

Stress Tensor Field Visualization for Implant Planning in Orthopedics

Christian Dick, Joachim Georgii, Rainer Burgkart, and Rüdiger Westermann

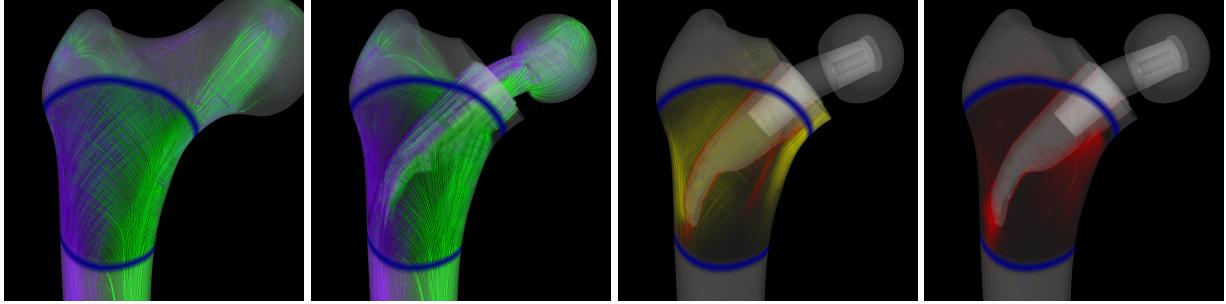


Fig. 1. Interactive visualizations (< 50 ms) of simulated stress tensor fields for a human femur under load, using our methods. Left: Principal stress directions and magnitudes in the physiological state (violet = tension, green = compression). Middle left: Principal stresses after a simulated implant surgery. Middle right: Change of normal stresses with respect to the principal stress directions of the physiological state (red = increase, yellow = decrease). Right: Change of shear stresses.

Abstract—We demonstrate the application of advanced 3D visualization techniques to determine the optimal implant design and position in hip joint replacement planning. Our methods take as input the physiological stress distribution inside a patient's bone under load and the stress distribution inside this bone under the same load after a simulated replacement surgery. The visualization aims at showing principal stress directions and magnitudes, as well as differences in both distributions. By visualizing changes of normal and shear stresses with respect to the principal stress directions of the physiological state, a comparative analysis of the physiological stress distribution and the stress distribution with implant is provided, and the implant parameters that most closely replicate the physiological stress state in order to avoid stress shielding can be determined. Our method combines volume rendering for the visualization of stress magnitudes with the tracing of short line segments for the visualization of stress directions. To improve depth perception, transparent, shaded, and antialiased lines are rendered in correct visibility order, and they are attenuated by the volume rendering. We use a focus+context approach to visually guide the user to relevant regions in the data, and to support a detailed stress analysis in these regions while preserving spatial context information. Since all of our techniques have been realized on the GPU, they can immediately react to changes in the simulated stress tensor field and thus provide an effective means for optimal implant selection and positioning in a computational steering environment.

Index Terms—Stress Tensor Fields, Biomedical Visualization, Comparative Visualization, Implant Planning, GPU Techniques.

1 INTRODUCTION

3D simulation and visualization methods for in-vivo bone stresses are of great importance in hip joint replacement planning, since they support the surgeon in finding the optimal design, size, and position of an implant during a preoperative design loop. This is due to the fact that an essential factor for the long-term stability of an implant is the load transmission to the adjacent bone stock. In particular, an optimized femoral stem should provide bone stress patterns that closely replicate the physiological stress state in order to avoid stress shielding with the consecutive effects of osteopenia, fracture and aseptic loosening [3, 17, 27, 31, 33].

In clinical practice, hip joint replacement planning is so far mainly done on 2D X-rays using simple, transparent template sheets with the outlines of the implants [41]. Using this two-dimensional approach,

rotational misalignment cannot be controlled and the implant position can only be revised in 2D planes. To overcome these limitations, 3D planning systems have been developed in the last years, which provide the surgeon with a 3D visualization of the implant position using patient-specific CT data [13, 17, 34, 36]. These systems are used for implant selection from a set of standard prostheses as well as for designing custom-made implants for abnormal anatomies. To accurately reproduce the planned implant position during surgery, computer-assisted navigation systems have been developed [1, 39, 41]. However, since the planning systems are merely geometry-based, i.e., do not provide any stress analysis, they do not allow for a patient-specific evaluation of an implant with respect to stress shielding.

Physically-based simulation of the mechanical response of a bone to an applied load (without or with an inserted implant) has been addressed in a number of research papers [3, 4, 35, 43]. Recent work has also shown the advantage of real-time simulation [11], which provides the possibility to obtain immediate feedback to changes of the implant shape and position as well as the exerted forces in the context of a computational steering environment. So far, these approaches visualize simulation results only by rendering scalar stress tensor norms on surfaces, or by using volume rendering of the respective scalar fields (see Figure 2). However, for medical purposes this kind of visualization is limited by the fact that it neglects important directional information in the simulated stress tensor fields. Furthermore, since an optimal implant should provide a bone stress distribution close to the physiological stress state, the surgeon needs a comparative visualiza-

• Christian Dick, Joachim Georgii, and Rüdiger Westermann are with Computer Graphics and Visualization Group, Technische Universität München, Germany, E-mail: {dick, georgii, westermann}@tum.de.

• Rainer Burgkart is with Klinik für Orthopädie u. Unfallchirurgie am Klinikum Rechts der Isar, Technische Universität München, Germany, E-mail: burgkart@tum.de.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

tion of the two stress distributions. To the best of our knowledge, such a comparative visual analysis of simulated 3D stress patterns to the physiological stress state has not yet been reported in the literature.

Contribution

To the best of our knowledge, we present the first interactive visualization approach for time-varying 3D stress tensor fields, and we demonstrate the application of this approach to find the most optimal implant shape, size and position in hip joint replacement planning. We use volume rendering to show stress magnitudes, and we combine it with transparent, shaded, and antialiased lines to indicate stress directions. To improve depth perception, these lines are rendered in correct visibility order, attenuated by the volume rendering. Since the proposed visualization techniques run entirely on the GPU, high frame rates can be achieved even for time-varying stress tensor fields. Therefore, the techniques provide an effective means for analyzing changing stress patterns as they are simulated in computational steering environments.

