Medical Image Processing

Module 4
Filtering and Transformations

Syllabus

- The filtering operation, the fourier transform, other transforms, discretization – resolution and artifacts, interpolation and volume regularization, translation and rotation, reformatting, tracking and image-guided therapy

Image Contrast

- *Global contrast* simply compares the ratio of difference between the highest and the lowest intensity values I_{max} and I_{min} of an image to the average intensity level given by the sum of I_{max} and I_{min}:

$$C_{\text{Michelson}} = \frac{l_{\max} - l_{\min}}{l_{\max} + l_{\min}}$$

- A somewhat better approach for measuring global contrast is the *root-mean square (rms) contrast*. Given an image (x,y) with $M \cdot N$ pixels and intensities $I(x, y)$, the expected value of I is

$$\bar{l} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} l(i, j),$$

and the *rms* contrast is

$$C_{\text{rms}}(f) = \sqrt{\frac{1}{MN-1} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(l(i, j) - \bar{l}\right)^2}.$$

Image Contrast

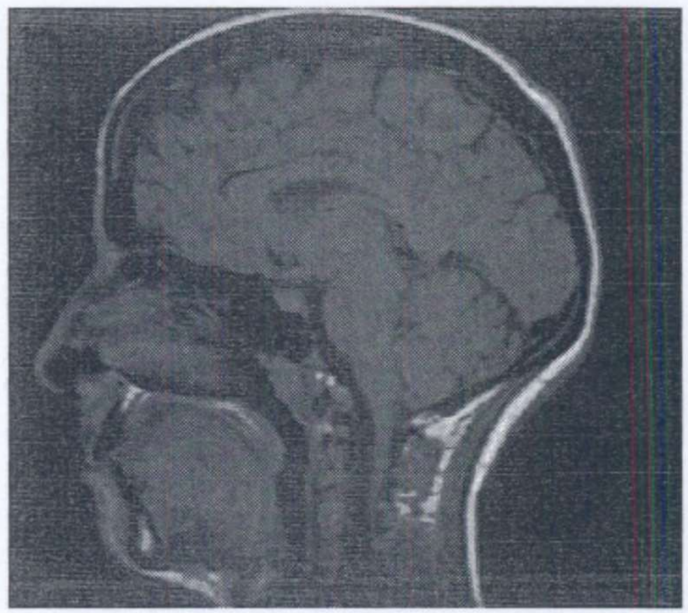


Figure .: An image with poor contrast.

Image Contrast

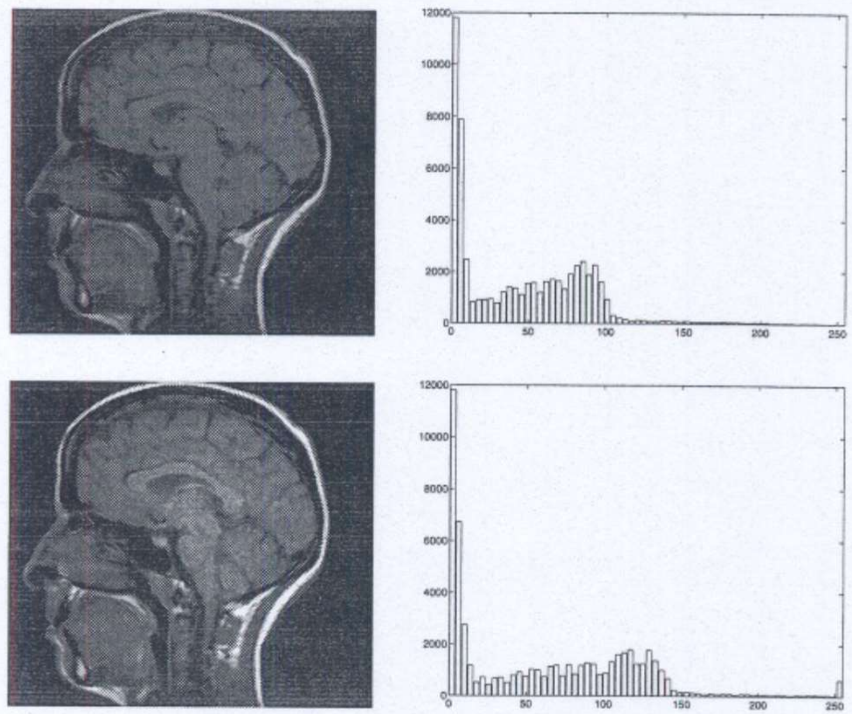


Figure The original image has very poor contrast since the gray values are in a very small range. Histogram scaling improves the contrast but leaves gaps in the final histogram. *Top:* Original image and histogram. *Bottom:* Image and resulting histogram after histogram scaling.

Image Contrast

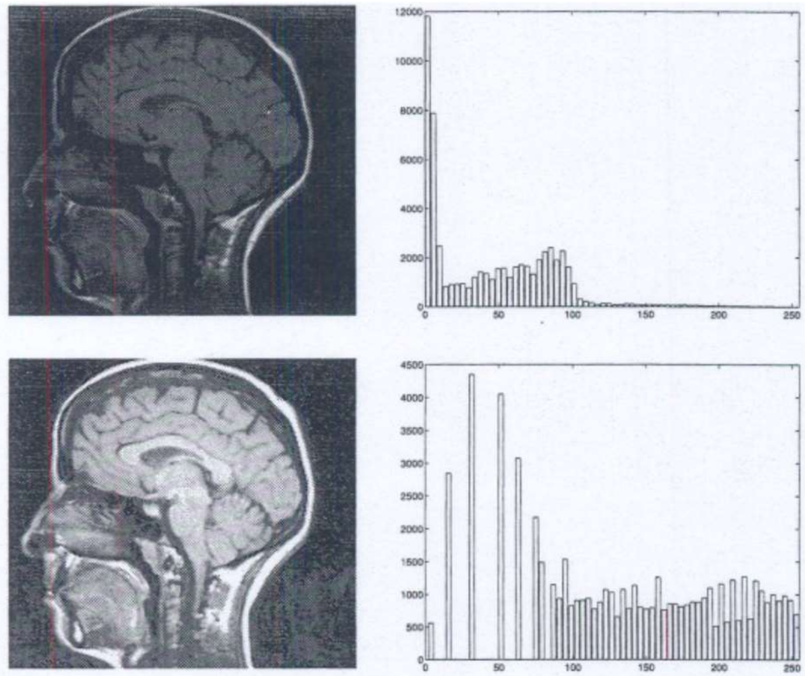


Figure The original image has very poor contrast since the gray values are in a very small range. Histogram equalization improves the contrast by mapping the gray values to an approximation to a uniform distribution. However, this method still leaves gaps in the final histogram unless pixels having the same gray levels in the input image are spread across several gray levels in the output image. *Top:* Original image and histogram. *Bottom:* Image and resulting histogram after histogram equalization.

Image Contrast

- Entropy still does not account for the fact that contrast should measure intensity differences between some foreground and background object. This can be computed using a *gray-level co-occurrence matrix (GLCM).*
- Co-occurrence calculates the normalized rates of co-occurring intensity values in a given neighborhood.
- The neighborhood is defined by the distance and direction between the two pixels.
- Hence, co-occurrence Ca,d is a two-dimensional function of intensities l_1 and l_2.
- $Ca,d\ (l1,\ l2)$ is the probability with which pixels with intensities $l1$ and $l2$ occur such that pixel $l1$ and $l2$ are d units apart at an angle of a with the x-axis.
- Co-occurrence matrices can be computed with different distances and different directions representing intensity changes between structures at different angles and with different sharpness at the edge.

Image Contrast

- For measuring contrast in a given image, co-occurrence is computed for a fixed distance (e.g., *d* = 1 pixel) and for arbitrary angles. *Cd (l*1*, l*2*)* is then the co-occurrence of pixels with gray levels *l*1 and *l*2 at distance *d* with an arbitrary angle
- For *d* = 1 this would be the four pixels of the 4-neighborhood. Contrast *C*GLCM is then defined as

$$C_{\text{GLCM}} = \frac{1}{I_{\max}^2} \sum_{i=0}^{I_{\max}} \sum_{j=0}^{I_{\max}} C_d(i, j)\left(1 + (i - j)^2\right) - 1$$

- Contrast thus weights the co-occurrences of two intensities by the difference betweenthe two. Higher differences indicating edges receive higher weights.

THE FILTERING OPERATION

- A filter in mathematics or signal processing describes a function that modifies an incoming signal. Since images are two- or three dimensional signals, a filter operation can, for instance, remove noise or enhance the contrast of the image.
- A filter modifies the original signal.
- To describe the performance a signal transferring system to take the performance on a single, point like source.
- The result of the filter on this point-like signal is called the Point Spread Function (PSF).
- If we want to know about the looks of the whole image, we simply have to apply the PSF to every single point in the original image.
- The process of blending the PSF with the original image is called convolution.
- In the convolution operation, the PSF is applied to every pixel, which of course might also affect the surrounding pixels in the resulting image. The convolution operation is denoted using the $\star$ sign.
- In various image processing operations performed by small matrices, which represent a convolution kernel.
- This kernel, can be considered a PSF, which is applied to every single pixel and re-distributes the gray values ρ in order to achieve the desired outcome.

THE FILTERING OPERATION

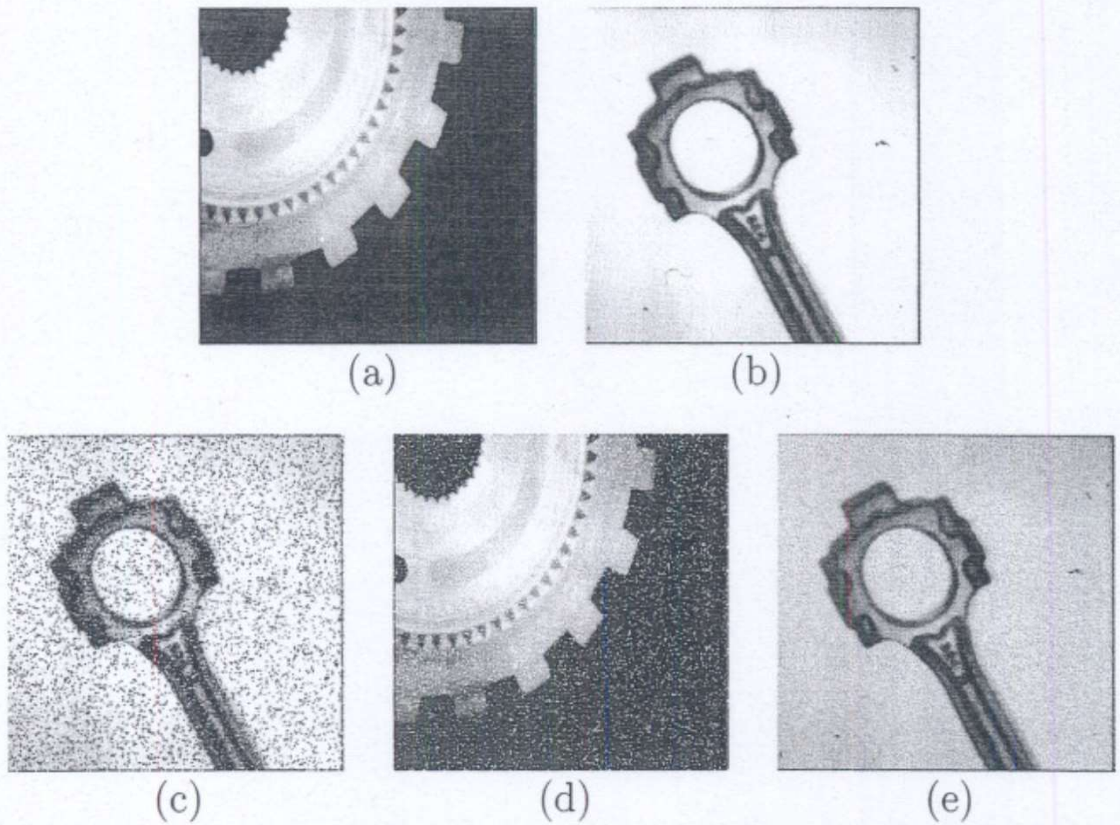


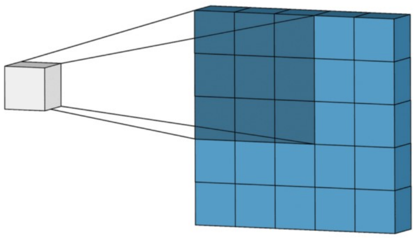
Figure Examples of images corrupted by salt and pepper, impulse, and Gaussian noise. (a) & (b) Original images. (c) Salt and pepper noise. (d) Impulse noise. (e) Gaussian noise.

THE FILTERING OPERATION

- Linear smoothing filters are good filters for removing Gaussian noise and in most cases, the other types of noise as well.
- A linear filter is implemented using the weighted sum of the pixels in successive windows.
- Typically, the same pattern of weights is used in each window, which means that the linear filter is spatially invariant and can be implemented using a convolution mask.
- If different filter weights are used for different parts of the image, but the filter is still implemented as a weighted sum, then the linear filter is spatially varying.
- Any filter that is not a weighted sum of pixels is a nonlinear filter.
- Nonlinear filters can be spatially invariant, meaning that the same calculation is performed regardless of the position in the image, or spatially varying.
- The median filter is a spatially invariant, nonlinear filter.

Kernel based smoothing and sharpening operations

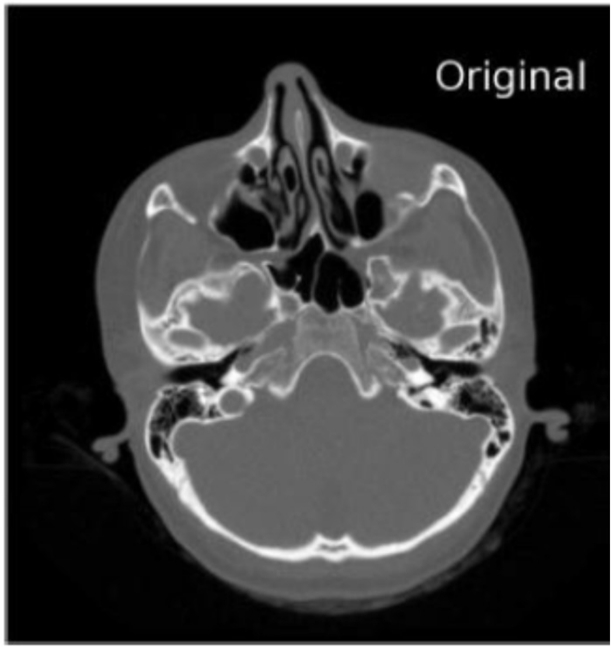
- A common task in image processing is the smoothing and sharpening of images; an image corrupted by noise – which can be considered ripples in the landscape of gray values ρ – may be improved if one applies an operation that averages the local surrounding of each pixel.
- Such a kernel K can, for instance, take the following form:



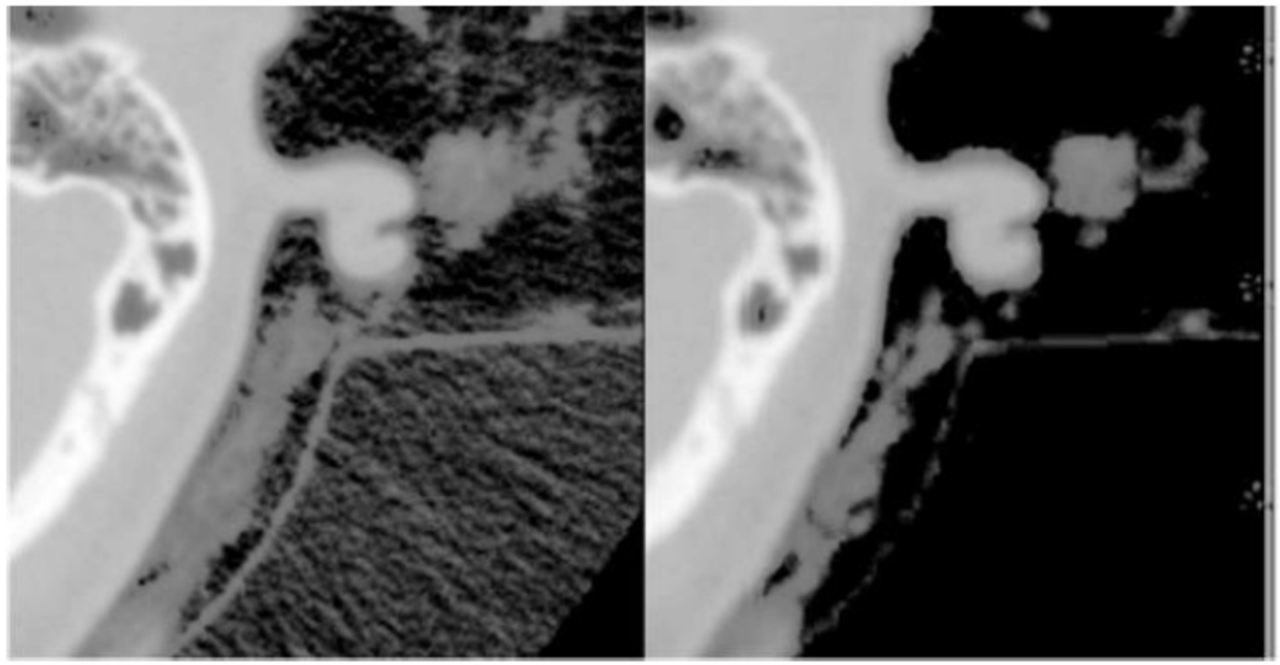
$$K_{\text{blur}} = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- The weighted average of the gray values surrounding each pixel in the original images is assigned to the same pixel position in the new image.
- Small fluctuations in the gray values are reduced by this averaging operation.
- However, the central pixel position is emphasized by receiving a higher weight than the others.
- If a kernel changes the sum of all gray values ρ, one might consider normalizing the whole image by a simple global intensity scaling to the same value Pi,j ρi,j found in the original image.

Kernel based smoothing and sharpening operations



The whole image (a CT slice in the vicinity of the skull base)



A detail from the SKULLBASE.DCM before and after the smoothing operation
On the left side, we see some reconstruction artifacts in the air surrounding the head. The smoothing filter Kblur successfully removes these artifacts by weighted averaging. For visualization purposes, the image intensities were transformed here in order to improve the visibility of low contrast details.

Kernel based smoothing and sharpening operations

- The opposite of smoothing is an operation called sharpening.
- In a sharpening operation, it is desired to emphasize edges in the images. Edges can be considered an abyss in the landscape of the image.
- A sharpening operator therefore does merely nothing if the surrounding of a pixel shows a homogeneous distribution of gray values.
- If a strong variation in gray values is encountered, it emphasizes the pixels with high intensity and suppresses the low intensity pixels.
- The classic sharpening operator looks like this:

$$K_{\text{sharp}} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Kernel based smoothing and sharpening operations

Figure shows after a sharpening operation using the K_{sharp} kernel.
Fine details such as the spongeous bone of the mastoid process are emphasized.

Kernel based smoothing and sharpening operations

- Both smoothing and sharpening operations are widely used in medical imaging.
- A smoothing operator can suppress image noise, as can be seen from Figure.
- In modalities with a poor SNR, for instance in nuclear medicine, this is quite common.
- Sharpening, on the other hand, is a necessity in computed tomography to enhance the visibility of fine detail,
- Figure shows two CT slices of a pig jaw; one is not filtered after reconstruction, the other one was filtered using the software of the CT.
- Besides the improvement in detail visibility and the increased visibility of reconstruction artifacts, we also witness a change in overall image brightness – a phenomenon. It is also an intensity scaling artifact

Kernel based smoothing and sharpening operations

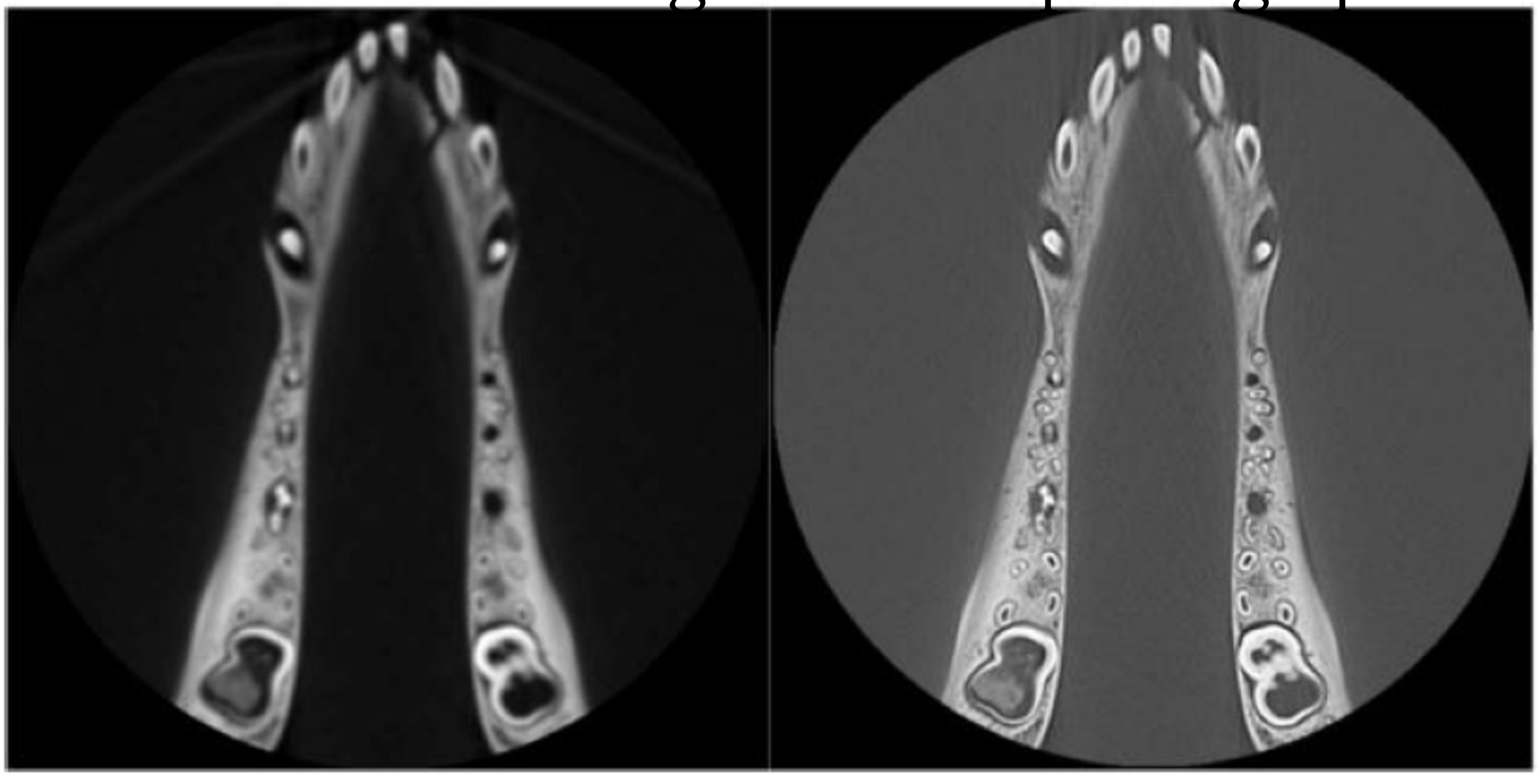


FIGURE : Two CT slices of a dry mandible of a pig. The left image was reconstructed without the use of a sharpening operator, the right slice was sharpened. Both images were directly taken from the DICOM-server connected to the CT and not further processed. Again, the change in overall image brightness from the use of the sharpening operator is clearly visible. This is, however, an intensity scaling artifact.

Kernel based smoothing and sharpening operations

- If we want to use more precise kernels, we may simply increase the number of pivoting values.
- For instance, a more sophisticated Gaussian kernel with five instead of three pivoting elements in each dimension looks like this:

$$K_{5\times 5\,\mathrm{Gauss}} = \frac{1}{256}\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- The single elements of the kernel $K_{5\times 5}$Gauss are approximated from the well known analytic expression for the Gaussian curve.

Differentiation and edge detection

- The numerical approximations of the derivation process can be used.
- differential expression df(x) / dx for a function f(x) = ρ becomes a finite difference:

$$\frac{df(x)}{dx} = \frac{\rho_{i+1} - \rho_i}{\Delta x}$$

- This is a forward difference . The numerator of the differential is defined as the difference between the actual value ρ_i and its next neighbor ρ_{i+1}.
- A function that maps from two or three coordinates to a scalar value (such as an image) features a derivative in each direction – these are the partial derivatives, denoted as $\partial I(x,y,z)/ \partial x$, $\partial I(x,y,z)/ \partial y$ and so on if our function is the well known functional formulation I(x, y, z) = ρ
- The forward difference is the equivalent of the partial derivative:

$$\partial I(x,y,z) \,/\, \partial x = \rho_{x+1,y,z} - \rho_{x,y,z},$$

where $\rho_{x,y,z}$ is the gray value at voxel position $(x, y, z)^T$;

the denominator x is one, and therefore it is already omitted.

