

Diplomarbeit
Winter 2006/2007

LOCAL STRAIN MAPPING FOR BONE MICROSTRUCTURE
UNDER LOAD

Author: David Christen

ETH Zurich
Institute for Biomedical Engineering

Supervisor: Prof. Dr. Ralph Mueller

Advisors: Romain Voide
Prof. Dr. Harry van Lenthe
Prof. Dr. Steve K. Boyd

Zurich, April 5, 2007

LOCAL STRAIN MAPPING OF BONE MICROSTRUCTURE UNDER LOAD

1. INTRODUCTION

Several mechanical testing devices have been developed to investigate bone mechanical competence. Among them, Image-Guided Failure Assessment (IGFA) is of particular interest for observing and locating local changes and deformations in bone structures under load. IGFA, also referred to as functional microimaging, utilizes micro-compression in combination with micro-computed tomography and allows direct three-dimensional visualization of bone deformation as well as fracture initiation and progression at the microscopic level. This tool is already very powerful and permits new insights into bone structure behavior under mechanical loading, but it can be greatly improved if the local strains and deformations of the bone structure could not only be visualized but also be quantified. A few computational methods to determine local strains from medical 3D images have already been developed and implemented as along with rigid and non-rigid image registration tools.

The goal of this project is to develop and validate a computational strain mapping algorithm based on existing digital image correlation codes or non-rigid image registration methods. The main tasks of the student will be to implement an image registration code on the local operating system. Once the code is running, the student will validate it and, finally, apply it to a set of existing 3D images in order to quantify local strains in trabecular bone.

2. TASK LIST

- Writing of detailed time-table.
- Literature orientation.
- Analysis of existing digital image correlation codes for strain mapping methods.
- Determining the best solution for the defined requirements.
- Implementing and installing the strain mapping code on the local operating system.
- Testing and evaluating the strain mapping code.
- Performing strain mapping on a set of existing data.
- Discussion of results, writing of a detailed report.

3. LITERATURE

1. R. Müller, S. C. Gerber and W. C. Hayes. Micro-compression: A novel technique for the nondestructive assessment of local bone failure. *Technology and Health Care*, 6, 1998, 433-444.
2. A. Nazarian and R. Müller. Time-lapsed microstructural imaging of bone failure behavior. *Journal of Biomechanics*, 37, 2004, 55-65.
3. B. K. Bay, T. S. Smith, D. P. Fyhrie and M. Saad. Digital volume correlation: Three-dimensional strain mapping using X-ray tomography. *Experimental Mechanics*, 39 (3), 217-226.
4. E. Verhulp, B. Van Rietbergen, R. Huiskes. A three-dimensional digital image correlation technique for strain measurements in microstructures. *Journal of Biomechanics*, 37, 2004, 1313-1320.
5. R. Müller, S. Boyd *et al.* Micro-mechanical evaluation of bone microstructures under load. *Proceedings of Spine*, 4503, 2002, 189-200.
6. C. Mattmann and A. Kuhn. Site-specific measurement of micro-structural bone adaptation and prediction of adapting bone architecture. Diplomarbeit, 2004, IBT, ETHZ.
7. S. Yerby *et al.* The effect of boundary conditions on experimentally measured trabecular strain in thoracic spine. *Journal of Biomechanics*, 31, 1998, 891-897.
8. T. McKinley, B. Bay. Trabecular bone strain changes associated with subchondral stiffening of the proximal tibia. *Journal of Biomechanics*, 36, 2003, 155-163.

Diplomarbeit im WSo6/07

Beginn: 9. Oktober 2006

Ende: 10. April 2007

Leitung:

Prof. Dr. R. Müller
Institute for Biomechanics

Betreuung:

Voide Romain
Tel.: 044/ 632 45 80
voider@ethz.ch

Prof. Dr. Steven K. Boyd
University of Calgary
Tel.: ++1 403 220 41 73
skboyd@ucalgary.ca

Prof. Dr. G. Harry van Lenthe
Tel.: 044/632 45 91
vanlenthe@ethz.ch

Zürich, 23. August 2006

Abstract

Strain mapping is an important tool for the evaluation of image guided failure assessment (IGFA) experiments. In this Diplomarbeit, a novel 3D strain mapping technique based on a deformable registration of micro-computed tomography images is presented and validated. Prior to the validation of the strain map itself, the local displacements computed by the Demons Deformable Registration had to be validated. Compression tests of five cylinders of dense aluminum foam were used for the validation. Comparable results were achieved for trabecular bone.

In a first approach to the validation of the local displacements, images of the aluminum samples were deformed using known deformations. The displacements computed by Finite Element Analysis (FEA) were applied to the images and it was tested if the deformable registration can recover the correct displacements from the deformed and undeformed images. Comparing the original displacements and the ones computed by the registration, a correlation of $R^2 = 0.87 \pm 0.06$ and an accuracy equal to the element size was measured when 2% compressive strain was applied to the samples. The good correlation showed that the deformable registration yielded physically meaningful deformations. The round-off error to integer positions when artificially applying displacements to the images and registration artifacts, due to the use of segmented images, had a negative impact on the accuracy, that was measured.

With the unsegmented images of the actual compression tests, the deformable registration showed to be even more precise. When the result of the registration of the first compression step of 1% applied strain was compared to the displacements computed by FEA, for two of the five samples, a correlation of $R^2 = 0.9$ and an accuracy of the displacements of 1/3 of the element size was measured. In the other samples, experimental artifacts, such rotations due to off-center loading or misaligned end-plates leading to an unwanted bending of the samples yielded deviating results for FEA and registration. However, a visual inspection showed, that the displacements computed by the deformable registration reproduced the correct deformations for all the samples. The registration error increased, if compression steps of 4% applied strain or more were used.

Very accurate displacement fields are needed for strain maps with a high spacial localization. Strain maps of complex microstructures such as trabecular bone are only helpful if the loading of individual structural elements is evaluated. While strain computed by FEA indicates deformed regions within individual trabeculi, strain maps can only characterize structures with an accumulation of strain. Due to this different localization of strain, a statistical validation was not possible. But visual inspection showed, that the characterization of regions with high strain (Von Mises strain above 5%) was reliable, as long as compression steps of up to 2% were used for the deformable registration.

3D strain mapping might also be used for the evaluation of IGFA experiments with different tissues and at higher resolutions as in nanoscopic imaging.

Contents

1	Introduction	1
1.1	Micro-computed tomography	1
1.2	Image registration	1
1.3	Local strain mapping	2
2	Previous work	3
2.1	Digital volume correlation by B. K. Bay <i>et al.</i>	3
2.2	3D digital image correlation by E. Verhulp <i>et al.</i>	5
2.3	Demons deformable registration	5
2.3.1	Recent implementations	6
3	Validation of the demons deformable registration using artificial displacements	8
3.1	Method	9
3.1.1	An introductory example	10
3.1.2	Forward and reverse warping	12
3.2	Results of the finite element analysis	13
3.3	Results of the reverse warping approach	16
3.4	Results of the forward warping approach	19
3.4.1	Displacements in the loading direction	20
3.4.2	Displacements perpendicular to the loading direction	25
3.5	Discussion	29
4	Validation of the demons deformable registration using experimental displacements	30
4.1	Results	32
4.2	Applying the registration to bigger compression steps	38
4.3	Applying the registration to trabecular bone	42
4.4	Discussion	46
5	Validation of local strain mapping	47
5.1	Method	47
5.2	Results	49

5.2.1	Strain mapping based on the registration of artificially deformed images	49
5.2.2	Strain mapping based on the registration of IGFA images	50
5.2.3	Strain mapping a sequence of IGFA compression steps	56
5.3	Discussion	59
6	Conclusion and outlook	60
6.1	Conclusion	60
6.2	Outlook	61
A	Documentation of the applications	65
A.1	DeformableRegistration	65
A.2	FEWarper	66
A.3	DFieldInverter	66
A.4	Correlate	67
A.5	DFieldGen	67
A.6	DispToAIM	67
A.7	Euclidean	68
A.8	ForwardWarp	68
A.9	ObjWarp	69
A.10	Res2AIM	69
A.11	Strainmap	70
A.12	SynthGrad	71
B	Description of the work flow	72
B.1	Deformable registration	72
B.1.1	Preprocessing	72
B.1.2	Registration	73
B.1.3	Post-processing	73
B.2	Strain mapping	73
B.2.1	Post-processing	74
B.3	Advanced applications	74
B.3.1	Combining the registration of subsequent compression steps	74
B.3.2	Strain map based on the deformed sample	76

Chapter 1

Introduction

1.1 Micro-computed tomography

Micro-computed Tomography (μCT) is a non-invasive, non-destructive 3D imaging method with a spatial resolution in the range of micrometers. Like in Radiology and traditional Computed Tomography, X-Rays are used. In Computed Tomography the 3D information is computed through the excess information of the many snapshots taken from various angles. The much higher resolution of μCT compared to ordinary Computed Tomography is achieved by scanning smaller samples and placing them very close to the X-Ray source. In biomedical engineering, among other applications μCT is used to analyze the micro structure of bone. Because of its non-destructive nature, μCT can not only be used for static measurements. It is ideal for dynamic measurements of microstructures (i.e. trabecular bone) under increasing load [1] and even in vivo measurements are feasible [2].

1.2 Image registration

In computer vision, the alignment of images of the same object taken at different times, under different conditions or under a different perspective is generally called image registration. It usually involves transferring the two data-sets into a common coordinate system to make them comparable.

It can be differentiated between undeformable and deformable registration. The first treats the images as rigid bodies and includes rigid body motions only. Usually metric for the similarity of the two images (i.e. correlation or mutual information) is used to find the optimal transformation [3]. This approach is used, for example, to make consecutive in vivo μCT measurements comparable [4].

In deformable image registration, there is no restriction to rigid body motions. The image is modeled as a elastic material instead. The matching of the individual image features yields local displacements. Algorithms can be classi-

fied into area-based and feature-based methods. In area-based algorithms, often called *digital image correlation* (DIC), subsections of the images are matched as in undeformable image registration. Local displacements are then computed by interpolating between the local transformations. Whereas feature-based algorithms try to match local features such as lines, gradients, corners or artificial markers.

Both approaches to the deformable registration depend on a high number of local features for a precise matching of the images. Most of the applications, especially in material testing, therefore involve so called speckle images. Labels are applied to the specimen so that a periodic speckle pattern appears in the image. Suitable labels are usually printed or sprayed and depend on the image modality and the kind of specimen. Only if the image involves enough local features a deformable registration of the images is possible. In this study, no additional labels were applied to the samples. The registration was based only on the intrinsic features of the specimens.

1.3 Local strain mapping

Local strain mapping at a structural level is important in order to understand the mechanical behavior of complicated micro-architectures such as trabecular bone. Macroscopic mechanical experiments with trabecular bone can not account for local phenomena like damages or bone remodeling. Also understanding failure mechanisms of trabecular bone under load requires knowledge about local strains. This in turn is required when implants should be optimized and adapted to individual bone micro-structure or in order to assess the impact of changes in the micro-architecture, as they occur in diseases like osteoporosis.

Local strain mapping at the micro-structural level was already done in 2D [5]. Also simulation tools such as *Finite Element Analysis* (FEA) can compute strain maps. But they are either limited to linear elastic material models or can only be computed for strongly simplified geometries or small sections of trabecular bone.

The components of local strain tensors can be computed from changes of the local displacements. This requires an exact knowledge about the displacement as computing local differences would strongly amplify any noise in the displacement field.

Chapter 2

Previous work

While there have been numerous applications of deformable registration and strain mapping such as using 2D images obtained by microscopy or other image modality, fewer applications presented are based on 3D images. There have been two implementations for Deformable Registration of μCT data. One was originally developed by B. K. Bay at University of California, the other by E. Verhulp at Eindhoven University of Technology. Both techniques use an area based approach to the deformable registration. This means that both are (deformably) registering the two images by nondeformably registering small portions of the images.

2.1 Digital volume correlation by B. K. Bay *et al.*

The Digital Volume Correlation (DVC) technique was first published in 1999 [6]. It is essentially an extension of a standard 2D digital image correlation method. The texture within the material was used for measuring the displacement field by sub-volume tracking. The method mainly consisted of three steps:

- The images of the loaded and unloaded samples were acquired using μCT with a spatial resolution of $35\mu m$. The images extended to about 500 voxels in each dimension.
- Discrete displacement vectors were measured by a correlation procedure. A first registration at integer voxel level was followed by a registration with sub-voxel precision. The sum of squared differences of the values of the two images was used to quantify the quality of the match of the two images. Sub-voxel registration equaled to an unconstrained minimization of the sum of squared differences and was done using the Levenberg-Marquardt algorithm. The interpolation of the image data was done with tricubic interpolation.

- The strain tensor field was computed from the displacement vector field. Beforehand the displacements were filtered by locally fitting a second order polynomial. Based on that a second order approximation of the strain tensor was computed.

An estimation of the measurement precision was made by registering repeated unloaded samples ¹. The authors reported a standard deviation of the displacement error of 0.035 voxels and a standard deviation of the strain tensor error of 0.03%. As they mentioned this is rather measuring the accuracy and reproducibility of the μCT measurement and has little to do with the accuracy of the registration. The strain-maps that were presented were sampled very sparse. They essentially lost their significance on the level of individual trabeculi. It can be assumed that the localization of the strain was omitted because the precision of the registration would not have allowed for it.

In this first study, the registration measured only translations and no rotations. This was changed in a later study [7]. During the registration process, the sub-images were not only displaced but also rotated. This made the displacement field smoother, especially in regions with high strain concentration. Plus, a normalized cross correlation of the two images was used as metric of the fit of the sub-images and the Broyden, Fletcher, Goldfarb and Shanno (BFGS) method [8] was used for its minimization.

Lately a comparison between the results from FEA and digital volume correlation was published [9]. The original algorithm (without considering rotations and with a sum of squared differences metric) was used in a new version, that was optimized for short computation times. A sample of human bone was scanned loaded and unloaded. First the displacements were computed by the registration. According to this, border conditions for a displacement controlled isotropic linear elastic FEA were determined. The results from FEA were averaged over a rather large volume and resampled at a resolution of $1.05mm$ before a linear regression analysis of the two estimations was done. For displacements in the loading direction a correlation coefficient $R^2 > 0.9$ was determined. Displacements perpendicular to the loading direction showed little correlation. An analysis of local strain tensor showed that the components in the loading direction correlated with R^2 between 0.77 and 0.327 depending on whether a femoral sample or a vertebral sample was analyzed. The other components of the strain tensor showed smaller correlation coefficients. It was concluded that DVC could not validate the results from FEA. Mostly because DVC could not measure strain at the level of individual trabeculi, as it used trabecular sized features for the registration.

¹An image was taken from the unloaded sample. The sample was then removed from the scanner and a small load was applied. Before the second scan was taken, the load was removed so that the sample would recover its original shape.

2.2 3D digital image correlation by E. Verhulp *et al.*

This approach aimed at the measurement of strain within a single trabeculi. The correlation algorithm is similar to the DVC. Instead of minimizing the sum of squared differences between the loaded and the unloaded image, the correlation coefficient is maximized. This is done using the BFGS optimization algorithm. Linear deformations of the sub-images are considered during the registration process, that includes translation, rotation and scaling in each dimension. Apart from that, they use the same method to compute the displacement field.

A volume meshing technique based on the marching-cubes algorithm is used to generate a tetrahedrae mesh, similar to a FEA mesh. For each vertex (equivalent to a node in a FE mesh), the displacement is interpolated and a Green Lagrange strain tensor is computed from the displacements within a certain estimation radius. According to these strain tensors the mesh is deformed. For visualization and further analysis, all the data (the original mesh, the deformed mesh, the nodal displacements and element strains) are passed to Mentat, a commercial FEA post-processing suite.

In this publication it was only visually tested that the deformations and rotations were correct. Also the strain map was not validated statistically. Instead the impact of the parameters was analyzed as well as the accuracy of the registration of undeformed aluminum foam samples. Nonetheless this was the first study to measure strain within single trabeculi.

2.3 Demons deformable registration

The Demons Deformable Registration was first proposed by J.-P. Thirion [10] [11]. Instead of the more common “attractor” based deformation techniques, this algorithm tries to push the model into a mold, as if it was an elastic fluid. The registration is an iterative process governed by an image-based force trying to align the images and an elasticity force restricting the difference of the displacements of neighboring portions of the image. Thus rather than describing the algorithm based on images, it can be understood as a deformable model (moving image) and a mold (fixed image). Demons in the sense of Maxwell’s demons in physics ² are assigned to a number of points on the contour of the mold. It computes locally an image force, pushing in the normal direction of the contour: If the point is inside the model, the push is inward, if the point is outside, the push is outward (See figure 2.1).

Instead of assigning demons only to contours of the fixed image, they are assigned to every point of the image or to a fixed grid, using a definition of

²Maxwell’s demons is a thought experiment suggesting a possible violation of the second law of thermodynamics. Situated on semi-permeable membrane, the demons are sorting locally two different types of molecules of a mixed gas, therefore decreasing the entropy of the system.

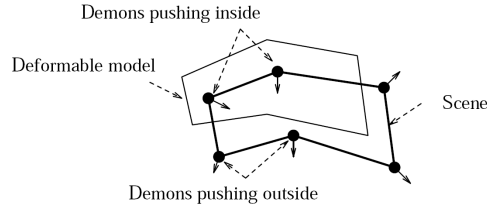


Figure 2.1: Image-based force of the demons. From [10].

the strength of the image force being dependent on the local gradients. Originally it was proposed to only consider the gradients of the fixed image, mainly for computational reasons, as they have to be computed only once for all the iterations, whereas the gradients of the moving image will change with every iteration. This leads to an implementation resembling a gradient descent optimization algorithm. But omitting the gradient of the moving image introduces some asymmetry and therefore making the convergence less reliable. The elasticity is usually modeled as a Gaussian filtration of the local displacements. The stronger the low pass filtering, the more rigid the material appears.

Each iteration consists of four steps:

- Compute the image forces
- Compute a deformation field for the whole image
- Gaussian filter the displacements
- Apply the deformation field to the image

Possible applications presented by Thirion include displaying cardiac motion, inter patient matching and matching patient images to a medical atlas [12]. The algorithm showed to be significantly faster than the previously proposed algorithms [13, 14], especially when a multi-resolution image pyramid was used, with smaller images for the first rather coarse registration steps and the full resolution only for the fine registration.

2.3.1 Recent implementations

In the Insight Toolkit (ITK)³, an open-source image segmentation and registration framework, the Demons Deformable Registration with the asymmetrical

³www.itk.org

formulation of the image force is implemented as one of many registration methods [15, 16]. This original implementation has been used to analyze time-varying 3D images.

In their Diplomarbeit C. Mattmann and A. Kuhn used ITK and its implementation of the Demons Deformable Registration to predict future adaptations of the bone micro-architecture using Simulated Bone Atrophy (SIBA) and in vivo μCT images [4]. As they found the conversion of the asymmetrical formulation not reliable enough, they implemented also the symmetrical formula in the ITK framework and contributed the additions to the project. They used the registration to find out how the architecture of the bone changed between a first and a second measurement. They then inverted the deformations, so that they could predict how the bone would look like in a third measurement.

They encountered problems with the registration as soon as the agreement of the images was not very good, i.e. when trabeculi were only little overlapping or resorbed in the remodeling process. The inversion of the deformation field worked good as long as there were no abrupt changes in the deformations.

Chapter 3

Validation of the demons deformable registration using artificial displacements

In order to validate the displacements computed by the Demons Deformable Registration, we applied known and mechanically meaningful synthetic deformations to μCT images and subsequently try to recover those deformations from the original and the digitally deformed image. Finite Element Analysis (FEA) was used to compute the displacements of every point of the sample in the first place. In this first step of the validation it was not important that those displacements were exactly the same as the ones the bone would have undergone, if it was physically compressed. As they were not compared to any experimental data. But we wanted the displacements to be easily controllable while not as strongly simplified as displacements computed with an analytical beam model would be, plus they still should be somehow related to the microstructure of the sample.

In order to quantify the error of the registration, all other error sources had to be minimized. In the following, two different approaches to the validation are presented differing only in how the displacements were applied to the original image. The validation was done using dense aluminum foam. We expected more homogeneous results than if the validation would have been done with bone samples. We used high density foam, because FEA was more stable and the bigger feature size makes it more representative to trabecular bone. For preliminary results we used only 3 out of 5 samples. For the more thorough part of the validation we used all the 5 samples.

