Image Analysis for Medical Visualization

Alejandro J. León Salas[1,2]

[1]Department of Software Engineering
University of Granada

[2]Virtual Reality Lab
University of Granada

15 February, 2020

Outline

Motivation

- Image analysis techniques extract clinically relevant information from radiological data. We focus on extracting information for generating **high-quality visualizations** and for deriving **quantitative information** about relevant structures.
- Overview of important image analysis tasks and some established solutions. We shall discuss not only algorithms that detect and analyze features in medical image data, but also **interaction techniques** that allow the user to guide an algorithm and to modify an existing result.

Overview

- Image analysis is often carried out as a pipeline of individual steps.
- This pipeline starts with **preprocessing** and **filtering**, which is designed to support subsequent algorithms.
- **Image segmentation** assigns labels (unique identifiers) of anatomical or pathologic structures to parts (regions) of the image data.

Overview

There are two basic strategies to address segmentation problems:

- The **edge-based approach**, which searches for discontinuities in the image data that belong to the border of the segmentation target structure.
- The **region-based approach**, in which the target structure is regarded as a homogeneous region determined by a search process guided by appropriate criteria for homogeneity.

Overview. Spatial domain

- The term **spatial domain** refers to the image plane itself, and and **Image Processing** (**IP**) methods in this category are based on direct manipulation of pixels in an image.
- Two principal categories of spatial processing are intensity transformations and spatial filtering.
- **Intensity transformations** operate on single pixels of an image, principally for the purpose of **contrast manipulation** and **image thresholding**.
- **Spatial filtering** deals with performing operations, such as **image sharpening**, by working in a neighborhood of every pixel in an image.

Overview. Spatial domain

- The spatial domain processes we discuss here can be denoted by the expression

$$g(x, y) = T[f(x, y)] \tag{1}$$

- where $f(x, y)$ is the input image, $g(x, y)$ is the output image, and T is an operator on f defined over a neighborhood of point (x, y). The operator can apply to a single image or to a set of images.

Overview. Spatial domain

- **Spatial (kernel) filtering**. The idea consists of moving the origin of the neighborhood from pixel to pixel and applying the operator T to the pixels in the neighborhood to yield the ouput at that location.
- For any specific location (x, y), the value of the output image g at those coordinates is equal to the result of applying T to the neighborhood with origin at (x, y) in f.
- For example, neighborhood is defined as a square of size 3×3, and operator T is defined as "compute the average intensity of the neighborhood". Assuming that the origin of the neighboorhood is at its center, if we consider an arbitrary location (x_i, y_j) then, the result, $g(x_i, y_j)$, at that location is computed as the sum of $f(x_i, y_j)$ and its 8-neighbors, divided by 9.

Overview. Spatial domain

- The smallest possible neighborhood is of size 1×1. In this case, g depends only on the value of f at a single point (x, y) and T in Eq. (2) becomes an **intensity transformation function** of the form

$$s = T(r) \tag{2}$$

- where, for simplicity in notation, s and r are variables denoting, respectively, the intensity of g and f at any point (x, y).

Overview. Histogram

- The **histogram** of a digital image with intensity levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value and n_k is the number of pixels in the image with intensity value r_k.
- **Histogram normalization**. To divide each of its components by the total number of pixels in the image, denoted by the product MN, where, as usual, M and N are the row (x) and column (y) dimensions of the image.
- A **normalized histogram** is given by $p(r_k) = n_k/MN$, for $k = 0, \ldots, L-1$. $p(r_k)$ can be considered as an estimate of the probability of occurrence of intensity level r_k in an image.

Overview. Histogram

- Histograms are the basis for numerous spatial domain processing techniques and can be used for image enhancement and segmentation.
- The horizontal axis of each histogram plot corresponds to intensity values, r_k. The vertical axis corresponds to values of $h(r_k) = n_k$ or $p(r_k) = n_k/MN$ if the values are normalized.
- Histograms may be viewed graphically simply as plots of $h(r_k) = n_k$ versus r_k or $p(r_k) = n_k/MN$ versus r_k.

Preprocessing and Filtering

- Radiological image data exhibit artifacts such as noise and inhomogeneities.
- Shading artifacts often occur in MRI data, and metallic implants, for example in the teeth, cause artifacts in CT and MRI data. Additionally, motion of the patients and breathing might locally decrease image quality.
- **Noise reduction** leads to more homogeneous regions, which might be delineated with less interaction effort. On the other hand, noise reduction might compromise the detection of small relevant features.

ROI selection

- A first step is the definition of a **Region Of Interest** (**ROI**) that comprises all relevant structures. We use the term ROI for the selection of subimages in 2D data as well as for the selection of subvolumes in 3D data, also known as Volumes Of Interest (VOI).
- The ROI usually has the shape of a cuboid.
- ROI selection accelerates subsequent computations and enhances visualization because irrelevant information is not included.

ROI selection

- The **interaction** to accomplish ROI selection should be a combination of:
  - Direct manipulation (drag the border lines of the ROI) in an appropriate visualization of the data.
  - The specification of numbers (precise input of ROI position and extent in each direction).
- It might be necessary for certain segmentation tasks to define **further ROIs** as a subset of the first ROI.
- These additional ROIs (possibly irregular-shaped) may serve as barriers that **reduce the search space** for segmentation information.

Resampling

- The anisotropic nature of many datasets can be a severe quality issue for image analysis and visualization techniques. Therefore, a resampling step is often included to transform data to an isotropic (regular) grid.
- The accuracy of the data should not be degraded. Therefore, resampling is driven by the highest resolution (usually in-plane) and interpolates additional data in the dimension with lower resolution (usually the z-dimension).
- Fast methods interpolate the value at a particular position by taking into account the neighboring voxels only (trilinear interpolation). Better results are achieved with triquadratic or tricubic interpolation.

Histogram Equalization: Contrast

- Histogram equalization is a method in image processing of contrast adjustment using the image's histogram.

Figure: Changes in the amount of contrast in a photo (CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=750555).

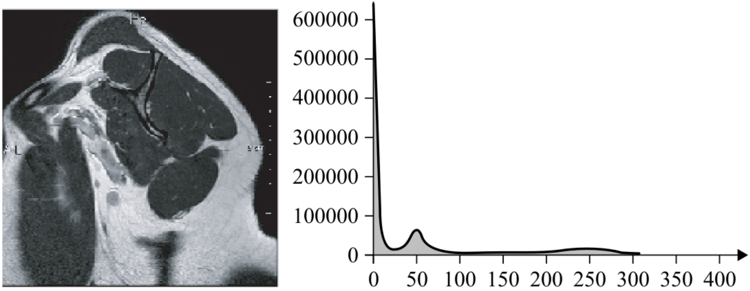
Histogram-related operations

- For many operations applied to image data, we employ the histogram to select appropriate parameters.
- Histograms are analyzed, for example, with respect to their **significant local maxima** (or simply **peaks**).
- The definition of a significant local maximum is difficult and application-dependent.
- In general, a local maximum that strongly exceeds adjacent values and is not too close to another significant local maximum is considered significant.

Histogram-related operations

- In medical image data, a peak in the histogram is often related to one tissue type.
- Depending on the tissue types and the imaging modality, the tissue types may overlap each other in the histogram, so that the number of peaks is lower than the number of tissue types.
- The following slide presents a slice of an MRI dataset of the shoulder region along with the histogram (of the whole dataset).
- The large peak in the left relates to background voxels, the second peak corresponds to muscles, and the smaller third peak to soft tissue structures with higher image intensity.

