Hierarchical Markov Random Fields Applied to Model Soft Tissue Deformations on Graphics Hardware

Christof Seiler[1], Philippe Büchler[2], Lutz-Peter Nolte[3], Mauricio Reyes[4] and Rasmus Paulsen[5]

Abstract Many methodologies dealing with prediction or simulation of soft tissue deformations on medical image data require preprocessing of the data in order to produce a different shape representation that complies with standard methodologies, such as Mass-spring networks, Finite Element Methods (FEM), etc. On the other hand, methodologies working directly on the image space normally do not take into account mechanical behavior of tissues and tend to lack physics foundations driving soft tissue deformations. This paper presents a method to simulate soft tissue deformations based on coupled concepts from image analysis and mechanics theory. The proposed methodology is based on a robust stochastic approach that takes into account material properties retrieved directly from the image, concepts from continuum mechanics and FEM. The optimization framework is solved within a Hierarchical Markov-Random Field (HMRF) which is implemented on the Graphics Processor Unit (GPU).

Introduction

One of the current problems with non-rigid registration techniques is their lack of physics foundations concerning mechanical properties and energies driving the deformations. Conversely, classical methods to compute soft tissue deformations

[1] University of Bern, ARTORG Center, 3014 Bern, Switzerland
christof.seiler@artorg.unibe.ch

[2] University of Bern, ARTORG Center, 3014 Bern, Switzerland
philippe.buechler@artorg.unibe.ch

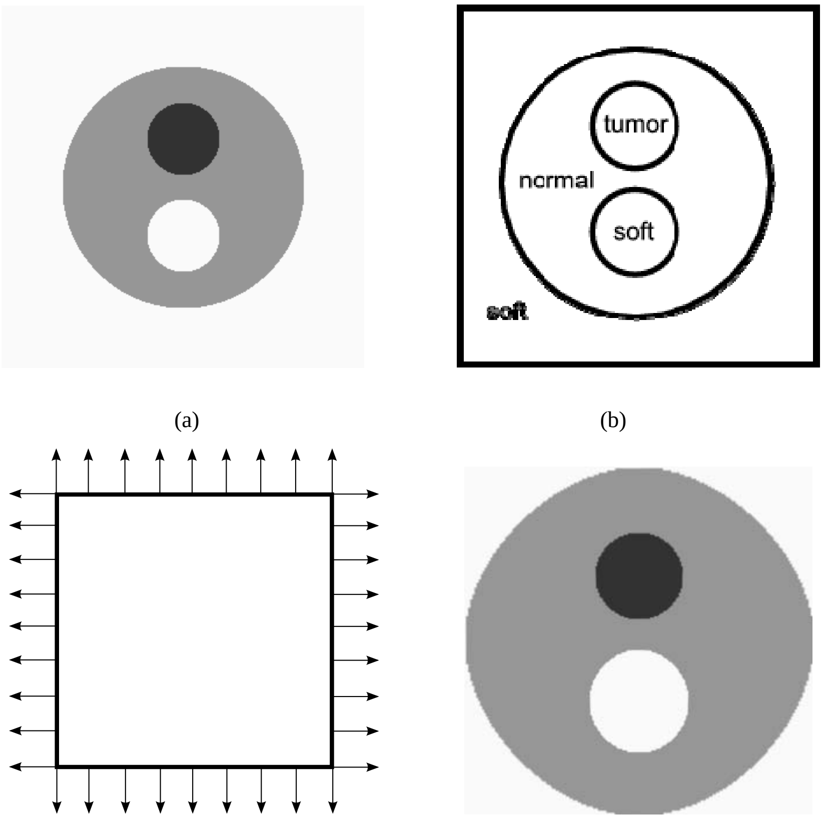
[3] University of Bern, ARTORG Center, 3014 Bern, Switzerland
lutz.nolte@artorg.unibe.ch

[4] University of Bern, ARTORG Center, 3014 Bern, Switzerland
mauricio.reyes@artorg.unibe.ch

