Slovak University of Technology in Bratislava
Faculty of Civil Engineering

Msc. Polycarp Omondi Okock

Dissertation Thesis Abstract

Efficient software solutions for medical image processing and analysis

to obtain the Academic Title of *philosophiae doctor (PhD.)*
in the doctorate degree study programme
9.1.9 Applied Mathematics

Bratislava 2021

Dissertation Thesis has been prepared at Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology in Bratislava.

Submitter: MSc. Polycarp Omondi Okock
Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering, STU, Bratislava

Supervisor: Prof. RNDr. Karol Mikula, DrSc.
Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering, STU, Bratislava

Readers: Prof. RNDr. Daniel Ševčovič, DrSc.
Katedra aplikovanej matematiky a štatistiky
FMFI UK, Bratislava

Doc. Ing. Gabriel Okša, CSc.
Informatics Departmenf of Institute of Mathematics
Slovak Academy of Sciences, SAV, Bratislava

Doc. RNDr. Peter Frolkovič, PhD.
Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering, STU, Bratislava

Dissertation Thesis Abstract was sent: .....

Dissertation Thesis Defence will be held on ..... at ..... am/pm at Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology in Bratislava, Radlinského 11.

Prof. Ing. Stanislav Unčík, PhD.
Dean of Faculty of Civil Engineering

Abstract

The dissertation thesis deals with development of efficient algorithms for medical image processing and analysis. Because of the vast application area of the medical image processing, we have focused the thesis on the important high-level shape segmentation method. Specifically, **Atlas based 3D medical image segmentation algorithm**.

There are two stages in the proposed algorithm. In the first stage we construct an atlas from precisely annotated training set of shapes. The main procedure in this stage is shape registration. A reference shape, selected from the sample such that it is closest to the sample mean is set and all the other shapes are registered to it. This is done by an efficient 3D shape registration using signed distance maps and Stochastic Gradient Descent method (or Block Coordinate Descent method). The resulting atlas is then compressed using PCA, leaving only shapes highlighting its main feature. Then using suitable set of limits, dependent on the distribution and the training set size, similar shape estimation is performed. These similar shapes estimated from atlas are referred to as eigenshapes.

In the second stage, we apply segmentation using a modified Geodesic Active Contours model. To discretize the model, we followed a semi-implicit scheme for time discretization and in space the so called flux-based level set finite volume [14] method. A balloon model that adds an inflation term has been added onto the model. Following on the same approach as [13, 22], we have involved the atlas by modification of the external force term. For finer control of the inflation term, we have added control parameter that allows us to effectively reduce or turn off its effect. Therefore the model evolves the segmentation towards the edges, level sets are expanded in outer normal direction by the inflation term and evolution is regularized by curvature and globally depending on the estimated shape of the current segmentation from the atlas.

We proposed a simple function to estimate differences between the current and previous segmentation. This algorithm effectively and automatically determines what iterations to reduce or turn off the inflation term influence. The setup is such that the atlas is only involved after turning off the inflation term. At this iteration, the atlas is involved to drive segmentation. For estimation of similar shape, we have proposed a probability based estimation [13] model and energy based model [22]. Both give similar result but for fast computation, the probability based model is preferred. Numerical experiments for both synthetic and real data have also been presented.

keywords: atlas, segmentation, registration, estimation, inflation term

Abstrakt

Dizertačná práca sa zaoberá vývojom efektívnych algoritmov pre spracovanie a analýzu medicínskych obrazových dát. Keďže aplikačná oblasť spracovania medicínskeho obrazu je veľmi rozsiahla, v dizertačnej práci sme sa zamerali na jej veľmi dôležitú časť, spracovanie tvaru a segmentáciu. Konkrétne na **algoritmus atlasom riadenej segmentácie medicínskeho 3D obrazu**.

Predložený algoritmus má dve fázy. V prvej fáze zostavíme atlas z presne anotovanej cvičnej množiny tvarov. Hlavnou metódou v tejto fáze je registrácia tvaru. Zvolí sa referenčný tvar vybraný zo vzorky tak, aby bol čo najbližšie k strednej hodnote vzorky a všetky ostatné tvary sa k nemu zaregistrujú. To je umožnené pomocou efektívnej registrácie 3D tvarov pomocou znamienkových vzdialenostných máp a metódou Stochastic Gradient Descent (alebo metódou Block Coordinate Descent). Výsledný atlas je potom popísaný pomocou Analýzy Hlavných Komponent (PCA), pričom zanecháva iba tvary zvýrazňujúce jeho najviac významné charakteristické vlastnosť. Potom sa pomocou vhodnej sady limitov v závislosti od distribúcie a veľkosti cvičnej množiny tvarov určí približný odhad podobnosti tvaru. Tieto podobné tvary odhadnuté z atlasu na označujú ako vlastné tvary.

V druhej fáze segmentujeme obraz použitím modifikovaného modelu Geodesic Active Contours. Pri diskretizácii modelu sme sledovali semi-implicitnú schému pre časovú diskretizáciu a tzv. metódu flux-based level set finite volume [14] pre diskretizáciu v priestore. Do modelu bol pridaný tzv. balónový model, ktorý pridáva tzv. „nafukovací člen“. Sledujúc rovnaký prístupu ako v [13, 22], sme atlas zapojili pomocou rozšírenia “vonkajšieho silového člena“. Pre lepšiu kontrolu „nafukovacieho člena“ sme pridali kontrolný parameter, ktorý nám dovoľuje efektívne redukovať alebo vypnúť jeho pôsobenie. Preto model vyvíja segmentačnú funkciu smerom k hranám v obraze a zároveň ju expanduje („nafukuje“) „nafukovacím členom“ v smere vonkajšej normály. Vývoj je regularizovaný krivosťou a globálne závisí od tvaru odhadovaného z atlasu na základe tvaru súčasnej segmentácie.

Navrhli sme jednoduchú funkciu na výpočet rozdielov medzi dvomi (súčasnou a predchádzajúcou) segmentáciou. Tento algoritmus efektívne a automaticky určí, kedy počas iteračného procesu treba obmedziť alebo úplne vypnúť vplyv „nafukovacieho člena“. Atlas dostáva vplyv, až po vypnutí „nafukovacieho člena“. V tejto iterácii je atlas zapojený do riadenia segmentácie. Pre výpočet podobného tvaru sme navrhli model odhadu založený na pravdepodobnosti [13] a model založený na energii [22]. Oba modely podávajú podobný výsledok, ale pre rýchly výpočet je model pravdepodobnosti vhodnejší. V tejto práci boli taktiež predstavené numerické experimenty so syntetickými aj so skutočnými dátami.

kľúčové slová: atlas, segmentácia, registrácia, odhad, inflation term

Contents

1 Introduction

The overall goal of the research is to develop efficient algorithms for medical image processing and analysis. Hence we focused on development of a high-level 3D image segmentation, specifically our work in the dissertation thesis is focused on development of **an atlas-based 3D medical segmentation algorithm**. In general atlas based segmentation have shown significant importance in clinical practices. Our inspiration came from [22], and we have picked some of the ideas and extended with our own to develop the algorithm. The proposed algorithm is able to deform in ways characteristic of the class of objects it represents while at the same time segmenting images with issues such as *low spatial (or temporal) resolution, ill-defined boundary, poor contrast, acquisition artifact or other noise*. The algorithm is a composite of several algorithms, optimized and tune specifically to improve efficiency of the atlas-based segmentation. The dissertation has mainly focused on 3D numerical algorithms.

There are two main phases needed to realize the proposed algorithm. The first objective was to extend previous works, see [22] to 3D data. All the algorithms have been written for 3D data. The change from 2D to 3D images, can lead to intensive computations and memory requirements. As a result, the second objective was to tune current algorithms to run more efficiently and incorporate into the algorithm newer and optimized ones. The third objective was to efficiently improve on the balloon model incorporated into the segmentation model. For images with weak or missing edge information or noised the inflation force from this model can overpower those forces.

The first phase involves atlas construction. Exact shapes created by an expert are used as samples to create the training set. They are represented using signed distance function. For efficiency, we have used Fast Sweeping method [11]. We first select a reference shape which is most similar to the samples mean shape. All the remaining samples are registered to this sample. For registration itself, we have developed three efficient optimization algorithms for calculating the 3D affine transformation matrix. They include a modified efficient standard gradient descent method, parallelized and efficient stochastic gradient descent method that requires few number of points and block coordinate descent method that makes it easier to find a group of transformation that registers the shapes efficiently. Depending on the nature of the problem being solved, any of the algorithm can be applied. They are all efficient, but from our experience for large volumes, stochastic gradient descent method is preferable.

After registration of the shapes, we have applied PCA approach (similarly to [13, 22]) to get a compressed atlas represented by eigenshapes. We have described deeper on how to select suitable limits for the feature vector of weights. This is to ensure that the estimated shapes deforms in only ways allowed by the class of objects they represents. These suitable limits are determined by the distribution and training set size. As a general case, basing the limits on the **empirical rule** is the safe approach (assumes normal distribution) but we have observed for a small sample size less stricter approach can be applied such as **Chebyshev's inequality**[2, 16] or **Mahalanobis distance** [3]. In our implementation of similar shape estimation we have followed the guidelines to achieve reasonable shape estimations.

Usually the atlas construction phase is the most computational intensive stage. The advantage is that it is only needed to be run once. It is also the reason why we have performed several computation optimizations in the stage. These include 'careful' selection of the narrowband δ, carried out registration only within bounding box around ROI during registration, calculation of distance values to the interface for only the points required in SGD and parallelization of the brute force method to calculate the distance values.

In the second phase, similar to [13, 22] we applied a modified GAC segmentation model. To discretize the model, we followed a semi-implicit approach for time discretization and in space the so called flux-based level set finite volume [14] method. A balloon model that adds an inflation term has been added onto the model. Following on the same approach as [13, 22], we have involved the atlas by modification of the external force term. For

finer control of the inflation term, we have added a control parameter that allows for to effective reduction or turn off its influence. This is inline with our objective to effectively manage its influence. Therefore the model evolves the segmentation towards the edges, level sets are expanded in outer normal direction by the inflation term and evolution is regularized by curvature and globally depending on the estimated shape from the atlas.

We proposed a simple estimation algorithm to estimate differences between the current and previous segmentation. This algorithm effectively and automatically determines what iterations to reduce or turn off the inflation term influence. The setup is such that the atlas is only involved after turning off the inflation term. At this iteration, the atlas is involved to drive segmentation. For estimation of similar shape, we have proposed a probability based estimation [13] and energy based [22] models. Both give similar result but for fast computation, the probability model is preferred.