3.1 Method

In a first step, a linear elastic Finite Element Analysis of the segmented μCT image was done with each voxel corresponding to a hexagonal element using *faim*, a custom made large scale FE solver. This yielded among others nodal displacement for the given loading condition being used. For this study we used displacement controlled compression in z-direction and applied 1% overall strain while the bottom was fixed in z-direction. The results of the analysis were written to a text file with the ending *.disp. The *disp_to_aim* application takes the *.disp file with the nodal displacements as an input, computes the displacement for every voxel (this corresponds to the computation of the element displacement) and writes the resulting displacement field as an *AIM* file.

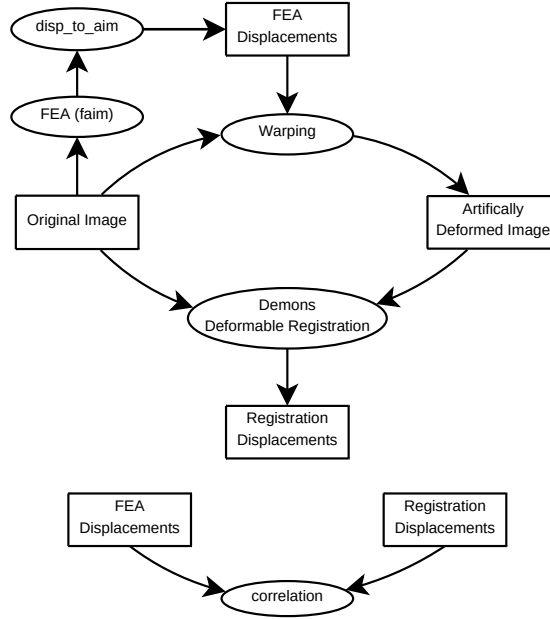


Figure 3.1: Flow Chart of the validation process

Next, the segmented image was warped according to the displacement field computed by the FEA. This was done either by inverting the displacement field and applying it using *reverse warping* or by applying it directly as described below as *forward warping*. The result was a deformed version of the original image.

The original and the deformed versions of the images were then used as inputs for the Demons Deformable Registration. In the ideal case, the registration of the original and the deformed images would yield exactly the displacements originally computed by the FEA and applied to the original image. As a measure for the quality of the registration, the correlation coefficient R^2 between the displacement field computed by the FEA and the one computed by the registration

algorithm was used. As the displacement orthogonal to the loading direction are usually small, correlation of those components is not a reliable measure for the correctness of the registration. As an alternative, we also evaluated the standard deviation of the registration error (SD_{err}). For the computation of the correlation factor as well as for SD_{err} and all other statistical evaluations, all the values outside of the material were removed by a mask and were not considered. The additional knowledge, that the portions of the image that do not contain any material are not moving, would lead to numbers suggesting a much better agreement than there actually is.

3.1.1 An introductory example

Looking at a simplified example can give a better understanding of the mechanism of the registration, the artifacts occurring and the trade-offs. Demons Deformable Registration was run on a synthetic model: A spherical cylinder ($320\mu m$ long, $320\mu m$ outer diameter, $160\mu m$ inner diameter, $10\mu m$ element size) was translated by $6\mu m$ in longitudinal (x) direction. See figure 3.2.

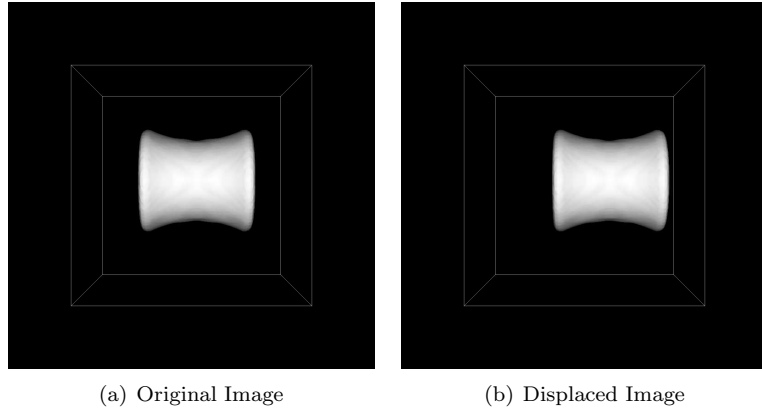


Figure 3.2: A spherical cylinder displaced by $6\mu m$ in longitudinal direction

The registration yields a displacement field with independent components for x, y and z direction. In this example, displacements in x direction are most interesting. In the following the mid-axial slice of the field of the displacements in x direction is discussed. Figure 3.3(a) shows the displacements computed by the registration. Ideally, this would be constant all over the cylinder and zero outside the cylinder. As the Demons Deformable Registration does not know anything about foreground or background, it models the background with the same properties as the foreground and yields a displacement field that is not zero outside the cylinder. Therefore the displacement field was masked with the original (binary) image of the cylinder (figure 3.3(b)).

A closer look at the displacement field in figure 3.3 shows that the maximum displacement was actually outside the cylinder (red, close to the right border of

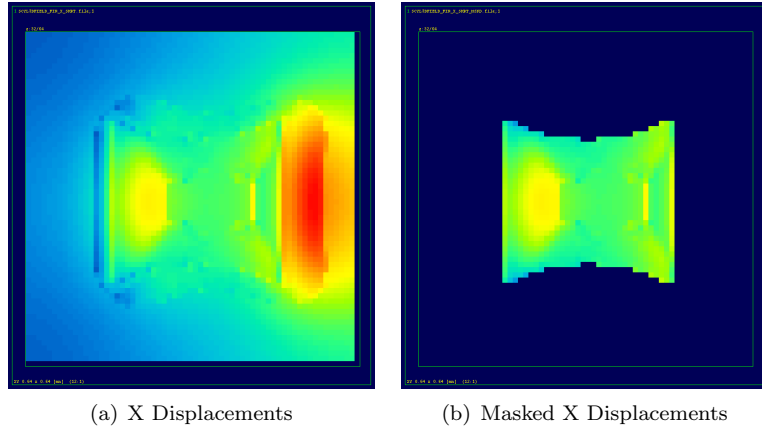


Figure 3.3: Midaxial slice of the field of the displacements in x direction resulting from registering images 3.2(a) and 3.2(b). Blue color corresponds to the minimal displacement and red to the maximal displacement.

the image). Because the foreground was moved to the right, also the background had to move. There might have been some effects of the close image border involved that additionally enhanced the displacement. Another artifact can be seen in the left of the figures 3.3(a) and 3.3(b): The displacements within the cylinder increased towards the core of the cylinder. This was because there was not only the left end of the cylinder moved to the right. Due to the curved sides of the cylinder, also in the middle of the cylinder were image-based forces pulling the points to the right trying to align the images. Gaussian filtration of the computed displacements after each iteration led to an accumulation at the core of the cylinder. This artifact would most likely disappear if the cylinder was larger or the parameters of the registration would have been chosen differently.

The *aperture problem* describes, that if one looks at the middle portion of an object like a straight cylinder, it can not be determined if it moved along its axis of symmetry. This effect was not very pronounced in this example, as the curved sides of the spherical cylinder being used helped to align it in longitudinal direction, but it can still be observed (see figure 3.4(b)): The aperture problem yielded higher displacements close to left and right ends of the cylinder.

The elasticity parameter d of the Demons Deformable Registration can be used to minimize this effect. It determines the sigma of the Gaussian filtration applied to the displacements after each Iteration. If it was chosen high like here ($d = 10.0$) the displacements were more homogeneous, as the impact of the force trying to align the images was distributed over a wider part of the image. This on the other hand led to a slower convergence requiring more iterations. If the elasticity parameter is chosen very low (i.e. $d < 1.0$) the images behave like a fluid. Here, this would lead to low displacement anywhere except close to the borders of the cylinder yielding to a much more pronounced effect of the aperture problem.

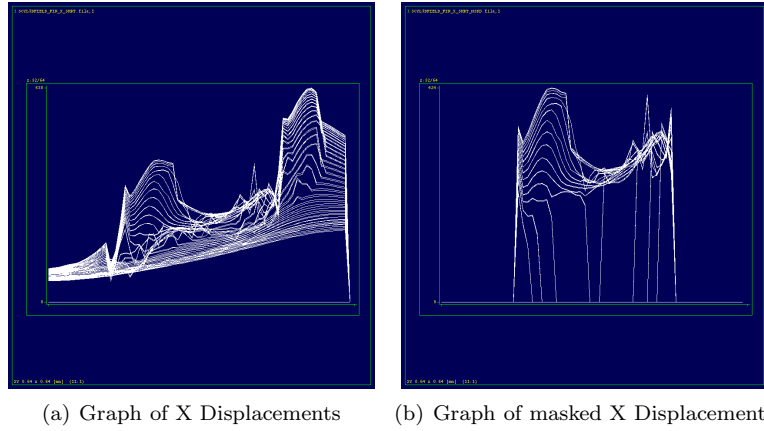


Figure 3.4: The displacements of figure 3.3 plotted row-wise. Each line represents the displacement along one line.

Binary images have to be Gaussian filtered before they can be registered, because too few voxels would have gradients and those gradients would be so high, that problems with numerical precision can occur. Good results were achieved using a Gaussian filtration with *sigma* between 2.0 and 3.0, so that a bigger part of the images had gradients and those were reasonably small.

3.1.2 Forward and reverse warping

Applying displacements to images, so called *warping* is a well established technique in Computer Graphics. But the apparently easiest approach, the so called forward warping, has some caveats. One iterates over the source image and writes the image content to the corresponding point in the output image:

```
for (u=0; u<umax; u++){
    for (v=0; v<vmax; v++){
        float x = u + fx(u,v);
        float y = v + fy(u,v);
        destination[x, y] = source[u, v];
    }
}
```

First of all, whenever a portion of the image is stretched or the displacements change rapidly, holes will occur in the output image. On the other hand, when a part of the image is shrinking, multiple values of the source image will be assigned to the same pixel of the destination image, with the question arising which value is to be assigned.

Therefore, in most computer graphics libraries reverse warping is implemented. This algorithm iterates over the destination image and tries to find the

corresponding pixel in the source image. This usually involves an inversion of the displacement field.

```
for (u=0; u<umax; u++){
  for (v=0; v<vmax; v++){
    float x = u + fx_inv(u,v);
    float y = v + fy_inv(u,v);
    destination[u, v] = source[x, y];
  }
}
```

Provided that the inverse of the displacement field can be computed, this reverse algorithm produces no holes. It can only lead to an oversampling of the source image. But the computation of the inverse displacement is usually not trivial. In ITK an iterative strategy is implemented. As a first guess, the negative displacement is used. This is a very good approximation as long as displacements are small and fairly homogeneous. But as soon as there are abrupt changes of the local deformations this estimation can be completely wrong (see figure 3.5). Thus the round trip error $err = ||f(f_{inv}(x, y))||$ is being minimized by looking at the displacements in the surrounding of the pixel in the source image.

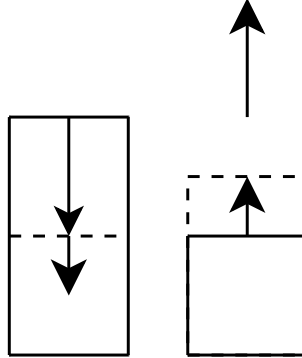


Figure 3.5: As depicted here, the inverse displacement field is not necessarily the same as the negative displacement field. A rectangle is shrunk to half its length. Then the negative displacement field of the transformation is applied to the shrunken shape. If the inverse displacements would have been applied, the rectangle would have recovered the original shape.

3.2 Results of the finite element analysis

The aluminum foam samples being used for this validation study had a diameter of $7mm$ and a length of $10.85 \pm 0.10mm$. The weight was $0.13 \pm 0.01g$. Volume density was $12.80 \pm 1.01\%$ and SMI was 2.22 ± 0.12 .

In a previous study the agreement between FEA and the measurements from the compression test experiment was evaluated. FEA showed a good prediction of the overall stiffness between 0 and 1 % overall strain. After 1%, nonlinear effects led to experimental results increasingly differing from the simulation (unpublished data).

Table 3.1 summarizes the results of the linear elastic Finite Element Analysis. All the displacements in x and y direction were small and only few points were displaced more than the element size of $30\mu m$. Some of the results, especially the displacements in x and y direction were highly stochastic and there was no reason for the inhomogeneity of the standard deviation of x displacements while the standard deviation of y displacements were rather homogeneous. The different mean displacements in z direction were caused by zones where most of the compression occurred: In some samples, this zone was close to the top plate, so that the middle and lower parts of the sample were not displaced much. Whereas in some samples, this zone was close to the bottom leading to larger displacements for a much bigger part of the sample.

As the analysis was linear elastic, we could scale all the results by a factor in order to simulate a compression at 2 or 4% strain, as long as we did not compare these scaled results with the experimental results, it did not matter, that linear elastic FEA is not valid in this range. And as a first step, we only tested if the registration could extract the previously applied FEA displacements.

Table 3.1: Statistical evaluation of the linear Finite Element Analysis for 1% compressive strain in z direction.

(a) Displacements in X Direction			(b) Displacements in Y Direction		
Sample	Mean (μm)	SD (μm)	Sample	Mean (μm)	SD (μm)
1	2.29	8.63	1	-6.62	12.04
2	5.10	15.49	2	7.76	10.21
3	-1.27	9.48	3	16.99	15.86
4	3.10	12.25	4	-2.73	11.25
5	-10.31	32.11	5	2.02	14.09
Total	-0.22 ± 6.09	15.59 ± 9.61	Total	3.48 ± 9.27	12.69 ± 2.27

(c) Displacements in Z Direction		
Sample	Mean (μm)	SD (μm)
1	37.80	45.03
2	43.63	44.29
3	55.44	42.33
4	75.04	32.90
5	59.43	42.33
Total	54.27 ± 14.52	41.38 ± 4.89

(d) Euclidean norm of the Displacements		
Sample	Mean (μm)	SD (μm)
1	40.86	45.33
2	49.73	42.70
3	61.12	41.96
4	78.80	28.26
5	73.74	35.01
Total	60.85 ± 15.90	38.65 ± 6.95

3.3 Results of the reverse warping approach

The inversion of the displacement field is a nontrivial operation and its error had to be assessed. A displacement field was inverted twice to quantify the error. Ideally this would lead to the original displacement field again. In order to visualize the impact of the error on the image a *round trip image* was created: The deformation field computed by the FEA and its inverse were subsequently applied to the image. The round trip image should look the same as the original image and deviations help understanding the weaknesses of the inversion. For the images in figure 3.6 only a sub image of the actual sample was used. For both, the figure and the statistical analysis in table 3.2, the inversion was done using 30 iterations.

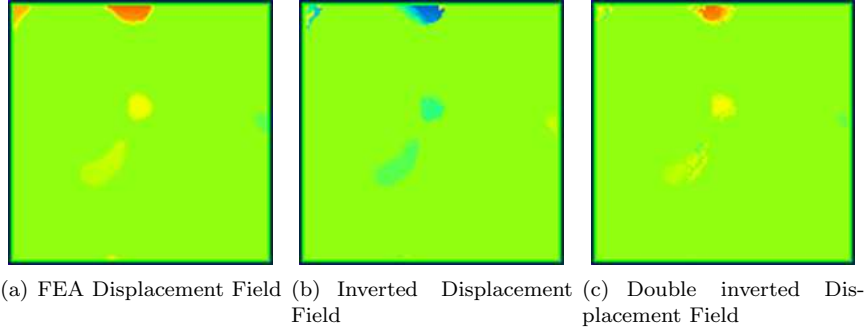


Figure 3.6: A layer of the x component of the displacement field. The sample was compressed by 5%. Blue colour corresponds to the maximal negative displacement ($-0.2mm$), green to zero displacement and red to maximal positive displacement ($0.2mm$).

Table 3.2: Correlation Factor of the original displacement field and the result of two consecutive inversions using ITK's `IterativeInverseDeformationFieldImageFilter`. Three samples have been compressed in z direction with 1% compressive strain

Comp.	R^2
X	0.95 ± 0.02
Y	0.93 ± 0.03
Z	0.92 ± 0.04

Figure 3.6 shows that the inversion is unreliable at the border of the material, even though correlation factors look very good (see table 3.2). The consequence can be seen in figure 3.7: The features are shrinking as the voxels on the surface do not stay aligned with the rest of the material during the warping process. This is due to the displacement that are not homogeneous enough for the inversion to work properly. This effect is especially dramatic in horizontal connections. That is because displacements are biggest in vertical direction

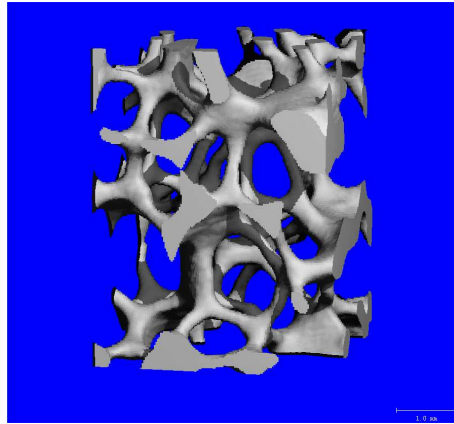
so that also the error of the inversion is biggest perpendicular to the connection. The good correlation values in table 3.2 show that the inversion algorithm works well in general whereas the visual inspection of the image reveals, that it gets unstable with abrupt changes of the displacements. This was not one of the goals when it was originally implemented. It should reliably invert displacement fields resulting from the demons deformable registration only. Those displacement fields do not have any discontinuities, as the algorithm computes a displacement field that is continuous at the surface of the sample trying to match the foreground as well as the background. Furthermore the displacements are Gaussian filtered after each step of the registration. (See figure 3.3(a) for an example displacement field).

In Figure 3.7(b) at the top of the sample some border effects can be observed (indicated with black arrows). The top of the sample does not move down in some places. The inversion algorithm depends on the surrounding and close to the border there might not be any neighborhood around for the iterative search.

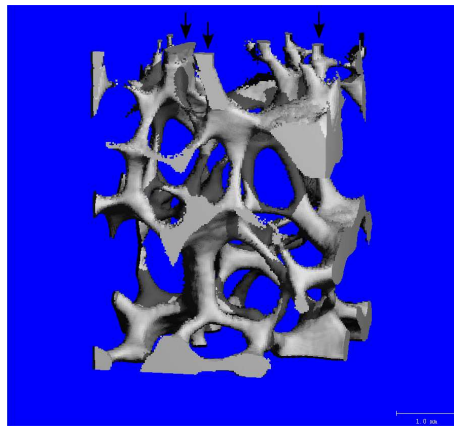
Table 3.3: Correlation factor R^2 of the original displacement field computed by FEA and the one computed by the Demons Deformable Registration. The deformed images was created by reverse warping the undeformed image by the FEA displacement field. Three samples have been compressed in z direction at 1% overall strain

Comp.	R^2
X	0.08 ± 0.05
Y	0.04 ± 0.02
Z	0.22 ± 0.11

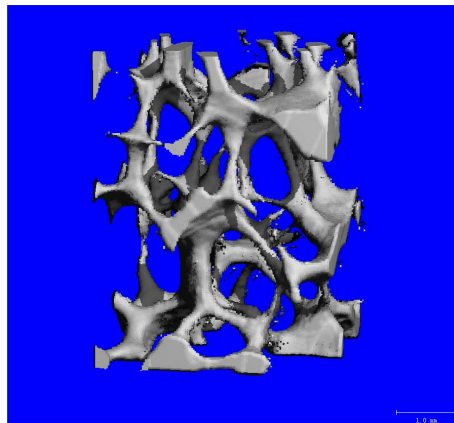
The registration was done with 25 iterations and an elasticity parameter of $d = 10.0$. Prior to this, the samples were Gaussian filtered with $\sigma = 3.0$. When correlating the output of the demons Deformable Registration with the original displacement fields computed by FEA, the error was too big (Table 3.3). It is difficult to estimate how much of the error got introduced by the warping process and what was the error of the registration. The negative impact of the reverse warping on these correlation factors was bigger than table 3.2 suggests, because the error of the inversion accumulates on the edges of the features (see figure 3.7), while the demons registration is mainly acting on the same regions, as gradients are high at the edges. On the other hand, not the whole difference to the results of the forward warping approach presented in the next section is due to the different warping algorithms. There were also other improvements. For instance the higher number of iterations used for the registration increased the correlation factors as well (see figure 3.11 for correlation factors achieved using forward warping and few iterations).



(a) Original Image



(b) Warped Image



(c) Round trip image

Figure 3.7: Original, warped and round trip image with 5% compressive strain applied in vertical direction.