The user can flexibly change a number of visualization parameters such as transparency and color to explore the underlying 3D stress tensor field. Furthermore, changes of normal and shear stresses with respect to the principal stress directions of the physiological state can be visualized, thus allowing for an immediate recognition of the regions where stress shielding occurs. To restrict the user’s attention to the relevant regions and to enable a precise, yet context-preserving analysis of the stress directions in these regions, a focus+context technique is used. Only in the focus region fine details of the stress directions are shown, while the stress directions in the context region are visualized with lines on a much coarser scale.

A first qualitative evaluation of our visualization techniques shows the importance of the proposed methodologies for preoperative implant selection and positioning. Even though the visualizations differ vastly from the current state-of-the-art, practitioners have indicated an immediate medical benefit from these visualizations.

2 RELATED WORK

In this section, we survey related work in the area of the visualization of second-order, symmetric tensor fields, with the most prominent examples of stress/strain as well as diffusion tensor fields. A 3D second-order, symmetric tensor is represented by a symmetric 3×3 matrix (six independent scalar values), and is thus uniquely characterized by its three eigenvalues and three mutually orthogonal eigenvectors. By sorting the three eigenvalues, the corresponding eigenvectors can be classified into the major/medium/minor eigenvector, resulting in three eigenvector fields. Most of the proposed visualization methods make use of the eigen-decomposition.

Previous work can be classified into glyph-based approaches, methods originating from vector field visualization based on line/surface tracing or line integral convolution, direct volume rendering techniques, topology-based visualization methods, and approaches based on the visualization of the physical effect of the tensor field on the underlying media.

Glyph based approaches map the tensor field’s local properties to the shape and visual appearance of graphical icons. Simple glyph-based visualization techniques are hedgehogs, which represent the three eigenvectors by short lines, the absolute eigenvalues by the line length, and the sign of the eigenvalue by the line color, or ellipsoids, with half axes aligned to the eigenvectors and scaled according to the (normalized) eigenvalues [24]. To avoid visual ambiguities, more advanced primitives are used, for example composite shapes [38] or superquadric tensor glyphs [18].

A broad class of methods adopts approaches from vector field visualization. These approaches can be divided into line and surface tracing methods, and techniques which are based on line integral convolution (LIC). Examples for the first category are hyperstreamlines [9], which are traced in the major eigenvector field, and which have an ellipsoidal cross section which is determined by the medium and minor eigenvalues, as well as streamlines, streamtubes and streamsurfaces used in the context of diffusion tensor field visualization [44, 48]. To visualize the differences between two diffusion tensor fields, da

Silva et al. [8] proposed color-coding of streamlines to visualize the distance between corresponding fibers. To allow for the interactive visualization of static tensor fields based on the tracing approaches, several GPU-based rendering methods were presented. This includes the ray-casting of thin threads splatted into a volume [37], stress nets, which are limited to 2D tensor fields [40], ray-casting of hyperstreamlines [30], rendering of streamtubes by textured triangle strips [25], and the rendering of streamtubes combined with a level of detail technique [29]. Kondratieva et al. [20] presented a GPU-based particle tracing method for diffusion tensor fields that even performs the tracing directly on the GPU.

LIC-based approaches are based on the adaption of the filter kernel parameters [32] or of the shape of the filter kernel domain [46] to the local tensor field, or they apply the LIC to each eigenvector field separately, and then overlay the resulting images [16]. Tensor field visualization techniques based on direct volume rendering [5, 19] use a complex mapping to obtain a color volume from the tensor field. Topology-based approaches [10, 15, 47] compute and visualize the degenerated points/lines and the separating lines/surfaces of a tensor field. Another method interprets the tensor field as a stress tensor field, and visualizes its deforming effect on the underlying media [6, 45].

Each of the proposed visualization methods has advantages and disadvantages. Due to the complex nature of a 3D tensor field—at each point, three directions and three scalar values have to be visualized—most of the proposed techniques suffer from occlusion and visual cluttering problems, which often reduces their application to the 2D case, and/or they are difficult to interpret, at least for the non-expert.

3 STRESS TENSOR FIELD VISUALIZATION

Stress is a measure of the internal forces acting within a deformable body. Given a point as well as a normal vector specifying the orientation of an infinitely small, *imaginary* area element passing through that point, the stress vector σ is defined as $\sigma = \frac{dF}{dA}$, where dF denotes the force vector which the material on the positive side of the normal exerts on the material on the negative side via the area element dA .

At each point, the state of stress is fully described by the stress vectors for three mutually orthogonal orientations of the area element. In particular, the stress tensor \mathcal{S} contains the stress vectors for the three orientations corresponding to the axes of a Cartesian coordinate system. For an arbitrary orientation of the area element specified by its normal vector n , the stress vector is determined by $\mathcal{S}n$. This vector can be decomposed into a normal stress and a shear stress component, acting orthogonally and tangentially on the area element, respectively. For each stress tensor, there are three mutually orthogonal orientations of the area element where the shear stress components vanish. For these orientations, the normal stresses are called the *principal stresses* of the stress tensor. Mathematically, the *principal stress magnitudes* are the eigenvalues of the stress tensor, and they are independent of the coordinate system in which the stress tensor is given. The associated eigenvectors are the *principal stress directions*, and they are given with respect to the current coordinate system. The sign of the principal stress magnitudes classifies the stresses into tension (positive sign) or compression (negative sign). However, since there are three principal stresses acting at each point, the classification is with respect to a specific direction.

