Open access • Book Chapter • DOI:10.1007/11505730_47

3D active shape models using gradient descent optimization of description length

Tobias Heimann, Ivo Wolf, Thomas N. Williams, Hans-Peter Meinzer

Institutions: German Cancer Research Center, University of Manchester

Published on: 10 Jul 2005 - Information Processing in Medical Imaging

Topics: Gradient descent, Singular value decomposition, Eigendecomposition of a matrix and Covariance matrix

3D Active Shape Models Using Gradient Descent Optimization of Description Length *

Tobias Heimann[1], Ivo Wolf[1], Tomos Williams[2] and Hans-Peter Meinzer[1]

[1] Div. Medical and Biological Informatics, German Cancer Research Center, 69120 Heidelberg, Germany t.heimann@dkfz.de
[2] Div. of Imaging Science, Stopford Building, Oxford Road, University of Manchester, M13 9PT, UK

Abstract. Active Shape Models are a popular method for segmenting three-dimensional medical images. To obtain the required landmark correspondences, various automatic approaches have been proposed. In this work, we present an improved version of minimizing the description length (MDL) of the model. To initialize the algorithm, we describe a method to distribute landmarks on the training shapes using a conformal parameterization function. Next, we introduce a novel procedure to modify landmark positions locally without disturbing established correspondences. We employ a gradient descent optimization to minimize the MDL cost function, speeding up automatic model building by several orders of magnitude when compared to the original MDL approach. The necessary gradient information is estimated from a singular value decomposition, a more accurate technique to calculate the PCA than the commonly used eigendecomposition of the covariance matrix. Finally, we present results for several synthetic and real-world datasets demonstrating that our procedure generates models of significantly better quality in a fraction of the time needed by previous approaches.

1 Introduction

Since their introduction by Cootes et al. [1], Active Shape Models (ASMs) have become popular tools for automatic segmentation of medical images. The main challenge of the approach is the point correspondence problem in the model construction phase: On every training sample for the ASM, landmarks have to be placed in a consistent manner. While manual labeling is a time-consuming but feasible solution for 2D models with a limited number of landmarks, it is highly impractical in the 3D domain: Not only is the required number of landmarks higher than in the 2D case, but it also becomes increasingly difficult to identify and pinpoint corresponding points, even for experts.

Several automated methods to find the correspondences in 3D have been proposed. Brett and Taylor [2] use a pairwise corresponder based on a symmetric version of the ICP algorithm. All training shapes are decimated to generate sparse polyhedral approximations and then merged in a binary tree, which is used to propagate landmark positions. Shelton [3] measures correspondence between surfaces in arbitrary dimensions by a cost function which is composed of three parts representing Euclidean distance, surface deformation and prior information. The function is minimized using a multi-resolution approach that matches highly decimated versions of the meshes first and iteratively refines the results. Paulsen and Hilger [4] match a decimated template mesh to all training shapes using thin plate spline warping controlled by a small set of manually placed anatomic landmarks. The resulting meshes are relaxed to fit the training shapes by a Markov random field regularization. Another approach based on matching templates is presented by Zhao and Teoh [5]: They employ an adaptive-focus deformable model to match each training shape to all others without the need for manually placed landmarks. The shape yielding the best overall results in this process is subsequently used to determine point correspondences, enhanced by a "bridge-over" procedure for outliers.

A common characteristic of these methods is that they base their notion of correspondence on general geometric properties, e.g. minimum Euclidean distance and low distortion of surfaces. A different approach is presented by Davies et al. [6] who propose to minimize a cost function based on the minimum description length of the resulting statistical shape model. In a recent comparison [7], this approach has shown to be superior to other correspondence methods. However, the optimization of the MDL criterion for 3D shapes is complex to implement and computationally expensive. In this paper, we present an optimized procedure for minimizing the description length which is easier to implement and more efficient than the current method.