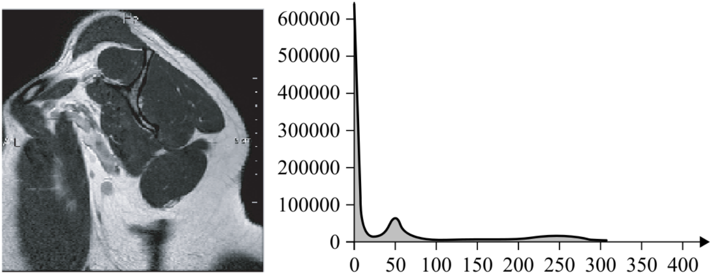
Histogram-related operations

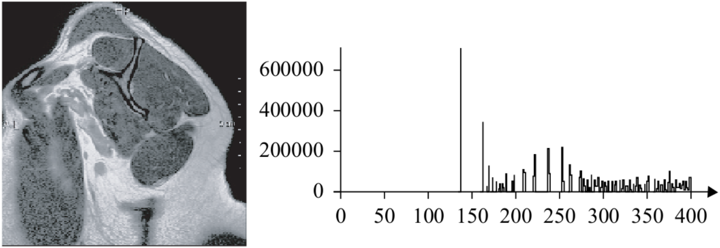


Histogram Equalization

- The **histogram equalization** is the most common histogram transformation technique and **enhances the image contrast**. With this transformation $\boldsymbol{T}$, an image is transformed such that a new histogram results, in which all intensity values are approximately constant.
- $\boldsymbol{T}$ is a monotonically increasing function in the whole interval of intensity values. This property ensures that pixels that were darker in the original image remain darker in the resulting image.
- Although the result is not a perfectly uniform histogram, the histogram of the output image is spread over a wider range of (gray) values.

Histogram Equalization





Histogram Equalization

- The method is useful in images with backgrounds and foregrounds that are both bright or both dark, and is especially indicated for x-ray images.
- Histogram equalization often produces unrealistic effects in photographs; however it is very useful for scientific images like thermal or x-ray images.
- Also can produce undesirable effects (like visible image gradient) when applied to images with low color (or intensity) depth. It will work the best in depths greater than or equal 16-bit.

Histogram Equalization: Implementation

- Remember that the **histogram** of a gray level image with intensity levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value and n_k is the number of pixels in the image with intensity value r_k.
- And remember that a **normalized histogram** is given by $p(r_k) = n_k/MN$, for $k = 0, \ldots, L-1$. $p(r_k)$ can be considered as an estimate of the probability of occurrence of intensity level r_k in an image.

Histogram Equalization: Implementation

- Hence, let p_x be any pixel in the image. The probability of an occurrence of a pixel of level i in the image is $p_x(i) = p(x = i) = n_i / MN$
- Let us also define the **cumulative distribution function** corresponding to p_x as $cdf(i) = \sum_{j=0}^{i} p_x(j)$.
- We would like to create a transformation of the form $g = T(f)$ to produce a new image g with a flat histogram. Such an image would have a linearized cumulative distribution function across the value range, i.e. $cdf_g(i) = iK$ for some constant K.

Histogram Equalization

- Histogram transformations may strongly enhance the perception and interpretation of images.
- Due to their strict monotonic behavior, image segmentation usually does not benefit from prior histogram transformations.
- Instead, for segmentation purposes, **noise reduction** is essential.

Noise Reduction

- Medical image data exhibit random noise due to stochastic processes in the image acquisition.
- Noise is characterized by a certain amplitude and distribution.
- The **noise level** is often measured as the **signal-to-noise ratio** in the whole image and depends on the imaged tissue and on its mean gray value (locally different).
- The noise level also depends on the spatial resolution of the data; high resolution data, such as CT data with $0.5mm$ slice thickness, exhibit more noise.
- In X-ray and CT imaging, the noise level depends on the amount of radiation.

Noise Reduction filters

- **Noise reduction filters** are employed to enhance the data. In general, it is assumed that noise occurs in high frequency. Therefore, **low pass filters** are used for noise reduction.
- The design of these filters is based on assumptions concerning the amplitude and distribution of noise.
- A variety of filters have been designed to reduce noise with a **Gaussian distribution**.

Noise Reduction filters

- The Gaussian filter kernel is represented by Equation (3), where μ represents the centroid of the function and σ represents the standard deviation and thus the width of the function.

$$G(x, \sigma, \mu) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{\frac{x-\mu}{\sigma^2}} \tag{3}$$

Noise Reduction filters

- The Gaussian is an example of the general concept of filtering a function g with a filter F , which is often represented as a **convolution**.

$$g'(x) = g(x) \otimes F(x) = \int_{-\infty}^{\infty} g(x) \cdot F(x - x_1)\, dx \quad (4)$$

where $\otimes$ denotes the **convolution operator**.

Noise Reduction filters

- To convolve discrete image data, the convolution is expressed as a weighted sum of the signal g over the filter kernel F with $2N+1$ elements.

$$g'(u) = \sum_{i=-N}^{N} g(u) \cdot F(u-i) \tag{5}$$

Noise Reduction filters

- Simple filters are **local filters** that modify the image intensity at each voxel by a combination of image intensities at neighboring voxels.
- **Discrete local filters** are characterized by a kernel, i.e. a **matrix** (array) of elements with the same size as the neighborhood being considered.
- Usually the matrix has the size $(2N+1) \times (2N+1) \times (2M+1)$, with N usually being 1, 2, or 3 and M being 0, 1, or 2. In cases of isotropic data, M and N usually have the same value, whereas in anisotropic data a smaller M is chosen to account for the larger extent in the z-direction.

Noise Reduction filters

- With $M = 0$, the filter is applied to each slice separately and has no effect on other slices.
- This is the typical situation when data have a highly anisotropic voxel spacing with a slice distance s being more than twice as large as the pixel distance r within a slice.
- One primary classification of filters
  - **Static filter**. The elements of the matrix are constant.
  - **Dynamic filter**. These filters adapt their content to local image characteristics.

Static Noise Reduction Filters

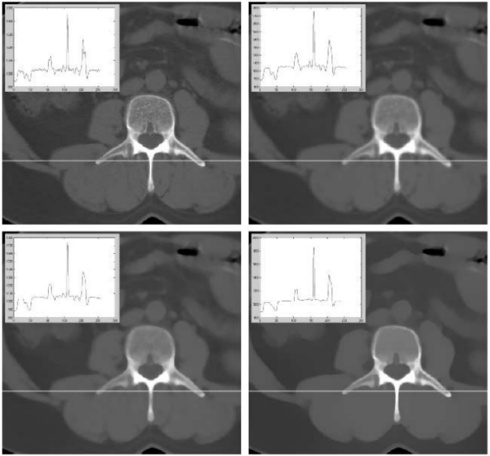
- Static noise reduction filters are scaled so that the sum of their elements is 1.
- Then, the mean gray value of the image remains the same after the filter was applied, thus maintaining the overall image intensity.
- The filter is applied by iterating over all pixels and replacing the image intensity with a weighted average of neighboring pixels.
- The weights of the neighboring pixels are characterized by the **filter matrix**.

Static Noise Reduction Filters

- **Average filter**. The simplest filter in which each neighbor voxel has the same influence. Better results are achieved with a filter that takes the distance to the central voxel into account.
- **Binomial filter** (called **Gaussian** in many IP systems), a discretized version of the Gaussian function.
- The elements of the filter are binomial coefficients.