3.4 Results of the forward warping approach

Trying to minimize the error introduced by the warping process, forward warping was implemented. Because ITK and most of the other computer graphics libraries include reverse warping only, it was implemented based solely on *AIM-PACK*. Aimpack is a platform independent library written in C for simplified I/O operations on 3D images. This approach worked surprisingly well with our images and displacement fields: The images were segmented, we were using a rather small compression and features were mostly compressed and not stretched. Compression often leads to cases, where multiple values from the original image are mapped onto the same voxel in the output image. This did not have any impact in our case, because the images were thresholded. Most of the holes occurring were only very small and could be filled with one iteration of dilation and erosion.

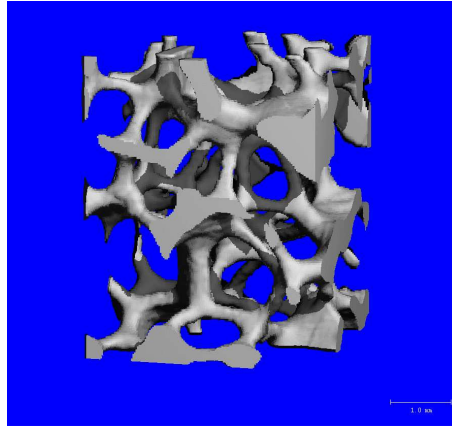


Figure 3.8: The same image as in figure 3.7, but deformed using forward warping. Also with 5% compressive strain applied in vertical direction.

The forward warped image could only be visually inspected. See figure 3.8. But possible errors were limited to round-off errors of the displacements to integer positions of the voxels. The error introduced by a single iteration of dilation and erosion is very small. In order to reduce further, one iteration of dilation and erosion was also applied to the undeformed image prior to the registration.

For the following statistical evaluation all 5 samples of dense aluminum foam had been used. If not stated otherwise, 100 iterations were used for the registration with an elasticity parameter of $d = 10.0$ and the images being Gaussian filtered with $\sigma = 3.0$ prior to the registration. As the results showed a remarkable improvement over the results presented in section 3.3, the samples were tested with 1, 2 and 4% applied strain.

3.4.1 Displacements in the loading direction

As described in section 3.2, significant displacements occurred only in z direction. So that this validation study is mainly based on this component of the displacement field. Displacements in x and y directions could be used as well, but only if grayscale images are registered, so that registration with sub-voxel precision is possible.

Because of the significant displacements, correlation is a good measure for the quality of the registration in z direction. For 1% compressive strain, an average correlation factor of $R^2 = 0.73 \pm 0.08$ was achieved. This is a good result considering the small displacements respective to the element size in a sample with only 1% strain: The top plate moved by 3 to 4 voxels, all other parts moved even less. Applying 2% strain, the average correlation factor increased to $R^2 = 0.87 \pm 0.06$. A very similar agreement was observed for 4% strain ($R^2 = 0.87 \pm 0.07$). See table 3.4 for details on the individual samples.

Instead of looking at the displacements in the loading direction, one could as well look at the euclidean norm of the displacements. The results were almost the same, as it can be seen when comparing tables 3.4 and 3.5.

Table 3.4: Correlation factor (R^2) of the z component of the original displacement field and the one computed by the registration. Each sample was compressed at 1, 2 and 4% strain.

Sample	1%	2%	4%
1	0.76	0.9	0.93
2	0.77	0.91	0.92
3	0.8	0.91	0.9
4	0.59	0.77	0.75
5	0.72	0.85	0.84
Total	0.73 ± 0.08	0.87 ± 0.06	0.87 ± 0.07

Table 3.5: Correlation factor (R^2) of the euclidean norm of the original displacement field and the one computed by the registration. Each sample was compressed at 1, 2 and 4% strain.

Sample	1%	2%	4%
1	0.81	0.92	0.94
2	0.81	0.93	0.93
3	0.76	0.90	0.90
4	0.51	0.76	0.74
5	0.67	0.84	0.82
Total	0.71 ± 0.12	0.87 ± 0.07	0.86 ± 0.08

The linear regression analysis yielded a slope between 0.69 (for 1 and 4% strain) and 0.81 (for 2% strain). See figure 3.9 for a regression analysis of a representative sample and table 3.6 for an overview of the results of the regres-

Regression analysis of the displacements in z direction for a sample with 2% strain

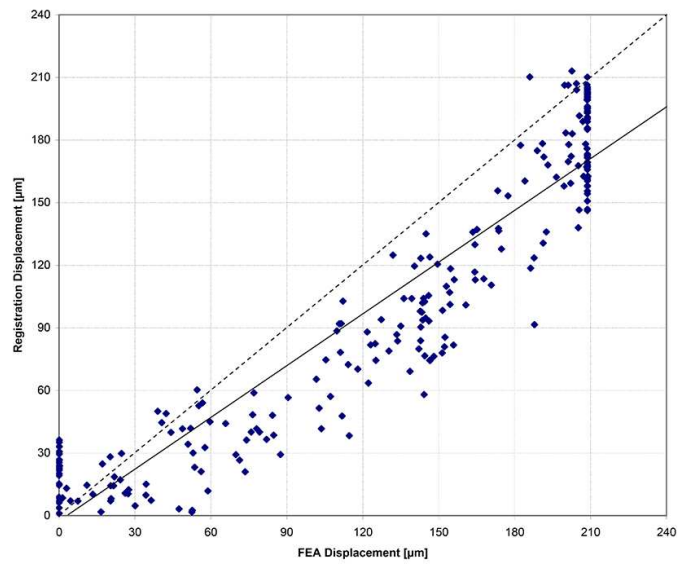


Figure 3.9: Regression analysis of the displacements in z direction using a sample compressed at 2% strain. 200 randomly chosen points within the object are plotted, while the analysis was based on the whole sample. Correlation factor measured was $R^2 = 0.91$, with a slope of 0.84 and an offset of $-5.66\mu m$. Standard deviation of the registration error was $26.4\mu m$.

sion analysis for all the samples. The slope was smaller than 1.0 because the demons registration is an iterative algorithm: The image forces trying to match the images is counteracted by an elasticity force that is distributing the effect of the image forces over a wider portion of the image. Therefore each individual step is a bit smaller than it would have to be for a perfect fit.

The slope deviating from 1.0 dramatically increased the mean error of the registration. For the samples compressed at 2% a mean error of about $20\mu m$ was observed (see table 3.7 for details). If the slope was corrected to 1.0, mean errors of only few micrometers were measured. Whereas the deviating slope did not have any influence on the standard deviation of the registration error. The fact that it can not be corrected for makes the standard deviation of the error an useful measure for the quality of the registration. While the standard deviation of the registration error increased only little from a 1% compression step to a 2% step, it almost doubled for 4% applied strain. This indicated again, that the precision of this validation approach was limited due to finite numerical precision and the resolution of the images. This limit was around $20\mu m$, just below the element size. The higher standard deviation of the error at 4% strain indicates, that the registration is less accurate for this step size.

Figure 3.10 shows how the displacements in z direction were distributed and how they were related to the mean registration error and its standard deviation. It shows that the number of voxels with low displacements was varying a lot. This was because the samples showed different behaviors when being loaded, as it was discussed in section 3.2. The top plate moved about $215\mu m$ and was counted in the second last or in the last bin, causing the fluctuation there. It showed, that the top and the bottom were registered more exactly than the center of the sample. This could have been caused by the displacements being less homogeneous or by the rather sparse material. The mean error showed a similar behavior, being more negative in the middle section of the sample.

Table 3.6: Slope/offset[μm] of the linear regression analysis of z displacements of FEA and registration for 1, 2 and 4% compressive strain.

Sample	1%	2%	4%
1	0.81/5.62	0.78/5.89	0.72/5.37
2	0.70/4.21	0.83/2.43	0.69/4.14
3	0.74/ - 1.27	0.84/ - 5.66	0.68/ - 4.52
4	0.66/3.81	0.83/ - 9.30	0.64/ - 2.03
5	0.67/1.84	0.80/ - 2.41	0.71/ - 4.68
Total	0.72 $\pm$ 0.06 /2.84 $\pm$ 2.67	0.81 $\pm$ 0.03 / - 1.81 $\pm$ 6.09	0.69 $\pm$ 0.03 / - 0.34 $\pm$ 4.79

Already after few iterations (10 to 30) the correlation factor reached an acceptable level (see figure 3.11), while it took more iterations for the standard deviation of the registration error to drop below the element size. It was found however, that convergence was much faster when grayscale images were being used.

Table 3.7: Mean error ($\pm$ standard deviation of the error) [μm] for the registration of z displacements for 1, 2 and 4% compressive strain.

Sample	1%	2%	4%
1	$-1.44(\pm 22.33)$	$-11.07(\pm 30.86)$	$-37.18(\pm 61.16)$
2	$-8.93(\pm 21.73)$	$-12.64(\pm 27.13)$	$-50.46(\pm 65.56)$
3	$-15.59(\pm 19.20)$	$-23.02(\pm 26.40)$	$-75.04(\pm 66.44)$
4	$-21.50(\pm 21.30)$	$-35.39(\pm 32.10)$	$-108.88(\pm 67.43)$
5	$-17.50(\pm 22.41)$	$-26.64(\pm 32.90)$	$-73.43(\pm 71.19)$
Total	-12.99 ± 7.90 (21.39 ± 1.31)	-21.75 ± 10.10 (29.88 ± 2.95)	-69.00 ± 27.40 (66.36 ± 3.61)

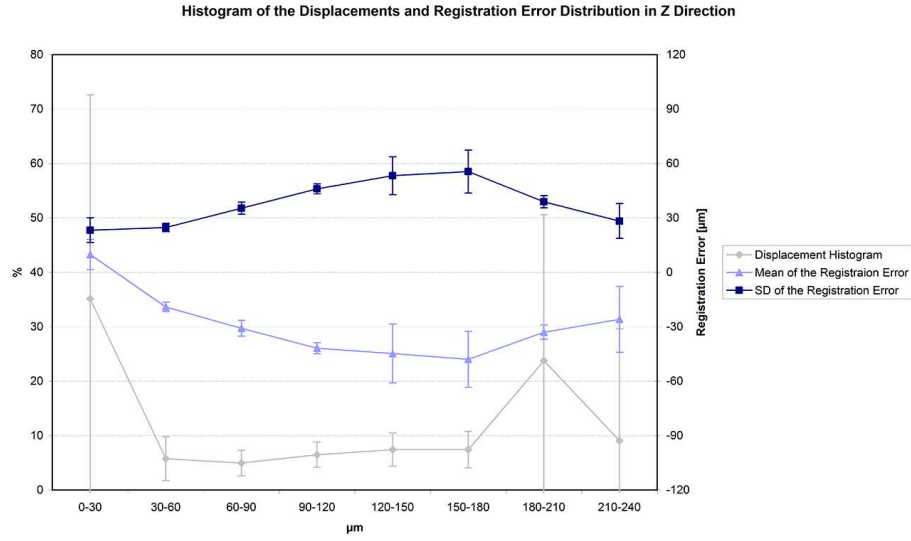


Figure 3.10: Histogram of the displacements in z direction originally computed by FEA (primary y axis) and a comparison of the mean and standard deviation of the registration error for z displacements (secondary y axis). The plot shows the results for all the samples at 2% strain. A bin size equal to the element size has been used.

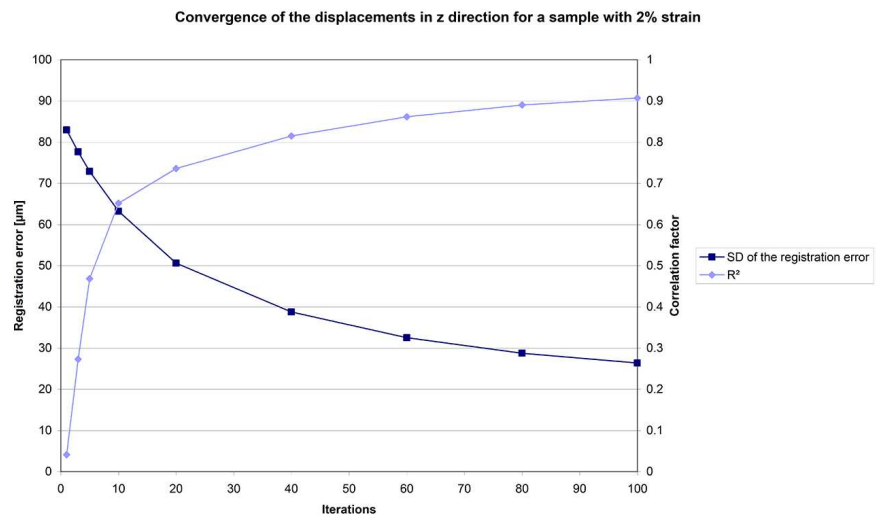


Figure 3.11: Standard deviation of the registration error and correlation factor between the original and the registration displacements in z direction for different numbers of iterations. This plot for a sample with 2% applied strain shows an exponential decrease towards a noise level below the element size while the correlation factor approaches 1.0 .

3.4.2 Displacements perpendicular to the loading direction

Displacements in x and y directions were very small: For 1% applied strain mean displacements were a few micrometers and standard deviations were around $15\mu m$, that is half of the element size. Even for 2% strain, standard deviation of the displacements just reached the element size. As binary images had to be used for this validation the registration could not be more precise than the element size. Therefore most of the displacements were covered in the numerical noise, mainly introduced by the warping process. In those cases correlation is not a significant measure for the accuracy of the registration of samples with small strain. For completeness, they were determined anyways: At 1% strain correlation factors of R^2 between about 0.1 and 0.2 were obtained. At higher strains, with x and y displacements becoming larger correlation factors increased as well. At 2 and 4% strain R^2 between about 0.3 and 0.5 had been measured (see figure 3.12). But the results for the individual samples were varying a lot.

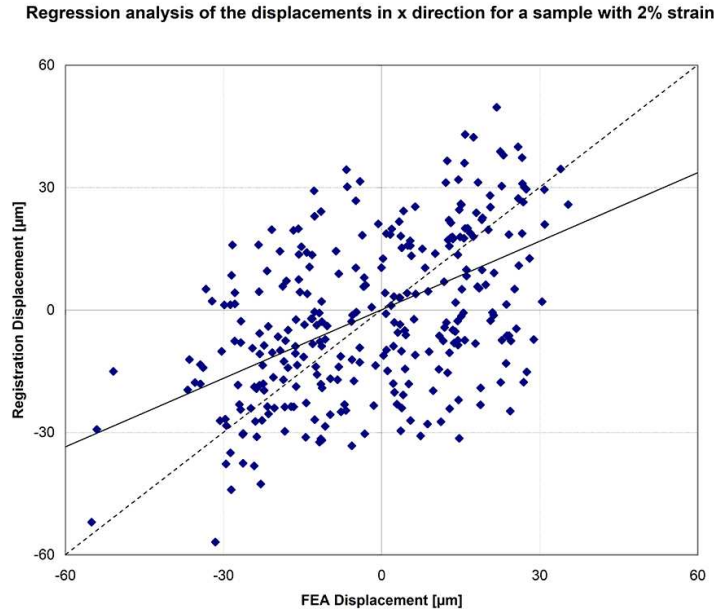


Figure 3.12: Regression analysis of the displacements in x direction of a sample compressed at 2%. Correlation factor measured was $R^2 = 0.29$, with a slope of 0.56 and an offset of $-0.55\mu m$. Standard deviation of the registration error was $18.4\mu m$.

One reason for the varying results was that displacements in x and y direction were highly stochastic (figure 3.13). They were also not necessarily normally distributed as one could assume.

In contrast to the displacements, the registration error closely resembled a

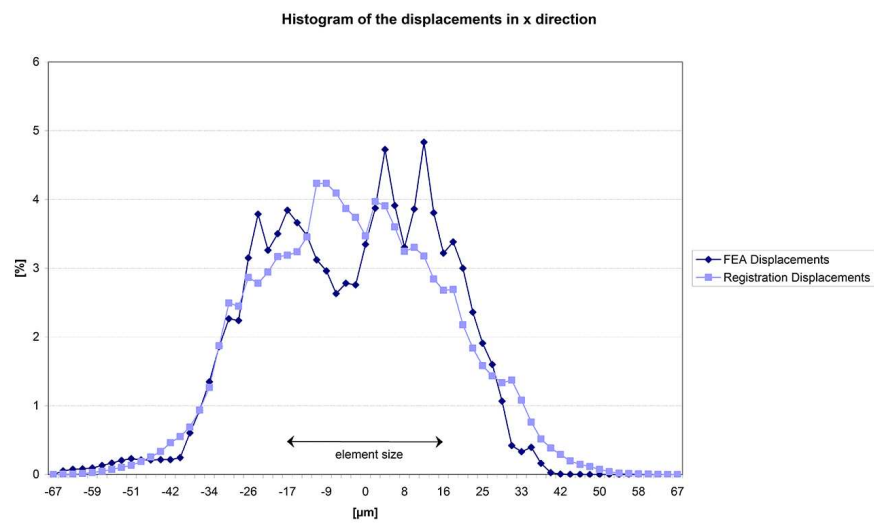


Figure 3.13: Typical histogram of the displacements in x direction of a sample loaded at 2% compressive strain.

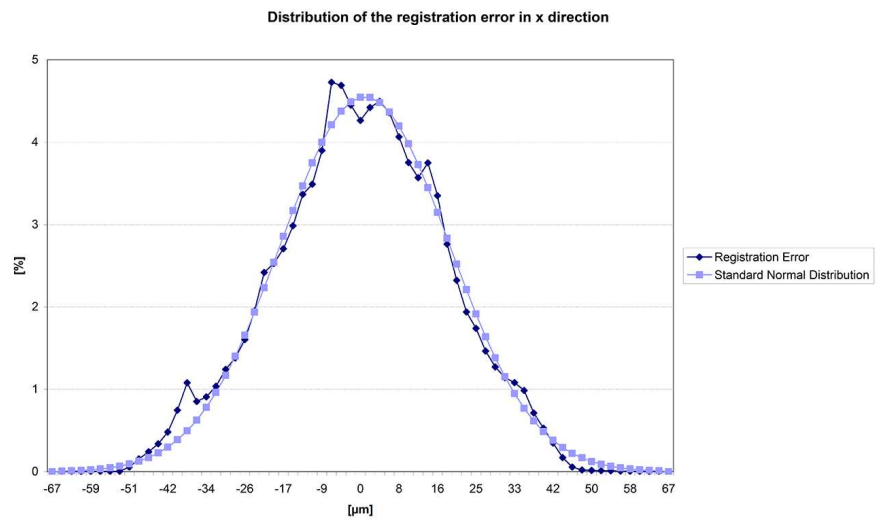


Figure 3.14: Typical Histogram of the registration error in x direction for a sample loaded at 2% compressive strain. The distribution shows good agreement with a normal distribution with $mean = 0.6\mu m$ and $SD = 18.4\mu m$.

normal distribution with a mean of a few μm and a standard deviation smaller than the element size for 1 and 2% strain. For 4% strain, the standard deviation of the registration error increased to 1.5 times the element size. The normal distribution with a small mean compared to the standard deviation indicates a stochastic cause for the registration error, i.e. the limit of the resolution. See figure 3.14 for the distribution of the registration error in x direction of a representative sample. Tables 3.8 and 3.9 show the mean registration error and its standard deviation for the displacements in x and y direction respectively for all the samples.

Table 3.8: Mean error ($\pm$ standard deviation of the error) [μm] for the registration of x displacements for 1, 2 and 4% compressive strain.

Sample	1%	2%	4%
1	-1.79(± 13.53)	-3.21(± 17.87)	-6.68(± 30.41)
2	-0.27(± 16.03)	-4.14(± 22.24)	-13.26(± 44.40)
3	0.97(± 16.49)	0.57(± 18.41)	3.24(± 29.53)
4	-1.01(± 17.20)	-4.17(± 21.35)	-9.62(± 34.94)
5	4.17(± 24.03)	8.62(± 41.84)	27.28(± 91.14)
Total	0.41 ± 2.33 (17.46 ± 3.93)	-0.47 ± 5.44 (24.34 ± 9.96)	0.19 ± 16.33 (46.08 ± 25.87)

Table 3.9: Mean error ($\pm$ standard deviation of the error) [μm] for the registration of y displacements for 1, 2 and 4% compressive strain.