2 Fundamentals

2.1 Active Shape Models

The most popular kind of ASMs uses point distribution models (PDMs), which represent each d-dimensional training sample as a set of n landmarks. For every sample, landmark positions are defined by a single vector $\mathbf{x}$, storing the coordinates for landmark i at (x_i, x_{i+n}, x_{i+2n}). The vectors of all training samples form the columns of the landmark configuration matrix $\mathbf{L}$. Applying principal component analysis (PCA) to this matrix delivers the principal modes of variation $\mathbf{p}_m$ in the training data. Restricting the model to the first c modes, all valid shapes can be approximated by the mean shape $\bar{\mathbf{x}}$ and a linear combination of displacement vectors:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{m=1}^{c} y_m \mathbf{p}_m \tag{1}$$

Cootes used an eigenvector decomposition of the covariance matrix of $\mathbf{L}$ to calculate the PCA [1], a method commonly employed for this purpose. However, the same results can also be achieved by a singular value decomposition (SVD), which is numerically more stable and thus more accurate when the covariance matrix is ill-conditioned [8].

Theorem 1 *Any $m \times n$ real matrix $\mathbf{A}$ with $m \geq n$ can be written as the product*

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}} \tag{2}$$

where $\mathbf{U}$ and $\mathbf{V}$ are column orthogonal matrices of size $m \times n$ and $n \times n$, respectively, and $\mathbf{D}$ is a $n \times n$ diagonal matrix. Then $\mathbf{U}$ holds the eigenvectors of the matrix $\mathbf{A}\mathbf{A}^{\mathbf{T}}$ and $\mathbf{D}^{\mathbf{2}}$ the corresponding eigenvalues.

Without calculating the covariance matrix, the PCA can thus be obtained by the SVD of the matrix $\mathbf{A} = \frac{1}{\sqrt{s-1}}(\mathbf{L} - \bar{\mathbf{L}})$, where s is the number of samples and $\bar{\mathbf{L}}$ a matrix with all columns set to $\bar{\mathbf{x}}$. In addition to the increased accuracy, the matrices $\mathbf{U}$ and $\mathbf{V}$ allow calculating gradient information for the eigenvalues which we will use during the optimization stage of the model-building process.

2.2 Correspondence by minimizing description length

A prerequisite for statistical shape models is a set of landmark points located at corresponding positions on all training shapes. To quantify this correspondence, the MDL approach introduced by Davies et al. [9] defines a cost function F which is based on the minimum description length of the generated model. In this work, we use a simplified version of the MDL as proposed by Thodberg [10], where F is defined as:

$$F = \sum_m \mathcal{L}_m \quad \text{with} \quad \mathcal{L}_m = \begin{cases} 1 + \log(\lambda_m/\lambda_{\text{cut}}) & \text{for } \lambda_m \geq \lambda_{\text{cut}} \\ \lambda_m/\lambda_{\text{cut}} & \text{for } \lambda_m < \lambda_{\text{cut}} \end{cases} \tag{3}$$

This formulation features one free parameter λ_{cut} which represents the expected noise in the training data. Since all shapes are rescaled to produce a mean shape with RMS radius $r = 1/\sqrt{n}$ for the PCA, the optimal value for λ_{cut} depends on the original average radius of the training shapes $\bar{r}$:

$$\lambda_{\text{cut}} = \left(\frac{\sigma}{\bar{r}}\right)^2 \ , \tag{4}$$

where σ is the standard deviation of noise in the training data. In coherence with the voxel quantization error, Thodberg choses $\sigma = 0.3$ and uses $\bar{r} = 100$ in all his experiments. While we adopt the same σ-value, we modify $\bar{r}$ depending on the resolution of the images from which the training shapes are extracted.

3 Mesh Parameterization

To define an initial set of correspondences and a means of manipulating them efficiently, we need a convenient parameter domain for our training shapes. For closed 2D objects, the natural choice for this parameter domain is the arc-length position on the contour: Choosing an arbitrary starting point and normalizing the total arc-length to 1, all positions on the contour (i.e. all potential landmark positions) can be described by a single parameter $p \in [0..1]$.

In order to minimize complexity for the parameterization of 3D shapes, we will restrict the discussion to closed two-manifolds of genus 0 (i.e. surfaces without holes and self-intersections). Objects of this class are topologically equivalent to a sphere and comprise most shapes encountered in medical imaging (e.g. liver, kidneys and lungs). The task is to find a one-to-one mapping which assigns every point on the surface of the mesh a unique position on the unit sphere, described by two parameters longitude $\theta \in [0..2\pi]$ and latitude $\phi \in [0..\pi]$.

