Computer- and robot-assisted Surgery

NCT
NATIONALES CENTRUM
FÜR TUMORERKRANKUNGEN
PARTNERSTANDORT DRESDEN
UNIVERSITÄTS KREBSCENTRUM UCC

getragen von:
Deutsches Krebsforschungszentrum
Universitätsklinikum Carl Gustav Carus Dresden
Medizinische Fakultät Carl Gustav Carus, TU Dresden
Helmholtz-Zentrum Dresden-Rossendorf

Lecture 5

Segmentation 2

Segmentation

- Point-based
  - Operations only on grey values
  - No global considerations
  - e.g.: Threshold methods

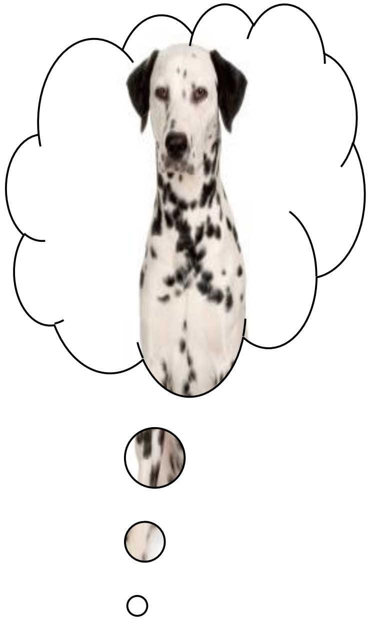
- Region-based
  - Every area of a region fulfills a certain homogeneity critera
  - e.g.: region growing

Segmentation

- Edge- and contour-based
  - Object has a clear edge
  - Goal: extraction und merging of edges
  - e.g.: Active Contours, Snakes

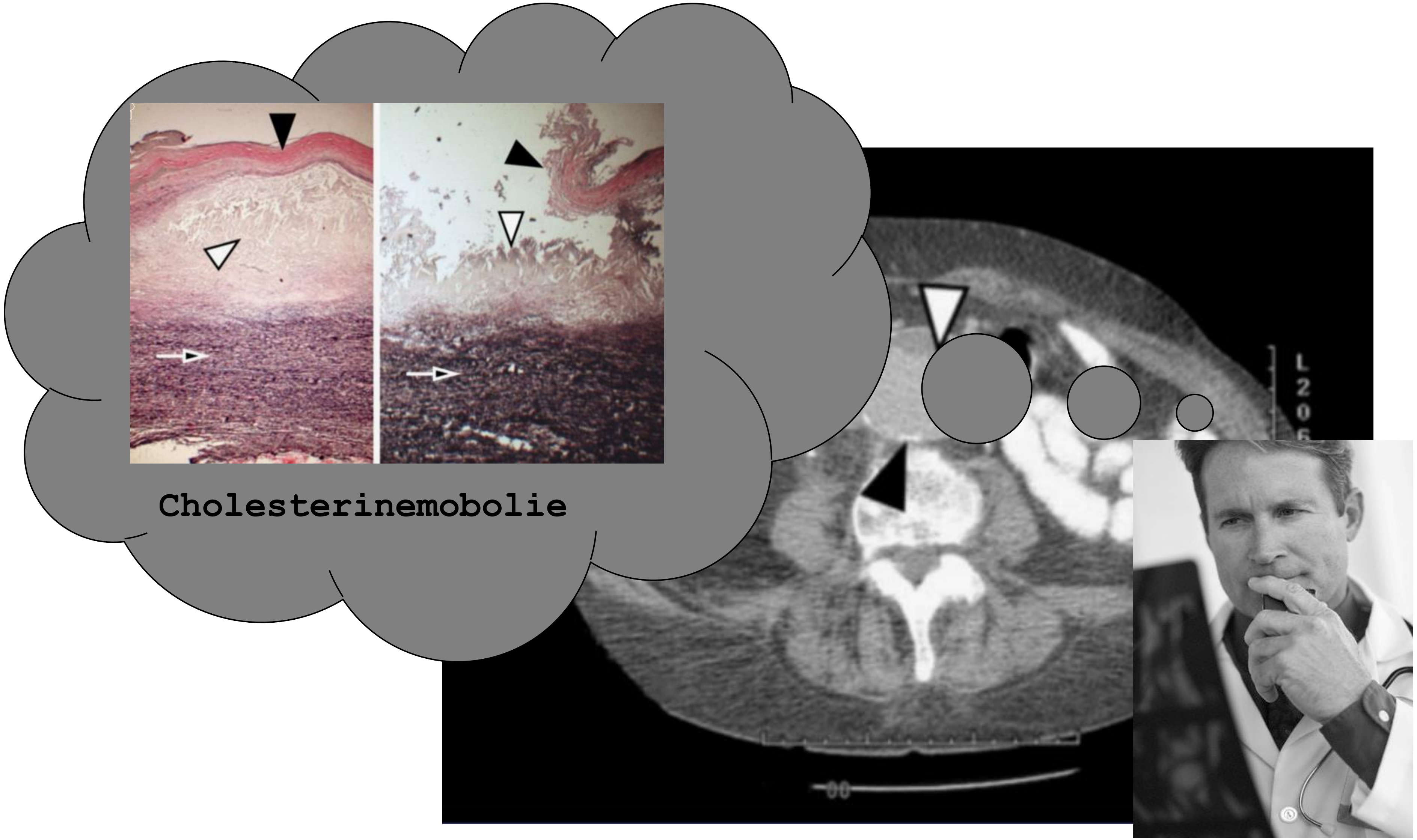
- Knowledge-/ model-based
  - Integration of problem specific a-priori-knowledge
  - Goal: Enhancement of segmentation by only considering „plausible“ results
  - z.B.: Point Distribution Models…

Introduction

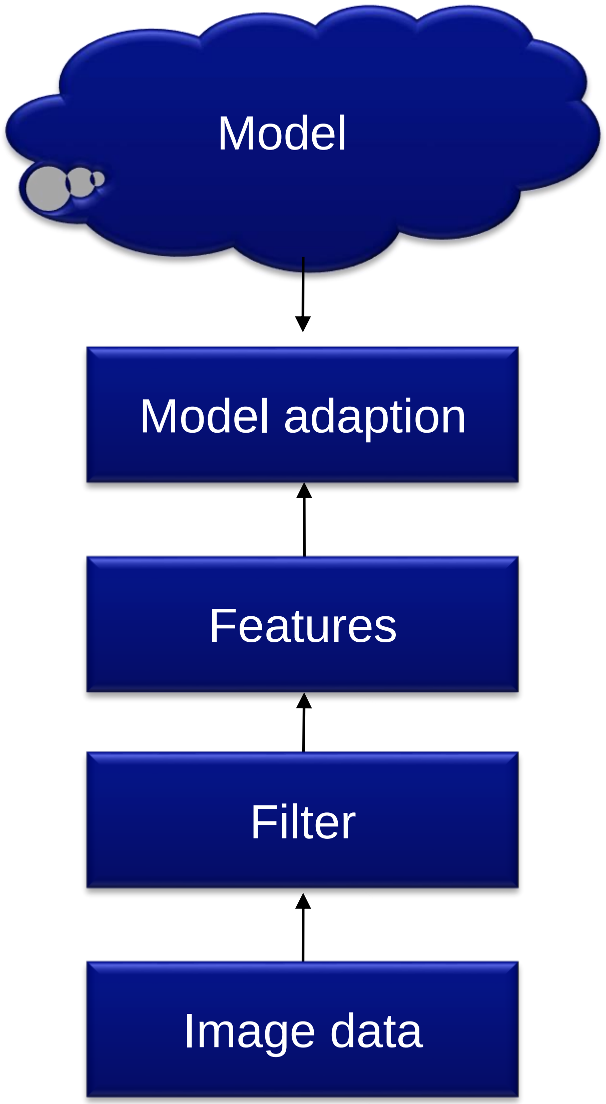


Look for a
Dalmatiner

Introduction



Model-based segmentation



Model-based segmentation

- What does model-based mean?
  - So far: only local information without knowledge

  - Now:
    - Consideration of the shape of the object
    - Simple geometric models (Lines, Ellipsoids…)
    - Deformable Models
    - Statistical Models

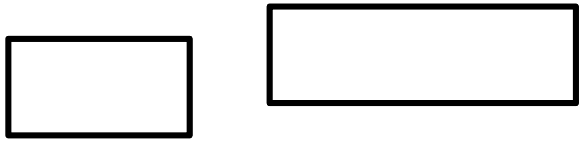
  - Goal: find model parameters and position/orientation in the image

What does „shape“ mean?

- „Configuration of a set of points that are invariant against specific transformations (Cootes 2004).

$$\mathrm{x} = (\mathrm{x}_0, y_0, z_0, \ldots, x_n, y_n, z_n)^T$$

- Geometric primitives: Circle, Square, Triangle…



same shape

Different shape

- Parametric shape: Contours/planes that can be presented via a function

HOUGH-TRANSFORMATION

Hough-Transformation

How can you detect known geometrical objects in an image?

Solution with Hough-Transformation

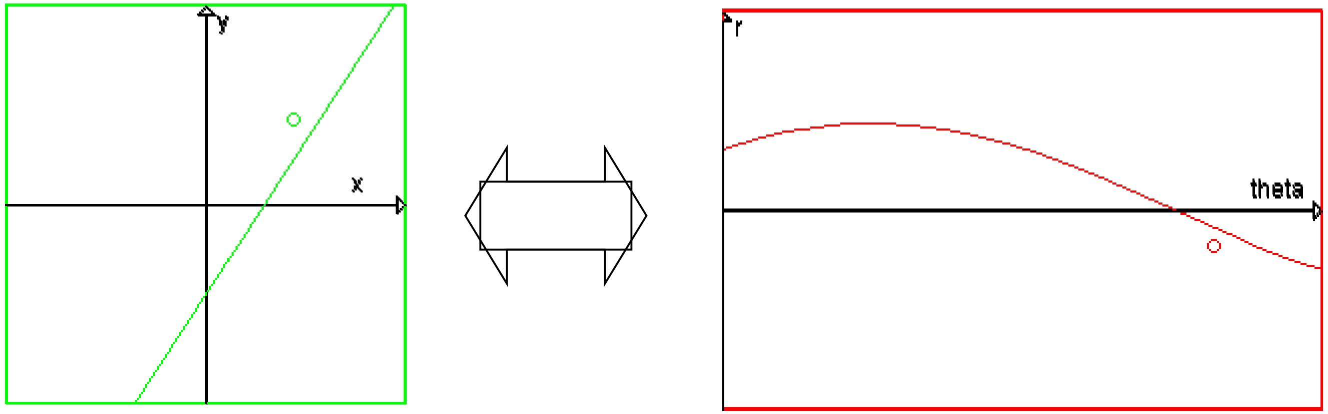
- Definition of a dual parameter space for an image:
  - Point in parameter space corresponds to an object in the image
  - Dimension parameter space = number of parameters to describe the object
- Transformation of all possible pixels on an edge into the parameter space
- Search for maxima in parameter space
- Backprojection in image space

Hough-Transformation: Detection of lines

- Definition of lines in Hessian normal form:

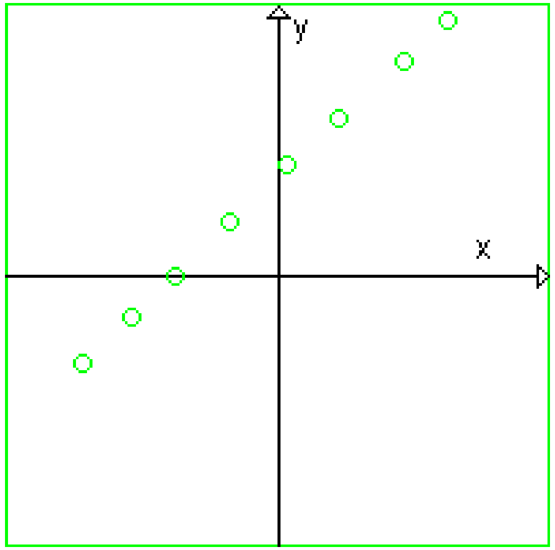
$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