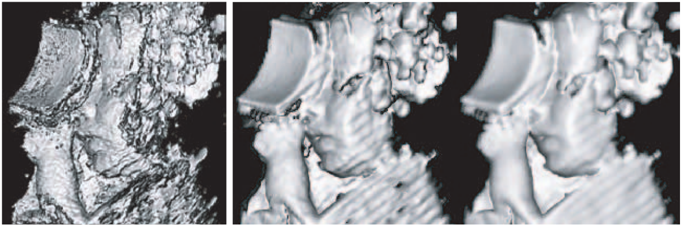
Binomial (Gaussian) filter

- The result of a 2D binomial filter with kernel size 5×5 applied to CT data (upper right) plus 5×5 median filter (lower left) and diffusion filter (lower right).



Binomial (Gaussian) filter

- The effect of Gaussian filtering on 3D visualizations derived from ultrasound data of a fetus. Left: original data, middle: after binomial filtering with a $5 \times 5 \times 5$ kernel, and right: after filtering with a $7 \times 7 \times 7$ kernel.



Binomial (Gaussian) filter

- The 5×5 filter has the following kernel matrix. To normalize the kernel elements, we divide them by the sum of all elements (256).

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

Binomial (Gaussian) filter

- **Separability** is an important property for the efficient application of filters with larger kernels to large volume data.
- Separable filters with a two (or three)-dimensional kernel can be replaced by combining two (or three) one-dimensional filters. This way, the quadratic (cubic) complexity can be reduced to a linear one.
- The drawback of implementing a filter for n dimensions as a sequence of separable one-dimensional filters is the additional memory consumption necessary to store intermediate results.

Main problem in using static local filters

- The inherent **problem of local filters** with a static matrix is that they are not adaptive. Features such as edges are not preserved and appear washed out.
- Static local filters do not exhibit adaptive behavior; neither the size of the filter nor its elements can be adapted to local image characteristics.

Dynamic Noise Reduction Filters

- Dynamic filters analyze and consider image intensities in the local neighborhood. These filters are computationally more demanding than static ones but allow to adapt to features in the data.
- **Median filter**. The voxels in the neighborhood of the current voxel are sorted into **bins** according to their image intensity, and the median is determined (intensity of the voxel that is in the middle of the sequence). The central voxel is then replaced by the median value.
- Compared to Gaussian filtering, extremely high and low outlier values do not significantly influence the result. The sorting stage, however, takes considerable time; therefore kernel sizes should not be too large.

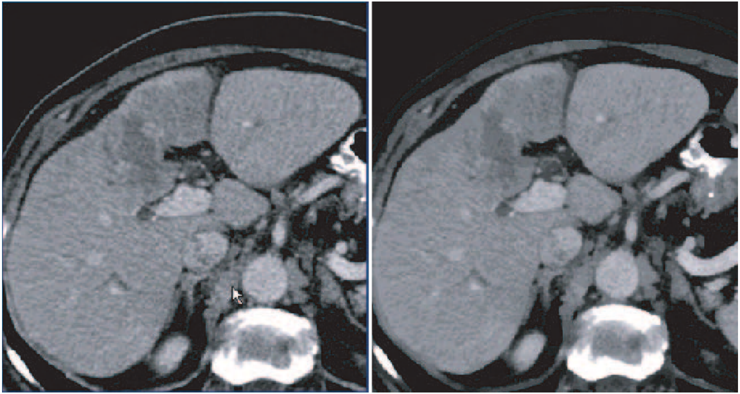
Dynamic Noise Reduction Filters

- σ-filter restricts the filter to voxels with an intensity that does not differ strongly from the local average.
- The filter considers the histogram of image intensities in the local neighborhood. If it turns out that the current voxel has a very high or very low value compared to the average in its neighborhood, it remains unchanged.

σ-filtering

- σ is the parameter of the filter that quantifies how strongly the image intensity may deviate from the average (avg). With $\sigma = 1$, all voxels deviating less than the standard deviation from the average ($avg \pm \sigma$) are considered.
- If the central voxel has an intensity inside the specified interval, the image intensity is replaced by its local average.

σ-filtering

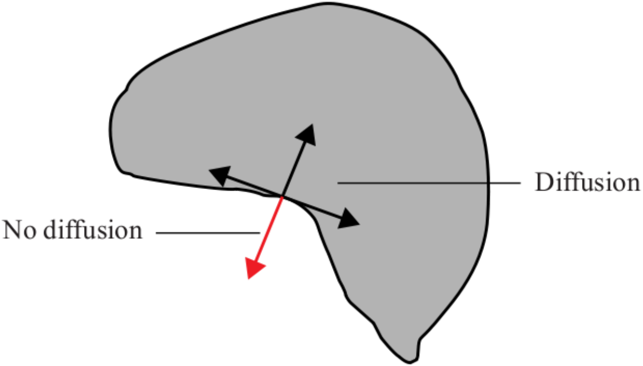


Diffusion Filtering

- A family of advanced edge-preserving filter techniques is diffusion filtering, where the **physical process of diffusion** is simulated.
- The goal of advanced diffusion filtering techniques is to restrict the diffusion to areas without edges as well as along edges.
- **Nonlinear anisotropic diffusion** effectively reduces the noise level and leads to clear boundaries of different regions.
- For the analysis of cerebral MRI data, diffusion filtering is a frequent initial step.

Diffusion Filtering

- Principle of non-linear anisotropic diffusion. Diffusion may occur along borders and inside homogeneous regions. No diffusion is applied perpendicular to edges.



Filtering at image boundaries

- With both static and dynamic filters, problems occur at the boundary of images where voxels do not have a complete neighborhood.
- For a matrix sized 5×5, the first and the last two rows and columns of an image are involved.
- In typical CT and MRI data, the border of the data contains background information that is not relevant. Therefore, virtual rows and columns filled with the background color can be added.
- If 3D kernels are used, at least the first and last slices of the volume do not have the neighborhood information relevant for filtering, then leave these slices unfiltered.

Gradient filtering

- Gradient filtering is a preprocessing operation useful for subsequent edge detection and edge-based segmentation methods.
- The gradient of a function f is the vector consisting of the partial derivatives in each direction.
- The gradient has a direction (the direction of steepest descent) and a magnitude (*mag*) that is determined as the length of the gradient vector.

$$mag(\nabla f) = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2} \tag{6}$$

Gradient filtering

- With respect to discrete image data, gradients have to be approximated by differences between neighboring voxels (**Finite Differences**).
- To yield an image with gradient magnitude per pixel, a suitable filter must be applied and the output image is known as **Gradient Image**.
- Basic gradient filter. To calculate the average intensity in the local neighborhood in a first step and sum the absolute differences of all pixels in the neighborhood.
- Widespread gradient filters are the **Sobel operator**, the **Roberts cross operator**, and the **Prewitt operator**.

Segmentation

- **Segmentation** is the task of decomposing image data into meaningful structures that are relevant for a specific task. Medical segmentation has two key aspects and a warning:
- Relevant objects should be identified, which means they must be recognizable as particular anatomical structures.
- They should be delineated, in the sense that their borders are precisely specified.
- The challenge is to combine the strength of the user with the potential of computer support.

Segmentation

- Technically, a unique label is assigned to each voxel representing the membership to a particular structure.
- In medical applications, it is usually sufficient to segment the relevant structures only, and not all anatomical structures, i.e. the **segmentation is incomplete**.
- In medical imaging the structures that should be delineated are known as **Target Structures**.

Segmentation issues

- **Significance of Segmentation**. We are primarily interested in segmentation as **prerequisite for visualization**.
- On the other hand, segmentation is sometimes applied to suppress a structure that hampers the visualization. As an example, bones in CT angiography data are often segmented to be removed from a visualization of contrast-enhanced vascular structures.
- Another application of segmentation is **quantitative image analysis**. For example, with respect to gray values, volumes, sizes, or shape parameters of relevant structures.