The mapping of an arbitrary shape to a sphere inevitably introduces some distortion. There are a number of different approaches which attempt to minimize this distortion, typically preserving either local angles or facet areas while trying to minimize distortions in the other. An overview of recent work on this topic can be found in [11].

For an initial parameterization, Davies uses diffusion mapping, a simplified version of the method described by Brechbühler [12] which is neither angle- nor area-preserving. Due to our optimization strategy (Sect. 4), our focus lies on preserving angles: Moving neighboring points on the parameterization sphere in a specific direction, we expect the corresponding landmarks on the training shape to move in a coherent direction as well. This behavior is guaranteed by conformal mapping functions, transformations that preserve local angles.

3.1 Creating a conformal mapping

Definition 1 *Each training sample for the ASM is represented as a triangulated mesh $K = (V, E)$ with vertices $u, v \in V$ and edges $[u, v] \in E$. The vertex positions are specified by $\mathbf{f} : V \to R^3$, an embedding function defined on the vertices of K. A second function $\mathbf{\Omega} : V \to R^3$ specifies the coordinates as mapped on the unit sphere, $\forall v \in V : |\mathbf{\Omega}(v)| = 1$.*

Gu et al. present a variational method to create a conformal parameterization in [13]. From an initial Gauss map, where $\mathbf{\Omega}(v)$ represents the normal vector of v, they use a gradient descent optimization to minimize the string energy of the mesh, defined as:

$$\mathcal{E}(K, \mathbf{\Omega}) = \sum_{[u,v] \in E} k_{u,v} \|\mathbf{\Omega}(u) - \mathbf{\Omega}(v)\|^2 \tag{5}$$

Minimizing the string energy with all edge weights $k_{u,v}$ set to 1 yields the barycentric mapping, where each vertex is positioned at the center of its neighbors. Subsequently, a conformal mapping can be obtained using edge weights

depending on the opposing angles α, β of the faces adjacent to $[u, v]$ as in:

$$k_{u,v} = \frac{1}{2}(\cot\alpha + \cot\beta) \tag{6}$$

During the optimization process, all vertices must constantly be projected back onto the sphere by $\mathbf{\Omega}(u) = \mathbf{\Omega}'(u)/|\mathbf{\Omega}'(u)|$. The formal correctness of this approach was later proved in [14].

3.2 Mapping landmarks

Following the preceding sections, the parameterization is defined by a spherical mesh with the same topology as the training sample. In order to obtain the 3D position for an arbitrary landmark at the spherical coordinates (θ, ϕ), which is generally not a vertex, we have to find the intersection between a ray from the origin to (θ, ϕ) and the parameterization mesh. Since mapping landmarks is the most computationally expensive part of the model-building process, an intelligent search strategy of ordering the triangles according to the likelihood of ray intersection speeds up the algorithm considerably. Intersected triangle indices for each landmark are cached and, in the case of a cache miss, neighboring triangles are given priority when searching for the ray intersection. To test a triangle for intersection, we use the method described in [15], which conveniently produces the barycentric coordinates of the intersection point. The same coordinates used on the respective triangle of the training mesh yield the final landmark position.

4 Optimizing Landmark Correspondences

With an initial conformal parameterization $\mathbf{\Omega}_i$ for each training sample i, we can acquire the necessary landmarks by mapping a set of spherical coordinates to each shape. To optimize the point correspondences with the MDL criterion, two possibilities are available: We can either change the individual $\mathbf{\Omega}_i$ and maintain a fixed set of global landmarks or modify individual landmark sets Ψ_i.

In this work, we opted for the first alternative, which has the advantage that the correspondence is valid for any set of points placed on the unit sphere. Therefore, it is possible to alter number and placement of landmarks on the unit sphere at any stage of the optimization, e.g. to better adapt the triangulation to the training shapes. Moreover, we do not need to worry about the correct ordering of landmarks: Since the valid set on the unit sphere is fixed, ensuring a one-to-one mapping to the training shapes is sufficient.