Differentiation and edge detection

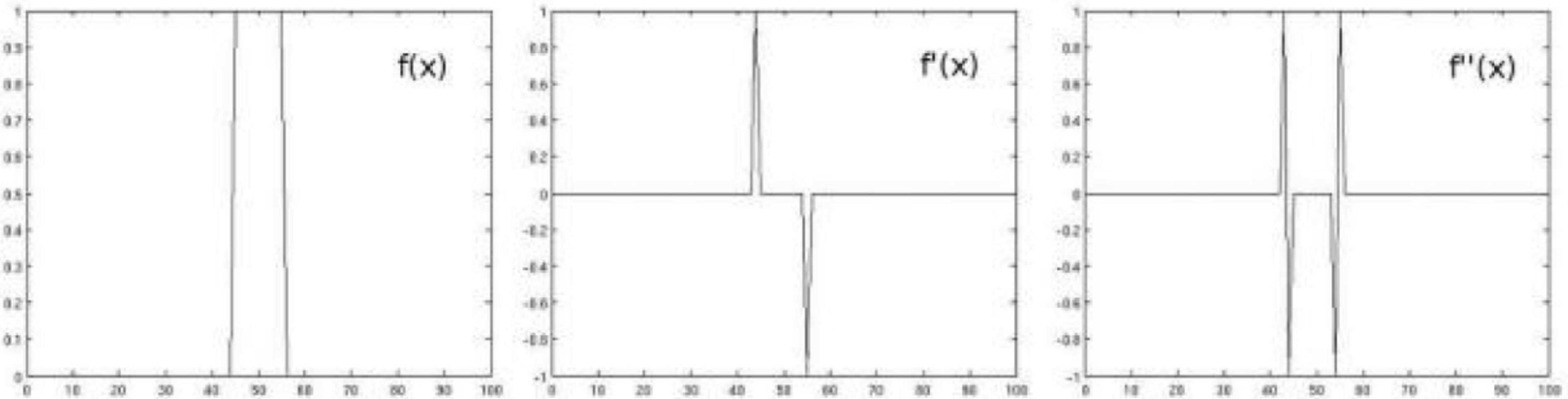


FIGURE: In the first plot, we see a simple rectangle function f(x). The first derivative $\frac{df(x)}{dx}$, computed by forward differentiation, is named $f'(x)$ in this illustration.
As we can see, the derivative takes a high value if something rapidly changes in f(x), and it becomes zero if neighboring values of f(x) are similar.
The same holds true for the second derivative $f''(x)$. In image processing, differentiation yields the edges of an image, whereas areas of similar gray values become black.

Differentiation and edge detection

- If we want to differentiate an image, we can define a kernel that produces the forward difference after convolution with the image. Such a kernel for differentiation in the x-direction is given as:

$$K_{\text{x-forward}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

- The correspondence between Kx-forward and df(x)/dx is obvious.
- The kernel subtracts the gray value ρx,y located at the central pixel from the gray value $\rho_{x+1,y}$ located to the right.
- Kx-forward therefore computes the partial forward difference of an image Ix,y.

Differentiation and edge detection

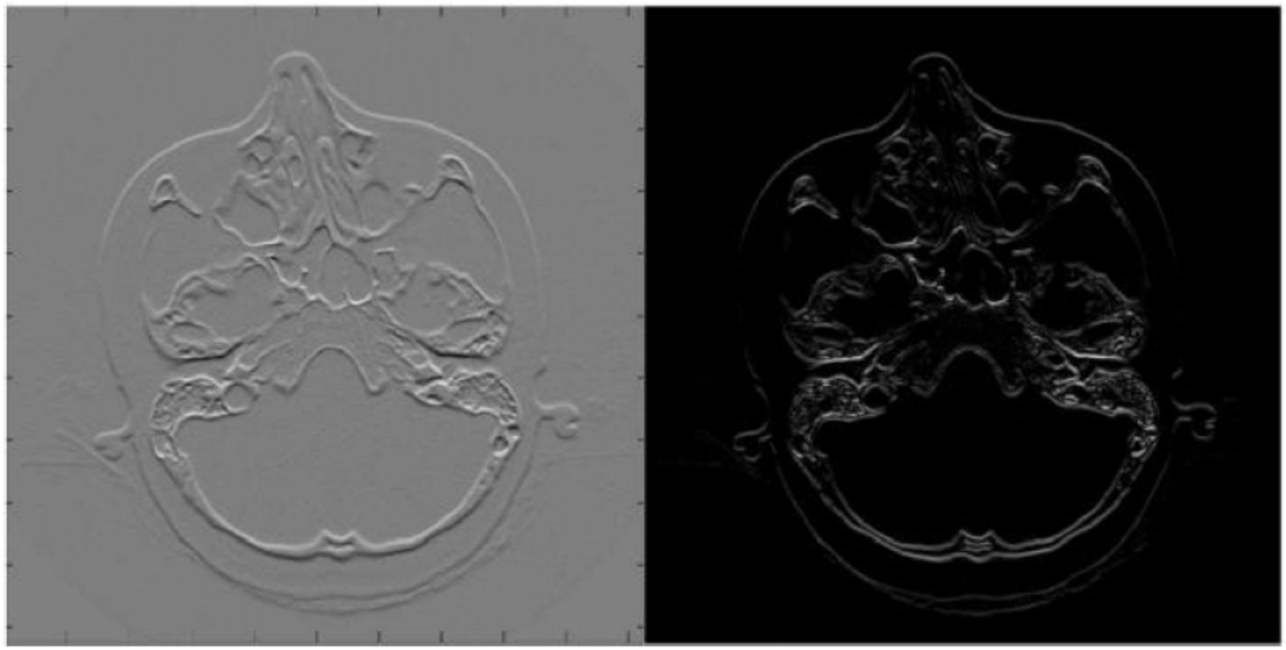


FIGURE: The left image shows the effects of $K_{x\text{-forward}}$ on the SKULLBASE.DCM-image. It is pretty evident why this filter is also called the bas-relief filter. When computing the absolute value of the convolution of $K_{x\text{-forward}}$ with SKULLBASE.DCM, the image on the right is produced. It does always give a positive value, no matter whether the image edges change from bright to dark pixel values or vice versa. In can also be seen that vertical edges like the skin surface close to the zygomatic arch is not emphasized; the x- and y-axis are swapped - this is a direct consequence of the matrix indexing conventions in MATLAB, where the first index is the column of the matrix image.

Differentiation and edge detection

- All the homogeneous gray value information is lost, and only the edges in the image remain.
- Due to the forward differentiation, the whole image shows some drift to the right hand side. Kx-forward is also known as the bas-relief kernel;
- While Kx-forward is an edge-detection filter in its simplest form, the dullness of the whole image is somewhat disturbing.
- The cause of this low contrast can be guessed from the middle image showing the first derivative of the rectangle function.
- When proceeding from a low value of f(x) to a high value, the derivative takes a high positive value; if f(x) is high and f(x + 1) is low, the derivative yields a high negative value.
- However, when detecting edges, we are only interested in the sudden change in image brightness – we are not interested in the direction of the change. A modification of Kx-forward is the use of the absolute value of the gradient.

Differentiation and edge detection

- If we want to know about the gradient – the direction of maximal change in our image I(x, y, z) = ρ – we have to compute all partial derivatives for the independent variables x, y, and z, and multiply the result with the unit vectors for each direction.

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^T$$

- it returns the gradient of a function as a vector.
- Computing the norm of the nabla operator and I(x, y, z) gives the absolute value of the maximal change in the gray scale ρ.

$$||\nabla I(x,y,z)|| = \sqrt{\left(\frac{\partial I(x,y,z)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y,z)}{\partial y}\right)^2 + \left(\frac{\partial I(x,y,z)}{\partial z}\right)^2}$$

Differentiation and edge detection

- The fact that this gradient length is always positive also makes the use of the absolute value of a differentiation kernel as proposed for Kx-forward unnecessary.
- The asymmetric nature of the forward differentiation, can be replaced by the average of the forward and the backward difference. This so-called central difference is given by

$$\frac{df(x)}{dx} = \frac{1}{2}\big(\underbrace{\rho_{i+1} - \rho_i}_{\text{Forward } \Delta} + \underbrace{\rho_i - \rho_{i-1}}_{\text{Backward } \Delta}\big)$$

Differentiation and edge detection

for a stepwidth of $\Delta x = 1$. This expression can be rewritten as $\frac{df(x)}{dx} = \frac{1}{2}\left(\rho_{i+1} - \rho_{i-1}\right)$. This central difference yields the following 2D convolution kernels for derivations in x and y

$$K_{\text{x-central}} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } K_{\text{y-central}} = \frac{1}{2}\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{5.8}$$

Therefore, the result of the operation

$$\sqrt{\left(K_{\text{x-central}} \star I(x,y)\right)^2 + \left(K_{\text{y-central}} \star I(x,y)\right)^2},$$

which is actually the length of the gradient of the image I(x, y) should give a rather good edge detection filter.

The $K_{\text{x-central}}$ and $K_{\text{y-central}}$ kernels obviously provide a nice edge detection filter,

Differentiation and edge detection

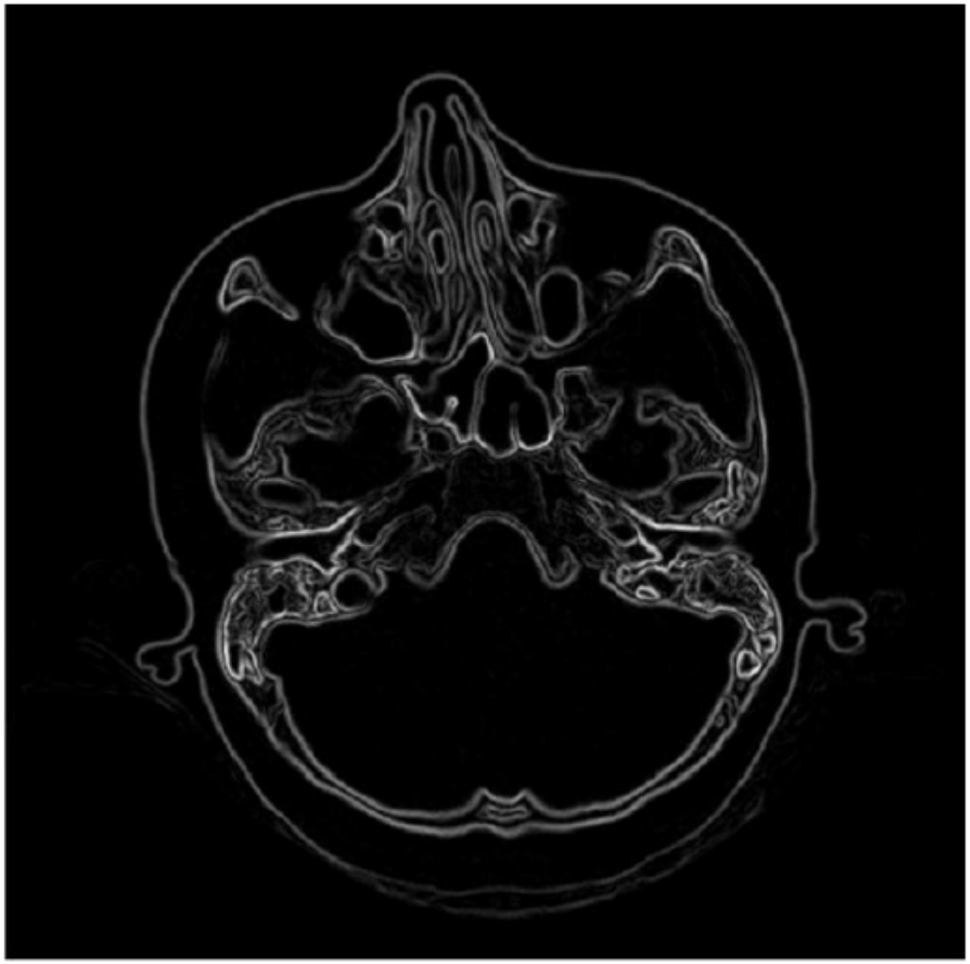


FIGURE: The effects of computing the length of the resulting gradient after convolution result of $K_{x\text{-central}}$ and $K_{y\text{-central}}$ with SKULLBASE.DCM.
This is already a pretty satisfying edge detector; due to the use of central differences, no shadowing effects appear.
The use of the total differential does respect changes in all directions, and the gradient length only yields positive results for the derivation result.

Differentiation and edge detection

- A simple smoothing kernel Kblur was enhanced by modelling a Gaussian curve over a 5 × 5 kernel.
- The general concept of connectedness can be introduced here; a pixel has four next neighbors – these are the four-connected pixels.
- Beyond that, there are also the eight-connected neighbors.
- If we add the contributions of the next but one neighbors to the center of the kernel, we are using the eight-connected pixels as well.
- Kx-central and Ky-central only utilize the four-connected neighbors, whereas Ksharp and Kblur are eight-connected.
- The neighbors on the corners of the kernel are located at a distance √2x from the center of the kernel with x being the pixel spacing.
- Therefore the weight of these image elements that are distant neighbors is lower. The concept of connectedness can of course be generalized to 3D.
- The equivalent to a four connected neighborhood is a six connected kernel (which also uses the voxel in front of and behind the central voxel), and if we also use the next but one voxels to construct our kernel, we end up with 26 connected voxels.

Differentiation and edge detection

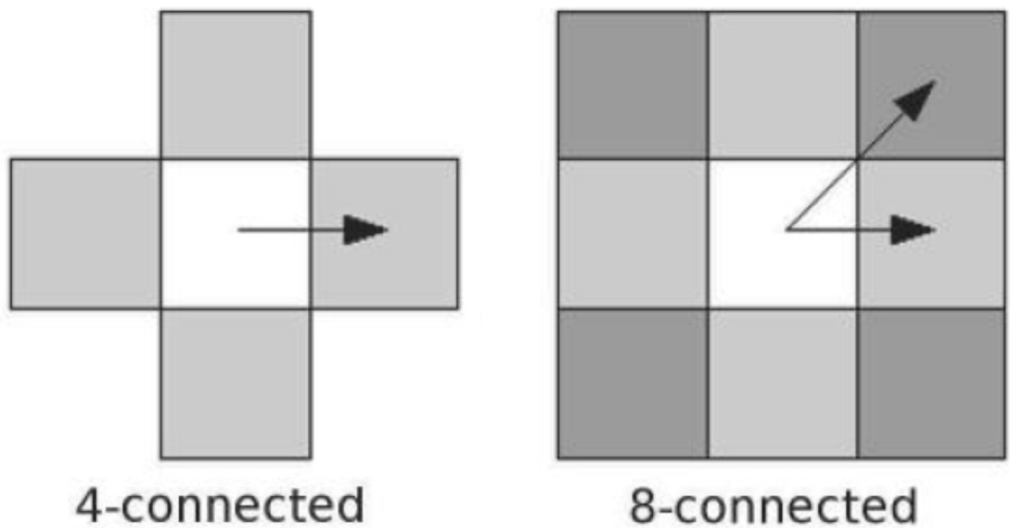


FIGURE: The four-connected and eight-connected neighborhood of a pixel. The contribution from the neighboring pixels depends on their distance; the pixels on the corner are at a distance that is larger by a factor √2 compared to the 4-connected neighbors. If a kernel utilizes information from these next but one neighbors, there has to be assigned a lower weight than to the nearest neighbors.

Differentiation and edge detection

these are the Sobel-kernels, which are among the best known edge detection operators in image processing.

$$K_{\mathrm{Sobel}_x} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ and } K_{\mathrm{Sobel}_y} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Differentiation and edge detection

- The kernels can be designed as a linear combinations of the kernels presented.
- The blurring, sharpening, and edge detection are among the most common and important operations in medical imaging.
- The kernel can be applied to the whole image, or one can divide the kernel in parts and apply the filter subsequently to the image.
- After fusing the outcome, the result is the same.
- If an image I(x, y) = I1(x, y)+s∗I2(x, y) is convolved with an arbitrary linear kernel K, the following identity is true:

$$K \star I(x,y) = K \star I_1(x,y) + s * K \star I_2(x,y)$$

Differentiation and edge detection

- The unsharp mask, which can be derived using blurring kernels such as $K_{5\times5\text{Gauss}}$ or K_{blur}; subsequently, the unsharp mask is subtracted from the original image.
- Therefore, an unsharp masking kernel K_{Unsharp} Mask using the simple smoothing filter K_{blur} looks like this:

$$K_{\text{Unsharp Mask}} = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Unity operator}} - w * K_{\text{blur}}$$

Differentiation and edge detection

- Enhancing the edges improves recognizing structures in images. Since automatic or interactive object delineation is a frequent task in image analysis, edge enhancement is often a prerequisite for tracking object boundaries.
- Edges are closely associated with the *intensity gradient* because the existence of an edge implies a local change of intensity.
- For a 2D image with continuous domain *(x, y)*, the gradient is a vector

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}(x, y) \quad \frac{\partial f}{\partial y}(x, y) \right)^{\mathrm{T}}.$$

- The gradient does not exist in the discrete domain *(i, j)*, but can be approximated by differences

$$\nabla f(i, j) \approx \begin{pmatrix} f(i, j) - f(i-1, j) \\ f(i, j) - f(i, j-1) \end{pmatrix}$$

$$\nabla f(i, j) \approx \begin{pmatrix} [f * D_x](i, j) \\ [f * D_y](i, j) \end{pmatrix}$$

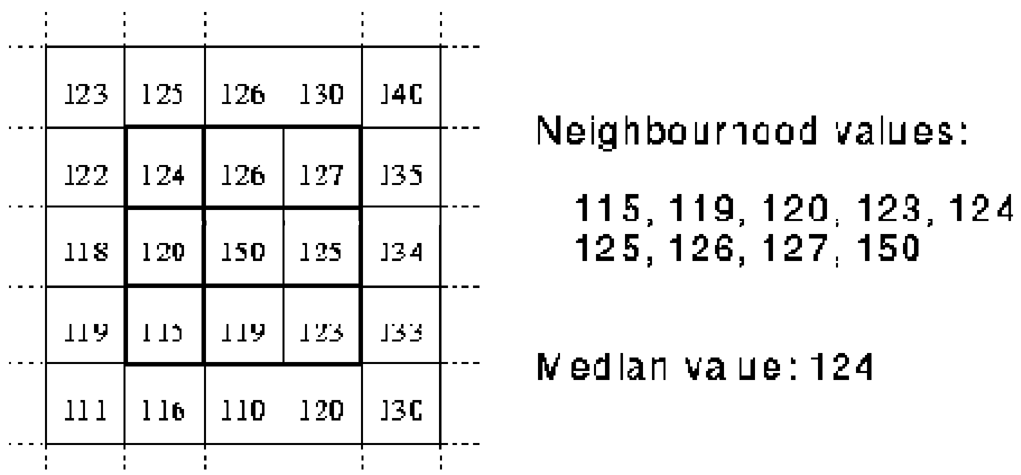
Since the edges and noise both have high frequency components, edge enhancement is often combined with smoothing. The two differences are computed by convolving the image with smoothing difference kernels

Non-linear filters

- The most helpful one is the median filter. The median of a set of random variables is defined as the central value in an ordered list of these variables.
- The ordered list of values is also referred to as the rank list.
- Let's take a look at a list of values xi for an arbitrary random variable, for instance xi ∈ 5, 9, 12, 1, 6, 4, 8, 0, 7, 9, 20, 666. The mean of these values is defined as

$$\bar{x} = \frac{1}{N} \sum_i x_i$$

- where N is the number of values
- xi ∈ 0, 1, 4, 5, 6, 7, 8, 9, 9, 12, 20, 666.
- The median of xi is 7.5 – the average of the values x6 and x7 in the ordered rank list.



Non-linear filters

- A median filter in image processing does something similar to a smoothing operation – it replaces a gray value ρ at a pixel location $(x, y)^T$ with an expectation value for the surrounding.
- It is remarkable that the median filter, however, does not change pixel intensities in an image – it just replaces some of them.
- As opposed to K_{blur} and $K_{5\times5Gauss}$, it does not compute a local average, but it uses the median of a defined surrounding.
- In general, the median filter for 3D and a kernel of dimension $n\times m\times o$ is defined in the following way:
  - Record all gray values ρ for the image elements in a surrounding of dimension $n\times m\times o$.
  - Sort this array of gray values.
  - Compute the median as the one gray value located in the middle of the sorted vector of gray values.
  - Replace the original pixel by this median.

Non-linear filters

- Compared to linear smoothing operators, the median filter retains the structure of the image content to a larger extent by preserving image edges while being very effective in noise suppression.
- The power of the median filter is demonstrated best when taking a look at "salt and pepper" noise, that is images stricken with black or white pixels, for instance because of defective pixels on the image detector.
- A median filter removes such outliers, whereas a simple blurring operation cannot cope with this type of noise.

Non-linear filters

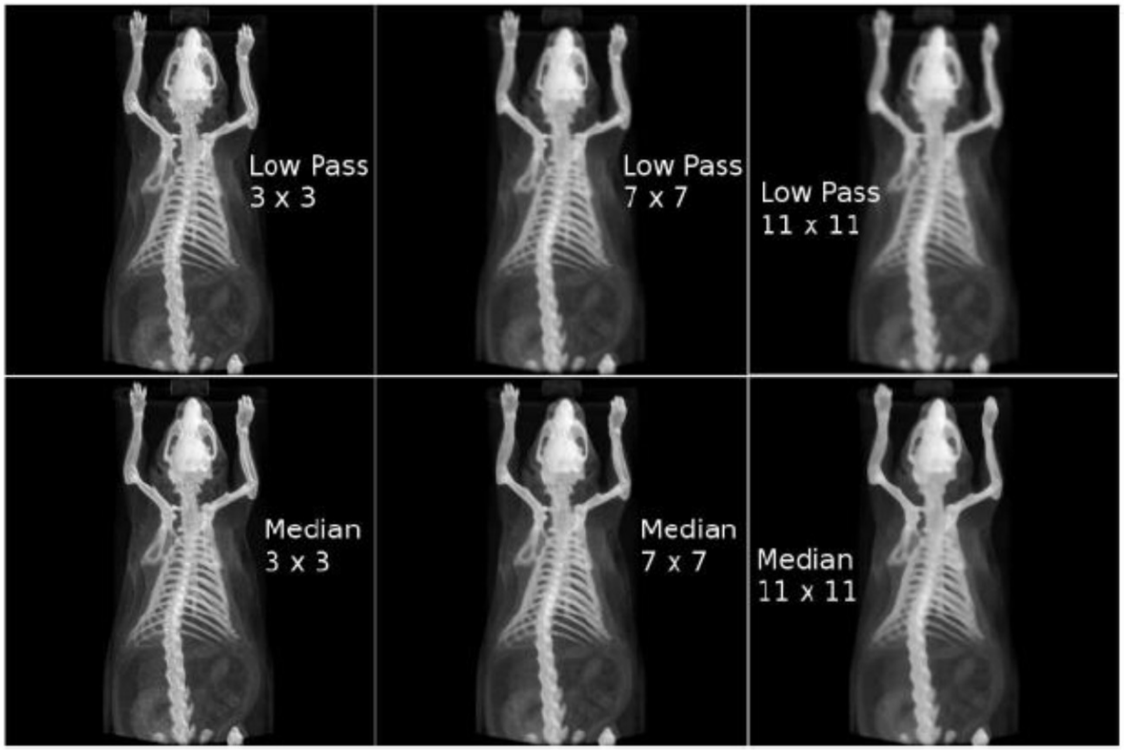


FIGURE: A topogram of a mouse from MicroCT with 733×733 pixels. The upper row shows the effects of low-pass filtering with varying kernel sizes. For comparison, the same image underwent median filtering with similar kernel sizes, which is shown in the lower row of images. While the median filter is a low-pass filter as well, the visual outcome is definitively different. The images can also be found on the CD accompanying this book in the JPGs folder.

Non-linear filters

- Not all non-linear filters are based on sorting or selection processes; an interesting filter that yields similar results compared to the median filter as a smoothing filter that retains general image structure is the anisotropic diffusion filter, which will be introduced as an example for a whole class of iterative filters.