Segmentation issues

- **Computer Support for Image Segmentation**. Most of segmentation approaches rely on some **homogeneity criteria** that are fulfilled for all voxels belonging to a certain structure.
- This **criterion** might be a certain **range of intensity values**—for example, with respect to the Hounsfield scale if CT data are involved.
- Another example for such a **criterion** is the **existence of a border line** characterized by large gradient magnitude.

Segmentation issues

- Segmentation methods usually compute a map that identifies which voxels belong to a target structure.
- Since this map may contain several segmented target structures it is also called a **label volume**, where each voxel in this auxiliary volume is considered a label of one (or more) segment.
- Approaches for segmentation:
  - **Manual segmentation**.
  - **Threshold- and region-based methods**.
  - **Edge-based methods**, in which the boundary of an object is detected.
  - **Model-based segmentation approaches**

Manual Segmentation

- In manual segmentation the user outlines the relevant structures on slices of radiological data with a pointing device.
- To modify the contour, it is often possible to redraw a particular portion, which replaces the previously drawn portion of the contour.
- This approach is robust (always applicable); however, it is time-consuming, not reproducible, and not precise, because the user often deviates slightly from the desired contour.
- Manual segmentation is widespread, particularly if objects are very difficult to delineate due to low contrasts and an unexpected shape. As an example, tumor segmentation is often performed manually.

Threshold-based segmentation

- A global threshold or an interval of a lower and upper threshold applied to the image intensity permits to generate a binary image.
- Equation (7) specifies the typical situation, in which a threshold interval is defined by an average value a and a tolerance ϵ. All voxel values $I(v)$ that lie within the resulting interval $[a-\epsilon, a+\epsilon]$ are selected.

$$|I(v) - a| \leq \epsilon \tag{7}$$

Threshold-based segmentation

- Threshold-based segmentation can be extended to using multiple intensity intervals $D_1, D_2, \dots D_n$.
- Any kind of threshold-based segmentation may be supported by the presentation and analysis of the image histogram, which may support the selection of thresholds.
- The most typical application of threshold-based segmentation is the **identification of bones in CT data**, as bones can be characterized by large Hounsfield values.
- The accuracy of threshold-based segmentation is limited, since **thin bones** usually are not correctly identified. This is due to the **partial volume effect**, which averages image intensities and leads to lower intensity values for voxels that only partially represent bony structures.

Threshold Selection

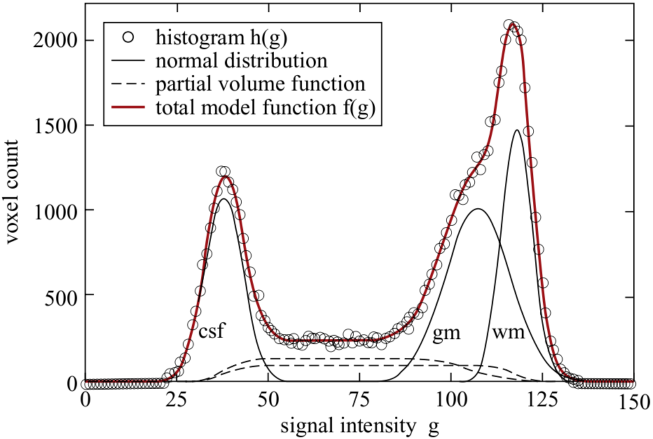
- There are a variety of methods to “suggest” meaningful threshold values. Most of these methods rely on the histogram of image intensities.
- A **local minimum in the histogram** often represents the threshold, which is optimal to distinguish two tissue types.
- The local minimum is a reasonable suggestion if the frequency at this position is low. If the image intensities of two tissue types strongly overlap, there might be no image intensity for which the histogram entry is small.

Threshold Selection

- If **knowledge about the tissue types in the data and their distribution of gray values** is available, threshold selection may be even further supported.
- Example. In cerebral MRI data three tissue types, corresponding to white matter, gray matter and cerebrospinal fluid, are available after segmenting the brain.
- The intensity distribution of these tissue types can be modeled as normal distribution with two parameters μ and α.
- Meaningful thresholds can be suggested by fitting the parameters (μ and α for each of the three tissue types) of the normal distribution to the image histogram.

Threshold Selection

- Fitting normal distributions to the histogram of brain voxels extracted from MRI data. The parameters are optimized by a least square fit.



Thresholding related issues

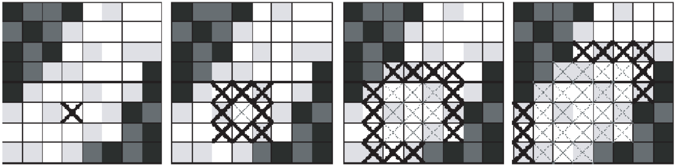
- The idea of threshold-based selection can be extended to derived information such as gradient magnitude, where two intervals for intensity and gradient magnitude are specified.
- A **Connected Component Analysis** (**CCA**) considers the binary image and initializes a first component with the first pixel.
- The algorithm recursively looks for adjacent pixels in the binary image and adds them to this component. If no more connected pixels are found, and there are still pixels that have not been visited, a new component is initialized.
- This process terminates when all pixels are processed and assigned to one region.

Region Growing

- A family of segmentation algorithms is based on the concept of **growing a connected volumetric region**.
- The standard approach of region growing is similar to threshold-based segmentation. Again, a threshold determines which voxels belong to the segmentation result.
- The major difference to intensity-based thresholding is that **one** connected component is considered, which corresponds to the combination of thresholding and the selection of one connected component by means of a Connected Component Analysis.

Region Growing. Basic idea

- The growing process is initiated by one or more **user-selected seed points** and aggregates successively neighboring voxels until a user-selected **inclusion criterion** is no longer fulfilled. In the image, X marks the voxels of the current active wavefront.



Region Growing. Basic idea

- The **inclusion criterion** is usually **a threshold for the intensity values**.
- The user simply specifies seed point(s) that certainly belong to the target structure and the threshold as additional parameter.
- The threshold selection is often a trial-and-error process; the segmentation is started with a certain threshold and then modified until a desired result is achieved.

Region Growing. Threshold selection

- The **problem** with threshold selection is that the user cannot employ his or her expert knowledge on anatomy and radiology.
- Basic region growing is based on a fixed condition, usually one or two thresholds. For the interactive adjustment of these parameters, the basic strategy is too slow.
- A solution approach is to use a **progressive region growing** in which the segmentation is performed for a whole range of thresholds in parallel. This can be done because the segmented voxel set for a higher threshold is completely included in the voxel set for a lower threshold.

Region Growing

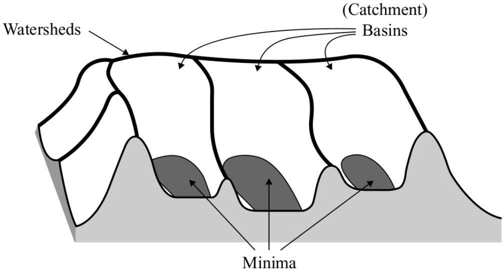
- Region growing is often used for the segmentation of contrast-enhanced vascular structures.
- For tracing vascular structures into the periphery (very thin structures) region growing terminates if at one voxel no neighboring voxel is found that satisfies the homogeneity criterion. This is common due to **partial volume effects**.
- To improve region growing for vessel segmentation, an initial segmentation result with an appropriate global threshold may be generated as a first step and incrementally expanded, guided by seed points specified by the user.

Watershed Segmentation

- Watershed segmentation is based on the idea of regarding an image as a topographic landscape with ridges and valleys.
- The elevation values of the landscape are typically defined by the gray values of the respective pixels or their gradient magnitude.
- Based on such a 3D representation, the watershed transform decomposes an image into regions called **catchment basins** (cuenca hidrográfica). For each local minimum, a catchment basin comprises all points whose path of steepest descent terminates at this minimum.