We have set different experiments, showing the advantages of the approaches we have undertaken. Specifically for very noisy synthetic and real images. The challenge with finding the correct set of parameters for segmentation have been made simpler by the proposed approach to automatically handle the inflation term. The estimated shapes are within the class of objects that we are segmenting. We plan for future developments to extend it to include other biological data.

The thesis is organized as follows. In Chapter 2, we describe the creation of the atlas. This is made possible by an efficient 3D registration model we developed particularly for it. In Chapter 3, we described the segmentation model. We first introduce the basic model and show how atlas is involved into this model. Along we also describe its numerical discretization. Finally in Chapter 4, we test the model on real data.

2 Atlas construction

The atlas is created from a set of 3D shapes. We have two stages to construct a usable atlas. In the first stage we describe an automatic efficient 3D registration technique. In the second stage we apply PCA-based approach to compress the atlas.

2.1 Automatic efficient 3D registration techniques

A reference shape is selected from the set and then we register the other shapes to the reference shape. The registration between two sets of shapes is described as follows.

We follow the general registration formulation which is as follows: let $\mathbf{F}, \mathbf{M} \subset \mathcal{R}^3$ be the fixed and moving shapes respectively. Our objective is to find a 3D affine transformation matrix, $\mathbf{A}$, that maps $\mathbf{M}$ to $\mathbf{F}$ while minimizing the dissimilarity measure between the transformed shape $\mathbf{A}(\mathbf{M})$ and the fixed shape $\mathbf{F}$. We use sum of squared differences (SSD) as the dissimilarity measure, minimized using gradient descent methods.

This methods have the advantage of by-passing the need for tedious manual labeling and developing point correspondence between set of shapes used to form the atlas. This technique has been described in our publication [1]. In this article, we published various optimzation techniques presented in the next section and the accompanying numerical results.

To represent a shape, we view it as a curve or surface and use level set methods [7] to describe it. The shape representation is given as a zero level set (interface) of some function. We use signed distance function to such zero level set in the registration procedure. We have implemented the Fast Sweeping method [11] and the "brute force" method to calculate the Euclidean distance between couple of points.

The goal of the registration is to find the transformation that maps one space to another space. Usually the shape we are mapping to is defined as the fixed shape whereas the shape that we are mapping to the fixed shape is referred to as the moving shape. Complexity is determined by what transformation model we are using. We have chosen to use the *affine* transformation model. This model is invariant to *translation*, *rotation* and *scaling*

transformations. We wish to find the *affine* transformation that minimizes the differences between the moving and fixed shape defined in $\mathcal{R}^3$.

Let $\mathbf{F}, \mathbf{M} \subset \mathcal{R}^3$ be the fixed and moving shapes respectively. The objective is to find the 3D affine transformation matrix, $\mathbf{A}$, that maps $\mathbf{M}$ to $\mathbf{F}$. The transformation matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{R}.$$

$\mathbf{R}$ is the 3D rotation matrix defined as

$$\begin{bmatrix} \cos\psi\cos\theta & -\cos\theta\sin\psi & \sin\theta & 0 \\ \cos\phi\sin\psi + \cos\psi\sin\phi\sin\theta & \cos\phi\cos\psi - \sin\phi\sin\psi\sin\theta & -\cos\theta\sin\phi & 0 \\ \sin\phi\sin\psi - \cos\phi\cos\psi\sin\theta & \cos\psi\sin\phi + \cos\phi\sin\psi\sin\theta & \cos\phi\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The rotation direction is anticlockwise with the angles ϕ →x direction, θ →y direction, ψ →z direction). Rotation in 3D is not commutative and the order of rotation is important. We chose the following **XYZ** order. The x rotation, r_x, y rotation, r_y and z rotation, r_z are defined as:

$$r_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, r_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, r_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore $\mathbf{R} = r_x \cdot r_y \cdot r_z$. $\mathbf{S}$, the scaling matrix and $\mathbf{T}$, the translation matrix, are given by:

$$\mathbf{S} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 & 0 \\ 0 & 0 & \frac{1}{s_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore the transformation matrix $\mathbf{A}$ is given as

$$\begin{pmatrix} \frac{\cos(\psi)\cos(\theta)}{s_x} & -\frac{\cos(\theta)\sin(\psi)}{s_x} & \frac{\sin(\theta)}{s_x} & -t_x \\ \frac{\cos(\phi)\sin(\psi)+\cos(\psi)\sin(\phi)\sin(\theta)}{s_y} & \frac{\cos(\phi)\cos(\psi)-\sin(\phi)\sin(\psi)\sin(\theta)}{s_y} & -\frac{\cos(\theta)\sin(\phi)}{s_y} & -t_y \\ \frac{\sin(\phi)\sin(\psi)-\cos(\phi)\cos(\psi)\sin(\theta)}{s_z} & \frac{\cos(\psi)\sin(\phi)+\cos(\phi)\sin(\psi)\sin(\theta)}{s_z} & \frac{\cos(\phi)\cos(\theta)}{s_z} & -t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The goal is to find the optimal set of parameters $\phi, \theta, \psi, s_x, s_y, s_z, t_x, t_y, t_z$ that minimizes the difference between the fixed shape $\mathbf{F}$ and moving shape $\mathbf{A}(\mathbf{M})$.

2.1.1 Registration optimization procedure

For functional defined on a parameter space we attempt to quantify the similarity between two distance maps. For our case we have chosen to use sum of squared differences (SSD). The optimization criterion, $E(\mathbf{A})$, is defined as:

$$E(\mathbf{A}) = \int_{\mathbf{D}} \left[\Phi_F(x,y,z) - \Phi_M(\mathbf{A}(x,y,z))\right]^2 dxdydz, \tag{1}$$

where $\Phi_{\mathbf{F}}(x,y,z)$, $\Phi_{\mathbf{M}}(x,y,z)$ are the fixed and moving distance maps respectively. $\mathbf{A}$ is the 3D affine transformation matrix and $E(\mathbf{A})$ is the **SSD/Energy** we want to minimize by applying the optimal $\mathbf{A}$. Similarly to [10] we reduce the calculation to a narrow band in the distance δ around the inputs. We define the narrow

band as:

$$N_\delta(\Phi_1, \Phi_{2)}) = \begin{cases} 1, & min(|\Phi_1|, |\Phi_2|) \leq \delta \\ 0, & min(|\Phi_1|, |\Phi_2|) > \delta. \end{cases} \tag{2}$$

The constrained optimization eq. (1) criterion becomes

$$E(\mathbf{A}) = \int_{\mathbf{D}} N_\delta(\Phi_F, \Phi_M) \left(\Phi_F(x,y,z) - \Phi_M((\mathbf{A}(x,y,z))\right)^2 dxdydz. \tag{3}$$

For minimization and calculation of particular components we chose gradient descent method for $\nabla E(\mathbf{A})$

$$\begin{aligned} \partial_{\mathbf{R}} E(\mathbf{A}) &= 2\int_{\mathbf{D}} N_\delta(\Phi_F, \Phi_M)(\nabla\Phi_M \cdot \nabla_R(\mathbf{Ax}))(\Phi_F - \Phi_M(\mathbf{A}))dxdydz \\ \partial_{\mathbf{S}} E(\mathbf{A}) &= 2\int_{\mathbf{D}} N_\delta(\Phi_F, \Phi_M)(\nabla\Phi_M \cdot \nabla_S(\mathbf{Ax}))(\Phi_F - \Phi_M(\mathbf{A}))dxdydz \\ \partial_{\mathbf{T}} E(\mathbf{A}) &= 2\int_{\mathbf{D}} N_\delta(\Phi_F, \Phi_M)(\nabla\Phi_M \cdot \nabla_T(\mathbf{Ax}))(\Phi_F - \Phi_M(\mathbf{A}))dxdydz \end{aligned} \tag{4}$$

For better understanding of the formula $\nabla E(\mathbf{A})$, the $\mathbf{x}$ in $(\mathbf{Ax})$ is defined as

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix},$$

and $\mathbf{Ax}$ is

$$\begin{pmatrix} \frac{x\cos(\psi)\cos(\theta)}{s_x} - \frac{y\sin(\psi)\cos(\theta)}{s_x} + \frac{z\sin(\theta)}{s_x} - t_x \\ \frac{x(\sin(\phi)\cos(\psi)\sin(\theta)+\cos(\phi)\sin(\psi))}{s_y} + \frac{y(\cos(\phi)\cos(\psi)-\sin(\phi)\sin(\psi)\sin(\theta))}{s_y} - \frac{z\sin(\phi)\cos(\theta)}{s_y} - t_y \\ \frac{x(\sin(\phi)\sin(\psi)-\cos(\phi)\cos(\psi)\sin(\theta))}{s_z} + \frac{y(\cos(\phi)\sin(\psi)\sin(\theta)+\sin(\phi)\cos(\psi))}{s_z} + \frac{z\cos(\phi)\cos(\theta)}{s_z} - t_z \\ 1 \end{pmatrix}$$

The grad of Φ_M, $\nabla\Phi_M$ is the row vector defined as

$$\nabla\Phi_M = (\partial_x\Phi_M, \partial_y\Phi_M, \partial_z\Phi_M, 0),$$

and gradient for the components are the following column vectors

$$\begin{aligned} \nabla_R(\mathbf{Ax}) &= \left(\partial_{r_x}\mathbf{Ax}, \partial_{r_y}\mathbf{Ax}, \partial_{r_z}\mathbf{Ax}, 0\right)^T \\ \nabla_S(\mathbf{Ax}) &= \left(\partial_{s_x}\mathbf{Ax}, \partial_{s_y}\mathbf{Ax}, \partial_{s_z}\mathbf{Ax}, 0\right)^T \\ \nabla_T(\mathbf{Ax}) &= \left(\partial_{t_x}\mathbf{Ax}, \partial_{t_y}\mathbf{Ax}, \partial_{t_z}\mathbf{Ax}, 0\right)^T \end{aligned}$$

In discrete case, we can obtain optimal affine transformation similarly. Let n be the number of points in the narrow band and $n \lll N$, where N is the total number of points. Functional eq. (3) can be rewritten in discrete formulation as

$$E_D(\mathbf{A}) = \frac{1}{n}\sum_{i=1}^{n} \left(\Phi_{F,i} - \Phi_{M,i}(\mathbf{A})\right)^2. \tag{5}$$

Approximation of gradient components in eq. (4) are given as

$$\partial_R E_D(\mathbf{A}) = 2\frac{1}{n}\sum_{i=1}^{n}(\nabla\Phi_{M,i}\cdot\nabla_R(\mathbf{Ax})(\Phi_{F,i}-\Phi_{M,i}(\mathbf{A})) \tag{6}$$

$$\partial_S E_D(\mathbf{A}) = 2\frac{1}{n}\sum_{i=1}^{n}(\nabla\Phi_{M,i}\cdot\nabla_S(\mathbf{Ax})(\Phi_{F,i}-\Phi_{M,i}(\mathbf{A}))$$

$$\partial_T E_D(\mathbf{A}) = 2\frac{1}{n}\sum_{i=1}^{n}(\nabla\Phi_{M,i}\cdot\nabla_T(\mathbf{Ax})(\Phi_{F,i}-\Phi_{M,i}(\mathbf{A}))$$

In addition, the standard gradient descent method can take many iterations to compute a local minimum with required accuracy. For larger domains, this process can be significantly slow. Because of this we have also implemented stochastic gradient descent [4]. These methods have been described in detail in our publication, see [1]. More we have implemented block coordinate descent [21] method described as follows.