Non-linear filters

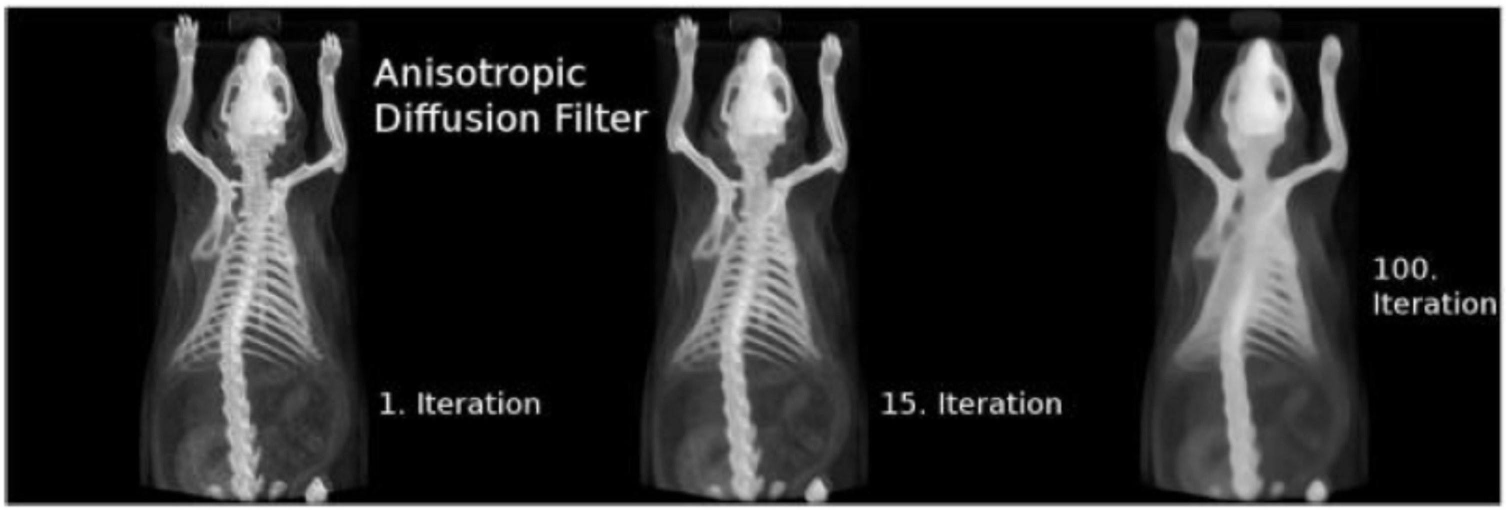
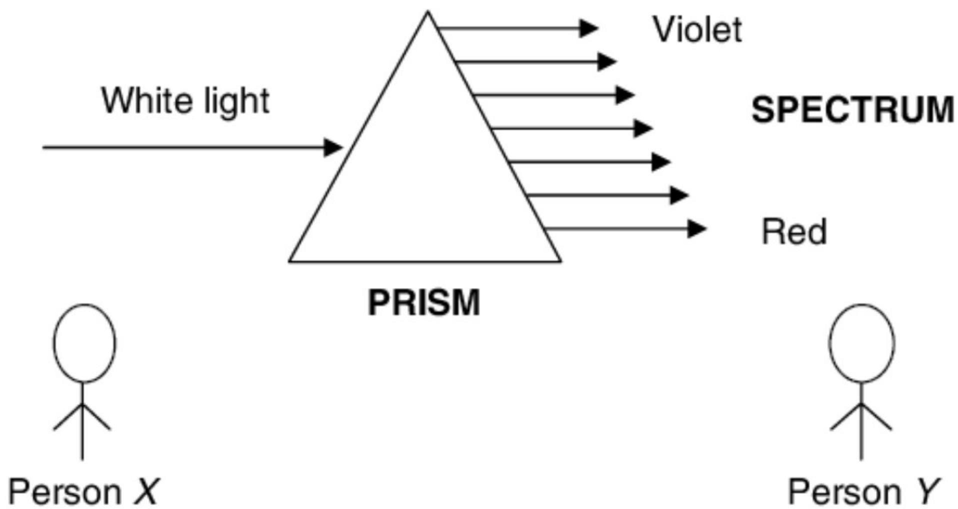


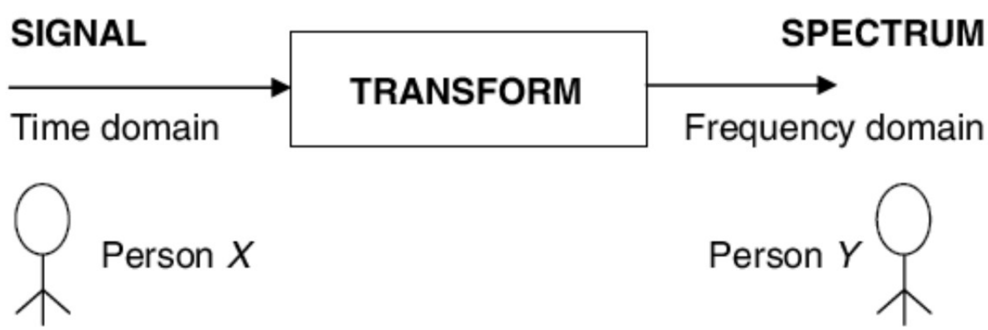
FIGURE: The same topogram as in above Figure after anisotropic diffusion filtering. As the diffusion process proceeds (as indicated by the number of iterations), one can see how the images get more blurred; still, structure is largely retained. The anisotropic diffusion filter is another example for a non-linear lowpass filter. Here, the large number of parameters allow for a very fine tuning of the degree of blurring and noise reduction.

THE FOURIER TRANSFORM

- The filters are used to modify an image in a specific manner
- For example, by enhancing discontinuities (sharpening) or attenuating detail (blurring).
- These filters act differently on different spatial frequencies.
- The term *spatial frequency* refers to the rate of change of the image values.
- An intensity trend that continues over most of the image consists of low spatial frequencies; conversely, an edge (a rapid change of intensity over a few pixels) contains high-frequency components.
- The frequency domain is a different representation of the same image data where the strength of periodic components (such as a pattern that repeats every 20 pixels) is extracted.
- The Fourier transform is a tool that converts the image with spatially arranged data into the same data arranged by periodicity, and thus, frequency.
- Any transform is characterized by the existence of an inverse transform that allows us to restore the original arrangement of the data.
- By using the Fourier transform, image data can be presented in a way that makes a number of image manipulations easier (or possible, in the first place): Images can be manipulated or filtered in the frequency domain.
- With the inverse Fourier transform, the filtered image is restored to its original spatial arrangement.

THE FOURIER TRANSFORM





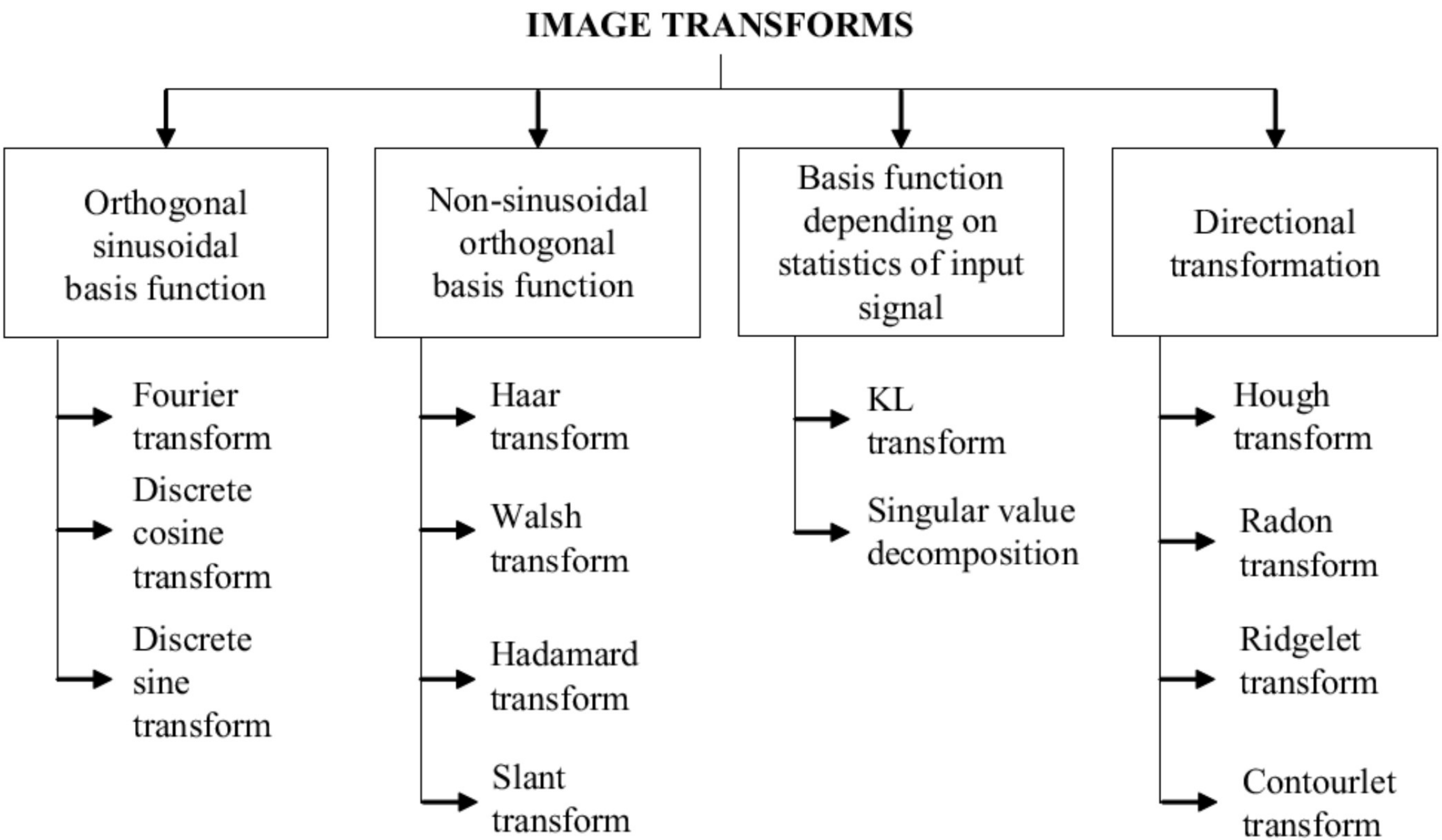


IMAGE TRANSFORMS
Orthogonal sinusoidal basis function
Fourier transform
Discrete cosine transform
Discrete sine transform
Non-sinusoidal orthogonal basis function
Haar transform
Walsh transform
Hadamard transform
Slant transform
Basis function depending on statistics of input signal
KL transform
Singular value decomposition
Directional transformation
Hough transform
Radon transform
Ridgelet transform
Contourlet transform

THE FOURIER TRANSFORM

- French mathematician Joseph Fourier found that any 2-periodic signal *f* (x) can be represented by an infinite sum of sine and cosine terms according to

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kx + \sum_{k=1}^{\infty} b_k \sin kx$$

- These Fourier Series are built, using cos kx, k =0, 1, . . . and sin kx, k = 0, 1, . . . as base functions
- where *ak* and *bk* are called the *Fourier coefficients*. The term 2-*periodic* indicates that the signal repeats after integer multiples of 2, that is, *f* (x) = *f* (*x* + 2*n*), where *n* is an integer number.

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} dx f(x) \cos kx, \ b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} dx f(x) \sin kx$$

- The periodicity is a consequence of the sine and cosine terms in Equation and will become a fundamental assumption in the Fourier analysis of signals and images. The cosine and sine functions in Equation form an orthogonal basis, since their inner product is zero:

THE FOURIER TRANSFORM

- Complex numbers have a number of special properties;
- Addition is carried out by applying the operation to the real and imaginary part separately.
- Multiplication is carried out by multiplying the doublets while keeping in mind that $i^2 = -1$.
- Complex conjugation is a very simple operation which consists of switching the sign of i, and it is denoted by an asterisk.
- In the case of our complex number c, the complex conjugate reads $c* = a - ib$.
- The norm of a complex number: $||c|| = \sqrt{a^2 + b^2}$.
- The norm operator yields a real number. The Real and Imaginary operator.
- These return the real and the imaginary part separately.
- The one identity that is of the utmost importance for us is the following; it is called
- Euler's formula: $e^{i\phi} = \cos\phi + \mathbf{i}\sin\phi$

THE FOURIER TRANSFORM

- for a complex representation of the Fourier series using the inner product

$$f(x) \bullet g(x) = \int_{-\pi}^{\pi} dx\, f(x) g^*(x)$$

- And base functi $e^{\mathbf{i}kx}, \; k = -\infty, \ldots \infty,$ which represent actually the plane waves.

$$f(x) = \frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} c_k e^{\mathbf{i}kx}$$

with Fourier coefficients

$$c_k = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} dx\, f(x) e^{-\mathbf{i}kx}$$

THE FOURIER TRANSFORM

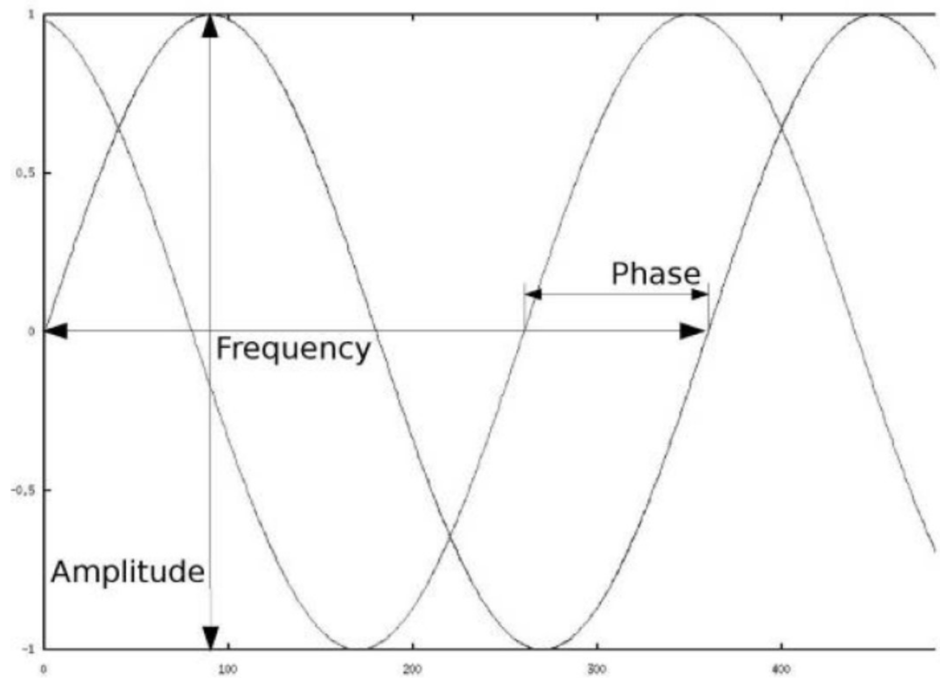


FIGURE: Two sines, which may be considered simple plane waves; the two variables that influence the shape of the sine are its frequency, and its amplitude. Furthermore, we have a second sine here with identical frequency and amplitude, but with a different phase. Periodical signals can be composed out of such simple waves with different frequencies, amplitudes and phases. The Fourier-transformation is a mathematical framework to retrieve these components from such a superimposed signal. Images can be considered 2D or 3D signals, and therefore it is possible to decompose them to simple planar signals as well. The sum of all frequencies and amplitudes is the spectrum of the signal.

THE FOURIER TRANSFORM

- The Fourier-transform of an arbitrary function f(x) as:

$$
\begin{aligned}
\hat{f}(k) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx f(x) e^{-\mathbf{i}kx} \\
f(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk \hat{f}(k) e^{\mathbf{i}kx} \\
k &\ \ldots \ \text{Wave number}
\end{aligned}
$$

- The Fourier transformation is an inner product of the function f(x) with the base functions e^{-ikx}. The wave number k is related to the wavelength λ of a wave by k = 2π /λ ; the wavelength by itself is related to the frequency ν as λ = c /ν where c is the propagation speed of the wave.
- A particular wave number kn is therefore the component of the function f(x) that is connected to a wave e^{-iknx}.
- The Fourier-transformation is therefore a transformation to a coordinate system where the independent variable x is replaced by a complex information on frequency (in terms of the wave number) and phase.
- The space of the Fourier transform is therefore also referred to as k-space.

THE FOURIER TRANSFORM

- To use the Fourier-transformation for image processing, then replace the integrals by sums. This is called the discrete Fourier transformation (DFT).
- In general our images are of finite size – they are not defined on a domain from $-\infty$ to ∞
- This problem can be resolved by assuming that the images are repeated, just like tiles on a floor.
- In an MR-tomograph, the original signal is measured as waves (in k- space), and is transformed to the spatial domain (the world of pixels and voxels) by means of a Fourier-transform.

THE FOURIER TRANSFORM

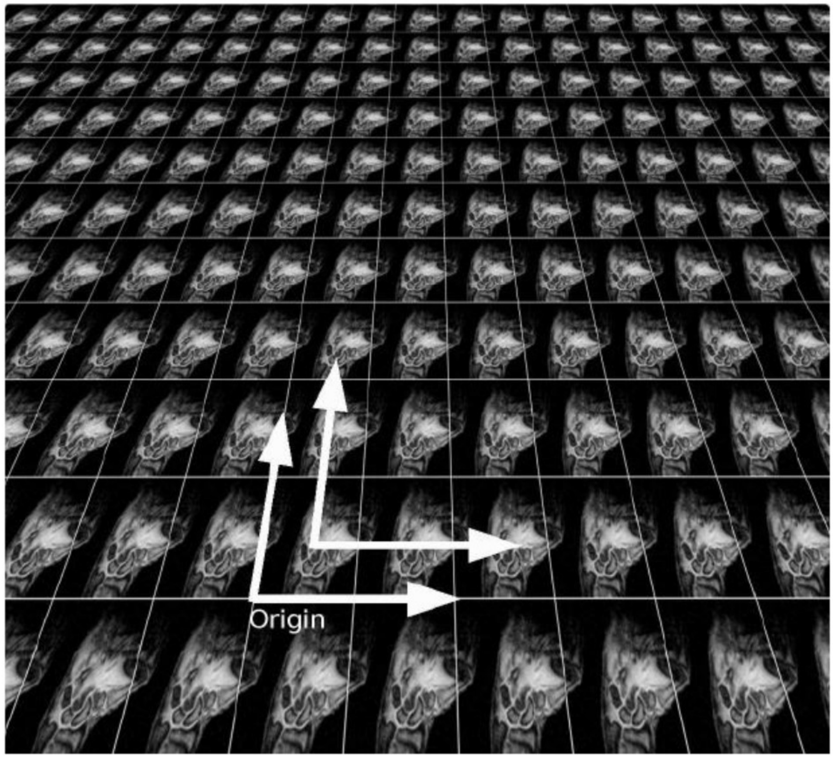


FIGURE: When applying a Fourier transformation to an image of finite size, it is assumed that the image is repeated so that the full spatial domain from $-\infty$ to ∞ is covered. An image, however, has a coordinate origin in one of the image corners; in order to establish symmetry, MATLAB provides a function called fftshift. This function shifts the origin of the coordinate system to the center of the image.

THE FOURIER TRANSFORM

The hand moves out of the field-of-view of the tomograph.
What occurs is a so called wraparound artifact.
The fingers leaving the image on the right side reappear on the left side of the image.

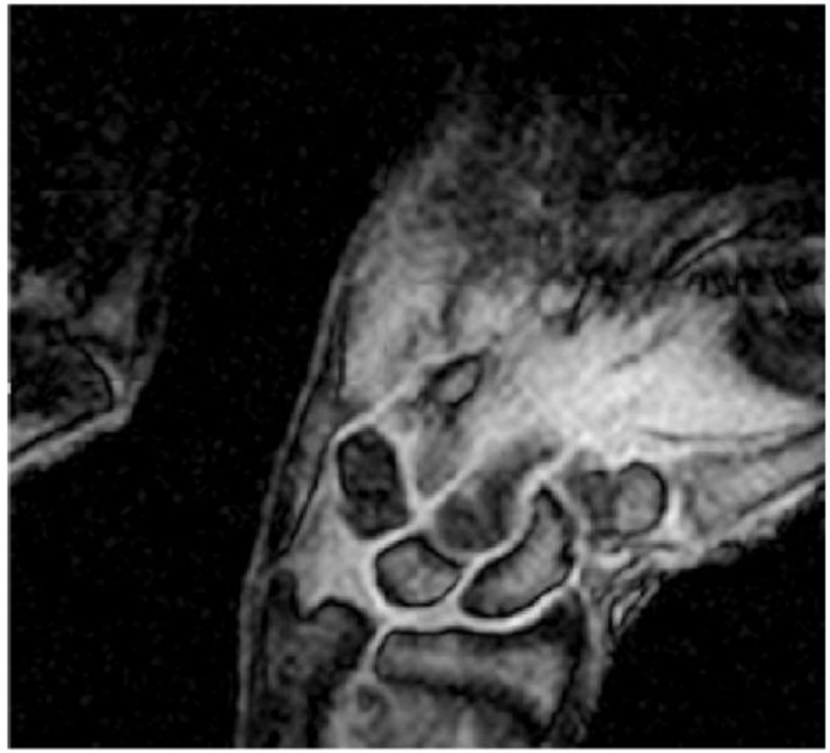


FIGURE: A wraparound artifact in MRI. In this dynamic sequence of a wrist in motion, a part of the hand lies outside the field-of-view. Since reconstruction of volumes in MR takes place using a Fourier transformation of the original signal, the image continues like a tile so that the transformation is defined on the whole spatial domain from $-\infty$ to ∞ – see also earlier Figure . The right part of the hand therefore re-appears on the left side of the image

THE FOURIER TRANSFORM

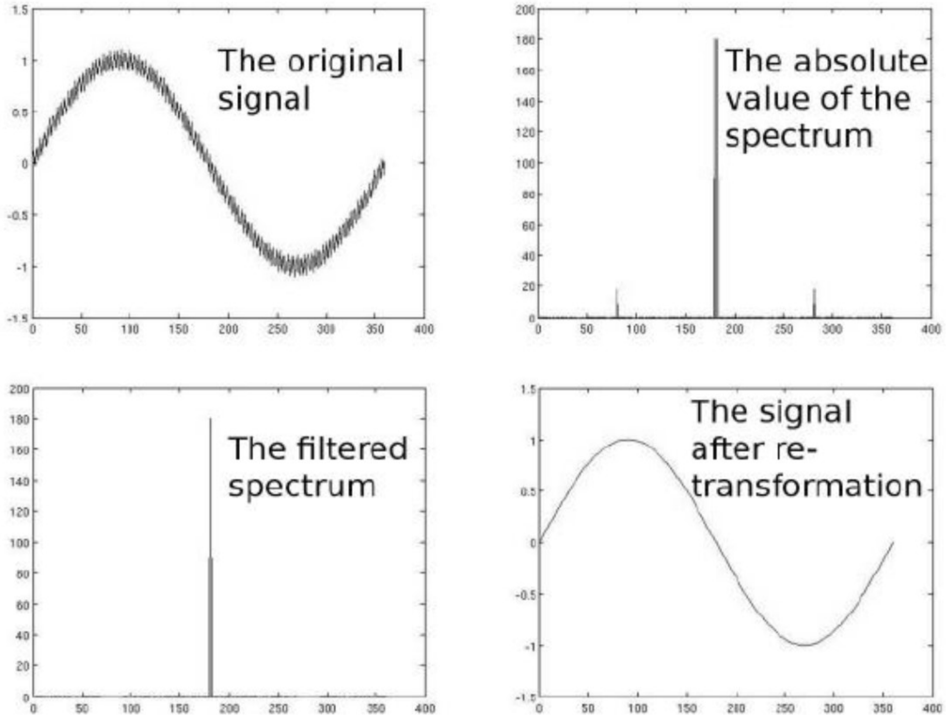


FIGURE : The four plots generated by the Sine_5.m script. The first plot shows the initial signal y – a sine overlaid by a smaller sine with higher frequency. The second and third plot show the absolute value of the spectrum. The frequency of the wx signal, which has a higher amplitude, is found in the middle of the plot. The higher frequency of the noise wxx is further to the left and the right of the origin of the coordinate system – it is lower since the amplitude of wxx is smaller. After removal of the contributions of wxx to the spectrum, the transformation back to the spatial domain shows the signal wx without the ripple added by wxx.

THE FOURIER TRANSFORM

- An image is a function defined on the 2D or 3D domain, it can be decomposed to plane or spherical waves; noise, for instance, is a signal with a high frequency.
- If we want to get rid of noise, we may reduce the weight of the higher frequencies (the components with a large value of the wave number k).
- If we want to sharpen the image, we can emphasize the higher frequencies.
- These operations are therefore obviously equivalent to K_{blur} and K_{sharp}, which were already called high-pass and low-pass filters.
- The advantage of blurring and sharpening in the Fourier domain, however, is the possibility to tune the operations very specifically. We can select various types of noise and remove it selectively,

THE FOURIER TRANSFORM

- The Fourier transformation has some properties of particular interest for signal and image processing; among those are:

Linearity: $w * f + g \mapsto w * \hat{f} + \hat{g}$.

Scaling: Doubling the size of an image in the spatial domain cuts the amplitudes and frequencies in k-space in half.

Translation: $f(x + \alpha) \mapsto \hat{f}(k)e^{ik\alpha}$. A shift in the spatial domain does not change the Fourier transform $\hat{f}(k)$ besides a complex phase. This property is also responsible for so-called ghosting artifacts in MR, where motion of the patient during acquisition causes the repetition of image signals

Convolution: $f(x) \star g(x) \mapsto \sqrt{2\pi} * \hat{f}(k) * \hat{g}(k)$. As you may remember from the introduction to this chapter, it was said that the convolution operation by mangling kernels into the image will be replaced by a strict and simple formalism. Here it is. In k-space, the convolution of two functions becomes a simple multiplication. Therefore, it is no longer necessary to derive large kernels from a function such as the Gaussian

THE FOURIER TRANSFORM

Differentiation: $\frac{d^n}{dx^n} f(x) \mapsto (\mathbf{i}k)^n \hat{f}(k)$. Computing the derivative of a function $f(x$ apparently also pretty simple. Again it becomes a simple multiplication.

The Gaussian: The Gaussian $G(x)$ retains its general shape in k-space. Under so circumstances, is even an *eigenfunction* of the Fourier transform;[4] in general, howe $\hat{G}(k)$ is a Gaussian as well.

Parseval's Theorem: The total energy in the image, which can be considered sum of squares of all gray values ρ, is maintained in k-space. Another formulatio $f \bullet g \mapsto \hat{f} \bullet \hat{g}$.