[5] Technical University of Denmark, DTU Informatics, 2800 Lyngby, Denmark
rrp@imm.dtu.dk

such as Finite Element Methods (FEM), Mass-spring networks, etc. suffer from the need of transforming the data into a volumetric mesh representation of the object. As a consequence, image segmentation and volumetric mesh generation are two inevitable steps to be performed. Whereas image segmentation can be performed semi- or fully automatically with some acceptable accuracy, automatic volumetric meshing algorithms are prone to errors and dependent on object topology, desired boundary conditions [1], etc. Therefore semi-manual techniques are often used, which is tedious and time consuming. Furthermore, jumping from the image space (i.e. voxels) to the shape space (i.e. mesh) involves approximations due to the discrete and different nature of both data representations [15]. In addition, voxel intensities have valuable information about material properties of the structure, which is required to be translated to the shape space (linear and non-linear interpolations, barycentric-based techniques, etc.) at the price of introducing further errors due to these approximations. The aforementioned limitations of these two approaches (mesh- and image-based techniques) are known by the scientific community and efforts have been made in this direction. In [2, 17] automatic FEM meshing techniques are proposed. The method generates a one-to-one representation of voxels into hexahedra elements and a posterior smoothing approach is applied to deal with jagged edges of the geometry. However, the high number of generated elements yield long computations making this technique only applicable to micro-CT images. In [11] an approach is presented for deformable registration of brain tumor images to a normal brain atlas, here the dissimilarity of the images hinders the usability of readily available deformable image registration techniques. Through statistical modelling of sought tumor-induced deformations the method combines a biomechanical model of tumor mass-effect and a deformable image registration technique. By doing this, the authors reported significant reduction in the registration error. Nonetheless, the reliability of the method depends on the statistical training performed on a set of FEM simulations using different parameters of the tumor mass-effect, such as tumor sizes, locations, external pressures, etc. Inspired by the work presented in [13] here we present a method to simulate soft tissue deformations in the image space that is based on a Maximum a Posteriori (MAP) model of the deformations, which considers novel developed energy terms to account for tissue material properties, boundary conditions and related confidence maps. Stochastic optimization is performed under a Markov Random Field (MRF) approach, which is further extended into a Hierarchical-MRF (HMRF) approach. The use of a HMRF approach enables us to use fast local optimizers, as opposed to global optimizers which are computationally very expensive. The hierarchical approach is also robust with respect to local minima [14]. Finally, MRF and HMRF are well suited for parallel implementation [9, 10], which has brought more attention with the advent of GPU-based application programming interfaces (API).

Test Data



(a) (b)

(c) (d)

Fig. 1. a) input image b) segmentation of the input image c) representation of boundary condition image with outward expansion and d) deformed input image.

A set of synthetic data is used to validate the method. This data is generated so it resembles an MRI image of a brain, as seen in Fig. 1a. To simplify notations three tissue types are considered. Namely, normal matter (gray color), tumoral tissue (black color) and soft tissue (white color), depicted in Fig. 1b. Secondly, a boundary condition image is constructed as seen in Fig. 1c, where the displacement vectors are shown. In the following, the boundary condition image is described by **y**. In this example, the boundary condition image describes an outward expansion. In real applications, the boundary condition image can for example be computed using a non-rigid registration algorithm [3, 16]. To be able to control the amount of confidence in the boundary condition image a confidence image is also provided. In the following, the confidence image is described by **c**. In this implementation, it is a binary vector-valued image where a value of 0 means no confidence in the boundary condition image at that pixel and a value of 1 means complete confidence. In the presented example the confidence image coincides with Fig. 1c except the vectors are normal vectors. The use of confidence fields is described in [12]. Finally, an image **m**, describing the local material properties is used. In our example, it corresponds to the input image where each pixel is assigned the Young's modulus stiffness based on the underlying tissue. A value of 40 MPa is used for tumors, 20 MPa for the normal tissue and 10 MPa for soft tissue. In a real application, the mechanical properties can be assigned using a segmentation and classification algorithm. In conclusion, the following images are used as input for the proposed method: segmented input image, boundary condition image **y**, confidence image **c** and mechanical properties image **m**.

Methods

The aim of our method is to compute a displacement field that conforms both to a given boundary condition image and a set of mechanical properties of the underlying tissue. In the following it is demonstrated how this is solved using Markov Random Field (MRF) regularization. Using MRF enables us to describe the underlying mechanical properties of the tissue using a prior energy term, and to set the boundary conditions using an observation energy term. Because no initial guess for the displacement vector field is available the field is set to null vectors at all sites.

The displacement vector field is described using a multivariate random variable **D**. A realization of the vector field is described by **d**. In addition, each vector in the displacement field is also described by a multivariate random variable $\mathbf{d}_s$, were *s* denotes the spatial location of the vector. This will be explained in more details in the next sections.