The coordinate descent method [21] evaluates the optimization criterion function at a single component of $\nabla E(\mathbf{A})$ at the current point in the narrow band. The current point is selected in the order they appear in the narrow band.

Let $\mathbf{w}_{\mathbf{A}_k}$ be the current component from our block of components $\mathbf{A}_k : k \in \{\mathbf{T},\mathbf{S},\mathbf{R}\}$. The component is selected in a cyclic fashion, in which $\mathbf{w}_{\mathbf{A}_0} = 0$ and

$$\mathbf{w}_{\mathbf{A}_{k+1}} = [\mathbf{w}_{\mathbf{A}_k} \bmod \mathbf{T}], k = 0, 1, 2\cdots \text{ and } \mathbf{T} = 3$$

Each component is modified at least once in τ_{max} iterations. The step size $\lambda_{\mathbf{A}_k}$ is sufficiently decreased as done in traditional line search algorithms. The coordinate descent algorithm is as follows

1. Set $k = 0, \tau = 0$, initial $\mathbf{w}^{\tau}_{\mathbf{A}_k}$ component from block of components $\mathbf{A}^{\tau}_k$ and $\lambda_{\mathbf{A}_k}$
2. Repeat until max no. of iterations, τ_{max}, or accepted tolerance for $E(\mathbf{A})$ is reached
   (a) **For each** $\mathbf{w}^{\tau}_{\mathbf{A}_k} : \tau = \{1, 2, \cdots, \tau_{max}\}$ **do**:

$$\mathbf{w}^{\tau}_{\mathbf{A}_k} := \mathbf{w}^{\tau-1}_{\mathbf{A}_k} - \lambda_{\mathbf{A}_k}\frac{1}{n}\sum_{i=1}^{n}\frac{\partial E_i(\mathbf{A}^{(\tau-1)}_k)}{\partial \mathbf{w}}$$

   (b) $k \leftarrow k + 1$

The atlas construction is an iterative procedure. The algorithm is summarized as follows:

1. Select a reference shape most similar to set of shapes mean shape.
2. Register the remaining shapes to the reference shape.
3. Re-calculate a new mean shape from the transformed shapes.
4. If the estimated mean shape has changed, return to step 1.

Convergence is declared when the mean shape has not changed significantly within an iteration.

2.2 Using PCA to compress the atlas

In this section, we describe how we use principal component analysis (PCA) to compress the atlas created from methods described in the previous section. We first discuss PCA implementation for the registered shapes, then calculation of eigenshapes. Eigenshapes are calculated from k dimensional eigenspace and highlights the main atlas features.

In Figure 1 registered shapes have been plotted in what we call an 'allowable shape domain', with the mean shape overlaid as a white mesh. It can be seen that some of the shapes are significantly different from the set

whereas some are very similar. This means that some of the shapes are highly correlated, therefore it should be possible to drop some of them without losing information. The process of removing these highly correlated shapes, we refer to it as compressing the atlas. We expect to only have shapes highlighting the main atlas features after removal of highly correlated ones. This process is known as dimensionality reduction.

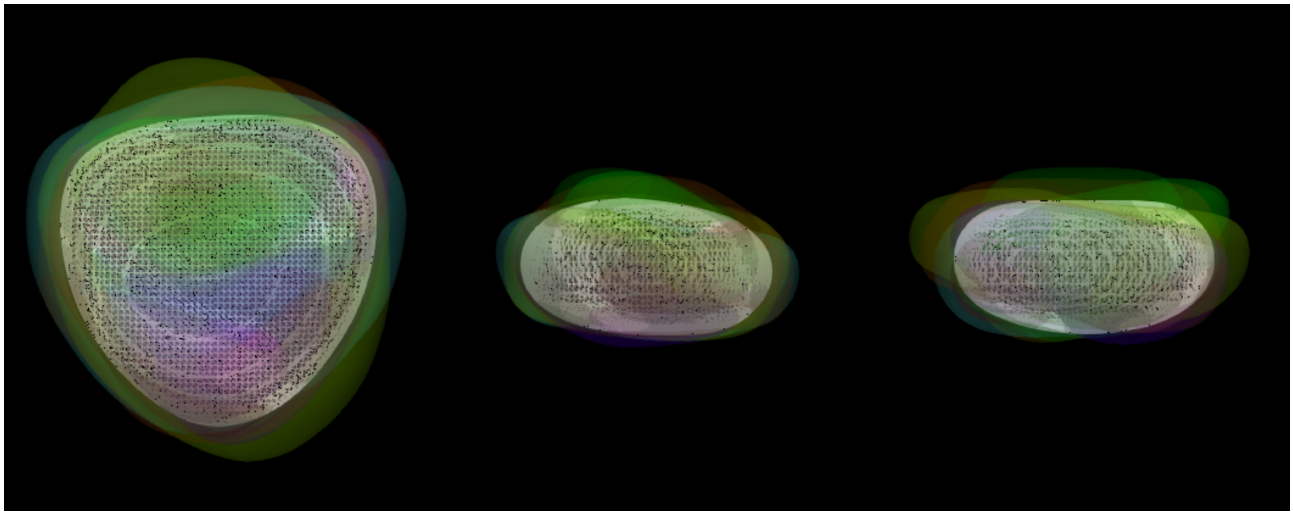


Figure 1: Aligned shapes with the mean shape represented by the white mesh

To compress the atlas by checking the correlation between the shapes we use PCA [15]. Every n^{th} aligned shape can be represented as an $N-$ dimensional point in the 'allowable shape domain'. Every $N-$ dimensional point in this domain gives a shape similar to the shapes in the original aligned atlas. Therefore by combining different $N-$ dimensional point, we can generate new shapes allowable' even though they are not in the original atlas.

Following similar steps as done in [6, 13, 22], we assume the 'allowable shape domain' is approximately ellipsoidal. We then proceed to calculate the mean aligned shape from set of transformed shapes. Given the set $\{s_1, s_2, \cdots, s_n\}$ of $N \times n$ of registered shapes, the mean aligned shape, $\bar{s}$, center of the allowable shape domain, is calculated as follows

$$\bar{s} = \frac{\sum_{j=1}^{n} s_j}{n}. \tag{7}$$

We translate each shape to the domain center by subtracting from $\bar{s}$ to form a new centered set $\mathbf{S}$. The new set of center aligned shapes,$\mathbf{S}$, is defined as

$$\mathbf{S} = \left[s_1 - \bar{s}\,, s_2 - \bar{s}\,\cdots\,, s_n - \bar{s}\right].$$

Our goal is calculate the principal axes of this ellipsoid. Different features of the atlas can be described by these principal axes. Through their linear combination we can generate new shapes. We use PCA to calculate the principal axes. It has been shown in the literature's [6, 15] that the longest axes of the ellipsoid can be described by the eigenvectors. Therefore, we calculate the eigenvectors of covariance matrix, Σ defined as follows:

$$\Sigma_{N\times N} = \frac{1}{n}\mathbf{S}\mathbf{S}^T.$$

The covariance matrix has the following properties

- Square
- Symmetric
- Positive semi-definite
- It can be very large, because N is the number of voxels in each shape level set representation.

As mentioned before the principal axes can be described by the eigenvectors of the covariancee matrix such that

$$\Sigma \mathbf{U} = \Lambda \mathbf{U}.$$

To find the eigenvectors of Σ, we use SVD such that

$$\Sigma = \mathbf{U}\Lambda\mathbf{U}^T,$$

any $\mathbb{M} \times \mathbb{N}$ matrix can be written as product of 3 matrices. Where

- $\mathbf{U}$ is an $\mathbb{M}\times\mathbb{M}$ matrix whose columns are orthonormal vectors. These orthonomral vectors are also referred to as eigenvectors
- Λ is an $\mathbb{N} \times \mathbb{N}$ diagonal matrix, whose diagonal values are known as singular values of Σ. They are also known as the eigenvalues.

For our case, $\mathbb{M} = \mathbb{N} = N$. We mentioned that Σ is very large since it is $N \times N$ matrix. Instead we consider the matrix

$$\mathbf{L}_{n\times n} = \frac{1}{n}\mathbf{S}^T\mathbf{S}.$$

- Both Σ and $\mathbf{L}$ are symmetric, but $\Sigma \neq \mathbf{L}$
- Σ is $N \times N$ while $\mathbf{L}$ is $n \times n$ matrix
- n is the number of aligned shapes, typically $n \lll N$

Let $\mathbf{e}$ be the eigenvector of $\mathbf{L}$ with an eigenvalue λ then

$$\begin{aligned} \mathbf{L}\mathbf{e} &= \lambda\mathbf{e} \\ \frac{1}{n}\mathbf{S}^T\mathbf{S}\mathbf{e} &= \lambda\mathbf{e} \\ \frac{1}{n}\mathbf{S}\mathbf{S}^T\mathbf{S}\mathbf{e} &= \lambda\mathbf{S}\mathbf{e} \\ \Sigma\left(\mathbf{S}\mathbf{e}\right) &= \lambda\left(\mathbf{S}\mathbf{e}\right) \end{aligned} \tag{8}$$

From eq. (8), $\mathbf{Se}$ is the eigenvector of Σ, with eigenvalue λ. Therefore by calculating the eigenvectors and eigenvalues of the matrix $\mathbf{L}$, we can calculate the eigenvectors and eigenvalues from these more efficiently.