Sample	1%	2%	4%
1	3.68(± 14.38)	7.88(± 20.55)	16.21(± 39.14)
2	-3.87(± 14.25)	-7.45(± 18.04)	-19.96(± 31.22)
3	-8.61(± 18.59)	-17.16(± 25.39)	-43.59(± 51.57)
4	2.57(± 16.57)	3.69(± 19.94)	9.51(± 33.38)
5	-2.25(± 17.67)	-4.28(± 24.10)	-12.53(± 44.34)
Total	-1.70 ± 5.00 (16.29 ± 1.94)	-3.47 ± 9.80 (21.60 ± 3.04)	-10.07 ± 23.99 (39.93 ± 8.28)

3.5 Discussion

As binary images had to be used for compatibility with FEA, a registration with sub voxel precision is very difficult. Even theoretically this is only possible using some knowledge about how the image could have deformed, i.e. an elastic material where neighboring points can not move independently. This finite precision of the registration combined with the round-off error introduced by the warping process, led to a standard deviation of the registration between half the element size and the element size.

Displacements in loading directions were much more significant for the behavior of the samples than displacements in x and y direction. Demons Deformable Registration showed to be able to reliably recover these displacements from the images. The applied deformations and the ones computed by the registration correlated with $R^2 = 0.87 \pm 0.06$ for compression steps of 2% strain and deviating only little even though the samples did not show a homogeneous behavior. The high correlation also confirms that the displacements computed by the registration are physically meaningful and not just any displacements that align the two images. Compression at 2% overall strain yielded optimal results, while also at 1 and 4% strain good and reliable results for the correlation were achieved ($R^2 = 0.73 \pm 0.08$ and $R^2 = 0.87 \pm 0.07$ respectively). The lower correlation factors in samples loaded with only 1% strain was most likely caused by the many displacements covered in the noise floor of forward warping the binary images. The doubling of the standard deviation of the registration error for 4% applied strain compared to 2% strain indicated a decreasing accuracy of the registration of such big compression steps.

Instead of the displacements in loading direction, the euclidean norm of the displacements had been evaluated as well. The results were essentially the same as in z direction. This was not very surprising, as the displacements in loading direction were much bigger than in the other directions and therefore they had a dominating influence on the euclidean norm.

In a uni-axial compression test, displacements perpendicular to the loading direction could only be computed from grayscale images with a high resolution as it will be shown in chapter 4. In the setting used in this chapter, x and y displacements were too small compared to the noise that was introduced by segmenting the images and the forward warping.

Chapter 4

Validation of the demons deformable registration using experimental displacements

As a next step after the Demons Deformable Registration was tested with digitally deformed images, we applied the Demons Deformable Registration to experimental data. In a first step of an Image Guided Failure Assessment (IGFA) experiment [1], the samples already used above were compressed at 1%. The two images of the two steps (0 and 1% compressive strain) were used as inputs for the registration, while an FE analysis based on the unloaded image and with the same boundary conditions as the experiment was done. This time, the displacements computed by FEA served as a reference, rather than as a gold standard. An agreement of the displacement fields computed by FEA and the registration would validate the displacement fields computed by the Demons Deformable Registration for IGFA experiments at small strain. Furthermore it would also show that linear elastic FE analysis predicts correct displacements for uni-axial compression test at up to 1% strain. As described in 3.2 the results of the linear elastic FEA are only valid for the first compression step, therefore subsequent compression step with higher strains could not be used for this comparison.

For the subsequent compression steps of the IGFA experiment, when 2, 4, 8 and 16% overall strain was applied to the sample, the correct registration was checked visually, as there was no reference available.

FEA and the registration again yielded two displacement fields. Similar to the previous section, the two displacement fields were compared to each other (See figure 4.1). But now it was not clear anymore, that the results from the FEA would match the experimental results. Although an approach to the validation of FEA has been presented already [17], it was not clear, that FEA

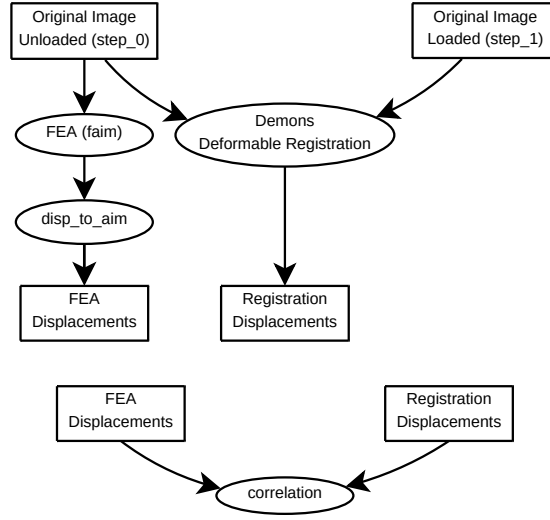


Figure 4.1: Flow Chart of the Experimental Approach

would predict correct displacements, for instance because of rotations caused by off-center loading or other experimental artifacts. Therefore in addition to the statistical analysis of the similarity of the displacement fields, it was visually assessed which displacement fields were accurate: The μCT image of the unloaded sample was warped using the displacement fields from FEA and the registration receptively. These two artificially deformed images were then compared to the image of the deformed sample from the experiment.

The statistical analysis was based on the displacements in z direction. Compared to a validation approach using artificial displacements, an experimental setup is much more difficult to be controlled precisely. As displacements in x and y direction were already very difficult to analyze in the artificial environment used in the last chapter, it would not make sense to do a statistical analysis on them here.

As greyscale images could be used for the registration, the parameters of the registration were chosen differently from before. Gaussian filtration was not necessary anymore and 60 Iterations were sufficient as the elasticity parameter was chosen lower as well ($d = 5.0$). The images were rather noisy (they were Gaussian filtered with $\sigma = 1.2$) so that a lot of noise was also in the displacement fields. A stronger low pass filtering of the images was omitted, as this would have removed very small features from the images. The displacements were Gaussian filtered with $\sigma = 5.0$ and a support of 10 voxels. Such a strong low pass filtration helps reproducing the physical displacements, as they do not vary a lot in elements close to each other. Figure 4.2 shows an example displacement field in the subsequent processing steps.

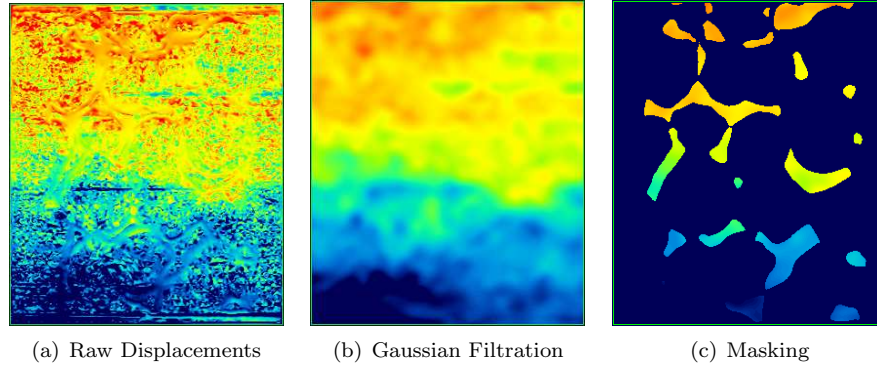


Figure 4.2: The processing steps of the z displacements from the registration of μCT images. 1% compressive strain was applied to the sample. The raw displacements computed by the Demons Deformable Registration are low pass filtered and masked using a segmented version of the image of the unloaded sample.

4.1 Results

It showed that for some samples the displacements from FEA and the registration were very similar, while for other samples, rather large deviations occurred. Of the 5 samples used in this study, two showed very similar displacement fields ($R^2 \approx 0.9$) while for two other sample correlation was good as well ($R^2 = 0.71$ and 0.62). In one sample the correlation was low (sample no. 5: $R^2 = 0.44$). See table 4.1 for details.

Table 4.1: Statistical analysis of the z component of the displacement fields computed by FEA and the registration.

Sample	R^2	Regression		Difference	
		slope	offset μm	Mean μm	SD μm
1	0.62	0.44	38.24	11.97	20.97
2	0.88	0.71	4.56	-10.02	13.53
3	0.93	0.79	1.62	-10.50	10.00
4	0.71	0.81	-45.45	-59.84	13.10
5	0.44	0.23	-52.04	-101.18	26.24

The results of the regression analysis show that for some samples there was a remarkable offset. This indicates that the sample was not exactly aligned in z direction when the first and the second image was taken. The correct alignment was often difficult to check and only in sample 5 a bad alignment was obvious. But the poor alignment and subsequently the offset from the regression analysis do not explain the lower correlation factors in those samples. The correlation is a second order moment and is corrected for deviating means and variances in the two populations. So that it has to be assumed, that also other experimental

artifacts played a role and that those effects were bigger, the worse the alignment was.

The differences between the z components of two displacement fields were rather small. The standard deviation of the difference between the two displacement fields were always smaller than the element size, while the mean difference was greatly enhanced by the alignment problems, as it behaved similarly to the offset of the regression analysis.

First the agreement of the computed displacement fields with the displacements that actually occurred in the experiment was tested visually: The FEA and registration displacements were applied to the μCT image of the unloaded sample and the result was compared to the image obtained from the experiment by scanning the sample at 1% applied strain. The artificially deformed and the real objects were then rendered and overlaid into a common 3D scene.

The FEA displacements were applied using forward warping, while the displacement field from the registration was applied directly to the surface mesh of the unloaded model. The error introduced by forward warping is bigger than applying the displacements directly to the surface mesh. But the FEA displacements could not be easily applied to the mesh: The vertices of the mesh are sometimes within the model and sometimes outside, because of finite numerical precision and because the meshes had to be simplified before they could be rendered. Because the displacement field from FEA is zero outside the model no meaningful displacement could be assigned to the vertices. But the rounding error involved in forward warping can only reach half of the element size.

In the following a deeper insight into the sample with the highest and the one with the lowest correlation factor will be given.

In the overlay renderings, the image of the physically compressed sample was rendered white. The artificially deformed sample, that is the unloaded sample warped by the displacements computed by FEA and the registration respectively, were overlaid as a blue colored object. Figure 4.3 shows the two renderings for sample 3, the one with the best agreement between FEA and registration. Both displacement fields were close to the actual displacements occurring. The object warped using the displacements from the registration showed an almost perfect fit with the image from the experiment (see figure 4.3(b)). It was random if the surface of the warped object was just outside the experimental sample or a bit inside leading to the speckled surface. This little variation can hardly be prevented and usually is not significant. The main reason for the poorer agreement of the object warped by FEA displacements were the rotations that occurred in the experiment. Also when the FEA displacements were being used for the warping (see figure 4.3(a)), a speckled surface can be seen, but only in the lower part of the sample. The upper section of the sample rotated slightly during the compression test, as it could be seen from the slice images of the μCT data and the x and y displacements from the registration. This was an experimental artifact that FEA could not compensate for.

The regression analysis confirmed the good agreement of displacements computed by FEA and the registration (see figure 4.4). Differences were very small, as it also can be seen from the low standard deviation of the difference between

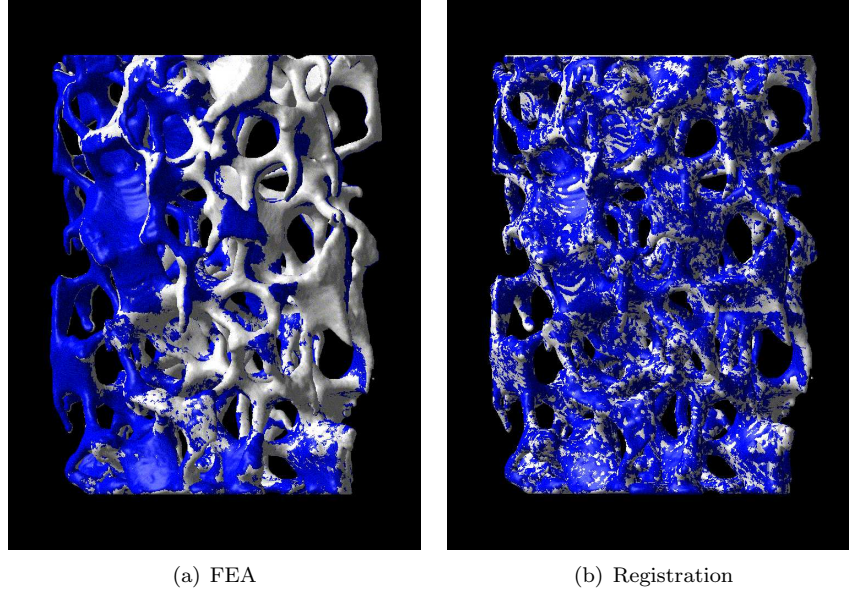


Figure 4.3: Overlay renderings for the sample with high correlation between the displacement fields. White is the physically compressed sample, blue is the unloaded sample warped by the displacements computed by FEA and registration.

FEA and registration of only $10.00\mu m$. It is difficult to explain the deviating results close to the top of the sample (in the top right corner of figure 4.4): It could be another experimental artifact or that the sample really behaved differently than FEA predicted. But in the overlay rendering in figure 4.3(b) it can be seen that the registration computes correct displacements also for the top of the sample.

Sample number 5 featured the lowest correlation between displacements computed by FEA and the registration ($R^2 = 0.44$). When the same overlay renderings as in figure 4.3 were done, it showed that warping the unloaded sample with the displacements computed by the registration again led to an object that agreed well with the sample loaded in the physical experiment (see figure 4.5(b)). Although the agreement is not quite as good for the sample 3, it still acceptable and the errors are small.

Contrary to the displacements computed by the registration, the displacements from the FEA only poorly reproduced the deformations that the sample underwent in the experiment (see the overlay rendering in figure 4.5(a)). The (blue) warped object is significantly displaced respective to the experimental data. Looking at the μCT images of the loaded and the unloaded sample reveals a very complicated deformation. It is as if the whole sample was bending slightly instead of being compressed homogeneously. This can happen when the plates on top and bottom of the sample are not exactly parallel in the beginning. They usually align during the first compression step, leading to bending in the

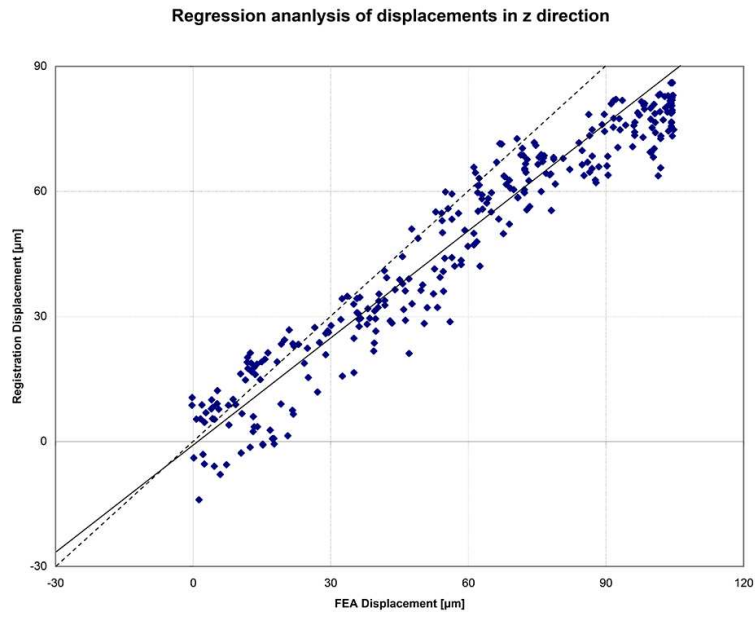


Figure 4.4: Regression analysis for the sample with good agreement between displacements computed by FEA and the registration. For the plot, 200 randomly chosen points within the sample were chosen, while the statistical analysis was based on all the points of the object. $R^2 = 0.93$, an offset of $1.62\mu m$ and a slope of 0.79 was measured.

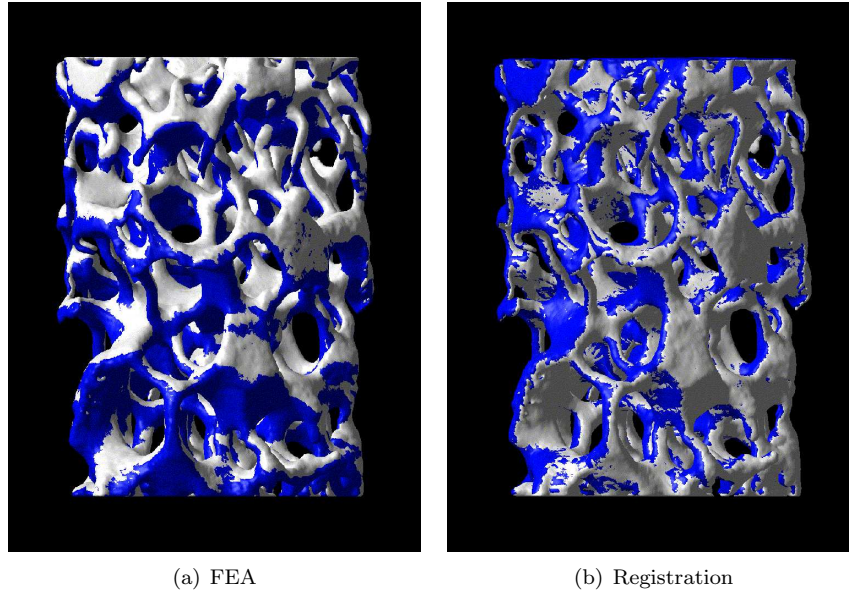


Figure 4.5: Overlay rendering for the sample with low correlation between the displacement fields.

sample. On top of that, the two scans (before and after the sample was loaded) were not taken at the same position in z , leading to a poor alignment. This can also be seen in the big offset of almost twice the element size that the regression analysis of the displacements from FEA and the registration yielded. From the μCT images itself, it was very difficult to determine the correct alignment due to the complex deformations. In this example hardly any rotation occurred.

The deviating results were also observed in the regression analysis (see figure 4.6). It showed a correlation of $R^2 = 0.44$ and a standard deviation of the difference between FEA and registered displacements of $26.24\mu m$, that is more than twice as much as it was determined for the sample with the best agreement. The large negative offset is mainly caused by the alignment problem discussed above and led to the negative displacements computed by the registration. Apart from this, the low slope of 0.23 showed that the experiment was not exactly reproducing an axial compression test with 1% applied strain.

In order to quantify the agreement between the sample deformed by the registration displacements and the loaded sample. The number of voxels, where the two samples did not fit each other. For this analysis the segmented image of the unloaded sample was forward warped by the displacements computed by the registration. This warped image was then subtracted from the image of the deformed sample. The result was an image with a value of zero in most of the voxels. But the value was positive, if the warped sample was outside the sample, and negative, if the warped sample did not cover the whole extent of

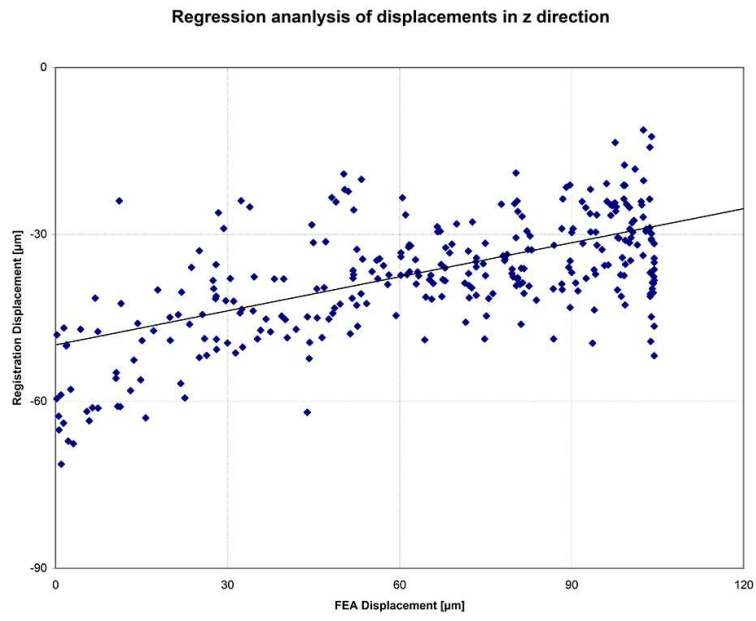


Figure 4.6: Regression analysis for the sample with poor agreement between displacements computed by FEA and the registration. $R^2 = 0.44$, an offset of $-52.04\mu m$ and a slope of 0.23 was measured.

the sample. The coefficient of the voxels deviating from zero divided by the total number of voxels in the sample, indicated the quality of the agreement. In this coefficient, most of the registration errors are counted doubled, as they usually yield a positive and a negative voxel. For the samples, where a speckle pattern could be observed in the overlay renderings, this coefficient was around 0.1. For all five samples a coefficient of 0.11 ± 0.04 was computed, meaning that less than 10% of the voxels of the samples are wrongly aligned. Considering the round-off error introduced by the forward warping, this is a very good result and confirms the visual observations.