Maximum a Posteriori and Markov Random Fields

The final objective is to find a vector displacement field $\hat{\mathbf{d}}$ that maximizes the posterior probability *P*,

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d}} P(\mathbf{d} \mid \mathbf{y}), \tag{1}$$

given a boundary condition image **y**. MRF are defined on a finite index set *S* with elements *s* called sites. Random variables are associated to each element in *S*. To be able to compute the posterior probability the theorem of Hammersley-Clifford is used, which establishes the equivalence between MRF and Gibbs distribution, a proof can be found in [4, 18]. With the help of the Gibbs distributions

$$P(\mathbf{d}) = \frac{1}{Z}\exp(-U(\mathbf{d})/T), \tag{2}$$

where *Z*, *U* and *T* are the normalizing constant, the energy function and the temperature, respectively, the probability can be defined in terms of energy functions. Energy functions are the summation of potential functions

$$U(\mathbf{d}) = \sum_{C\in\mathcal{C}} V_C, \tag{3}$$

where V_C are defined on local neighborhood systems *N*. A local neighborhood system is a subset of *S*, usually only a few sites are involved. *C* is called a clique if any two different elements of *C* are neighbors, so *C* is a subset of *N*. The set of cliques will be denoted by $\mathcal{C}$. The Markov property of MRF is expressed as [5]

$$P(\mathbf{d_s} \mid \mathbf{d_t}, t \neq s) = P(\mathbf{d_s} \mid \mathbf{d_t}, t \in N_s)\forall s \in S, \tag{4}$$

where N_s is a neighborhood system at site *s*, $\mathbf{d_s}$ and $\mathbf{d_t}$ are vectors at site *s* and *t*, respectively. It means that the probability distribution of each site only depends on the state of neighboring sites. In Bayesian terms the posteriori probability can be written as

$$P(\mathbf{d} \mid \mathbf{y}) = \frac{P(\mathbf{y} \mid \mathbf{d})P(\mathbf{d})}{P(\mathbf{y})}. \tag{5}$$

In image processing applications *P(***d***)* stands for the a priori energy term, which represents prior knowledge about the configuration of **d**. *P(***y|d***)* represents the observation function, which describes the relation between **y** and **d**. *P(***y***)* is the density of **y**, which is a constant when **y** is given. The combination of (2), (3) and (5) yields

$$P(\mathbf{d} \mid \mathbf{y}) \propto \exp(-U_{\text{total}}/T), \tag{6}$$

where

$$U_{\text{total}} = U_{\text{prior}} + U_{\text{observation}}, \tag{7}$$

is the sum of observation and prior energy. To maximize (6) one can minimize (7).

Prior Energy

In the current implementation, the prior energy is based on the mechanical properties of the underlying tissue. The prior energy should therefore be formulated so it is at its minimum when the deformation of the tissue conforms to the expected mechanical properties. Mechanical properties of the tissue are modeled using a finite difference approach, where the local tissue characteristics are based on Young's modulus. The used energy function is then

$$U_{\text{prior}}(\mathbf{d},\mathbf{m}) = \sum_{C_1\in \mathrm{C}} V_{C_1} + \sum_{C_2\in \mathrm{C}} V_{C_2} + \sum_{C_3\in \mathrm{C}} V_{C_3} + \sum_{C_4\in \mathrm{C}} V_{C_4}, \tag{8}$$

where C_1, C_2, C_3 and C_4 are four different kind of cliques. Each involving three pixels, which can be visualized as in Fig. 2.

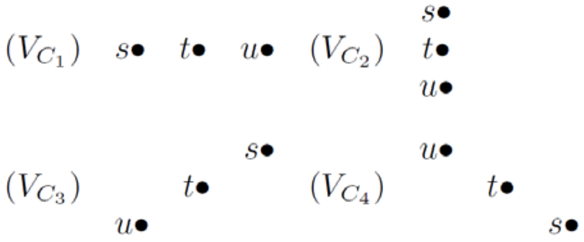


Fig. 2. Four cliques for the prior energy.