THE FOURIER TRANSFORM

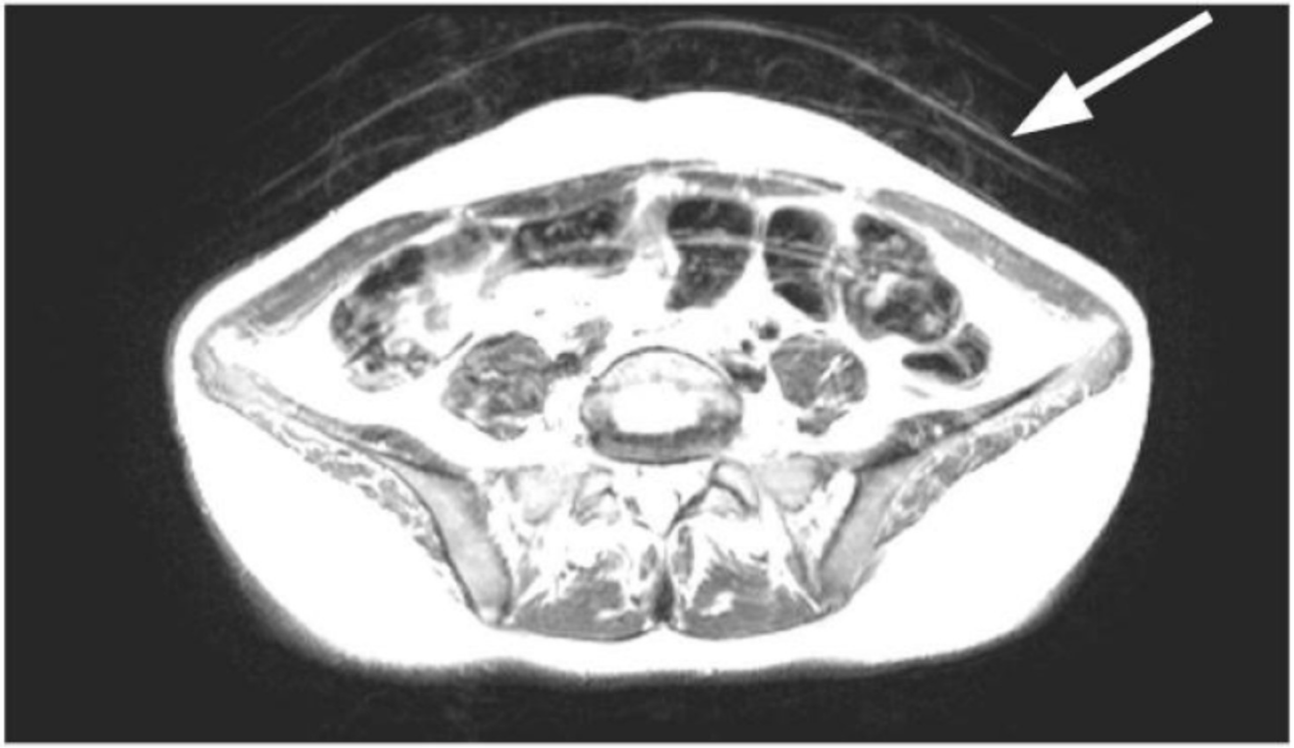


FIGURE: The translation property of the Fourier transform at work. As said before, the MR signal is acquired in k-space and transformed to the spatial domain for volume reconstruction. When patient motion during image acquisition occurs, the result is the repetition of image structures in the reconstructed volume (see arrow). This T2-weighted image of the pelvis was windowed in such a manner that the ghosting artifact – the repetition of the abdominal wall in the upper part of the image – becomes prominent. Ghosting artifacts in this area of the body are common due to breathing motion.

THE FOURIER TRANSFORM

FIGURE: The topogram of a mouse taken from small animal CT, and the absolute value of the Fourier-spectrum after a logarithmic intensity transform. Some of the details in the left image from the spatial domain leave a recognizable trace in the spectrum on the right-hand side. For instance, the low frequencies in the y-direction are more dominant than in the x-direction since the mouse occupies more space in the y-direction. Furthermore, the ribcage gives a significant signal of high intensity at a higher frequency, and the bow-shaped structures in the medium frequency range indicate this.

Image processing in the frequency domain

- In medical imaging, periodical artifacts may also occur, for instance in MR – imaging, or in CR plates, which may be stricken by discretization artifacts.
- An example of a bandstop filter, which removes some of the middle frequencies in the image,
- Of course, we can also introduce blurring and sharpening by removing the high or low frequencies in the image.
- The maximum wave number k, which represents the highest frequency in the image, is defined by the resolution of the image since the smallest signal given in the image is a single point.

Image processing in the frequency domain

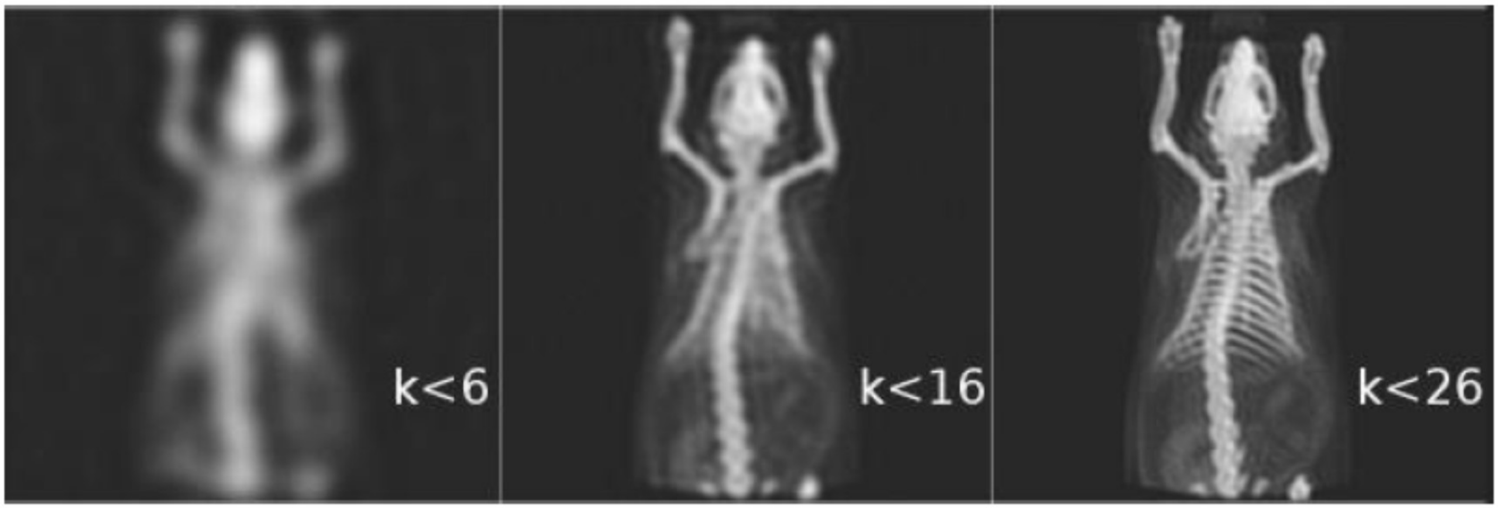


FIGURE: The output from script MouseFourierLP_5.m. Instead of smoothly suppressing higher frequencies like the K_{blur} kernel does, we simply cut off the frequencies, which leads to a lower image quality. A more sophisticated method would produce images just like

Image processing in the frequency domain

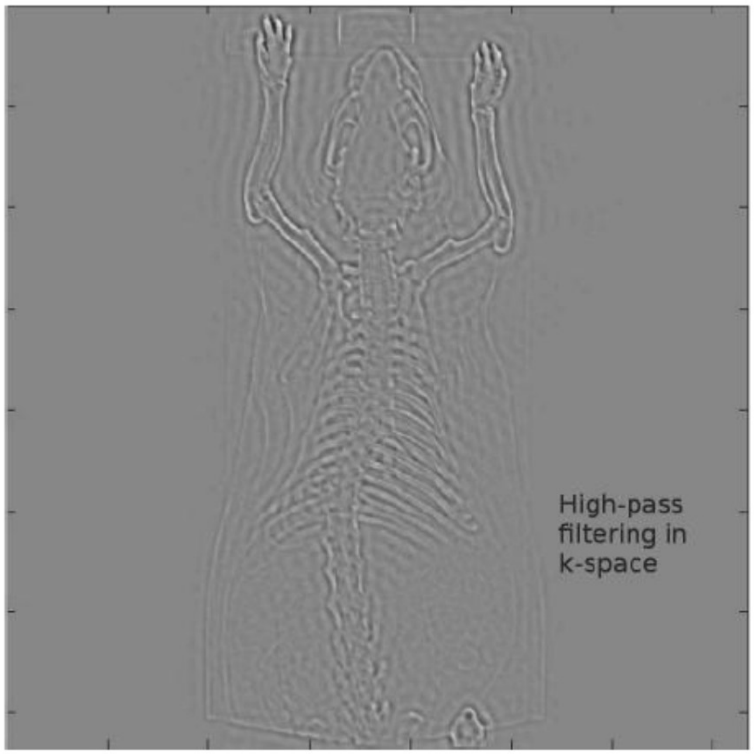


FIGURE : The results of a simple high pass filter in k-space applied to MouseCT.jpg.

Image processing in the frequency domain

- Important property of the Fourier-transformation is the convolution-operation; where small 2D-functions like Kblur were convolved with the image.
- In k-space, convolution becomes a multiplication.
- We can convolve kernels of all kinds (some of them have names such as Hann, Hamming, Butterworth and so on) by simply transforming both the kernel and the image to k-space, and multiply the two.
- After retransformation to the spatial domain, we get the effect of the convolution operation.
- Remember that $K_{5\times5\text{Gauss}}$, where a simple 5×5 kernel already requires 25 operations per pixel for convolution.
- Furthermore, we can invert the convolution operation; if we know the exact shape of our PSF, we can apply the inverse process of the convolution operation in k-space, which is actually a multiplication with the inverse of the PSF.
- This process is called deconvolution or resolution recovery.

Modelling properties of imaging systems – the PSF and the MTF

- The ability to distinguish two point sources depending on their distance is given by the so-called modulation transfer function (MTF).
- The MTF gives a score whether the two images overlap as a function of the spatial frequency that separates the two PSF-like images in term of cycles over range.
- The MTF is the absolute value of the Fourier transform of the PSF
- A narrow Gaussian will not distort the image a lot – fine detail will remain distinguishable; therefore the MTF is wide.
- While the Fourier-transform of a narrow Gaussian is still a Gaussian, it becomes a very wide distribution.
- If we apply a wide Gaussian PSF, the MTF will become narrow since fine details vanish.

Modelling properties of imaging systems – the PSF and the MTF

FIGURE: M13, the great globular cluster in the Hercules constellation. The ability to resolve two stars as separated depends on the PSF of the optical system used. In a poor optical system like the spherical mirror presented in the introduction, the image would not look like that since the amount of overlap between two images of the nearby stars would fuse them. The ability to distinguish two close objects is measured by the modulation transfer function, the MTF.

Modelling properties of imaging systems – the PSF and the MTF

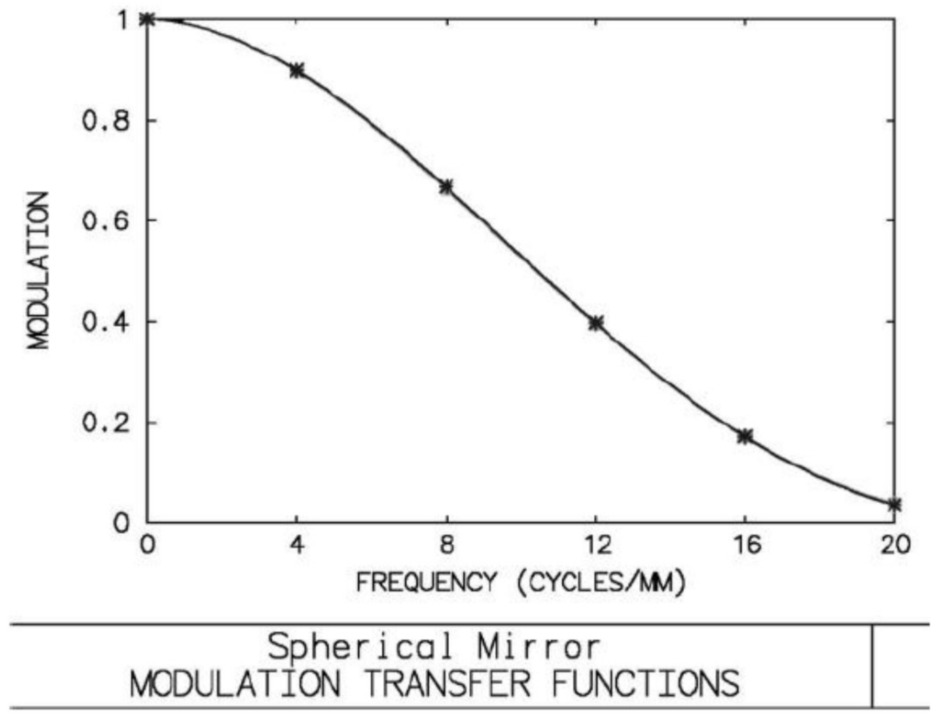


FIGURE : The modulation transfer function of the spherical mirror. The graph was, again, produced using the OSLO optical design software.

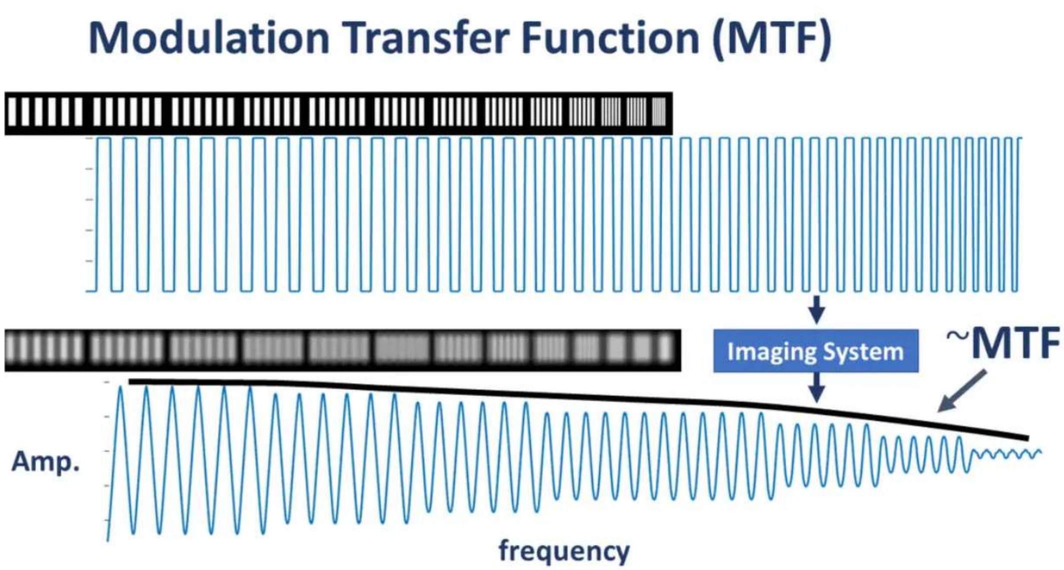
The MTF gives a figure of the capability of a signal-transducing system to resolve two point-shaped objects.

The higher the spatial frequency (given in cycles/mm in the image plane in this example), the lower the resolution of the system.

The MTF is coupled to the PSF by means of the Fourier-transformation; a wide PSF gives a narrow MTF and vice versa.

Modelling properties of imaging systems – the PSF and the MTF

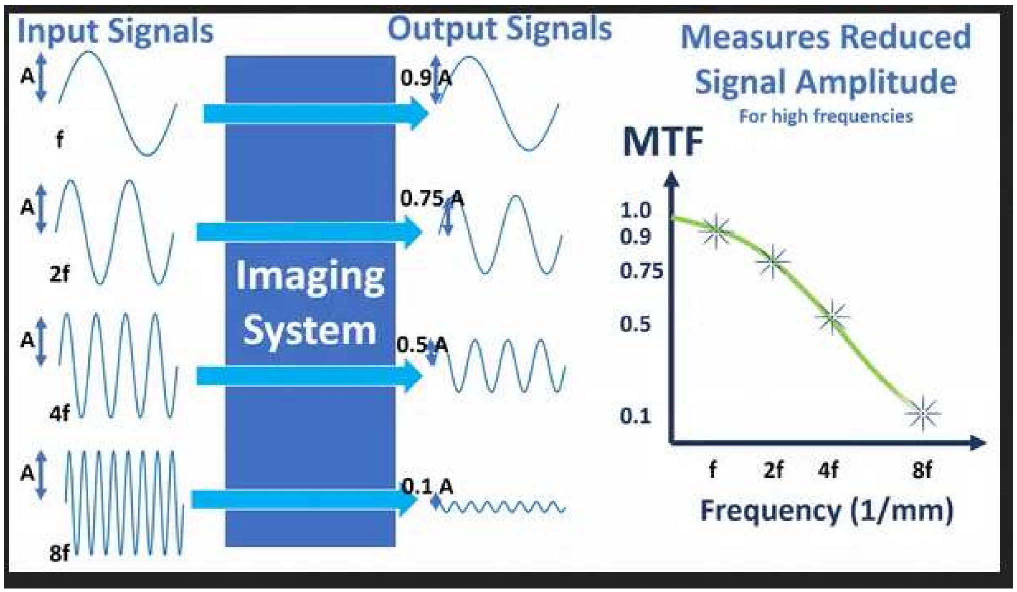
- The modulation transfer function (MTF) describes the frequency behavior of the system and is a curve that has lower values for high frequencies which represent the small image structures.
- Small image structures correspond to high frequencies, and large image structures correspond to low frequencies.
- The modulation transfer function (MTF) is the Fourier transform (frequency space representation) of the point spread function (PSF).



The wider bar patterns have a lower frequency (fewer line pairs per mm) and the narrow bar patterns have a higher frequency (more line pairs per mm).

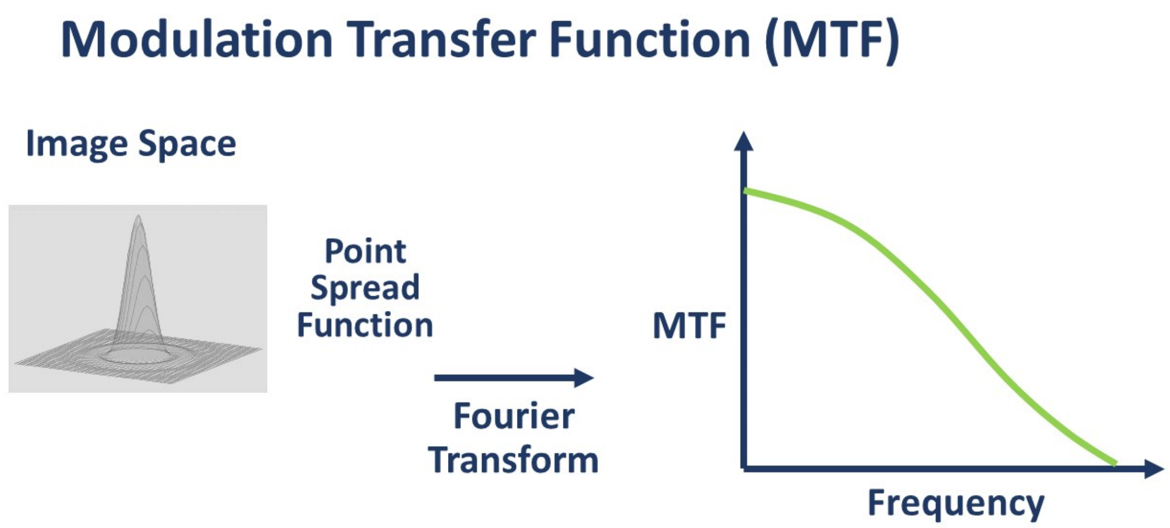
Modelling properties of imaging systems – the PSF and the MTF

The **modulation transfer function (MTF)** is the spatial frequency response of an imaging system or a component. It is the contrast at a given spatial frequency relative to low frequencies**.**
MTF = output signal amplitude / input signal amplitude



Modelling properties of imaging systems – the PSF and the MTF

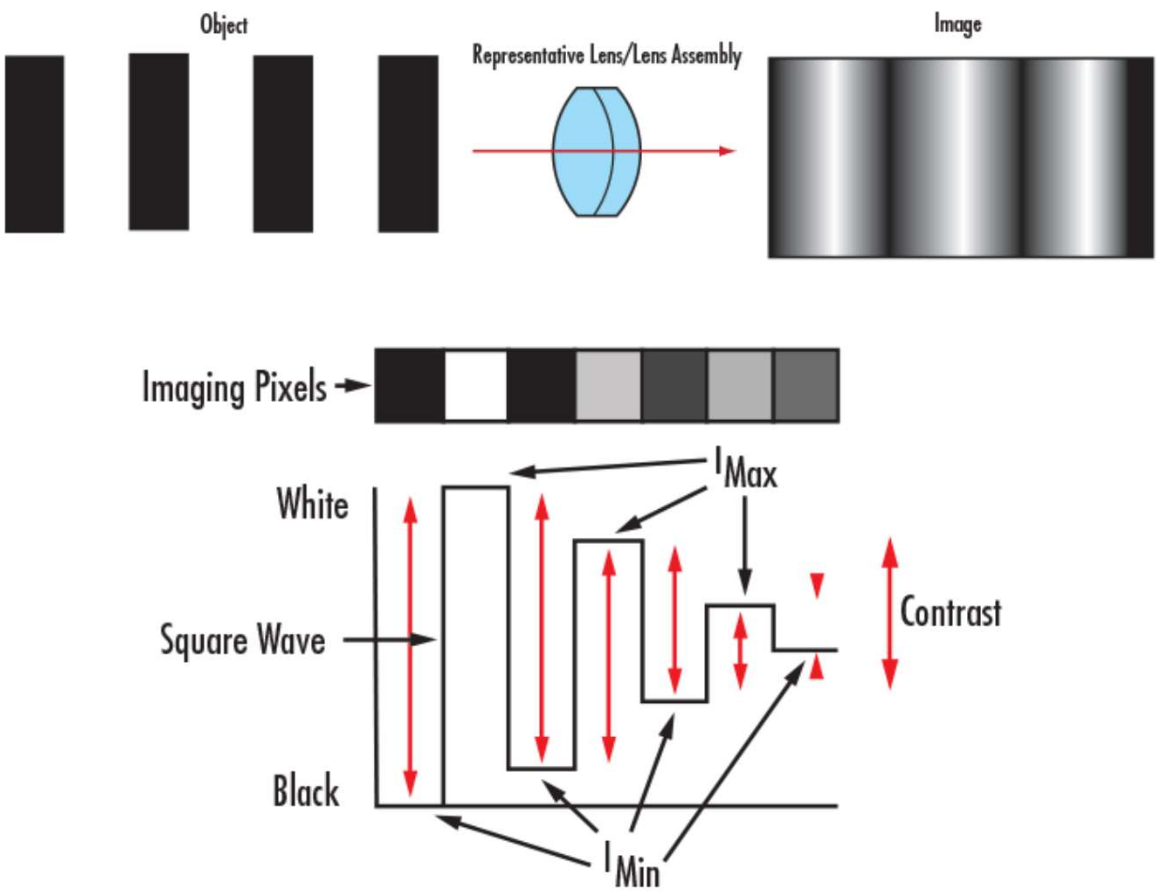
The point spread function (PSF) is a blurring function that is measured in the image space. The PSF is often assumed to be symmetric. If we take the Fourier Transform of that point spread function, we get the MTF or the Modulation Transfer Function.
This is the same Fourier transform that is used in MRI imaging where the data is collected in Frequency space and the Fourier transform is used to convert to image space (image reconstruction).



Modelling properties of imaging systems – the PSF and the MTF

- On the radiogram, objects having different sizes and opacity are displayed with different gray-scale values. MTF is responsible for converting contrast values of different-sized objects (object contrast) into contrast intensity levels in the image (image contrast).
- MTF is a useful measure of true or effective resolution, since it accounts for the amount of blur and contrast over a range of spatial frequencies.
- At higher spatial frequency MTF falls towards 0 , this corresponds to the poor visibility of the small structures. Conversely, at lower spatial frequency MTF is closer to 1 and it represents the ability to clearly visualize large structures.
- MTF would be affected at a given spatial frequency as follows:
  - Movement unsharpness will degrade the MTF
  - Increasing the size of the focal spot will degrade the MTF (due to geometric reasons)
  - Magnification will degrade the MTF (due to geometric reasons)
  - Screens degrade the MTF because of the speed of light in the screen
  - Screens have a lower MTF than film
  - Fast screen-films have lower MTF than slower screen-film

Modelling properties of imaging systems – the PSF and the MTF



The MTF of a lens, as the name implies, is a measurement of its ability to transfer contrast at a particular resolution from the object to the image. In other words, MTF is a way to incorporate resolution and contrast into a single specification. As line spacing decreases (i.e. the frequency increases) on the test target, it becomes increasingly difficult for the lens to efficiently transfer this decrease in contrast; as result, MTF decreases

Modelling properties of imaging systems – the PSF and the MTF

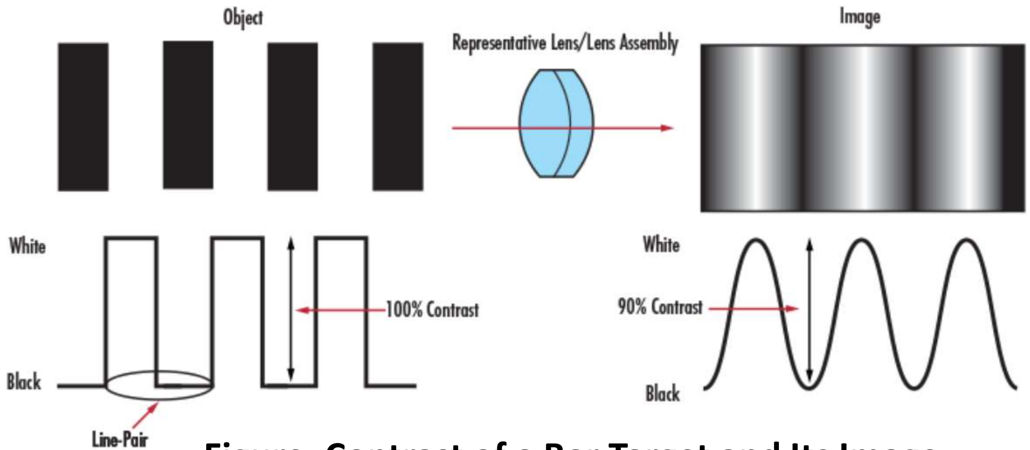


Figure: Contrast of a Bar Target and Its Image

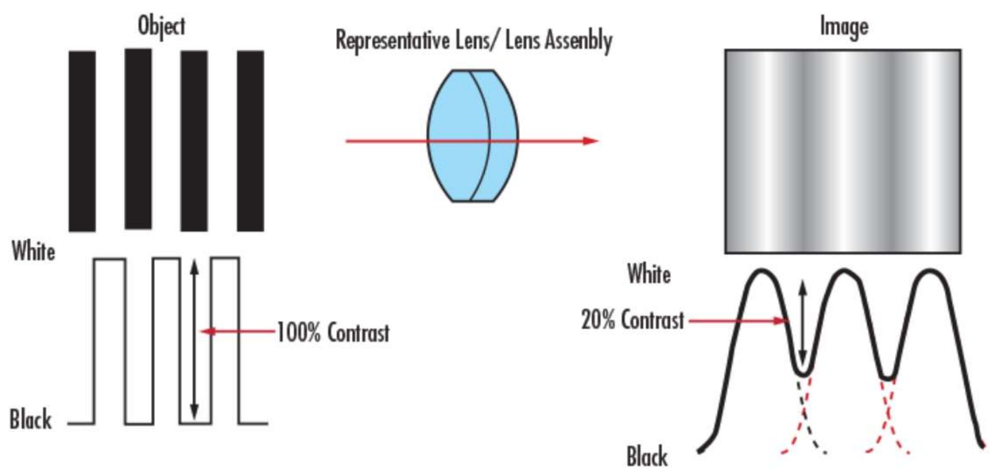


Figure : Contrast Comparison at Object and Image Planes

OTHER TRANSFORMS
The Hough transform

- The Hough transform is a transformation that can be used for any geometrical shape in a binary image that can be represented in a parametric form.
- A classical example for a parametric representation is, for instance, the circle.
- A circle is defined by the fact that all of its points have the same distance r from the center of the circle (M_x,M_y).
- A parametric representation of a circle in a Cartesian coordinate system i

$$\sqrt{(x - M_x)^2 + (y - M_y)^2} = r.$$

- Center coordinates M_x,M_y and radius r are the parameters of the circle.