4.1 Re-parameterization

To modify the individual parameterizations in an iterative optimization process, we need a transformation function of the type $\mathbf{\Omega}' = \Phi(\mathbf{\Omega})$. In [6], Davies et al. use symmetric theta transformations for that purpose: Employing a wrapped Cauchy kernel with a certain width and amplitude, landmarks near the kernel

position are spread over the sphere, while landmarks in other regions of the surface are compressed. By accumulating the effects of thousands of kernels at different positions, arbitrary parameterizations can be created.

While this re-parameterization method produces the required effect, it is an inefficient means of modifying surface parameterizations. The main disadvantage is that it is a global modification, i.e. adding one new kernel modifies all landmark positions on the object. Intuitively, it would be desirable to keep established landmark correspondences stable. Therefore, we suggest a new method for modifying parameterization functions based on kernels with strictly local effects.

We will assume that we know a principal direction $(\Delta\theta, \Delta\phi)$ in which the vertices of a local neighborhood on the parameterization mesh should move to improve landmark correspondences. Then we define a Gaussian envelope function to change each spherical coordinate by $c(x, \sigma) \cdot (\Delta\theta, \Delta\phi)$ with

$$c(x, \sigma) = \begin{cases} e^{\frac{-x^2}{2\sigma^2}} - e^{\frac{-(3\sigma)^2}{2\sigma^2}} & \text{for } x < 3\sigma \\ 0 & \text{for } x \geq 3\sigma \end{cases} \tag{7}$$

The variable x denotes the Euclidean distance between the center of the kernel and the specific vertex of the parameterization mesh, while σ specifies the size of the kernel. The movements are cut off at 3σ to limit the range and keep the modification local. During the course of the optimization, σ is decreased to optimize larger regions at the beginning and details at the end. Three examples for possible kernel configurations with different σ-values are shown in Fig. 1.

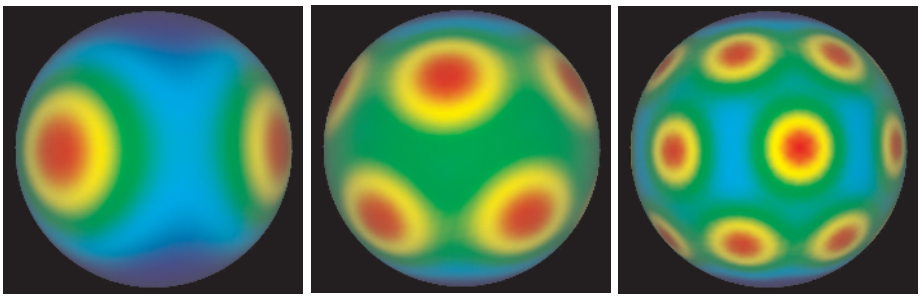

Fig. 1. Kernel configurations for σ values of 0.4, 0.3 and 0.2. Red colors mark regions with large vertex movements, blue ones those with no modification.

The proposed method of modification does not work if a kernel includes one of the poles of the spherical parameterization mesh ($\phi = 0$ or $\phi = \pi$) because vertices would all move either toward or away from this point, depending on $\Delta\phi$. Nevertheless, the positions of the different kernels have to change in the course of the optimization in order to guarantee an equal treatment for all vertices of the parameterization mesh. This limitation is overcome by defining

specific kernel configurations as shown in Fig. 1, which do not cover the pole sections of the sphere. By keeping these configurations fixed and instead rotating all parameterizations and the global landmark collection by a random rotation matrix, the relative kernel positions are changed without touching a pole. The random rotation matrices for these operations are acquired using the method described in [16].

4.2 Calculating MDL gradients

Given a kernel at a certain position, we need the direction $(\Delta\theta, \Delta\phi)$ for the movement which minimizes the cost function. Since all modifications of the parameterization change landmark positions on the training sample, the first step is to quantify the effect landmark movements have on the MDL value. As shown in [17], the work of Papadopoulo and Lourakis on estimating the Jacobian of the SVD [18] can be used for that purpose, calculating the gradients of the MDL objective function with respect to individual landmarks.