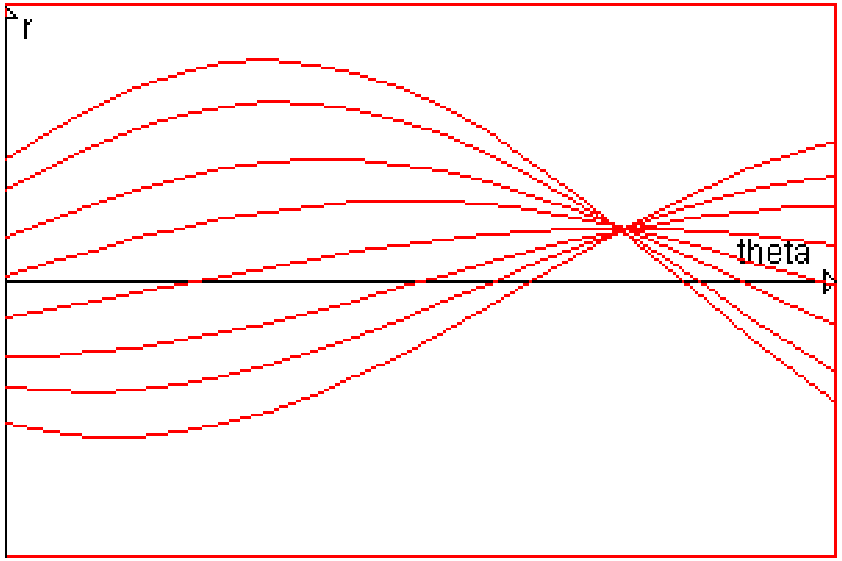
- Pixels of edges correspond to sinusoid in parameter space
- Point in parameter space corresponds to line in image space



Hough-Transformation: Detection of lines

- Realisation of the parameter space as accumulator array
- For every pixel on an edge:
  - Increase accumulator cell on the corresponding sinusoid
- Line corresponds to maxima in parameter space

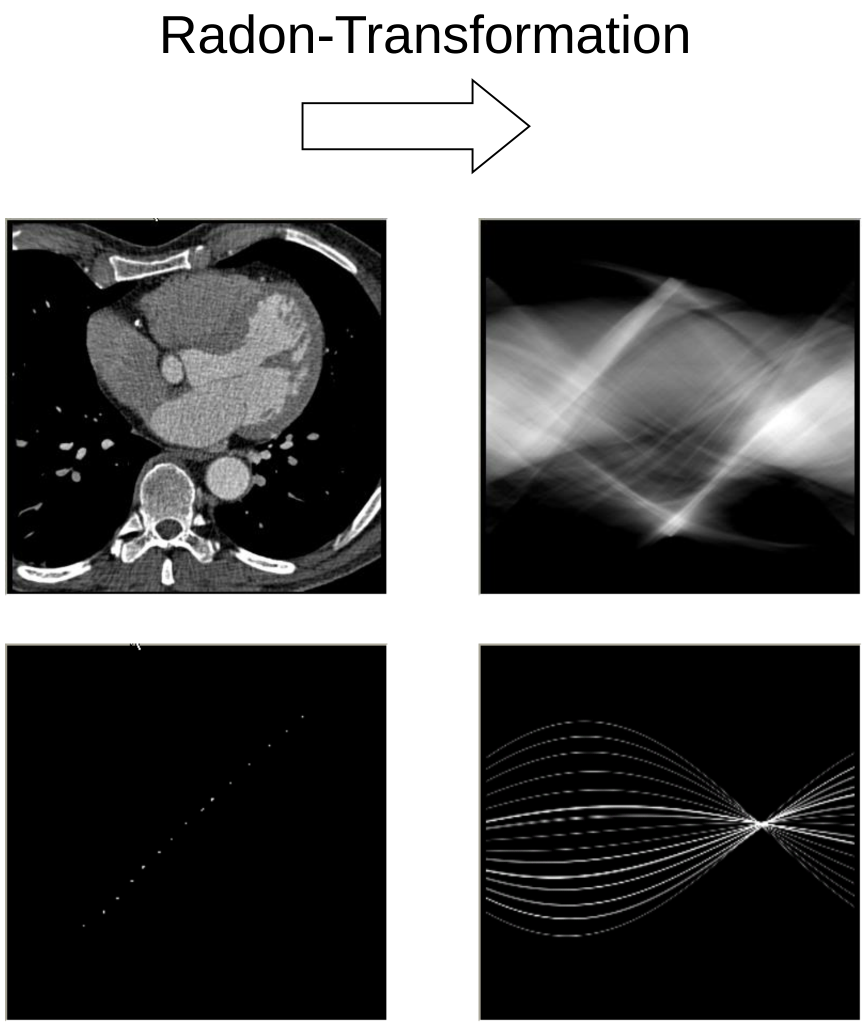




Analogy Radontransformation

- Hough-Transformation is the discrete analogon of the Radon-Transformation.

- The line integral is substituted with the sum of the edge pixel



Hough-Transformation: Detection of circles

- Extension of the Hough-Transformation for lines to other parametric shapes possible

Detection of circles

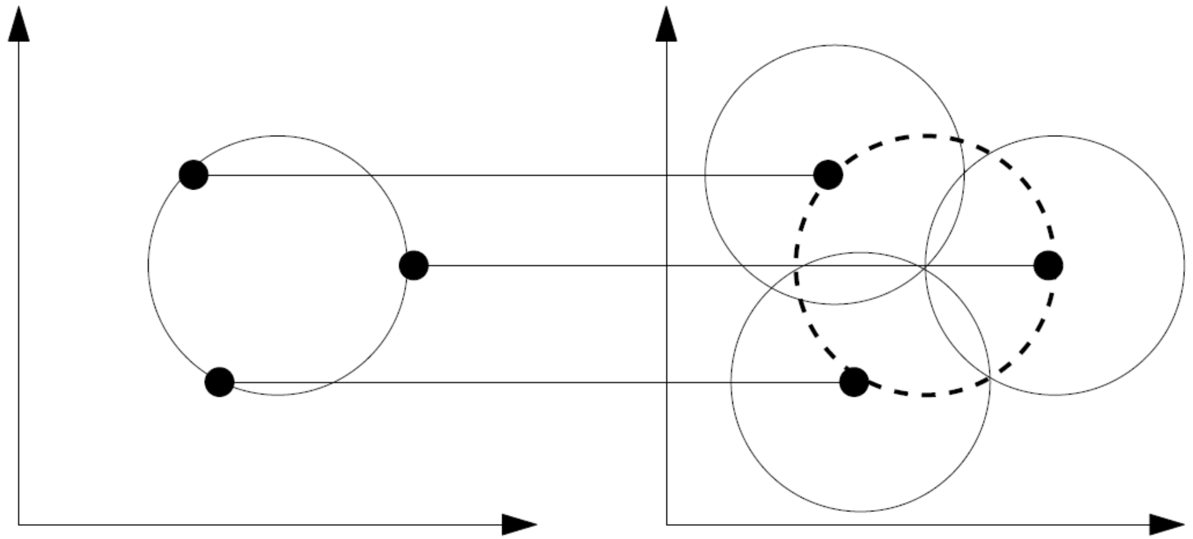
- Circle equation:

$$\mathrm{r}^2 = x^2 + y^2$$

- unknow r => 3D parameter space (x,y,r)
- For circles with known radius r - analog to line transformation

Hough-Transformation: Detection of circles

- For every edge pixel:
  - Increase accumulator cell on the corresponding circle
- Circle is maxima in parameter space

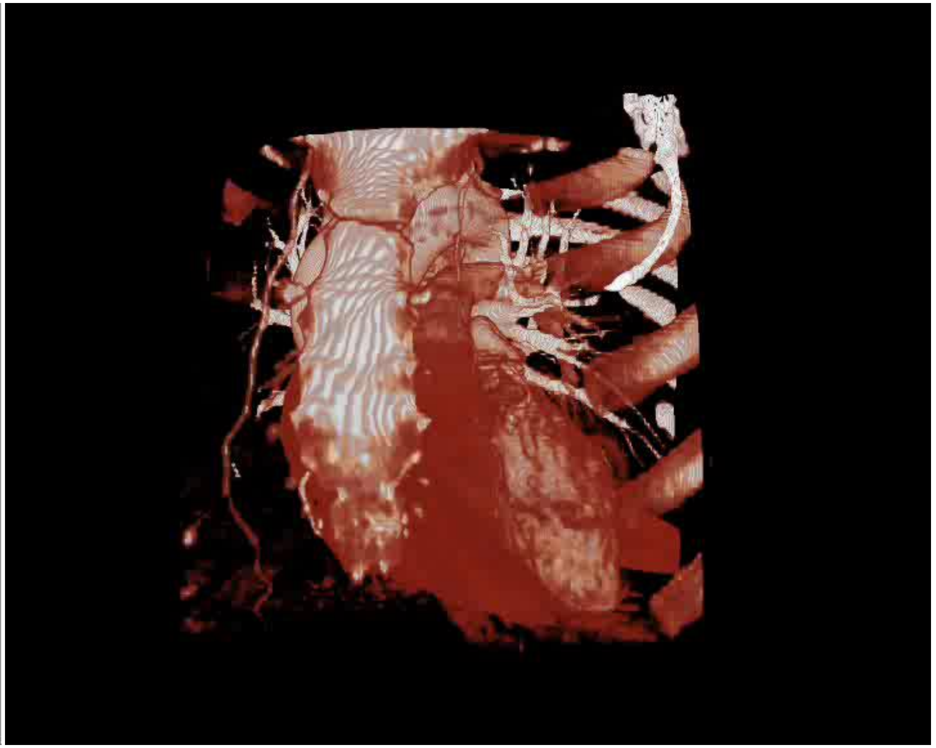


Hough-Transformation: Conclusion

- Voting method for detection of parametric objects in binary edge images
- Lines: Analogies to Radon-Transformation
- Dimension of Hough space rises with numbers of parameters

→ High complexity

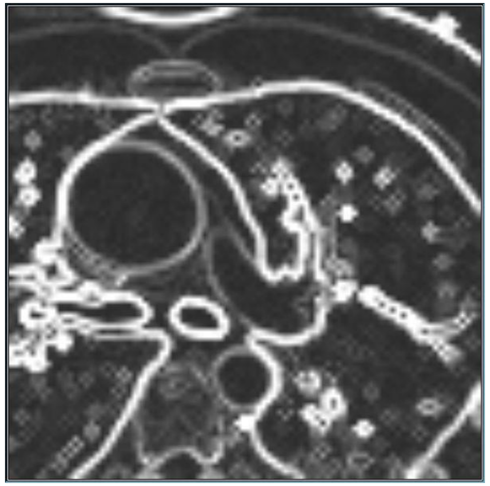
Practical example: Detection of Aorta in CTA image data



Detection of Aorta in CTA image data

Detection of Aorta in CTA image data

Edge filter:

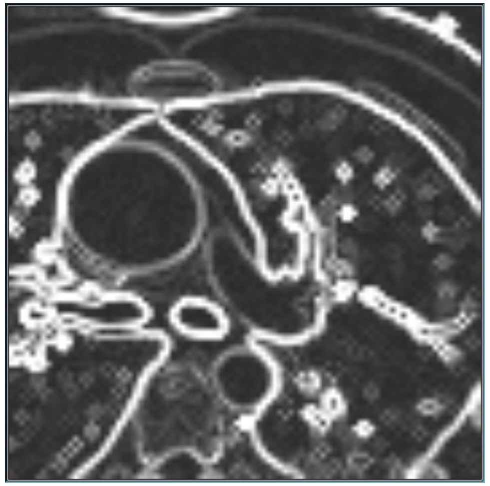


$$G_y = I * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad G_x = I * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G = \sqrt{\mathbf{G_x}^2 + \mathbf{G_y}^2}$$