3.1 Stress Computation

In the numerical simulation of an elastic material the stress is derived from the deformation field. Using a finite element discretization, the deformation field u is computed by solving the equation $Ku = f$, where K is known as the stiffness matrix, and f denotes the external forces [2]. It is worth noting that in our application the occurring displacements are typically so small, that—from a visualization point of view—they do not result in any perceivable change of the body’s shape and thus can be ignored in the visualization. Once the displacement field is computed, we determine the internal stresses in the body using the element stress matrices

$$S = \frac{1}{V} \int_{\Omega} DB \, dx.$$

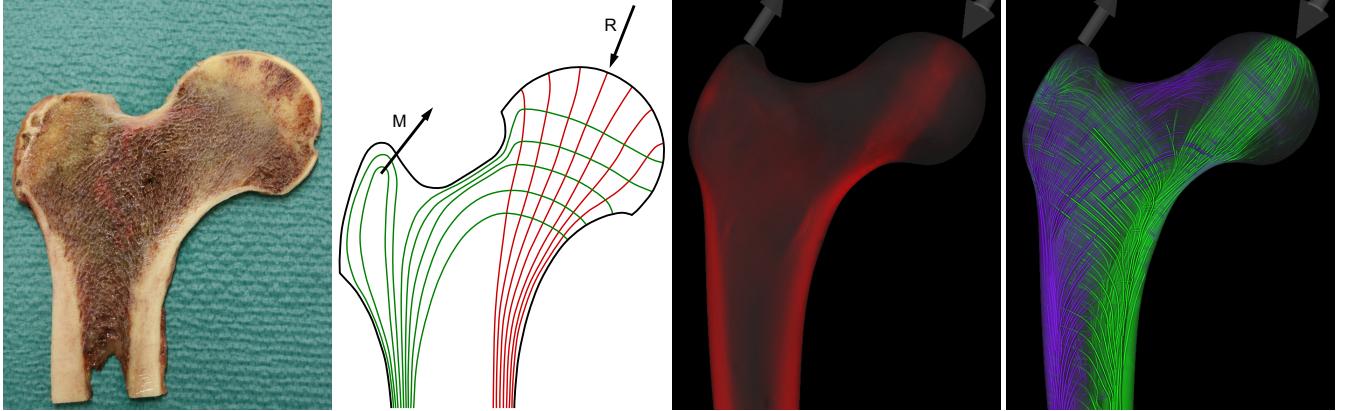


Fig. 2. From left to right: Cross section of a human femur showing the cortical and trabecular structures. Schematic overview of the principal stress directions in 2D (according to Pauwels [28]). 3D volume rendering of the scalar von Mises stress norm. Visualization of the principal stresses using the method proposed in this work.

Here, $V = \int_{\Omega} 1 dx$ is the volume of the finite element Ω , D is the underlying material law, and B is the element strain matrix [2]. By applying the displacement vectors of the element's vertices (linearized into a single vector) to this matrix, an averaged stress tensor \mathcal{S} over the domain of the element is computed.

In the current work we employ a computational steering environment for implant planning in orthopedics [11] to simulate the stress tensor field in quasi real-time whenever the exerted external forces or the shape or position of the implant are changed. The stress computation is based on a finite element model consisting of hexahedral finite elements aligned in a 3D Cartesian grid, which is built from patient-specific CT data in a preprocess. From the computed stress tensors, which are assumed to be located at the centers of the finite elements, we derive a continuous stress tensor field by means of componentwise trilinear interpolation.

3.2 Principal Stresses

A human femur is made of two different types of bone tissue—compact *cortical bone* forming its outer shell, and spongy *trabecular bone* occupying its interior (see left image in Figure 2). These two different types of bone tissue result from a natural optimization process, which allows the femur to be load-bearing and light-weight at the same time. Bone is living tissue which is subject to a life-long remodeling process, consisting of bone resorption and bone formation, and driven by the mechanical load situation on the bone [23]. For an unchanging load situation, bone resorption and formation are at equilibrium. This equilibrium is disturbed, however, if the mechanical load situation is changed (for example due to bone growth). A larger (smaller) deformation causes an increase of bone formation (resorption), which will decrease (increase) the deformation, until the equilibrium is restored. In this way, the bone adapts itself to changes in the load situation to provide optimal mechanical stability at light weight. As a consequence of this natural adaption process, the trabeculae of the spongy bone are aligned along the principal stress directions (Wolff's law [42]). This can be clearly seen in the left and right image of Figure 2, which show a cut through a real femur and the visualization of the principal stresses in a simulated physiological stress distribution, respectively. Since the trabecular structures—and thus the principal stress directions in the physiological state—are well-known by orthopedic surgeons from their education and daily work with X-rays (see middle left image in Figure 2, which is taken from medical literature and which shows a 2D sketch of the principal stress directions in a human femur), visualizing principal stresses gives the surgeon an intuitive way to judge the stress distribution resulting from a simulated implant surgery. In particular, this approach is much more intuitive than previous approaches based on the scalar von Mises norm [2], which is used in the middle right image of Figure 2.

Following these ideas, we combine volume ray-casting for the visualization of stress magnitudes, color-coded according to tension and

compression, with the rendering of transparent lines for the visualization of stress directions.

For the volume rendering of the principal stress magnitudes, we use the following mapping to assign a color to the stress tensor at a given point. First, we compute the tensor's eigenvalues and eigenvectors. We then determine for each principal stress magnitude σ_i , $i \in \{1, 2, 3\}$ a color contribution (RGB_i, α_i) as follows (using non-associated colors): The color RGB_i is determined by the classification into tension (violet) and compression (green), and the respective opacity α_i is proportional to the absolute stress magnitude, i.e.,

$$RGB_i = \begin{cases} \text{violet} & \text{if } \sigma_i \geq 0 \\ \text{green} & \text{if } \sigma_i < 0 \end{cases}, \quad \alpha_i = \text{saturate}(c \cdot |\sigma_i|),$$

where c is a user-specified scaling factor. In this way, regions with low stresses are almost transparent and do not occlude regions with high stresses. Since stress magnitude is finally represented by the brightness of the associated color $\alpha_i \cdot RGB_i$, the violet and the green base color are selected such that they have the same luminance. The three color contributions at a point are accumulated to get the final color (RGB, α) for the point according to

$$RGB = \frac{\sum_{i=1}^3 \alpha_i \cdot RGB_i}{\sum_{i=1}^3 \alpha_i}, \quad \alpha = \text{saturate}\left(\sum_{i=1}^3 \alpha_i\right).$$

To add directional information to the visualization, we trace lines along the three eigenvector fields, with six traces originating from each seed-point (two for each principal stress direction and its opposite direction). We choose the seed-points on a regular Cartesian grid, which is restricted by the bone surface, and we slightly jitter each seed position. To accurately integrate the lines of principal stress, we apply a fourth-order Runge-Kutta scheme with fixed step size. During tracing, the eigenvector-decomposition is computed on-the-fly for each sample point. To ensure directional consistency within a trace (note that the eigenvector direction is not uniquely determined), we flip computed eigenvectors if the angle to the current line direction is larger than 90° . We restrict the length of the traces to a user-specified limit, since the principal stress directions are a local property of the tensor field. The line is colored according to the principal stress magnitude in the trace direction, using the same color mapping as is used for the volume rendering. In this way, stress lines representing low stresses are transparent and thus do not occlude stress lines with high stresses.