Watershed Segmentation. Basic idea

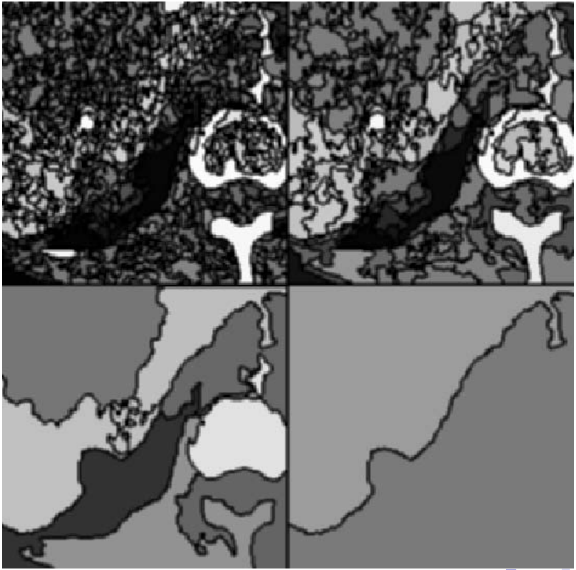
- In watershed the intensity values define hills and basins. **Watersheds** (divisorias de aguas) are the **border lines that separate basins** from each other.
- The watershed transform decomposes an image completely and thus assigns each pixel either to a region or a watershed.
- For segmentation purposes, basins may be flooded in order to combine corresponding regions.



Watershed Segmentation

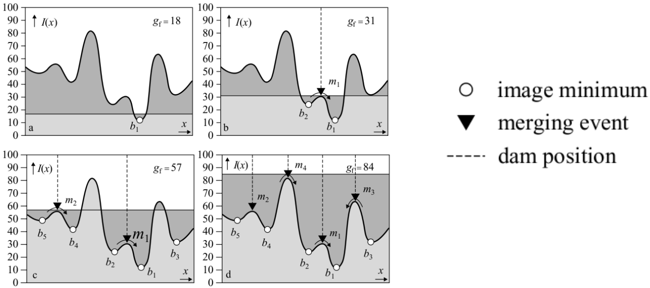
- **Oversegmentation problem**. With noisy medical image data, typically a large number of small regions arise that are much smaller than the anatomical target structures.
- The oversegmentation problem must be solved by using some criteria for merging regions, and the user must be provided with facilities to influence the segmentation result.

Watershed Segmentation



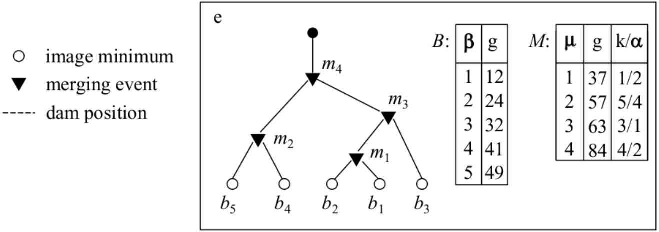
Watershed Segmentation. Merging

- The decomposition of the image into regions is the basis for merging them. The basins are merged at their watershed locations by flooding them. While some regions merge early (with low flooding level), other regions are merged later (with higher flooding levels).



Watershed Segmentation. Merging

- As the intensity values define hills and basins. For segmentation purposes, basins may be flooded in order to combine corresponding regions.
- As result, a **merge tree** arises. We use the **merge tree** to support interactive merging. This tree consists of the original catchment basins as leaves and of intermediate nodes that represent merging events. A merging event is characterized by the nodes merged and by the flood level necessary for merging.



B:

β	g
1	12
2	24
3	32
4	41
5	49

M:

μ	g	k/α
1	37	1/2
2	57	5/4
3	63	3/1
4	84	4/2

Watershed Segmentation. Merging

- Often no examined flooding level is sufficient to segment target structures precisely.
- **Solution: Marker-based watershed** The user may specify image locations that belong to the target structure (include points), or that do not belong the target structure (exclude points).
- If the user specifies an include point and an exclude point, an additional watershed is constructed at the maximum level between them. The watershed prevents the regions represented by the include and exclude points from being merged.

Livewire Segmentation

- While region growing directly generates the region that belongs to the target structure, **livewire** ("**Intelligent Scissors**") and its variant, **live-lane**, are edge-based segmentation methods.
- As a result, the **contours** of the target structure in each slice of a 3D dataset are available.
- Livewire is based on the definition of a **cost function**, and it selects paths with minimal costs.
- Minimal cost paths are computed by Dijkstra's graph search algorithm.

Livewire Segmentation

- Interpretation of an image for Dijkstra's algorithm.
- An image is represented as a graph, with vertices representing image pixels and edges representing costs of connections between neighboring pixels.
- The edges are directed and the orientation is opposite to each other. The "inside" of a directed edge is considered to be the left of this edge. Costs are assigned to every directed edge.
- Intensity to the left and to the right, gradient magnitude and direction, and the Laplacian zero crossing (the approximated second-order derivative of the image data) may be part of the **cost function** computation.

Livewire Segmentation. Cost function

- Equation (8) is a general cost function for the computation of the local cost of an edge connecting the pixels p and q. It is a weighted sum of different components:
- The Laplacian zero crossing $f_z(q)$, which indicates proximity of the pixel q to an edge.
- $f_g(q)$, which represents the gradient magnitude at pixel q.
- $f_d(p, q)$, which represents the gradient direction.

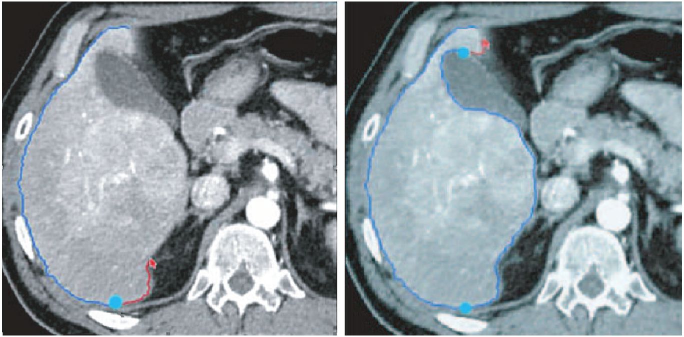
$$l(p, q) = w_z \cdot f_z(q) + w_g \cdot f_g(q) + w_d \cdot f_d(p, q) \qquad (8)$$

Livewire Segmentation. Procedure

- Since the graph is directed, it is essential that the user draws the contour in the same way that the cost function computation assumes (usually counterclockwise).
- Costs are computed from a user-selected control point to the current mouse position. The contour associated with the lowest accumulated cost is highlighted.
- If the path deviates from the target object's boundary, the mouse is moved back until the suggested contour and the desired contour coincide and an additional control point is marked interactively.
- Thus, a first path segment is specified. The second control point becomes the new seed point and is used to define the next path segment. This process is repeated until the boundary is closed.

Livewire Segmentation

- Two stages in the segmentation of an organ with livewire.



Livewire and Shape-based Interpolation

- Livewire is a segmentation approach operated in a slice-oriented way, then is well accepted by radiologists and their technical assistants.
- However, the original livewire approach is still rather laborious for the segmentation of larger 3D objects. Depending on the image resolution and the object size, an object might well be part of 100 or more slices (aprox. 20-30 min to segment).
- Another problem is the **3D slice consistency**, since the defined contours of neighboring slices may deviate significantly.

