

Forschungsschwerpunkt S92

# Industrial Geometry

<http://www.ig.jku.at>



FSP Report No. 70

## **The CMA-ES on Riemannian manifolds to reconstruct shapes in 3D voxel images**

S. Colutto, F. Frühauf, M. Fuchs and  
O. Scherzer

May 2008

**FWF**

Der Wissenschaftsfonds.





# The CMA-ES on Riemannian manifolds to reconstruct shapes in 3D voxel images

Sebastian Colutto\*    Florian Frühauf\*    Matthias Fuchs\*  
Otmar Scherzer\*<sup>†</sup>

May 29, 2008

## Abstract

The reconstruction of shapes in images is a demanding question in image analysis. We consider a variational formulation of this task on shape manifolds, in particular the Riemannian manifold of 3D-MReps. Moreover we introduce statistical shape knowledge into the formulation by incorporating a Riemannian metric as regularization term. This leads to an optimization problem for a functional defined on the shape manifold. We generalize the concept of the CMA-ES to Riemannian manifolds and apply this technique to region-based and edge-based segmentation of 3D voxel images. Moreover, we compare the CMA-ES to a standard gradient descent method for the region-based segmentation.

**Keywords:** CMA-ES, medial axis representation, image segmentation, principle geodesic analysis

## 1 Introduction

This paper was motivated by the problem of detecting geometries in 3D voxel images under the assumption that a-priori knowledge about the shape of the solution is available. We consider a variational approach, i.e., we assume the solution to be a minimizer of an accordingly chosen energy functional. There exist two main approaches to segment images via an energy formulation: the *Mumford-Shah* functional [21, 4] and the *Snake* energy [15]. We also refer to the first one as a region-based approach because it identifies regions of homogeneous contrast as parts of an image. The second one is an edge-based approach which leads to segmentations which are characterized by the properties of the edges between them.

These two energies have successfully been used for a wide range of applications. However, in case of noisy image data or the presence of partial occlusions, segmentation approaches which rely only on image features do not give satisfying results. This has led to the development of a large variety of shape reconstruction techniques which are based on shape and image statistics learned from training data.

Cootes et al. [6] introduced statistics of shapes represented by landmarks. More recent approaches are Chen et al. [5], Cremers et al. [7], Fang and Chan [8], Gastaud

---

\*University of Innsbruck, Infmath Imaging, Technikerstrasse 21a/2, 6020 Innsbruck, Austria

<sup>†</sup>Radon Institute of Computational and Applied Mathematics, Altenberger Str. 69, 4040 Linz, Austria

et al. [12], Leventon et al. [19], Rousson and Paragios [24, 25] and Tsai et al. [27]. All these works consider a combination of region- or edge-based segmentation and statistical energies which penalize deviations from previously learned training statistics. In this work we reconsider our approach [11] and combine an arbitrary segmentation energy with a *shape regularization* which is formulated for shape spaces parametrized by a Riemannian manifold. The regularization is based on the Principal Geodesic Analysis (PGA) and the Mahalanobis distance on Riemannian manifolds.

As a shape model we use a parametric 3D medial axis representation (3D-MRep), c.f. Pizeret al. [22] and Fletcher et al. [10]. The parameter domain of 3D-MReps constitute a finite dimensional Riemannian manifold. Each element of this shape space defines a surface  $\gamma: S^2 \rightarrow \mathbb{R}^3$ .

Following [11] the above shape representation and the region- and edge-based reconstruction of objects in 3D voxel images leads to a complex optimization problem. The approach we chose to solve this problem is based on the Covariance Matrix Adaption Evolution Strategy (CMA-ES). The CMA-ES is a population-based probabilistic optimization method for non-linear and non-convex functionals, which are defined on  $\mathbb{R}^N$ . It avoids local minima and is suitable to minimize non-separable functionals with rugged landscapes, like discontinuities or sharp bends. There are a lot of applications in engineering and inverse problems, where the CMA-ES turns out to be a good choice, for instance Abudhahir and Baskar [1], Bayer and Finkel [2], Mera [20], Kern et al. [17]. The behavior of CMA-ES is well studied and compared to other evolution strategies by Kern et al. [18, 16].

Originally the CMA-ES is formulated for functionals on finite dimensional vector spaces. We generalize the CMA-ES to Riemannian manifolds and apply it to our formulation of the shape reconstruction problem. The feasibility of this approach is demonstrated by the reconstruction of geometries in artificial image data and MRI scans of the cerebrum. We compare the results to the minimization of the reconstruction energy using a gradient based approach.

This leads to the following summary of the main contributions of this paper:

- The generalization of the CMA-ES to Riemannian manifolds.
- The application of statistical shape regularization on manifolds [11] to 3D voxel data.
- The minimization of the shape reconstruction energy using the CMA-ES.

The outline of this paper is as follows: In Section 2 we repeat the definition of shape spaces and the approximated PGA on shape spaces. Following [11] we define the Mahalanobis distance on shape spaces and the associated regularization functional. This is followed by a description of the shape space of 3D-MReps. In Section 3 we introduce the CMA-ES on Riemannian manifolds. The application of the CMA-ES to 3D-MReps is discussed in Section 4. In particular we consider the Mumford-Shah and the Snake energies and define regularization functionals based on the Mahalanobis distance in shape spaces. In Section 5 we discuss the details of our implementation of the proposed algorithm and finally present the results of two different experiments in Section 6.

## 2 Shape spaces

In this section we first define the notion of shape spaces and outline some of their properties. This part introduces basic concepts from Riemannian geometry. For further details we refer the reader to [14]. The second part is devoted to statistics on shape spaces.

### 2.1 Riemannian manifolds

In this paper we assume  $M$  to be a Riemannian  $C^\infty$  manifold of dimension  $N \geq 1$ . For  $p \in M$  we call  $T_pM$  the tangent space of  $M$  at  $p$  and  $TM := \bigcup_{p \in M} T_pM$  the tangent bundle of  $M$ . The Riemannian metric on a manifold  $M$  is a smoothly varying inner product  $\langle \cdot, \cdot \rangle_p$  on  $T_pM$ . This inner product defines the norm

$$|v| = \langle v, v \rangle_p^{\frac{1}{2}} \quad \text{for } v \in T_pM.$$

The length  $l$  of a given smooth curve  $\kappa : [0, 1] \rightarrow M$  is computed by integrating the norm of the tangent vectors along  $\kappa$ , i.e.,

$$l = \int_0^1 |\dot{\kappa}(t)| dt = \int_0^1 \langle \dot{\kappa}(t), \dot{\kappa}(t) \rangle_{\dot{\kappa}(t)}^{\frac{1}{2}} dt,$$

where  $\dot{\kappa}(t) = \partial \kappa(t) / \partial t$ . The distance  $d_M$  between two points  $p, q \in M$  is defined as the minimum length over all piecewise differentiable curves between  $p$  and  $q$ . A curve is said to be *geodesic* if it locally minimizes the length between points. A manifold is said to be complete if all geodesics extend indefinitely. This implies that between any two points a length-minimizing geodesic exists. In the following sections we always assume  $M$  to be complete. We refer to  $M$  as a *shape space*.

Let  $p \in M$  and  $v \in T_pM$ , then there exists a unique geodesic  $\kappa : [0, 1] \rightarrow M$  such that  $\kappa(0) = p$  and  $\dot{\kappa}(0) = v$ . The *Riemannian exponential map*  $\text{Exp}_p : T_pM \rightarrow M$  is defined by

$$\text{Exp}_p(v) = \kappa(1).$$

This implies that the exponential map locally preserves distances, i.e.

$$d_M(p, \text{Exp}_p(v)) = |v|,$$

for  $v$  small enough. Moreover, there exists a non-empty neighborhood  $V_p \subset T_pM$  of the origin such that the exponential map is a diffeomorphism on  $V_p$  and any two points  $q, q' \in \text{Exp}_p(V_p) =: U_p$  can be connected by a uniquely determined geodesic which is entirely contained in  $U_p$ . We call  $U_p$  a *normal neighborhood* of  $p$  and the inverse of  $\text{Exp}_p$  on  $U_p$  the *Riemannian logarithmic map*:

$$\text{Log}_p : U_p \rightarrow V_p, \quad \text{Log}_p := \text{Exp}_p^{-1}.$$

Thus for  $q \in U_p$  the geodesic distance between  $p$  and  $q$  is given by

$$d_M(p, q) = |\text{Log}_p(q)|. \tag{1}$$

In the following we give the exponential and the logarithmic map for certain specific shape spaces  $M$ . Let  $p, q \in M$  and  $v \in T_pM$ . First we consider the case  $M = \mathbb{R}^3$  with the Euclidean metric. Then we have

$$\text{Log}_p(q) = q - p \quad \text{and} \quad \text{Exp}_p(v) = p + v.$$

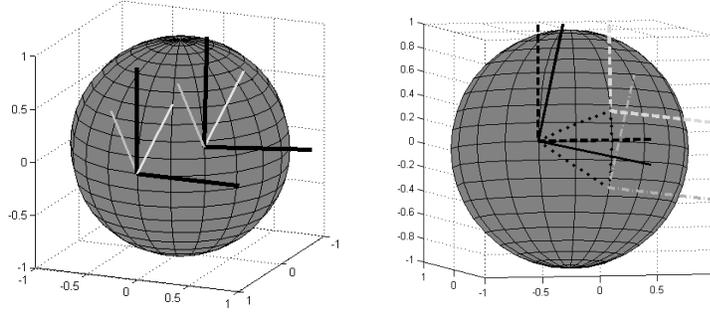


Figure 1: Illustration of the parallel transport.

For  $M = \mathbb{R}^+$  with the metric  $d_{\mathbb{R}^+}(p, q) = |\log(p) - \log(q)|$  for  $p, q \in \mathbb{R}^+$ , the Riemannian maps are given by

$$\text{Log}_p(q) = \log\left(\frac{q}{p}\right) \quad \text{and} \quad \text{Exp}_p(v) = p \exp(v).$$

For the computation of the Riemannian logarithmic map on the unit sphere  $M = S^2$  with the induced metric we use the *azimuthal equidistant projection* [26].

## 2.2 Parallel transport on normal neighborhoods

The definition of a normal neighborhood  $U_p$  of a point  $p \in M$  implies that every other point in the neighborhood can be connected to  $p$  by a uniquely defined geodesic. This allows us to transport tangent vectors and bilinear forms from  $T_p M$  to any other tangent space at a point in  $U_p$  using the parallel transport [11].