The Hough transform

- As a parametric representation of a straight line we choose the Hesse normal form.
- It is simply given by the normal vector of the line which intersects with the origin of a Cartesian coordinate system.
- The parameters are the polar coordinates of this normal vector n.
- The Hough transform inspects every non-zero pixel in a binary image and computes the polar coordinates of its pixel, assuming that the pixel is part of a line.
- If the pixel is indeed part of a line, the transform to polar coordinates will produce many similar parameter pairs.

The Hough transform

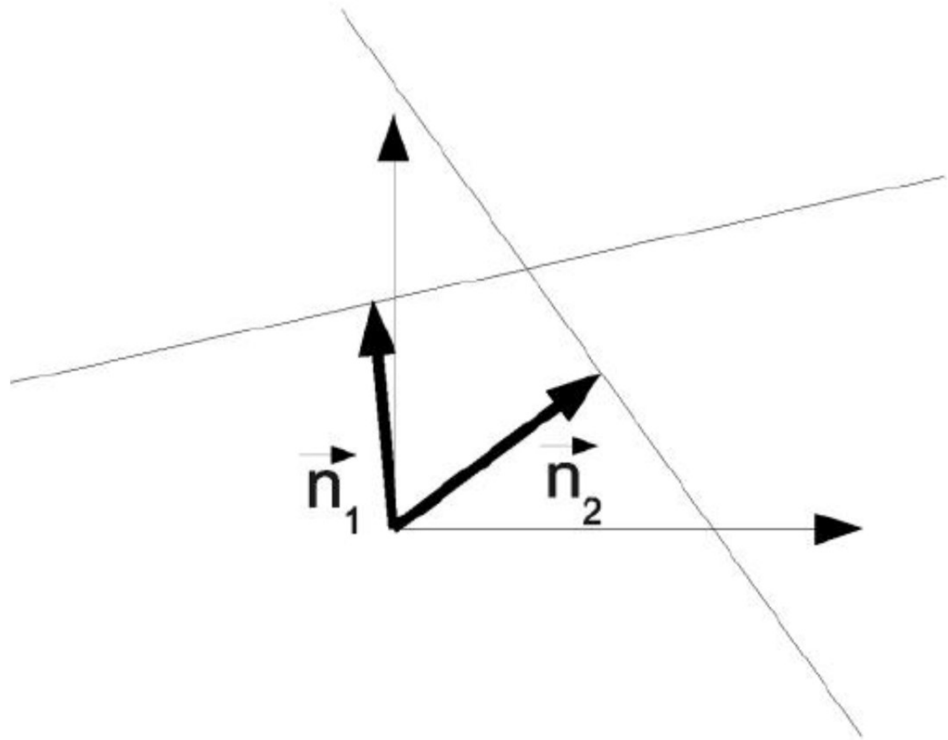


FIGURE: The Hesse normal form representation of lines in a Cartesian coordinate system. Each line is represented by a normal vector ~n that intersects the origin of the coordinate system. The polar coordinates of the intersection point of ~n and the line are the parameters of the line.

The Hough transform

- The trick in the case of the Hough transform is to plot the parameters of the shape in a new coordinate system which is spanned by the two parameters of the normal form, the length r of the normal vector $\underset{n}{\rightarrow}$ and the angle ϕ enclosed by $\underset{n}{\rightarrow}$ and the x-axis.
- Pixels belonging to a line will occupy the same location in Hough-space.
- Since the Hough transform acts only on binary images, the varying brightness of points in Hough-space is a measure of the number of pixels with the same normal vector.

The Hough transform

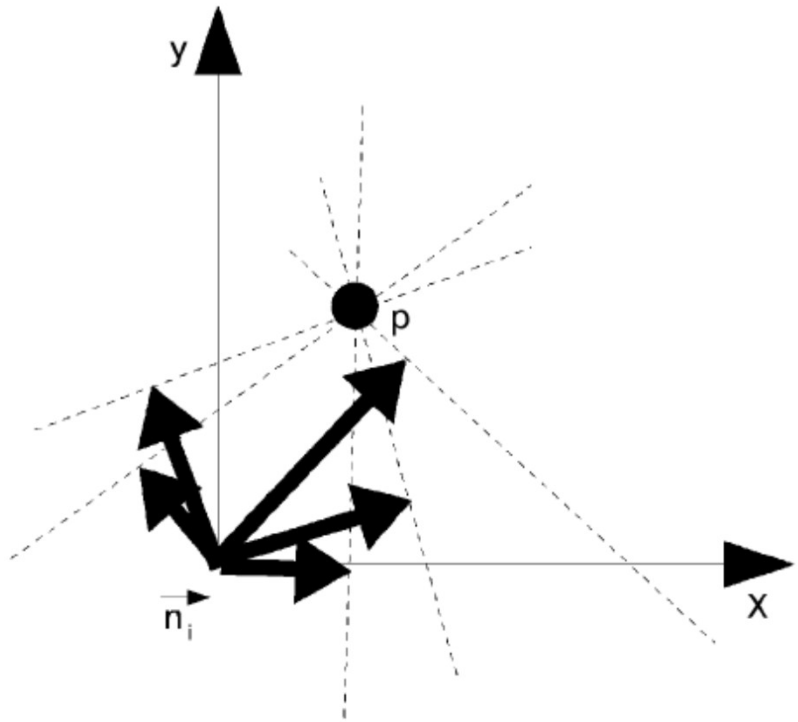
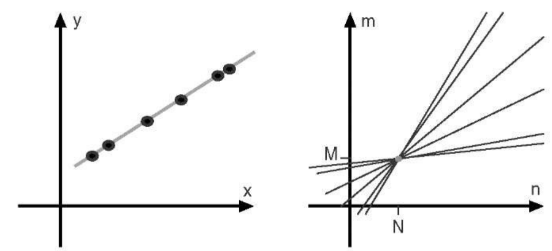


FIGURE: The basic principle of the Hough – transform is to store the parametric representation of the normal vectors $\vec{n}$ of lines that may pass a non-zero pixel p. If a series of pixels lies on a line, the location of the parameters for the normal vector of this very line will be densely populated. The parameters for less populated normal vectors are being disregarded prior to back-transformation.

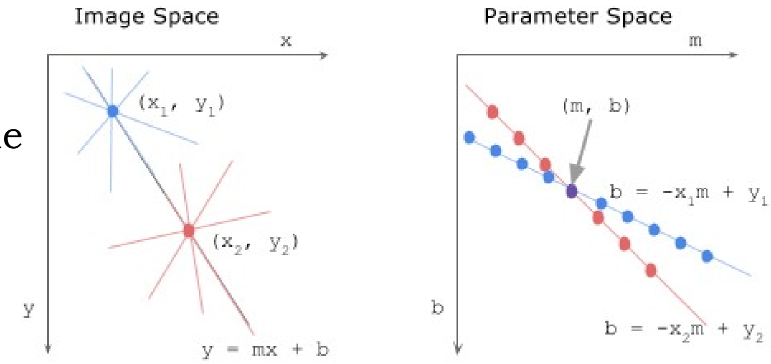
The Hough transform

More points that on same line tends to more lines in Hough domain and that will increase voting to the intersection point indicating that there is many points belongs to line in image domain with that slope and y-intercept.

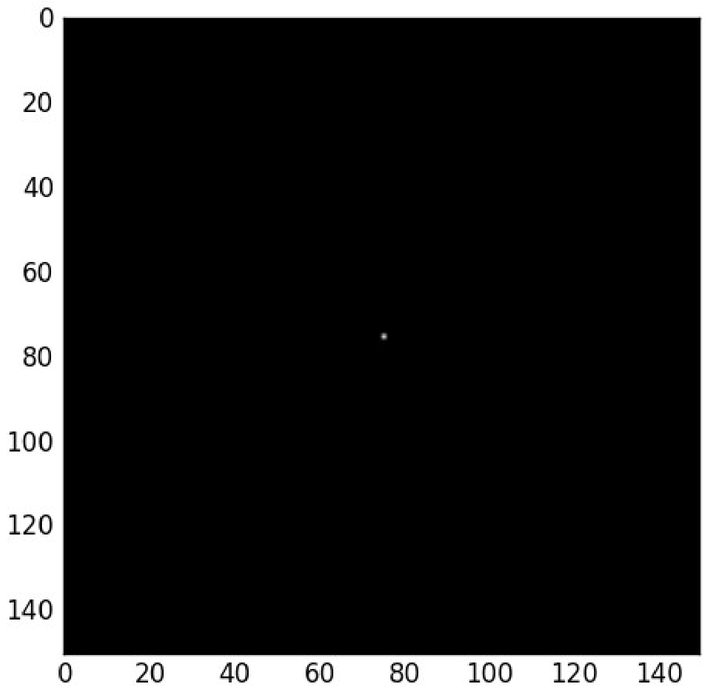


Blue point in image domain was mapped to the blue line in Hough domain. The same for red point. Intersection point of blue and red lines in hough domain has the values m and b of the line

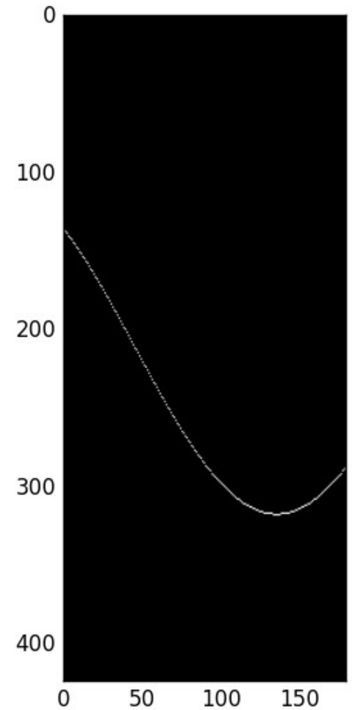
$$y=mx+b$$



The Hough transform



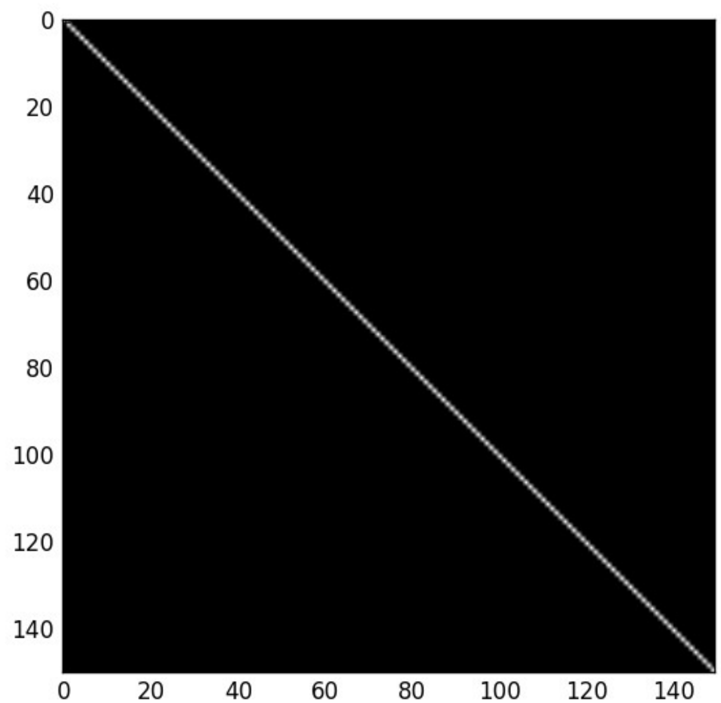
Original Image



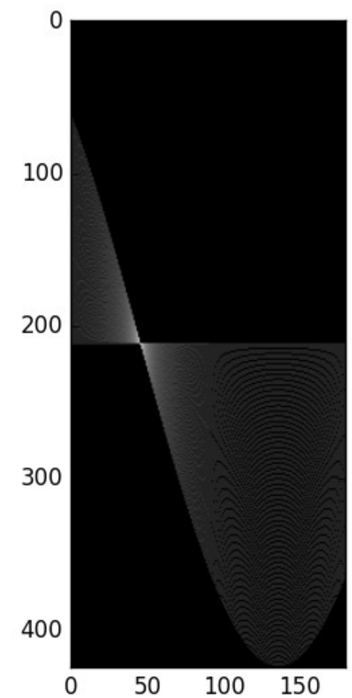
Hough Transform

the edge point is mapped to a curve in hough domain

The Hough transform



Original Image



Hough Transform

All curves in hHough domain have only one intersection point, so there is only one line in the image domain

The Hough transform

FIGURE: The binary image which will undergo a Hough-transform in Example The representation in Hough-space can be found in Figure next

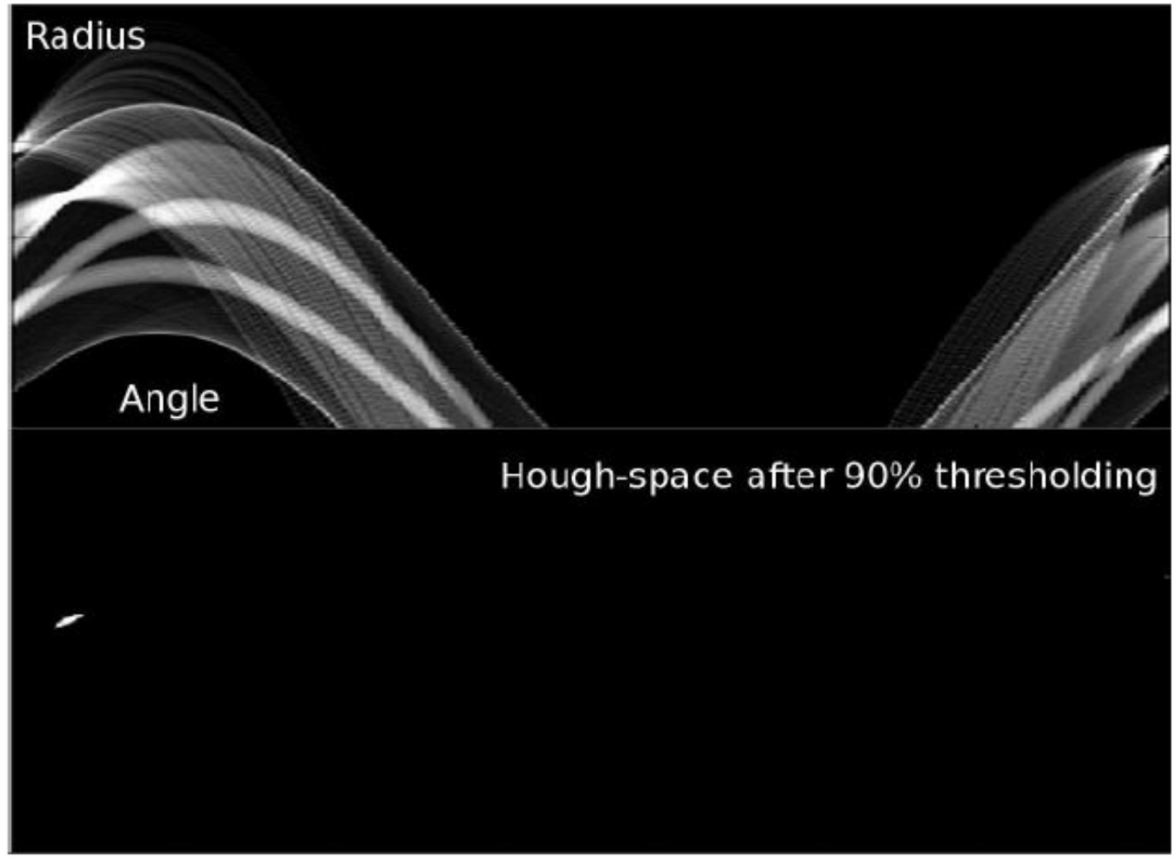


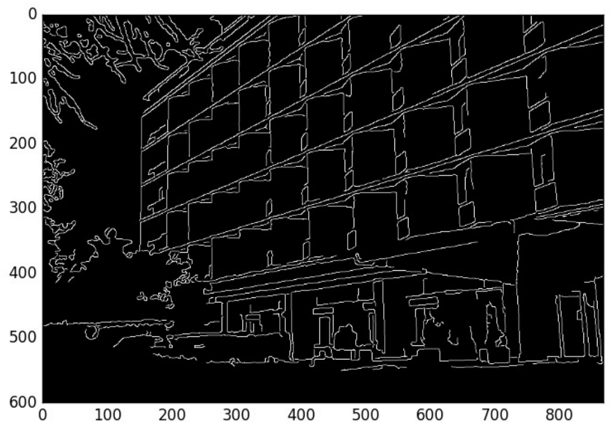
FIGURE: An intermediate result of Hough_5.m. The binary image is transformed to Hough-space; after thresholding the resulting image, only the areas with the highest concentration of parameter pairs remain. In this representation, the abscissa represents the angles θ, and the ordinate is the parameter r.

The Hough transform

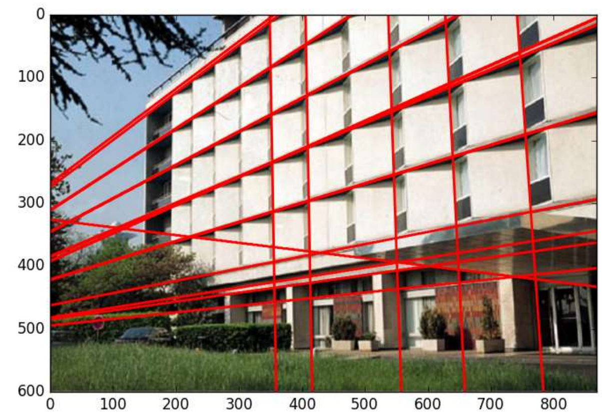
- The pixels that lie on a line appear as bright spots in the image. By setting all pixels below a given threshold to zero, only the bright spots in the parameter image remain.
- Next, one can define a so-called accumulator cell; this is a square in parameter space which assigns a single parameter pair to the area in the parameter space covered by the accumulator, thus narrowing down the number of possible parameters.
- While the detection of lines is a classical example of the Hough-transform, the principle is applicable to every shape that can be modeled in a parameter representation.
- In medicine, the Hough transform can also be very helpful in identifying structures in segmented images, for instance when identifying spherical markers in x-ray images.

The Hough transform

original image

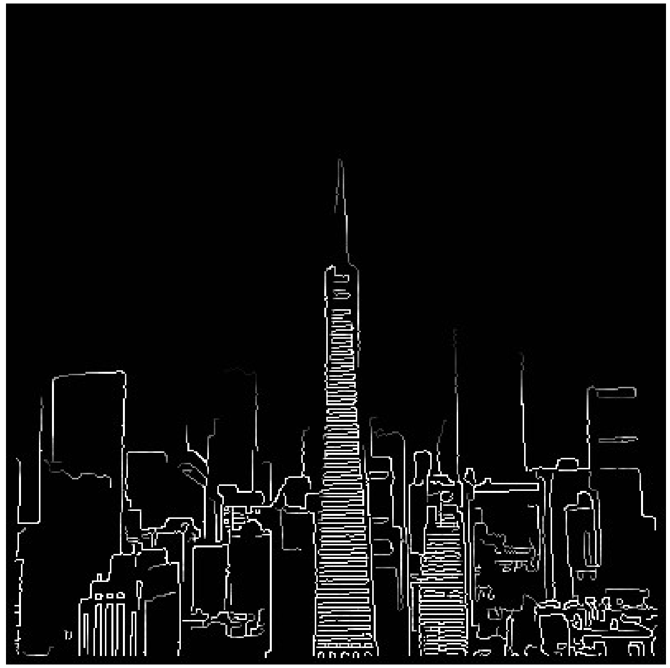


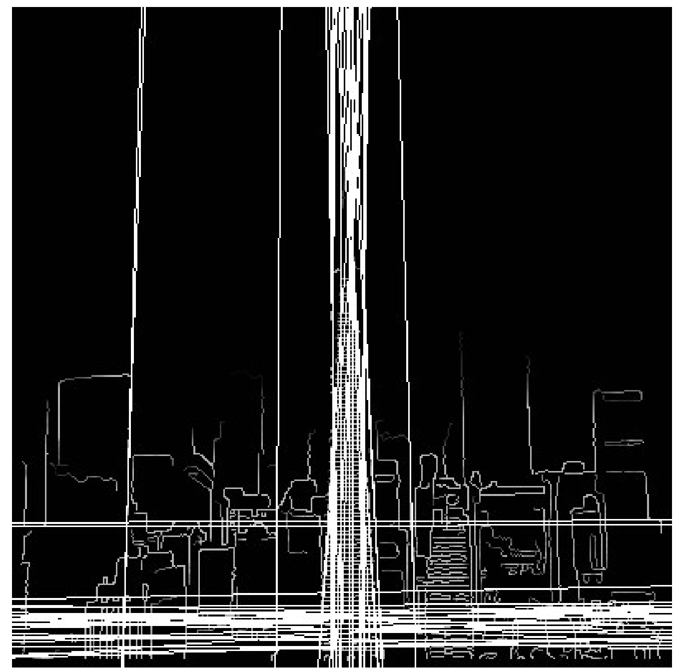
canny image



final image after detecting lines using Hough transform

The Hough transform



The distance transform

- It also operates on binary images; here, the nearest black pixel for each non-black pixel is sought.
- The value of the non-black pixel is replaced by the actual distance of this pixel to the next black pixel.

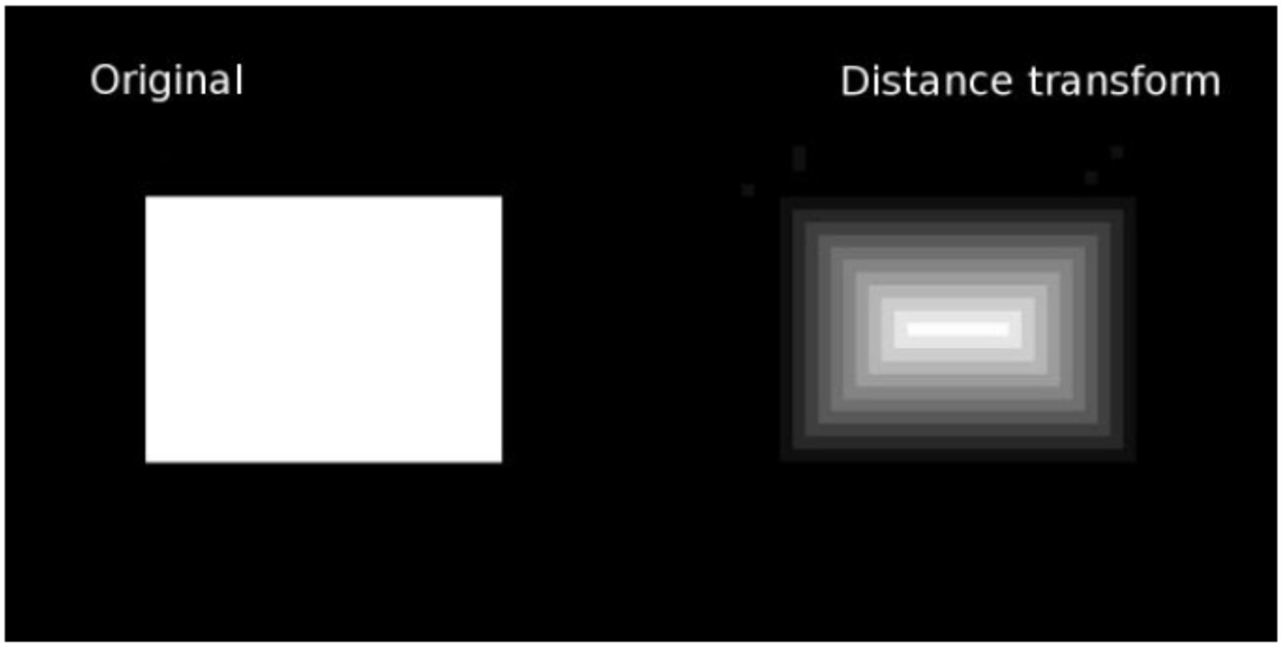


FIGURE: The result of the distance transform. The innermost pixels of the rectangle are those with the greatest separation to the border of the rectangle; therefore they appear as bright, whereas the pixels closer to the order become darker.