The calculation of the singular value derivatives does not add a significant computational overhead. Given the centered and un-biased landmark configuration matrix $\mathbf{A}$ from Sect. 2.1, the derivative for the m-th singular value d_m is calculated by:

$$\frac{\partial d_m}{\partial a_{ij}} = u_{im} \cdot v_{jm} \tag{8}$$

The scalars u_{im} and v_{jm} are elements of the matrices $\mathbf{U}$ and $\mathbf{V}$ from (2). Since our MDL cost function uses $\lambda_m = d_m^2$, we can derive the MDL gradients as

$$\frac{\partial F}{\partial a_{ij}} = \sum_m \frac{\partial \mathcal{L}_m}{\partial a_{ij}} \quad \text{with} \quad \frac{\partial \mathcal{L}_m}{\partial a_{ij}} = \begin{cases} 2u_{im}v_{jm}/d_m & \text{for } \lambda_m \geq \lambda_{\text{cut}} \\ 2d_m u_{im} v_{jm}/\lambda_{\text{cut}} & \text{for } \lambda_m < \lambda_{\text{cut}} \end{cases} \tag{9}$$

This derivation yields a 3D gradient for every landmark, revealing the influence of its movements on the cost function. Two examples of the resulting gradient fields are visualized in Fig. 2.

4.3 Putting it all together

The final step is to transform the calculated gradient fields into optimal kernel movements $\mathbf{k} = (\Delta\theta, \Delta\phi)$ on the parameterization mesh. Using the chain rule, we get:

$$\frac{\partial F}{\partial \mathbf{k}} = \frac{\partial F}{\partial a_{ij}} \frac{\partial a_{ij}}{\partial \mathbf{k}} \tag{10}$$

We use finite differences to estimate the surface gradients $\partial a_{ij}/\partial \mathbf{k}$.

Both Davies [19] and Thodberg [10] describe cases in which the MDL optimization can lead to landmarks piling up in certain regions or collapsing to a point. Davies keeps one shape as a master example with fixed landmarks to prevent this effect while Thodberg suggests adding a stabilizing term to the cost

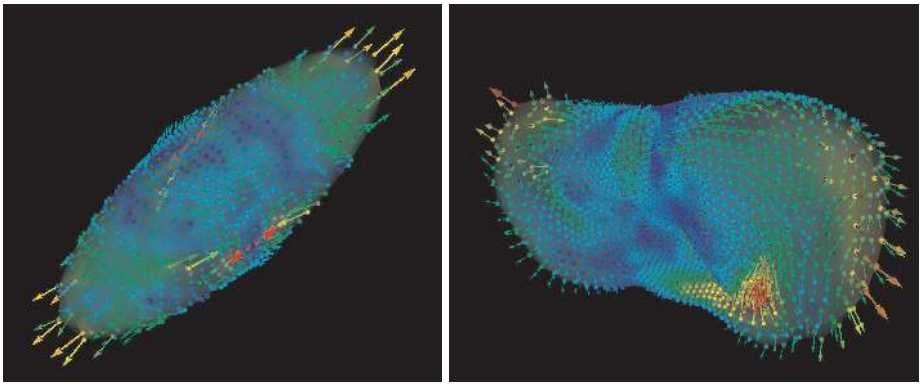

Fig. 2. Gradients of the MDL cost function visualized for two sample shapes. The value of the directional derivative is color-coded ranging from blue for weak gradients to red for the strongest gradients.

function. Since we have never observed the problematic behavior with our new re-parameterization, we do not employ any of these methods.

In addition to modifying the mapping functions $\mathbf{\Omega}_i$ by re-parameterization, other variables which influence landmark positions can be included in the optimization. The rotation of each mapping $\mathbf{\Omega}_i$ determines the position of the first landmark on the training shape and the relative orientation of all others. By calculating gradients for rotating the parameterization mesh around the three Euclidean axes and using those instead of the surface gradients $\partial a_{ij}/\partial \mathbf{k}$ in (10), we have an efficient method to optimize this variable.