The four types of cliques are assigned with potential functions to described the biomechanical process of deformation:

$$V_{C_1} = \left(\frac{m_t}{m_u}(d_{x_t} - d_{x_s}) - (d_{x_u} - d_{x_t})\right)^2 + \left(\frac{m_t}{m_s}(d_{x_t} - d_{x_u}) - (d_{x_s} - d_{x_t})\right)^2 \tag{9}$$

$$V_{C_2} = \left(\frac{m_t}{m_u}(d_{y_t} - d_{y_s}) - (d_{y_u} - d_{y_t})\right)^2 + \left(\frac{m_t}{m_s}(d_{y_t} - d_{y_u}) - (d_{y_s} - d_{y_t})\right)^2 \tag{10}$$

$$V_{C_{3,4}} = \Big| \underbrace{\frac{m_t}{m_u}(\mathbf{d_t}-\mathbf{d_s})}_{(a)} - \underbrace{(\mathbf{d_u}-\mathbf{d_t})}_{(b)} \Big|^2 + \Big| \underbrace{\frac{m_t}{m_s}(\mathbf{d_t}-\mathbf{d_u})}_{(c)} - \underbrace{(\mathbf{d_s}-\mathbf{d_t})}_{(d)} \Big|^2, \quad (11)$$

where $\mathbf{d_i} = [\mathbf{d_{xi}}\ \mathbf{d_{yi}}]^\mathbf{T}$ and m_i are the displacement vector at site *i* and the biomechanical property image sampled at *i*, respectively. Indicated by letters (a) to (d) there are four parts that define the prior knowledge, analog for (9) and (10). Parts (a) and (b) describe the change of Young's modulus in one direction and (c) and (d) the change in the opposite direction. The goal of the prior energy is to reach:

$$\frac{m_t}{m_u}(\mathbf{d_t}-\mathbf{d_s}) = (\mathbf{d_u}-\mathbf{d_t}), \quad (12)$$

at every site. The relative change at site *t* is equal to the relative change at *u* by a ratio of m_t/m_u. The differences $(\mathbf{d_t}\text{-}\mathbf{d_s})$ and $(\mathbf{d_u}\text{-}\mathbf{d_t})$ describe the elasticity in terms of relative displacements at site *t* and *u*, respectively. To assign an energy value to the relative difference this translates to

$$\left| \frac{m_t}{m_u}(\mathbf{d_t}-\mathbf{d_s}) - (\mathbf{d_u}-\mathbf{d_t}) \right|^2 . \quad (13)$$

This means that the minimum energy at site *t* is given by 0. The bigger the difference the higher the energy. Also by taking the square of the energy value, values >1 contribute more than values <1. The energy function also detects non-monotonic fields. A monotonic field is defined as

$$|\,\mathbf{d_s}\,| \leq |\,\mathbf{d_t}\,| \leq |\,\mathbf{d_u}\,| \quad (14)$$

or

$$|\,\mathbf{d_s}\,| \geq |\,\mathbf{d_t}\,| \geq |\,\mathbf{d_u}\,|. \quad (15)$$

Example 1. [Zero energy at site *t*] In this example a situation is presented where two neighboring pixel have different Young's modulus values. The minimum energy is reached when (12) is fulfilled. A possible configuration for three displacement vectors $\mathbf{d_s}$, $\mathbf{d_t}$ and $\mathbf{d_u}$ is given.

$$\mathbf{d_s} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ mm, } \mathbf{d_t} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \text{ mm, } \mathbf{d_u} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \text{ mm, } m_t = 20 \text{ MPa, } m_u = 10 \text{ Mpa}$$

$$\frac{20}{10}\left(\begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) = \left(\begin{pmatrix} 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix}\right)$$

Example 2. [Non-monotonic field at site *t*] In the second example both pixel have the same Young's modulus. Inserting the example values into (13) gives the energy value.

$$\mathbf{d_s} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ mm, } \mathbf{d_t} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \text{ mm, } \mathbf{d_u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ mm, } m_t = 10 \text{ MPa, } m_u = 10 \text{ MPa}$$

$$\left|\frac{10}{10}\left(\begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) - \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix}\right)\right|^2 = 2.8$$

Even though the difference between sites and the biomechanical properties are the same, a non-zero value is calculated. Non-monotonic fields cause overlapping of sites, which is not realistic.