Assume a basis  $e_1, \dots, e_N$  of  $T_p M$  which is orthonormal w.r.t.  $\langle \cdot, \cdot \rangle_p$  and  $q \in U_p$ . Let  $\kappa : [0, 1] \rightarrow U_p$  be the unique geodesic connecting  $p$  and  $q$ . Then there exists a unique field of frames  $(e_i)_{1 \leq i \leq N} : [0, 1] \rightarrow TM$  denoted by

$$(e_1(t), \dots, e_N(t)),$$

which is parallel to  $e_1, \dots, e_N$  and satisfies  $e_i(0) = e_i$ ,  $1 \leq i \leq N$ . We define  $e_i(q) := e_i(1)$  for  $1 \leq i \leq N$  and hence we have maps  $e_i : U_p \rightarrow TM$ , where  $(e_1(q), \dots, e_N(q))$  is an orthonormal frame in  $T_q M$  for every  $q \in U_p$ .

Now we consider a tangent vector  $v_p \in T_p M$ ,  $v_p = \sum_{i=1}^N v_i e_i$ . The *transport*  $v_q$  of  $v_p$  to  $T_q M$  is defined by

$$v_q = \sum_{i=1}^N v_i e_i(q). \quad (2)$$

Next we assume a bilinear form  $b_p$  on  $T_p M$ . Then  $b_p$  defines a matrix  $B \in \mathbb{R}^{N \times N}$  w.r.t.  $e_1, \dots, e_N$  by

$$B_{ij} = b_p(e_i, e_j). \quad (3)$$

We define the *transport*  $b_q$  of  $b_p$  to  $T_q M$  by

$$b_q(v_q, w_q) = \sum_{i,j=1}^N v_i w_j B_{ij} \quad \text{for} \quad v_q = \sum_{i=1}^N v_i e_i(q), \quad w_q = \sum_{i=1}^N w_i e_i(q). \quad (4)$$

This transformation of  $b_p$  to  $b_q$  is compatible to the transport of tangent vectors, i.e.,

$$b_p(v_p, w_p) = b_q(v_q, w_q) \quad \text{for } v_p, w_p \in T_p M, q \in U_p.$$

As suggested by the notation the parallel transport is reversible, i.e., transporting  $v_q$  and  $b_q$  back to  $T_p M$  results in  $v_p$  and  $b_p$ . It is important to keep in mind that in either way the transport depends on its origin.

The parallel transport on the  $S^2$  is illustrated in Figure 1. On the left hand side the transport of  $e_1, e_2$  and two vectors in  $T_p S^2$  to the tangent space at a second point  $q \in S^2$  is shown. The right image illustrates that the parallel transport depends on its origin. The image shows the parallel transport of the frames  $e_1, e_2 \in T_p S^2$  to  $q$ . The initial frame (black dotted lines) is transported to  $T_q S^2$  (dotted bright lines). Then we transport the resulting frame from  $q$  to a third point  $r \in S^2$  (gray dash-dotted lines). After that this frame is transported back to  $T_p S^2$ . The resulting frame (black solid lines) is different from the initial one.

### 2.3 Approximated principal geodesic analysis

In this subsection we study a generalization of the linear *Principal Component Analysis* (PCA) to shape spaces, called *Approximated Principal Geodesic Analysis* (aPGA) [11, 10].

First we repeat the concept of the PCA. Let  $p_1, \dots, p_S \in \mathbb{R}^N$  be a set of data points. The mean  $\mu \in \mathbb{R}^N$  of this set is computed as

$$\mu = \arg \min_{p \in \mathbb{R}^N} \sum_{i=1}^S (p - p_i)^2 = \frac{1}{S} \sum_{i=1}^S p_i.$$

The PCA is then used to find a sequence of linear subspaces that optimally represent the variability of the data, i.e., we search for an orthonormal basis  $v_1, \dots, v_N$  of  $\mathbb{R}^N$  which satisfies

$$\begin{aligned} v_1 &= \arg \max_{|v|=1} \sum_{i=1}^S \langle v, p_i - \mu \rangle^2, \\ v_k &= \arg \max_{\substack{|v|=1 \\ v \perp \langle v_1, \dots, v_{k-1} \rangle}} \sum_{i=1}^S \sum_{j=1}^{k-1} (\langle v_j, p_i - \mu \rangle^2 + \langle v, p_i - \mu \rangle^2). \end{aligned}$$

Then the vector  $v_1$  determines the direction of the largest variance of the data set, the vector  $v_2$  the direction of the second largest variance and so on. The vectors  $v_1, \dots, v_N$  are the ordered eigenvectors of the covariance matrix

$$\Sigma = \frac{1}{S} \sum_{i=1}^S (p_i - \mu)(p_i - \mu)^T,$$

which is symmetric and positive semidefinite. The corresponding eigenvalues  $\lambda_k$  of  $\Sigma$  describe the variability of the points  $p_1, \dots, p_S$  in the directions  $v_k$ .

This concept can be generalized to shape spaces as follows. Let  $p_1, \dots, p_S \in M$ . The straightforward definition of the mean  $\mu' \in M$  is

$$\mu' := \arg \min_{p \in M} \sum_{i=1}^S d_M^2(p, p_i). \quad (5)$$

Note that in general, without imposing additional constraints, the existence and uniqueness of such minimizers are not guaranteed.

Next we search for *principal geodesics* on  $M$  as an analogy to the principal components in the linear setting. A submanifold  $H$  of  $M$  is said to be geodesic in  $M$  if all geodesics in  $H$  are also geodesics in  $M$ . We want to compute geodesic submanifolds  $H_1, \dots, H_N$  which are normal to each other and reflect the directions of the principal variances of the data. In the following we give a formal definition of the principal geodesic submanifolds. First we define the projection operator on a closed subset  $H \subset M$  by

$$\pi_H : M \rightarrow H, \quad p \mapsto \arg \min_{h \in H} d_M^2(h, p).$$

Let further be

$$v'_1 := \arg \max_{\substack{v \in T_{\mu'} M \\ |v|=1}} \sum_{i=1}^S d_M^2(\mu', \pi_{H_1(v)}(p_i)), \quad (6)$$

where  $H_1(v) = \text{Exp}_{\mu'}(\langle v \rangle)$  is the exponential of the linear hull of  $v$ . We set

$$v'_k := \arg \max_{\substack{v \in T_{\mu'} M \\ |v|=1 \\ v \perp \langle v'_1, \dots, v'_{k-1} \rangle}} \sum_{i=1}^S d_M^2(\mu', \pi_{H_k(v)}(p_i)), \quad 2 \leq k \leq N, \quad (7)$$

where  $H_k(v) = \text{Exp}_{\mu'}(\langle v'_1, \dots, v'_{k-1}, v \rangle)$  is the exponential of the linear hull of the vectors  $v'_1, \dots, v'_{k-1}, v$ . The tangent vectors  $v'_1, \dots, v'_N$  define the *principal geodesics*. We call  $\mu', v'_1, \dots, v'_N$  the *Principal Geodesic Analysis* (PGA) of the data  $p_1, \dots, p_S$ .

In the following we assume that the data points  $p_1, \dots, p_S$  lie in a sufficiently small neighborhood  $U$ . Then we can approximate

$$d_M(q, q') = |\text{Log}_q(q')| \approx |\text{Log}_{p_1}(q) - \text{Log}_{p_1}(q')|$$

for  $q, q' \in U$ . Inserting this approximation in the formal definition of  $\mu'$  in (5) yields the *approximated mean*

$$\mu = \text{Exp}_{p_1} \left( \frac{1}{S} \sum_{i=1}^S \text{Log}_{p_1}(p_i) \right). \quad (8)$$

Next, we replace  $\mu'$  by  $\mu$  in (6) and (7) and apply the approximation

$$d_M(q, q') = |\text{Log}_q(q')| \approx |\text{Log}_\mu(q) - \text{Log}_\mu(q')|$$

for  $q, q' \in U$ . This leads to the following expressions for the tangent vectors defining the *approximated principal geodesics*:

$$v_1 = \arg \max_{\substack{v \in T_\mu M \\ |v|=1}} \sum_{i=1}^S \langle v, \text{Log}_\mu(p_i) \rangle_\mu^2,$$

$$v_k = \arg \max_{\substack{v \in T_\mu M \\ |v|=1}} \sum_{i=1}^S \sum_{j=1}^{k-1} \left( \langle v_j, \text{Log}_\mu(p_i) \rangle_\mu^2 + \langle v, \text{Log}_\mu(p_i) \rangle_\mu^2 \right)$$

for  $2 \leq k \leq N$ . Thus, the  $v_1, \dots, v_k$  are the principal components of the logarithms of the data in the tangent space at  $\mu$ . This motivates the following definition of the *Approximated PGA*:

**Definition 1**

Assume data  $p_1, \dots, p_S \in U$  as above.

1. Define the approximated mean  $\mu$  as in (8).
2. Choose an orthonormal basis  $e_1, \dots, e_N$  of  $T_\mu M$  and define the covariance matrix  $\Sigma \in \mathbb{R}^{N \times N}$  by

$$\Sigma = \frac{1}{S} \sum_{i=1}^S w_i w_i^T \quad \text{where} \quad \text{Log}_\mu(p_i) = \sum_{j=1}^N w_{i,j} e_j, \quad 1 \leq i \leq S.$$

Then  $(\mu, \Sigma)$  is called the approximated PGA (aPGA) of the data  $p_i, 1 \leq i \leq S$ , w.r.t. the basis  $e_1, \dots, e_N$ .

If  $(v_k, \lambda_k), 1 \leq k \leq N$ , are the eigenvectors and eigenvalues of  $\Sigma$ , then

$$\text{Exp}_\mu(-\lambda_k v_k), \mu, \text{Exp}_\mu(\lambda_k v_k)$$

correspond to the mean and to the elements of the  $k$ -th approximated principal geodesic whose distance to  $\mu$  is  $\lambda_k$  w.r.t. the Riemannian metric on  $M$ .