In many of the samples, slight rotations could be observed, that led to x and y displacements much bigger than the displacements that were computed by FEA. Already a rotation of 0.5 degrees leads to displacements in the order of the element size in the outer parts of the sample (assuming a diameter of 200 voxels) which is already more than twice the average standard deviation of the x and y displacements computed by FEA for the 5 samples (see section 3.2 for details on the FEA results). The exact amount of rotation could be calculated from the displacement fields computed by the registration. But it is not likely that one could fully compensate for the rotation in order to compare the x and y displacements from FEA and the registration. The displacements from FEA were probably too small for that.

4.2 Applying the registration to bigger compression steps

In order to ensure the precision of the registration also for bigger compression steps than 1% applied strain, the next compression steps were being analyzed. The samples loaded at 2, 4 and 8% were registered to the image of the unloaded sample. The accuracy of the registration for those loading steps could only be visually assessed, as there was no reference one could compare the displacements to. Linear elastic FEA is only valid, as long as linear mechanical effects are dominating in the sample being loaded. But the experimental data showed that this only holds for first compression step (1% strain).

The registration was done using 80 Iterations. Images were only minimally Gaussian filtered ($\sigma = 1.2$). The elasticity parameter was chosen slightly lower ($d = 3.0$). After the registration, the displacements were Gaussian filtered using $\sigma = 5.0$. For the biggest compression step, that is the registration of step 0 and step 4 (loaded at 8% applied strain), also increasing the number of iterations to 100 did not lead to a better convergence of the registration.

The overlay renderings were done as in the previous section. In order to simplify the assessment of the accuracy of the registration, an image showing the sample in the loaded (white) and unloaded (blue) state is given on the left. This visualizes the starting position of the registration. As above, the image on the right shows the unloaded sample warped by the displacements computed by the registration (blue) and the loaded sample (white).

The renderings showed that the registration was accurate until a compression step of 4% overall strain. It could not handle a compression step of 8%. The registration was not converging, because the initial guess, that is the image of the unloaded sample, was not close enough to the image of the loaded sample. It showed that also in the 4% step, some small features in the upper part of the sample were not registered correctly, while all the bigger structures were aligned. The left side of the middle part of the same rendering shows how well the algorithm can cover non linear effects such as buckling and bending. Also rotations showed to be recovered correctly.

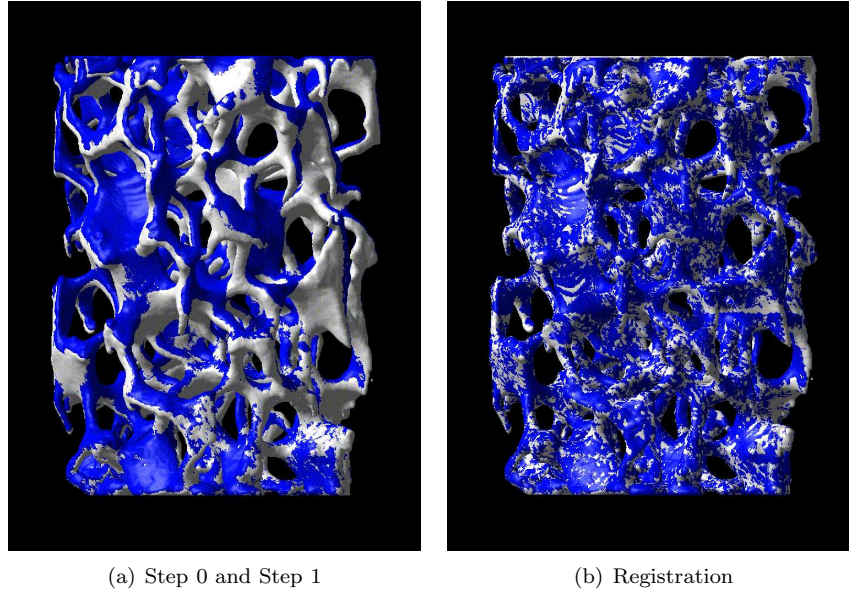


Figure 4.7: Overlay renderings for a sample compressed at 1% overall strain. White is the loaded sample. In the left image, the loaded sample is overlaid with the unloaded sample (blue), in the right image the mesh of the unloaded sample is warped by the displacement field computed by the registration.

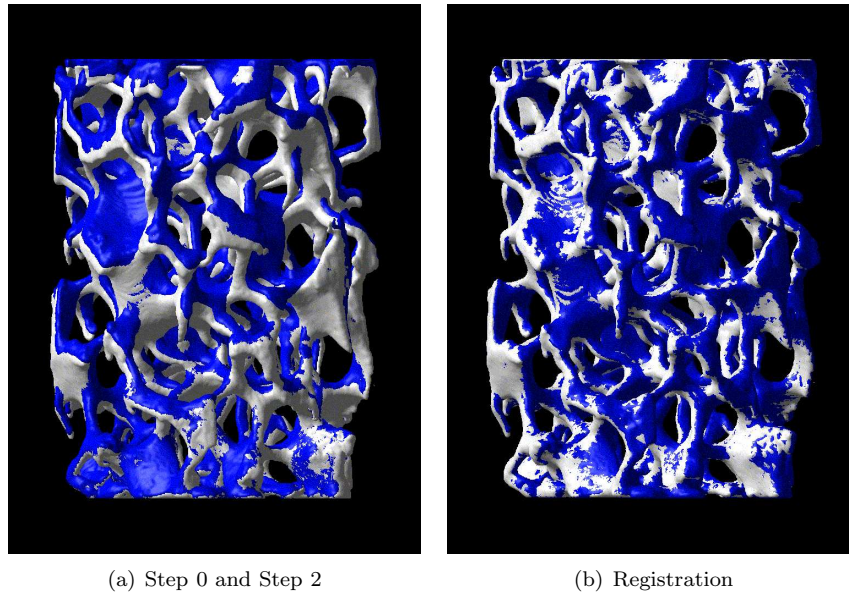
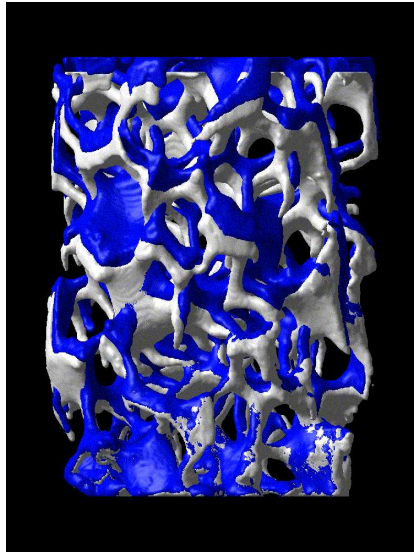
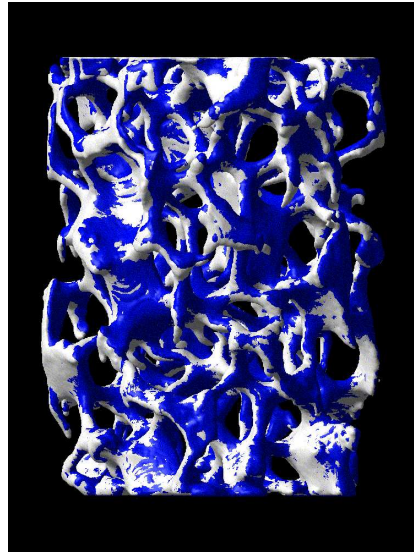


Figure 4.8: Overlay renderings analogous to 4.7 for a sample compressed at 2% overall strain.

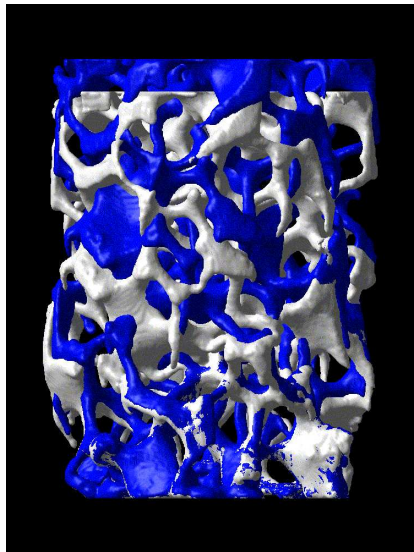


(a) Step 0 and Step 3

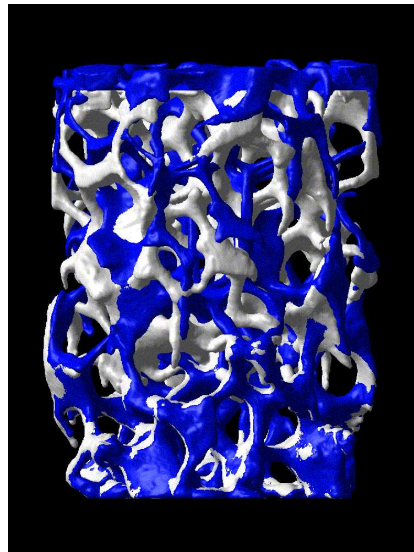


(b) Registration

Figure 4.9: Overlay renderings analogous to 4.7 for a sample compressed at 4% overall strain.



(a) Step 0 and Step 4



(b) Registration

Figure 4.10: Overlay renderings analogous to 4.7 for a sample compressed at 8% overall strain.

4.3 Applying the registration to trabecular bone

The registration was also tested to yield accurate results for trabecular bone. Compared to the aluminum foam used previously, the structure of trabecular bone strongly varies. It can be rod-like as low density aluminum foam or rather plate-like and any combination of the two. It had to be ensured, that the registration would yield correct results independent of the structure.

It was assumed, that on the one hand, additional features within the bone tissue could possibly increase the precision of the registration, whereas on the other hand, the registration would become less reliable in coarse plate-like structures due to the lack of features.

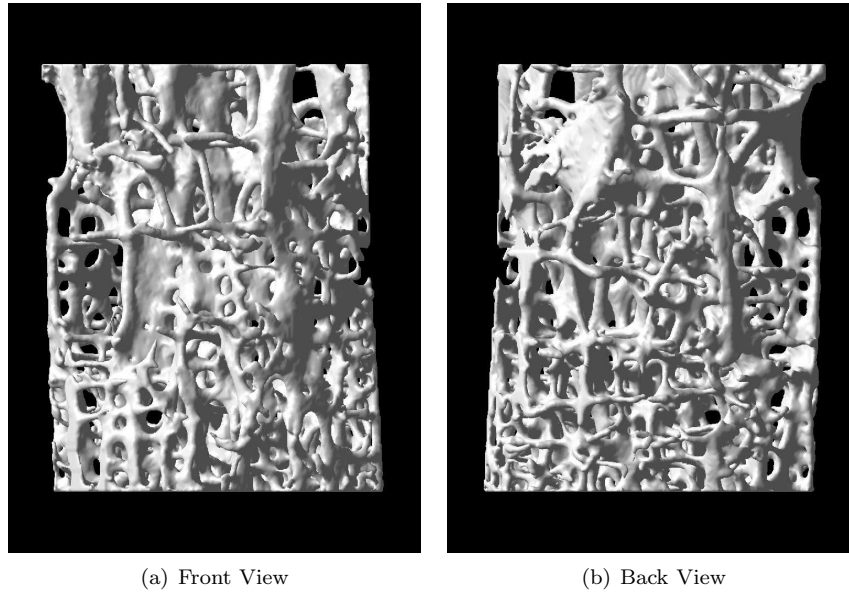


Figure 4.11: The unloaded sample of trabecular bone seen from two sides. There are plate-like regions, as well as regions, where rod-like structures are dominating.

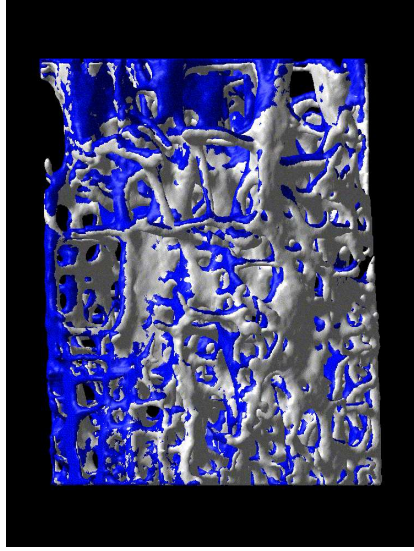
A sample of a human vertebra (L3) was used, as it showed rod-like as well as plate-like structures. The sample was 10.6mm long and 8mm in diameter. It was scanned at 0, 2, 4 and 8% applied compressive strain, always at a resolution of $34\mu\text{m}$. The registration was run using 100 Iterations and an elasticity parameter of 5.0. As the greyscale images were used for the registration, no additional Gaussian Filtration of the images was required.

As in the last section, the correct registration could only be visually assessed. The same overlay renderings were generated as in the last section, but because of the inhomogeneity of the sample, it is always showed from the front and from the back.

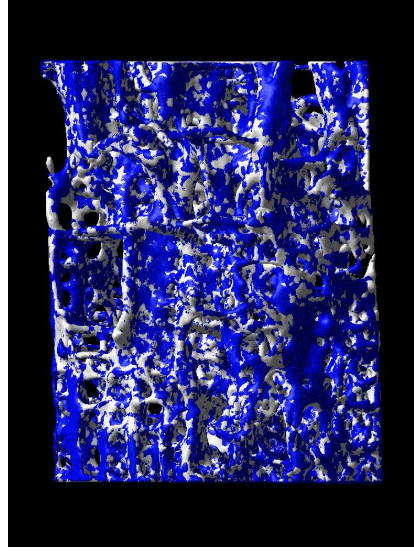
The renderings in figure 4.13 show an excellent agreement of the object warped by the registration displacements for the sample compressed at 2%. In

figure 4.12(a) it can be seen, that plate-like regions are moving slightly less than the rod-like regions. But the different structures have little impact on the performance of the registration.

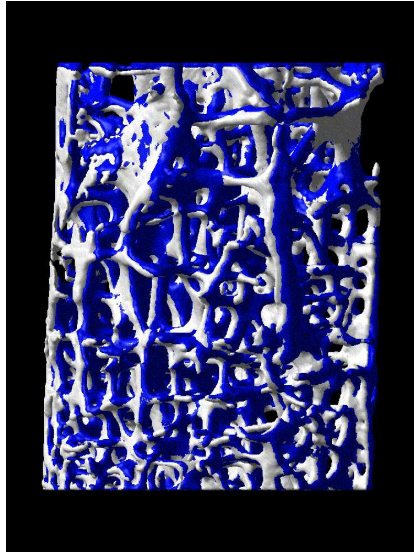
When a compression step of 4% was registered (see figure 4.13), the resulting displacement field did not match the actual deformations of the sample. Especially the top of the sample was poorly registered. When the elasticity parameter d was reduced to 3.0, generally leading to a faster convergence, the results improved. The region of good alignment became bigger, but the top of the sample still was not properly aligned.



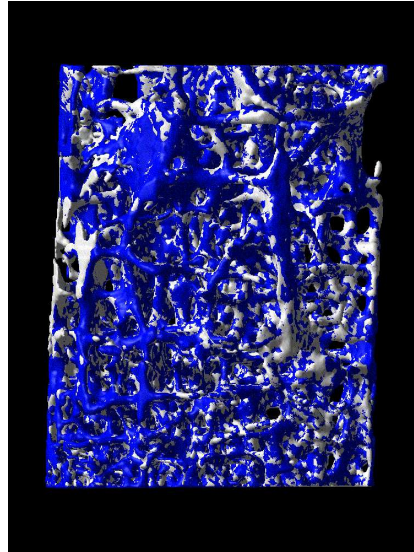
(a) Front View: Step 0 and Step 1



(b) Front View: Registration

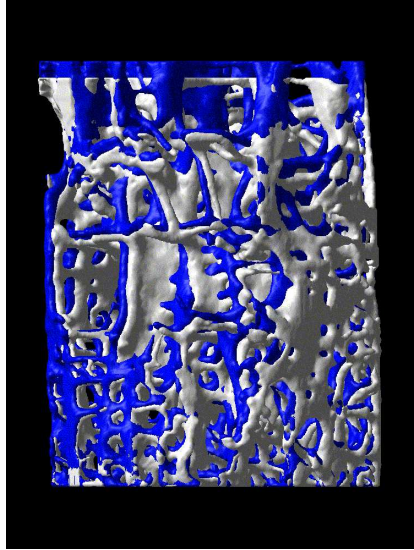


(c) Back View: Step 0 and Step 1

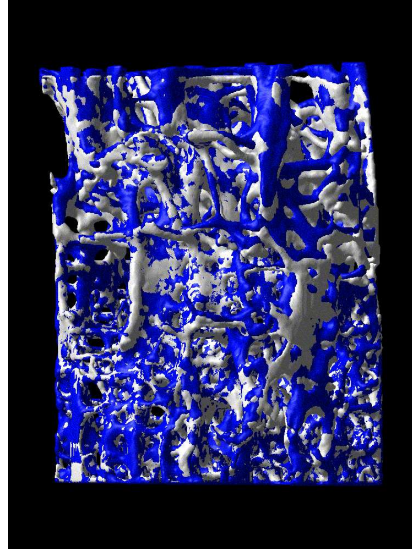


(d) Back View: Registration

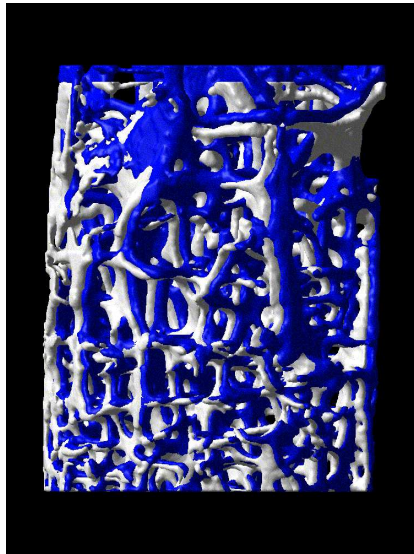
Figure 4.12: Overlay renderings for the trabecular bone sample compressed at 2% overall strain. White is the loaded sample. In the left image, the loaded sample is overlaid with the unloaded sample (blue), in the right image the mesh of the unloaded sample is warped by the displacement field computed by the registration.



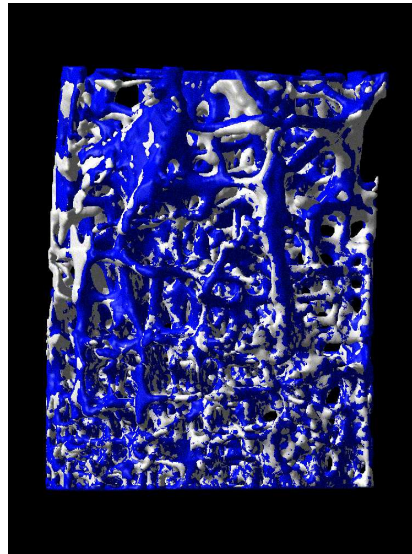
(a) Front View: Step 0 and Step 2



(b) Front View: Registration



(c) Back View: Step 0 and Step 2



(d) Back View: Registration

Figure 4.13: Overlay renderings for the trabecular bone sample compressed at 4% overall strain. White is the loaded sample. In the left image, the loaded sample is overlaid with the unloaded sample (blue), in the right image the mesh of the unloaded sample is warped by the displacement field computed by the registration.

4.4 Discussion

Applying the Demons Deformable Registration to the greyscale images obtained in IGFA compression test yielded displacements that accurately reproduce the deformations the samples underwent in the test. This was done visually: From the images of the unloaded and the compressed sample two surface meshes were generated. The displacements computed by the registration were applied to surface mesh of the unloaded sample and the resulting mesh was rendered in a common 3D scene with the mesh from the loaded sample. The registration showed to be very resistant to experimental artifacts such as unwanted rotations, poor alignment of the subsequent μCT scans and other unwanted effects.

It also showed that linear elastic FEA predicts correct displacements for small compressive strains ($\epsilon \leq 1\%$). 2 of the 5 samples showed very good agreement between FEA and the registration ($R^2 = 0.9$). In those samples both methods described the correct deformations. In the other samples the experiments did not reproduce a standard compression test. It became clear that IGFA experiments at low strains are very difficult to conduct precisely enough so that the displacements from the experiment could be directly compared to the results from FEA.