Observation Energy

The current application uses the observation energy to set the boundary conditions:

$$U_{\text{observation}}(\mathbf{y}, \mathbf{c}, \mathbf{d}) = \sum_{C_5 \in \mathrm{C}} V_{C_5} + \sum_{C_6 \in \mathrm{C}} V_{C_6}, \tag{16}$$

where C_5 and C_6 are cliques at single sites *s* and

$$V_{C_5} = p_{x_s} \mid y_{x_s} - d_{x_s} \mid \tag{17}$$

$$V_{C_6} = p_{y_s} \mid y_{y_s} - d_{y_s} \mid, \tag{18}$$

where **p** is the penalty vector field image. The penalty image is proportional to the confidence image

$$p_{x_s} = Kc_{x_s}, \quad p_{y_s} = Kc_{y_s}, \tag{19}$$

where $\mathbf{c_s} = [c_{xs}\ c_{ys}]^T$, $\mathbf{p_s} = [p_{xs}\ p_{ys}]^T$, and K is the penalty constant. By issuing a penalty, a low energy value is sensitive to the difference between the boundary vector and the displacement vector. Hence, the probability is high when the displacement vector at a site coincides with the boundary condition vector whenever boundary conditions are available, indicated by the confidence vector.

Hierarchical Markov Random Fields

The idea is to propagate the geometrical information through hierarchical levels. Low resolution scale levels express global geometrical features whereas high resolution scale levels are used for fine tuning. Geometrical information is needed to converge to a global solution. By using a hierarchical approach, local energy optimizers can be used to find global solutions. Local optimizers are stable and predictable whereas global optimizer do not guarantee convergence and the computation time is enormous, this is further discussed in the next section.

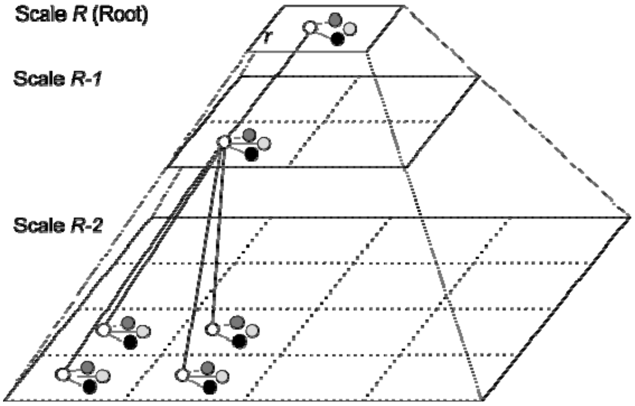


Fig. 3. Dependency graph corresponding to a quadtree structure. With white, light gray, dark gray and black circles representing displacement vectors, boundary condition vectors, confidence vectors and mechanical property vectors, respectively.

In [14] HMRF for segmentation is presented, it follows the adaptation to the presented method. Let *G = (S,L)* be a graph composed of a set *S* of nodes and a set *L* of edges. A tree is a connected graph that has no cycle. Each node has a unique parent node, except the root node *r*. A quadtree, as illustrated in Fig. 3, is a special case of a tree. Each node in a quadtree has four child nodes except the leave nodes, which are terminal nodes. Set *S* can be partitioned into subsets we call *scale levels*. The subset, $S = S^0 \cup S^1 \cup \ldots \cup S^R$, are distinguished according to the path length from each node to the root. S^0 is the subset with the most elements and $S^R = \{r\}$ the one with the least elements. A subset contains 4^{R-n} sites, where *R* represents the scale and *n* the distance from the root node *r*.

Different interpolation methods are used for projecting the images from one scale level to the next. **d** is interpolated linearly, because linear elastic behavior was used to describe the material properties of the image. On the other hand, for the interpolation of **m**, **y** and **c**, nearest neighbor interpolation is used to preserve information at interfaces between different tissue types and prevent over-smoothing. In Fig. 3 the propagation between scale levels is shown, starting on top with the root element *r*. To each white circle $\mathbf{d_s} \in \mathbf{d}$, three circles $\mathbf{m_s} \in \mathbf{m}$, $\mathbf{y_s} \in \mathbf{y}$ and $\mathbf{c_s} \in \mathbf{c}$, are attached, representing interpolated pixel values at corresponding locations *s*.

Energy Minimization

In practice, the crucial point of optimization is a good initial guess. With initial guesses far off the global minimum, local energy optimizers like Iterated Conditional Modes (ICM) [18] will most likely stay at a local minimum and global energy optimizers like Markov Chain Monte Carlo (MCMC) with Simulated Annealing (SA) [18] methods will take a long time; if the cooling schedule is not adjusted to the problem, it might even stop at a local minimum as well. In the current approach, the overall geometrical structure of tissues and their corresponding mechanical properties are crucial in obtaining a good initial guess. A top-down hierarchical approach is used to supplement the local nature of the used energy terms. Each level of the hierarchy is used as an initial guess for the next level. Iterated Conditional Modes is used as local optimizer within each hierarchy level because a good initial guess from the previous level can be assumed.