**2.4 Mahalanobis distance on shape spaces**

For a symmetric, positive definite matrix  $\Sigma$  the Mahalanobis distance  $d_\Sigma : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$  is defined as

$$d_\Sigma^2(x, y) = (x - y)^T \Sigma^{-1} (x - y),$$

where  $x, y \in \mathbb{R}^N$ . In the Euclidean case  $\Sigma$  is the (possibly empirical) covariance matrix of a normal distribution centered at the origin. This definition can also be expressed via the inner product  $\langle \cdot, \cdot \rangle_\Sigma$  given by

$$\langle v, w \rangle_\Sigma = v^T \Sigma^{-1} w, \quad v, w \in \mathbb{R}^N.$$

Then

$$d_\Sigma^2(x, y) = \langle x - y, x - y \rangle_\Sigma.$$

Now we generalize this concept to shape spaces following [11]. Again assume  $\Sigma \in \mathbb{R}^{N \times N}$  to be a symmetric, positive definite matrix and  $\mu \in M$ . Let  $e_1, \dots, e_N$  be a fixed orthonormal basis of  $T_\mu M$  w.r.t.  $\langle \cdot, \cdot \rangle_\mu$  and  $U_\mu$  a normal neighborhood of  $\mu$ . Note that  $U_\mu$  is not necessarily “small”. In case of the sphere e.g. normal neighborhoods can always be chosen such that they cover the whole manifold with the exception of one point. The matrix  $\Sigma^{-1}$  defines a (symmetric, positive definite) bilinear form  $b_{\Sigma^{-1}, \mu}$  on  $T_\mu M$  by

$$b_{\Sigma^{-1}, \mu}(v_\mu, w_\mu) = v_\mu^T \Sigma^{-1} w_\mu \quad \text{for} \quad v_\mu = \sum_{i=1}^N v_i e_i, \quad w_\mu = \sum_{i=1}^N w_i e_i.$$

We call the family of the transports of  $b_{\Sigma^{-1}, \mu}$  to  $U_\mu$  as in (4) the *Mahalanobis metric* and denote them by

$$b_{\Sigma^{-1}, p} \quad \text{for} \quad p \in U_\mu.$$

The Mahalanobis distance  $d_{M,\mu,\Sigma}$  from  $p$  to  $q$  in  $U_\mu$  w.r.t.  $\mu$  and  $\Sigma$  is then given by

$$d_{M,\mu,\Sigma} = \inf_{\substack{\kappa:[0,1] \rightarrow U_\mu \\ \kappa(0)=p \\ \kappa(1)=q}} \int_0^1 \sqrt{b_{\Sigma^{-1},\kappa}(\dot{\kappa}, \dot{\kappa})} dt.$$

Note that unlike to the situation on vector spaces, the Mahalanobis distance on shape spaces depends not only on  $\Sigma$  but also on  $\mu$ . In [11] it is proven that this generalization corresponds to the classical Mahalanobis distance in the Euclidean setting and that it equals the standard Riemannian metric on  $M$  if  $\Sigma = \text{Id}_N$ .

If  $\Sigma$  and  $\mu$  are the covariance matrix and the mean of a given probability distribution, respectively, then the Mahalanobis distance  $d_{M,\mu,\Sigma}$  reflects the shape of this distribution, i.e., the distance between a point  $p \in M$  and  $\mu$  is small, if  $p$  is located in a "probable direction". We use this property in our regularization functional because it allows us to penalize deviations from a reference element depending on the distribution of the training data.

## 2.5 A Mahalanobis regularization functional

In this section we introduce regularizations on shape spaces. Let  $R_\alpha : M \rightarrow [0, \infty]$ ,  $\alpha > 0$ , be a family of continuous maps. Assume that  $\mathcal{D} := \bigcap_{\alpha > 0} \text{dom}(R_\alpha) \neq \emptyset$ . We call  $(R_\alpha)_{\alpha > 0}$  a regularization if it satisfies the following conditions:

1. For every  $p \in \mathcal{D}$

$$\lim_{\alpha \rightarrow 0} R_\alpha(p) = 0$$

holds.

2. The regularization of every unbounded sequence tends to infinity, i.e., there exists some  $p_0 \in M$  such that for every  $\alpha > 0$

$$\lim_{k \rightarrow \infty} d_M(p_0, p_k) = \infty \quad \text{implies} \quad \lim_{k \rightarrow \infty} R_\alpha(p_k) = \infty.$$

3. For every  $p \in \mathcal{D}$ , there exists  $c > 0$  such that for all  $q \in M$  and all  $\alpha > 0$

$$R_\alpha(q) \leq R_\alpha(p) \quad \text{implies} \quad d_M(p_0, q) \leq c.$$

Let  $d_{M,\mu,\Sigma}$  be the Mahalanobis distance defined by  $\mu \in M$  and a covariance  $\Sigma \in \mathbb{R}^{N \times N}$  as in Section 2.4. Then, the map

$$p \mapsto \alpha d_{M,\mu,\Sigma}^2(\mu, p) \tag{9}$$

qualifies as a regularization [11].

## 2.6 The shape space of 3D-MReps

The following definition of the 3D-MRep shape space is based on the definition of the (continuous) medial axis transform that has been introduced by Blum [3]. The initial idea of parametric MReps is due to Pizer et al. [22]. It uses a discretization of the continuous medial axis of an object instead of boundary representations.

In our implementation, every instance of an MRep in three dimensions is represented by

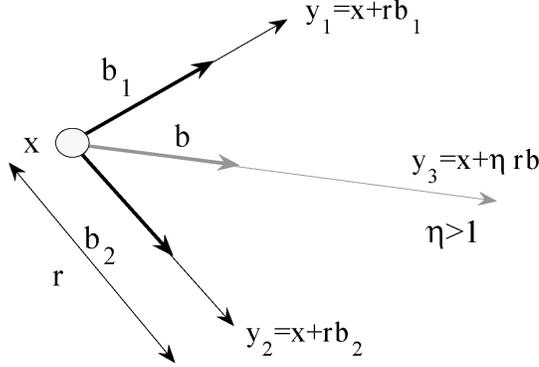


Figure 2: Illustration of the atom parameters of a boundary atom with  $\eta > 1$ .

1. a regular  $n_1 \times n_2$  quadrilateral *mesh* and
2. the  $n_1 n_2$  *atoms*, centered on the mesh vertices.

The following information is provided for every atom of a shape instance:

1. The *position*  $x \in \mathbb{R}^3$  of the atom.
2. The *radius*  $r \in \mathbb{R}^+$  of the atom.
3. Two *boundary vectors*  $b_1, b_2 \in S^2$ , pointing to the two implied boundary points of the surface. For *border atoms* (i.e. atoms on the boundary of the mesh), an *additional boundary vector*  $b \in S^2$  is defined as the bisector of the two given boundary vectors.
4. A parameter for border atoms, called  $\eta \in \mathbb{R}^+$ , specifying the elongation factor along the additional boundary vector.

An illustration of a boundary atom and its parameters is shown in Figure 2. The *shape space of 3D-MReps* is defined as follows:

**Definition 2**

Let  $n_1, n_2 \geq 2$ . The product space

$$\begin{aligned}
 M & := \underbrace{\left( \underbrace{\mathbb{R}^3}_{\text{position}} \times \underbrace{\mathbb{R}^+}_{\text{radius}} \times \underbrace{S^2 \times S^2}_{\text{boundary vectors}} \right)}_{\text{inner atoms}}^{(n_1-2)(n_2-2)} \\
 & \times \underbrace{\left( \underbrace{\mathbb{R}^3}_{\text{position}} \times \underbrace{\mathbb{R}^+}_{\text{radius}} \times \underbrace{S^2 \times S^2}_{\text{boundary vectors}} \times \underbrace{\mathbb{R}^+}_{\text{boundary elongation}} \right)}_{\text{boundary atoms}}^{2((n_1-1)+(n_2-1))}
 \end{aligned}$$

is called the shape space of 3D-MReps.

On the left hand side of Figure 5 an example of an 3D-MRep is displayed. Every instance of  $M$  defines a surface via its parameters. Thus we can introduce a map  $\psi$  that represents the boundary of an instance of 3D-MReps. This leads to the following definition of the *shape model*:

**Definition 3**

Let  $M$  be the shape space of 3D-MReps,  $p \in M$  and  $\Omega \subset \mathbb{R}^3$  a domain. Let furthermore be

$$\psi : M \rightarrow \mathcal{C}(S^2, \Omega),$$

where  $\psi$  is a closed surface that is homeomorphic to a sphere. Then we call  $(M, \psi)$  a 3D-MRep shape model.

### 3 The CMA-ES on Riemannian manifolds

The CMA-ES [16, 18, 13] is an evolutionary optimization method for non-linear, non-convex functionals  $\mathbb{R}^N \rightarrow \mathbb{R}$  based on iterative random sampling of arguments. In each step random points are sampled from a multivariate normal distribution. The best sample points are used to estimate a new mean and a new covariance matrix which define the according normal distribution. We refer to the set of random points which are generated in one iteration as a *generation*.

One important property of the CMA-ES is that the covariance of a generation is not completely determined by the results of the previous generation. The amount of the update relative to the old values is controlled by *learning rates*. Thus, the sampling points of the next generation are created depending on the local behavior of the functional *and* the sampling characteristics of the previous generations.

The major advantage of this optimization method compared to gradient based methods is that non-convex functionals or functionals with rugged search landscape can be minimized efficiently [13]. In Figure 3 a 2-dimensional section of the simplified Mumford-Shah functional on 3D-MReps is shown. There we can see such a rugged search landscape.

In the following we give a short review of the CMA-ES on vector spaces [13]. As stated above, the CMA-ES iteratively samples a population of points from a normal distribution. This distribution is determined by its mean and a covariance matrix multiplied by a step size. Each of these variables — the mean, the covariance matrix and the step size — are updated in each iteration. In addition, the CMA-ES on vector spaces stores information about the “directions” of the previous iterations. These directions are the accumulated and normalized differences of the means of two subsequent generations. In [13] the author refers to the family of these directions as the *evolution paths*. The evolution path directions “remember” information of the previous generations and are one component of the updates of the covariance and the step size in each iteration. This leads to two different evolution paths, one for the update of the covariance and one for the update of the step size.

The goal of this section is to generalize the CMA-ES to Riemannian manifolds. We sample tangent vectors in the tangent spaces at the mean and project them onto the manifold  $M$  by means of the Riemannian exponential map. Moreover, the covariance matrix is replaced by a positive definite bilinear form on the tangent bundle of  $M$ . The challenge is to determine the mean value (on  $M$ ) for the next iteration and to transport the covariance and the current evolution path directions to the new location on the manifold. Under the assumption that each generation is sampled within a normal neighborhood of the mean, this is done by utilizing the parallel transport introduced in Section 2.2.

Before proceeding with the CMA-ES on Riemannian manifolds we review an alternative approach to minimize functionals defined on  $M$ . The Riemannian logarithmic map  $\text{Log}_p$ ,  $p \in M$ , is a diffeomorphism of normal neighborhoods of  $p$  to the

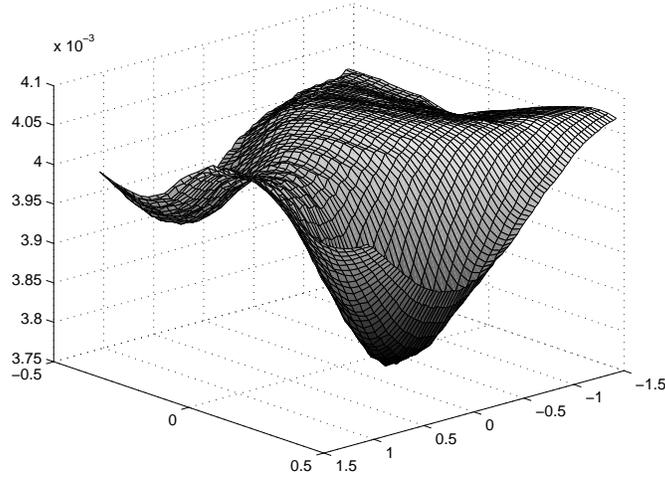


Figure 3: Illustration of a 2-dimensional section of the simplified Mumford-Shah functional which we use in Section 6.1. The plot includes local minima and hence justifies the use of the CMA-ES as a method for optimization.

their images in the tangent space  $T_pM$ . If  $U_p$  is such a normal neighborhood and  $V_p := \text{Log}_p(U_p)$  then

$$F' : V_p \rightarrow \mathbb{R}, \quad q \mapsto F(\text{Exp}_p(q))$$

is a reformulation of  $F$  on  $T_pM$ , i.e., on a vector space. The obvious alternative to minimizing  $F$  by using the CMA-ES on  $M$  is to perform the classical CMA-ES for  $F'$  on  $T_pM$ . However, the drawback of this approach is that the samples of *all* generations are required to be contained in  $V_p$  whereas in our case it is sufficient that the samples of *one* generation are contained in a common normal neighborhood.