Livewire and Shape-based Interpolation

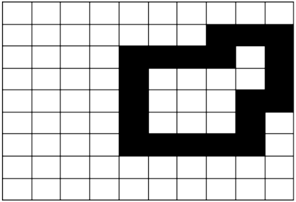
- To reduce the interaction effort and potential slice inconsistencies, livewire has been combined with interpolation methods.
- Interpolation is employed **to skip intermediate slices** and reduce the specification of control points to a subset of **key slices**.
- If the interpolation is carried out for intermediate slices S_1 and S_2 , a binary image of the two slices is generated, with “1” representing the pixels on the contour and “0” representing all other pixels.

Livewire and Shape-based Interpolation

- Based on these binary images, a **distance transform image** is computed (entries represent the Euclidean distance to the object contour with positive values for inside pixels and negative values for outside pixels).
- The interpolation is performed as a gray value interpolation of the distance values. In intermediate slices, the pixels with an assigned "0" are regarded as interpolated contours.
- The selection of key slices should be carried out carefully. If the topology changes between consecutive key slides it is advisable to specify more contours. Topology changes typically occur if, for example, the target structure consists of a closed contour in one slice and two contours in other slices, e.g., in case of **branching structures**.

Livewire segmentation

- A contour (on the left) and its distance transform image (on the right).



4	3	2	2	2	2	1	1	1	1
4	3	2	1	1	1	1	0	0	0
4	3	2	1	0	0	0	0	1	0
4	3	2	1	0	1	1	1	1	0
4	3	2	1	0	1	2	1	0	0
4	3	2	1	0	1	1	1	0	1
4	3	2	1	0	0	0	0	0	1
4	3	2	1	1	1	1	1	1	1
4	3	2	2	2	2	2	2	2	2

Livewire segmentation

- The **result of livewire segmentation** is a stack of contours represented as polygons or parametric curves.
- To use these results for a quantitative analysis as well as for visualization, it is necessary to convert the contours to regions, i.e. **sets of voxels** that represent the inner part of the target structure.
- The transformation of a (closed) contour to a region is carried out by a **filling algorithm**. Since a livewire contour is a (possibly partial) directed line loop, the inside can be determined easily.

Livewire segmentation: pros and cons

- Livewire is a general segmentation method. It works best for compact and large objects that do not have many indentations.
- Livewire may also be used to correct segmentation results, even if they were generated using other methods.
- Livewire is also less suited if the data are rather inhomogeneous, such as MRI data. In such data, a completely different cost function might be required in different regions of an image.

Model-based Segmentation Methods

- These strategies all employ knowledge of the **size** and **shape** of objects or of gray level distributions for the segmentation.
- Characteristic landmarks, symmetry considerations, or typical orientations of the target structure may also be employed.
- Some of these objects have a rather fixed location in relation to other's. This relative location might be employed for the segmentation.

Active contour models (Snakes)

- **Active contour models**, a widespread **variant of** the general approach of **fitting deformable models** to the segmentation target structure.
- Deformable models are based on a flexible geometric representation, such as B-splines, that provides the degree of freedom necessary to adapt the model to a large variety of shapes.
- The process of fitting the model to the target structure is guided by physical principles and constraints, which restrict, for example, the curvature along the boundary.
- The application of **deformable models** is guided by principles of **elasticity theory**.

Active contour models (Snakes)

- Initial contours are algorithmically deformed towards edges in the image and they approximate the shape of object boundaries under the assumption that the boundaries are smooth.
- Active contour models rely on an initial contour, which is either supplied by the user or derived from **a priori** knowledge:
    - Geometric constraints.
    - Data constraints such as range of expected gray level.
    - Object shapes.

Active contour models (Snakes)

- Starting from the initial contour, an **energy function** is minimized based on contour deformation and external image forces. A local minimum based on the initial contour is accepted as a valid solution.
- The energy function with a parametric description of the curve $v(s) = (x(s), y(s))^T$, where $x(s)$ and $y(s)$ represent the coordinates along the curve $s \in [0, 1]$ is described by Equation (9).

$$E_{contour} = \sum_{0}^{1} [E_{int}(v(s)) + E_{ext}(v(s))] ds \tag{9}$$

Active contour models (Snakes)

- The inner energy E_{int} (Eq. (10)) represents the **smoothness of the curve** and encodes expectations concerning the smoothness and elasticity of the target structure's contour. High α values, for example, contract the curve. Usually, α and β are constant.

$$E_{int} = \alpha(s)|\frac{dv}{ds}|^2 + \beta(s)|\frac{d^2v}{ds^2}|^2 \tag{10}$$

Active contour models (Snakes)

- The external energy E_{ext} counteracts the inner energy and is derived by the gray values and the gradient of the image according to Equation (11).
- w_1 and w_2 are weights that represent the influence of the gray value $(f(x, y))$ and the gradient $\nabla(G)$. The gray values are assumed to be normally distributed. The σ value characterizes the standard deviation of this distribution.

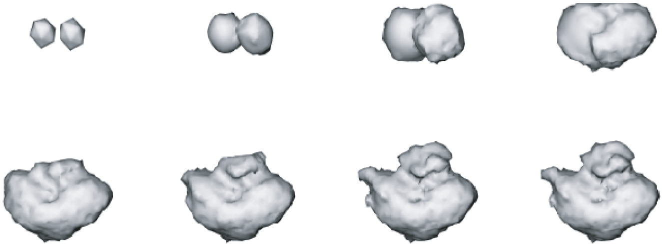
$$E_{ext} = w_1 f(x, y) - w_2 |\nabla(G_\sigma(x, y))|^2 \tag{11}$$

Active contour models (3D Snakes)

- **Balloon segmentation** is the extension to 3D of snakes and is based on deforming surfaces instead of contours.
- This is accomplished by interactively fitting a polygonal representation of the initial shape to the target structure. There are two approaches to initiating the segmentation process:
    - The user selects small volumes that are iteratively inflated until the forces converge.
    - The user specifies enclosing volumes that are iteratively deflated.

Active contour models (3D Snakes)

- This image illustrates the inflation approach with an application to tumor segmentation.



Level Sets

- If we consider region growing as a dynamic process in which the progressing boundary can be regarded as a **wave front** propagating through the target object, then we understand the key idea of level sets.
- The **wave propagation** is guided by image features such as image intensity and gradient.
- Level sets could be considered an implicit formulation of deformable models. However, the contour is not manipulated directly.
- The contour is embedded as the zero level set in a higher dimensional function, the level set function $\Psi(X, t)$.

Level Sets

- The level set function is evolved under control of a partial differential equation. The evolving contour can be determined by extracting the zero-level set.

$$\Gamma((X), t) = \Psi(X, t) = 0 \tag{12}$$

- Applying Equation (12) determines all points at height 0 of the embedding function.
- Level set segmentation is able to handle complex anatomical shapes with arbitrary topology.

Postprocessing of Segmentation Results

- **Threshold-based and region-oriented** segmentation methods in particular often produce results that need some postprocessing to be accurate. For example, holes within a segmentation result that should be filled.
- There are typically two tasks in the postprocessing stage:
  - To correct minor errors in the segmentation output by **adding and removing voxels**.
  - To achieve high-quality visualizations of the segmentation result: the segmentation result should be enhanced to support **smooth visualizations**. The goal here is not to add or remove voxels completely but to transform the binary segmentation result to a multivalued segmentation result.