The distance transform

- The distance transform is an operator normally only applied to binary images. The result of the transform is a graylevel image that looks similar to the input image, except that the graylevel intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point.
- One way to think about the distance transform is to first imagine that foreground regions in the input binary image are made of some uniform slow burning inflammable material. Then consider simultaneously starting a fire at all points on the boundary of a foreground region and letting the fire burn its way into the interior. If we then label each point in the interior with the amount of time that the fire took to first reach that point, then we have effectively computed the distance transform of that region. Figure 1 shows a distance transform for a simple rectangular shape.

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

$\Longrightarrow$

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

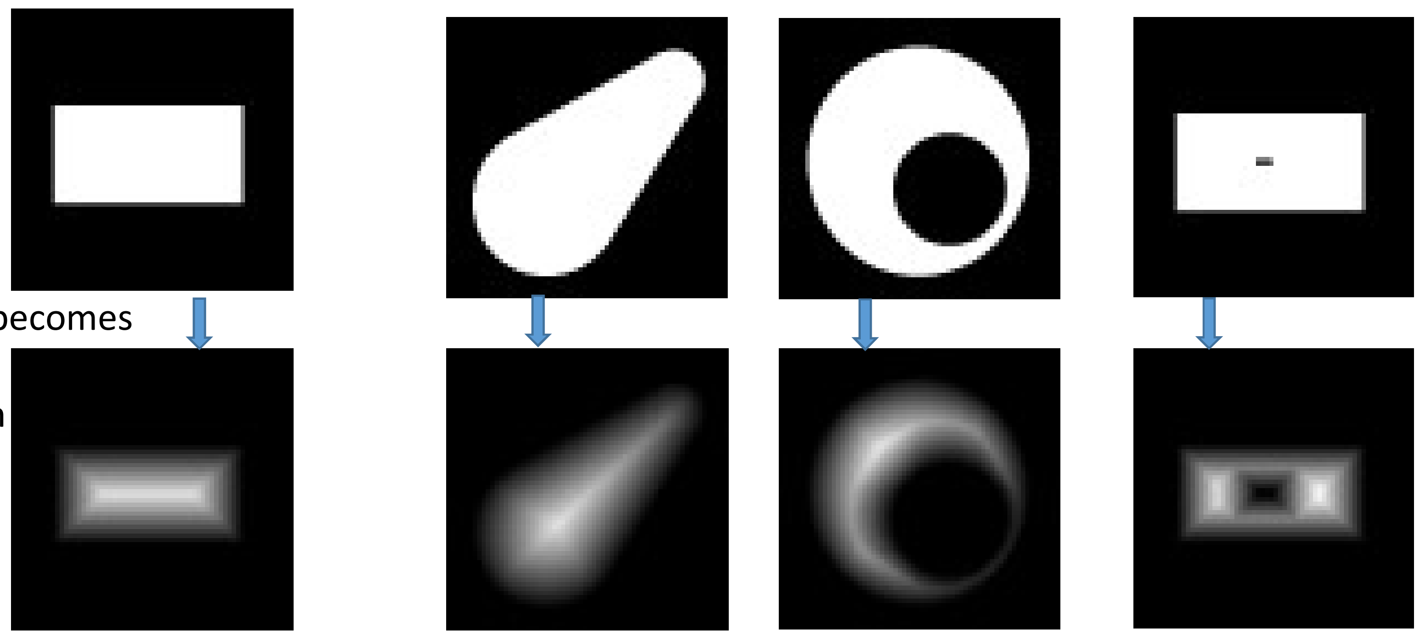
Figure The distance transform of a simple shape

The distance transform

The binary image

becomes

when a distance transform is applied



perform multiple successive erosions with a suitable structuring element until all foreground regions of the image have been eroded away

DISCRETIZATION – RESOLUTION AND ARTIFACTS

- A change in image size therefore introduces artifacts, which may be considered rounding errors. Figure gives an illustration.
- The original image is a PET-scan with 128×128 pixels in-plane resolution of a gynecological tumor located in the pelvis.
- The left part of the image was resized without interpolation – the single pixels appear blocky and enlarged.
- If we apply an interpolation procedure, which distributes the pixel intensities in a more sophisticated manner, we get the result on the right hand side.
- In short, enlarging the image consists of putting a finer mesh of image elements on the image data.
- If we don't distribute the image intensities in a smooth manner, a blocky appearance is inevitable.
- It is to be emphasized that interpolation does not add image information, but improves visualization of image data.
- Here, a rotation operation does not transfer all gray values from the original image to the rotated image due to discretization artifacts.

DISCRETIZATION – RESOLUTION AND ARTIFACTS

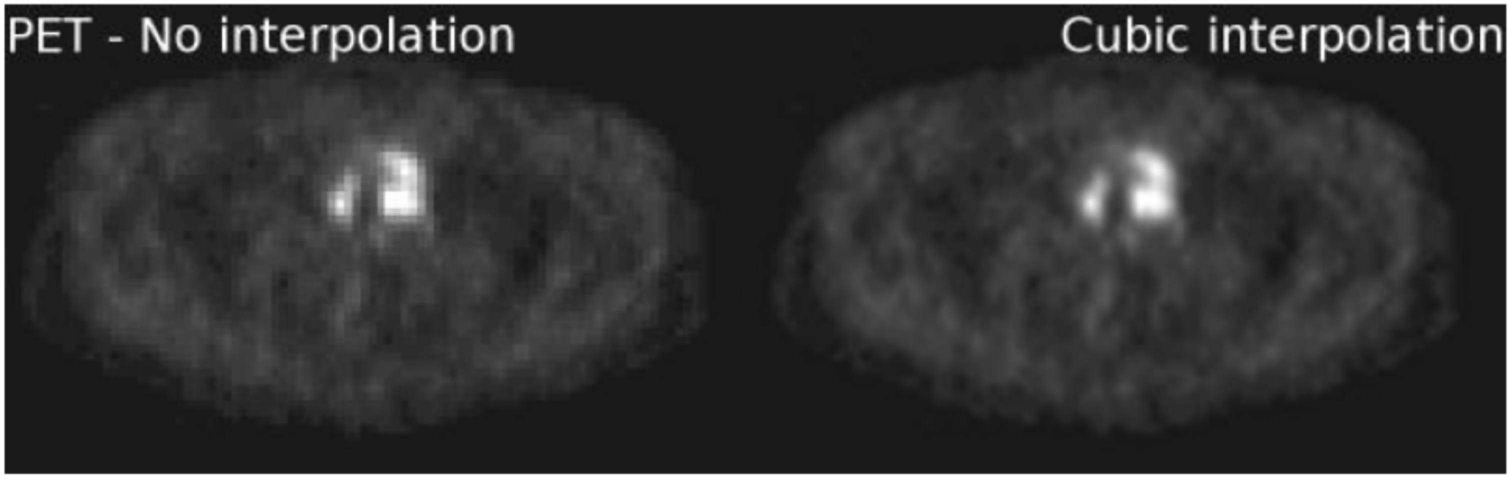


FIGURE: The effects of interpolation. The original image is an FDG PET slice of a tumor located in the lower abdomen. The sub-image of the slice shown here was resampled from 84× 50 pixels to 638×398 pixels, without and with interpolation. The left image fills the new pixels by the values from the nearest neighbor in the old image, whereas the right image distributes gray values ρ using a bicubic interpolation model. While there is no new information in the right image, visualization and interpretation is improved.

INTERPOLATION AND VOLUME REGULARIZATION

- Interpolation: the easiest way to interpolate between two discrete data pairs $(x_i, \rho_i)^T$ and $(x_j , \rho_j)^T$ is to draw a straight line.
- The general shape of a linear function is $\rho = \rho_0+ax$ where ρ_0 is the value that is connected to the doublet $(0, \rho_0)$T – this is the position on the ordinate where the straight line intersects it. a is the gradient of the function.
- If we interpolate, we can rewrite this function as

$$\rho = \rho_0 + \frac{\rho_1 - \rho_0}{x_1 - x_0} (x - x_0)$$

INTERPOLATION AND VOLUME REGULARIZATION

- The term bilinear interpolation refers to the generalization of this method to a function in two variables.
- Here, a linear interpolation is carried out in one direction, and then the other direction follows.

$$
\begin{aligned}
\rho_{x,y_0} &= \frac{x_1 - x}{x_1 - x_0}\rho_{00} + \frac{x - x_0}{x_1 - x_0}\rho_{10} \\
\rho_{x,y_1} &= \frac{x_1 - x}{x_1 - x_0}\rho_{01} + \frac{x - x_0}{x_1 - x_0}\rho_{11} \\
\rho_{x,y} &= \frac{y_1 - y}{y_1 - y_0}\rho_{x,y_0} + \frac{y - y_0}{y_1 - y_0}\rho_{x,y_1}
\end{aligned}
$$

We interpolate first in the x-direction, starting from coordinates y_0 and y_1. The final value is given by an interpolation in ydirection between the previously interpolated values.
Simplify the whole thing by assuming the difference between pixel positions $x1 - x0$ and $y1 - y0$ to be one.

INTERPOLATION AND VOLUME REGULARIZATION

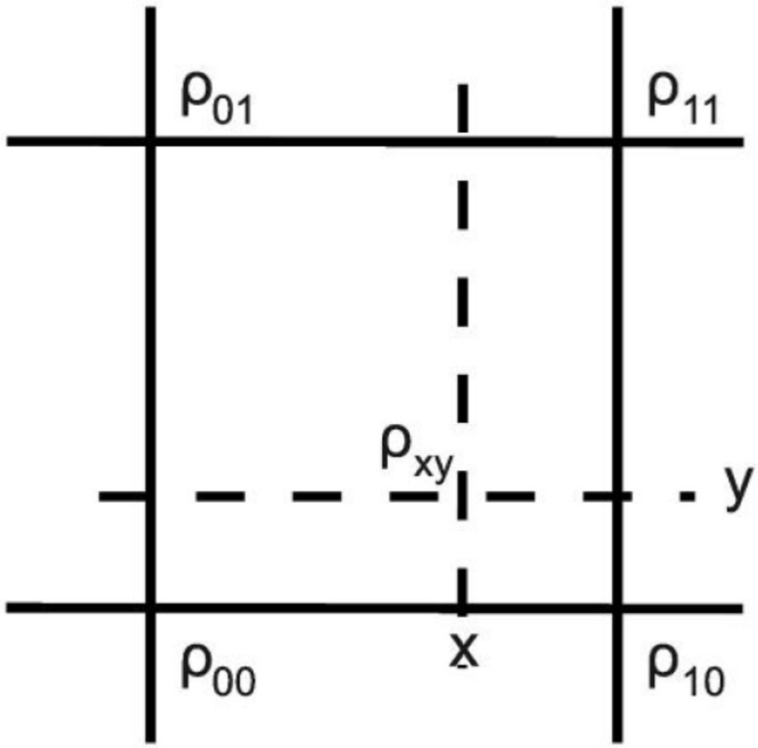


FIGURE: The basic principle of bilinear interpolation on a regular grid.
Four nodes with known function values $\rho_{00} \ldots \rho_{11}$ are given. The function value ρ_{xy} on an arbitrary position $(x, y)^T$ is estimated by performing a linear interpolation in both directions.

INTERPOLATION AND VOLUME REGULARIZATION

$$\begin{aligned}
\rho_{x,y_0} &= \frac{x_1 - x}{x_1 - x_0}\rho_{00} + \frac{x - x_0}{x_1 - x_0}\rho_{10} \\
\rho_{x,y_1} &= \frac{x_1 - x}{x_1 - x_0}\rho_{01} + \frac{x - x_0}{x_1 - x_0}\rho_{11} \\
\rho_{x,y} &= \frac{y_1 - y}{y_1 - y_0}\rho_{x,y_0} + \frac{y - y_0}{y_1 - y_0}\rho_{x,y_1}
\end{aligned}$$

therefore simplifies to

$$\begin{aligned}
\rho_{x,y} = \; & (y_1 - y)\left((x_1 - x)\,\rho_{00} + (x - x_0)\,\rho_{10}\right) + \\
& (y - y_0)\left((x_1 - x)\,\rho_{01} + (x - x_0)\,\rho_{11}\right).
\end{aligned}$$

Equation can be written as a product of vectors and matrices

if we assume x_0 and y_0 to be zero and x_1 and y_1 to be one:

$$\rho_{xy} = \begin{pmatrix} 1 - y \\ y \end{pmatrix}^T \begin{pmatrix} \rho_{00} & \rho_{10} \\ \rho_{01} & \rho_{11} \end{pmatrix} \begin{pmatrix} 1 - x \\ x \end{pmatrix}$$

INTERPOLATION AND VOLUME REGULARIZATION

- Subdivide the original pixel into smaller units, and the gray value of these smaller pixels is computed from the four gray values $\rho_{00} \dots \rho_{11}$.
- Expand the bilinear interpolation to the 3D-domain.
- In such a case, the interpolated values for ρ_{xyz0} and ρ_{xyz1} are computed for the corresponding interpolated in-slice pixels for neighboring slices.
- Finally, a linear interpolation in the intra-slice z-direction is carried out.

$$\rho_{xyz} = \begin{pmatrix} (1-x)(1-y)(1-z) \\ (1-x)(1-y)z \\ (1-x)y(1-z) \\ (1-x)yz \\ x(1-y)(1-z) \\ xy(1-z) \\ x(1-y)z \\ xyz \end{pmatrix}^T \begin{pmatrix} \rho_{000} \\ \rho_{001} \\ \rho_{010} \\ \rho_{011} \\ \rho_{100} \\ \rho_{110} \\ \rho_{101} \\ \rho_{111} \end{pmatrix}$$

INTERPOLATION AND VOLUME REGULARIZATION

- The position of the eight gray values $\rho_{000} \ldots \rho_{111}$ can be found in Figure

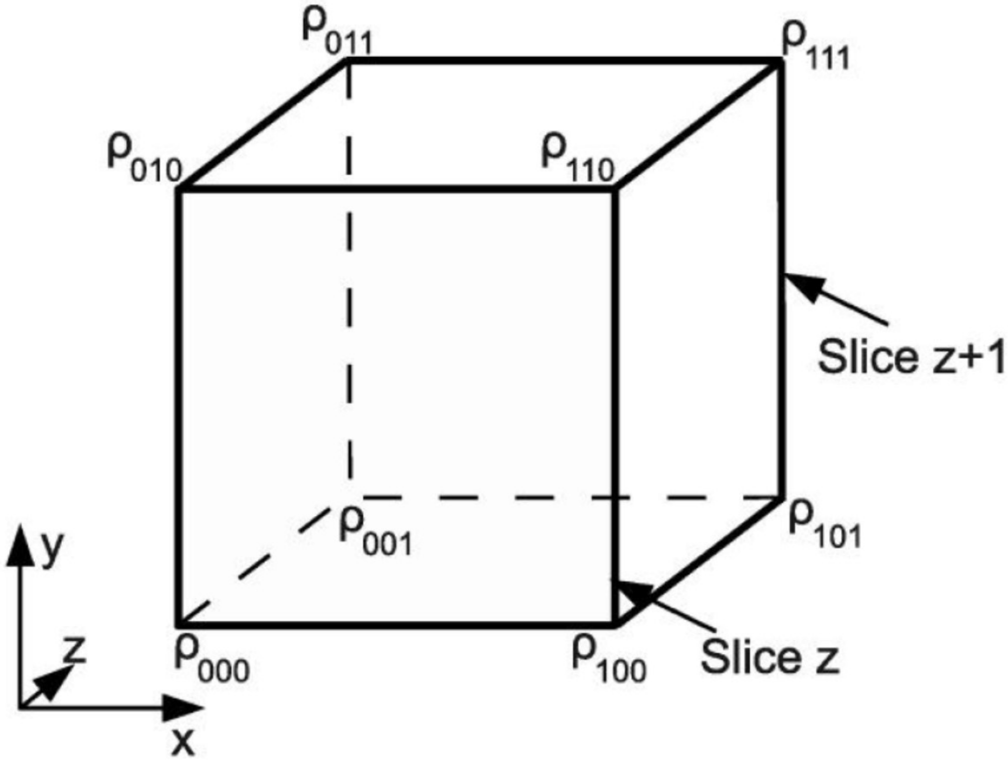


FIGURE : The conventions for performing a trilinear interpolation according to Equation

INTERPOLATION AND VOLUME REGULARIZATION

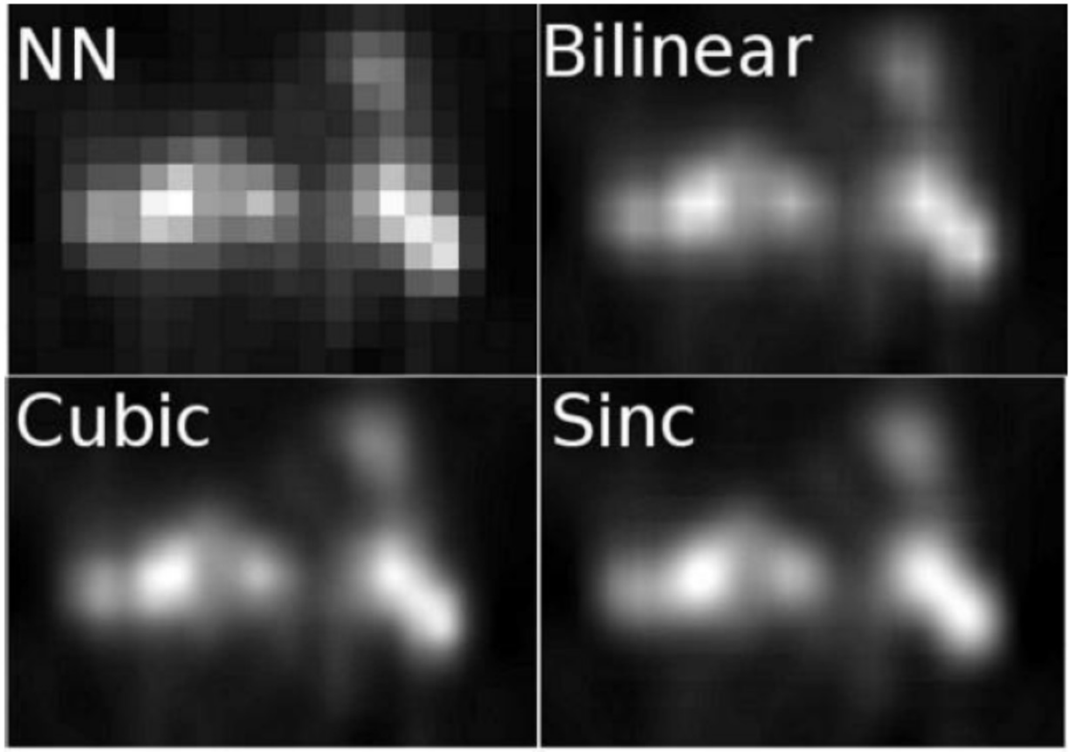
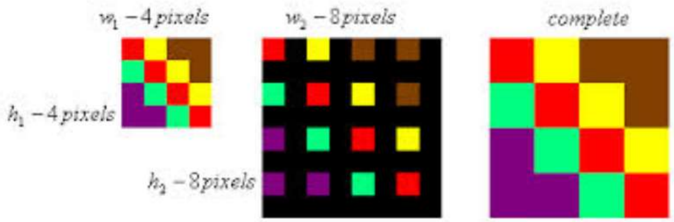


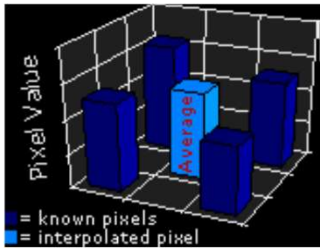
FIGURE: Different interpolation techniques, presented on a subimage from the PET image. The image was magnified fifteen times; at such a magnification, differences in interpolation models are evident. The image on the outer left was not interpolated; what follows are images that underwent bilinear interpolation, cubic interpolation, and interpolation using a sinc-function. All images were generated using GIMP. The image of the bilinear interpolation also shows an important property of this method: despite its name, bilinear interpolation is not a linear operation – it is the product of two linear operations. Therefore it is only linear in the direction of the coordinate axes; this results in the starshape artifacts visible in the example.

INTERPOLATION AND VOLUME REGULARIZATION

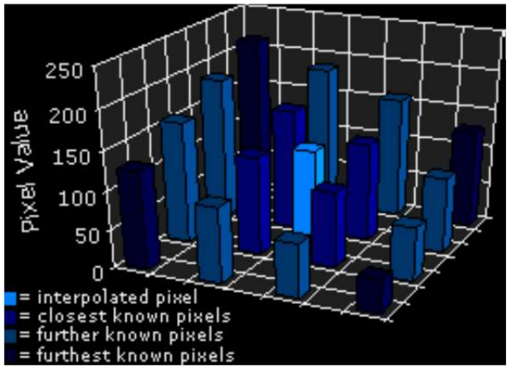
Nearest neighbor



Bilinear



Bicubic



TRANSLATION AND ROTATION

Rotation in 2D – some properties of the rotation matrix

- When rotating a single point given by $\vec{x} = (x, y)^T$, it is a useful approach to use a matrix R which is applied to this vector – the result is a new vector $\vec{x}'$.
- Such a matrix, which rotates a point in 2D by an angle φ is given by

$$R = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}$$

- The transformation of our 2D-point $\vec{x}$ can now be written as

$$\vec{x}' = R\vec{x}$$

Rotation in 2D – some properties of the rotation matrix

- Multiplication of matrices is another one of those basics; in order to multiply a matrix such as R from Equation with a vector $\underset{x}{\rightarrow}$, we have to form the inner products of the matrix rows with the vector, and each of those inner products gives a component of the resulting vector.
- Inner product can be computed as a sum of products of the corresponding vector elements. If we perform the matrix multiplication in equation, we get

$$\vec{x}' = \begin{pmatrix} \cos(\phi)x - \sin(\phi)y \\ \sin(\phi)x + \cos(\phi)y \end{pmatrix}$$

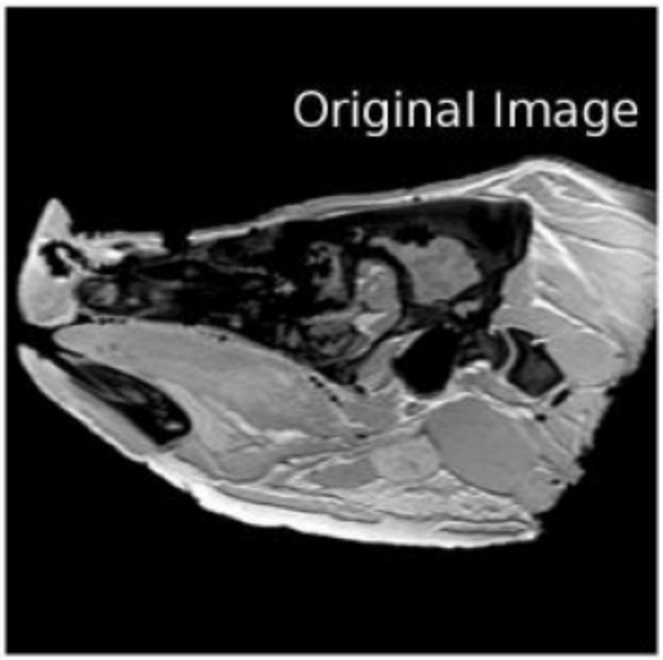
- That is the only time we will fully write down the transform of a position $\underset{x}{\rightarrow}$ to another position.
- When dealing with more complicated spatial transformations, we would drown in chaos if we would continue to deal with sines and cosines.