In the following we introduce the notations and parameters used in the description of the algorithm in the next section. First we assume a functional  $I : M \rightarrow \mathbb{R}$  which we want to minimize. We denote the current iteration by  $k \in \mathbb{N}$ . The mean value and the step size at the  $k$ -th iteration are accordingly given by  $\mu^k \in M$ , and  $\sigma^k > 0$ . The positive definite bilinear form  $b_{\mu^k}$  on  $T_{\mu^k}M$  represents the covariance. Moreover we assume the two evolution path directions  $v_{c,\mu^k}, v_{\sigma,\mu^k} \in T_{\mu^k}M$ . The proposed algorithm depends on the same parameters as in the Euclidean setting. These include the population size  $m_1$  and the number of selected samples  $m_2$ . The recombination of the selected samples is controlled by *recombination coefficients*  $w_i$ ,  $1 \leq i \leq m_2$ , where

$$w_1 \geq \dots \geq w_{m_2} > 0 \quad \text{and} \quad \sum_{i=1}^{m_2} w_i = 1.$$

Furthermore the update of the evolution path directions is controlled by the learning rates  $0 < c_c, c_\sigma \leq 1$  and the number of *effective samples*  $m_{\text{eff}}$ . The parameters  $1 \leq \mu_{\text{cov}}$  and  $0 < c_{\text{cov}} \leq 1$  control the update of the covariance. Finally, the computation of the step size depends on the damping parameter  $d_\sigma \approx 1$ . With exception of the population sizes  $m_1$  and  $m_2$  we set all parameters to the values recommended in [13].

### 3.1 One evolution step

In the following we give a detailed description of a single step of the proposed algorithm. Our goal is to minimize the energy of the objective function  $I$ . I.e. we try to move  $\mu^k$  closer to a global minimum of  $I$ . In addition, we update  $b_{\mu^k}$ ,  $v_{c,\mu^k}$  and  $v_{\sigma,\mu^k}$  and propagate them to the tangent space at  $\mu^{k+1}$ .

We assume that an orthonormal basis  $e_1, \dots, e_N$  of  $T_{\mu^k}M$  is fixed. Then the bilinear form  $b_{\mu^k}$  can be expressed by a symmetric, positive definite matrix  $C$  as in (3). Furthermore, all vectors in the tangent space  $T_{\mu^k}M$  can be assumed to be coordinates in  $\mathbb{R}^N$  (w.r.t.  $e_1, \dots, e_N$ ). We start by sampling  $m_1$  tangent vectors  $v_{1,\mu^k}, \dots, v_{m_1,\mu^k} \in T_{\mu^k}M$  from the multivariate normal distribution defined by  $\sigma^k C$ :

$$v_{i,\mu^k} \sim \mathcal{N}(0, (\sigma^k)^2 C), \quad 1 \leq i \leq m_1. \quad (10)$$

If the  $v_{i,\mu^k}$  are not completely contained in a normal neighborhood of  $\mu^k$  we continue to sample until this criterion is met. We assume that the  $v_{i,\mu^k}$  are ordered such that the functional values of their exponentials increase, i.e.,

$$I(\text{Exp}_{\mu^k}(v_{i,\mu^k})) \leq I(\text{Exp}_{\mu^k}(v_{j,\mu^k})) \quad \text{for } 1 \leq i \leq j \leq m_1.$$

Then we define  $v'_{\mu^k}$  as the recombination of the  $m_2$ ,  $0 < m_2 \leq m_1$ , best samples:

$$v'_{\mu^k} := \sum_{i=1}^{m_2} w_i v_{i,\mu^k} \in T_{\mu^k}M.$$

Next we compute the updated path directions according to

$$v'_{c,\mu^k} := (1 - c_c) v_{c,\mu^k} + \sqrt{c_c(2 - c_c)m_{\text{eff}}} \frac{1}{\sigma^k} v'_{\mu^k} \in T_{\mu^k}M$$

and

$$v'_{\sigma,\mu^k} := (1 - c_\sigma) v_{\sigma,\mu^k} + \sqrt{c_\sigma(2 - c_\sigma)m_{\text{eff}}} \frac{1}{\sigma^k} C^{-\frac{1}{2}} v'_{\mu^k} \in T_{\mu^k}M.$$

The current path directions  $v'_{c,\mu^k}$  and  $v'_{\sigma,\mu^k}$  correspond to an exponentially smoothed accumulation of the previous directions. Both directions are normalized w.r.t. the current step size  $\sigma^k$ . The learning rates  $c_c$  and  $c_\sigma$  control the amount of the exponential smoothing. In contrast to  $v'_{c,\mu^k}$  the evolution path direction  $v'_{\sigma,\mu^k}$  for the estimation of the step size is independent of the direction of the eigenvectors of the covariance. This is achieved by multiplication with the root of the covariance  $C^{-\frac{1}{2}}$ .

Then we update the covariance matrix

$$C' := \underbrace{c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \frac{1}{(\sigma^k)^2} \sum_{i=1}^{m_2} w_i v_{i,\mu^k} v_{i,\mu^k}^T}_{\text{rank-}m_2\text{-update}} + \underbrace{(1 - c_{\text{cov}})C}_{\text{old covariance}} + \underbrace{\frac{c_{\text{cov}}}{\mu_{\text{cov}}} v'_{c,\mu^k} (v'_{c,\mu^k})^T}_{\text{rank-one-update}}.$$

The update of  $C$  is essentially the same as in the vector space setting [13] and consists of three terms: the rank- $m_2$ -update, the scaled covariance matrix of the last generation and the rank-one-update.

The rank- $m_2$ -update estimates the variances of the  $m_2$  best sampling points to their true mean, i.e., the mean of the original sampling distribution. It is called rank- $m_2$ -update because the rank of the matrix is at most  $\min(m_2, N)$ . For the rank- $m_2$ -update

to be a robust estimator of the covariance matrix, the population size  $m_2$  has to be large enough. However, large populations increase the runtime of a single step of the algorithm. Therefore, the estimated covariance is combined with the covariance computed in the previous step using exponential smoothing (old covariance).

The rank-one-update estimates the covariance based on the current evolution path direction. Consider a lot of good sample which are concentrated along a geodesic through the mean but are equally distributed on *both* sides of the mean. This situation does not necessarily indicate a direction towards a minimum, because in that case the optimal samples would concentrate on *one* side of the mean. In contrast to the rank- $m_2$ -update the rank-one-update does not increase the eigenvectors of  $C$  into such a directions. In other words, the rank-one-update incorporates sign information which is not present in the rank- $m_2$ -update. Note that for  $c_c = 1$ ,  $m_{\text{eff}} = 1$  and  $m_2 = 1$  the rank- $m_2$ -update and the rank-one-update coincide.

In the definition of  $C'$  the parameter  $c_{\text{cov}}$  controls the exponential smoothing of the influence of the old covariance and  $\mu_{\text{cov}}$  determines the relative weighting between the rank-one-update and rank- $m_2$ -update.

The step size

$$\sigma^{k+1} := \sigma^k \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{|v'_{\sigma, \mu^k}|}{\mathbb{E}|\mathcal{N}(0, \text{Id})|} - 1\right)\right)$$

is again adapted as in the vector space setting. Here,  $\mathbb{E}|\mathcal{N}(0, \text{Id})|$  is the expectation of the length of a  $\mathcal{N}(0, \text{Id})$ -distributed vector in  $\mathbb{R}^N$ , i.e.,

$$\mathbb{E}|\mathcal{N}(0, \text{Id})| = \sqrt{2} \frac{\Gamma((N+1)/2)}{\Gamma(N/2)}.$$

In [13] the author explains that the goal is to increase the step size if the updates of  $v_{\sigma, \mu^k}$  over successive generations point into similar directions and to decrease the step size if they cancel each other.

At this point we are ready to propagate the mean to the next point on  $M$ . We define

$$\mu^{k+1} := \text{Exp}_{\mu^k}(v'_{\mu^k}) \in M.$$

Then we transport the information we have to provide for the next iteration to  $T_{\mu^{k+1}}M$ . The updated covariance matrix  $C'$  defines a bilinear form  $b'_{\mu^k}$  on  $T_{\mu^k}M$ . We define

$$b_{\mu^{k+1}} := \text{the parallel transport of } b'_{\mu^k} \text{ to } \mu^{k+1}.$$

In the same spirit we update

$$v_{c, \mu^{k+1}} := \text{the parallel transport of } v'_{c, \mu^k} \text{ to } \mu^{k+1}, \quad \text{and}$$

$$v_{\sigma, \mu^{k+1}} := \text{the parallel transport of } v'_{\sigma, \mu^k} \text{ to } \mu^{k+1}.$$

This procedure is iterated until a stopping criterion is fulfilled. Such a criterion is for instance, that the functional value  $I(\text{Exp}_{\mu^k}(v_{i, \mu^k}))$  is similar for all  $i = 1, \dots, m_1$ .