4 COMPARISON TO PHYSIOLOGICAL STRESS STATE

One major problem in implant surgery is the change of the bone stress distribution due to the insertion of the implant. In particular, the stiffening of the bone by the implant as well as an unphysiological load transmission from the implant to the bone lead to a removal of stress from certain regions in the bone, which is called *stress shielding*.

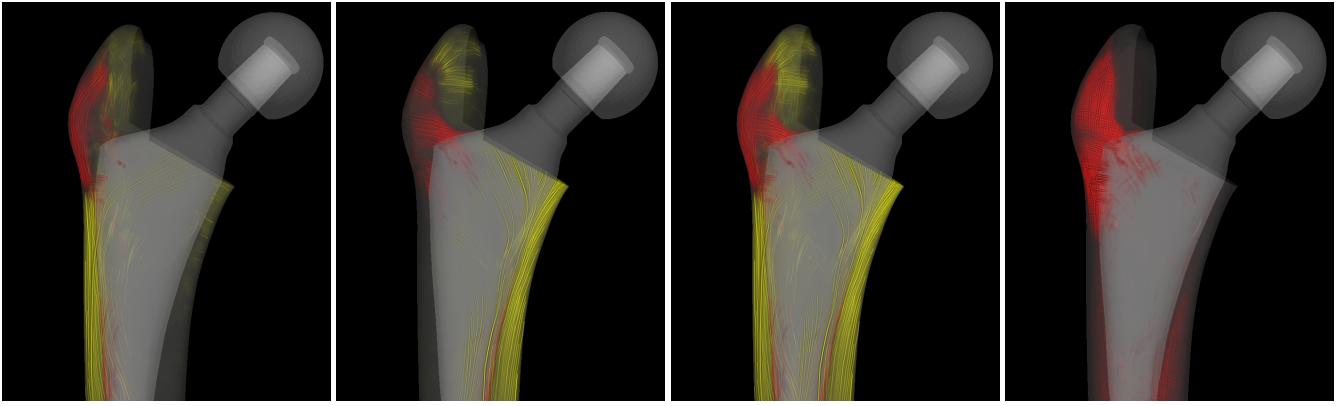


Fig. 3. Different options for the comparative visualization of the stress distribution after a simulated implant surgery and the physiological stress distribution (red = increase, yellow = decrease). Left: Change of normal stresses, tension only. Middle left: Change of normal stresses, compression only. Middle right: Change of normal stresses, both tension and compression. Right: Change of shear stresses.

Due to the bone's adaption to changed stress patterns, stress shielding causes degeneration of bone tissue, which finally may lead to fracture or loosening of the implant. Paradoxically, degeneration of bone tissue also occurs when the stress exceeds a certain critical stress magnitude [23]. Therefore, the ultimate goal of the medical procedure is to keep the stress distribution after insertion of an implant as close as possible to the physiological state. We address this requirement by providing a novel method to visualize differences between the simulated stress tensor field resulting from a virtual implant surgery and the physiological stress distribution. Our idea is to use the directions of the trabecular structures—which correspond to the principal stress directions in the physiological state—as a reference frame at each point for decomposing the stress tensors of both fields into normal stress and shear stress components, and then to visualize differences with respect to these components. It is worth noting that by using these reference frames, in the physiological state the normal stresses are principal stresses and the shear stresses vanish, corresponding to an optimal loading of the trabecular structures.

As a first step, our approach requires a model of the trabeculae. While in principle the main trabeculae are visible in high-resolution CT scans, we choose a model-driven approximation here. We define the reference directions by means of the principal stress directions that arise from the default loading of the intact bone (standing position). This model can be obtained by using the available simulation backend. It is worth noting that the intent of the proposed comparative visualization method is to show changes of normal and shear stresses *with respect to the trabecular directions*. Since the visualization's physical significance thus depends on the accuracy of the reference frames, let us mention here that we could employ any other, more accurate trabeculae model, too.

Using the trabeculae model, we can decompose any stress tensor \mathcal{S} at a given sample point with respect to the local reference frame e_i , $i \in \{1, 2, 3\}$, i.e., the trabecular directions, yielding a stress vector for each direction as illustrated in Figure 4. Each stress vector is then further decomposed into the normal stress component (magnitude σ_i) along

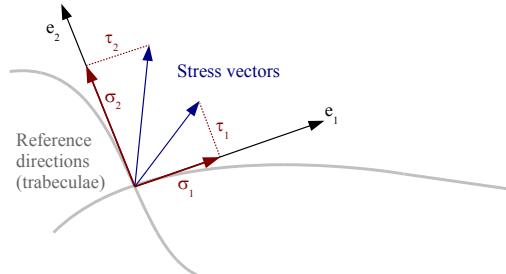


Fig. 4. Decomposition of the stress tensor (with respect to the reference directions e_1 and e_2) into normal stresses σ_1 and σ_2 and shear stresses τ_1 and τ_2 . The figure shows the 2D case for simplicity.

the direction e_i and the orthogonal shear stress component (magnitude τ_i) by

$$\begin{aligned}\sigma_i &= e_i^T \mathcal{S} e_i, \quad i \in \{1, 2, 3\}, \\ \tau_i &= \| \mathcal{S} e_i - \sigma_i e_i \|_2, \quad i \in \{1, 2, 3\}.\end{aligned}$$

The visualization maps the differences of the absolute normal/shear stress magnitudes to colors of the respective principal stress lines in the physiological stress distribution, using a similar color mapping as described in the previous section, but with different base colors. Analogously to the visualization of the principal stresses, these lines are attenuated by the volume rendering of the accumulated values for all directions. Since changes of normal/shear stress magnitudes with respect to the physiological stress distribution are highly important for the surgeon, we use highlight colors red and yellow to visualize the increase and decrease of the stress magnitudes, respectively. Due to the transparent rendering of the lines with their alpha value depending on the magnitude of the differences, only those lines are visible, which reflect a significant change in the load transmission and thus are of relevance for the surgeon. Since stresses within the implant are not of primary interest in the preoperative planning phase, these stresses can be hidden in the visualization as it is shown in Figure 3.