The variance described by the each eigenvector are explained by the magnitude of the corresponding eigenvalue. The eigenvalues normally decrease fast, meaning majority of the variation can be explained by the first few eigenvectors. Similar to the approaches in [6, 13, 22], we calculate k components such that the sum of their total variations, $\lambda_T = \sum_{j=1}^{n} \lambda_j$, explains a large proportion of the variations. We do this by setting a threshold, $\mathbf{T} \in (0, 1)$ such that the sum of ordered eigenvalues is $\leq \mathbf{T}$:

$$\sum_{j=1}^{k} \frac{\lambda_j}{\lambda_T} \leq \mathbf{T}$$

2.2.1 Eigenshapes formulation

We mentioned that the columns of $\mathbf{U}$ are orthonomral vectors also known as the eigenvectors. The eigenvectors span an eigenspace and any shape in 'allowable shape domain' can be converted to the eigenspace. This is done as follows. Let y be a new 'allowable' shape, it can be represented by k principal components in k-dimensional space as follows:

$$\alpha_k = \mathbf{U}_{k\times N}^T \left(y - \bar{s}\right), \tag{9}$$

where α_k is a feature vector of weights for each eigenshape. Any shape in the aligned set can be approximated using the linear combination of the mean aligned shape and weighted sum of the k principal components as follows

$$\tilde{y} = \mathbf{U}_{N\times k}\alpha_k + \bar{s} \tag{10}$$

Shapes generated using eq. (10) are known as eigenshapes. Calculation in the eigenspace, generates shapes highlighting the main atlas features and we can generate a range of shapes in the 'allowable shape domain'. Working in the dimension k is much easier since $k \ll n \lll N$ and through this way we have managed to compress our atlas to just the main k significant features.

2.2.2 Selection of feature vector α_k

From eq. (10), we can generate new shapes by varying the feature vector within reasonable limits so that the new shapes are similar to those in the atlas. As explained by [6], the parameters maybe linearly independent though there may be non-linear dependencies still present. To determine the limits of α_k, we look at the distribution and number of shapes used to generate the atlas.

We can base the limits on **empirical rule** also known as **68-95-99.7** rule [12] given that we assumed the domain is ellipsoidal and centered the shapes around the center of the ellipsoidal allowable domain. In this case we assume the atlas shapes follow a normal distribution. Using the **three-sigma rule of thumb** [12], it expresses that nearly all values lies within 3 standard deviations of the mean and it is useful to treat 99.7% probability as near certainty. These relation can be seen from the following Table 1

Range	**% explained**
$\mu \pm \sigma$	68.268949
$\mu \pm 2\sigma$	95.449973
$\mu \pm 3\sigma$	99.730020
$\mu \pm 4\sigma$	99.993666

Table 1: For various values of standard deviation, σ, the percentage of values that lie within the mean, μ

Since the variance of α_k over the atlas can be shown approximated by the eigenvalues, λ_k, suitable limits using the empirical rule will be

$$-3\sqrt{\lambda_k} \leq \alpha_k \leq 3\sqrt{\lambda_k}. \tag{11}$$

Unlike empirical rule, which applies to normal distributed data, we can consider the weaker **Chebyshev's inequality**[2, 16]. Let X be a random variable with known $\mu > 0$ and $\sigma > 0$. Then for any real number $k > 0$

$$\Pr\left(|X - \mu| \geq k\sigma\right) \leq \frac{1}{k^2}.$$

Only the case $k > 1$ is useful. It states that distribution of values lies within $\mu \pm k\sigma \leq \frac{1}{k^2}$ (or at least $1 - \frac{1}{k^2}$). It has the advantage of being applied to any distribution given that the mean and variance are defined. See Table 2

Range	**% explained**
$\mu \pm \sigma$	0
$\mu \pm 3\sigma$	88.8889
$\mu \pm 5\sigma$	96
$\mu \pm 9\sigma$	98.7654

Table 2: For various values of standard deviation, σ, the percentage of values that lie within the mean, μ *if we additionally know that the distribution is normal.*

The **Chebyshev's inequality**[2, 16] despite the advantage of being applicable to any arbitrary distribution, it is still dependent on the sample size n. It generally improves as the sample size increases. See Table 3

n	**Range**	**% Explained**
10	$\mu \pm 13.58\sigma$	95
100	$\mu \pm 4.96\sigma$	95
	$\mu \pm 140\sigma$	99
500	$\mu \pm 4.55\sigma$	95
	$\mu \pm 11.16\sigma$	99
1000	$\mu \pm 4.51\sigma$	95
	$\mu \pm 10.53\sigma$	99

Table 3

Alternatively, taking in to account the correlation of the shapes in the atlas we can consider **Mahalanobis distance**[3]. It measures the distance between a point p and distribution D. The distance is zero if the p is at the mean of D and grows as the p moves from D along the each principal component axis. Together with the k principal components, we can choose a set of parameters α_t, $t \in \{1, k\}$, such that Mahalanobis distance, D_M from the μ is less than a suitable set max, $D_{\max}$ [6]

$$D_M^2 = \sum_{t=1}^{k} \left(\frac{\alpha_t^2}{\lambda_t} \right) \leq D_{\max}^2. \tag{12}$$

With this approach eq. (12), if shapes are distributed normally then D_M follows a chi-squared distribution and $D_{\max}$, see [6] can be set to include a suitably large proportions of realizable shapes.

3 3D segmentation model

The described general segmentation model is similar to [22]. Let $u(\mathbf{x}, t)$ be level-set function, where $\mathbf{x}(t) = [x(t), y(t), z(t)]$ denotes position vector of a surface Γ, at time t. We observe evolution of $u(\mathbf{x}, t)$ such that at each time t the surface Γ represents the same isosurface of $u(\mathbf{x}, t)$ as follows

$$u(\mathbf{x}, t) = c \tag{13}$$

where c is a constant. The total differential of the level-set function becomes

$$\frac{du(\mathbf{x}(t), t)}{dt} = \frac{du(x(t), y(t), z(t), t)}{dt} = 0, \tag{14}$$

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \frac{\partial u(\mathbf{x}, t)}{\partial x}\frac{\partial x(t)}{\partial t} + \frac{\partial u(\mathbf{x}, t)}{\partial y}\frac{\partial y(t)}{\partial t} + \frac{\partial u(\mathbf{x}, t)}{\partial z}\frac{\partial z(t)}{\partial t} = 0, \tag{15}$$

which can be rewritten in the form

$$\begin{aligned} &\frac{\partial u(\mathbf{x}, t)}{\partial t} + \frac{d\mathbf{x}(t)}{dt} \cdot \nabla u(\mathbf{x}, t) = 0, \\ &u(\mathbf{x}, 0) = u^0(\mathbf{x}), \quad \text{for} \quad x \in D, \end{aligned} \tag{16}$$

where $u^0(\mathbf{x})$ is an initial condition. Let us denote $\mathbf{V}(\mathbf{x}, t) = \frac{d\mathbf{x}(t)}{dt}$, as the force term. Then the Eq. (16) becomes

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \mathbf{V}(\mathbf{x}, t) \cdot \nabla u(\mathbf{x}, t) = 0. \tag{17}$$

The force term controls the evolution of the level-set funtion. It is composed of the external force term and and the curvature term. We have defined it similar to [22] as follows

$$\mathbf{V}(\mathbf{x},t) = \mu_1((1-\lambda)g_2\mathbf{N} - \lambda\nabla g_1) + \mu_2 g_1 \kappa \mathbf{N}, \tag{18}$$

where μ_1, μ_2 and λ are parameters, $\mathbf{N} = \dfrac{\nabla u}{|\nabla u|}$ is the outer unit normal vector, $\kappa = -\nabla \cdot \dfrac{\nabla u}{|\nabla u|}$ is curvature, $g_1 = g_1(|\nabla G_\sigma \star I|) = \dfrac{1}{1 + k_1|\nabla G_\sigma \star I|^2}$ and g_2 defined as

$$g_2 = g_2(I, \rho, k_2) = \frac{1}{1 + k_2(I-\rho)^2}. \tag{19}$$

Both g_1 and g_2 are scaling functions, where g_1 is an edge detector function. g_2 is an inflation term function, with $k_2 > 0$ a parameter, ρ is the average value computed from voxels inside the initial segmentation surface and I is voxel value. Both ρ and I can be a reasonably chosen value, characterizing color, intensity or texture of segmented object in the image. **This is useful because different organs can have different colors, textures or intensity values**.

3.1 Discretization of the segmentation model

From eq. (17), we expand it to full form

$$u_t + \left(\mu_1((1-\lambda)g_2\frac{\nabla u}{|\nabla u|} - \lambda\nabla g_1)\right) \cdot \nabla u - \mu_2 g_1 \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right)\frac{\nabla u}{|\nabla u|} \cdot \nabla u = 0. \tag{20}$$

Simplified further to

$$u_t + \mu_1 \nu \cdot \nabla u - \mu_2 g_1 |\nabla u| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) = 0, \tag{21}$$

where $\nu = (1-\lambda)g_2\dfrac{\nabla u}{|\nabla u|} - \lambda\nabla g_1$.

Time discretization

In order to discretize eq. (21) in time, we apply semi-implicit approach that guarantees unconditional stability with respect to the diffusion term. Let suppose that the equation in time intervcal $I = [0, T]$ and in N equal time steps. The time step is denoted as $\tau = \frac{T}{N}$. The time discretization is then as follows

$$\frac{u^{n+1} - u^n}{\tau} + \mu_1 \nu^n \cdot \nabla u^n - \mu_2 g_1 |\nabla u^n| \nabla \cdot \left(\frac{\nabla u^{n+1}}{|\nabla u^n|}\right) = 0 \tag{22}$$

Space discretization

To discretize eq. (22) in space, we apply the so called flux-based level set finite volume [14] method. Consider the product rule as follows

$$\nabla \cdot (u^n \nu^n) = u^n \nabla \cdot \nu^n + \nu^n \nabla \cdot u^n,$$

we can express $\nu^n \cdot \nabla u^n = \nabla \cdot (u^n \nu^n) - u^n \nabla \cdot \nu^n$ and replace in eq. (22). We integrate eq. (22) over a finite volume p.

$$\begin{aligned} 0 &= \int_p \frac{u^{n+1} - u^n}{\tau} dx \\ &- \mu_1 \int_p u^n \nabla \cdot \nu^n dx \\ &- \mu_2 \int_p g_1 |\nabla u^n| \nabla \cdot \left(\frac{\nabla u^{n+1}}{|\nabla u^n|} \right) dx \end{aligned} \tag{23}$$

Integrating

$$\int_p \frac{u^{n+1} - u^n}{\tau} dx$$

becomes

$$\int_p \frac{u^{n+1} - u^n}{\tau} dx = m(p) \frac{u_p^{n+1} - u_p^n}{\tau}, \tag{24}$$

where $m(p)$ is the measure of finite volume p.