Hierarchical Markov Random Fields on GPU

Even though ICM is used as optimization method, computation of the entire quadtree is expensive. As will be explained in the following sections around 70 iterations per scale level are needed to reach convergence. By implementing the method on graphics hardware the computation time can be reduced drastically. Due to the fact that MRF is only looking at neighborhood information it is fairly easy to implement it in a highly parallelized way.

Implementation

In [9, 10] the general way of MRF on graphics hardware is presented. For the implementation a new extension for OpenGL, EXT_framebuffer_object [6, 7], was used. Code execution on the GPU is done through shader programs. The shader program is executed on each pixel simultaneously. As input arguments the shader program has access to read-only memory; for output values the shader program has access to write-only memory. The EXT_framebuffer_object enables alternating between read and write buffers directly without copying the buffers back and forth after one iteration has been completed. Two textures are created as read/write data structures on the GPU. One framebuffer object is created and the two textures are attached. Using the glDrawBuffer extension, ping-pong rendering is implemented. Each scale level is computed until the finest scale level is reached.

The current implementation has the two restrictions: First, the number of iterations needs to be predefined before the simulation is started. Second, one can only read or write to memory and not both, this means that we can not update the image during the iteration process.

The pseudo code for the implementation is listed in Alg. 3.7. U_{total} is calculated according to (7) and Γ is a set of values describing the next estimate, e.g. Γ ={(0.1,-01),(0.001,0.001),…}. By varying Γ one can set the step size for the optimization method. Line 3 to 6 are the projection from one scale level to the next. Line 7 to 10 is the ICM minimization. The loop will continue until the full resolution image is reached.

Computation on the quadtree (with notations from Fig. 3) and optimization with ICM:

Data: boundary condition image **y**, confidence image **c**, mechanical property image **m**

Result: displacement vector field $\hat{\mathbf{d}}$

```
   initialize d to null vector field
   for n=R to 0 do
• projection of d from scale $n$+1 to $n$ by linear interpolation
• projection of y from scale 0 to $n$ by nearest neighbor interpolation
• projection of c from scale 0 to $n$ by nearest neighbor interpolation
• projection of m from scale 0 to $n$ by nearest neighbor interpolation
• repeat
• foreach site $s \in S$ do
• $\mathbf{d}_s \leftarrow \arg\min_{e \in \Gamma} (U_{\text{total}}(\mathbf{y}_s, \mathbf{c}_s, \mathbf{d}_s + e, \mathbf{m}_s))$
• until $U_{\text{total}}$ stabilizes
```

Experiments and Results

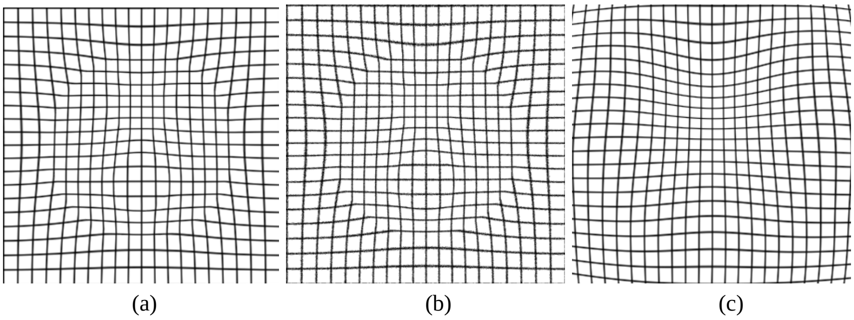


Fig. 4. Deformation grids: a) HMRF method b) ABAQUS and c) elastic registration method.

In general non-rigid deformation algorithms do not include mechanical property information which results in non-realistic deformation. Therefore a comparison with an existing elastic deformation registration method [3] was made. As will be explained in Sec. 4.1 a FEM model was taken as the ground truth deformation. The elastic registration method was used to register the deformation image received from ABAQUS[6] with the input image. The resulting displacement vector fields are shown as deformation grids in Fig. 4a-c. The HMRF model and ABAQUS show similar deformations whereas the elastic registration method does not consider mechanical properties, which can be seen by looking at tissue interfaces. The elastic registration image looks smooth because the change of biomechanical properties at the interfaces is not taken into account.