Detection of Aorta in CTA image data

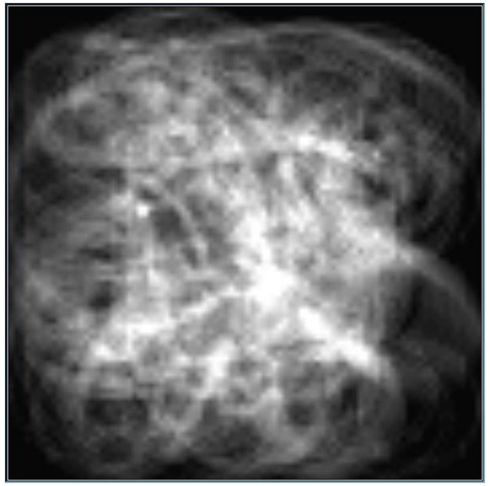
Binary edge image:



$$B(x,y) = \begin{cases} 1, & \text{wenn } |G(x,y)| > t \\ 0 & \text{sonst} \end{cases}$$

Detection of Aorta in CTA image data

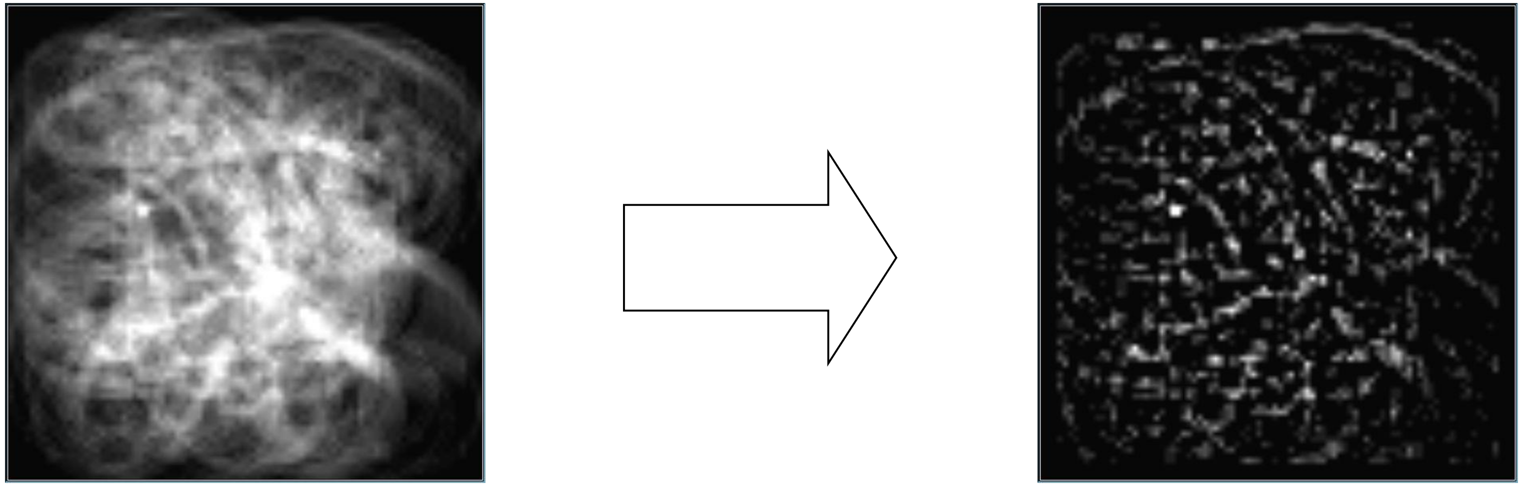
Hough-Transformation:



- Hough-Transformation for circles
- Radius is not know a-priori nicht:
  - Separate Hough-Transformation for different Radi

Detection of Aorta in CTA image data

Hough-Transformation filtering:

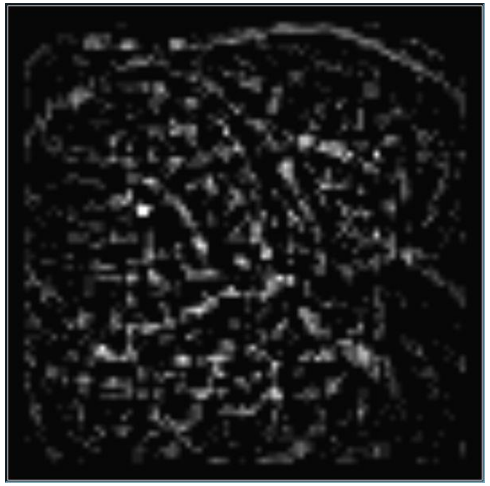


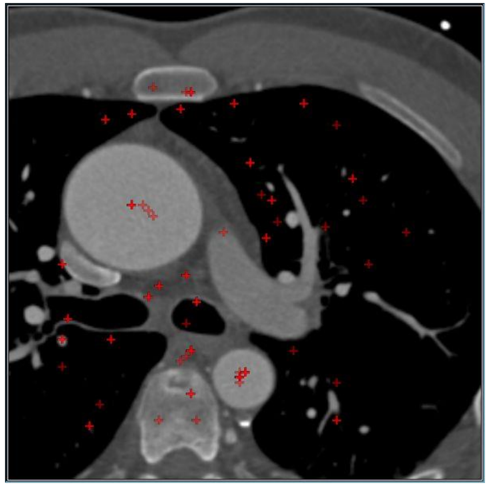
- Highlight Peaks
- Difference to the local average:

$$\mathrm{H}_{\mathbf{filtered}} = H * \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Detection of Aorta in CTA image data

Detection of Hough-Maxima:

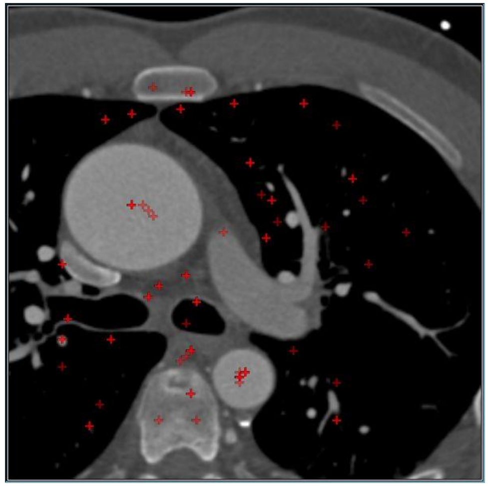


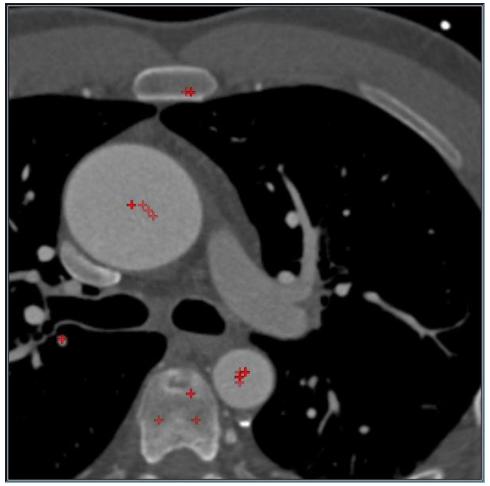


- for all Radi

Detection of Aorta in CTA image data

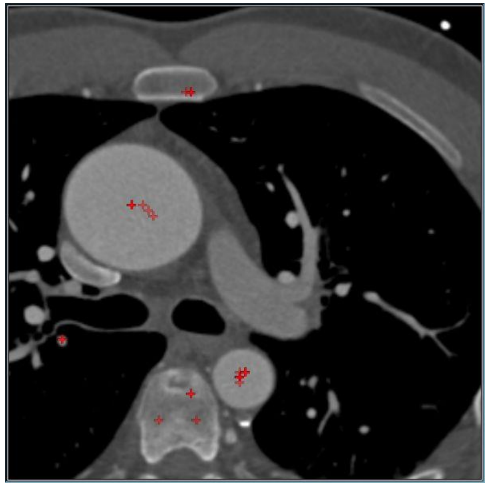
Begrenzung auf plausible Grauwerte:

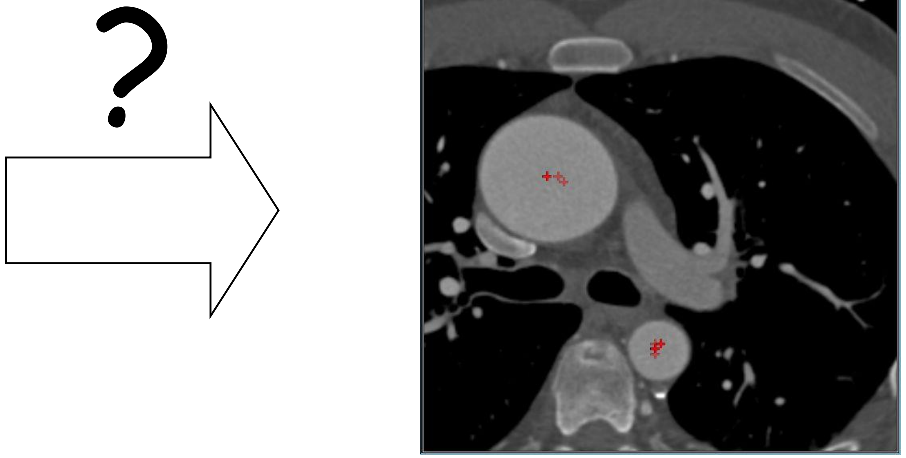




- Limitation of the candidates according to grey value (contrast agent)

Detection of Aorta in CTA image data





- Solution: Comparison with ellipsoid model

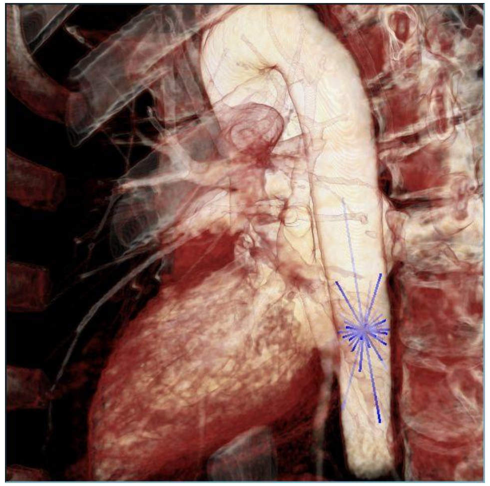
Detection of Aorta in CTA image data

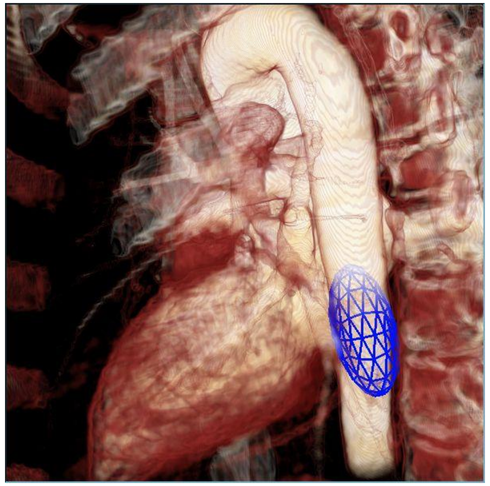
- Sample surrounding with spherical search rays

- Identify vessel contour

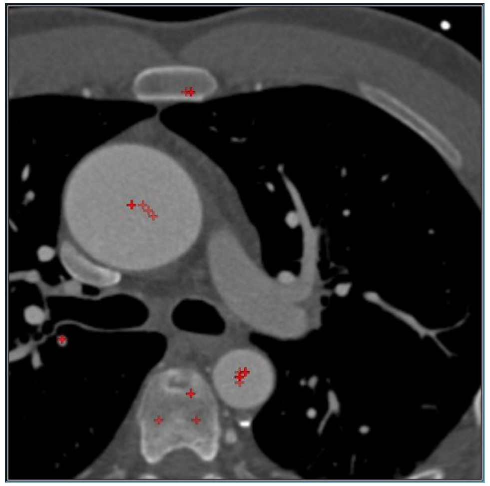
- Principal component analysis of the detected contour points:
  - Eigenvectors and Eigenvalues of the covariance matrix