### 3.2 Initialization

The algorithm has to be initialized by a first guess for  $\mu^0$ . Then we set  $b_{\mu^0} = \langle \cdot, \cdot \rangle_{\mu^0}$  and  $v_{c, \mu^0} = v_{\sigma, \mu^0} = 0$ . In accordance with the recommendation for the vector space

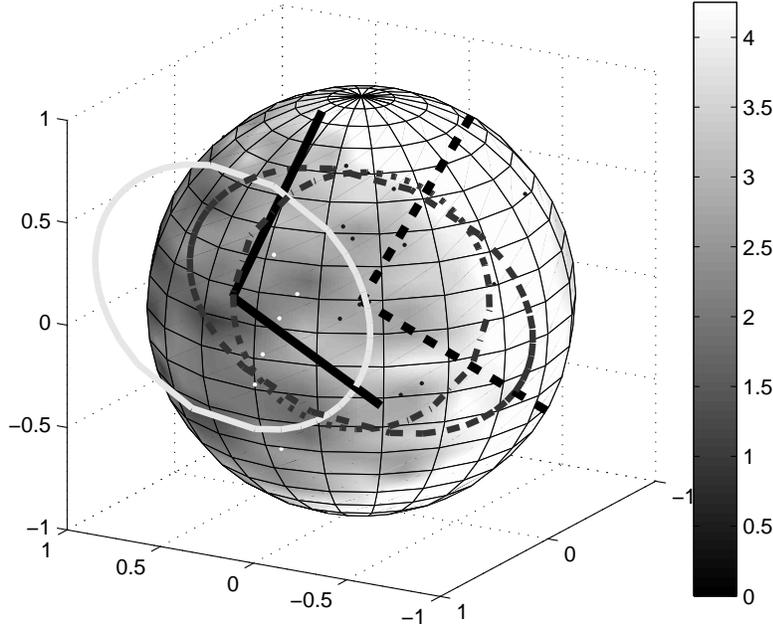


Figure 4: We illustrate the first iteration of the CMA-ES on  $S^2$ . We start with any initial guess  $\mu^0 \in S^2$ . The orthonormal basis  $e_1, e_2$  of  $T_{\mu^0}S^2$  is shown by the dashed black lines. Additionally the initial bilinear form  $b_{\mu^0}$  is indicated by the dashed-dotted circle in  $T_{\mu^0}S^2$ . Then 40 tangent vectors  $v_{i,\mu^0}$  are sampled. Here we displayed  $\text{Exp}_{\mu^0}(v_{i,\mu^0})$  by dots, where the  $m_2 = 10$  best are white colored while the others are black colored. The updated bilinear form  $b'_{\mu^0}$  in  $T_{\mu^0}S^2$  and the transported bilinear form  $b_{\mu^1}$  in  $T_{\mu^1}S^2$  is shown by the dashed and the light ellipse, respectively. The transported orthonormal basis is displayed by the black lines.

setting in [13], a reasonable choice of  $\sigma^0$  is such that the minimum of  $I$  is contained in the set

$$\{p \in M \mid d_M(\mu^0, p) < 3\sigma^0\}.$$

Moreover, the step size  $\sigma^k$  has to be sufficiently small, such that the generated tangent vectors  $v_{i,\mu^k}$ ,  $1 \leq i \leq m_1$  in (10) have a good enough chance to lie in a normal neighborhood of  $\mu^k$ .

### 3.3 The CMA-ES on $S^2$

We consider randomly generated functionals  $I : S^2 \rightarrow \mathbb{R}$ . The functionals have some local minima with values greater than zero, while the global minimum has value zero. An example of such a functional is shown in Figure 4. We want to find the global minimum by using the CMA-ES. Therefore we start with some  $\mu^0 \in S^2$  and initialize the ES with  $\sigma^0 = 1.05$ . In Figure 4 the first iteration step is displayed. After the

iteration procedure stops in a minimum, we restart the algorithm for 20 times with different  $\mu^0 \in S^2$ . In the example in Figure 4 the global minimum was found in all test runs. We iterate this process with different functionals  $I$ . The global minimum was detected in 63% of all cases, while a steepest descent algorithm found the global minimum in less than 2% of all cases.

## 4 Application

We investigate two methods to detect objects in voxel images based on 3D-MRep shape models. Let  $f : \Omega \rightarrow \mathbb{R}$  be a given image, where  $\Omega \subset \mathbb{R}^3$  is a domain. Moreover let  $\gamma : S^2 \rightarrow \Omega$  be a differentiable Jordan submanifold, which is homeomorphic to the unit sphere  $S^2$ .

### 4.1 A simplified Mumford-Shah functional

We consider a region-based segmentation method, i.e., we search for volumes in voxel images, which are characterized by a significant difference between the mean image intensity inside the volume and the mean intensity of the background. An example of this situation is the synthetic voxel data presented in Figures 6, 7 and 8 of Section 6.1. An object inside the synthetic voxel image is represented by values of 200 (the dark area), while the background intensity is set to 50. This clear contrast between inside and outside values makes a region-based segmentation method, like the one described below, applicable.

A Jordan submanifold  $\gamma$  separates  $\Omega$  into a volume inside and outside  $\gamma$ . We denote the inside volume as  $\mathcal{J}(\gamma)$  and the volume outside as  $\mathcal{O}(\gamma)$ . According to [4, 11] the simplified Mumford-Shah functional is defined as

$$I^{SMS}(\gamma) = \int_{\mathcal{J}(\gamma)} (u_1(\gamma) - f)^2 dx + \int_{\mathcal{O}(\gamma)} (u_2(\gamma) - f)^2 dx,$$

where

$$u_1(\gamma) = \frac{1}{|\mathcal{J}(\gamma)|} \int_{\mathcal{J}(\gamma)} f dx \quad \text{and} \quad u_2(\gamma) = \frac{1}{|\mathcal{O}(\gamma)|} \int_{\mathcal{O}(\gamma)} f dx. \quad (11)$$

Formulating this functional for the 3D-MRep shape model  $(M, \psi)$  yields  $F : M \rightarrow [0, \infty]$  defined by

$$F(p) = I^{SMS}(\psi(p)). \quad (12)$$

Thus,  $F$  maps a shape  $p$  to the simplified Mumford-Shah energy of its boundary  $\psi(p)$ . The goal is to find a shape  $p$ , which minimizes  $F$ .

Moreover we want to include statistical information about the appearance of a shape model into the solution process. According to Section 2.5 the Mahalanobis distance has appropriate properties. We define

$$I_\alpha(p) = F(p) + \alpha d_{M, \mu, \Sigma}^2(\mu, p). \quad (13)$$

The existence of a minimizer  $p_\alpha$  of  $I_\alpha$  for finite dimensional  $M$  is proven in [11]. Moreover,  $p_\alpha$  converges to  $\bar{p}$  for  $\alpha \rightarrow 0$ , if  $\bar{p}$  is a minimizer of  $F$ .

We use a CMA-ES and a gradient descent method to calculate a minimizer of  $I_\alpha$ . For the gradient based approach we have to evaluate the first derivative of  $I_\alpha$  w.r.t.

$p$ . Theorem 2 in Appendix A states that  $F$  is differentiable if  $f$  is continuous and  $\psi : M \rightarrow \mathcal{C}(S^2, \Omega)$  is differentiable. The derivative into direction  $\rho$  is then given by

$$DF(\psi(p))(\rho) = DI^{SMS}(\psi(p))D\psi(p)(\rho). \quad (14)$$

Because the Mahalanobis distance is differentiable [11], the regularization functional  $I_\alpha$  is differentiable.

The differences between the evolution strategy and the gradient based approach to minimize (13) are:

- For the CMA-ES we do not need to evaluate the gradient of  $I_\alpha$ . The computation of  $D\psi$  has to be done numerically and hence is computationally expensive for large  $N$ .
- The gradient descent method is trapped in local minima, whereas the CMA-ES is able to get around them (c.f. Section 3.3).

## 4.2 Edge-Based segmentation

Region-based segmentation cannot be applied, if the objects in the voxel data have nearly the same intensity as the surrounding volume. To overcome this problem, an edge-based segmentation method is introduced below. There are a lot of examples of such problems, for instance the detection of yeast cells in microscope images [11] or the segmentation of the cerebellum in 3D MRI voxel data in Section 6.2. We choose an edge-based segmentation method based on the Snake energy as introduced by Kass et al. [15]. We consider the functional

$$I_\beta(\gamma) = - \int_\gamma |\nabla f(\gamma(\sigma, \tau))| d\sigma d\tau + \frac{\beta}{2} \text{Surface}(\gamma)$$

for a Jordan submanifold  $\gamma$  and the voxel data  $f$ . The first term of this functional forces  $\gamma$  to be at locations where the gradient of  $f$  is high and the second term penalizes the surface area of  $\gamma$ .

Again we reformulate the functional for the 3D-MRep shape model  $(M, \psi)$  and define  $F : M \rightarrow [0, \infty]$  by

$$F_\beta(p) = I_\beta(\psi(p)). \quad (15)$$

We minimize this functional with the CMA-ES to avoid the calculation of the derivative  $DF$ .

## 5 Implementation

We implemented a library which provides functions to visualize and modify 3D-MReps. Furthermore it provides the integration routines necessary to implement the energy functionals for the automatic segmentation of voxel images. The library is an object oriented framework written in C++. For the visualization OpenGL is utilized and the GUI is based on Qt by Trolltech.

## 5.1 Computation of $\psi(p)$ for 3D-MReps

This section is concerned with the generation of the boundary surface that is implied by an instance of 3D-MReps (see Section 2). In the middle of Figure 5 the implied boundary of the shape model on the left hand side is represented as a closed triangulation. This polyhedral representation approximates a smooth  $G^1$  continuous surface that has been created using *Gregory patches*. Gregory patches [23] can be used to construct a smooth closed surface in  $\mathbb{R}^3$ . In the following we summarize this interpolation process.

The idea of (bicubic) Gregory patches is based on the definition of a *bicubic Bézier surface*

$$s(t_1, t_2) = \sum_{i=0}^3 \sum_{j=0}^3 s_{ij} B_i^3(t_1) B_j^3(t_2), \quad t_1, t_2 \in [0, 1],$$

where  $B_i^3$  and  $B_j^3$  are the Bernstein polynomials of degree 3 and  $s_{ij}$  are coordinates of the control points. A *bicubic Gregory patch* is defined as follows [9]:

$$s(t_1, t_2) = \sum_{i=0}^3 \sum_{j=0}^3 s_{ij}(t_1, t_2) B_i^3(t_1) B_j^3(t_2),$$

where the control points are functions of  $t_1$  and  $t_2$  defined by

$$\begin{aligned} s_{11}(t_1, t_2) &= \frac{t_1 s_{11}^{(t_2)} + t_2 s_{11}^{(t_1)}}{t_1 + t_2} \\ s_{21}(t_1, t_2) &= \frac{(1-t_1) s_{21}^{(t_2)} + t_2 s_{21}^{(t_1)}}{1-t_1+t_2} \\ s_{12}(t_1, t_2) &= \frac{t_1 s_{12}^{(t_2)} + (1-t_2) s_{12}^{(t_1)}}{t_1 + 1-t_2} \\ s_{22}(t_1, t_2) &= \frac{(1-t_1) s_{22}^{(t_2)} + (1-t_2) s_{22}^{(t_1)}}{1-t_1+1-t_2} \\ s_{ij}(t_1, t_2) &= s_{ij} \text{ for all other pairs } (i, j) \end{aligned}$$

Here, the  $s_{ij}^{(k)} \in \mathbb{R}^3$ ,  $1 \leq i, j, k \leq 2$ , are the eight fixed inner control points that have to be computed by the interpolation algorithm. Note, that this generalization of a bicubic Bézier surface achieves tangent plane continuity for an irregular quadrilateral interpolation mesh [23]. In our application the mesh is irregular because vertices of valence 3 exist.