Ground Truth

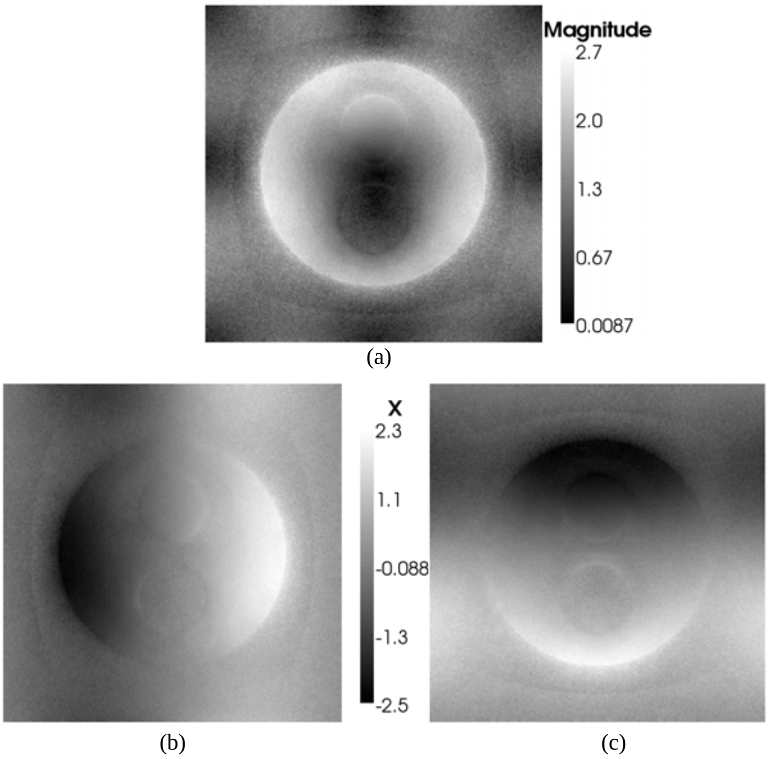
To validate the results of the simulation the same mechanical deformation in a FEM model in ABAQUS was implemented and used as ground truth image. A comparison of data using a root mean square deviation (RMSD) was realized,

$$RMSD(x) = \sqrt{\frac{1}{|S|}\sum_{s\in S}|\mathbf{a}_{\mathrm{s}} - \hat{\mathbf{d}}_{\mathrm{s}}|^2}, \tag{20}$$

where $\mathbf{a}_{\mathrm{s}}$ and $\hat{\mathbf{d}}_{\mathrm{s}}$ are the displacement vector obtained from ABAQUS and our model, respectively.

[6] ABAQUS is a software tool to create FEM models

Quality



(a)

(b) (c)

Fig. 5. Experimental data obtained to evaluate quality compared to FEM, 4255 elements of type CPS3 (3-node linear) and CPS4R (4-node bilinear, reduced integration with hourglass control) are used for the FEM calculations.

Table 1. ABAQUS and HMRF displacement vector field images differences: a) magnitude of the vector b) *x* component of the vector and c) *y* component of the vector.

Method	Mean energy value	RMSD error FEM - HMRF	Iterations	Computation time
HMRF	0.19404	1.44 pixel	700	1.63 sec.
FEM	-	-	-	1.30 sec.

The comparison with FEM results are shown in Table 1. In Fig. 5a the difference image with ABAQUS results is shown. It can be seen that the region that differ the most are around the outer corner of the synthetic brain. Further it is interesting to look at the two Fig. 5b-c in which the difference per component is depicted. In both images the results show that most of the differences occur at the outer border of the synthetic brain, which confirm the interpretation of Fig. 5a.

Computation Time

Table 2. Comparison of computation times between GPU and CPU implementation with 70 iteration steps.

Image Size	GPU Time	CPU Time	Speed-up
300 x 300	1.63 sec.	91.39 sec.	56.07
600 x 600	6.42 sec.	363.89 sec.	56.68
1200 x 1200	26.25 sec.	1445.94 sec.	55.08

In Table 2 a listing on the performance gain with GPU compared to CPU is shown. Three different image sizes have been examined, a speed-up factor of around 55 was found. The number of iterations per scale level is set to 70.

The experiments were performed on a system with ATI, Radeon X1300 Series, and Intel, Core^{TM} 2 Duo E6300 1.86 GHz.