- Approximation of the ellipsoid is calculated via eigen-vectors and eigen-values





Detection of Aorta in CTA image data



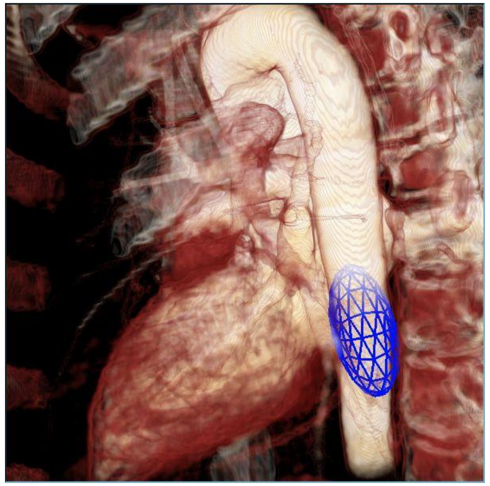
- Comparison with ellipsoid model:
  - Excentricity close to zero
  - Radius of the ellipsoids must be similar to radius of the Hough-Transformation

Detection of Aorta in CTA image data

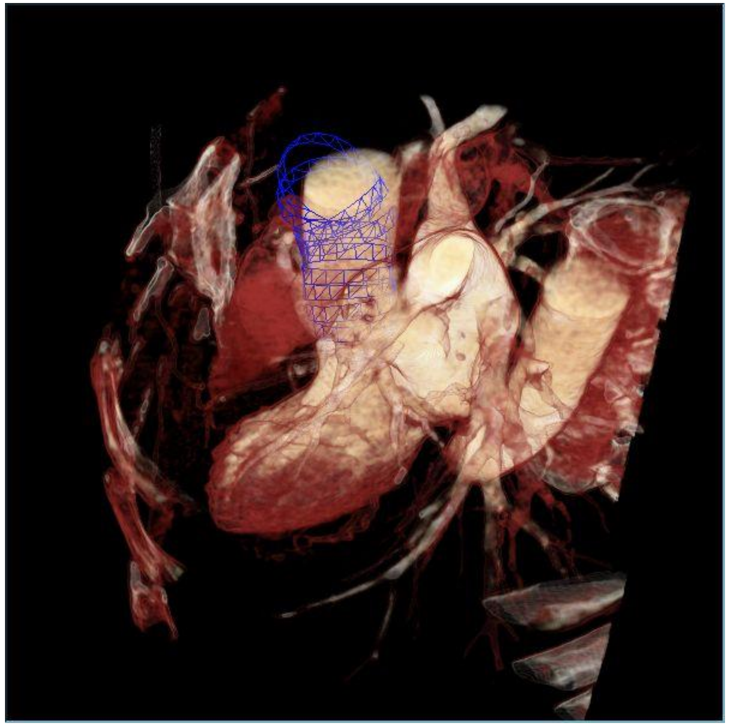
Cylinder chain model of the Aorta

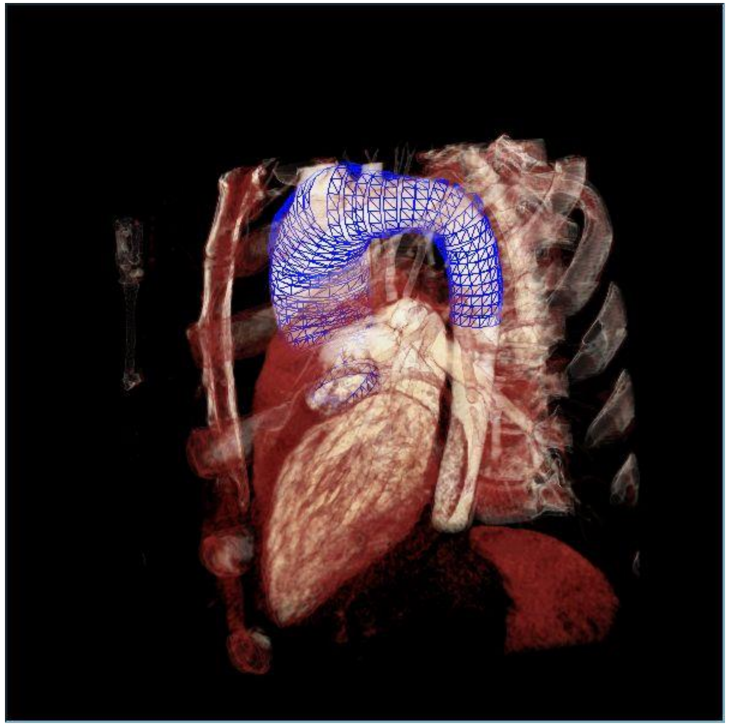
- Starting with Hough-Maxima:
  - Push the ellipsoid model
  - Build a cylinder segment chain

- Selection of the correct chain depending on
  - (Aortic)-arc included?
  - Length of the chain
  - Radius maxima?



Detection of Aorta in CTA image data



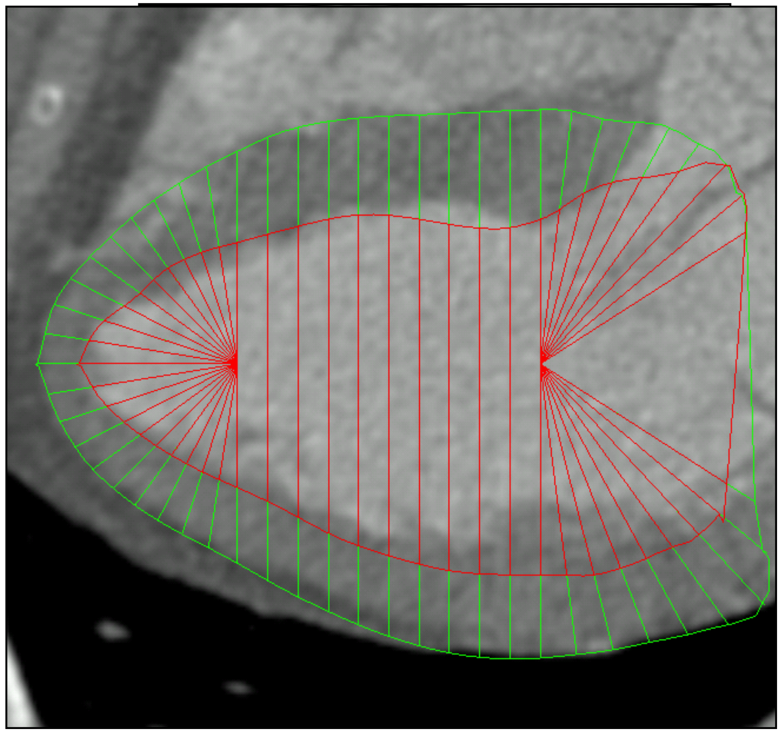


POINT DISTRIBUTION MODELS

Point Distribution Models

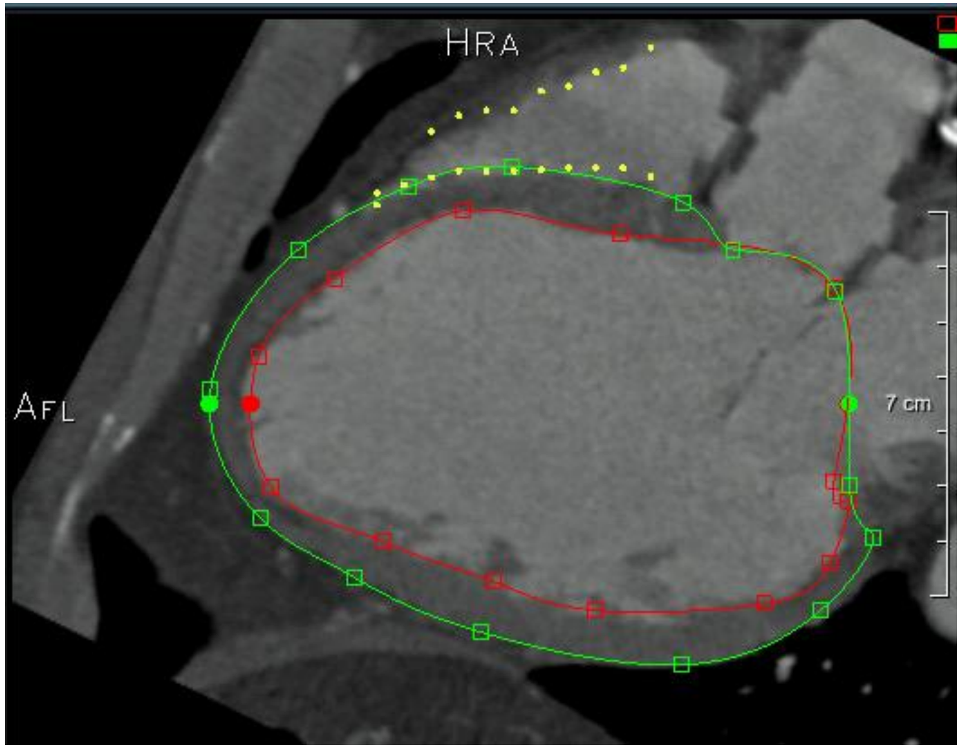
- Description of shape and shape variance of an object (Cootes et al. 1999)
- Based on different training dataset corresponding landmarks are learned
- Calculation of the average shape
- Modeling of possible deformation

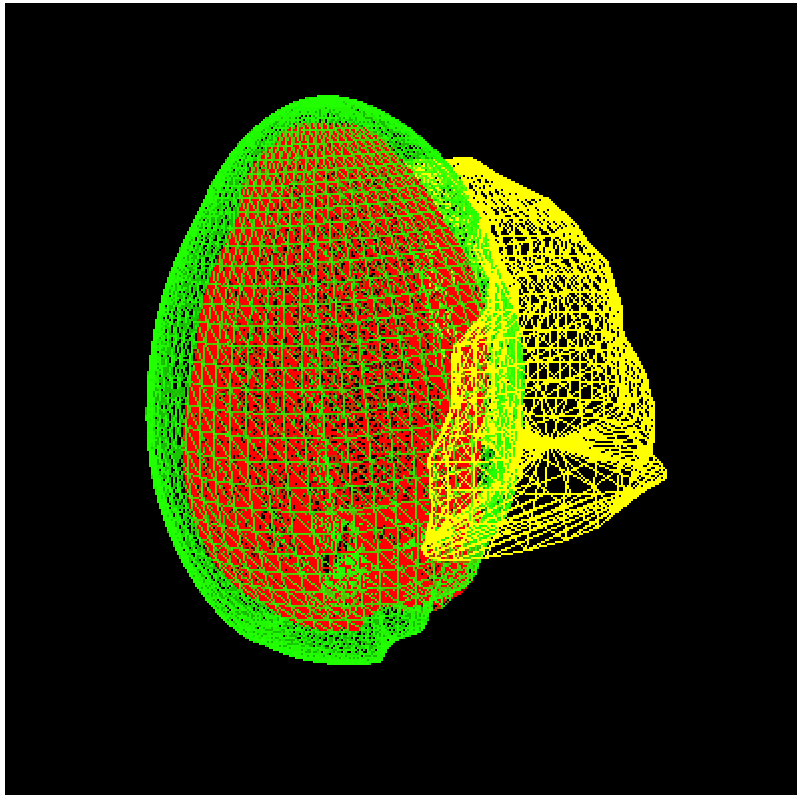
Point Distribution Models – Corresponding Landmarks



- 1:1 Correspondence is a basic requirement for statistical shape models

Point Distribution Models – Corresponding Landmarks

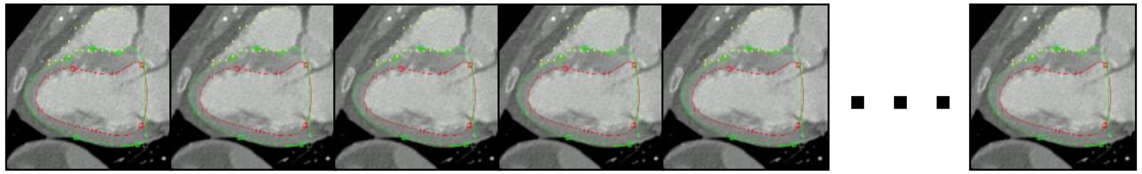




- 1:1 Correspondence is a basic requirement for statistical shape models

Point Distribution Models

- Calculation of *m* shapes.