For the advection part,

$$\mu_1 \int_p \nabla \cdot (u^n \nu^n)\, dx - \mu_1 \int_p u^n \nabla \cdot \nu^n dx,$$

we use divergence theorem in both terms and constant representation of solution in finite volume p in the second term, we obtain

$$\approx \mu_1 \int_{\partial p} (u^n \nu^n) \cdot \mathbf{n}_{\partial p} dS - \mu_1 u_p^n \int_{\partial p} \nu^n \cdot \mathbf{n}_{\partial p} dS. \tag{25}$$

Using integral fluxes, see [14, 18, 20], we define inflows and outflows through the voxel sides as

$$\nu_{pq}^{in} = \min\left(\nu_{pq}^n, 0\right) , \nu_{pq}^{out} = \max\left(\nu_{pq}^n, 0\right).$$

where ν_{pq}^n is define as

$$\begin{aligned} \nu_{pq}^n &= \int_{\partial p} \nu^n \cdot \mathbf{n}_{\partial p} dS \\ &= \int_{e_{pq}} (1-\lambda) g_2 \frac{\nabla u^n}{|\nabla u^n|} - \lambda \nabla g_1 \cdot \mathbf{n}_{\partial p} dS \\ &\approx \frac{m(e_{pq})}{m(\sigma_{pq})} \left[(1-\lambda) g_2 \frac{u_q^n - u_p^n}{|\nabla\, u_{pq}^n|} - \lambda \nabla g_1 \right], \end{aligned} \tag{26}$$

where e_{pq} denotes edge between finite volumes p and q, $\mathbf{n}_{\partial p}$ is the normal vector to e_{pq} from p to q, $m(e_{pq})$ is measure of edge between finite volumes p and q and $m(\sigma_{pq})$ denotes measure of line between centers of finite volumes p and q.

We define an approximate gradient, ∇g_1 in finite volume p using central differences [20]:

$$\nabla g_{1_p} = (G_{pe}, G_{pn}, G_{pt}) = (-G_{pw}, -G_{ps}, -G_{pb})$$

where

$$\begin{aligned} -G_{pw} &= G_{pe} \approx \frac{g_e - g_w}{2h}, \\ -G_{ps} &= G_{pn} \approx \frac{g_n - g_s}{2h}, \\ -G_{pb} &= G_{pt} \approx \frac{g_t - g_b}{2h}, \end{aligned}$$

and g_q is the value of the g_1 in $q \in N_p$. For g_2 function, we approximate the average of neighbouring points as follows:

$$g_2 = \frac{g_{2_p} + g_{2_q}}{2}.$$

Substituting these in eq. (26), we get ν_{pq}^n define as

$$\nu_{pq}^n = \frac{m(e_{pq})}{m(\sigma_{pq})} \left[(1-\lambda) \frac{g_{2_p} + g_{2_q}}{2} \frac{u_q^n - u_p^n}{|\nabla u_{pq}^n|} - \lambda G_{pq} \right]. \tag{27}$$

We can then approximate eq. (25) by using upwind principle as in [14, 20] to obtain the following

$$\begin{aligned} \mu_1 \int_{\partial p} (u^n \nu) \cdot \mathbf{n}_{\partial p} dS - \mu_1 u_p^n \int_{\partial p} \nu \cdot \mathbf{n}_{\partial p} dS \approx \\ \mu_1 \sum_{q \in N_p} \nu_{pq}^{in} u_q^n + \mu_1 \sum_{q \in N_p} \nu_{pq}^{out} u_p^n - \mu_1 \sum_{q \in N_p} \nu_{pq}^{in} u_p^n - \mu_1 \sum_{q \in N_p} \nu_{pq}^{out} u_p^n = \\ \mu_1 \sum_{q \in N_p} \nu_{pq}^{in} \left(u_q^n - u_p^n \right), \end{aligned} \tag{28}$$

where N_p is an index set containing all the neighboring finite volumes q to p.

For discretization of the diffusion term, we consider $|\nabla u_n|$ as a constant on finite volume p, to become $|\nabla u_p^n|$. Then apply divergence theorem, approximating the derivative, $\nabla u^{n+1} \approx \frac{u_q^{n+1} - u_p^{n+1}}{m(\sigma_{pq})}$ and $|\nabla u^n|$ on the edges, denoted as $|\nabla u_{pq}^n|$ using diamond cell scheme [17, 18]. This is similar to approximating the mean curvature term as done in [19]

$$\mu_2 \int_p g_1 |\nabla u^n| \nabla \cdot \left(\frac{\nabla u^{n+1}}{|\nabla u^n|} \right) dx \approx \mu_2 g_1 |\nabla u_p^n| \sum_{q \in N_p} \frac{m(e_{pq})}{m(\sigma_{pq})} \frac{u_q^{n+1} - u_p^{n+1}}{|\nabla u_{pq}^n|}. \tag{29}$$

Combining eq. (24), (28), (29), we can obtain discrete form of eq. (23) and making u_p^n the subject as:

$$\begin{aligned} u_p^n \quad = \quad & u_p^{n+1} + \mu_1 \frac{\tau}{m(p)} \sum_{q \in N_p} \nu_{pq}^{in} \left(u_q^n - u_p^n \right) - \\ & \mu_2 \frac{\tau}{m(p)} g_1 |\nabla u_p^n| \sum_{q \in N_p} \frac{m(e_{pq})}{m(\sigma_{pq})} \frac{u_q^{n+1} - u_p^{n+1}}{|\nabla u_{pq}^n|} \end{aligned} \tag{30}$$

The solution of eq. (30) is then obtained by using Gaus-Seidel iterative method. For boundary values, we impose Neumann boundary condition.

3.2 Atlas involvement in the segmentation model

In section 3.1, we presented discretization of the segmentation model eq. (21). To involve atlas information in the model, we follow the same approach as [13, 22]. We modify the force term since it controls the evolution of the level-set function. First of all, the force term 18 can be broken into influence terms as shown in Figure 2

We involve the atlas info by modifying the external force term to include the difference between the current (u) and estimated ($\tilde{u}$) segmentation in the normal direction. The estimated segmentation $\tilde{u}$ is calculated using the method described in Section 3.3. The extended external force will be as follows

$$\mu_1(\gamma((1-\lambda)g_2\mathbf{N} - \lambda\nabla g_1) + (1-\gamma)(u-\tilde{u})\mathbf{N}), \tag{31}$$

where γ is parameter that weighs the influence of the edges and the expanding term compared to influence of the atlas. The modified force term with alas involvement is shown in Figure 3. We have chosen γ similar to [13,

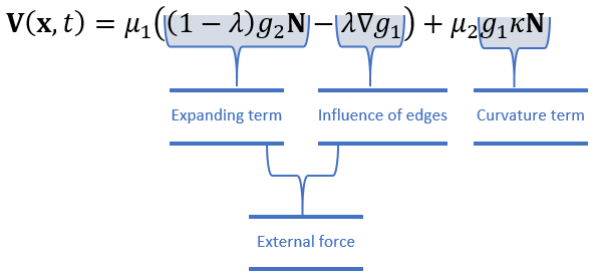


Figure 2: The composition of the force terms in the segmentation model

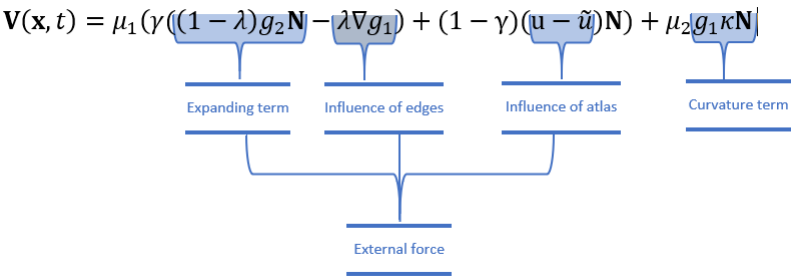


Figure 3: Composition of the force terms with atlas involvement in the segmentation model

22] (eq. (32)). We follow the same approach as [13] and threshold distance d_1 is approximately 4 or 5 times d_2. See Fig. 4 illustrates selection of γ parameter.

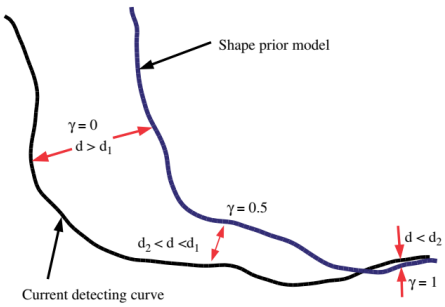


Figure 4: Illustration of γ selection

$$\gamma = \begin{cases} 0, & |u-\tilde{u}| > d_1 \\ 1, & |u-\tilde{u}| < d_2 \\ 0.5, & \text{else.} \end{cases} \tag{32}$$

where d_1 , d_2 are distance such that $d_1 > d_2$.

After modification of the external force with involvement of atlas, the discrete ν_{pq}^n in eq. (27) becomes

$$\begin{aligned} \nu_{pq}^n \quad = \quad & \frac{m(e_{pq})}{m(\sigma_{pq})}\left(\gamma\left((1-\lambda)\frac{g_{2_p}+g_{2_q}}{2}\frac{u_q^n-u_p^n}{|\nabla u_{pq}^n|} - \lambda G_{pq}\right)\right) + \\ & \frac{m(e_{pq})}{m(\sigma_{pq})}\left((1-\gamma)(u-\tilde{u})\frac{u_q^n-u_p^n}{|\nabla u_{pq}^n|}\right) \end{aligned} \tag{33}$$

where $(u - \tilde{u})$ is defined as an average of differences between current and estimated segmentation.

$$(u - \tilde{u}) = \frac{(u_p - \tilde{u}_p) + (u_q - \tilde{u}_q)}{2}$$

3.3 PCA-based estimation models

We propose two simple approaches of estimating shape from the eigenspace domain. The first approach is based on the difference (energy) between the current segmentation and estimated segmentation shape, see [22]. The second approach is based on maximizing the likelihood probability function by evolving the current segmentation towards the estimated shape, see [13]. In each approach, the limits of the feature vector α_k are set as described in Section 2.2.2.

Let u be the current segmentation step result and $\tilde{u}$ be estimate of the current segmentation from the atlas. The estimate $\tilde{u}$ is obtained by first registering the current segmentation, u, with the mean shape $\bar{\phi}$. The registered results is then estimated (eigenshape formulation) as described in Section 2.2.1. In this case u corresponds to y and $\bar{\phi}$ to $\bar{s}$ of eq. (9) and 10 respectively. After estimating, inverse transformation is applied to return it to the initial space of the current segmentation.

Energy based estimation model

The goal is to minimize the following functional, defined as E_{shape}

$$E_{shape} = \int_D (\tilde{u} - u)^2 \, dA, \tag{34}$$

which can be written in the form

$$E_{shape} = \int_D \left(U_k \alpha + \bar{\phi} - u\right)^2 dA, \tag{35}$$

where U_k is a matrix consisting of the first k eigenvector columns, $\bar{\phi}$ is the mean shape of the eigneshapes and α vector represents the k eigenshape weights.