The Gregory patch interpolation algorithm uses adjacent quadrilateral patches, each consisting of 4 boundary points and the corresponding normals, and joins them (i.e. computes the eight inner control points as mentioned above), such that a closed  $G^1$  surface is generated. The following scheme is used to construct a quadrilateral interpolation mesh including surface normals from 3D-MRep parameters. Suppose that we are given a shape instance  $p \in M$  with a regular  $n_1 \times n_2$  mesh of atoms. Every inner atom (see Section 2.6) defines two boundary points,  $y_1$  and  $y_2$ , and corresponding normals,  $z_1$  and  $z_2$ , which are computed as follows:

$$y_1 = x + rb_1, \quad y_2 = x + rb_2, \quad z_1 = rb_1, \quad z_2 = rb_2.$$

Every boundary atom defines an extra boundary point  $y_3$  and a normal  $z_3$  in the direction of  $b$ :

$$y_3 = x + \eta rb, \quad z_3 = rb.$$

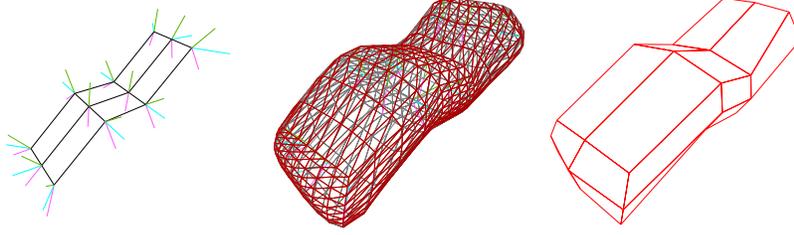


Figure 5: An instance of the 3D-MRep shape space (*left*) and its implied boundary (*middle*) and the initial interpolation mesh that is used to construct the surface (*right*). Note that the interpolation mesh is an *irregular* quadrilateral mesh, since it has vertices with valence 3 at each of the 4 corners of the figure.

From the equations above, we first obtain a mesh for the upper and lower side of an object using only the boundary vectors  $b_1$  and  $b_2$ , respectively, for all atoms. After that, a mesh for the border of the object is computed combining the vectors  $b_1$ ,  $b_2$  and  $b$  for all boundary atoms. All three meshes are then interpolated using the Gregory patch interpolation algorithm [23] to yield a set of Gregory patches interpolating all given boundary points and normals. Figure 5 shows an illustration of the initial interpolation mesh on the right hand side.

After the surface points and normals have been interpolated with this algorithm, a triangulation is constructed by evaluating the Gregory patches equidistantly along each parametric direction  $t_1$  and  $t_2$ , respectively.

## 5.2 Computation of the average inner/outer gray values

In this section we briefly describe how the average inner and outer values  $u_1$  and  $u_2$  in (11) w.r.t. an 3-dimensional input image  $f : \Omega \rightarrow \mathbb{R}$  and a closed surface  $\gamma$  are computed. This step contributes most to the total complexity of the segmentation algorithm. In the following we assume that  $\Omega = [0, a_1] \times [0, a_1] \times [0, a_1]$  for  $a_i > 0$ ,  $1 \leq i \leq 3$ .

To compute the sum of inner and outer gray values, i.e.,  $\int_{\mathcal{J}(\gamma)} f dx$  and  $\int_{\mathcal{O}(\gamma)} f dx$ , we take advantage of the divergence theorem

$$\int_{\mathcal{J}(\gamma)} \nabla \cdot G dx = \int_{\gamma} G \cdot dv, \quad (16)$$

where  $G : \Omega \rightarrow \mathbb{R}^3$  and  $v$  denotes the outer normal of  $\mathcal{J}(\gamma)$ . In our case  $\gamma = \psi(p)$  as described in Section 5.1. We compute the divergence field  $G$  for  $f$  as follows:

$$G(x_1, x_2, x_3) := \frac{1}{3} \begin{pmatrix} \int_0^{x_1} f(\zeta, x_2, x_3) d\zeta \\ \int_0^{x_2} f(x_1, \zeta, x_3) d\zeta \\ \int_0^{x_3} f(x_1, x_2, \zeta) d\zeta \end{pmatrix}. \quad (17)$$

Finally, applying (16) to (17), yields

$$\int_{\mathcal{J}(\gamma)} f dx = \int_{\gamma} G dv. \quad (18)$$

In the implementation we only compute  $G$  as in (17) once, and then evaluate the integral on the right hand side of (18) at each step to compute  $\int_{\mathcal{J}(\gamma)} f dx$  and  $\int_{\mathcal{O}(\gamma)} f dx$ ,

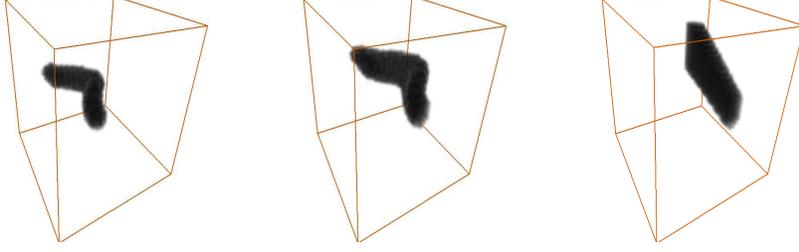


Figure 6: Three training shapes out of 30. The dark area illustrates the training sets  $w_i$ .

respectively. For the average inner and outer gray values  $u_1$  and  $u_2$  in (11) we have to normalize the latter integrals, i.e., divide them by  $|\mathcal{J}(\gamma)|$  and  $|\mathcal{O}(\gamma)|$ . These volumes can be calculated by applying the above method with the unit function  $\mathbb{1}$ .

## 6 Results

### 6.1 Segmentation of synthetic voxel images

In this subsection we apply the simplified Mumford-Shah functional as described in Section 4.1. We consider 30 synthetic voxel images  $f_i : \Omega \rightarrow \mathbb{R}$ ,  $1 \leq i \leq 30$ , which are scaled and shifted indicator functions of training sets  $\omega_1, \dots, \omega_{30} \subset \Omega$ :

$$f_i = 50 + 150 \chi_{\omega_i}, \quad 1 \leq i \leq 30.$$

A point  $x \in \Omega$  belongs to the training sets  $\omega_i$ , if the distance from  $x$  to a given curve in  $\mathbb{R}^3$  is smaller than a specified number. In detail, let  $\kappa_i : [0, 1]^2 \rightarrow \Omega$ ,  $1 \leq i \leq 30$  be the family of surface parametrizations

$$(t_1, t_2) \mapsto \begin{pmatrix} a_{1,i}t_1 + a_{2,i} \\ (a_{3,i}t_1 + a_{4,i})(a_{5,i}t_2 + a_{6,i}) + a_{7,i} \\ a_{8,i} \sin(\pi \cdot a_{9,i}t_1^{a_{10,i}})(a_{11,i}t_2 + a_{12,i}) + a_{13,i} \end{pmatrix},$$

where the parameters  $a_{i,j}$  are randomly distributed. Then the sets  $\omega_i$  are defined by

$$\omega_i = \{x \in \Omega : |x - \kappa_i| \leq c\}.$$

In Figure 6 three examples of the training shapes are shown.

As 3D-MRep shape space we choose

$$M = (\mathbb{R}^3 \times \mathbb{R}^+ \times S^2 \times S^2 \times \mathbb{R}^+)^8,$$

where the atoms are on a  $4 \times 2$  quadrilateral mesh. Then we minimized the simplified Mumford-Shah functional, i.e. (12), for all training data. Note that we do not use the regularization term here.

In Figure 7 the results for one of the training data exemplarily illustrates the differences between the CMA-ES and the gradient descent method. Both images are the result of 1000 iterations of the respective minimization methods. In CMA-ES we set

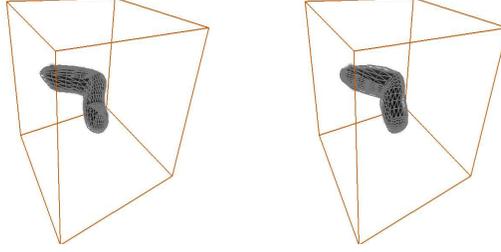


Figure 7: The results of the minimization of the simplified Mumford-Shah functional using the CMA-ES (*left*) and the gradient descent method (*right*). The surface of the results are displayed by the gray meshes, while the true training set is illustrated by the dark area. The functional values for the CMA-ES and the gradient descent method are  $F(p) = 1.9 \cdot 10^{-4}$  and  $F(p) = 7.0 \cdot 10^{-4}$ , respectively.

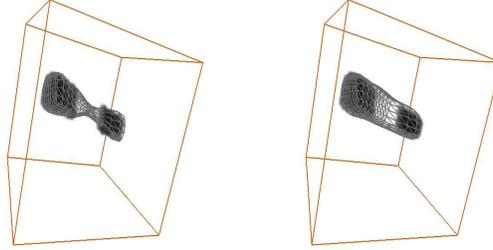


Figure 8: Minimization of the Mumford-Shah energy  $F$  (left) and Mumford-Shah regularization functional  $I_\alpha$  (right). Again the surface of the results are displayed by the gray meshes and the training set is shown by the dark area. In the left image the Mumford-Shah energy is  $F = 1.2 \cdot 10^{-3}$ , on the right  $F(p) = 1.5 \cdot 10^{-3}$  and  $R_\alpha = 3.9 \cdot 10^{-8}$ .

$m_1 = 200$ . Because each iteration of the gradient descent methods requires the numerical evaluation of the gradient of  $F$  w.r.t. the  $9 \cdot 8 = 72$  independent shape parameters, the computational complexities of both approaches are comparable. As we can see in Figure 7, the CMA-ES yields a more balanced triangulation than the gradient descent method.

Applying the above method to the 30 synthetic voxel images  $f_i$ , we yield 30 different shape instances  $p_i$ . These are used to calculate an aPGA, as outlined in Section 2.3. These statistics define the corresponding Mahalanobis regularization as in (9) and in consequence the regularized Mumford-Shah energy  $I_\alpha$  as in (13). We use this functional to segment a voxel data set, where parts of the information is occluded as shown in Figure 8. First, we apply the CMA-ES to minimize the simplified Mumford-Shah functional (without regularization, i.e.,  $\alpha = 0$ ) of this data set. The result is shown in the left image of Figure 8. Driven by the image data, the area between the two objects is minimized such that the recovered surface looks like a dumbbell. This effect does

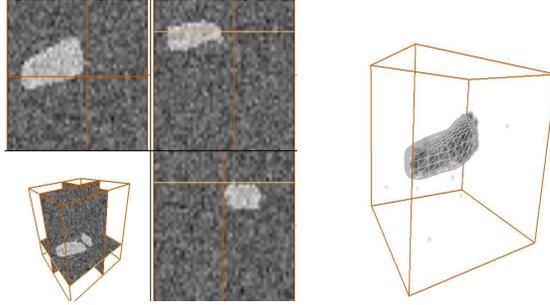


Figure 9: On the right hand side we see the surface of the result, which is displayed by the gray mesh. The training set (without noise) is illustrated by the dark area. On the left hand side we indicate the noisy voxel data by showing single slices of the data.

not appear, when we minimize the regularization functional  $I_\alpha$  with  $\alpha = 5 \cdot 10^{-4}$ . The result is shown in the right image of Figure 8 .