- Each with *n* points $p_k(x_k,y_k,z_k)^T$ R^3

$$\mathbf{x}_i = (x_1, y_1, z_1, \dots, x_n, y_n, z_n)^T \in \mathcal{R}^{3n}$$

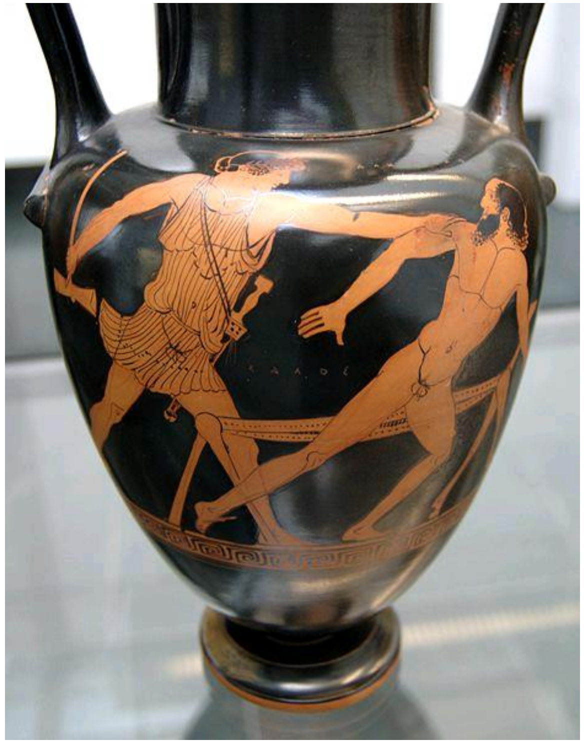
$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) \in \mathcal{R}^{3n \times m}$$

Point Distribution Models: Prokrustes Analysis

- Statistical form analysis requires common coordinate system
- Different orientation of the training data can be seen as natural variance
- Better: Align data before statistical form analysis

- Prokrustes Analysis: Iterative method to align training data
- Minimization of quadratic distance compared to average shape

$$D = \sum |x_i - \bar{x}|^2$$



Quelle: Wikipedia

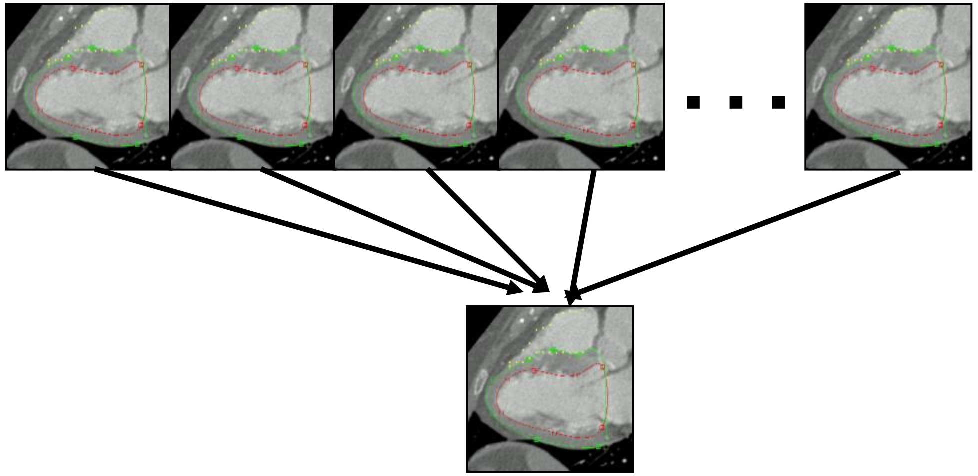
Point Distribution Models: Prokrustes Analysis

- Choose a dataset as reference
- Align all other datasets to reference dataset
- Calculate average shape of all registered data sets
- Calculate Prokrustes distance $\mathrm{D} = \sum |x_i - \bar{x}|^2$
- If D is bigger than a threshold choose average shape as reference

Point Distribution Models

- Calculation of the average shape:

$$\bar{\mathbf{x}} = \frac{1}{m} \sum \mathbf{x}_i$$



Initialisation of the model

Point Distribution Models

- Modeling of shape variance
  - How does a point move on average?

- Covariance matrix

$$\mathbf{S} = \frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i - \overline{\mathbf{x}}) \cdot (\mathbf{x}_i - \overline{\mathbf{x}})^T = \mathbf{X} \cdot \mathbf{X}^T, \quad \mathbf{S} \in \mathcal{R}^{3n \times 3n}$$

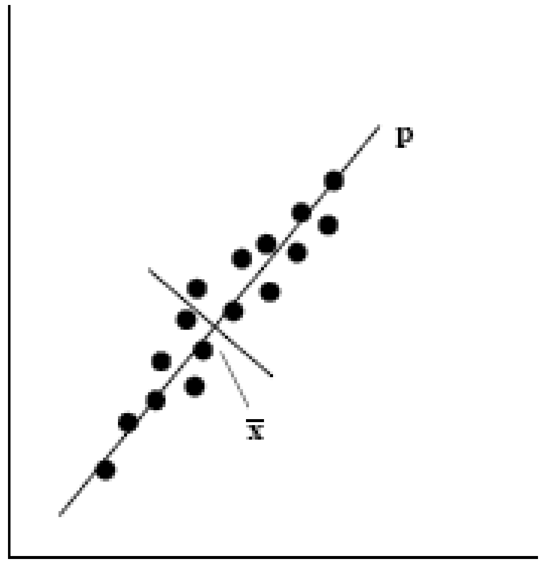
⇒ Describes the variance of all shape vectors

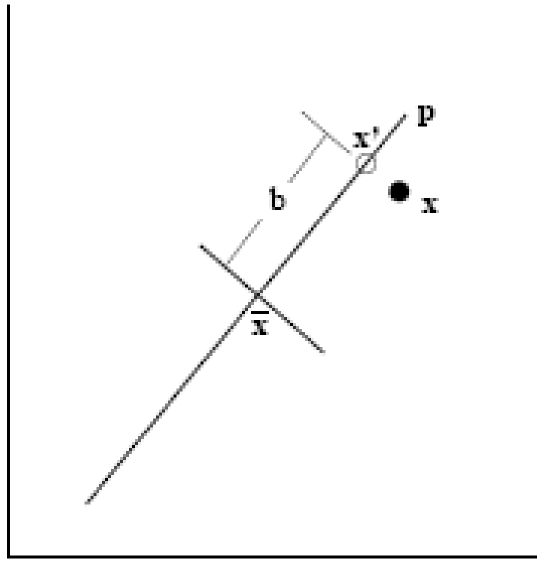
Point Distribution Models

- Calculation of the main deformation direction

  ⇨ Karhunen-Loève-Transformation (engl. Principle Component Analysis, PCA)

  - Movement is included in covariance matrix S
  - Eigenvectors of **S** are main deformation directions





Point Distribution Models

- Calculation of the main deformation direction
- Calculation of Eigenvectors ϕ_i and eigenvalues λ_i of **S**
  - Sort eigenvectors such that
  - $$\lambda_i \geq \lambda_{i+1}$$
  - main deformation directions are described through the first *t* Eigenvectors

$$\mathbf{\Phi} = (\phi_1 | \phi_2 | \ldots | \phi_t)$$

- How big should t be?
  - Choose t such that 98% of all shape variations are described

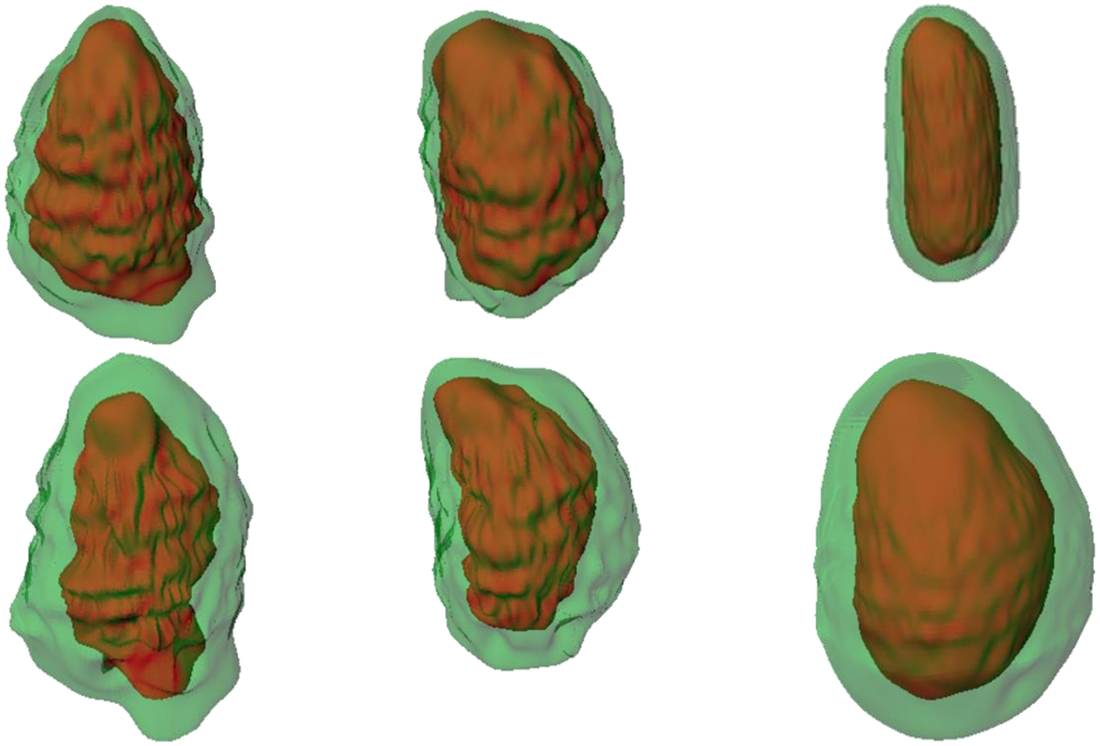
$$V_T = \sum \lambda_i$$

$$\sum_{i=1}^{t} \lambda_i \geq f_v V_T$$

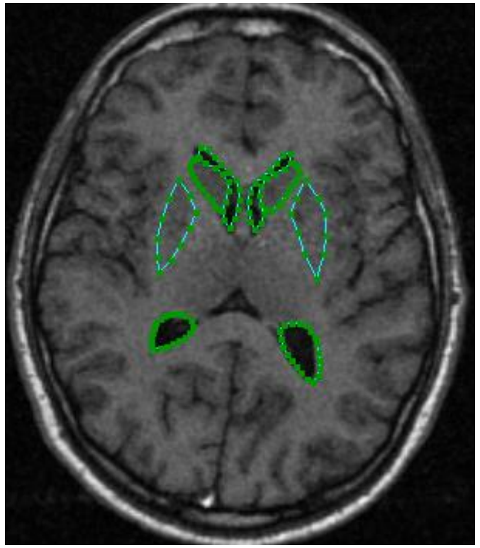
Point Distribution Models

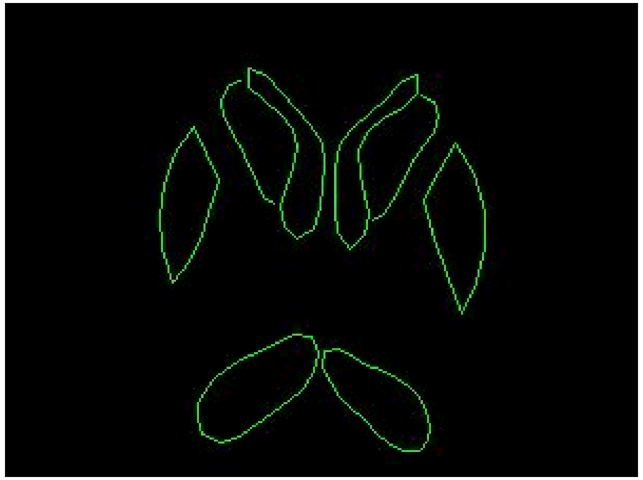
- Approximation of possible model shapes