We provide four different visualization modes in order to reduce the information provided to the observer at once, which are shown in Figure 3:

Tension view: Change of normal stresses that are classified as tension in the physiological state.

Compression view: Change of normal stresses that are classified as compression in the physiological state.

Normal view: Change of normal stresses (tension and compression).

Shear view: Change of shear stresses.

5 GPU-BASED IMPLEMENTATION

To ensure immediate updates of the visualization whenever visualization parameters or the underlying stress tensor field are changed, we exploit computational, bandwidth, and rendering capacities on the GPU for stress tensor field visualization. In combination with a finite element simulation method which provides interactive update rates, the GPU implementation enables the surgeon to interactively explore the effects of different implant shapes and positions as well as varying loads on the stress distribution inside the patient-specific bone.

5.1 Eigen-Decomposition

As described in Section 3.2, our visualization method requires the computation of the principal stresses of a given stress tensor, i.e., its eigenvalues and eigenvectors. Our implementation uses the eigen-solver algorithm proposed by Hasan et al. [14], which allows for the analytical computation of the eigenvalues and eigenvectors of a 3×3 symmetric, positive definite matrix. For a GPU implementation, an

analytical solver is preferable to an iterative algorithm, since a constant runtime better fits to the GPU’s lock-step execution of parallel threads. On the GPU, Hasan et al.’s algorithm has recently been used for eigen-decomposition in the context of DT-MRI visualization [20], but in contrast to diffusion tensors, which are positive definite, i.e., which only have positive eigenvalues, stress tensors have eigenvalues of arbitrary sign. Fortunately, from the proof given by Hasan et al. it can be deduced that the algorithm is also applicable to non-positive definite 3×3 symmetric matrices. In this case, however, the algorithm does not implicitly return the eigenvalues in ascending order so that an explicit sorting of the three eigenvalues is required.

5.2 Line Tracing

The GPU-based computation of the stress lines is similar to the particle advection method presented in [21]. We use a fixed number of vertices per line, as well as a fixed step size. First, the seed-points are computed on the CPU and are uploaded into a buffer on the GPU. Then, starting from the seed-points, the lines are successively traced on the GPU by using a multi-pass “ping-pong” technique, which in each rendering pass simultaneously performs one trace step for each line.

The lines’ vertices are stored in two texture sets, which are alternately used as input and output. For each vertex, we store its position, the tangential direction of the stress line at that position (i.e., the eigenvector), the stress magnitude (i.e., the eigenvalue), as well as a number encoding which eigenvector field (major/medium/minor) the respective line follows. Therefore, each texture set consists of two four-component, floating-point textures. The tensor field is stored in two three-component, floating point 3D textures. This enables us to directly use the GPU’s texture sampling capabilities for trilinear, componentwise interpolation within the tensor field. The eigen-decomposition of the interpolated tensor is computed on-the-fly by employing the method described in the previous section.

Tracing lines with a length of N vertices requires N rendering passes. In pass 0, the seed-point of each line is fetched from the buffer and written into texture set 0. In all other passes $i = 1, 2, \dots, N-1$, each line’s previous vertex $i-1$ is fetched from texture set $(i-1) \bmod 2$, and each line’s new vertex i is computed using fourth-order Runge-Kutta integration. The new vertices are stored in texture set $i \bmod 2$.

5.3 Rendering

For the GPU-based visualization of the 3D stress tensor field, we combine volume ray-casting and transparent line rendering. While the volume rendering is used to simultaneously visualize the three stress magnitudes at each point, the lines are used to visualize selected stress directions as well as the stress magnitudes along these directions. The transparent lines and the volume are rendered in correct visibility order, which considerably improves depth perception, since the lines are attenuated with increasing depth (see Figure 5 for a comparison of different rendering techniques).

The correct visibility order is achieved by using the stencil-routed k-buffer proposed by Myers and Bavoil [26], in a similar way as it was used by Dick et al. [11] for the simultaneous rendering of transparent surfaces and volumes. The stencil-routed k-buffer can capture up to eight incoming fragments per pixel in a single rendering pass. It is thus significantly faster than depth-peeling, which can only capture one fragment per pixel in each pass. In our application, four rendering passes, i.e., up to 32 fragments per pixel, are typically sufficient.

The k-buffer consists of a multisampled texture accompanied by a stencil buffer, which is used to route the fragments falling into one pixel to individual subsamples in the texture. To capture more than eight fragments per pixel, the geometry has to be rendered successively into multiple k-buffers using different initial stencil values for each buffer. Since the maximum number of incoming fragments to a pixel, i.e., the number of required rendering passes, is not known a priori, we employ occlusion queries to detect at runtime whether an overflow of the k-buffer has occurred, i.e., whether further rendering passes are required. For details, we refer the reader to the original work [26]. In our application, where we use the k-buffer to render

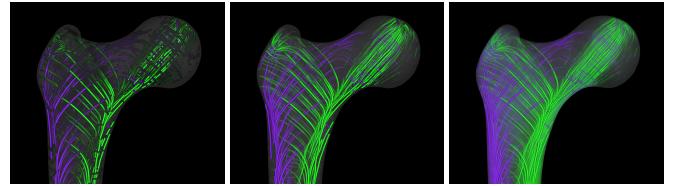


Fig. 5. Comparison of different rendering techniques: Left: Opaque lines without antialiasing. Middle: Transparent lines with antialiasing. Right: Transparent antialiased lines with volume attenuation (proposed method).

transparent geometry in correct visibility order, we reduce the maximum number of incoming fragments per pixel by discarding all line fragments with an alpha value < 0.02 during rendering, since these fragments would only have an insignificant contribution to the final image.