Discussion and Future Work

A method to simulate soft tissue deformations in the image space using physical-based concepts and image processing techniques was presented. The method combines mechanical concepts into a Bayesian optimization framework which has been defined and solved under a HMRF approach and implemented for the GPU. The main advantages of the proposed methodology over previous ones is its mesh-free characteristic, which is normally needed to perform accurate mechanical simulations of tissues. The use of HMRF proves to be an appealing technique to solve the proposed stochastic problem. the reasoning is twofold: On one hand it was found that local minima are avoided by using a hierarchical approach, which in turn allows for a speed-up since it is now possible to use local optimizers rather than slow global optimizers (e.g. Monte-Carlo optimization). Secondly, the nature of the Markovian approach resulted in a straightforward implementation in the GPU, which has been remarked by others [9, 10]. The experiments showed that by using HRMF at the GPU level, stable results can be attainable in a very reasonably computation time, simplifying the soft-tissue-deformation simulation pipeline. Making this method very appealing for medical applications.

In future experiments the focus will be on more complex geometry and non-synthetic image data. In these regards ongoing work includes its extension to 3-D models. An implementation with OpenGL and Cg shader programming was realized. In the future a transfer to the Brook [8], project from Stanford University, or CUDA, NVIDIA, framework will be considered. A strong influence of the number of hierarchical levels on the number of iterations needed for the model to converge are assumed, this topic will also be further investigated in the future.

References

1. Bachtar F, Chen X, Hisada T (2006) Finite element contact analysis of the hip joint. Medical and Biological Engineering and Computing 44(8):643–651
2. Boyd SK, Müller R (2006) Smooth surface meshing for automated finite element model generation from 3D image data. Journal of Biomechanics 39(7):1287–1295
3. Carreras IA, Sorzano C, Marabini R, Carazo J, De Solorzano CO, Kybic J (2006) Consistent and Elastic Registration of Histological Sections Using Vector-Spline Regularization. Computer Vision Approaches to Medical Image Analysis, pp. 85–95
4. Geman D (1990) Random Fields and Inverse Problems in Imaging. In: Saint-Flour Lectures 1988, Lecture Notes in Mathematics, pp. 113–193
5. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(6):721–741
6. Göddeke D (2005) GPGPU Basic Math Tutorial. Technical report, FB Mathematik, Universität Dortmund
7. Green S (2005) The OpenGL Framebuffer Object Extension. NVIDIA Corporation, GDC
8. Gummaraju J, Rosenblum M (2005) Stream Programming on General-Purpose Processors. In: Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture, Barcelona, Spain
9. Jodoin PM, Mignotte M (2006) Markovian segmentation and parameter estimation on graphics hardware. Journal of Electronic Imaging 15(3)
10. Jodoin PM, St Amour JF, Mignotte M (2005) Unsupervised Markovian segmentation on graphics hardware. Pattern Recognition and Image Analysis, pp. 444–454
11. Mohamed A, Zacharaki EI, Shen D, Davatzikos C (2006) Deformable registration of brain tumor images via a statistical model of tumor-induced deformation. Med Image Anal 10(5):752–763
12. Murino V, Castellani U, Fusiello A (2001) Disparity map restoration by integration of confidence in Markov random fields models. In: International Conference on Image Processing
13. Paulsen RR (2004) Statistical Shape Analysis of the Human Ear Canal with Application to In-the-Ear Hearing Aid Design. PhD thesis, Technical University of Denmark
14. Provost JN, Collet C, Rostaing P, Pérez P, Bouthemy P (2004) Hierarchical Markovian segmentation of multispectral images for the reconstruction of water depth maps. Comput. Vis. Image Underst 93(2):155–174
15. De Putter S, van de Vosse FN, Gerritsen, FA, Laffargue F, Breeuwer M (2006) Computational Mesh Generation for Vascular Structures with Deformable Surfaces. International Journal of Computer Assisted Radiology and Surgery 1(1):39–49
16. Rueckert D, Sonoda LI, Hayes C, Hill DLG, Leach MO, Hawkes DJ (1999) Nonrigid registration using free-form deformations: Application to breast MR images. IEEE Transactions on Medical Imaging 18(8):712–721

17. Sigal IA, Hardisty MR, Whyne CM (2008) Mesh-morphing algorithms for specimen-specific finite element modeling. Journal of Biomechanics, In press.
18. Winkler G (2006) Image Analysis, Random Fields and Markov Chain Monte Carlo Methods. Springer, Second edition