Displacements in x and y direction from FEA and the registration could not be compared to each other in this setup. The displacements computed by FEA were very small (less than the element size) so that images at higher resolutions would have been necessary. Furthermore most of the samples were slightly rotating when being loaded, mostly because of off-center loading. These rotations were not considered in the FE analysis while the registration computed correct displacements in x and y direction also if rotation was involved in the compression step.

The registration yielded accurate displacements for loading steps up to 2% overall strain. The visual inspection showed that the displacements from registering loading steps of 1% reproduced the actual deformations better than when larger loading steps were registered. This contradictory result to section 3.4 indicates that numerical errors were reduced. In loading steps of 4% strain, some smaller features with big displacements were not correctly aligned.

When trabecular bone was used instead of aluminum foam, the registration showed to yield displacements that reproduced the deformations of the sample at least as good as with aluminum foam. There was no decrease in precision observed for plate-like regions of the trabecular bone. Also for trabecular bone, the registration was accurate for up to 2% strain steps. The registration of a 4% compression step did not converge.

Chapter 5

Validation of local strain mapping

In this study local strains were computed similar to the original publication of DVC [6]. This approach was already implemented in a previous Semester Arbeit by R. Jost [18]. Similar to the present study, this project aimed at the computation of local strains based on displacements computed by Demons Deformable Registration. But there was an incompatibility issue with the floating point number format, so that the proper functioning of the code could not be tested within the Semester Arbeit. For this validation study, the application was completely rewritten because of run-time errors in the original implementation and for a simplified maintenance.

5.1 Method

The computation of the local strain map was done in 4 steps:

- Gaussian filtration of the displacements
- Computing local gradients in the displacement field
- Computing the local Green Lagrange strain tensor
- Computing principal strains and Von Mises strain

Computing derivatives always dramatically amplifies small errors. Because of the finite precision of the computation there are always some small inaccuracies in the data. In order to reduce the impact of this amplification, the data always should be low-pass filtered before the gradients are computed. In the implementation that was used here, the displacements were first Gaussian filtered with $\sigma = 5.0$ and a support of 10 voxels, before gradients were computed, based on the central difference with a step size $h = 2$ voxels.

$$\partial f[x] = \frac{f[x+h] - f[x-h]}{2h} \quad (5.1)$$

There are other algorithms to compute local gradients: The Difference of Gaussian Filter (DOG) combines the computation of the gradient and the smoothing with a Gaussian function into one operation. The images are smoothed with the derivative of a Gaussian function. The derivative of a Gaussian function can be written as two identical Gaussian functions, one shifted to the left and changed in sign, the other shifted to the right. Another possibility are polynomial filters. They compute a least square fit of a higher order polynomial. The linear term of the fitted polynomial can then be used as the gradient. All three approaches produce very similar results.

At each coordinate $\vec{x} = (x_1, x_2, x_3)^T$, three values for the three components of the displacement vector $\vec{u} = (u_1, u_2, u_3)^T$ were computed by the registration. Therefore 9 gradients ($\partial u_i / \partial x_j$) had to be computed. So that 9 runs of differently oriented DOG filters or polynomial filters would have been necessary. This makes the combination of the filtering and the computation of the derivatives less interesting.

Strain ε usually is defined as the change in length $\Delta L = L_1 - L_0$ divided by the original length L_0 [19].

$$\varepsilon = \frac{L_1 - L_0}{L_0} \quad (5.2)$$

This definition is only valid for small strains (typically $< 2\%$) as long as the change in volume can be assumed to be negligible. This did not hold in our case: Overall strains of up to 4% were applied and previous studies showed that local strains can reach 8 times the applied strain [20]. Therefore a second order formulation for strain as defined by Lagrange had to be used.

$$\varepsilon = \frac{L_1^2 - L_0^2}{2L_0^2} \quad (5.3)$$

Both formulations can as well be computed based on local changes in the displacement field [21]. If strain was assumed to be small, the strain tensor could be linearly approximated:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{du_i}{dx_j} + \frac{du_j}{dx_i} \right) \quad (5.4)$$

The Lagrange formulation for strain results in the Green Lagrange strain tensor. (Note the symmetry of the tensor)

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{du_i}{dx_j} + \frac{du_j}{dx_i} + \frac{du_k}{dx_i} \frac{du_k}{dx_j} \right) \quad (5.5)$$

As it can be seen from formula 5.4 and 5.5 assuming small strain is equivalent to neglecting higher order terms of the strain formulation.

Von Mises strain was used in order to locate regions of strain concentration and to make the strain map comparable to FEA. A formulation similar to the formulation of Von Mises stress was used:

$$\varepsilon_{vm} = \sqrt{\frac{(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2}{2}} \quad (5.6)$$

With ε_i being the three principal strains or equivalently the eigenvalues of the strain tensor. Therefore the eigenvalue problem of the strain tensor had to be solved first. Because of the symmetry in the strain tensor, the Jacobi method was used [8]: A sequence of orthogonal similarity transformations iteratively reduces the off-diagonal entries of the matrix. The accumulation of the product of the transformations gives the matrix of eigenvectors. This algorithm is reported to be very stable for real symmetric matrices.

Other than Von Mises stress, Von Mises strain has no mechanical meaning, except that it summarizes the strain tensor in a scalar value. There are other formulations, mostly energy based like strain equivalent density (SED), that are more significant, but most of them include material parameters (i.e. Young's modulus E). As the Demons Deformable Registration makes no assumption about the actual material properties they also should not be used for the strain mapping.

Generally strain is mapped onto the undeformed sample, so that results could be compared to FEA. However, if strain is mapped onto the sample in the deformed state, it gives a better understanding of the actual deformations in the sample.

5.2 Results

First it was verified, that the strain mapping yielded correct results for displacement fields with synthetic gradients. The strain mapping was also tested based on the displacement fields computed by FEA comparing the result from strain mapping with the strain computed by FEA ($R^2 > 0.9$). In both cases strain mapping yielded the expected results.

5.2.1 Strain mapping based on the registration of artificially deformed images

The combination of deformable registration and strain mapping had to be validated as well. This showed to be a bit more difficult than one could assume. The computation of strain is based on very small changes in the displacement field. Looking at two points 4 voxels or $120\mu m$ apart, 2% strain equals a relative displacement of $2.4\mu m$. That is 10 times less than the image resolution ($30\mu m$). Such small changes can hardly be computed accurately using deformable image registration. Based on the measurements in section 3.4 using binary images, a registration error of up to about the element size had to be assumed for 2%

compression steps. The registration error for relative displacements in neighboring points was less than that, because the elasticity force modeled in the Demons Deformable Registration restricts neighboring points from moving independently. But the round-off error introduced by the forward warping is completely stochastic and has to be assumed to be up to half the element size, that is $15\mu m$.

Despite these limitations, it was tried to statistically validate the displacements using the validation setup from chapter 3 with the artificially deformed images. But when the strain computed based on the displacement field from the deformable registration was correlated with the strain computed by FEA, low correlation coefficients resulted. The correlation was very low for all samples and all compression steps (see table 5.1). However Correlation of data superimposed by noise of the same order of magnitude is not a reliable measure for the accuracy, as it was already discussed in section 3.4.2.

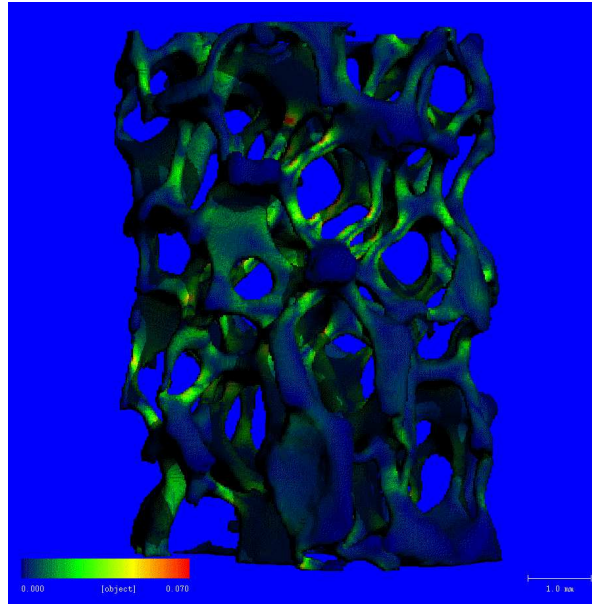
But also a visual comparison of the strain map computed from the registration of the artificially deformed samples with the strain computed by FEA showed that the strain map was not very meaningful (see figure 5.1(b)). There have been several effects, limiting the precision of the strain mapping: The artifacts in the displacement fields were much stronger if binary images were used (see section 3.1.1) than if grayscale images were used for the registration. Also, the FEA often did not yield distinct regions of high strain. Instead strain was uniformly distributed all over the sample with high peaks in very small regions, while strain computed by strain mapping was less localized. And finally already the round-off error introduced by the forward warping was much bigger than the relative displacements that would have had to be computed.

Table 5.1: Correlation factor (R^2) of the Von Mises strain computed by FEA and the strain computed by strain mapping based on the displacements computed by deformable registration. Each sample was analyzed at 1, 2 and 4% applied strain.

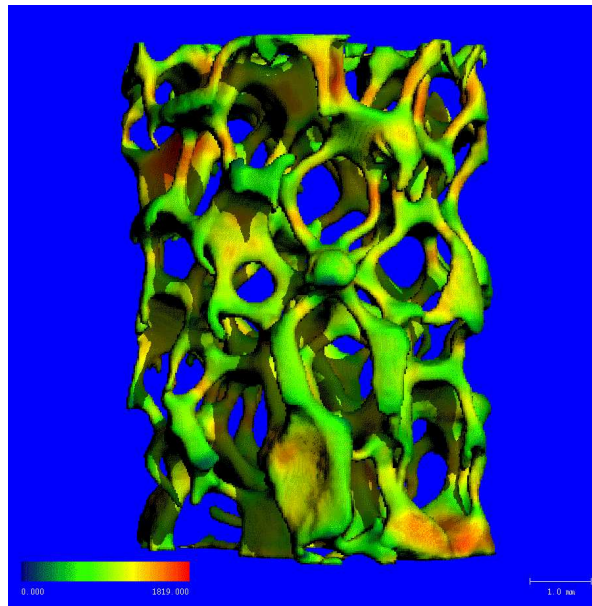
Sample	1%	2%	4%
1	0.07	0.08	0.08
2	0.05	0.05	0.03
3	0.09	0.08	0.04
4	0.07	0.06	0.02
5	0.05	0.04	0.01
Total	0.07 ± 0.02	0.06 ± 0.02	0.03 ± 0.03

5.2.2 Strain mapping based on the registration of IGFA images

To circumvent these problems, it was tried to validate the strain map based on the setup described in chapter 4 using displacements for the registration of images from IGFA experiments. In this setup a standard deviation of the difference between the displacements computed by the registration and the dis-



(a) FEA



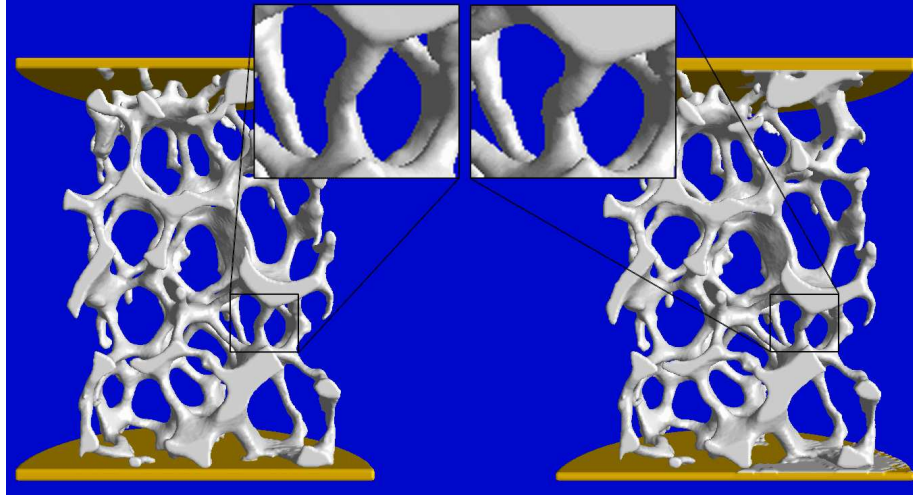
(b) Strain Map

Figure 5.1: Von Mises strain computed by FEA compared to the strain map based on the registration of the image artificially deformed according to the displacements computed by FEA.

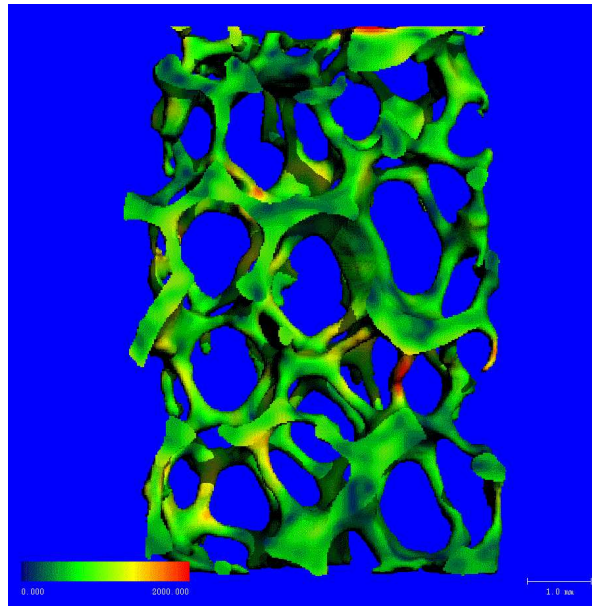
placements computed by FEA of only 1/3 of the element size was computed for two of the five samples. A strain map was computed based on the registration of the unloaded sample and the same sample loaded at 1% compressive strain. But also with this approach a statistical validation was not possible. Not even for the samples that showed remarkably good agreement of the displacements computed by FEA and the registration. As already stated above, correlation is not a meaningful measure if the data are superimposed by noise of the same order of magnitude. Because of the different localization of the strain in FEA and strain mapping, also measuring the standard deviation of the difference between the strain computed by FEA and registration did not yield meaningful results. The comparison of the absolute results for local strain computed by strain mapping and FEA was further complicated by a problem with the quantification of the local strain using strain mapping. Depending on the strength of the low-pass filtering, the strength of the local gradients in the displacement field changed, which in turn led to varying results for the local strain. It is difficult to estimate how the resulting strains have to be scaled for a given setup. However, for the implementation that was used here, using a Gaussian filtration with $\sigma = 5.0$ voxels, scaling the strains with 0.3 to 0.5 yielded best results. Such a scaling factor has to be determined manually: A heavily loaded element of the structure has to be found, ideally a rod-like connection with compression or tension only. The change in length has to be measured manually and compared to the strain computed by the strain mapping. Therefore this scaling parameter can not be quantified precisely.

Instead of validating the strain maps statistically, it was visually tested that the strain maps were meaningful. The strain maps based on the displacements computed by the registration of experimental images, were compared to animations showing how the sample deformed when it was loaded. It was tested that the strain map can reliably characterize regions with high strain. In samples compressed at 1 and 2% strain, high strain was defined as a von mises strain above 5%. This also included that no parts with small deformations were marked as regions with high strain. Therefore we looked for regions with big deformations in the deformed sample and verified that also the strain map indicates high strains there. On the other hand, we looked for regions of false strain, that is regions of high strain in the strain map while there is no deformation visible in the images of the sample.

Figures 5.2 and 5.3 were chosen, because regions of big deformations are easily visible. The sample is shown undeformed and with 2% strain applied and the strain map is depicted for this compression step. A step size of 2% was chosen, mostly to make the deformation more visible. The characterization of regions with high strain (Von Mises strain $> 5\%$) was reliable for 2% compression steps. Also for compression steps of 1% applied strain, the strain mapping yielded good and reliable results (see figure 5.4(a)), only that for such a small compression step the deformations were hardly visible in the sample. All parts with big deformations were accurately classified as regions with high strain and very few regions with small deformations showed false strain. Those regions of false strain were basically limited to the proximity of the end plates. They were

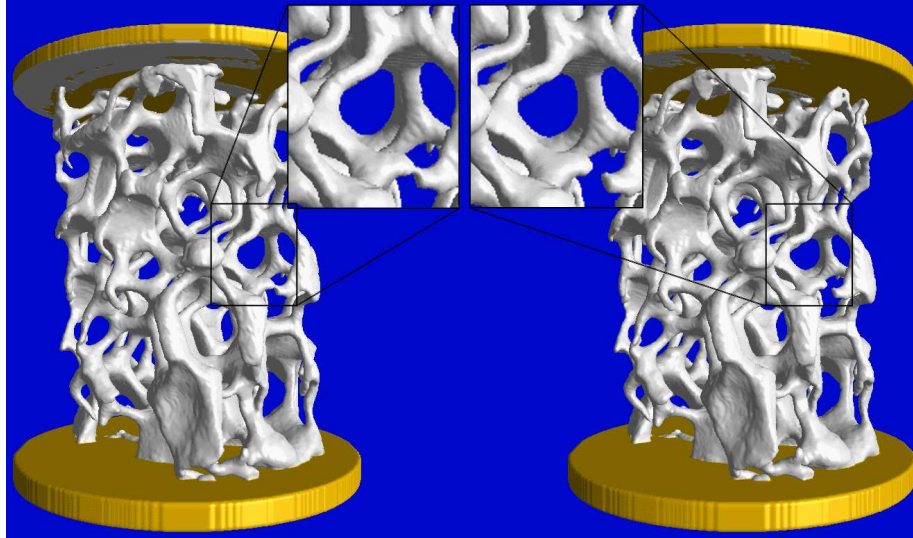


(a) The unloaded sample and with 2% applied strain

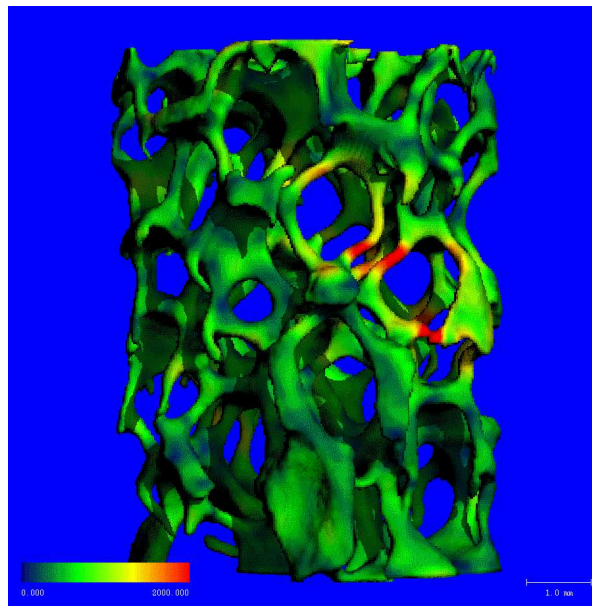


(b) Strain Map

Figure 5.2: Strain map based on the registration of IGFA images of a sample compressed at 2% strain.

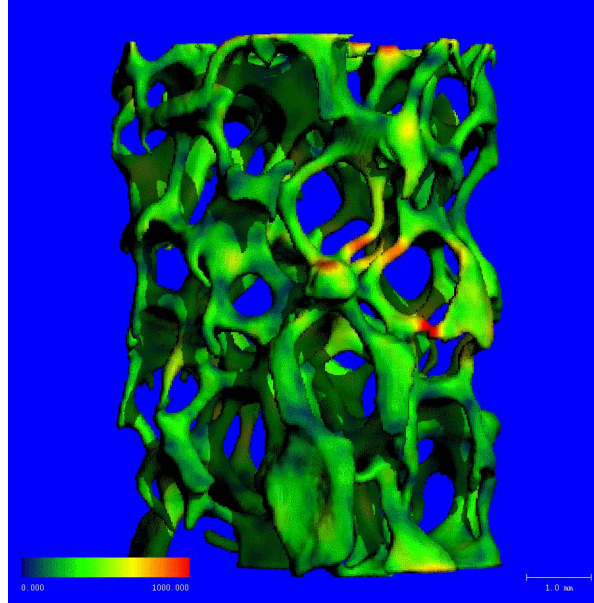


(a) The unloaded sample and with 2% applied strain

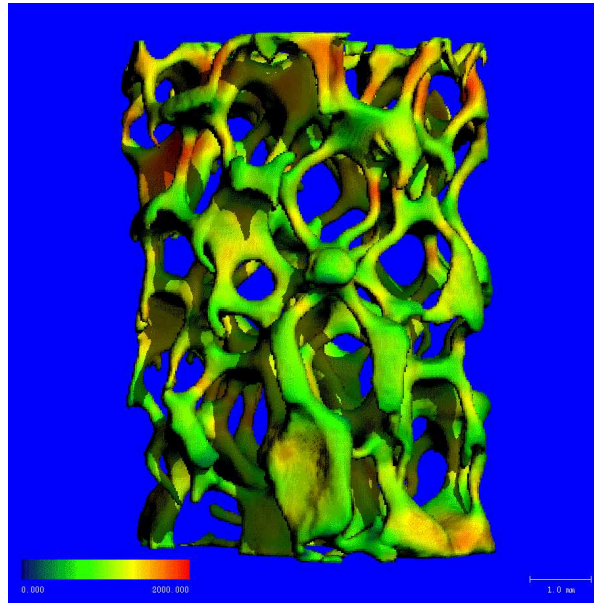


(b) Strain Map

Figure 5.3: Strain map based on the registration of IGFA images computed for a sample compressed at 2% strain.