The visualization is generated in multiple passes. First, all opaque scene geometry is rendered into the frame buffer, with depth testing being enabled. The depth buffer content is later used during ray-casting to correctly clip transparent geometry and the volume at the opaque geometry. Then, all transparent geometry is rendered into the screen-aligned k-buffers, with depth testing as well as front/back face culling being disabled, so that all fragments are captured. The transparent geometry consists of the bone and the implant surface mesh, as well as the transparent stress lines. For each bone and implant fragment, we store its depth and normal in camera space, as well as an object ID which is later used to assign material colors. For the lines, we store the depth in camera space as well as the RGBA color of the fragment. All of these values are encoded into 2×32 bits, requiring a two-component unsigned integer texture format.

To render the stress lines, which have been traced using the method described in Section 5.2 and stored in two textures, the lines’ vertices are fetched from these textures in the vertex shader. We render shaded and antialiased lines to improve perception. This is achieved by employing the geometry shader to expand each line segment into a screen aligned quad, which is conceptually similar to the method proposed by Merhof et al. [25]. The expansion is performed in screen space, so that the lines are rendered with a specific pixel width. We apply transparency-based antialiasing using a box filter kernel [7]. Employing non-associated colors, this is implemented by multiplying a fragment’s alpha value with $\text{saturate}(w - d)$ in the fragment shader, where $w > 1.5$ denotes half the line width and d the distance of the fragment’s center from the line’s centerline, both values measured in pixels. To enhance the lines’ silhouettes, we multiply the fragment’s RGB color components by $\text{saturate}((w - 1.5) - d)$, which introduces thin black borders along the lines.

After all transparent fragments have been captured, we use a full-screen render pass to perform the volume rendering as well as to simultaneously create the final image. For each pixel, in the fragment shader we first fetch the transparent fragments corresponding to that pixel and sort them according to ascending depth. Furthermore, we clip at the opaque geometry by using the pixel’s depth buffer value. If the view ray corresponding to the pixel does not intersect the volume, the fragments lying in front of the opaque geometry are blended front-to-back, and the result is blended with the frame buffer content. Otherwise, we ray-cast the volume along the ray and simultaneously incorporate the transparent fragments in correct depth order. At each sample point, two texture fetches are performed to obtain the componentwise interpolated stress tensor, and its eigenvalues are computed to derive the sample’s color. This color is accumulated along the ray with the transparent fragments via front-to-back blending. If the ray reaches the depth of the opaque geometry it is terminated, and the accumulated color value is blended with the frame buffer content.

Due to the discrete tensor field voxelization, the volume and the stress lines can appear slightly outside of the bone surface mesh. We therefore improve the visualization by clipping the volume and line rendering at the bone surface. This is implemented by maintaining a flag during ray-casting, which specifies whether the cur-

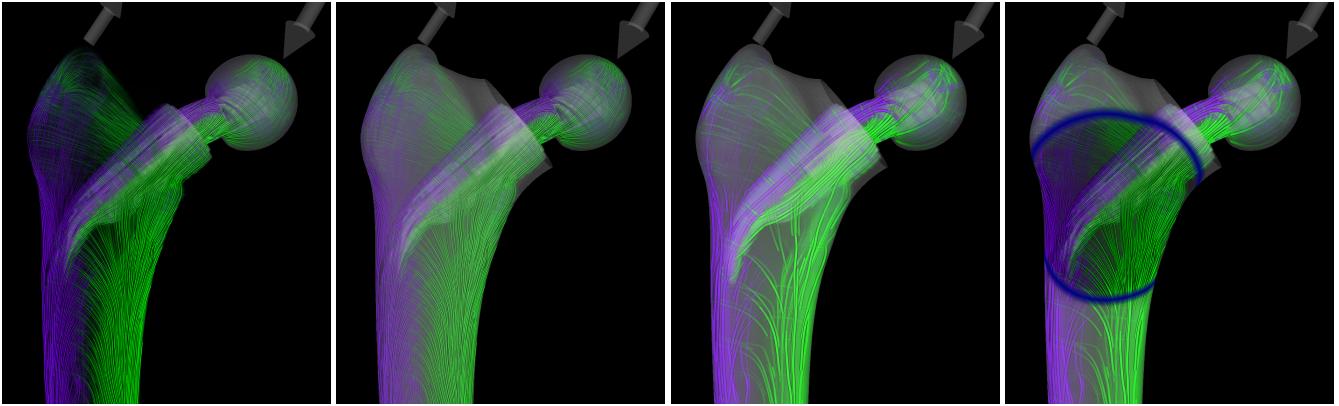


Fig. 6. Focus+context visualization of 3D stress tensor fields. Left: Spatial context is lost if only the stress lines are shown. Middle left: Increasing the opacity of the bone's surface improves the spatial perception but washes out the lines. Middle right: Fewer and thicker lines improve the perception of the global stress distribution but do not allow for a detailed analysis. Right: Focus+context visualization provides the spatial context and the global stress distribution, and allows for a detailed stress analysis in the focus region.

rent sampling/fragment position is inside or outside the bone surface, and which is toggled whenever a transparent bone mesh fragment is blended. Using this flag, the color contribution of the volume or a transparent line fragment is only blended if the respective position is lying inside of the bone. The removal of the stress visualization within the implant mesh is implemented analogously.

6 FOCUS+CONTEXT

To emphasize important regions in the data and to support a detailed stress analysis in these regions, we employ a lens-like focus+context metaphor as proposed by Krüger et al. [22] for scalar volume rendering. We also use a sphere-shaped focus region that snaps onto the bone surface, but in contrast to showing different structures in the context and the focus region, we show the same structures at different resolutions. In the context region, the density and thickness of the stress lines is decreased and increased, respectively, while the opacity of the bone's surface is increased.

This particular visualization approach is motivated by the observation that a detailed stress analysis requires fine details of the stress directions, which can be achieved by using a high seeding density of the stress lines. Furthermore, the bone's surface has to be completely transparent to optimally reveal these details. By only showing the stress lines, however, spatial perception is significantly reduced, especially if the user navigates around the bone. As a consequence, we increase the surface's opacity in the context region, yielding a considerably improved perception of the spatial relationships. The problem now is, that in the context region the blending of the bone's surface with the finely detailed directional structures results in a rather flat visualization showing low contrast and little structural information. To circumvent this drawback we reduce the seeding density of the stress lines in the context region, at the same time making them thicker and thus more distinguishable. By doing so, we provide important spatial context information of the bone as well as the global stress distribution, at the same time enabling a precise analysis of the focus region as demonstrated in Figure 6.