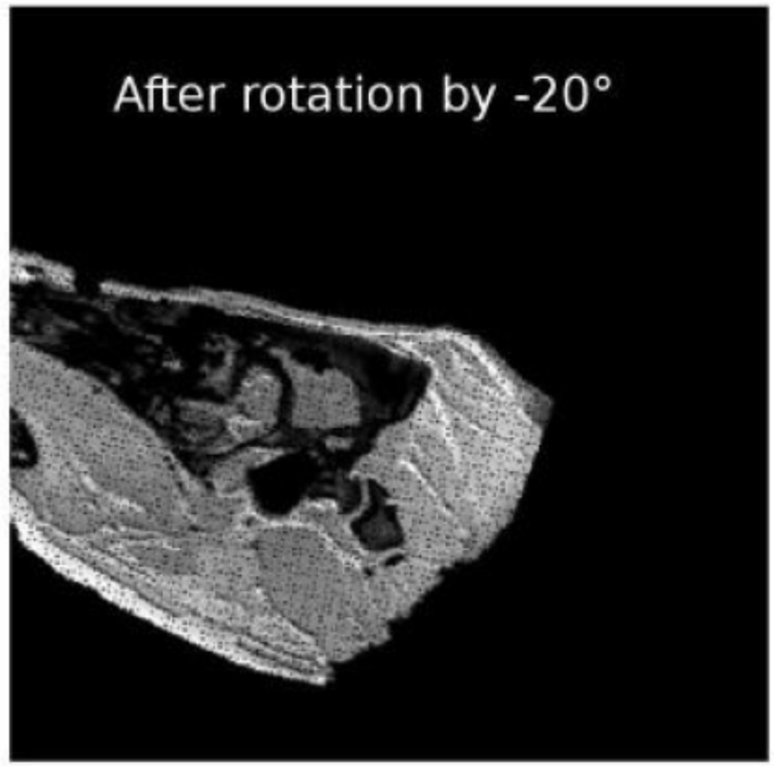
Rotation in 2D – some properties of the rotation matrix

- When multiplying two matrices, the number of matrix columns has to match the rows of the other matrix – otherwise, the matrix product cannot be computed. The components of the resulting matrix are given as the inner products of the respective rows and columns of the matrices.
- Matrix multiplication is not commutative.
- Matrix multiplication is associative.
- The inverse only exists if A is not singular. A matrix is singular if its determinant is zero.
- Multiplication in MATLAB or Octave is always matrix multiplication.
- For rotation matrices, we will use radians as the unit of choice for specifying angles
- Rotation matrices have two special properties. First, a rotation transformation does not change the length of a vector. Therefore, its determinant is always 1. The other property of the rotation matrix is that it can easily be inverted.
- We usually carry out matrix multiplication from left to right.

Rotation in 2D – some properties of the rotation matrix

- If we want to rotate an image by an angle ϕ, all we have to do is apply Equation $\vec{x}' = R\vec{x}$ to every single pixel position $\vec{x}$, and to store its gray value ρ at the new position $\vec{x}'$





Rotation in 2D – some properties of the rotation matrix

- The second operation which allows for changing the location of pixels in an image – translation.
- Compared to rotation, translation – the shifting of image content by a constant vector $\vec{\Delta r} = (\Delta x \; \Delta y)^T$ is simple. We just add this vector to every pixel location $\underset{x}{\rightarrow}$ by adding $\vec{\Delta r}$.
- Applying a translation before applying the rotation gives a different result compared to translation followed by rotation.
- We can also choose the center of rotation by a translation; when rotating an image,

Rotation in 2D – some properties of the rotation matrix

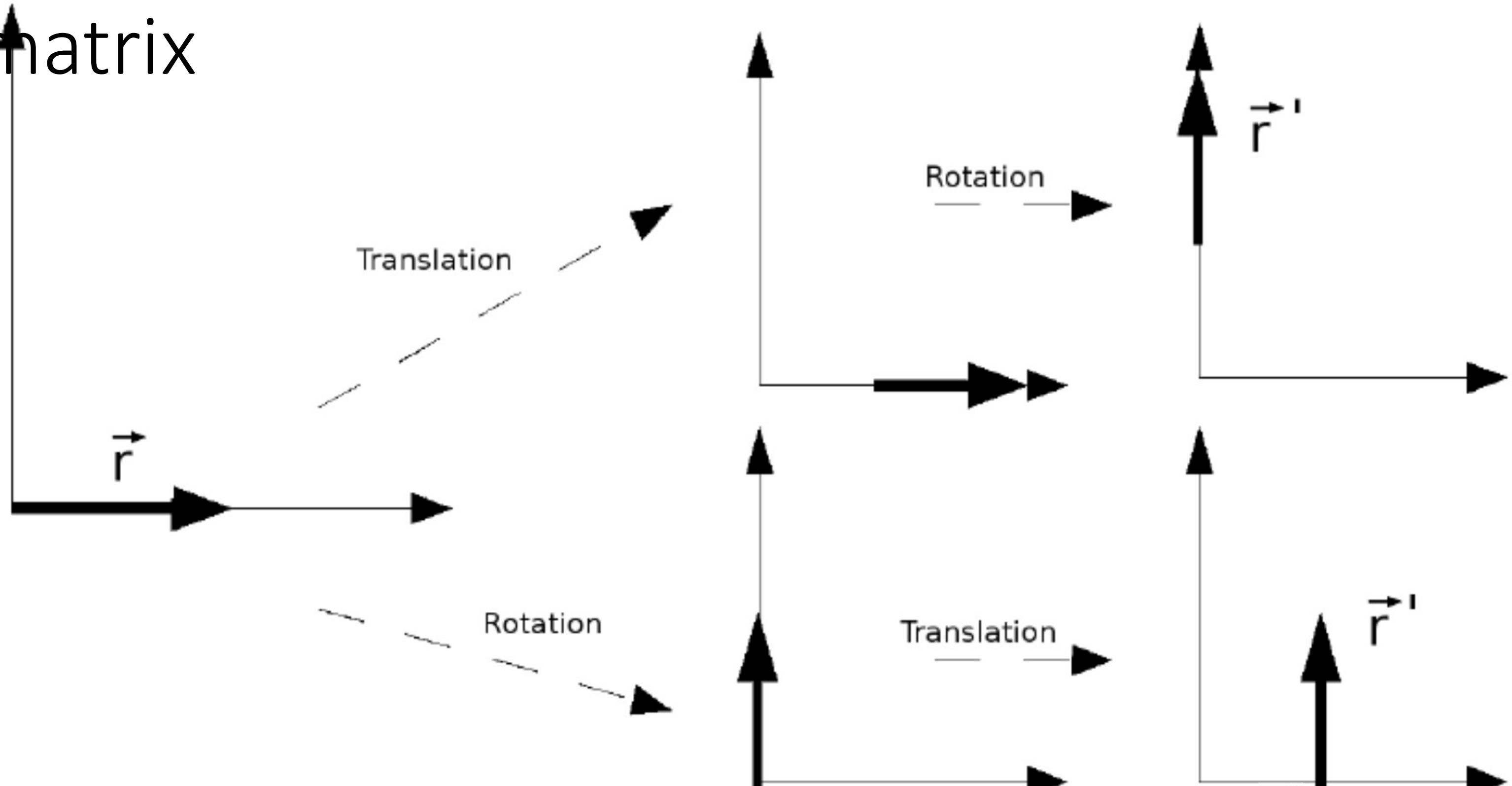


FIGURE: The effects of a combined rotation and translation on a simple vector $\vec{r}$ located on the abscissa of a Cartesian coordinate system. If translation by one unit is applied prior to rotation by 90° counterclockwise, the upper position for the transformed vector $\vec{r}$ is the outcome. Otherwise, the lower situation is the result. Translation and rotation do therefore not commute.

Rotation in 2D – some properties of the rotation matrix

Move the center of the image (or the centroid of a ROI as given in Equation) to the origin of the coordinate system; if the image dimensions are width W and height H, this translation vector is given as $\vec{\Delta c} = -\left(\frac{W}{2}, \frac{H}{2}\right)^T$.

Next, the shifted coordinates are rotated.

Finally, the rotated coordinates are shifted back to the image domain by a translation $-\vec{\Delta c}$. This is actually the inverse transform of shifting the pixel coordinates to the image origin.

Rotation and translation in 3D

- A medical imaging is three dimensional in many cases.
- We have three degree of freedom (dof) in rotation and three dof in translation; apparently, life gets more complicated if you add more dof.
- The handling of rotations is not that much different from the 2D case.
- The Euler angles which act as parameters for the rotation around the axes of a 3D Cartesian coordinate system.
- The rotation matrix that rotates a point in space around the x-axis looks like this:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_x) & -\sin(\phi_x) \\ 0 & \sin(\phi_x) & \cos(\phi_x) \end{pmatrix}$$

Rotation and translation in 3D

- A Gaussian stays a Gaussian when applying the Fourier transform.
- It changes its width, but not its shape.
- In special cases, it is even an eigenfunction.
- A matrix is a transformation applied to a vector with a finite number of elements.
- The eigenvector retains its direction, but it may be stretched by a factor λ that represents the eigenvalue.
- If we want to find the eigenvectors and eigenvalues of a matrix, we have to solve the following linear system of equations: $R_x\vec{x} = \lambda\vec{x}.$
- We can make sure by computing the transpose and the determinant of Rx that it is a rotation indeed.
- Therefore, the eigenvalue λ has to be one since a rotation cannot change a vector's length. Without further math, we can easily verify that every vector

$$\vec{x} = (x_i,\, 0,\, 0)^T$$

that is, all vectors collinear to the x-axis – is an eigenvector of R_x.

Rotation and translation in 3D

- The corresponding matrices for rotation around the y- and z- axis are:

$$R_y = \begin{pmatrix} \cos(\phi_y) & 0 & -\sin(\phi_y) \\ 0 & 1 & 0 \\ \sin(\phi_y) & 0 & \cos(\phi_y) \end{pmatrix}$$

and

$$R_z = \begin{pmatrix} \cos(\phi_z) & -\sin(\phi_z) & 0 \\ \sin(\phi_z) & \cos(\phi_z) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation and translation in 3D

- By using Rx, Ry, and Rz, we can now compose rotations in 3D by successively applying these three matrices.
- An arbitrary rotation of a point in 3D - space with coordinates $\vec{x}$ is simply given as $\vec{x}' = R\vec{x}$, where R is a combination of the three matrices Ri.
- The only problem lies in the fact that matrix multiplication is not commutative, therefore the product $R_{xyz} = R_x R_y R_z$ gives a completely different rotation than $R_{zyx} = R_z R_y R_x$.
- If we provide Euler-angles as parameters for 3D rotation, we also have to define a convention on the order of applying these angles.
- Euler angles are handy since they can be easily interpreted, but they can cause terrible chaos when handling them.
- In short, we will be acquainted with a different parameter set that is not very demonstrative but extremely powerful.

Rotation and translation in 3D

- Let's take a look at the translation problem; shifting positions by a vector $\vec{\Delta s}$ using a translation matrix T should look like this:

$$T\vec{x} = (x_1 + \Delta s_1,\, x_2 + \Delta s_2,\, x_3 + \Delta s_3)^T$$

- We can augment the 3× 3 matrix T to a 4× 4 matrix by adding a row and a column; the same is done with the vector $\vec{x}$. the fourth element of $\vec{x}$ is said to be one, and the added row and column of the augmented matrix T contains zeros besides the main diagonal element t_{44}, which is also set to 1.
- The translation operation now looks like this:

$$T\vec{x} = \begin{pmatrix} 1 & 0 & 0 & \Delta s_1 \\ 0 & 1 & 0 & \Delta s_2 \\ 0 & 0 & 1 & \Delta s_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + \Delta s_1 \\ y + \Delta s_2 \\ z + \Delta s_3 \\ 1 \end{pmatrix}$$

- At the cost of augmenting the translation operator T , we do have a matrix that gives us a translation if we only look at the first three components of the resulting vector

Rotation and translation in 3D

- Augment the rotation matrices. We therefore get

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi_x) & -\sin(\phi_x) & 0 \\ 0 & \sin(\phi_x) & \cos(\phi_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- and similar matrices Ry and Rz; again, all we did was an augmentation by adding an additional row and column with all elements besides the diagonal element equal zero.
- A little problem, however, arises; if we add additional matrices like a perspective projection operator the fourth element of the resulting transformed position will usually not be 1.
- In such a case, we have to renormalize the result of our operation by multiplying all four elements of the resulting 4 × 1 vector with the inverse of the fourth element.
- This representation of the spatial transforms is called homogeneous coordinates

Rotation and translation in 3D

- Another important operation can also be carried out using homogeneous coordinates – scaling of coordinates, which is specially important for medical imaging since we usually encounter volumes where the in-plane resolution differs from the slice spacing.
- If we want to scale voxel positions, we can use the scaling matrix S:

$$S = \begin{pmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 \\ 0 & 0 & \sigma_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- where σ_i gives the scaling factor in each dimension.

Rotation and translation in 3D

- we can use a product of matrices for all rigid-body transformations in 3D – that is, all transformations that map each voxel position in the same manner.
- These are also called affine transformations. This matrix, which we may call volume transformation matrix V can be composed as $V = TR_xR_yR_zS$ or $V = R_zR_yR_xTS$, or in another sequence.
- If one chooses to scale the coordinates using a scaling matrix S, this one should always be carried out first, otherwise the whole operation would be completely pointless.
- It is also noteworthy that both S and T have a non-zero determinant, therefore they are invertible. The inverse matrices S^{-1} and T^{-1} are given as:

$$S^{-1} = \begin{pmatrix} \frac{1}{\sigma_x} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_y} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } T^{-1} = \begin{pmatrix} 1 & 0 & 0 & -\Delta s_1 \\ 0 & 1 & 0 & -\Delta s_2 \\ 0 & 0 & 1 & -\Delta s_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation and translation in 3D

- The inverse of a product of two matrices A and B is given as $(AB)^{-1} = B^{-1}A^{-1}$.
- The inverse of a transformation $V = TRS$ is therefore given as $V^{-1} = S^{-1}RT\,T^{-1}$.
- Finally, the centering of the rotation operation can be written as $T^{-1}RT$.
- Its inverse is $T^{-1}RT\ T$.
- The true power of the use of matrices lies in the fact that we can combine transformations – the matrices are not commutative, but they are associative.
- A series of transformations can be written down as a product of matrices.
- The resulting matrix can be computed and applied to the 3D coordinates in a given application.

Principal axis transform

- An image coordinate transformation is called *rigid*, when only translations and rotations are allowed. If the transformation maps parallel lines onto parallel lines it is called *affine*. If it maps lines onto lines, it is called *projective*.
- Finally, if it maps lines onto curves, it is called *curved* or *elastic*. Each type of transformation contains as special cases the ones described before it, *e.g.,* the rigid transformation is a special kind of affine transformation. A composition of more than one transformation can be categorized as a single transformation of the most complex type in the composition, *e.g.,* a composition of a projective and an affine transformation is a projective transformation, and a composition of rigid transformations is again a rigid transformation.

Principal axis transform

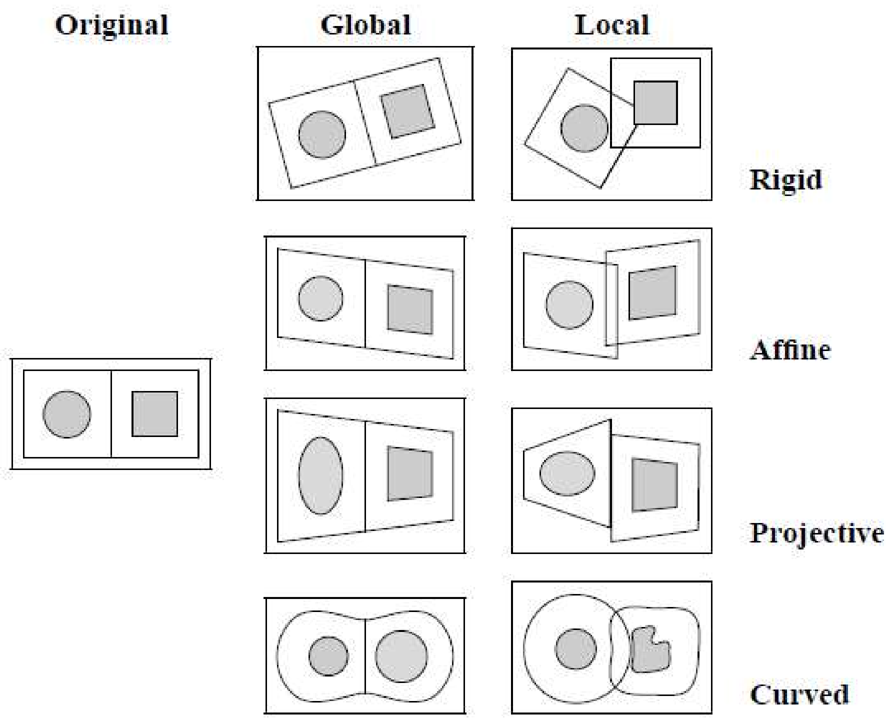


Figure: Examples of 2D transformations

Principal axis transform

- The statistics Principal component analysis (PCA) is special case of a 2D transformation, used in Principal axis transform.
- In a PCA procedure, a set of random variable pairs xi and yi, which may be correlated, is transformed to a subset of uncorrelated variables.
- Basically, correlated variables are connected by a correlation coefficient.
- If we plot all values of xi and yi in a so-called scatterplot, as shown in figure

Principal axis transform

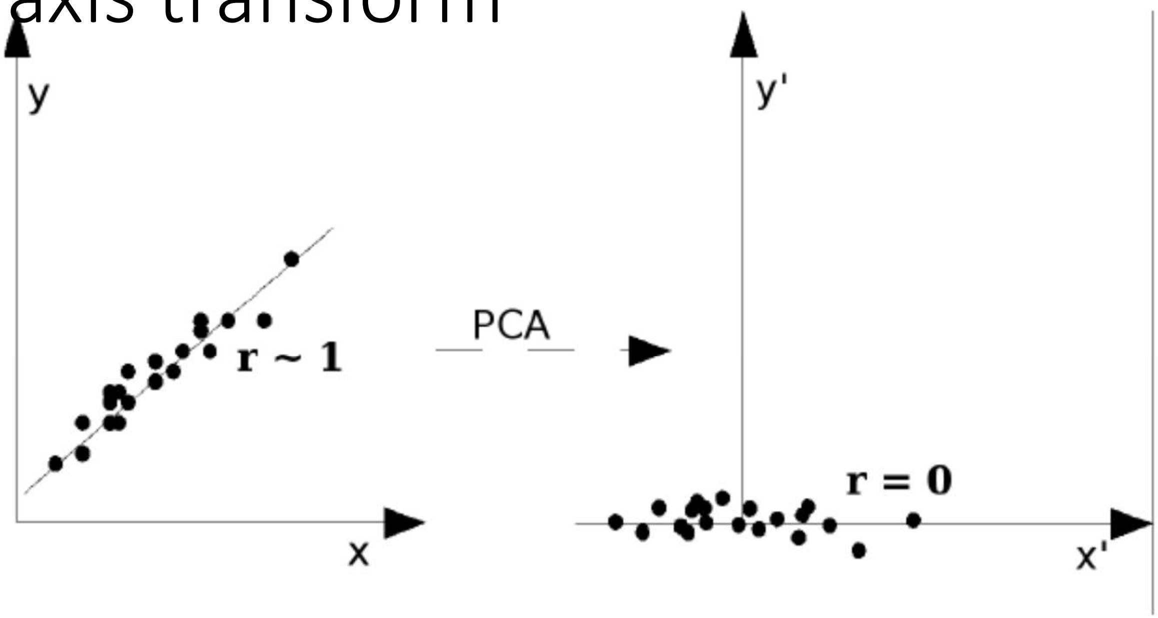


FIGURE The principle of the PCA transform. Pairs of variables x_i and y_i are plotted in so-called scatterplot. If a linear functional correspondence between x_i and y_i exists, the correlation coefficient r is either close to 1 or -1, depending on the gradient of the functional correspondence. This is the situation on the left hand side. The goal of the PCA is to transform the variables (x_i, y_i) in such a manner that the correlation coefficient becomes minimal. In this new frame of reference, the variables are uncorrelated to the largest extent possible. For uncorrelated variables, the correlation coefficient is zero. Since r is closely connected to the slope of the regression line, this slope becomes zero as well (see also the right hand side of the illustration).

Principal axis transform

- The grade of dependency between values of xi and yi can be defined by Pearson’s correlation coefficient r, which is defined as

$$r = \frac{\overbrace{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}^{\text{Covariance of x and } y}}{\underbrace{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2}}_{\text{Standard deviation of x}} \underbrace{\sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}}_{\text{Standard deviation of } y}}$$

Principal axis transform

- Pearson's correlation coefficient $r \in \{-1 \ldots 1\}$ gives the amount of linear dependency of two independently measured variables xi and yi. If r = 1, a high value xi causes a high value of yi, and the dependence of the two is strictly linear.
- The standard deviation of xi and yi, usually called σ_x or σ_y is a measure of the variability of single values versus the expectation value $\bar{x}$ or $\bar{y}$.
- It is also known as the width of a Gaussian if the values of xi or yi follow a normal distribution – therefore, approximately 68% of all values xi can be found in an interval $\bar{x} \pm \sigma$.
- The expectation value $\bar{x}$ in a normal distribution is, of course, the average value of xi. The covariance is something new – it measures to what extent a large deviation of a single value xi from $\bar{x}$ causes a large deviation of the corresponding value yi from $\bar{y}$.
- The correlation coefficient is tightly connected to linear regression analysis, where the straight line that describes the linear dependency of the variable pairs xi and yi is defined as the optimum model for the functional correspondence of the variables.

Principal axis transform

- The gradient a of the regression line y = ax + b is given as:

$$a = r\frac{\sigma_y}{\sigma_x}.$$

- The PCA-transform is indeed a rotation and a translation in the coordinate system spanned by the values of x and y.

Principal axis transform

- The PCA is carried out as follows.

Compute the average vector $\vec{E}$ from all vectors containing the paired values $(x_i,\, y_i)^T$.

The *covariance matrix* C is given as the averaged sum of the matrix product

$$C(x,y) = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} x_i \\ y_i \end{pmatrix}(x_i,\, y_i) - \vec{E}\vec{E}^T.$$

The eigenvectors and eigenvalues of $C(x,y)$ are computed, and the eigenvectors are sorted in a descending manner according to their eigenvalues. A matrix R is formed where the columns represent the eigenvectors.

The PCA transformation is carried out by computing $R\left((x_i,\, y_i)^T - \vec{E}\right)$.

Principal axis transform

Figure shows the initial image; after carrying out the PCA, the image is aligned alongside the main axis where most image information is found

The segmented mouse after a principal axes transform

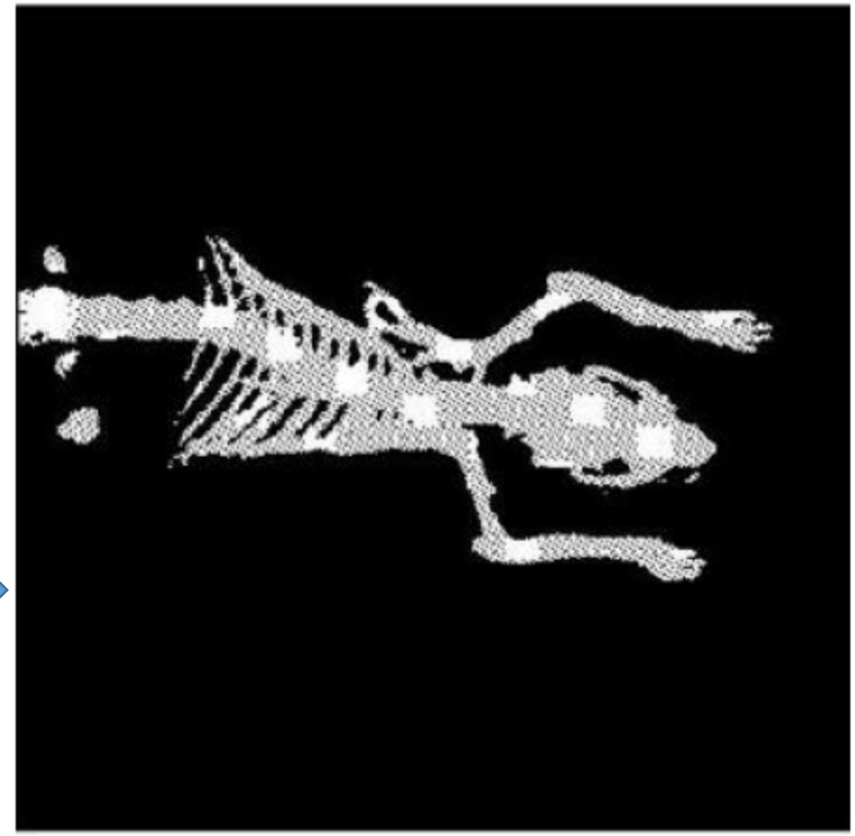
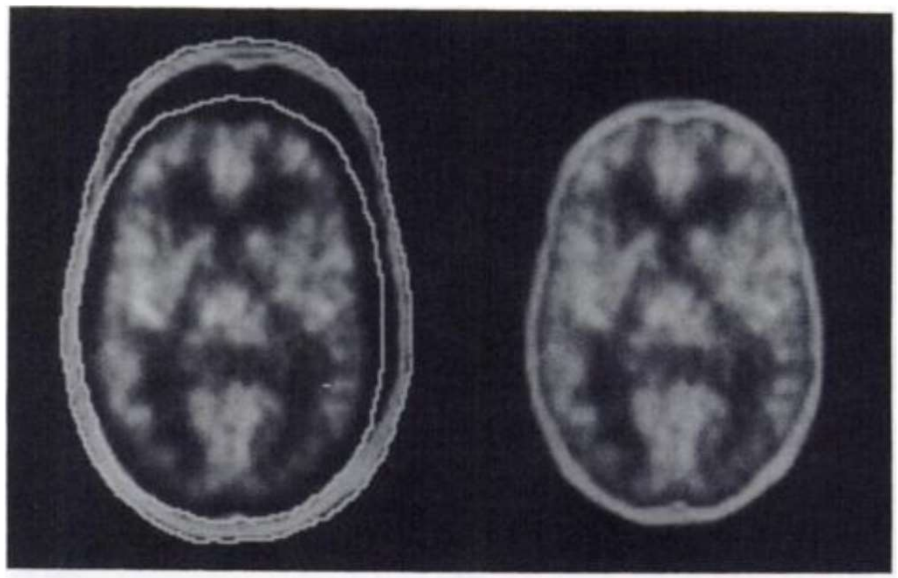


FIGURE: The initial image used for a demonstration of the PCA on binary images. The topogram of a mouse was segmented using a simple thresholding operation, and rotated. The PCA on the coordinates of non-zero pixels yields a spatial transform in 2D that aligns the principal axis – that is, the axis that contains the most pixels

Discretization artifacts are clearly noticeable

Principal axis transform



Alignment of a PET and CT scan on the same patient within the trans axial(x-y) plane. Axial (z) alignment was done externally with a head holder. One combined PET and CT slice is shown before(left) and after(right) alignment. In addition to the coordinate transformation,the CT image is scaled.