At last we consider the reconstruction of noisy data without regularization. We added normal distributed noise with standard deviation 40 to synthetic voxel data. In the left image of Figure 9 we can see the noisy data set and in the right image the result of the minimization with the CMA-ES of  $F$  is shown.

### 6.1.1 Complexity

In the following, we will briefly evaluate the complexity of the segmentation algorithm used in the above examples. Considering time complexity, the implementation of the above algorithm has two major obstacles:

1. The *computation of the average inner and outer values*  $u_1$  and  $u_2$ .
2. The *numerical differentiation of*  $\psi(p)$ .

Referring to Section 5.2, the implementation takes advantage of the divergence theorem, to minimize the time complexity for the computation of  $u_1$  and  $u_2$ . As we can see from this Section, the size of the input image  $f$  is only crucial in the initialization stage of the algorithm, whereas in each step of the minimization, only the boundary of the current model has to be parsed. This leads to the fact, that the time complexity of the main segmentation algorithm (i.e. the iteration process) does not depend on the size of the input image. The computation of  $u_1$  and  $u_2$  thus is only influenced by the number of faces of the boundary triangulation. However, it has to be mentioned, that we have to store the divergence field for the input image in advance, which can be of significant size. For example for the synthetic images above, the input size is  $50^3 = 125000$  voxels. The divergence field is a 3-dimensional vector field of the same size, i.e. the total size of the resulting array is  $125000 \cdot 3 = 375000$ . Assuming, that we are using double precision floating point values with 8 bytes size each, the divergence field has a total size of about 2.86 MB in memory. Considering the relatively low resolution of the input image, this value is large (e.g. for an input image of dimension  $256^3$ , the resulting size in memory would be 384 MB!).

For the numerical differentiation of  $\psi$ , we use a standard finite difference scheme. This step can be very time consuming, as we have to compute the numerical derivative in each parameter direction to get the gradient. Thus, it mainly depends on the

dimensionality of the model. Looking at the examples above, the input model has 72 shape parameters, implying 72 calculations of the finite difference for each step of the minimization algorithm.

Finally, we will compare the time for one iteration of the CMA-ES and the gradient descent method. All segmentations were carried out on a standard desktop PC using an AMD processor with 2.4 GHz and 2 GB RAM. Using the same parameters as in the Section above, one step of the CMA-ES took about 1.2 seconds, while the steepest descent algorithm lasted 0.4 seconds per iteration.

## 6.2 Segmentation of the cerebellum

In this example we are concerned with the segmentation of the cerebellum in 3D MRI voxel data  $f$ , which is displayed in the pictures in the left column of Figure 10. Here we see that the intensities in the cerebellum, the cerebrum and other tissue are more or less the same. Moreover, the intersection between the cerebellum and the cerebrum is not visible. To use the edge-based segmentation method as discussed in Section 4.2, we have to preprocess  $f$ . These preparatory steps include to threshold  $f$  under and over certain values and implement appropriate erosion and dilatation algorithms afterwards. We denote the resulting image by  $\tilde{f}$  and in the right column of Figure 10 we displayed  $|\nabla\tilde{f}|$ . The boundary of the cerebellum is clearly visible at the bottom, the front and backside. The boundary between cerebellum and cerebrum is available fractional.

Here, the cerebellum is modeled by

$$M = (\mathbb{R}^3 \times \mathbb{R}^+ \times S^2 \times S^2)^3 \times (\mathbb{R}^3 \times \mathbb{R}^+ \times S^2 \times S^2 \times \mathbb{R}^+)^{12}$$

as a shape space of 3D-MReps, where the atoms are on a  $5 \times 3$  quadrilateral mesh. Then we minimize the functional (15) with the CMA-ES method. The result is shown in Figure 11. The segmentation is very good apart from the region around the center of the cerebellum. In this area the chosen shape model is too inflexible to adapt itself to the true geometry of the volume we want to detect. Consequently, a model with more atoms in those regions would be needed to capture those details.

## Acknowledgments

The work of S.C., M.F., F.F. and O.S. has been supported by the Austrian Science Foundation (FWF), projects FSP9202-N12, FSP9203-N12 and FSP9207-N12. The work of F.F. was also supported by the Industrial Mathematics Competence Center (IMCC).

We would like to thank Wolfgang Recheis and Michael Verius from the Medical University Innsbruck for providing the MRI data sets of the human brain.

## A A simplified Mumford-Shah functional

In this section we investigate the differentiability of  $I^{SMS}$  introduced in (4.1). We denote the set of all differentiable maps  $S^2 \rightarrow \Omega$  by  $\mathcal{C}^1(S^2, \Omega)$ , the vector product in  $\mathbb{R}^3$  by  $\times$  and the inner product in  $\mathbb{R}^3$  by  $\langle \cdot, \cdot \rangle$ .

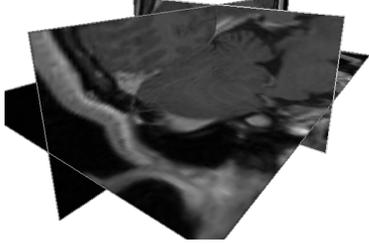
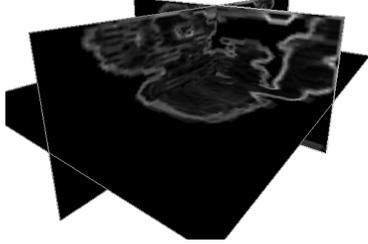
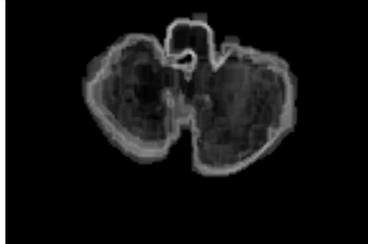
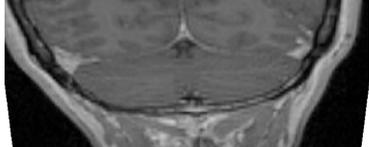
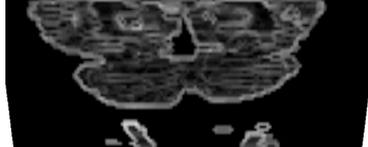
	original MRI voxel data	gradient of the data after the preprocessing
3D		
transversal		
coronal		
sagittal		

Figure 10: In the left column slices of the MRI voxel image  $f$  are shown. The corresponding slices of  $|\nabla \tilde{f}|$  can be seen in the right column.

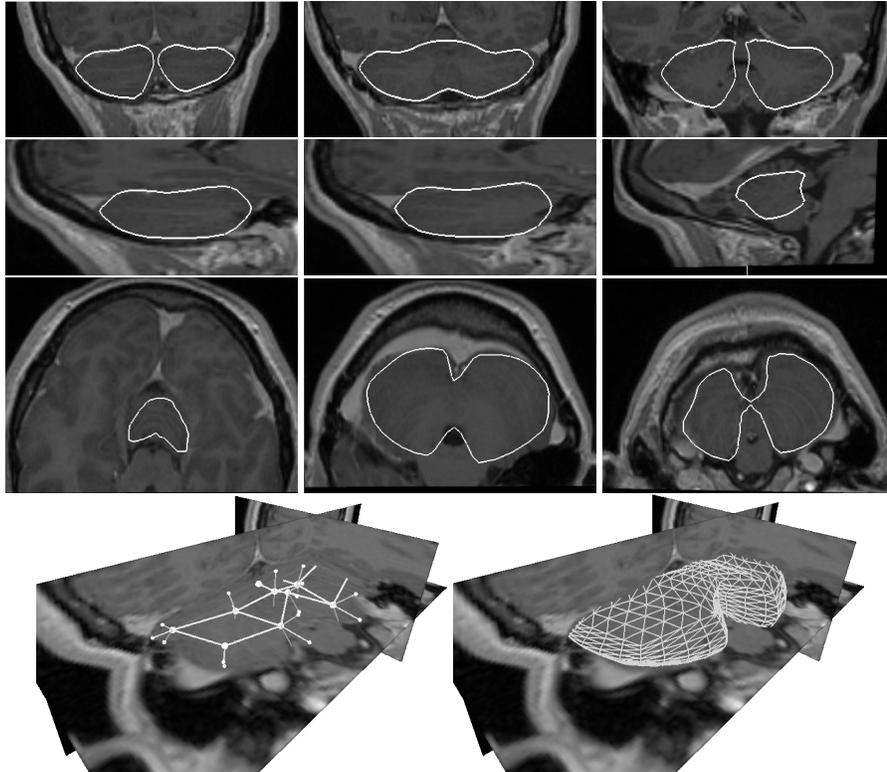


Figure 11: The result of the segmentation with the CMA-ES is shown. In the first three rows slices of the MRI data are displayed with the segmented lines. In the bottom row are a part of the MRep model (*left*) and the resulting surface (*right*).