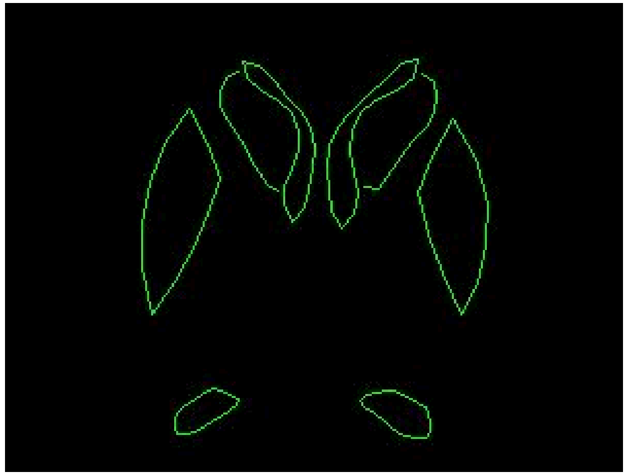
$$\mathbf{x} \approx \overline{\mathbf{x}} + \mathbf{\Phi} \cdot \mathbf{b}, \quad \mathbf{b} \in \mathcal{R}^t, \quad \mathbf{\Phi} \in \mathcal{R}^{n \times t}$$

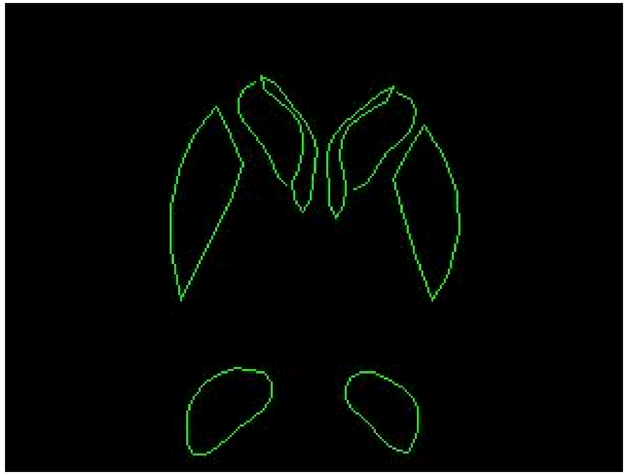


Point Distribution Model





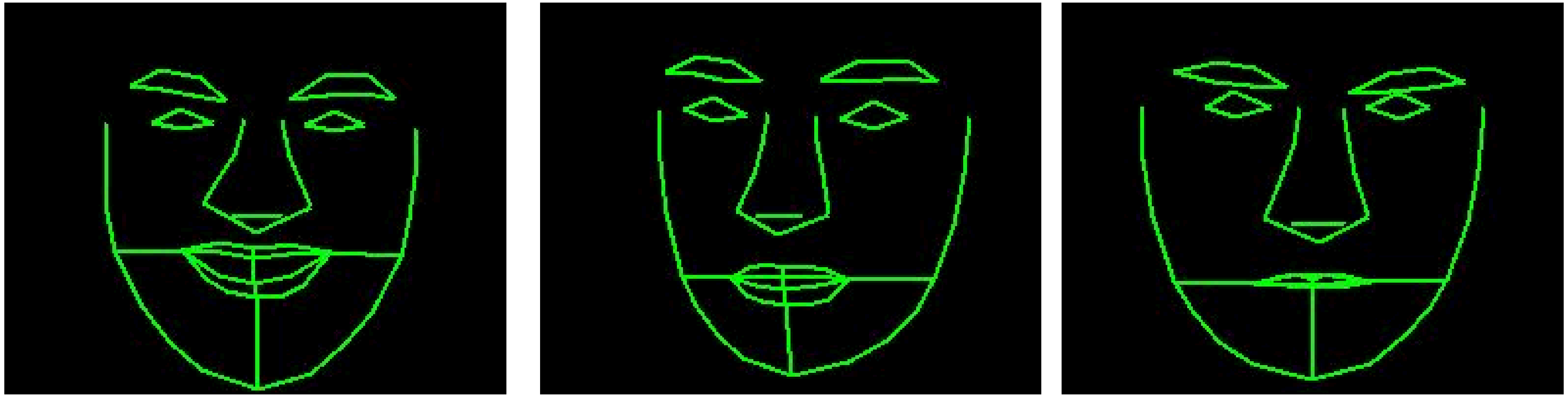




Quelle: T.F. Cootes

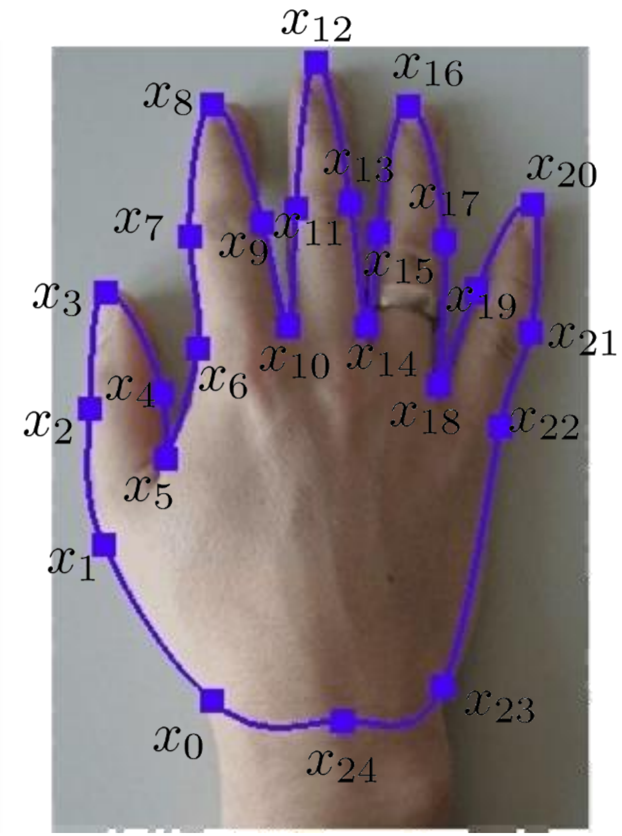
Point Distribution Model

Quelle: T.F. Cootes



Example: Hand PDM
Acquisition of training data

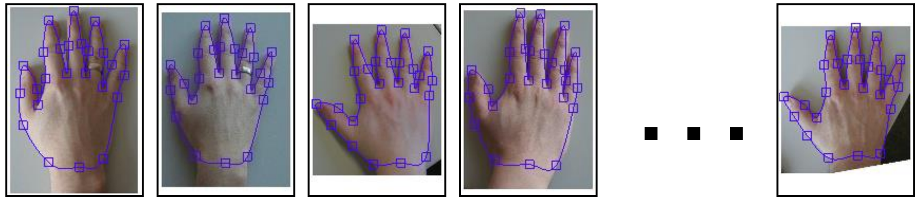
training data



Example: Hand PDM
Acquisition of training data

Training data

- m=15 Shapes



- For every shape a shape vector is build:

$$\mathbf{x}_i = (x_0, y_0, \ldots, x_{24}, y_{24})^T \in \mathcal{R}^{50}$$

- Training data matrix:

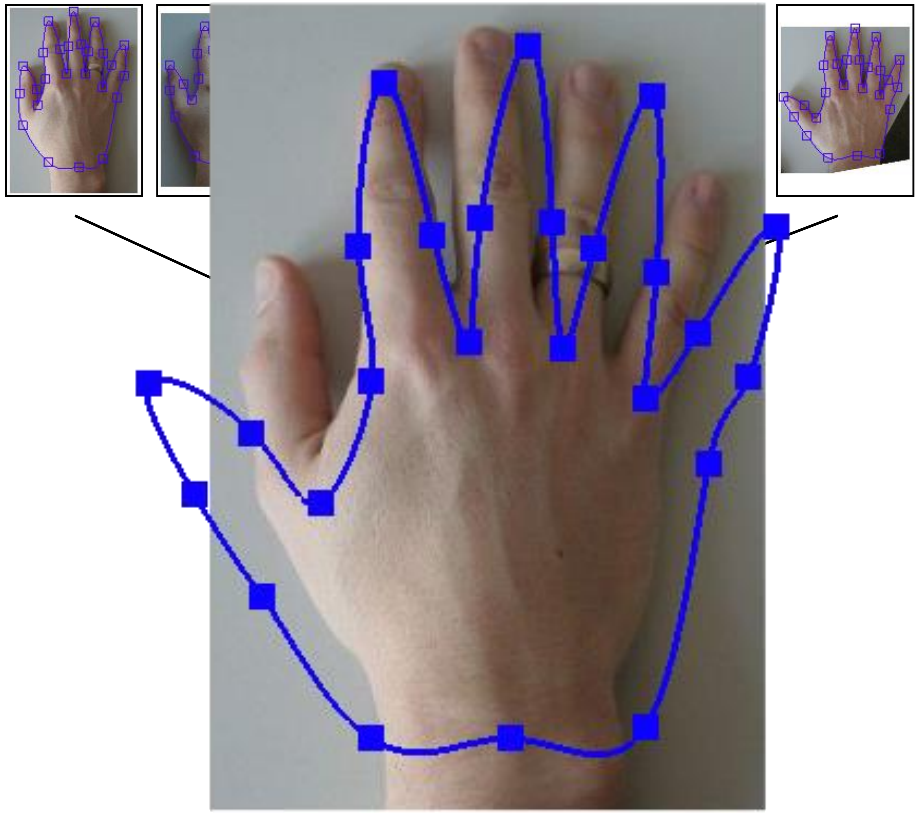
$$\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{14}) \in \mathcal{R}^{50\times 15}$$

Example: Hand PDM
Average Model

- Calculation of the average shape

$$\bar{\mathbf{x}} = \frac{1}{m} \sum \mathbf{x}_i$$

$$\bar{\mathbf{x}}, \mathbf{x}_i \in R^{50}$$



NCT

Example: Hand PDM
Modelling of the shape variances

- Modelling of the shape variance
- Calculation of the covariance matrix

$$\mathbf{S} = \frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i - \overline{\mathbf{x}}) \cdot (\mathbf{x}_i - \overline{\mathbf{x}})^T = \mathbf{X} \cdot \mathbf{X}^T, \quad \mathbf{S} \in \mathcal{R}^{50\times 50}$$

- m = Number of trainings vectors = 15

- PCA detects main deformation directions
  - Calculation of eigenvectors ϕ_i and eigenvalues **S**

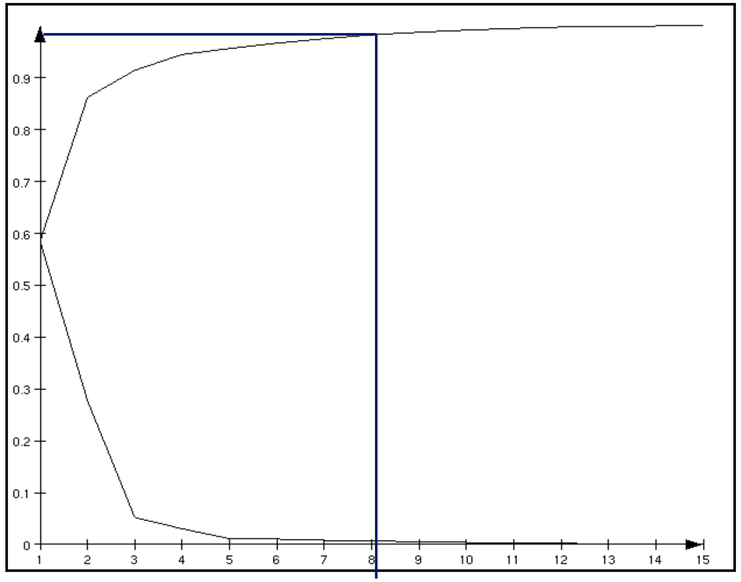
Example: Hand PDM
Modelling of the shape variances

- Main deformation direction are described through the first *t* eigenvectors

$$\mathbf{\Phi} = (\phi_1|\phi_2|\ldots|\phi_t)$$

$$\mathrm{x} \approx \overline{\mathbf{x}} + \mathbf{\Phi}\cdot\mathbf{b}, \quad \mathbf{b}\in\mathcal{R}^t, \quad \mathbf{\Phi}\in\mathcal{R}^{2n\times t}$$

- How do you choose *t*, so that 98% of all variations are covered?