Other possibilities for optimization include scale and rotation of the individual training shapes, which are normally determined by a generalized procrustes matching. While we do optimize scale in our procedure, we did not notice significant improvements in the resulting MDL values due to this step.

5 Results

5.1 Datasets

We tested the presented method on several synthetic and real-life datasets. Synthetic data has the advantage that the global minimum of the MDL function is known, since it can be calculated from the correspondences inherent for generated data. A tabular description of all employed datasets is given in Tab. 1.

5.2 Performance measures

In [19], Davies describes three measures to quantify the quality of the created shape model: Generalization ability, specificity and compactness. The same mea-

Table 1. The collection of datasets used for the evaluation.

	Cuboids	Ellipsoids	Lungs	Livers
Origin	synthetic	synthetic	clinical	clinical
Mean size (radius in voxels)	100	100	25	70
Number of samples	20	20	18	21
Perceived sample variance	low	medium	medium	high
Sample complexity (# vertices)	486	962	3250–5000	1500–2000
Model complexity (# landmarks)	642	642	2562	2562

sures are also used in the comparison of different correspondence methods by Styner et al. [7].

Generalization ability quantifies the capability of the model to represent new shapes. It can be estimated by performing a series of leave-one-out tests on the training set, measuring the distance of the omitted training shape to the closest match of the reduced model. Specificity describes the validity of the shapes the model produces. The value is estimated by generating random parameter values from a normal distribution with zero mean and the respective standard deviation from the PCA. The distance of the generated shape to the closest match of the training set is averaged over a number of 10,000 runs. Compactness simply measures the accumulative variance of the model. All measures are defined as functions of the number of modes or parameters used by the model and displayed as piecewise linear graphs. Smaller values indicate better models.

While generalization ability and specificity are well-established qualities inherent to a good model, compactness is an implementation specific measure: The MDL approach assumes that low variances result in a good ASM, but this is no imminent truth. We therefore restrict our evaluation to the first two measures.

5.3 Comparison with current standard

For all datasets described in Sect. 5.1, Active Shape Models have been generated using the gradient descent technique (GD) proposed in this paper and using the current standard approach (STD) by Davies [19]. Our GD-algorithm was implemented in C++ and run on a 3.0GHz Intel Pentium 4 with Windows XP and 512MB of memory. The code makes use of the Hyper-Threading architecture to optimize several samples concurrently. The STD-experiments were performed using the original Matlab code on a 2.8GHz Intel Xenon with Linux and 2GB of memory. After optimization, the global landmark sets of the GD-optimized models were adjusted to match the landmark distributions of the STD-models. For the evaluation, all models were rescaled to the same dimensions as the average training sample. The results of the experiments are summarized in Tab. 2. For our GD-optimization, we list additional intermediate values for the point at which the MDL values surpasses the results of the STD-method.

The GD-optimization reaches the same MDL values as the converged STD-method up to 5,000 times faster and produces distinctly better final results. Gen-

eralization ability and specificity values for all datasets are displayed in Fig. 3. In accordance with the MDL values, models optimized with our GD-method exhibit significantly better generalization ability and specificity values.

Table 2. Resulting MDL values at different stages of optimization using the gradient descent (GD) and standard (STD) method for all datasets. Times are given in hours and minutes.

	Cuboids		Ellipsoids		Lungs		Livers	
Optimization	Time	MDL	Time	MDL	Time	MDL	Time	MDL
Initial values	0:00	1305	0:00	1288	0:00	1216	0:00	2323
STD (converged)	63:24	1297	63:24	1284	432:00	1203	432:00	2275
GD (intermediate)	0:01	1246	0:01	1254	0:05	1180	0:07	2263
GD (converged)	0:36	1243	2:43	1247	17:03	1160	14:45	2140

6 Conclusions

As demonstrated in the preceding section, our gradient descent optimization produces significantly better models than the current standard approach while at the same time being several orders of magnitude faster. Highly detailed models containing over 2,500 landmarks can be successfully optimized in less than 20 hours on a normal desktop PC. This significant performance gain opens up new possibilities for building larger and more detailed 3D ASMs. Excluding the platform difference, a major part of the improvements can be attributed to our novel method of local parameter modification controlled by the estimated gradients of the MDL cost function. As the lower MDL values after optimization indicate, our method is less sensitive to convergence to local minima than the original approach. It offers an efficient, robust and versatile approach to automatic model building that should further propagate the use of 3D ASMs in clinical practice. To represent more complex shapes (e.g. brain ventricles), the mesh surface could be cut and parameterized over multiple domains instead of a single sphere.