The functional is minimized by varying α. The gradient of E_{shape} w.r.t α is defined as follows

$$\nabla_\alpha E_{shape} \propto \frac{2}{N} U_k^T \left(U_k \alpha + \bar{\phi} - u\right). \tag{36}$$

The optimal coefficient vector α is obtained through an iterative process along the gradient descent direction

$$\alpha_{t+1} = \alpha_t - \epsilon \nabla_\alpha E_{shape}, \tag{37}$$

where $\epsilon > 0$ is equivalent to the step length. The resulting vector α is used to estimate shape similar to the current segmentation from the atlas as follows

$$\tilde{u} = U_k \alpha + \bar{\phi},$$

which is derived from eq. (10).

Probability based estimation model

The model is based on the methods described in [8, 9, 13]. It expresses the probability of the feature vector α. In the reduced subspace, the feature vectors α is assumed to follow a Gaussian distribution

$$p(\alpha) = \frac{1}{(2\pi)^{\frac{k}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\{-\frac{1}{2} \alpha^T \Sigma_k^{-1} \alpha\}, \tag{38}$$

where Σ is diagonal matrix whose diagonal values are the first k eigenvalues. For the current segmentation step result u, the probability density function in the subspace is $p(\alpha)$. To maximize the probability, the geodesic active contour is evolved towards the shape pattern. For computational convenience, the negative logarithm of $p(\alpha)$ is computed and defined as an energy term,$E_{shape}(\alpha)$:

$$\begin{aligned} E_{shape}(\alpha) &= -\log\left[p(\alpha)\right] \\ &= \log\left[(2\pi)^{\frac{k}{2}}|\Sigma_k|^{\frac{1}{2}}\right] + \left[\frac{1}{2}\alpha^T\Sigma_k^{-1}\alpha\right] \\ &\propto \alpha^T\Sigma_k^{-1}\alpha. \end{aligned} \tag{39}$$

Minimizing $E_{shape}(\alpha)$ is equivalent to maximizing the probability function. The minimum is searched along the gradient descent direction. The initial α_0 is set as described in eq. (9) i.e.

$$\alpha_0 = \mathbf{U}_k^T\left(u - \bar{\phi}\right).$$

Thereafter, we search for the minimum:

$$\begin{aligned} \alpha_{t+1} &= \alpha_t - \epsilon\nabla_\alpha E_{shape}(\alpha_t) \\ &= \alpha_t - \epsilon\Sigma_k^{-1}\alpha_t, \end{aligned} \tag{40}$$

where ϵ is equivalent to the step length. The resulting vector α is used to estimate shape similar to the current segmentation from the atlas, similar to the previous section or eq. (10).

Similar shape estimation results

In our tests, we have compared results when estimating a shape included in the atlas construction and estimation of shape not included in the atlas construction. Let a represent an atlas constructed using stochastic gradient descent method (see our publication [1]) and b an atlas constructed using block coordinated descent method, (see Section 2) respectively. Registration of the shape to be estimated and atlas mean was performed using the block coordinated descent method. To measure the similar shape estimation, we used two accuracy measures between the estimated shape and the exact shape. These measures are *Mean Hausdorff distance* (mhd with voxel spacing of $h = 1.0$) and L2-norm difference.

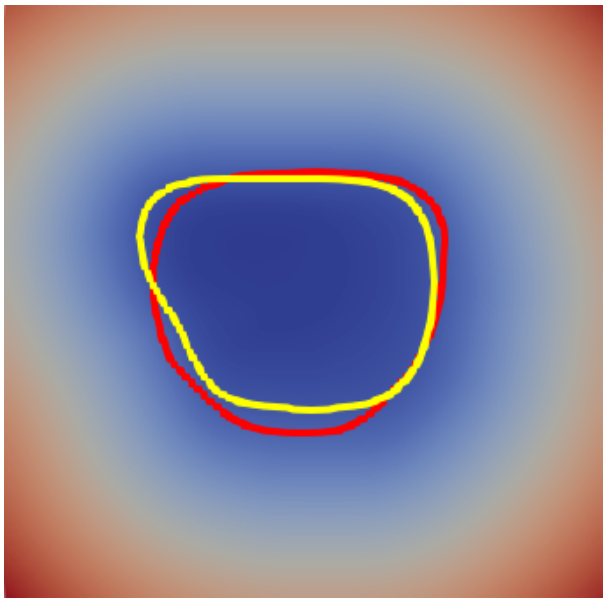
		Estimation model		
Atlas	Measure	Energy	Probability	Registration
a	L^2-norm	0.2836	0.40512	0.1484
	mhd	0.4961	0.7158	
b	L^2-norm	0.4144	0.3963	0.1885
	mhd	0.7778	0.7953	

Table 4: Comparison of estimation models when estimating a shape included in the atlases

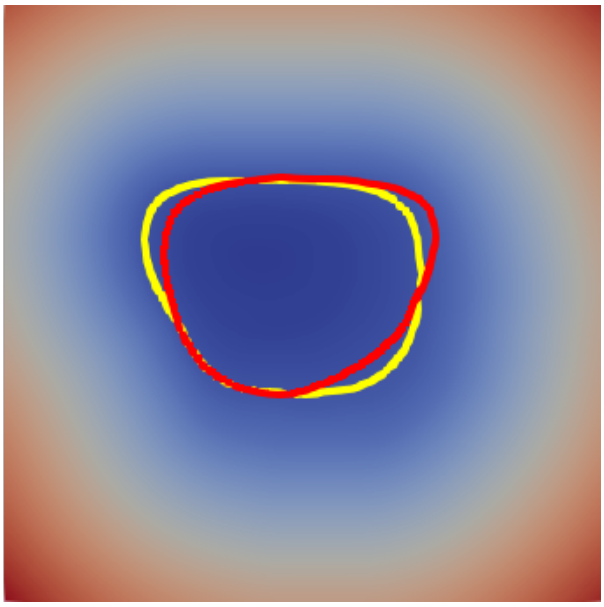
The goal was to show either of the proposed estimation models can be used during estimation of shapes from the atlas and successful estimation of shapes not included in the atlas (unknown). The results are summarized in Table 4 and 5. Visual results are shown Figure 5 for estimated shapes not in the atlas. These results are for **segmentation without involvement of the atlas**.

		Estimation model		
Atlas	Measure	Energy	Probability	Registration
a	L^2-norm	1.5213	1.5279	1.3020
	mhd	1.0927	0.9092	
b	L^2-norm	1.7315	1.3990	1.0284
	mhd	1.1984	1.2142	

Table 5: Comparison of estimation models when estimating a shape not included in the atlases



Energy based model estimation.



Probability based model estimation.

Figure 5: Comparison of estimation models for a shape not in atlas. Slice view of the distant function for exact shape. Red curve is estimated while yellow curve is the exact shape.

3.4 Improvement and handling of the inflation function g_2

The goal is to find the specific iteration to reduce or turn off g_2. It is difficult to determine the extrema of the curve of the differences between current, u_p^{n+1} and previous, u_p^n segmentation while the segmentation is still running. To solve this, we built a function estimating the next ϖ differences from the currently calculated differences. We refer to ϖ as the offset value. The estimation is started only after the number of iterations during the segmentation process is greater than ϖ.

Let y be a discrete curve of values calculated from L^2-norm of the difference between u_p^{n+1} and u_p^n at each iteration (time step) n. It is defined as follows

$$y^{n+1} = \frac{1}{N} \sum_{p=1}^{N} \left(u_p^{n+1} - u_p^n \right)^2 \tag{41}$$

where N is the total number of points in their level set representaion. The y curve values are noisy. To remove this noise, we perform *curve fitting* by constructing a smoothing function to approximately fit the data. Let y_{smooth} be the curve fitted by applying moving average of order m as follows

$$y_{\text{smooth}}^n = \frac{1}{m} \sum_{i=-k}^{k} y^{(n+i)}, \tag{42}$$

where $m = 2k + 1$.

The smoothed curve will be used as an input to the estimating function instead of the original noisy curve. In addition, the smooth curve will be used to detect the curve's extrema. We also define the function, y_{diff} as follows

$$y^n_{\text{diff}} = y^{n+1}_{\text{smooth}} - y^n_{\text{smooth}} \tag{43}$$

With the above definitions, we now define the function to estimate, denoted as y_{est}.

$$\begin{aligned} d^n_{\text{diff\_smooth}} &= \frac{1}{\xi}\sum_{i=-k}^{k} y^{n+i}_{\text{diff}} \\ y^n_{\text{est}} &= y^n_{\text{smooth}} + \varpi d^n_{\text{diff\_smooth}}, \end{aligned} \tag{44}$$

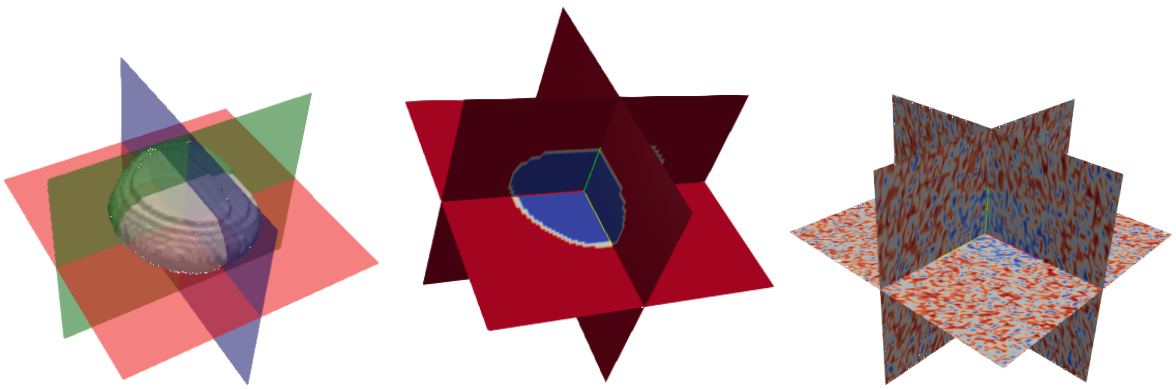
We have set $\xi = 2k + 1$ and ϖ is the offset. For our setup, we used $k = 2$ or $\xi = 5$. With the above setup, we find the first iteration n to turn off g_2, such that y^n_{est} becomes non-positive. The estimate corresponds to the global minima of the estimated curve. The advantage of this approach is that we don't have to wait for the segmentation process to finish then determine the global minima from the calculated values.

After determining what iteration to turn off g_2, we try to determine the iteration to automatically reduce its influence. From our early experiments, we observed our initial choice was approximately close to the curve's global maxima. From our experience, the global maxima is not the best measure for choosing the iteration to reduce the influence. This is because in some experiments, this can be in the first early steps, which we found too early to reduce or turn off g_2 influence. Instead, we have chosen to base the iteration to reduce the influence on the mid-area of the curve y.