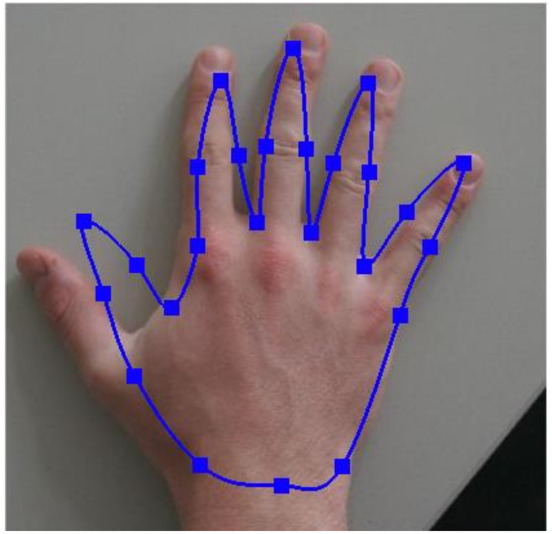


Example: Hand PDM
Limitation to plausible shapes

- Model should guarantee limitations to plausible shapes
- But: arbitrary parameter b allows also „arbitrary shapes“

  Limitation of b

$$|b_i| \leq 3 \cdot \sqrt{\lambda_i}$$

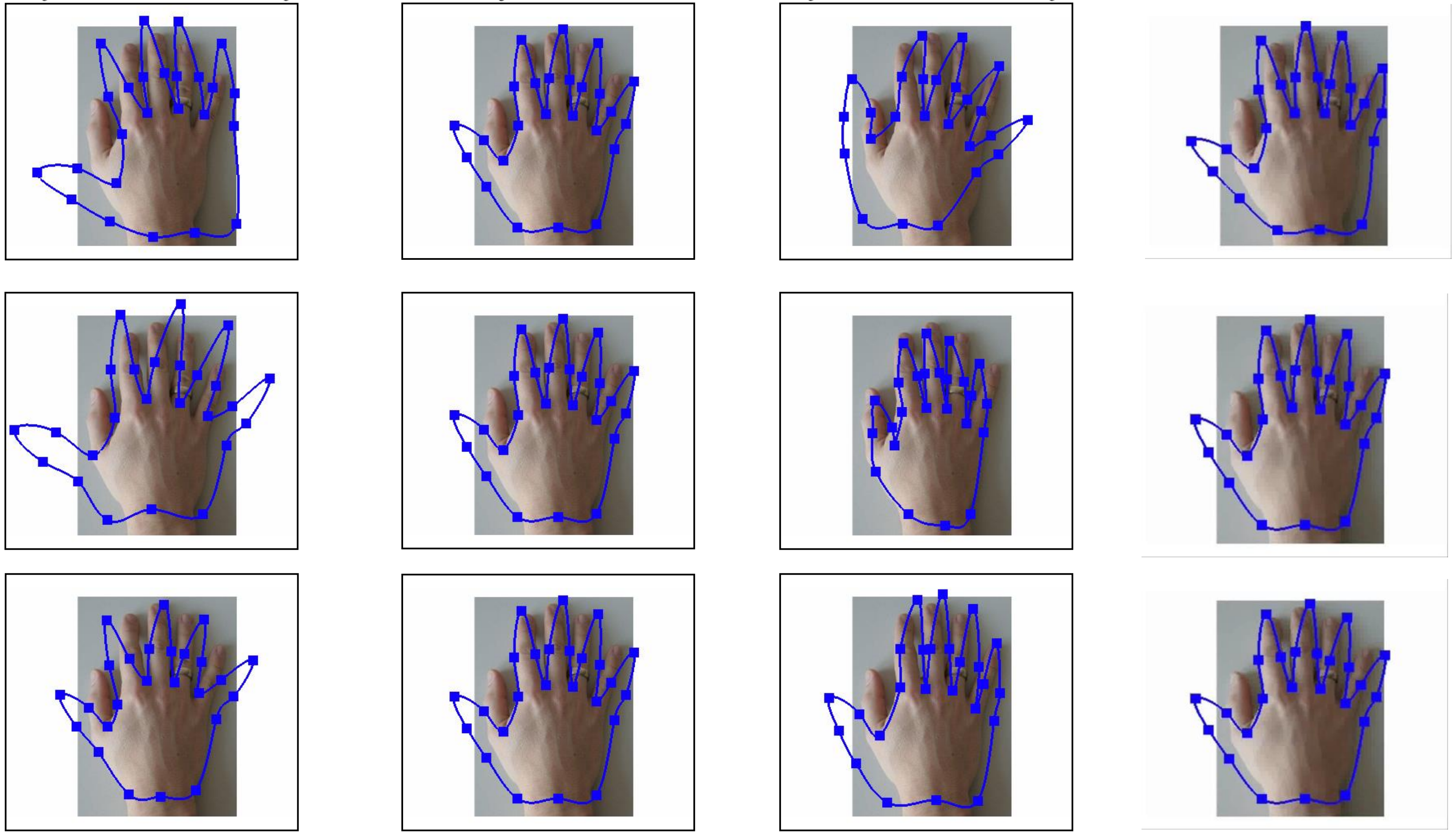


- Background: 3-times standard deviation = outlier.

Example: Hand PDM
Approximation of possible model shapes

$$\mathbf{x} \approx \overline{\mathbf{x}} + \mathbf{\Phi} \cdot \mathbf{b}, \quad \mathbf{b} \in \mathcal{R}^{t}, \quad \mathbf{\Phi} \in \mathcal{R}^{2n \times t}$$

$b_i = -3 \cdot \sqrt{\lambda_i}$ $\qquad$ $b_i = 0$ $\qquad$ $b_i = +3 \cdot \sqrt{\lambda_i}$



b_0

b_1

b_2

Point Distribution Models
Modelling of local shape variances

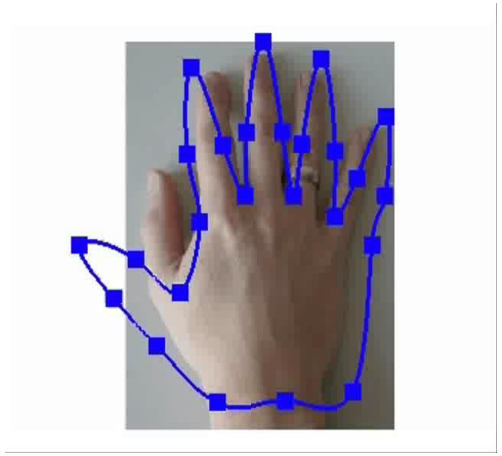
Global vs. Local shape models PCA leads to global shape models

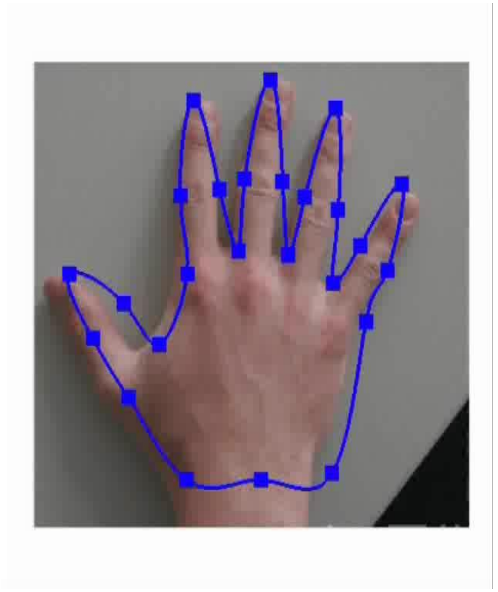
Only global modelling leads to problems regarding segmentation

Local shape models

- Rotation of the principle component matrix leads to local shape vectors

$$\mathbf{R}_{\mathbf{orthomax}} = \arg max \left(\sum_{j=1}^{k} \sum_{i=1}^{p} (\mathbf{\Phi R})_{ij}^{4} - \frac{1}{p} \sum_{j=1}^{k} \left(\sum_{i=1}^{p} (\mathbf{\Phi R})_{ij}^{2} \right)^{2} \right)$$





ACTIVE SHAPE MODELS

Active Shape Models

How can PDMs be applied for segmention?

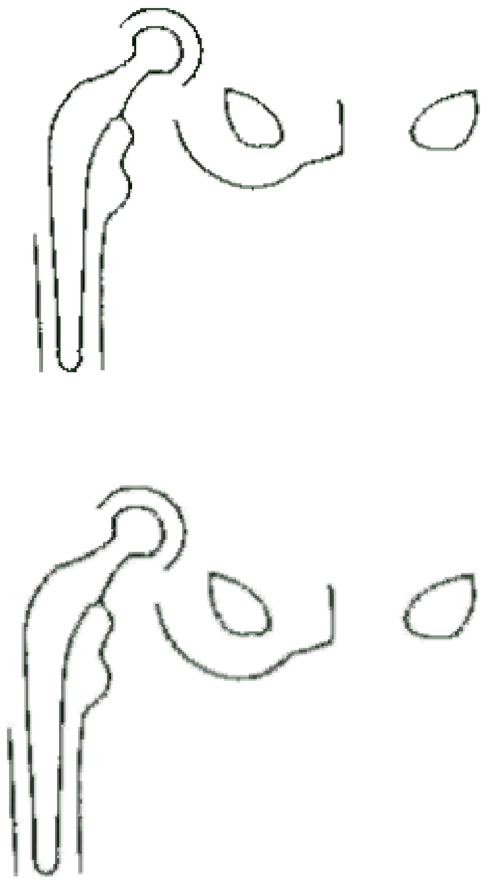
- Active Shape Models (ASM) use PDMs for segmentation of image data
  - Initialisation of the model: $\mathbf{x} = \overline{\mathbf{x}}$
  - Align model points along the normal (e.g. gradient)
  - Re-calculation of the parameter vector:

$$\mathbf{b} = \mathbf{\Phi}^{-1} \cdot (\mathbf{x} - \overline{\mathbf{x}}) = \mathbf{\Phi}^{T} \cdot (\mathbf{x} - \overline{\mathbf{x}})$$

  - Map points on the model:

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi} \cdot \mathbf{b}$$

Active Shape Models



Quelle: T.F. Cootes

Active Shape Models

- Alignment of model points to the image, possible criteria:
  - Gradient
  - Grey value
  - …
  - But: gradient could be missing/not visible

Modelling of local grey value structure

ASM – Mahalanobis Distance

- Modelling of local grey value structure – Mahalanobis Distance
  - „Learning“ of the local grey value profile along the normal for every model point p_i
  - Calculation of the average grey value profile for p_i

$$\overline{\mathbf{g}}_i = \frac{1}{m}\sum_{j=1}^{m} g_{im}$$

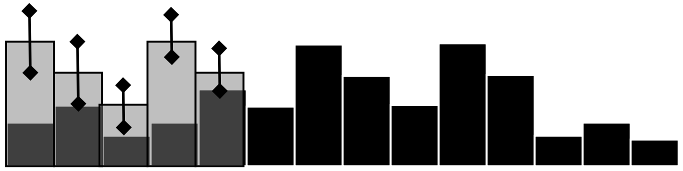
  - Calculation of the covariance matrix

$$S_i = \frac{1}{m}(g_{i1}|g_{i2}|\ldots|g_{im}) \cdot (g_{i1}|g_{i2}|\ldots|g_{im})^T$$

ASM – Mahalanobis Abstand

- Mahalanobis distance as quality factor for ASM adaption
  - Sampling of grey value profile g_s
  - Align grey value model to g_s
  - Calculation of Mahalanobis distance

$$f(g_s) = (g_s - \overline{g_i})^T \cdot S^{-1} \cdot (g_s - \overline{g_i})$$



Active Shape Models: Conclusion

Advantage

- A-priori Knowledge allows „meaningful“ segmentation
- Result also if image quality is reduced
  (e.g. little gradient information)

Disadvantage

- Models have to be trained
  - More data -> better results, more robustness
- Solution might not be optimal
- Segmentation only looks at object shapes, texture inside is discarded

ACTIVE APPEARANCE MODELS

Active Appearance Models

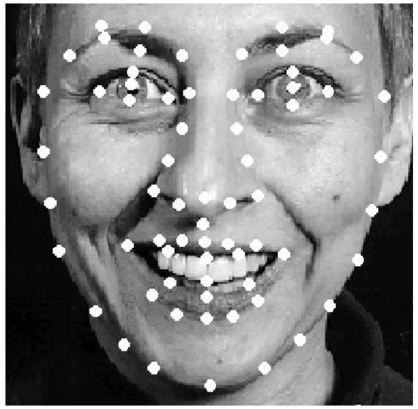
- Active Shape Models only use object shape for segmentation
- Can the texture inside the object be used?