Future research will investigate in how far the established correspondences can be used to reorganize landmarks after the optimization in order to represent the geometry of the model optimally with a minimum number of points. Additionally, the stability of our re-parameterization method against landmark collapse has to be verified using a larger number of test datasets.

Acknowledgements

The work reported here was partially funded by the German Research Foundation through SFB 414: Information Technology in Medicine. Special thanks to Rhodri H. Davies for many inspiring discussions about MDL optimization. The original MR lung data was kindly provided by Christian Plathow, the 3D meshes were created by Max Schöbinger. The CT liver data and segmentations were supplied by Matthias Thorn and Jan-Oliver Neumann.

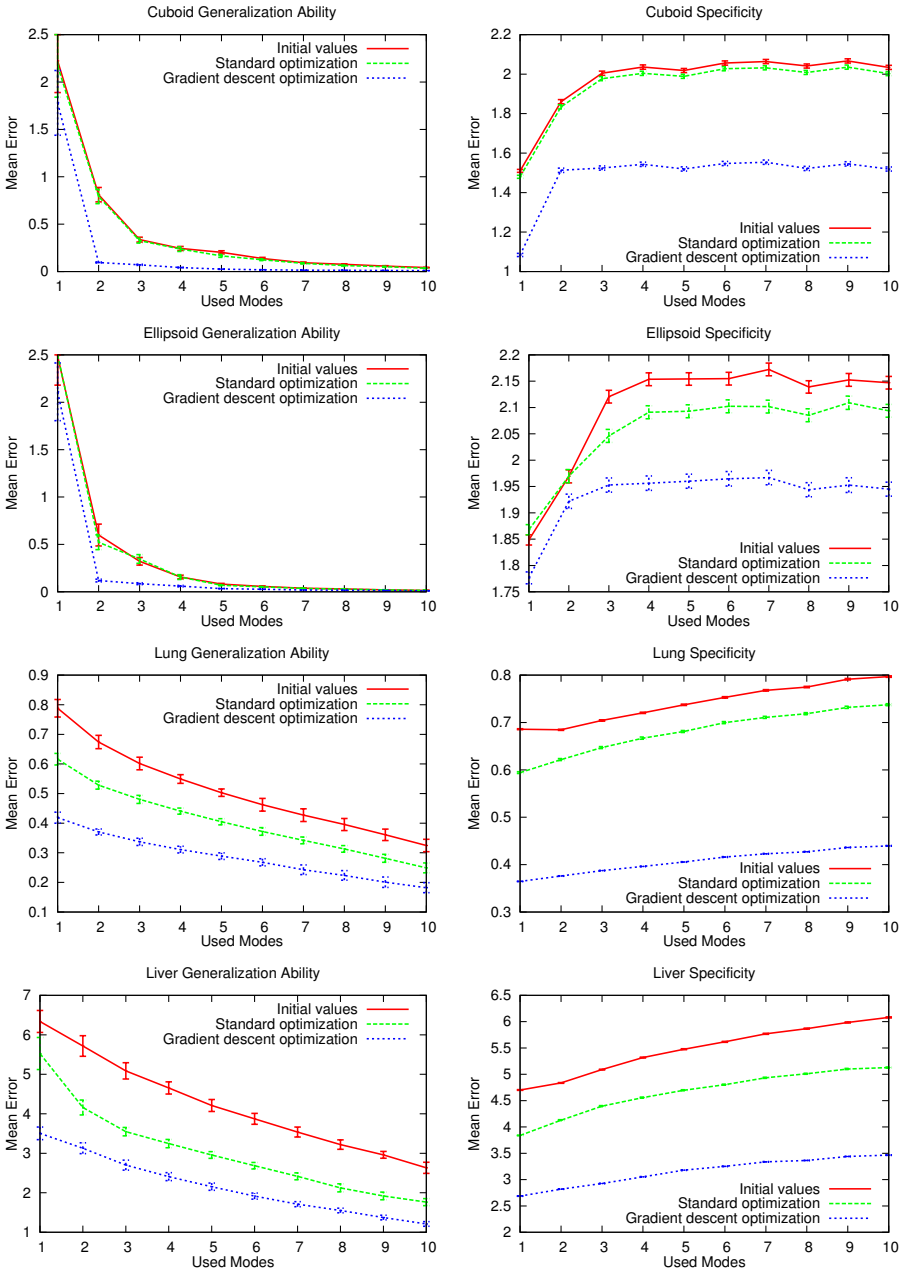

Fig. 3. Graphs of generalization ability and specificity for different numbers of modes for all datasets. In addition to the results after standard optimization and our gradient descent method, the initial values before optimization are displayed as orientation.

References

1. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models – their training and application. Computer Vision and Image Understanding **61** (1995) 38–59
2. Brett, A.D., Taylor, C.J.: A method of automated landmark generation for automated 3D PDM construction. Image and Vision Computing **18** (2000) 739–748
3. Shelton, C.R.: Morphable surface models. Int. Journal of Computer Vision **38** (2000) 75–91
4. Paulsen, R.R., Hilger, K.B.: Shape modelling using markov random field restoration of point correspondences. In: Proc. IPMI. (2003) 1–12
5. Zhao, Z., Teoh, E.K.: A novel framework for automated 3D PDM construction using deformable models. In: Proc. SPIE Medical Imaging. Volume 5747. (2005)
6. Davies, R.H., Twining, C.J., Cootes, T.F., Waterton, J.C., Taylor, C.J.: 3D statistical shape models using direct optimisation of description length. In: Proc. European Conference on Computer Vision, Part III, Springer (2002) 3–20