Morphological Image Analysis

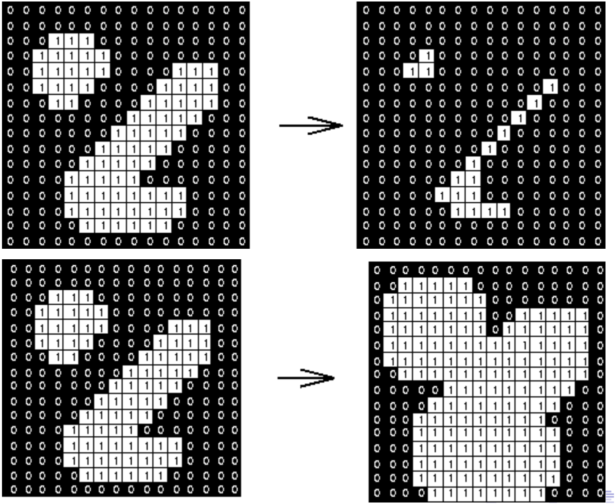
- Morphological operators add or remove voxels inside or from the border of the segmentation.
- This filters operate on binary image data (1 represents a voxel that is part of the segmentation result and 0 represents other voxels).
- Similar to noise reduction filters, morphological filters are described by a kernel matrix with a certain size. The elements, however, are restricted to 1 or 0 to produce binary results.

Morphological Image Analysis

- The simplest morphological operators are **erosion** and **dilation**.
- **Erosion** with a 3×3 kernel, with all elements set to 1, removes all pixels with a neighbor pixel outside the segmentation result and, hence, shrinks the segmentation. Typically, erosion is used to remove erroneous (“false”) connections between regions.
- **Dilation** grows the segmentation (thus closes holes and bumps in the boundary) if at least one voxel below the kernel is set.

Morphological Image Analysis

- Two examples of erosion (top) and dilation (bottom) operators.



Morphological Image Analysis

- Since erosion and dilation reduce or increase the segmentation result, **opening and closing operators** are introduced that maintain the overall size.
- **Opening** consists of an erosion and a subsequent dilation. Note that the image really changes via these two operations; dilation is not exactly the inverse transformation of erosion. Opening is employed to remove small objects and to reduce convex bulges.
- **Closing** dilates the segmentation result first and erodes it subsequently. Closing fills concave notches (muescas o cortes en V) and closes small holes.

Morphological Image Analysis

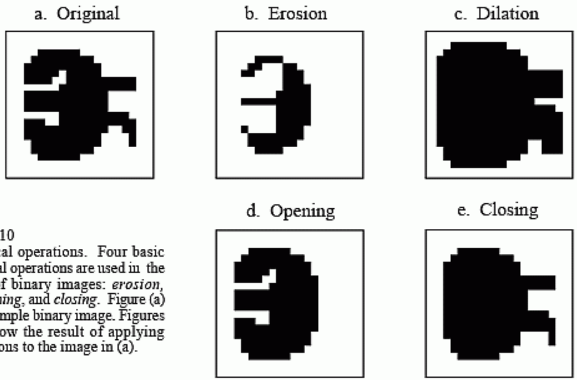


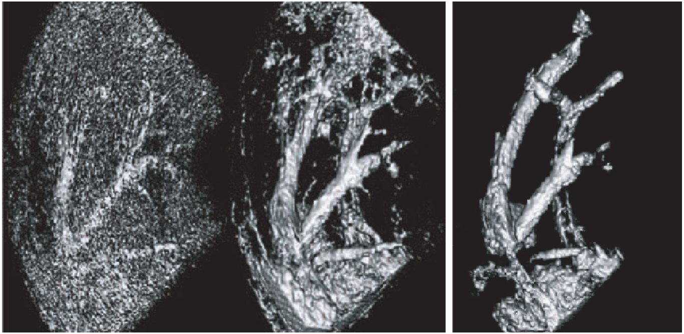
FIGURE 25-10
Morphological operations. Four basic morphological operations are used in the processing of binary images: *erosion*, *dilation*, *opening*, and *closing*. Figure (a) shows an example binary image. Figures (b) to (e) show the result of applying these operations to the image in (a).

Noise Reduction with Morphologic Operators

- Morphological operators can be useful as part of the **noise reduction process**.
- In particular, morphologic operators are useful if the data contain not only stochastic noise but also speckles of a certain size that need to be removed for further analysis and visualization.
- Iterative erosion may be used to separate regions that contain the relevant information from a noisy environment.

Noise Reduction with Morphologic Operators

- 3D visualization of ultrasound data from vascular structures in the liver. From left to right: original, after 20 iterations of an erosion and after 30 iterations.



Smoothing Segmentation Results for Visualization

- **Smoothing** attempts to improve the appearance of visualizations based on the segmentation results.
- Usually, 3D binary segmentation results are used as input for an **isosurface visualization**.
- The **segmentation result** is characterized by a data structure called a **segmentation or label volume**. One value represents voxels belonging to the segmentation mask and a second value represents background voxels.
- The intermediate value is used as isovalue for the surface generation.

Smoothing Segmentation Results for Visualization

- The simple isosurface generation suffers from strong **aliasing artifacts**, which become obvious as discontinuities in the surface normals. These artifacts are due to the discontinuity of the inside-outside function.
- **Solution**. To "smooth" the boundary in the segmentation result at the voxel level.
- We will show a simple algorithm that is specially indicated for isosurface extraction.

Smoothing Segmentation Results for Visualization

- We denote with v_1 the value of the segmentation result and with v_2 the value of the background voxels. t_{iso} represents the isovalue computed as $(v_1 + v_2)/2$.
- The method is based on morphologic operations performed in a defined distance of the original object boundary.
- The morphologic operations are constrained in a way that maintain the original in-out classification of the segmentation step.
- Each voxel near the object boundary is assigned a new value v, with $v_1 \leq v \leq v_2$ using the following algorithm (See next slide).

Smoothing Segmentation Results for Visualization

- A reference mask V_{ref} is created by eroding the segmentation result. After the erosion, all voxels of V_{ref} are assigned the value v according to Equation (13), which in essence moves the isosurface closer to the background value and thus farther away from the original segmented object.

$$v = v_2 - (v_2 - v_1)\frac{1}{3} \tag{13}$$

Smoothing Segmentation Results for Visualization

- Two dilation operations are performed. The boundary voxels of the reference mask are tracked as reference voxels with the dilation front. Each voxel v that is added through dilation is therefore associated with a reference voxel $v_{ref} \in V_{ref}$ and acquires the value v

$$v = v_{ref} - (v_2 - v_1)\frac{d}{3} \tag{14}$$

with d being the Euclidean distance from v to v_{ref} and v_{ref} the value of the associated voxel from V_{ref}.

Smoothing Segmentation Results for Visualization

- **Image on the next slide**.
- **Top row**: Smoothing binary segmentation results (left) through morphologic operators and with $v_1 = 0, v_2 = 100, and t_{iso} = 50$. The dark gray voxels in the left part (middle image) would be removed by an erosion. This is avoided to preserve the object shape. In a last step, voxels that belong to the segmentation result and have a value below 50 are corrected (right image).
- **Bottom row**: A real world example. Bright red represents the original label value v_2 of the segmentation, dark red the smoothed voxel value. White indicates the isosurface. Binary input segmentation (left image), smoothing after erosion (middle image), smoothing after erosion with correction (right image).