(a) Strain map for 1% applied strain



(b) Strain map for 4% applied strain

Figure 5.4: Strain map based on the registration of IGFA images computed for a sample compressed at 1 and 4% strain.

most likely caused by some artifacts in the displacement field computed by the registration.

Strain maps were only correct for the registration of compression steps up to 2%. In samples loaded at 4% strain, strain maps did not show meaningful strains anymore (see figure 5.4(b)). This was due to the decreasing precision of the registration for 4% compression steps. It was also observed, that in regions with big displacements, strain was overestimated. As registration errors are bigger the bigger displacements occur, these errors in the displacement field can cause false strain. In the compression tests that were analyzed in this study, those regions of false strain were in the upper part of the samples. The magnitude of those artifacts dramatically increased with bigger compression steps, i.e. 4% applied strain (see figure 5.4(b)).

As it can be seen in figures 5.4(a) and 5.1(a), the localization of the strain in the strain map is much less precise than the localization of the strain computed by FEA. While FEA shows strain for small parts of the connections, the strain map based on the deformable registration can only indicate which connections are loaded with high strain. It had to be assumed that relative displacements of neighboring points can not be more precise than $5\mu m$ or 4 to 5% strain when grayscale images of the IGFA compression test are used as in this setup. A straight forward solution to this problem would be to compute the central difference over a bigger distance, i.e. considering points 10 voxels apart. However this would lead to a loss in terms of the localization of strain. The same effect was observed when very strong low-pass filtering was used: Increasing the sigma of the Gaussian filtration and the step size of the central differences have the same impact on the localization, while a stronger low-pass filtering yielded superior results for the quantification. It is a trade-off between the localization of strain and its quantification, for which the optimal parameters for the Gaussian filtration and the computation of the gradients vary depending on the exact setup of the testing.

5.2.3 Strain mapping a sequence of IGFA compression steps

Loading steps can also be registered subsequently. Instead of computing the displacements between the unloaded sample and the sample loaded at 4%, the registration can be split into parts: The images of the unloaded sample and the sample with 2% strain applied can be registered and the images of the sample with 2 and 4% strain applied can be registered. Because the displacements of the second registration step are computed based on the sample compressed at 2%, the displacement field has to be translated back to the undeformed sample. This works best using inverse warping (see section 3.1.2), because the transformation has to apply the inverse of the displacement field computed by the registration of the unloaded sample and the sample loaded at 2%. Once the displacement field is mapped back to the unloaded sample, the displacements of the first and the second loading step can be summed up. This yields a displacement field, that describes the total deformations between the unloaded sample and the sample

loaded at 4% strain. It can be used to compute a strain map of the sample compressed at 4% strain.

All the previous strain maps presented in this report have been visualized based on the undeformed sample, so that they could be compared to FEA results. However, strain maps visualized based on the deformed sample can be more intuitive and give a better understanding of the actual deformations in the sample.

There are two ways to compute the strain map based on the deformed sample. If the deformation fields and strain maps are already computed based on the undeformed sample, the strain maps can be warped by the inverse of the displacement field for the respective compression step using inverse warping. The warping might introduce some artifacts close to the end-plates of the sample. A better way would be to start the registration with the images swapped. This would compute the displacements from the loaded sample to the undeformed sample. The strain map computed from this displacement field is already based on the deformed state of the sample. But the inaccuracies of the first approach are usually too small to justify the start of a new deformable registration.

Figure 5.5 gives an impression, how strain mapping can be used to visualize and analyze deformations of subsequent loading steps in an IGFA experiment. The images of first two loading steps with 1 and 2% strain applied were registered directly. The other registration steps were between 2 and 4% and between 4 and 8% applied strain. The displacement fields of these last two compression steps were mapped back onto the undeformed sample and combined with the previously computed displacement field for the first two compression steps. Those combined displacement fields described the deformations from the unloaded sample to the samples with 1, 2, 4 and 8% strain applied. A strain map was computed based on those displacement field. The strain map was then warped using the inverse of the displacement field of the respective strain step, so that the final strain maps were based on the deformed sample.

It was an exception that the registration of the sample compressed at 4 and 8% converged. Usually the deformable registration of 4% compression steps did not yield results, that were accurate enough to compute strain maps.

The coloring of the strain maps can either be adapted to the applied strain or the maximal strain for each compression step, or it can be kept constant for all the compression steps. The later makes the individual steps more comparable, while the first gives a more detailed coloring of the strain map. Ideally the color-map would contain more than the 4 colors, that were used here.

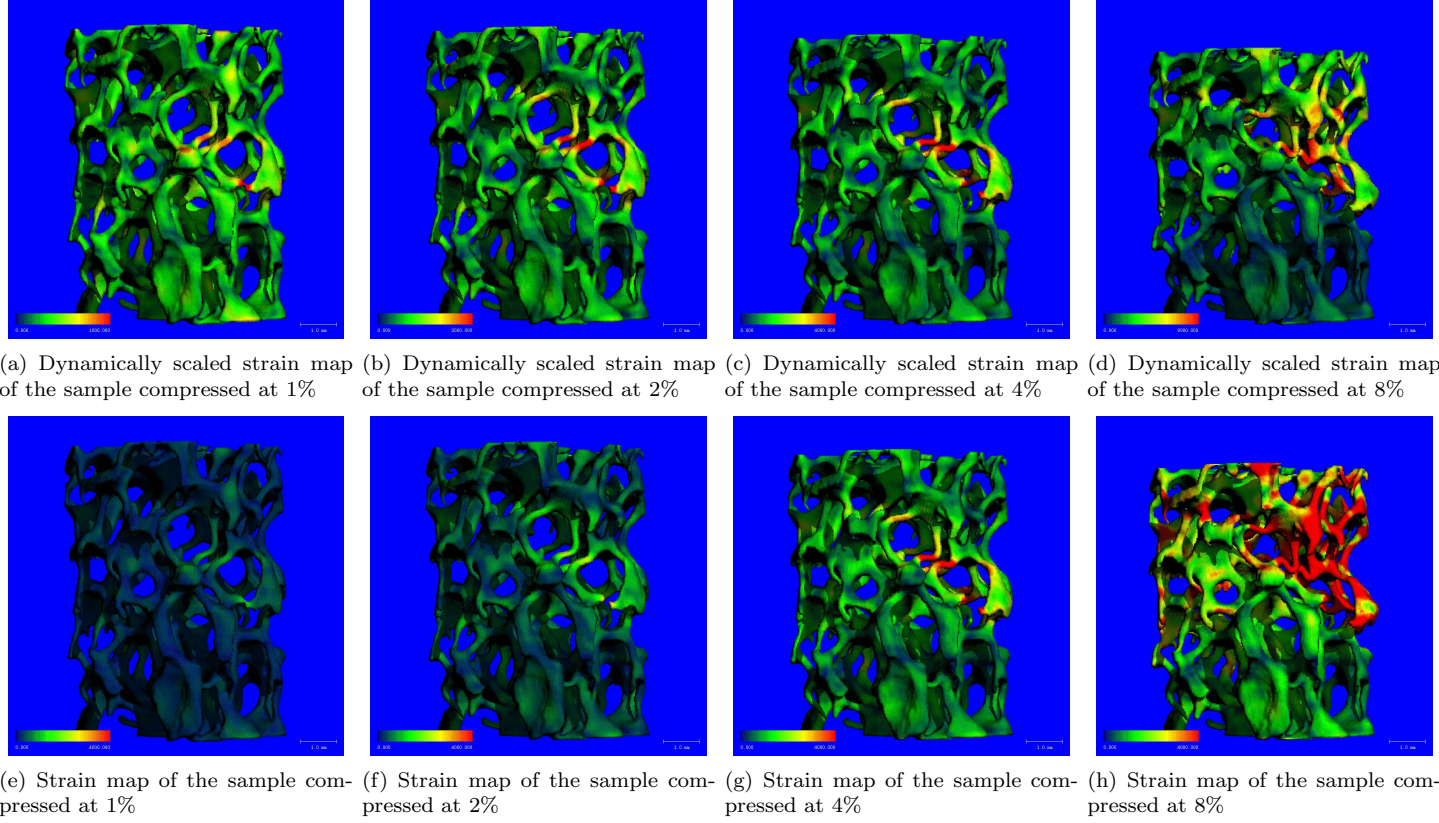


Figure 5.5: Strain maps based on the registration of IGFA images computed for a sample compressed at 1 to 8% strain. The strain was mapped onto the samples in the deformed state. In the upper row the coloring of the strain is adapted to the applied strain, while it is kept constant in the lower renderings.

5.3 Discussion

Strain maps computed directly from FEA displacements were equal to the strain computed by FEA. Therefore the strain mapping itself was correct. What was more difficult to validate was the combination of deformable registration and strain mapping.

Computing strain maps from local gradients in the displacement field needs very precise displacement fields, because local gradients are very sensitive to any artifacts in the displacement field. In order to compute strain with a localization, that is as good as possible, relative displacements of points close to each other have to be used. Therefore the relative displacements become very small. If for example 2% strain is applied to a region, two points, that are 4 voxels apart, undergo a relative displacement of less than 1/10 of the element size. However, the registration error in terms of relative displacements of neighboring points is usually smaller than the standard deviation of the registration error.

Strain maps based on the displacements computed by the deformable registration of artificially deformed images, as described in chapter 3, were not even qualitatively correct. Mostly because the error in terms of local displacement introduced using this setup was too big. A standard deviation of the registration error of almost the element size was measured. This error was introduced by the round-off error in forward warping and additional registration artifacts because binary images had to be used.

Compared to the registration of artificially deformed images, the registration of grayscale images from IGFA experiments showed a much smaller registration error (1/3 of the element size). This also yielded strain maps with a better quality. But also for the samples, where the displacements computed by the registration and by FEA were very similar, a statistical comparison of the strain map and the strain computed by FEA indicated poor agreement. This was mostly due to the different localization of the strain. While FEA computes strain for individual voxels, strain maps can only indicate which connections in the microstructure have high strain. Furthermore strain maps could not be computed precisely enough, so that strain could be exactly quantified at each position in the sample. Only a characterization of regions with high strain, here defined as Von Mises strain $> 5\%$, was possible.

It was visually validated that strain maps based on the displacements computed by the registration of images from IGFA compression test could reliably characterize regions with high strain (Von Mises strain $> 5\%$) for compression steps of 1 and 2% applied strain. Regions of false strain, that is where the strain map indicates high strain, although no deformations are visible in the images, were limited to regions close to the end plates of the samples and can be explained by artifacts that occurred during the registration. The registration of 4% compression steps was usually not accurate enough for the computation of strain maps. However, the displacements computed by an incremental registration of the subsequent compression steps can be combined. With this technique, also strain maps for samples compressed at 8% strain or even more can be computed.

Chapter 6

Conclusion and outlook

6.1 Conclusion

The local displacement fields computed by the Demons Deformable Registration were successfully validated. When binary images of the unloaded sample were artificially deformed using displacements computed by FEA, the displacements computed by the registration of the original and the deformed image showed a very good agreement with the displacements, that were previously applied to the image ($R^2 = 0.87 \pm 0.06$). For the whole process of applying displacements to the original image and registering the two images, the error in terms of local displacement was less than the element size. This also showed, that the Demons Deformable Registration did not just align the images somehow, but that the deformations were actually physically meaningful.

When grayscale images of IGFA compression test were used, the registration yielded displacements that very accurately reproduced the deformations of the aluminum foam samples up to a strain step of 2%. For bigger compression steps, not all regions of the images were correctly aligned and eventually the registration did not converge anymore. The displacements computed by the registration of the first compression step (from 0 to 1% applied strain) were also compared to the displacements computed by a linear elastic FEA simulation of the same compression test. For 2 of the 5 samples, a very good agreement between the displacements computed by FEA and the registration was observed ($R^2 = 0.9$). Furthermore, between the two versions of the local displacement fields (FEA and registration) a very small standard deviation of the difference of about 1/3 of the element size was measured. This indicates the upper limit of the average registration error. Due to experimental artifacts, the agreement between the displacements computed by FEA and the registration was lower for the other three samples. These unwanted influences included rotations in the sample due to off-center loading and misaligned end plates, leading to an unnatural bending of the sample. Both artifacts were not compensated in the FE analysis. If more samples had been tested, this could validate displacements

computed by linear elastic FEA for IGFA compression tests in the linear range.

The registration was also successfully tested using IGFA experiments of trabecular bone.

The strain maps based on displacements computed by the Demons Deformable Registration could be validated only visually. The exact quantification of local strain was not possible. However, when grayscale images were used for the registration, a reliable characterization of regions with high strain, here defined as Von Mises strain above 5%, for compression steps of up to 2% applied strain was achieved.

6.2 Outlook

The impact of different image resolutions was not evaluated in this study. It would be helpful to find out, how the registration error would change relative to the resolution of the images. It can be assumed that as long as the ratio between the features to be tracked and the resolution is within an ideal range, the registration error compared to the element size would stay the same. However, the boundaries of this interval would have to be found. This investigation should be based on grayscale images, as the deformable registration showed a different performance for segmented and unsegmented images. To be able to look at a bigger range of resolutions, only images of smaller parts of the aluminum foam samples would have to be used, due to limitations in memory and computation time.

The statistical validation of strain mapping at the microstructural level is still an unsolved problem. A possible approach would be to show, that FEA and strain mapping compute high strains for the same structural elements and that in both methods, strain in thicker or plate-like structures is smaller than in rod-like elements. This can be achieved by combining strain maps and FEA results with a structure specific skeletonization [22]. Based on the skeleton of the structure, an average strain can be computed for each rod. This would help dealing with the different localization of strain in strain maps and FEA and make them easier to compare.

A combination of undeformable registration and deformable registration would improve the accuracy of the displacements computed for big compression steps (i.e. 4% compressive strain). Undeformable registration could be used to compute a first guess of the displacement field, so that fewer iterations of deformable registration would yield a displacement field with a superior precision. Also, the impact of poorly aligned images of subsequent loading steps would be minimized. There are basically two ways to combine the two registration modalities: The undeformable registration can be used to deform the images prior to the deformable registration, so that top and bottom of the samples are artificially aligned already. Or an initial guess of the displacement field can be computed by the undeformable registration and passed to the Demons Deformable Registration as a starting point of the iteration. The ITK implementation of the Demons Deformable Registration allows setting an

initial displacement field, but the application interface has to be extended. Implementing undeformable registration is simplified by the ITK framework. A 3D visualization of the images is needed with the ability to pick landmarks in the images. Ideally this visualization uses a multi-resolution image pyramid to speed up the visualization of big images. A separate application can then either directly align the images, so that the landmarks fit best, or compute a transformation or equivalently a displacement field for the alignment.

The current implementation of Demons Deformable Registration using the ITK framework is not running on VMS, because ITK is not ported to the VMS operating system yet. Also, the registration showed a rather big memory consumption and was not very fast. Registering images with $300 \times 300 \times 350$ voxels needed 1.4 GB of memory and took 9-10 hours. Both, the memory requirement and the computation time restricts the registration to moderate sized images. Comparing the amount of memory used by the registration to the size of a floating point version of the same image (about 110 MB) shows that some excess copies of the images or of the displacement field must be present, as the two images and the 3 components of the displacement field account for only about $1/3$ of the 1.4 GB. An implementation of the Demons Deformable Registration that is not based on ITK could help saving memory and could reduce the computation time by reducing the number of layers in the hierarchy and by an optimized distribution of the computation on several threads. Such a stand alone implementation could easily be ported to VMS and integrated into IPL, which would help establishing strain mapping as a standard analysis tool for IGFA experiments.

In a longer perspective, strain mapping could be used as a benchmark for FEA and help the development of a non-linear FE solver. As soon as strain mapping is statistically validated, it could also be used for the validation of non-linear FEA.

Strain mapping has also many interesting applications in other tissues than trabecular bone, such as cortical bone or soft tissues like muscles, tendons and cartilage. Using higher resolutions, as in nanoscopic imaging, 3D strain mapping of micro-cracks would help understanding the initialization of structural failure in trabecular bone. At this resolution scale, an analysis of the impact of strain on the function of osteocytes could also yield a better understanding of mechanocellular interactions. In general, strain mapping of a rather continuous structure, such as cortical bone in nanoscopic CT images, yields better results than strain mapping of thin rod-like structures. Lacunae and vessels could serve as features for the registration.

Bibliography

- [1] A. Nazarian and R. Muller. Time-lapsed microstructural imaging of bone failure behavior. *Journal of Biomechanics*, 2004.
- [2] H. G. Simank, M. Stuber, R. Frahm, L. Helbig, G. H. van Lenthe, and R. Muller. The influence of surface coatings of dicalcium phosphate (dcpd) and growth and differentiation factor- 5 (gdf-5) on the stability of titanium implants in vivo - non-destructive micromotion tests and histomorphometric analysis in an animal model. *Biomaterials*, 27:3988–3994, 2006.
- [3] S. K. Boyd, S. Moser, M. Kuhn, R. J. Klinck, P. L. Krauze, R. Muller, and J. A. Gasser. Evaluation of three-dimensional image registration methodologies for in vivo micro-computed tomography. In *Ann. Biomed. Eng.*, volume 34, pages 1587–1599, 2006.
- [4] C. Mattmann and A. Kuhn. Site-specific measurement of micro-structural bone adaptation and prediction of adapting bone architecture. Master’s thesis, ETH Zurich, 2004.
- [5] D. Nicolella, D. Moravits, A. Gale, L. Bonewald, and J. Lankford. Osteocyte lacunae tissue strain in cortical bone. *Journal of Biomechanics*, 39(9):1735–1743, 2006.
- [6] B. K. Bay, T. S. Smith, D. P. Fyhrie, and M. Saad. Digital volume correlation: Three-dimensional strain mapping using x-ray tomography. *Experimental Mechanics*, 39(3):217–226, 1999.
- [7] Tait S. Smith, Brian K. Bay, and Mark M. Rashid. Digital volume correlation including rotational degrees of freedom during minimization. *Experimental Mechanics*, 2002.
- [8] William H. Press, William T. Vetterling, Saul A. Teukolsky, and Brian P. Flannery. *Numerical Recipes in C++: the art of scientific computing*. 2002.
- [9] R. Zauel, Y. N. Yeni, B. K. Bay, X. N. Dong, and D. P. Fyhre. Comparison of the linear finite element prediction of deformation and strain of human cancellous bone to 3d digital volume correlation measurements. *Journal of Biomechanical Engineering*, 2006.