The $n+\varpi$ estimated values are copied from y_{est} to y. Then use *Trapezoid rule* [5] to calculate the area under the new curve. Then we calculate the mid-area and find the closest iteration to this mid-area. At this iteration, we set to reduce g_2 influence. In the next subsection, we show results from applying the novel automatic approach to handle g_2 and compare it to the first two attempts we tried on the 85% noisy images.

Results of automatic selections of reduce and turn off steps for g_2 function (no atlas)

In the experimental setup, we tested on noisy synthetic bladder images. The exact dimensions for the each bladder image was $100 \times 100 \times 40$ voxels. We created artificial noisy bladder images by adding varying high level of noise to the exact images. We chose salt and pepper where 85% of the images was affected with the noise. From our experiments with different types of noise and associating levels, *salt and pepper* noise levels >= 75% segmentation failed. Figure 6 shows the one of the test data and the resulting noisy image with 85% noise levels slice view. In our setup, we used a set of 3 bladders.



(a) Exact bladder image (b) Slice view of exact bladder (c) Slice view of 85% noisy bladder

Figure 6: First bladder image

In table 6, we show the found values versus our handpicked values in our original attempt for the 85% noisy images.

Bladder	1	2	3
reduce iteration	15	15	15
turn off iteration	35	35	35

(a) Manual selected iterations to change g_2 influence.

Bladder	1	2	3
reduce iteration	15	15	14
turn off iteration	37	38	32

(b) Automatically calculated iterations to change g_2 influence.

Table 6: Comparison of manual versus automatic selection of steps to reduce and turn off g_2 influence

In Table 7 we show the original segmentation results without applying automatic handling of g_2 approach.

Bladder	1	2	3
L^2-norm	36.7712	67.3418	45.0444

Table 7: L^2-norm after segmenting 85% noisy images

In Table 8, we show new results of segmentation with the automatic handling of g_2. In the setup, after the first estimated zero has been found and mid-area determined, the segmentation process is restarted automatically from the initial iteration. Once the segmentation process reaches the found iterations, automatic g_2 reduction and turning off is applied without user intervention.

Bladder	1	2	3
L^2-norm	1.1380	6.2533	0.2299

(a) Segmenting 85% noisy images with both steps set manually

Bladder	1	2	3
L^2-norm	1.2245	5.2651	0.1906

(b) Segmenting 85% noisy images with both steps set automatically

Table 8: Comparison or segmentation L^2-norm results between manual and automatic handling of g_2.

The segmentation results shown in Tables 7 and 8 do not use the atlas based model. The goal was to first improve the segmentation model, then after apply the atlas-based segmentation model. In the next sections, we show results after applying the atlas-based segmentation model to real data.

4 Atlas-based 3D segmentation results

This final section, presents segmentation results after applying the atlas-based segmentation model to real data. The real data are three patient pelvis CT images with bladder as the region of interest. Three cropped volumes of dimensions $100 \times 100 \times 40$ voxels are extracted from the CT images. The cropped volume contains the region of interest. The goal is to segment the real bladder shapes from these patient CT images.

The iterations to reduce or turn off g_2 influence calculated are presented in Table 9. The presented results, see

Patient Image	1	2	3
reduce iteration	31	30	34
turn off iteration	68	64	60

Table 9: Automatically set iterations to change g_2 influence.

Figures 7, 8 and 9 show the atlas involvement significantly improves the results. In general, segmentation deforms similarly to the class of objects it is trying to segment. In some of the results presented, the segmentation was able to fill in the missing boundary information for the bladder extracted using model. The threshold distances for γ were set to $d_1 =\sim 20$, $d_2 =\sim 4$ to 5 respectively.

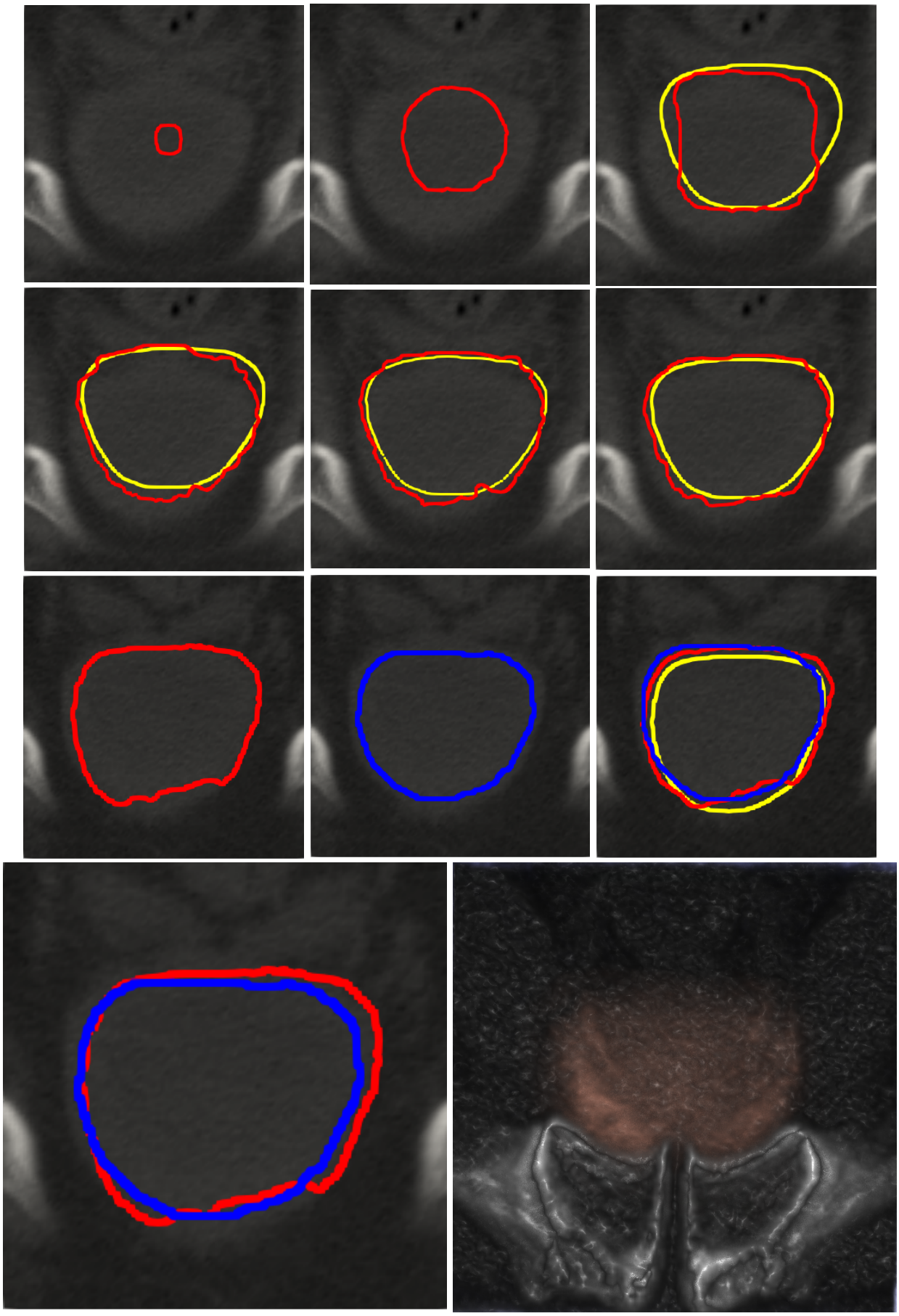


Figure 7: Patient 1 bladder segmentation. Red curve is the segmented, yellow curve is the estimated shape and blue is the exact shape. Final segmented bladder (red) is shown in the cropped pelvis image.

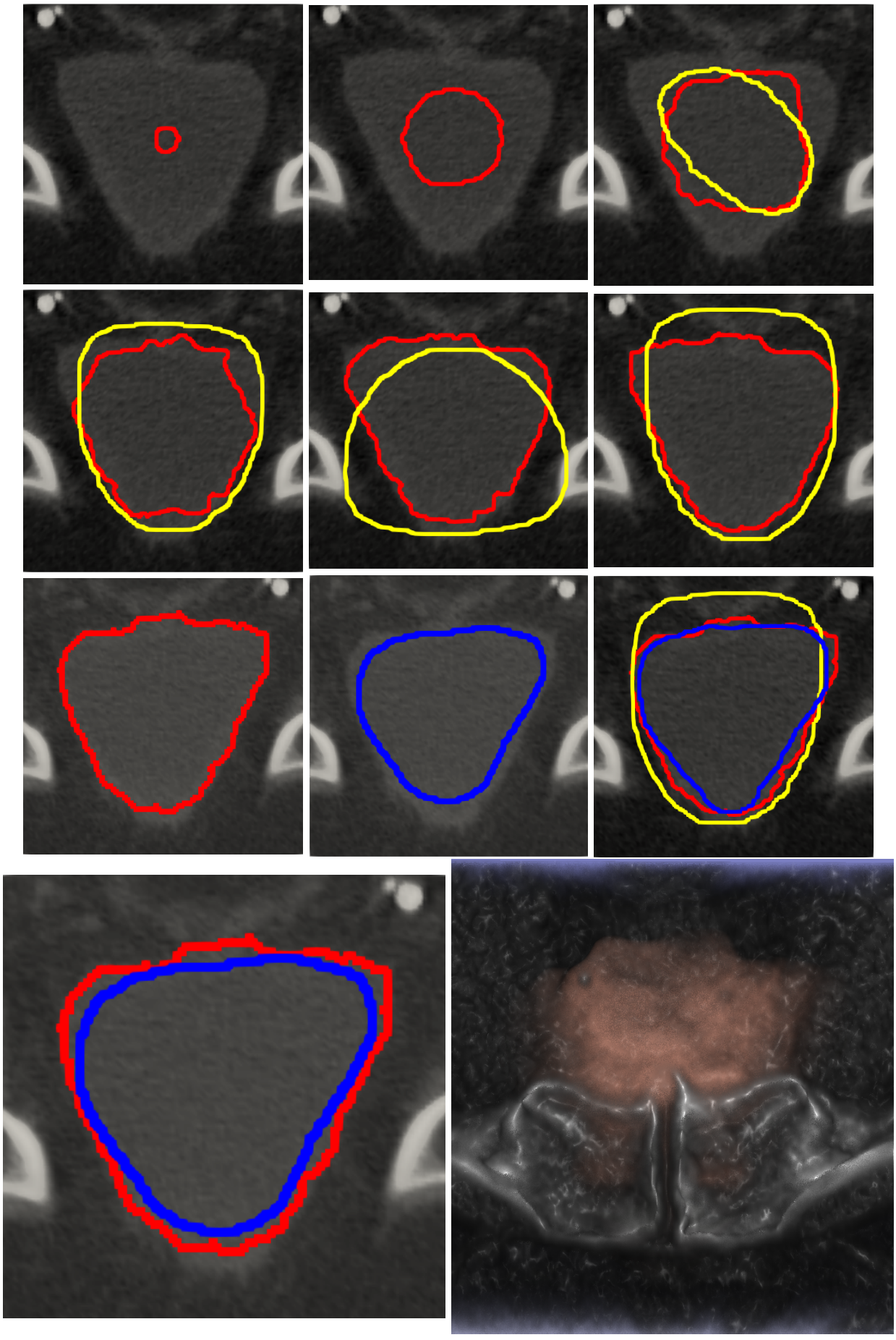


Figure 8: Patient 2 bladder segmentation. Red curve is the segmented, yellow curve is the estimated shape and blue is the exact shape. Final segmented bladder (red) is shown in the cropped pelvis image.