Decreasing the seeding density of the stress lines is implemented by increasing the spacing of the regular Cartesian grid that is used for the placement of the seed-points as described in Section 3.2. Only within the focus region we insert additional seed-points located in the middle between the existing ones. To achieve a visually continuous visualization, i.e., to avoid that lines suddenly pop up or disappear during movements of the focus region, we proceed as follows: The additional seed-points are placed within a sphere region located at the focus region center, but with a radius of the sum of the radius of the focus region plus the trace length. For the stress lines originating from the *original* seed-points, we use a total line width of 10 pixels, with the exception of line segments located within the focus region, for which we use a line width of 6 pixels (with a smooth transition at the border of the focus region). For the stress lines originating from the *additional*

seed-points, we always use a total line width of 6 pixels, but for line segments located outside of the focus region, we set the alpha value to 0 (again with a smooth transition at the border of the focus region).

7 RESULTS AND EVALUATION

We have integrated the proposed visualization techniques for 3D stress tensor fields into a virtual implant planning tool [11], which is based on an implicit multigrid finite element approach [12] for the simulation of the stresses in a human femur under load without and with an inserted implant. The tool allows the user to interactively adapt the shape, size and position of the implant as well as exerted forces.

All images were created in less than 50 ms on a standard desktop PC, equipped with an Intel Core 2 Quad Q9450 2.66 GHz processor, 8 GB of RAM, and an NVIDIA GeForce GTX 280 graphics card with 1024 MB of local video memory. The view port size was 1280×1024 . The resolution of the tensor field is $89 \times 83 \times 182$ voxels at a spacing of $1.5 \text{ mm} \times 1.5 \text{ mm} \times 2 \text{ mm}$. The trace length is 40 mm, the step size is 0.5 mm, and the seed points are located on a Cartesian grid with a spacing of 12 mm in the context region and 6 mm in the focus region.

In Figure 1, we demonstrate the effectiveness of our visualization techniques for the precise analysis of 3D stress tensor fields. First, we visualize tension (violet) and compression (green) as they arise in a healthy femur under a typical load. Next, we show the same data set after a virtual implant surgery. It is clearly visible how the implant changes the stress distribution in the surrounding bone tissue. To allow for a more precise comparison, we apply our novel comparative visualization technique in the next image. We use colors red and yellow to visualize the increase and decrease of the normal stresses. In this image, the stress shielding in the lower cortical head region becomes apparent (yellow region). The last image visualizes the shear stresses that arise mainly around the implant.

The relevance of the visualization techniques for orthopedic surgeons is demonstrated in Figure 7. We show comparative stress visualizations for two implant types and positions. It is clearly visible that the common G2 implants yield to stress shielding in large areas of the femur, especially in the cortical region close to the cut surface. The images demonstrate that modern shorter CUT implants yield a better physiological load transmission. However, these implants are more sensitive to correct positioning. If the implant does not have direct contact to the cortical region, the load transmission exhibits higher stress shielding and higher shear stresses, which increases the risk of fracture or aseptic loosening. These examples demonstrate that our visualization techniques help the surgeon to understand and evaluate the complex mechanical situation arising in hip joint replacement surgery. Consequently, the surgeon can optimize the shape, size and position of the implant to ensure the best possible long-term prognosis for the patient.

To obtain a feedback on the importance of the proposed methodologies to practitioners and the additional benefits they can derive, we