Smoothing Segmentation Results for Visualization

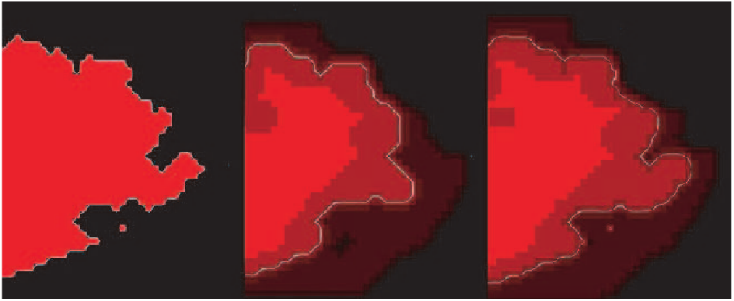
0	0	0	0	0	0	0
0	0	0	100	100	100	0
0	100	100	100	100	100	0
0	100	100	100	100	100	0
0	0	0	100	100	100	0
0	0	0	0	0	0	0

→

0	6	26	33	33	26	6
20	33	54	66	66	54	26
33	66	66	100	100	66	33
33	66	66	100	100	66	33
20	33	54	66	66	54	26
0	6	26	33	33	26	6

→

0	6	26	33	33	26	6
20	33	49	66	66	54	26
33	66	66	100	100	66	33
33	66	66	100	100	66	33
20	33	49	66	66	54	26
0	6	26	33	33	26	6



Registration and Fusion of Medical Image Data

- **Image registration** is the process of aligning images so that corresponding features can easily be related.
- In many clinical scenarios, it is crucial to mentally combine information extracted from different sources to draw conclusions. Registration is essential to compare image data and to analyze different image data in a **common frame**.
- The image data to be analyzed and compared often relate to the **same patient but** are acquired at **different points in time**, or they are acquired with **different imaging modalities**.

Registration and Fusion of Medical Image Data

- **Registration of image data acquired at different points in time**. In dynamic imaging, image data are acquired at different time points. Follow-up studies after initial treatment involve the acquisition of images at various stages, such as three, six, and twelve months after a therapy. The evaluation of these images involves a comparison, for example, with respect to tumor growth.
- **Registration of pre- and intraoperative data**. Therapy monitoring is based on intraoperative imaging. It is desirable to relate intraoperative data to analysis results derived from preoperative data.
- **Multimodal registration**. A wide area of application is multimodal image registration, in which different acquisition techniques, such as CT and MRI, are used complementarily.

Registration and Fusion of Medical Image Data

- These application areas have in common the need to geometrically adapt several images to each other or to some kind of model.
- **Goal of image registration**. To deform or transform one dataset to optimally match another dataset that provides different information as well as similar information.
- The dataset to which a new dataset is adapted is referred to as **reference data**.

Registration and Fusion of Medical Image Data

- **The Registration Problem**: “**Transform a floating image dataset geometrically so that it fits optimally to a given reference image under a given aspect**.”
  - **Transformation**: geometric transformation of voxel coordinates.
  - **Fitting**: requires a quantification by means of a **similarity measure**.
  - **Optimally**: The transformation should be accomplished in such a way that the similarity measure is maximized.
  - **A given aspect**: The **criteria for optimal matching** are chosen such that particular structures are matched as good as possible.

Registration: Transformation

- **Transformation**. “Global” and “local” transformations are discriminated.
- **Global Transformations**. Translation and rotation of **all** coordinates are examples for global transformations. These transformations are described by a **small set of parameters** that is applied to all coordinates. The modification of one parameter has an influence on all voxels.
- **Local Transformations**. These transformations are described by a **large set of parameters** that correspond to a **mesh of control points**. Modifications of a single parameter (control point) affect only a local neighborhood.

Registration: Transformation

- **Comparison: Global versus Local Transformations**.
- A global transformation can be used to compensate for simple movements, while local movements can in principle account for any complex movement.
- The computational effort for a global transformation is low: a matrix-vector operation is carried out for each coordinate. In contrast, elastic transformations are computationally highly demanding.
- In general, a global transformation is used as a preprocesssing step for a local transformation.

Registration: Fitting

- **Similarity measures** characterize how similar two images are. Basically, similarity measures based on intensities of voxels and on geometric measures are discriminated:
- **Intensity-based similarity**. Gray values of voxels in the floating image are compared with voxels in the reference image.
- **Geometry-based similarity**. Positions of voxels in the floating image are compared to those in the reference image.

Registration: Discussion

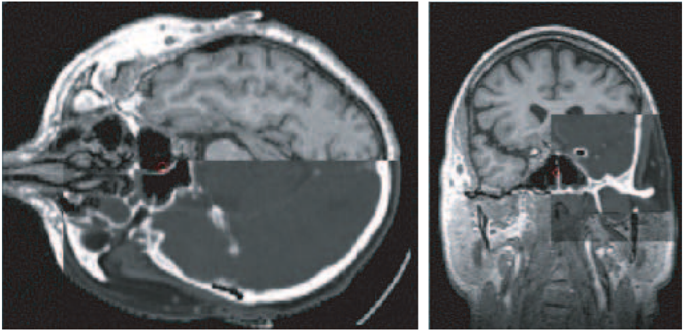
- **Local registration processes are often guided by some constraints**. For example, the volume of certain tissue, in particular a tumor, should be preserved. Without constraints, arbitrary deformations may arise, resulting in misleading visualizations.
- **Local registration is a time-consuming process**. Several attempts have been made to accelerate local registration by employing graphics hardware, parallel computing or a priori knowledge relating to the specific question.
- In the clinical routine, **landmark-based registration** is primarily used. This is probably due to the fact that with this kind of registration it is relatively easy to control which regions of the target image are mapped to certain portions of the reference image.

Integrated Visualization of Registered Image Data

- Once the target image is transformed to the reference image, both image data can be explored in an integrated visualization.
- The integrated visualization also serves as a visual control of the registration quality.
- Integrated visualizations may be explored in slice-based views (next slide) or in 3D visualizations (next next slide) where clipping planes might be used to restrict which portions are visible from the reference image and from the target image.

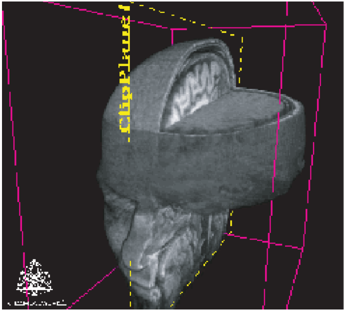
Integrated Visualization of Registered Image Data

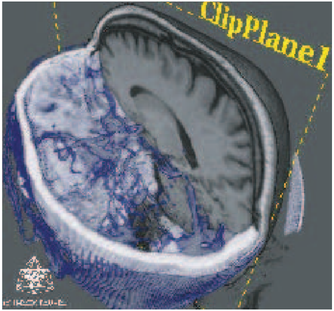
- Fusion of T1-weighted MRI and CT angiography data.



Integrated Visualization of Registered Image Data

- Fusion of T1-weighted MRI and MRI angiography data (left). Fusion of T1-weighted MRI with CT angiography data.



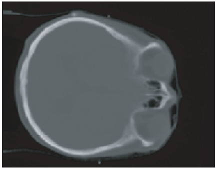
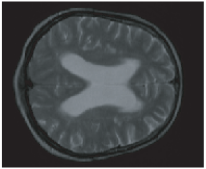


Integrated Visualization of Registered Image Data

- In both 2D and 3D visualization, movable and **scalable lenses** are frequently used to control which portions of the data are visible.
- Neurosurgical interventions are an example where multimodal information (in particular CT and MRI data) are frequently acquired.
- The registration and integrated visualization of both image data provide an overview on soft tissue structures within the context of skeletal data.

Integrated Visualization of Registered Image Data

- The fusion of skeletal structures from CT (left) and cerebral soft tissue from MRI data (right) allows to generate integrated 3D visualizations highlighting the cerebrospinal fluid inside the skull.


Integrated Visualization of Registered Image Data

- Fusion of PET and CT data to convey the location of a neck tumor. The tumor is visualized based on its characteristic signal in PET data, while the skeletal structures are extracted from CT data.