- [10] J.-P. Thirion. Fast non-rigid matching of 3D medical images. Technical Report 2547, INRIA, May 1995.
- [11] J.-P. Thirion. Image matching as a diffusion process: an analogy with maxwell’s demons. *Medical Image Analysis*, 2(3):243–260, 1998.
- [12] Benoit M. Dawant, S. L. Hartmann, Jean-Philippe Thirion, Frederik Maes, Dirk Vandermeulen, and P. Demaerel. Automatic 3d segmentation of internal structures on the head in mr images using a combination of similarity and free form transformations: Part i, methodology and validation on normal subjects. *IEEE Trans. Med. Imaging*, 18(10):909–916, 1999.
- [13] D. L. Collins, P. Neelin, T. M. Peters, and A. C. Evans. Automatic 3d intersubject registration of mr volumetric data in standardized talairach space. *J Comput Assist Tomogr*, 18(2):192–205, 1994.
- [14] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. 3d brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.
- [15] Teo Popa, Luis Ibanez, Kevin Cleary, and Kenneth H. Wong. Itk implementation of deformable registration methods for time-varying (4d) imaging data. volume 6141, page 61412J. SPIE, 2006.
- [16] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, <http://www.itk.org/ItkSoftwareGuide.pdf>, second edition, 2005.
- [17] Su R., Campbell G. M., and Boyd S. K. Establishment of an architecture-specific experimental validation approach for finite element modeling of bone by rapid prototyping and high resolution computed tomography. *Medical Engineering and Physics*, 2007.
- [18] Reto Jost. Strain calculation in trabecular bone on three dimensional micro ct-images. ETH Zurich, 2006. Semesterarbeit.
- [19] Turner C H and Burr D B. Basic biomechanical measurements of bone: a tutorial. 1993.
- [20] R. Muller, T. Bosch, D. Jarak, M. Stauber, A. Nazarian, M. Tantillo, and S. Boyd. Micromechanical evaluation of bone microstructures under load. *SPIE*, 2002.
- [21] University of Michigan. Computational modeling of biological tissues. Website. <http://www.engin.umich.edu/class/bme506/bme5062000/bme506formlec/largedeflec/largedef.htm>.
- [22] Martin Stauber and Ralph Muller. Volumetric spatial decomposition of trabecular bone into rods and plates—a new method for local bone morphometry. *Bone*, 2006.

Appendix A

Documentation of the applications

A.1 DeformableRegistration

Usage: DeformableRegistration [-args] image1.aim image2.aim

Calculates the displacement field mapping image 2 onto image 1.

Gaussian filtering of the images before the registration

-g arg1 : sigma [in voxel]

A value 0 zero means no Gaussian smoothing.

Deformable registration settings

-l arg1 : number of levels (default 1)

-i arg1... : set number of iterations at each level (default 5)
-> number of arguments = 'number of levels'.

-d arg1... : elasticity parameter for each level (default 1.0)
-> number of arguments = 'number of levels'.

-s arg1,2,3 : set the starting shrink factors (default 1 1 1)
-> number of arguments = 'image dimension'.

output

-w filename : write the deformable field to 3 aim-files
(one for each component).

Only one basic file name has to be specified.

The filter adds _X, _Y, _Z to this name.

-t : write the deformation field as short data aim-files.

The application is directly based on the program written by C. Mattmann during her Diploma Thesis [4]. Changes to the original version include that all the visualization capabilities were removed and support for writing the files with short data was added.

It should be noted however, that this Program doesn't map image 1 onto image 2, but image 2 onto image 1. This is due to the reverse warping and

registration as it is implemented in ITK [16]. Only if the registration is combined with ITK's warping framework, it looks like image 1 was mapped onto image 2, because the registration and the reverse warping cancel out each other. But whenever the displacement field is to be analyzed, the correct direction of the mapping should be considered.

A.2 FEWarper

```
FEWarper -i InputFile
          -dx DeformationField in X Direction
          -dy DeformationField Y
          -dz DeformationField Z
          [-m Maximal Displacement] or [-s CustomScalingfactor]]
          [-f [-n NumIterations] [-s StopValue]]
          [-l]
          -o OutputFile
Warps an Image using a Deformation Field
```

It is assumed that the displacement field is of type IEEE Float and contains the actual displacements in mm or data of type SHORT. For Displacement Fields of type Short a scaling factor of 32768 (i.e. 1mm = SHRT_MAX) is assumed. This can be overridden by specifying the Displacement mapped to SHRT_MAX using "-m" or by a custom scaling factor using "-t".

Warping is always performed in the reverse direction. If forward warping is needed, use the -f option. (i.e. to physically apply a deformation field). The number of iterations and a StopValue (in mm) can be specified for the inversion. Default is -n 5 -s 0.00. With the -l option linear interpolation is used instead of nearest neighbor.

This Application uses the ITK warping framework. Additionally the displacement field can be inverted before the warping is done, so that in the end, the image is forward warped. The itkIterativeInverseDeformationFieldImageFilter was originally written by Corinne Mattmann and is used for the inversion of the displacement field.

A.3 DFieldInverter

```
DFieldInverter
          -dx DeformationField in X Direction
          -dy DeformationField Y
          -dz DeformationField Z
          [-n NumIterations] [-s StopValue]
          -o OutputFile
```

In order to evaluate the error introduced by the inversion of the displacement field, this application was implemented. It takes a deformation field, computes its inverse using `itkIterativeInverseDeformationFieldImageFilter` and writes the result to three output files.

A.4 Correlate

```
USAGE: correlate file1 file2 [maskfile]
      [-l lower bound (abs)] [-u upper bound(abs)]
      [-p numValuesToPrint]
```

Using AIMPack and "Numerical Recipes in C++" [8], this application computes statistics for the 2 AIM files, as well as comparing them. If a mask file is used, only the voxels within the mask are considered for the analysis. Statistical parameters for the two images are mean, variance and standard deviation. Results of the statistical comparison of the two images involve covariance, linear correlation coefficient, regression analysis (slope and offset), mean difference and standard deviation of the difference. Also Fisher's z-Transformation as well as Student's t-distribution and probability are computed, but those measures were not verified and might include errors.

The application is based on templates and can handle char, short and float data type. The images do not need the same origin or dimensions as the points are analyzed according to their global coordinates. The only restriction is that the data of the two images have to be of the same type.

Additional options include lower and upper boundaries for the values considered in the analysis. Points are only considered in the analysis, if the value of the respective point in the *file1* is within the lower and upper boundaries. Otherwise the point is ignored, as if it was outside the mask. The boundaries are in absolute values.

With the "-p" option, for a number of points within the mask, the corresponding values in the two images are printed. Those points are randomly chosen.

A.5 DFieldGen

```
USAGE: DFieldGen numChessFields value_mm file-in file-out
```

This generator of chess field patterns was used to test the correct warping of images. The geometry is taken from the input image, so that the generated file can easily be used to deform the input image. The second parameter specifies the value of the on-fields, that is usually a displacement in millimeters.

A.6 DispToAIM

```
USAGE: disp_to_aim -i inputaimfile
```

```

-m meshfile
-d dispfile
[-s scalefactor]
-o outputfiles

```

Converting the FAIM output into AIM files with the displacement fields

The output is written to 'outputfiles_X.aim', 'outputfiles_Y.aim', 'outputfiles_Z.aim'. The data is written as IEEE floats and displacements are indicated in mm.

The output of the large-scale linear elastic FE solver FAIM is usually a text file with a ".disp" ending. In FAIM, elements correspond to voxels. It contains information about the parameters, that were used, as well as nodal displacements and element stress and strain. In order to compute the element displacements, the nodes have to be associated to the correct elements and the displacements have to be interpolated. The scale factor can be used to scale all the displacements. This is equivalent to scaling the applied displacements or the applied force.

A.7 Euclidean

USAGE: euclidean

```

[-x Deformations in X Direction
-y Deformations in Y Direction
-z Deformations in Z Direction]
or [-d Deformations]
[-o Output Filename]

```

From the three components of the displacement field, the euclidean norm is computed. Either the individual components of the displacements are specified, in this case also the output file name is required, or the "-d" option is used. With this option, the beginning of the file name specified after "-d" is extended with "_X.AIM" (or analogous with y and z components) and by default the output is written to "_EUCL.AIM".

A.8 ForwardWarp

USAGE: forward_warp

```

-i Input File
[-x Deformations in X Direction
-y Deformations in Y Direction
-z Deformations in Z Direction]
or [-d Deformations]
-o Output Filename

```

The input image is deformed according to the displacements specified, using a forward warping approach. Input image and the displacements need to have

the same size and global coordinates. As described above, the files containing the components of the displacement field can be specified individually ("-x", ...) or the "-d" option can be used.

A.9 ObjWarp

```
USAGE: obj_warp -i inputfile
          [-d deformations] or [-x -y -z]
          -o outputfile
```

Object files (".obj") contain a surface mesh of an object previously extracted from an AIM. This surface mesh can be directly deformed without deforming the image first and subsequently extracting the deformed object. This eliminates the round-off error involved in forward warping.

The object file is text based and contains the coordinates of all the vertices and how the faces connect them. The coordinates are multiplied by a factor of 2, because the marching cubes algorithm is used to extract the surface from the image. In order to deform the mesh, the coordinates of the vertices are adapted.

Object files can be rendered using *viz3D* and *avs*.

A.10 Res2AIM

*** RESULTS_TO_AIM ***

Format: results_to_aim resfile [-h] [-l]

```
-h          : Help information.
-l          : List of parameters and output ranges.

-i          : Input aim file.
-d          : Input disp file.
-o          : Output aim file.
             Default: AimInfile + parameter based extension
-f arg1     : Scale results with Reaction Force of arg1 N.
-p arg1     : Parameter to plot.
-r arg1     : Range of results: predefined options
```

Usage: results_to_aim -i test.aim -d test.disp -p 7 -r 14

OR results_to_aim -i test.aim -d test.disp -o out.aim -p 7 -r 14

Overview of arguments for -p option

1: stress_11	11: strain_11
2: stress_22	12: strain_22
3: stress_33	13: strain_33
4: stress_12	14: strain_12
5: stress_23	15: strain_23
6: stress_31	16: strain_31
7: stress_vm	17: sed

```

18: strain_eff
19: princ_strain_1
20: princ_strain_2
21: princ_strain_3
22: strain_vm

```

Overview of arguments for -r option

```

0      : float output
1      : min          - max
2      : -max(abs)    - max(abs)
3      : min          - 0
4      : 0            - max
11     : mean-2*std   - mean+2*std
12     : -abs(mean)-2*std - abs(mean)+2*std
13     : mean-2*std   - 0
14     : 0            - mean+2*std
21 arg1 arg2 : arg1   - arg2

```

The conversion tool *res2aim* was extended to write the results as floating point values without any conversion or scaling to AIM files. Also the computation of the principal strains and Von Mises strain was implemented.

The computation of the eigenvalues of the strain tensor, that is needed for the principal strains, is done using the Jacobi method. For this, the C++ library from [8] was used.

A.11 Strainmap

USAGE: strainmap

```

[-x Deformations in X Direction
 -y Deformations in Y Direction
 -z Deformations in Z Direction]
or [-d Deformations]
-o Output Filename

-1 Write exx
-2 Write eyy
-3 Write ezz
-4 Write exy
-5 Write exz
-6 Write eyz
-7 Write Von Mises Strain
-a (default) Write all Strains

```

Based on the displacements, a local strain map is computed. The displacements can be individually specified or with the "-d" option, as described above. The output is written to the specified output file name extended by a parameter specific abbreviation (i.e. "_EXX.AIM"). By default, all six values of the strain tensor plus Von Mises strain are written. However, the results can also be individually specified and combined with each other.

Note that the displacement fields have to be heavily low pass filtered before the strain mapping (i.e. Gaussian filtration with $\sigma = 5.0$ and a support of 10 voxels).

Using the central differences with a step size of 2 voxels, gradients are computed in all three components of the displacement field. Based on those gradients, a second order approximation is used to calculate a local strain tensor (Green-Lagrange strain tensor). In order to compute Von Mises strain, principal strains are calculated. The NR library [8] is used to solve the eigenvalue problem of the strain tensor.

A.12 SynthGrad

USAGE: synth_grad x_increment y_increment z_increment file-out

increment (mm displacement) per 10microns

An AIM file with 100x100x100 voxels and a resolution of $10\mu m$ is generated with the gradients specified. This was used to verify the results of the strain mapping.

Appendix B

Description of the work flow

This chapter should document how strain maps can be computed and help other members of the group to perform a deformable registration and strain mapping of their images.

B.1 Deformable registration

If possible, the unsegmented versions of the images should be used for the registration. They yielded displacement fields, that were much more accurate, than binary images.

B.1.1 Preprocessing

The images used for the registration ideally are aligned, meaning that there is a part of the images, that stays fixated. In compression test, this usually is the bottom of the sample.

If the test is more complex, images might have to be aligned manually prior to the registration. A poor alignment can decrease the accuracy of the registration dramatically.

Ideally the images of the various loading steps should show the same part of the sample. If this is not the case, the accuracy of the displacements decreases in the regions of the images, where additional parts show up or parts disappear. This can also introduce false strain in those regions. The false strain can be reduced if the image is extended by some empty slices in the critical regions.

The computation of the registration linearly increases with the number of voxels, as long as there is enough memory. The registration of images with 30 million voxels (i.e. 300x300x350) took about 10 hours and needed 1.4 GB of memory. With 4GB of memory, images of up to 90 million voxels (i.e. 450x450x450) can be registered and the registration would take about 30 hours.

Before the images are transferred to the Linux system and the registration can be started, the images of the various compression steps need to have the

same dimensions.

B.1.2 Registration

By default, the Gaussian filtered images with the ending "gfilt.aim" are Gaussian filtered with $\sigma = 1.5$ and a support of 1 voxel.

```
~/example/samplename_step_0_gfilt.aim
~/example/samplename_step_1_gfilt.aim
```

The registration can then be started with

```
>DeformableRegistration -g 1 -l 1 -i 80 -d 3.0 -s 1 1 1 \
-w ~/example/samplename_step_01_dfield \
~/example/samplename_step_1_gfilt.aim \
~/example/samplename_step_0_gfilt.aim
```

The result are 3 AIM files containing the three components of the displacement field, with the displacements in millimeters.

```
~/example/samplename_step_01_dfield_x.aim
~/example/samplename_step_01_dfield_y.aim
~/example/samplename_step_01_dfield_z.aim
```

Ideally, the registration is executed using the *batch* command.

The parameters used above yielded best results for grayscale images.

B.1.3 Post-processing

After the registration, the displacement files have to be transferred back to VMS. They have to be Gaussian filtered using *XIPL*, i.e. with $\sigma = 5$ and a support of 10 voxels.

```
~/example/samplename_step_01_dfield_x_g5.aim
~/example/samplename_step_01_dfield_y_g5.aim
~/example/samplename_step_01_dfield_z_g5.aim
```

B.2 Strain mapping

The strain mapping is also done on VMS.

```
>strainmap
-x samplename_step_01_dfield_x_g5.aim
-y samplename_step_01_dfield_y_g5.aim
-z samplename_step_01_dfield_z_g5.aim
-o samplename_step_01_smap
-7
```

With the "-7" option, only the Von Mises strain (in percent) is written to disk.

```
samplename_step_01_smap_vme.aim
```

B.2.1 Post-processing

In order to prepare the strain map for the visualization, it has to be scaled and converted to short (16-bit integer) data type. After that, the strain map has to be masked using a segmented image of the unloaded sample. In *XIPL*:

```
/read
  -name                df
  -filename             samplename_step_01_smap_vme.aim
/norm_max
  -input               df
  -output              dfn
  -max                 3.2767
  -type_out            short
/gobj_maskaimpeel_ow
  -input_output         dfn
  -gobj_filename       samplename_step_0_thres.aim
  -peel_iter           0
/write_v020
  -name                dfn
  -filename             samplename_step_01_smap_vme_shrt_mskd.aim
  -compress_type       bin
  -version_020         true
```

In the resulting file a value of 1000 corresponds to 1% strain. The scaling factor might have to be adapted if strains above 30% occur.

samplename_step_01_smap_vme_shrt_mskd.aim

The final strain map can be visualized using *uct_3d*.

B.3 Advanced applications

B.3.1 Combining the registration of subsequent compression steps

Based on the unsegmented images, the displacement of the first and the second loading step can be computed.

```
~/example/samplename_step_0_gfilt.aim
~/example/samplename_step_1_gfilt.aim
~/example/samplename_step_2_gfilt.aim

>DeformableRegistration -g 1 -l 1 -i 80 -d 3.0 -s 1 1 1 \
  -w ~/example/samplename_step_01_dfield \
  ~/example/samplename_step_1_gfilt.aim \
  ~/example/samplename_step_0_gfilt.aim

>DeformableRegistration -g 1 -l 1 -i 80 -d 3.0 -s 1 1 1 \
```

```

-w ~/example/samplename_step_12_dfield \
~/example/samplename_step_2_gfilt.aim \
~/example/samplename_step_1_gfilt.aim

~/example/samplename_step_01_dfield_x.aim
~/example/samplename_step_01_dfield_y.aim
~/example/samplename_step_01_dfield_z.aim
~/example/samplename_step_12_dfield_x.aim
~/example/samplename_step_12_dfield_y.aim
~/example/samplename_step_12_dfield_z.aim

```

On VMS, the displacements then have to be Gaussian filtered with $\sigma = 5.0$ and a support of 10 voxels.

```

~/example/samplename_step_01_dfield_x_g5.aim
~/example/samplename_step_01_dfield_y_g5.aim
~/example/samplename_step_01_dfield_z_g5.aim
~/example/samplename_step_12_dfield_x_g5.aim
~/example/samplename_step_12_dfield_y_g5.aim
~/example/samplename_step_12_dfield_z_g5.aim

```

Back on Linux, the displacements of the second compression step have to be mapped to the undeformed sample.

```

>FEWarper \
-i ~/example/samplename_step_12_dfield_x_g5.aim \
-dx ~/example/samplename_step_01_dfield_x_g5.aim \
-dy ~/example/samplename_step_01_dfield_y_g5.aim \
-dz ~/example/samplename_step_01_dfield_z_g5.aim \
-o ~/example/samplename_step_02m_dfield_x_g5.aim
>FEWarper \
-i ~/example/samplename_step_12_dfield_y_g5.aim \
-dx ~/example/samplename_step_01_dfield_x_g5.aim \
-dy ~/example/samplename_step_01_dfield_y_g5.aim \
-dz ~/example/samplename_step_01_dfield_z_g5.aim \
-o ~/example/samplename_step_02m_dfield_y_g5.aim
>FEWarper \
-i ~/example/samplename_step_12_dfield_z_g5.aim \
-dx ~/example/samplename_step_01_dfield_x_g5.aim \
-dy ~/example/samplename_step_01_dfield_y_g5.aim \
-dz ~/example/samplename_step_01_dfield_z_g5.aim \
-o ~/example/samplename_step_02m_dfield_z_g5.aim

~/example/samplename_step_02m_dfield_x_g5.aim
~/example/samplename_step_02m_dfield_y_g5.aim
~/example/samplename_step_02m_dfield_z_g5.aim

```

On VMS, the two displacement fields can be added using *XIPL*.

```
/read
```

```

        -name                in
        -filename             samplename_step_02m_dfield_x_g5.aim
/read
        -name                in1
        -filename             samplename_step_01_dfield_x_g5.aim
/add_aims
        -input1              in
        -input2              in1
        -output              out
/write_v020
        -name                out
        -filename             samplename_step_02ma_dfield_x_g5.aim
        -compress_type       bin
        -version_020         true

```

Resulting in the total displacement fields for the sample at step 1 and 2.

```

samplename_step_01_dfield_x_g5.aim
samplename_step_01_dfield_y_g5.aim
samplename_step_01_dfield_z_g5.aim

samplename_step_02ma_dfield_x_g5.aim
samplename_step_02ma_dfield_y_g5.aim
samplename_step_02ma_dfield_z_g5.aim

```

From those displacement fields, strain maps can be computed.

```

>strainmap
-x samplename_step_01_dfield_x_g5.aim
-y samplename_step_01_dfield_y_g5.aim
-z samplename_step_01_dfield_z_g5.aim
-o samplename_step_01_smap
-7

>strainmap
-x samplename_step_12ma_dfield_x_g5.aim
-y samplename_step_12ma_dfield_y_g5.aim
-z samplename_step_12ma_dfield_z_g5.aim
-o samplename_step_02ma_smap
-7

```

The normalization and masking procedure is the same as in B.2.1.

B.3.2 Strain map based on the deformed sample

There are two ways to compute strain maps based on the deformed sample. If the strain maps were already computed based on the undeformed sample as described above, the unmasked version of the strain map can be warped by the displacement field.

```
samplename_step_01_smap_vme.aim
```

```
>FEWarper -f -n 50 -l \  
-i ~/example/samplename_step_01_smap_vme.aim \  
-dx ~/example/samplename_step_01_dfield_x_g5.aim \  
-dy ~/example/samplename_step_01_dfield_y_g5.aim \  
-dz ~/example/samplename_step_01_dfield_z_g5.aim \  
-o ~/example/samplename_step_01_smap_invv_vme.aim
```

Depending on the size of the images, this may take up to an hour, as the displacement field has to be inverted for the actual warping. The result is a strain map based on the deformed sample. The post processing is then similar to B.2.1, except that the segmented image of the deformed sample "_step_1_thres.aim" has to be used for the masking.

Another way would be to swap the images of step 0 and 1 already for the registration. The rest of the work flow would be as described in the beginning of this chapter, with the exception that the strainmap has to be masked with the image of the deformed sample.