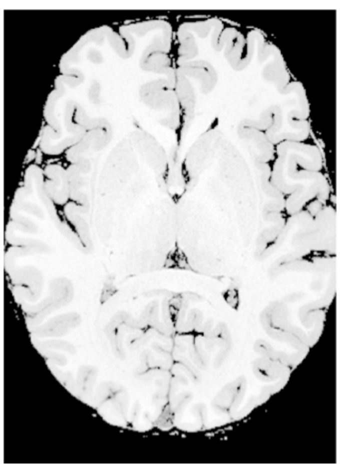
- Goal: create a flat histogram
- How: apply an appropriate intensity map□depending on the image content method will be generally applicable

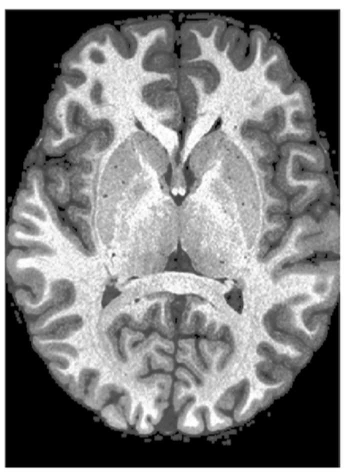




Histogram equalization example

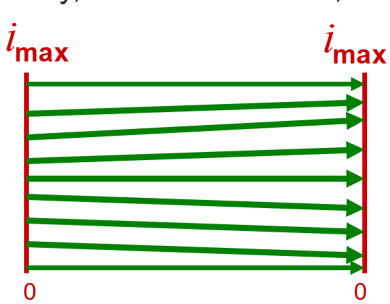






Histogram equalization - principle

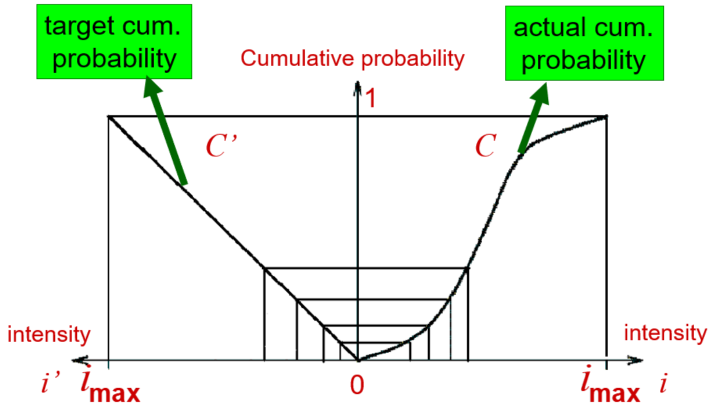
- Redistribute the intensities, 1-to-several (1-to-1 in the continuous case) and keeping their relative order, as to use them more evenly
- Ideally, obtain a constant, flat histogram



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!