The quaternion representation of rotations

- The order in which rotations are carried out is not arbitrary since the single rotation matrices do not commute.
- In aviation, where the rotation around the x-axis is called roll, the rotation around the y-axis is the pitch, and the z-axis rotation is called yaw, this is even regulated by an official norm (otherwise, aviation control would yield catastrophic results in five out of six cases).
- This parameterization in Euler-angles gives rise to even more problems since in a certain position, one of the Euler-angles becomes arbitrary.
- This is the so-called Gimbal lock problem. And finally, it is extremely cumbersome to interpolate rotation motion from angles
- Therefore, a more convenient parametrization of the group of rotations in 3D space would be helpful; it should be unambiguous, and it should allow for some sort of commutative math in order to allow for adaptive filtering. This parameters are called unit quaternions.

The quaternion representation of rotations

- In order to give a tangible interpretation of unit quaternions, it is useful to introduce another possible way of giving rotation parameters – the eigenvector of the rotation matrix R and the angle of rotation around this single vector.
- The eigenvector $\vec{u}$ of length 1 is the one axis defined by all points that do not change their position when carrying out the rotation.
- The only remaining degree of freedom is the one angle ϕ that gives the amount of the clockwise rotation around $\vec{u}$. A quadruple q derived from $\vec{u}$ and ϕ can be represented as

$$q = (q_0,\, q_1,\, q_2,\, q_3)^T = \left(\cos\left(\frac{\phi}{2}\right),\, \vec{u}^T \sin\left(\frac{\phi}{2}\right)\right)^T$$

Unit quaternions fulfill the identity $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$

The quaternion representation of rotations

- Quaternions are a quadruple of one scalar and three complex numbers.
- The imaginary unit i now has two more brothers, usually named j and k.
- For all of those, the identity $i^2 = j^2 = k^2 = -1$ is true.
- A quaternion q is given as q = q0+iq1+jq2+kq3 and can therefore be expressed as $q = (q_0, q_1, q_2, q_3)^T$, analog to a complex number.
- The multiplication of two unit quaternions is not commutative, but it is anti-commutative; the multiplication of two quaternions u and v yields the following result: uv = w or vu = −w.
- Compared to the behavior of rotation matrices, this is a huge simplification.

The quaternion representation of rotations

- In medical image processing, namely in image-guided therapy and registration, quaternions play an important role.
- First, almost all tracking systems give their output as quaternions.
- Second, one can easily determine the eigenvector of rotation and the associated eigenvalue of an arbitrary complex transform in 3D space.
- And finally, one can interpolate the angulations in a complex 3D trajectory, for instance in biomechanics.
- Therefore, we need to know how a quadruple of unit quaternions q can be expressed as a rotation matrix.
- This is done by carrying out the following calculation:

$$R = \begin{pmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_0q_2+q_1q_3\right) \\ 2\left(q_0q_3\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_1q_3-q_0q_2\right) & 2\left(q_0q_1+q_2q_3\right) & 1-2\left(q_1^2+q_2^2\right) \end{pmatrix}$$

The quaternion representation of rotations

- The inverse operation is,

Compute the trace of the rotation matrix: $\mathrm{tr}R = \sum_{i=1}^{3} R_{ii}$.

Compute four values $p_0 \ldots p_3$ as follows:

$$
\begin{aligned}
p_0 &= 1 + \mathrm{tr}R \\
p_1 &= 1 + 2R_{11} - \mathrm{tr}R \\
p_2 &= 1 + 2R_{22} - \mathrm{tr}R \\
p_3 &= 1 + 2R_{33} - \mathrm{tr}R
\end{aligned}
$$

Determine the maximum $p_{\max}$ of $\{p_0 \ldots p_3\}$.

Compute $p_s = \sqrt{p_{\max}}$.

The quaternion representation of rotations

- The index s of p_{max} determines which of the four following computations has to be carried out:

$$
\begin{aligned}
s = 0: \quad q_0 &= \frac{1}{2} p_0 \\
q_1 &= \frac{1}{2p_0} \left(R_{32} - R_{23} \right) \\
q_2 &= \frac{1}{2p_0} \left(R_{12} - R_{21} \right) \\
q_3 &= \frac{1}{2p_0} \left(R_{21} - R_{12} \right)
\end{aligned}
$$

The quaternion representation of rotations

$$
\begin{aligned}
s = 1: \quad q_0 &= \frac{1}{2p_1}(R_{32} - R_{23}) \\
q_1 &= \frac{1}{2}p_1 \\
q_2 &= \frac{1}{2p_1}(R_{21} + R_{12}) \\
q_3 &= \frac{1}{2p_1}(R_{13} + R_{31})
\end{aligned}
$$

$$
\begin{aligned}
s = 2: \quad q_0 &= \frac{1}{2p_2}(R_{13} - R_{31}) \\
q_1 &= \frac{1}{2p_2}(R_{21} + R_{12}) \\
q_2 &= \frac{1}{2}p_2 \\
q_3 &= \frac{1}{2p_2}(R_{32} + R_{23})
\end{aligned}
$$

The quaternion representation of rotations

$$
\begin{aligned}
s = 3: \quad q_0 &= \frac{1}{2p_3}(R_{21} - R_{12}) \\
q_1 &= \frac{1}{2p_3}(R_{13} + R_{31}) \\
q_2 &= \frac{1}{2p_3}(R_{32} + R_{23}) \\
q_3 &= \frac{1}{2}p_3
\end{aligned}
$$

If q_0 is negative, all values $q_0 \ldots q_3$ should be multiplied with -1.

Check if the result is still a unit quaternion.

the unit quaternion is a representation of the eigenvector of an arbitrary rotation matrix R as given by an arbitrary product.
The inverse operation is an algorithm that derives the eigenvector and the angle of rotation of R.

REFORMATTING

- We have seen important visualization method for 3D volume image data.
- The tomographicimages must not be considered stacks or collections of single images; they are, indeed, images defined on a 3D domain.
- The fact that 3D images are acquired in a slice wise manner in many imaging systems, does, however, not simplify this model.
- However, CBCT and SPECT, for instance, do not work in such a manner.
- Rather than that, they acquire projection data, and the volume is reconstructed from a multitude of these projections.
- MR machines do acquire data as slices, but the position of the imaging plane is determined using the MR machine parameters and can be chosen arbitrarily

REFORMATTING

- The other problem lies in the fact that visualization of a function $I(x, y, z) = \rho$ is not as trivial as in the case of 2D images.
- A possibility to visualize the 3D datasets is to compute new slices.
- Another possibility is the use of visualization techniques such as rendering.
- However, the generation of new slices is the most important visualization technique.
- It is referred to as reformatting. We can distinguish between orthogonal reformatting – this is the derivation of new slices from the volume alongside the main anatomical directions – axial/transversal, sagittal and coronal.
- However, the exactness of this definition is limited. If a patient lies in the scanner at a slightly tilted position (a CT machine, for instance, can tilt the whole unit with an angle of up to 15◦), the axial direction is not coincident with the orientation of the body stem.
- Reformatting of slices in arbitrary orientations is called oblique reformatting. And finally, we can define an arbitrary 2D manifold, which intersects the volume image.
- This is generally referred to as curved reformatting.

REFORMATTING

- Figure shows such an image; this is a so-called pantomographic reformatting from dental radiology, where a CT of the mandible was reformatted in such a manner that the reformatting plane follows the arc of the jaw, thus producing an image similar to panoramic x-ray.

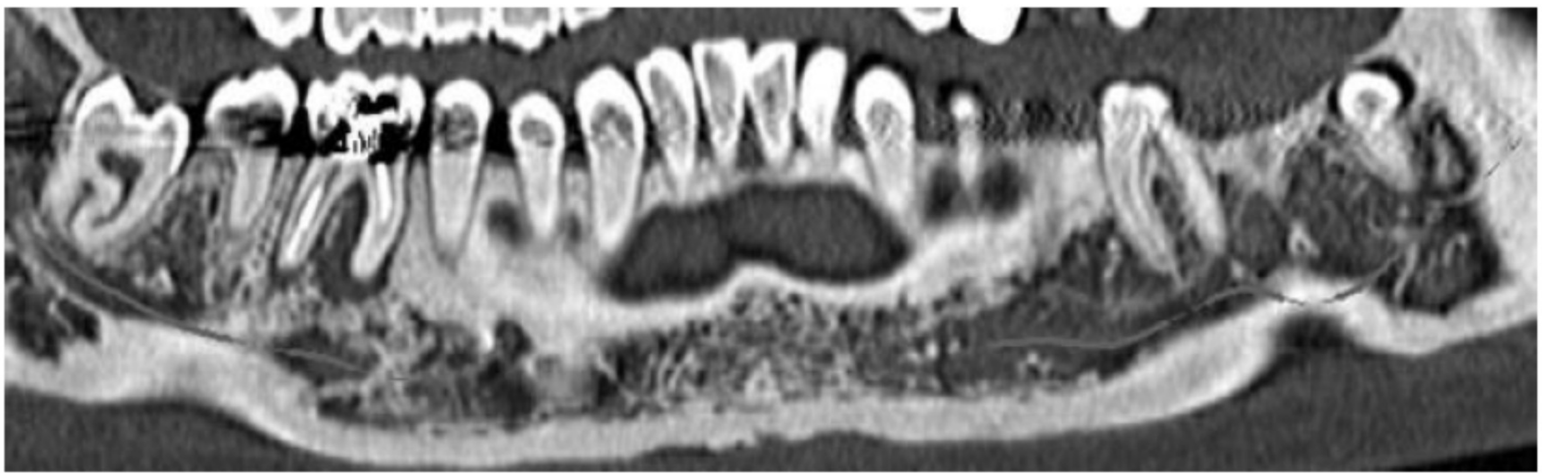


FIGURE: A curved reformatting from dentistry. Here, a standard CT of the mandible was acquired, and a curved reformatting plane matching the arc of the jaw is defined. Therefore, a slice image of the whole jaw is provided, similar to the well-known panoramic x-ray used in dentistry for more than 80 years.

REFORMATTING

- The derivation of orthogonal and oblique sections is actually rather simple. We have our data volume, built up from voxels and defined in a Cartesian coordinate system.
- We may consider the x-y plane to be the initial cutting plane.
- All we have to do is rotate and translate this plane to the desired position, and to transfer the gray values ρ from a voxel at the position (x, y, z)T that intersects a pixel in the cutting plane to exactly this pixel in the 2D image.
- Interpolation of an appropriate gray value from the surrounding voxels is usually a necessity.

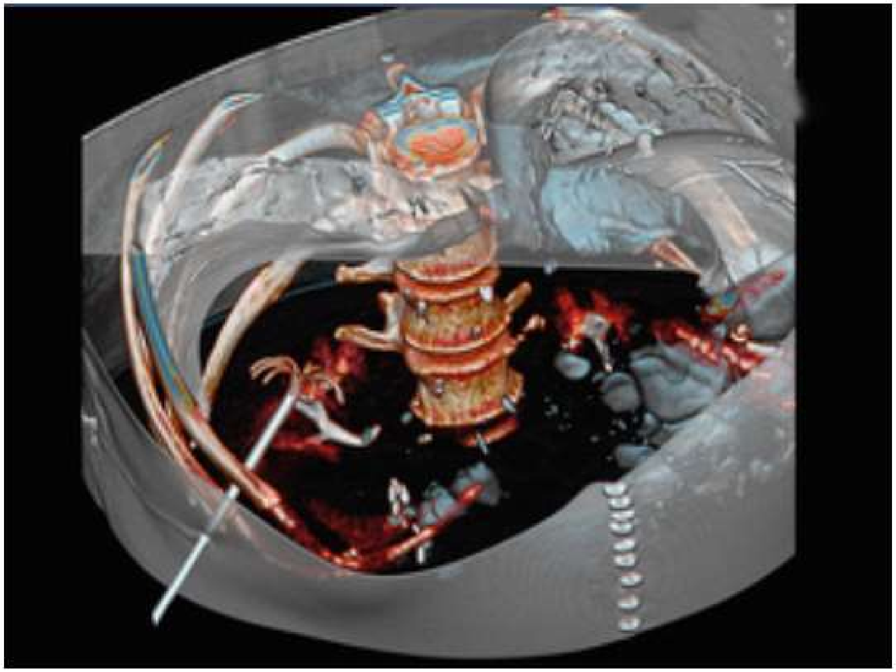
TRACKING AND IMAGE-GUIDED THERAPY

- With the tools to handle spatial transforms at hand, we can proceed to an application field where diagnostics and therapeutic measures are joined – image-guided therapy, often also referred to as stereotactic surgery, frameless stereotaxy, neuro navigation, or computer-aided surgery.
- Here, we use three components to guide the physician during an intervention or a surgical procedure:

1. Tracking of patient position and 3D pose of instruments.
2. Registration of the patient coordinate system to the reference frame of an image dataset.
3. Visualization of the actual situation by generating views from the registered image data.

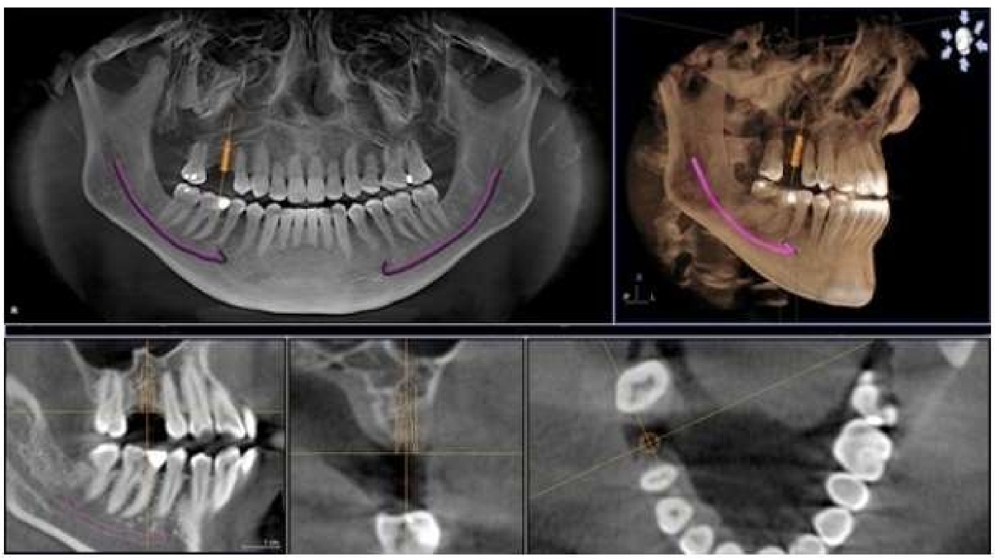
TRACKING AND IMAGE-GUIDED THERAPY

Example: In cancer treatment volume rendering play an important role. Patient is scanned prior to treatment and visualized as a 3D volume this allows accurate localization of tumour and minimize the adverse effect on the normal tissues. It also used to figure out the progress of treatment. By measuring the size and the shape of the tumour to see the effectiveness of the treatment.



TRACKING AND IMAGE-GUIDED THERAPY

- **Dental implant planning**
- This is another application for image guided therapy. A software simulator based on volume rendering will use the scanned skull of the patient to plan the process of medical implant and get the best orientation and size of new implant. Actually this is the core work of one of a good market companies (360 Imaging)



TRACKING AND IMAGE-GUIDED THERAPY

- Tracking in general refers to the real-time measurement of six degrees of freedom of several rigid bodies. Several techniques for tracking exist.

1. Optical Tracking
2. Electromagnetic Tracking
3. Mechanical Tracking
4. Ultrasonic Tracking
5. Passive HF-Tracking
6. Inertial Tracking
7. Image-based Tracking

Optical Tracking

- Several calibrated cameras acquire an image of active or reflective optical beacons of known geometry.
- By reconstructing the 3D position of at least two (for 5 dof) or three beacons (for 6 dof), one can derive the volume transformation V of the rigid body carrying the beacons relative to the camera coordinate system; an algorithm for deriving an affine transformation from known position pairs in two coordinate systems is used to track multiple rigid bodies (usually referred to as tools) since these are either connected by cable to the control unit of the camera, or they feature a unique configuration of markers in the case of wireless systems.

FIGURE : A passive optical tracking probe. The tracking system flushes the digitizing volume with infrared light, which is reflected by the four spheres. Due to the known unique configuration of the beacons on the tool, the tracker can distinguish between several tools.
Since the system is passive, no additional wiring is necessary.

Optical Tracking

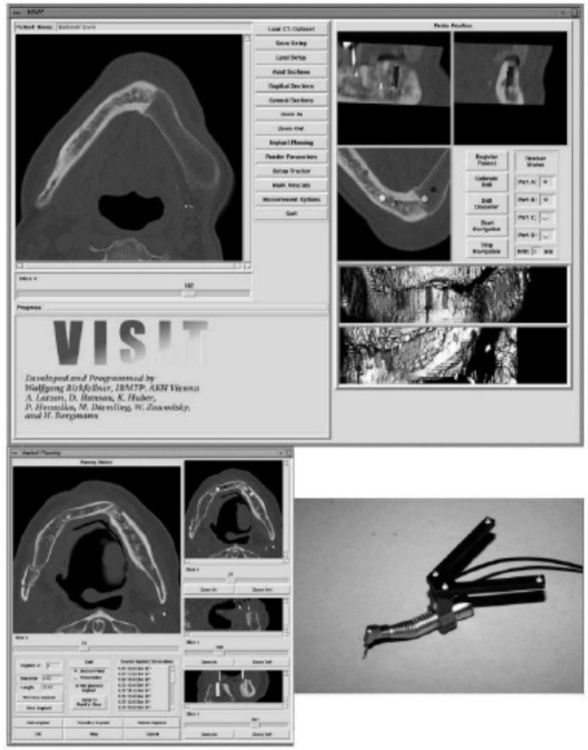


FIGURE: Some screenshots of VISIT, a system for image-guided cranio- and maxillofacial surgery developed by the author. The top screenshot shows the main screen of the program, where a dental CT scan can be browsed, and planned sites for dental implants are shown in dependence of surgical drill position on oblique reformatted slices and volume renderings. After preoperative planning (lower left screenshot) of implant channels on the CT, the patient was registered using a point-to-point registration technique. The position of the surgical drill (lower right image) is reported by an optical tracking system; the tool for the optical tracker is rigidly attached to the drill

Optical Tracking

FIGURE: An image of a camera bar of an optical tracking system. This is a system featuring two planar cameras with a filter that blocks visible light. The digitizer volume is flooded with infrared light from an array of light-emitting diodes surrounding the camera. Reflective spheres on the tool show up in the camera images. By knowing the relative configuration of the beacons and the projection geometry of the camera system, one can derive an affine transformation in six dof.

Optical Tracking

- Optical tracking systems are the most widespread tracking technology, with typical accuracy in the range of 0.3 mm; their main disadvantage lies in the fact that a free line-of-sight between the tool and the camera is necessary
- Another type of optical tracking system that is not very widespread in medicine sweeps the digitizer volume with laser fans; the beacons on tools are replaced by photoresistors in these systems.

Electromagnetic Tracking

- Another widespread technology is electromagnetic tracking, where the field strength or the magnetic flux of a specially designed electromagnetic field is measured by small sensors like search coils or flux-gate sensors.
- The electromagnetic field permeates soft tissue and does therefore not require a free line-of-sight.
- Furthermore, the sensors can be built with an extremely small form factor that allows for the use of electromagnetic trackers within flexible instruments such as endoscopes.
- A drawback of electromagnetic trackers, however, is the fact that the reference field may be distorted by conductive or ferromagnetic materials, which depreciates the accuracy achievable with these devices.
- Still, these systems provide the technology of choice in current image – guided therapy systems if flexible instruments are to be tracked within the body.

Electromagnetic Tracking

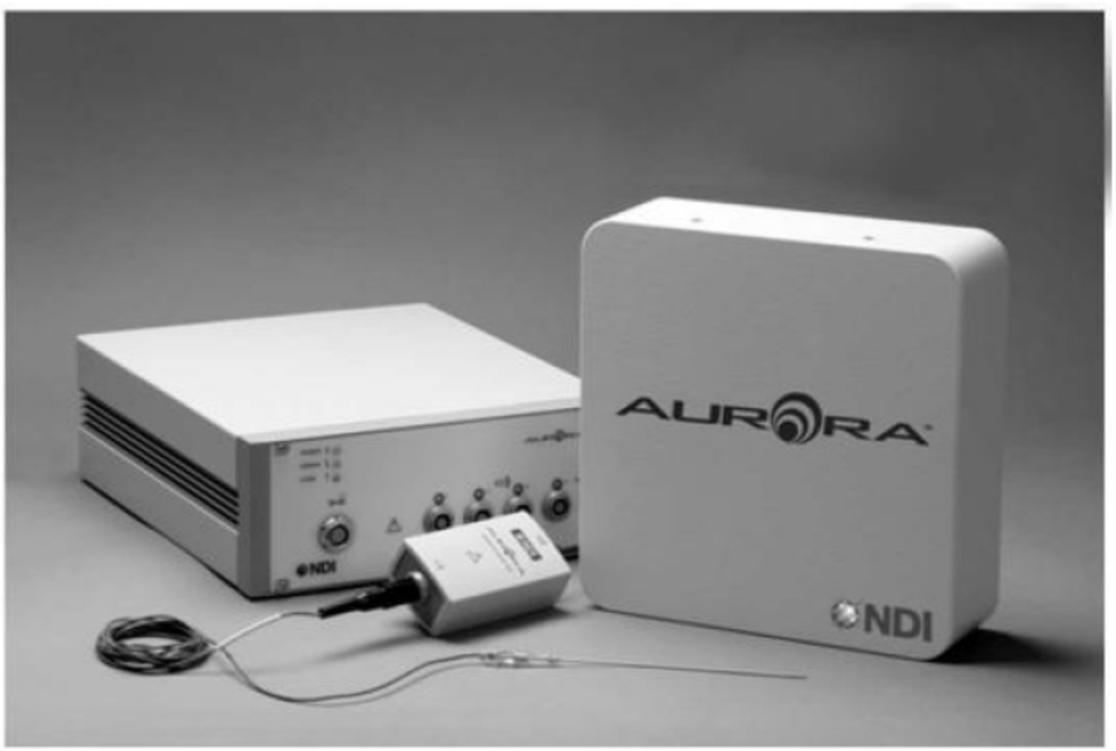


FIGURE: An electromagnetic tracking system. The field emitter generating the electromagnetic reference field can be found in the right half of the image. A needle containing a small position sensor located at the needle tip is shown in front of the control hardware which processes the raw sensor data and interfaces to a computer.

Mechanical Tracking

- The earliest systems for computer-aided neurosurgery consisted of a single passive arm with encoders on each joint.
- Since the relative position of the joints relative to each other is known, the position of the instrument tip at the end of the device can be determined from the kinematic chain.
- In neurosurgery, the patient's calvaria is fixed by means of a so-called Mayfield clamp,1 therefore one tracked tool is sufficient.
- These systems are very accurate, but bulky, difficult to sterilize, and were more or less replaced by optical trackers.
- Nevertheless, position information from decoders still plays an important role in medical robotics.

Ultrasonic Tracking

- A system similar to optical trackers can be built using ultrasoundemitters and microphones.
- The ultrasound emitter acts like an optical beacon, and an array of microphones takes the position of the camera in an optical tracker.
- However, the ultrasound tracker suffers from the same problem as the optical tracker, which is the requirement for a free line-of-sight between the emitter and the detector.
- Furthermore, changes in air humidity and temperature affect the speed of sound, which also affects tracker accuracy.
- Therefore, ultrasonic tracking systems have not found widespread usage in image-guided therapy despite the fact that cost-effective systems are available.

Passive HF-Tracking

- An interesting approach consists of small passive transponders which emit a signal when being exposed to an electromagnetic field of high frequency.
- These transponders, contained in small glass capsules, can be implanted near a tumor in healthy tissue.
- By monitoring the position of the beacon during radiotherapy, internal organ motion can be monitored to spare healthy tissue.
- The positioning of the emitter is somewhat delicate, therefore tracking applications for a large range of motion appear difficult and were not yet realized.

Inertial Tracking

- One can derive position from a known starting point by integrating acceleration twice in the time domain.
- The same holds true for angulation, where an integration of angular velocity yields a change in rotation.
- Acceleration and angular velocity can be measured by special sensors.
- However, the time integration step yields an inevitable drift over a longer period of time.
- Therefore, such inertial trackers were only used as supporting systems in augmented or virtual reality application, where they provide input for adaptive filtering of position data.

Image-based Tracking

- Finally, it is possible to retrieve motion directly from medical images if the imaging modality is located within the operating room.
- By applying registration techniques presented in Chapter 9, 3D motion can, for instance, be derived from 2D projective data.
- In the case of x-ray imaging, this may cause considerable additional dose.
- But also monitoring of interventional progress by US- and MR-imaging is used in clinical routine.

- *Common to all tracking systems is the fact that their output is usually in quaternions, which have to be converted to rotation matrices R.*
- *The clinical application fields of image-guided therapy range from neurosurgery, ear-, nose-, and throat surgery, craniomaxillofacial surgery, orthopedics, plastic and trauma surgery to interventional radiology, radiotherapy, and surgical training.*

Thank You