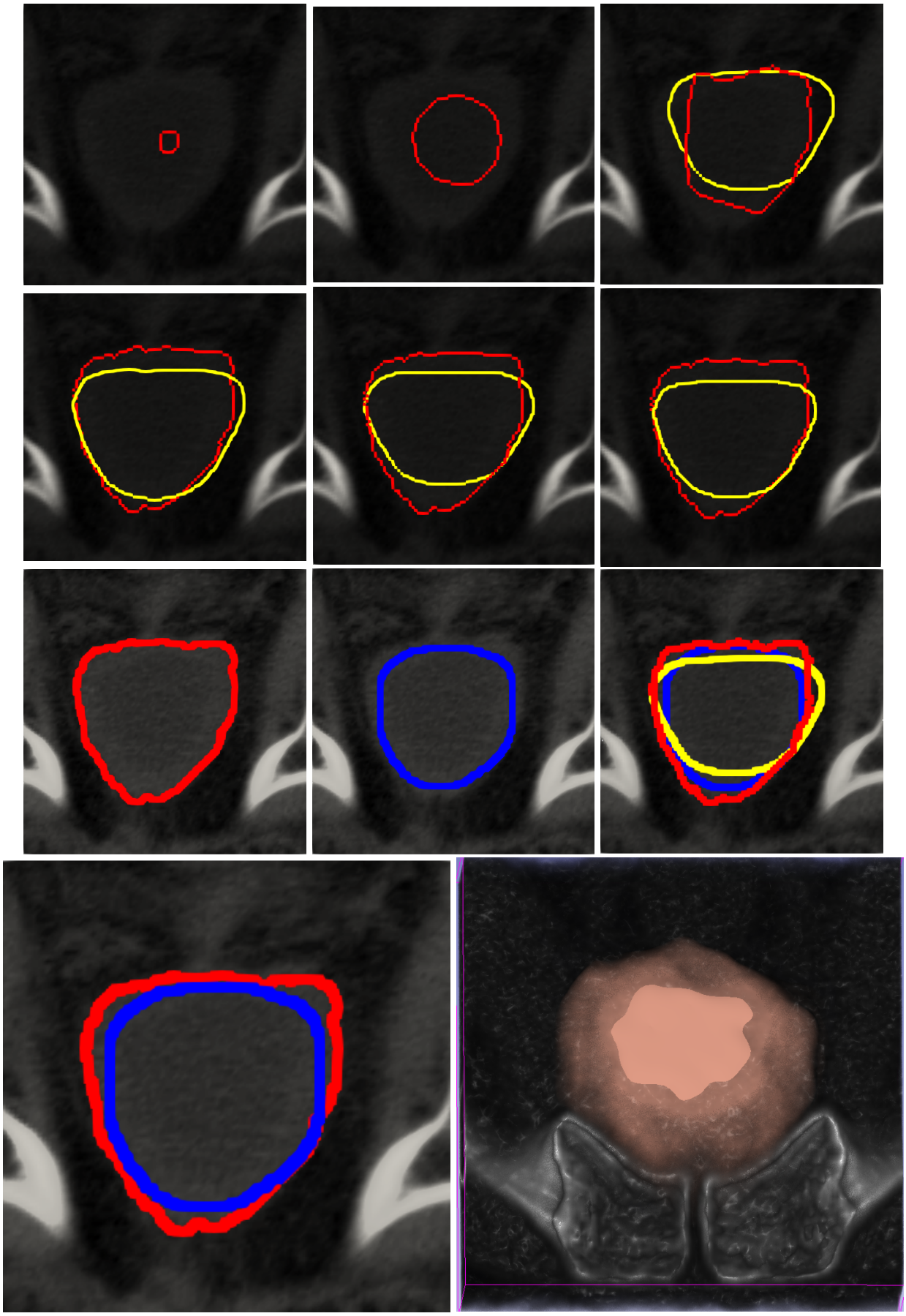


Figure 9: Patient 3 bladder segmentation. Red curve is the segmented, yellow curve is the estimated shape and blue is the exact shape. Final segmented bladder (red) is shown in the cropped pelvis image.

References

[1] Polycarp Omondi Okock, Jozef Urbán, and Karol Mikula. "Efficient 3D Shape Registration by Using Distance Maps and Stochastic Gradient Descent Method". In: *Tatra Mountains Mathematical Publications* 75.1 (1Apr. 2020), pp. 81 –102. DOI: https://doi.org/10.2478/tmmp-2020-0006. URL: https://content.sciendo.com/view/journals/tmmp/75/1/article-p81.xml.

[2] P. Tchébychef. "Des valeurs moyennes (Traduction du russe, N. de Khanikof." fre. In: *Journal de Mathématiques Pures et Appliquées* (1867), pp. 177–184. URL: http://eudml.org/doc/234989.

[3] P. C. Mahalanobis. "On the Generalized Distance in Statistics". In: *Proceedings of National Institute of Sciences (India)* 2.1 (1936), pp. 49–55.

[4] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *Ann. Math. Statist.* 22.3 (Sept. 1951), pp. 400–407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

[5] Kendall Atkinson. *An Introduction to Numerical Analysis*. 2nd ed. John Wiley & Sons, Jan. 1989. ISBN: 978-0-471-62489-9.

[6] T.F. Cootes et al. "Active Shape Models-Their Training and Application". In: *Computer Vision and Image Understanding* 61.1 (1995), pp. 38 –59. ISSN: 1077-3142. DOI: https://doi.org/10.1006/cviu.1995.1004. URL: http://www.sciencedirect.com/science/article/pii/S1077314285710041.

[7] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1999. ISBN: 9780521645577. URL: https://books.google.sk/books?id=ErpOoynE4dIC.

[8] Michael Emmanuel Leventon. "Statistical Models in Medical Image Analysis". PhD thesis. Massachusetts Institute of Technology 2, Apr. 2000.

[9] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. "Statistical shape influence in geodesic active contours". In: *5th IEEE EMBS International Summer School on Biomedical Imaging, 2002.* 2002, 8 pp.–. DOI: 10.1109/SSBI.2002.1233989.

[10] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. "Matching Distance Functions: A Shape-to-Area Variational Approach for Global-to-Local Registration". In: *Computer Vision — ECCV 2002*. Ed. by Anders Heyden et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 775–789. ISBN: 978-3-540-47967-3.

[11] Hongkai Zhao. "Fast sweeping method for Eikonal equations". In: *Math. Comput.* 74 (Apr. 2005), pp. 603–627. DOI: 10.1090/S0025-5718-04-01678-3.

[12] E.W. Grafarend. *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models*. Walter de Gruyter, 2006. ISBN: 9783110162165. URL: https://books.google.sk/books?id=0Q7st5Cb41oC.

[13] Wen Fang and Kap Luk Chan. "Incorporating shape prior into geodesic active contours for detecting partially occluded object". In: *Pattern Recognition* 40.8 (2007). Part Special Issue on Visual Information Processing, pp. 2163 –2172. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2006.12.014. URL: http://www.sciencedirect.com/science/article/pii/S0031320306005292.

[14] Peter Frolkovič and Karol Mikula. "Flux-based level set method: A finite volume method for evolving interfaces". In: *Applied Numerical Mathematics* 57.4 (2007), pp. 436 –454. ISSN: 0168-9274. DOI: https://doi.org/10.1016/j.apnum.2006.06.002. URL: http://www.sciencedirect.com/science/article/pii/S0168927406001310.

[15] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Applied Multivariate Statistical Analysis. Pearson Prentice Hall, 2007. ISBN: 9780131877153. URL: https://books.google.sk/books?id=gFWcQgAACAAJ.

[16] "Bienaymé, Irénée-Jules". In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 42–42. ISBN: 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_31. URL: https://doi.org/10.1007/978-0-387-32833-1_31.

[17] Olga Drblíková and Karol Mikula. "Semi-implicit diamond-cell finite volume scheme for 3D nonlinear tensor diffusion in coherence enhancing image filtering". In: *Finite Volumes for Complex Applications V: Problems and Perspectives, ISTE and WILEY, London* (2008), pp. 343–350.

[18] Karol Mikula and Mariana Remesikova. "Finite volume schemes for the generalized subjective surface equation in image segmentation". In: *Kybernetika* 4 (Jan. 2009).

[19] R. Eymard, A. Handlovičová, and K. Mikula. "Study of a finite volume scheme for the regularized mean curvature flow level set equation". In: *IMA Journal of Numerical Analysis* 31.3 (2011), pp. 813–846. ISSN: 1464-3642. DOI: 10.1093/imanum/drq025.

[20] Karol Mikula, Michal Smíšek, and Róbert Špir. "Parallel algorithms for segmentation of cellular structures in 2D+Time and 3D morphogenesis data". In: *Proceedings of the Conference Algoritmy* (2015), pp. 416–426. URL: http://www.iam.fmph.uniba.sk/amuc/ojs/index.php/algoritmy/article/view/352.

[21] Stephen J. Wright. "Coordinate descent algorithms". In: *Mathematical Programming* 151.1 (2015), pp. 3–34. ISSN: 1436-4646. DOI: 10.1007/s10107-015-0892-3. URL: https://doi.org/10.1007/s10107-015-0892-3.

[22] Jozef Urbán. "The new improvements of atlas based image segmentations". PhD thesis. Slovak University of Technology in Bratislava, July 2016.

Zoznam publikačnej činnosti

Okock, P., Urbán, J., and Mikula, K. (2020). Efficient 3D Shape Registration by Using Distance Maps and Stochastic Gradient Descent Method. Tatra Mountains Mathematical Publications 75, 1, 81-102, Available From: Sciendo https://doi.org/10.2478/tmmp-2020-0006.

Polycarp O. Okock, Urban Jozef, Markjoe O. Uba. (2019, February 15). ImageInLib v1.0.0 Release (Version v1.0.0). Zenodo. http://doi.org/10.5281/zenodo.2566361.