- Active Appearance Models expand Active Shape Models taking texture into consideration

Active Appearance Models: Statistical texture model

From shape to texture model:

- Acquisition of training data (similar to shape model)
- Landmarks are selected (similar to shape model)
- Deformation of the image: landmarks are aligned with average shape.
- Sampling of deformed image. Pixel values define texture vector g_{im}



Quelle: T.F. Cootes

Active Appearance Models
Statistical texture model

Normalisation of texture vector n:

- Global differences in brightness needs to be minimized -> normalisation of texture vectors

$$\mathrm{g} = (\mathrm{g}_{im} - \beta \mathbf{1})/\alpha$$

$$\alpha = \mathbf{g}_{im} \cdot \bar{\mathbf{g}} \qquad \beta = (\mathbf{g}_{im} \cdot \mathbf{1})/n$$

- Recursive definition of normalistion
- Start with arbitrary g_{im}
- Iterative improvement

Active Appearance Models
Statistical texture model

Statististical texture model:

- Acquisition of texture vectors of the training data $(\mathrm{g}_0, \ldots, \mathbf{g}_n)$
- Normalisation of the texture vector
- Calcuation of the covariance matrix

$$\mathbf{S}_{\mathbf{g}} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{g}_i - \overline{\mathbf{g}}) \cdot (\mathbf{g}_i - \overline{\mathbf{g}})^T = \mathbf{G} \cdot \mathbf{G}^T$$

- Eigenvectors describe texture model

$$\boldsymbol{\Phi}_{\mathbf{g}} = (\phi_{g1} | \phi_{g2} | \ldots | \phi_{gt})$$

- Lineare model of possible texture vectors:

$$\mathbf{g} = \bar{\mathbf{g}} + \Phi_g \cdot \mathbf{b}_g$$

Active Appearance Models
Combined Shape-Texture model

- Shape and texture are described with parameter b_g and b_s

- Combination Shape-Texture vector

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \cdot \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \cdot \Phi_s^T (\mathbf{x} - \bar{\mathbf{x}}) \\ \Phi_g^T (\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix}$$

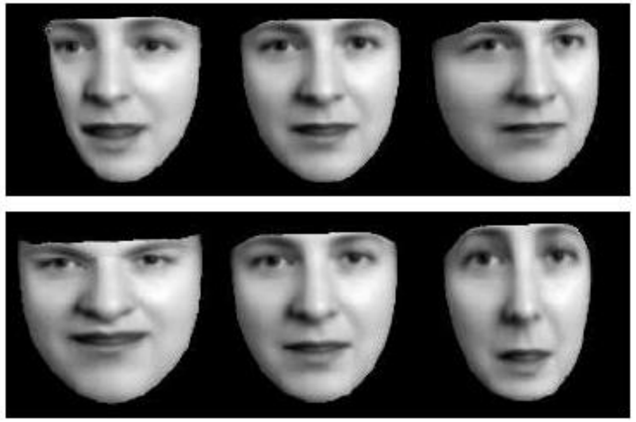
- Diagonal matrix W is weighting factor regarding different unities of grey value and shape
- Application of additional PCA leads to appearance model

$$\mathrm{b} = \Phi_c \cdot c$$

- Φ_c = Eigen vectors of the appearance model
- c = appearance vector

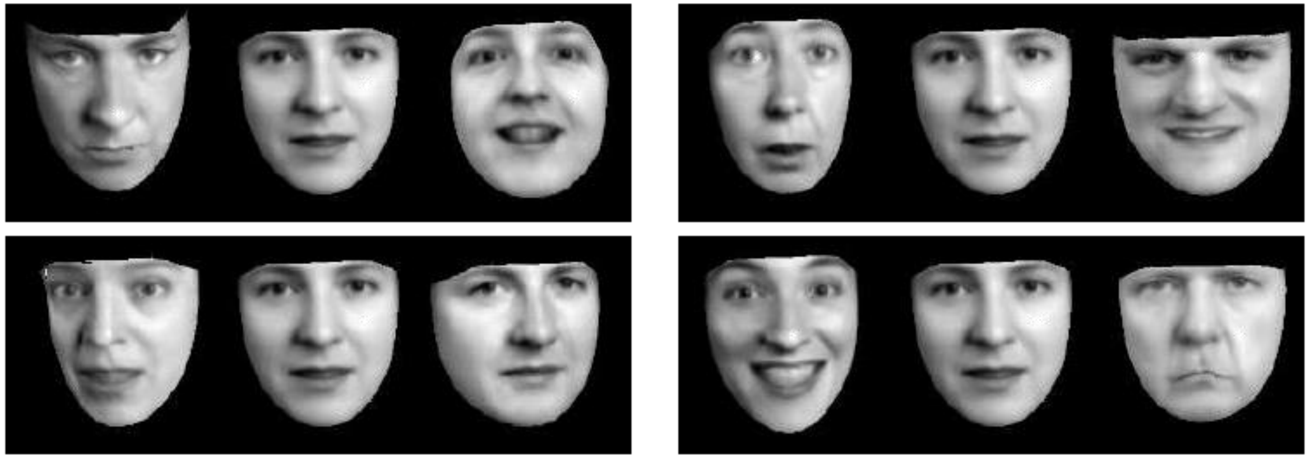
Active Appearance Models
Shape and Texture variation

Variation of the first two Shape Eigenvectors



Variation of the first two Texture Eigenvectors

Variation of the first four Appearance Eigenvectors



Quelle: T.F. Cootes

Active Appearance Models
Shape and Texture variation

$$-2\sqrt{\lambda_1} \longleftarrow b_1 \longrightarrow 2\sqrt{\lambda_1}$$

$$-2\sqrt{\lambda_2} \longleftarrow b_2 \longrightarrow 2\sqrt{\lambda_2}$$

Quelle: T.F. Cootes

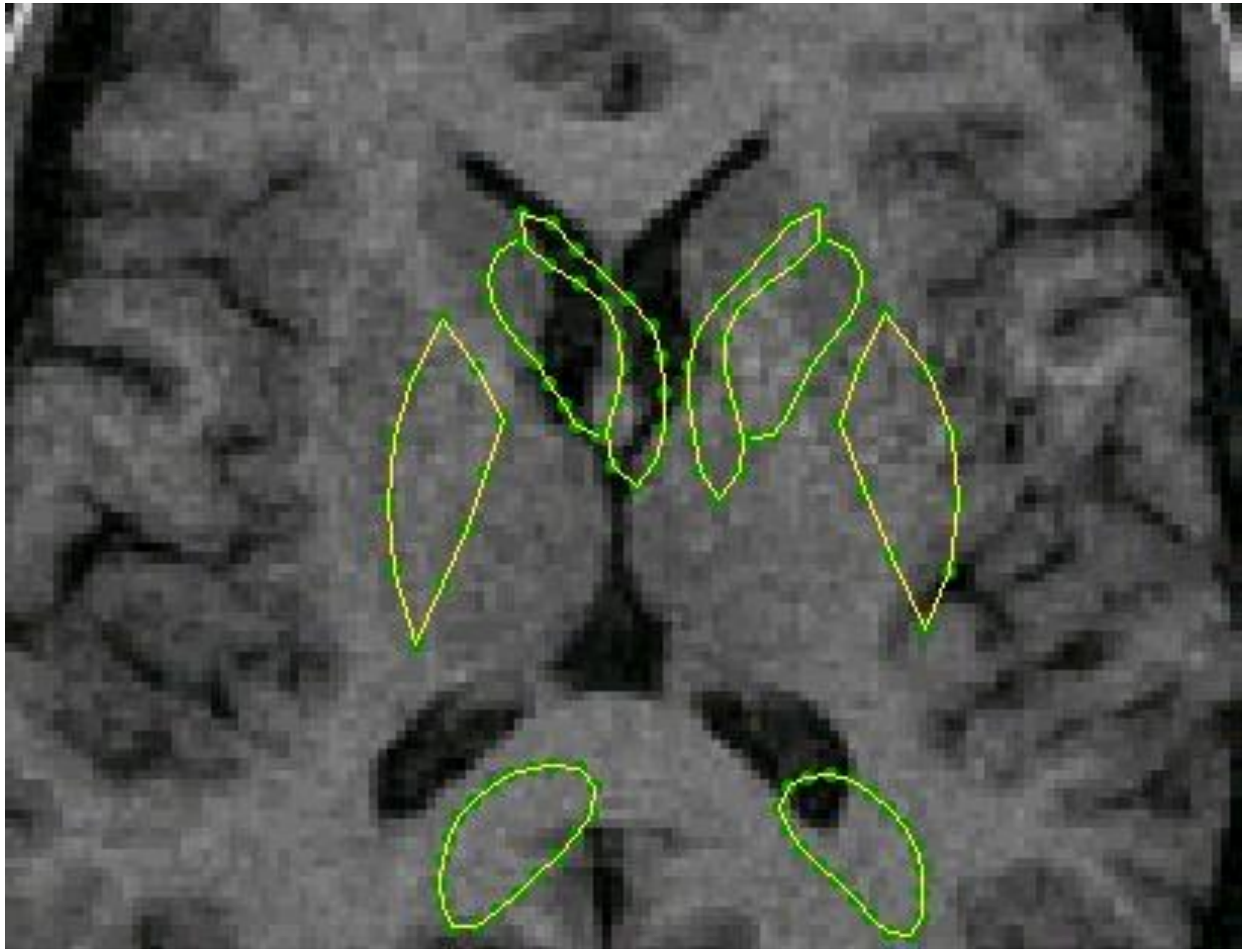
Active Appearance Models
Adaption of the model

Quelle: T.F. Cootes

Active Appearance Models
Adaption of the model

Quelle: T.F. Cootes

Active Appearance Models
Adaption of the model



Quelle: T.F. Cootes

Active Appearance Models

Conclusion

- Segmentation via synthesis
- Extension of Active Shape Models
- AAMs take also the texture into account, not only the shape
- Training is necessary

Literatur

- http://users.cs.cf.ac.uk/Paul.Rosin/CM0311/dual2/hough.html
- T. Cootes, C. Taylor, D. Cooper, and J. Graham. *Active shape models -- their training and application*. Comput Vis Image Underst, 61(1):38--59, 1995.
- T. Cootes, G. Edwards, C. Taylor. *Active Appearance Models*. Transactions on Pattern Analysis and Machine Intelligence, 23: 681--385, 2001
- T. Cootes & C. Taylor. *Statistical Models of Appearance for Computer Vision* . http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf, 2004
- http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/