**Lemma 1**

Let  $g : \Omega \rightarrow \mathbb{R}$  be a bounded function and let  $\gamma \in \mathcal{C}^1(S^2, \Omega)$  be a Jordan submanifold. Then the functional

$$I(\gamma) := \int_{\beta(\gamma)} g dx$$

is well-defined and Gateaux differentiable in  $\gamma$ . For a direction  $\rho \in \mathcal{C}^1(S^2, \Omega)$  the derivative is given by

$$\begin{aligned} DI(\gamma)(\rho) &= \int_0^1 \int_0^1 g(\gamma(t_1, t_2)) \left\langle \frac{\partial \gamma(t_1, t_2)}{\partial t_1} \times \frac{\partial \gamma(t_1, t_2)}{\partial t_2}, \rho(t_1, t_2) \right\rangle dt_1 dt_2 \\ &= \int_0^1 \int_0^1 g \circ \gamma \langle \gamma_{t_1} \times \gamma_{t_2}, \rho \rangle dt_1 dt_2. \end{aligned}$$

PROOF. Let  $h > 0$  and

$$\psi : [0, 1]^2 \times [0, 1] \rightarrow \Omega, \quad \psi(t_1, t_2, s) = \gamma(t_1, t_2) + hs\rho(t_1, t_2).$$

Then the functional determinant  $A := \det(J(\psi)(t_1, t_2, s))$  of  $\psi$  is

$$\begin{aligned} A &= h \left\langle \underbrace{\frac{\partial \gamma(t_1, t_2)}{\partial t_1} \times \frac{\partial \gamma(t_1, t_2)}{\partial t_2}}_{=: n_\gamma}, \rho(t_1, t_2) \right\rangle + \\ &\quad h^2 s \left\langle \underbrace{\left( \frac{\partial \gamma(t_1, t_2)}{\partial t_1} \times \frac{\partial \rho(t_1, t_2)}{\partial t_2} \right) - \left( \frac{\partial \gamma(t_1, t_2)}{\partial t_2} \times \frac{\partial \rho(t_1, t_2)}{\partial t_1} \right)}_{=: n_{\gamma, \rho}}, \rho(t_1, t_2) \right\rangle + \\ &\quad h^3 s^2 \left\langle \underbrace{\frac{\partial \rho(t_1, t_2)}{\partial t_1} \times \frac{\partial \rho(t_1, t_2)}{\partial t_2}}_{=: n_\rho}, \rho(t_1, t_2) \right\rangle. \end{aligned} \quad (19)$$

Now let  $h > 0$  be so small that  $\gamma + h\rho$  is a Jordan submanifold. Then we have

$$\begin{aligned} \int_{\Omega} (\chi_{\beta(\gamma+h\rho)} - \chi_{\beta(\gamma)}) g dx &= \\ \int_0^1 \int_0^1 \int_0^1 \text{sign}(\langle n_\gamma(t_1, t_2), \rho(t_1, t_2) \rangle) |A| g(\psi(t_1, t_2, s)) dt_1 dt_2 ds. \end{aligned} \quad (20)$$

The differential quotient of  $I$  w.r.t.  $\gamma$  into direction  $\rho$  is now

$$\begin{aligned} DI(\gamma)(\rho) &= \lim_{h \rightarrow 0} \frac{1}{h} (I(\gamma + h\rho) - I(\gamma)) \\ &= \lim_{h \rightarrow 0} \frac{1}{h} \left( \int_{\Omega} (\chi_{\beta(\gamma+h\rho)} - \chi_{\beta(\gamma)}) g dx \right). \end{aligned}$$

We apply now (19) and (20) to this formula and get

$$\begin{aligned} &\lim_{h \rightarrow 0} \frac{1}{h} \left( \int_{\Omega} (\chi_{\beta(\gamma+h\rho)} - \chi_{\beta(\gamma)}) g dx \right) \\ &= \lim_{h \rightarrow 0} \int_0^1 \int_0^1 \int_0^1 \text{sign}(\langle n_\gamma(t_1, t_2), \rho(t_1, t_2) \rangle) \left| \langle n_\gamma(t_1, t_2), \rho(t_1, t_2) \rangle + \right. \\ &\quad \left. hs \langle n_{\gamma, \rho}(t_1, t_2), \rho(t_1, t_2) \rangle + h^2 s^2 \langle n_\rho(t_1, t_2), \rho(t_1, t_2) \rangle \right| \\ &\quad g(\gamma(t_1, t_2) + hs\rho(t_1, t_2)) dt_1 dt_2 ds \\ &= \int_0^1 \int_0^1 \left\langle n_\gamma(t_1, t_2), \rho(t_1, t_2) \right\rangle g(\gamma(t_1, t_2)) dt_1 dt_2. \end{aligned}$$

The result is linear and continuous w.r.t.  $\rho$ .  $\square$

We use this lemma to prove that the simplified Mumford-Shah functional (4.1) is Gateaux differentiable.

**Theorem 2**

Let  $f : \Omega \rightarrow \mathbb{R}$  be a bounded function and  $\gamma \in \mathcal{C}^1(S^2, \Omega)$  be a Jordan submanifold. Then the functional  $I^{SMS}$  is Gateaux differentiable in  $\gamma$ . For a direction  $\rho \in \mathcal{C}^1(S^2, \Omega)$  the derivative is given by

$$\begin{aligned} DI^{SMS}(\gamma)(\rho) &= \int_0^1 \int_0^1 \left( (u_1(\gamma) - f(\gamma(t_1, t_2)))^2 - (u_2(\gamma) - f(\gamma(t_1, t_2)))^2 \right) \\ &\quad \left\langle \frac{\partial \gamma(t_1, t_2)}{\partial t_1} \times \frac{\partial \gamma(t_1, t_2)}{\partial t_2}, \rho(t_1, t_2) \right\rangle dt_1 dt_2 \\ &= \int_0^1 \int_0^1 \left( (u_1(\gamma) - f \circ \gamma)^2 - (u_2(\gamma) - f \circ \gamma)^2 \right) \langle \gamma_1 \times \gamma_2, \rho \rangle dt_1 dt_2. \end{aligned}$$

PROOF. First we rewrite  $I^{SMS}$  as

$$I'(a, b, \gamma) = \int_{\beta(\gamma)} (a - f)^2 dx + \int_{\Theta(\gamma)} (b - f)^2 dx$$

for  $a, b \in \mathbb{R}$  since  $u_1(\gamma)$  and  $u_2(\gamma)$  are constant values depending on  $\gamma$ . Obviously the derivative of  $I^{SMS}$  exists, if  $u_1, u_2$  and  $I'$  are differentiable. The derivatives of  $I'$  w.r.t.  $a$  and  $b$  are

$$\begin{aligned} D_a I'(a, b, \gamma)(c) &= 2 \int_{\beta(\gamma)} (a - f) c dx, \\ D_b I'(a, b, \gamma)(c) &= 2 \int_{\Theta(\gamma)} (b - f) c dx. \end{aligned} \tag{21}$$

Note that for any  $c \in \mathbb{R}$  we obtain that  $D_a(u_1(\gamma), b, \gamma)(c) = 0$  and  $D_b(a, u_2(\gamma), \gamma)(c) = 0$ . Thus we can omit the calculation of the derivatives  $Du_1(\gamma)$  and  $Du_2(\gamma)$ , since according to the chain rule these derivatives appear in  $DI^{SMS}(\gamma)$  only in combination with  $D_a I'$  and  $D_b I'$ , respectively.

Next we examine the derivative of  $I'$  w.r.t.  $\gamma$ . Using Lemma 1 we obtain

$$\begin{aligned} D_\gamma I'(a, b, \gamma)(\rho) &= \int_0^1 \int_0^1 (a - f(\gamma(t_1, t_2)))^2 \langle \gamma_1(t_1, t_2) \times \gamma_2(t_1, t_2), \rho(t_1, t_2) \rangle dt_1 dt_2 \\ &\quad - \int_0^1 \int_0^1 (b - f(\gamma(t_1, t_2)))^2 \langle \gamma_1(t_1, t_2) \times \gamma_2(t_1, t_2), \rho(t_1, t_2) \rangle dt_1 dt_2 \end{aligned}$$

Now we apply the chain rule and get the derivative of  $I^{SMS}$  in direction  $\rho$  as

$$\begin{aligned} DI^{SMS}(\gamma)(\rho) &= D_\gamma I'(u_1(\gamma), u_2(\gamma), \gamma)(\rho) + D_a I'(u_1(\gamma), u_2(\gamma), \gamma) Du_1(\gamma)(\rho) + \\ &\quad D_b I'(u_1(\gamma), u_2(\gamma), \gamma) Du_2(\gamma)(\rho). \end{aligned}$$

According to the comment after (21) the latter two terms vanish and thus the theorem is proven.  $\square$

## References

- [1] A. Abudhahir and S. Baskar. An evolutionary optimized nonlinear function to improve the linearity of transducer characteristics. *Meas. Sci. Technol.*, 19:045103, 2008.
- [2] P. Bayer and M. Finkel. Optimization of concentration control by evolution strategies: Formulation, application, and assessment of remedial solutions. *Water Resources Research*, 43:W02410, 2007.
- [3] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- [4] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10:266–277, 2001.
- [5] Yunmei Chen, Hemant D. Tagare, Sheshadri Thiruvankadam, Feng Huang, David Wilson, Kaundinya S. Gopinath, Richard W. Briggs, and Edward A. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.
- [6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [7] Daniel Cremers, Florian Tischhäuser, Joachim Weickert, and Christoph Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.
- [8] Weng Fang and Kap Luk Chan. Incorporating shape prior into geodesic active contours for detecting partially occluded object. *Pattern Recognition*, 40(7):2163–2172, 2007.
- [9] Gerald Farin. *NURB Curves and Surfaces*. A K Peters, Ltd., 1995.
- [10] P.T. Fletcher, C. Lu, S.M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23:995–1005, 2004.
- [11] M. Fuchs and O. Scherzer. Regularized reconstruction of shapes with statistical a priori knowledge. *International Journal of Computer Vision*, 79(2):119–135, 2008.
- [12] Muriel Gastaud, Michel Barlaud, and Gilles Aubert. Combining shape prior and statistical features for active contour segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):726–734, 2004.
- [13] N. Hansen. The CMA evolution strategy: A tutorial. Link from 03.07.2008: <http://www.bionik.tu-berlin.de/user/niko/cmatutorial.pdf>, 2007.
- [14] Sigurdur Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. Number 80 in Pure and Applied Mathematics. Academic Press, 1978.

- [15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.
- [16] S. Kern and N. Hansen. Evaluating the CMA evolution strategy on multimodal test functions. In *Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *LNCS*, pages 282–291. Springer, 2004.
- [17] S. Kern, N. Hansen, and P. Koumoutsakos. Fast quadratic local meta-models for evolutionary optimization of anguilliform swimmers. *EUROGEN 2007, Finland*, 2007.
- [18] S. Kern, S.D. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms — a comparative review. *Natural Computing*, 3:77–112, 2004.
- [19] Micheal E. Leventon, W. Eric L. Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 316–323, June 2001.
- [20] N.S. Mera. Passive gamma tomography reconstruction of layered structures in nuclear waste vaults. *Inverse Problems*, 23:385–403, 2007.
- [21] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–684, 1989.
- [22] Stephen M. Pizer, Andrew L. Thall, and David T. Chen. M-Reps: A new object representation for graphics. Technical Report TR99-030, 17, 1999.
- [23] M.A. Puso and T.A. Laursen. A 3-d contact smoothing algorithm method using gregory patches. *International Journal for Numerical Methods in Engineering*, 2002.
- [24] Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In *ECCV 2002 Proceedings, Part II*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2002.
- [25] Mikael Rousson and Nikos Paragios. Prior knowledge, level set representations & visual grouping. *International Journal of Computer Vision*, 76:231–243, 2007.
- [26] John P. Snyder. *Map Projections — A Working Manual*. Washington, DC : U.S. Government Print, 1987.
- [27] Andy Tsai, Anthony Yezzi, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric L. Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22(2):137–154, 2003.