7. Styner, M., Rajamani, K.T., Nolte, L.P., Zsemlye, G., Székely, G., Taylor, C.J., Davies, R.H.: Evaluation of 3D correspondence methods for model building. In: Proc. IPMI. (2003) 63–75
8. Kalman, D.: A singularly valuable decomposition: The SVD of a matrix. College Math Journal **27** (1996) 2–23
9. Davies, R.H., Twining, C.J., Cootes, T.F., Waterton, J.C., Taylor, C.J.: A minimum description length approach to statistical shape modelling. IEEE trans. Medical Imaging **21** (2002) 525–537
10. Thodberg, H.H.: Minimum description length shape and appearance models. In: Proc. IPMI. (2003) 51–62
11. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In Dodgson, N.A., Floater, M.S., Sabin, M.A., eds.: Advances in Multiresolution for Geometric Modelling. Mathematics and Visualization. Springer, Berlin, Heidelberg (2005) 157–186
12. Brechbühler, C., Gerig, G., Kübler, O.: Parametrization of closed surfaces for 3-D shape description. Computer Vision and Image Understanding **61** (1995) 154–170
13. Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.T.: Genus zero surface conformal mapping and its application to brain surface mapping. In: Proc. IPMI. (2003) 172–184
14. Gotsman, C., Gu, X., Sheffer, A.: Fundamentals of spherical parameterization for 3D meshes. ACM Trans. Graph. **22** (2003) 358–363
15. Möller, T., Trumbore, B.: Fast, minimum storage ray-triangle intersection. Journal of Graphics Tools **2** (1997) 21–28
16. Arvo, J.: Fast random rotation matrices. In Kirk, D., ed.: Graphics Gems III. Academic Press (1992) 117–120
17. Ericsson, A., Åström, K.: Minimizing the description length using steepest descent. In: Proc. British Machine Vision Conference. (2003) 93–102
18. Papadopoulo, T., Lourakis, M.I.A.: Estimating the Jacobian of the singular value decomposition: Theory and applications. In: Proc. European Conference on Computer Vision, Springer (2000) 554–570
19. Davies, R.H.: Learning Shape: Optimal Models for Analysing Shape Variability. PhD thesis, University of Manchester, Manchester, UK (2002)