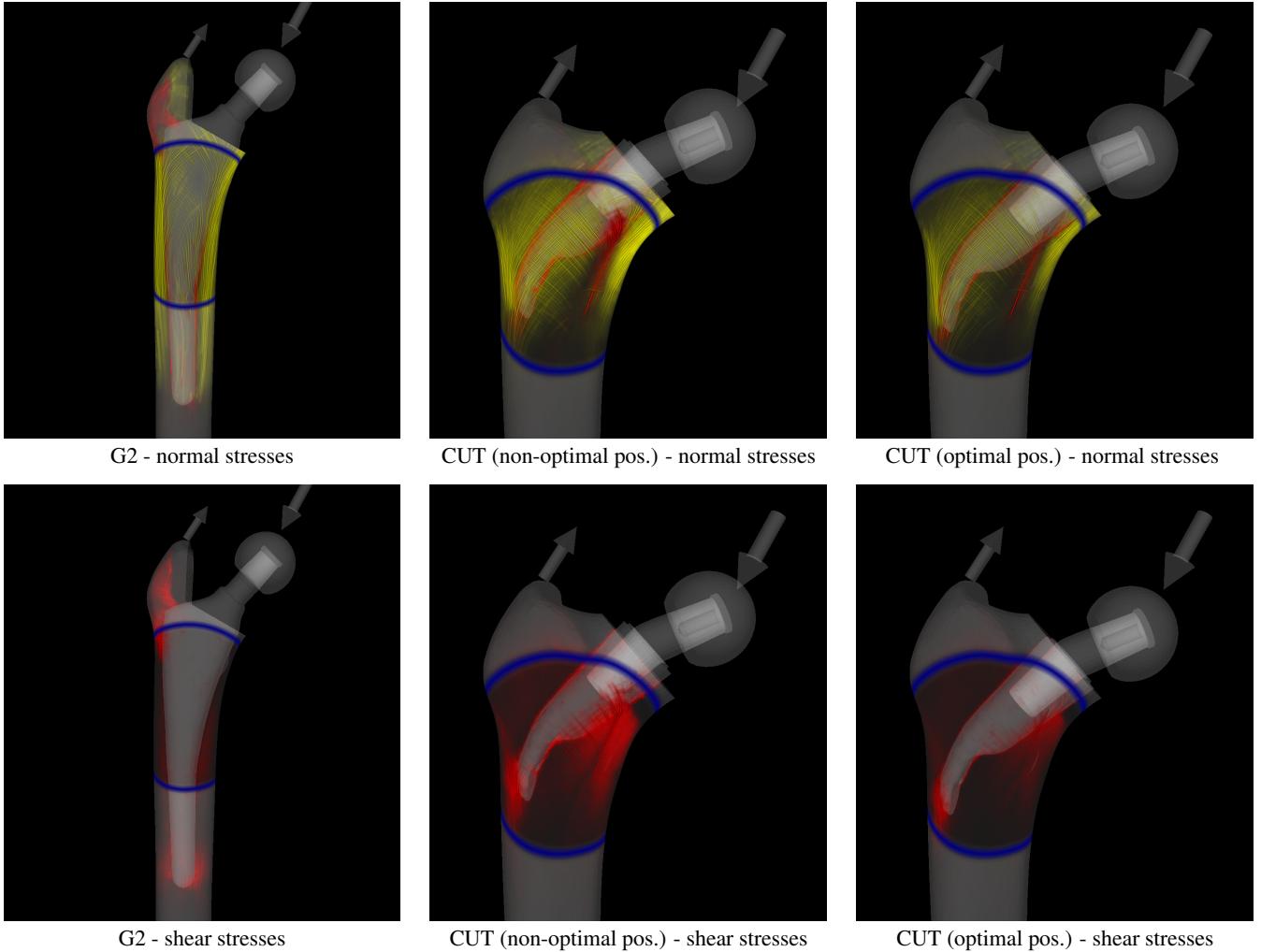


Fig. 7. Visualization of the stress distributions for varying implant types and positions in the femur. It is clearly visible that the CUT implant reduces the stress shielding in the cortical region and yields a more physiological load transmission than the G2 implant (left column). Optimal positioning of the CUT implant yields less stress shielding as well as reduced shear stresses around the implant (middle vs. right column).

have pursued a first qualitative evaluation of the orthopedic implant planning tool. We have shown the tool to five experienced orthopedic surgeons (assistant or associate professors) at the Klinikum Rechts der Isar, Technische Universität München. Specifically, we have shown to them visualizations of the stress distribution in two patient-specific data sets without and with three different implants, each inserted at two slightly different positions. We have asked them to judge whether the results can be easily understood and are in line with their intuitive expectation, in particular with respect to an immediate assessment of the patient-specific differences in the stress distributions induced by different implant types and positions.

All surgeons rated the proposed stress visualization methods as highly relevant in practice, especially because an intuitive perception of the different kinds of stresses, i.e., tension and compression, as well as their main directional structure in combination with the stress magnitudes is given. All of them stated that the comparative visualization of the stress distributions provides an intuitive and clearly visible rating of different implants, and that the use of such a support tool for preoperative implant selection can significantly reduce the risk of stress shielding. An even stronger statement has been made by all surgeons with respect to the use of the tool in the preoperative planning phase for patients with anatomical anomalies, e.g., pronounced hip-joint dysplasia and bone deformation, or in the context of revision interventions. All five surgeons have indicated strong interests to use the tool for this purpose, even though it causes an additional temporal expense. Due to the more precise and predictable operative care, this temporal expense would be accepted. Two surgeons were pointing

out the value of the tool in education, because all variants and results thereof can be tested, compared, and analyzed virtually.

The focus+context technique was considered to be an interesting feature of the system. However, there were some concerns with respect to its usability in clinical practice, since the focus region was placed manually in the demonstration. The majority of the surgeons requested that the focus region should be automatically placed at a medical point of interest (e.g., at the calcar region at the proximal medial femur), a feature we will integrate in the future. Two surgeons were critical concerning the current realization of the tool, because it does not yet support a default mechanism to automatically place the implant into a physiologically meaningful initial position. It was also seen problematic, whether the tool can really be used to precisely analyze the most optimal position of a selected implant. Here it was conjectured that the fine-granular adjustment as provided by the tool might be difficult to be mimicked in reality.

8 CONCLUSION

We have presented advanced, yet interactive, visualization techniques for 3D stress tensor fields. To the best of our knowledge, this is the first approach that allows for the interactive visual exploration of time-varying 3D stress tensor fields. Due to the sophisticated visualization options provided, surgeons get deep insights into the highly complex stress tensor data sets arising in preoperative implant planning environments. Especially, we have demonstrated the importance of comparing the stress distribution after an implant has been inserted with the physiological stress state, and we have provided powerful novel

visualization techniques for this purpose. These techniques allow the surgeon to optimize the shape, size and position of the implant with respect to the specific patient, which finally improves the long-term prognosis of the patient.

ACKNOWLEDGMENTS

The first author is funded by the International Graduate School of Science and Engineering (IGSSE) of Technische Universität München.

REFERENCES

- [1] L.-P. Amiot and F. Poulin. Computed tomography-based navigation for hip, knee, and spine surgery. *Clinical Orthopaedics and Related Research*, 421:77–86, 2004.
- [2] K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, 2002.
- [3] B.-A. Behrens, C. J. Wirth, H. Windhagen, I. Nolte, A. Meyer-Lindenberg, and A. Bouguesa. Numerical investigations of stress shielding in total hip prostheses. *Journal of Engineering in Medicine*, 222(5):593–600, 2008.
- [4] M. Bessho, I. Ohnishi, J. Matsuyama, T. Matsumoto, K. Imai, and K. Nakamura. Prediction of strength and strain of the proximal femur by a CT-based finite element method. *Journal of Biomechanics*, 40(8):1745–1753, 2007.
- [5] A. Bhalerao and C.-F. Westin. Tensor splats: Visualising tensor fields by texture mapped volume rendering. In *Proc. MICCAI*, pages 294–302, 2003.
- [6] E. Boring and A. Pang. Interactive deformations from tensor fields. In *Proc. IEEE Visualization*, pages 297–304, 1998.
- [7] E. Chan and F. Durand. *GPU Gems*, volume 2, chapter Fast Prefiltered Lines. Addison-Wesley, 2005.
- [8] M. J. da Silva, S. Zhang, Ç. Demiralp, and D. H. Laidlaw. Visualizing diffusion tensor volume differences. In *Proc. IEEE Visualization, Work in Progress*, pages 16–17, 2001.
- [9] T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.
- [10] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *Proc. IEEE Visualization*, pages 140–147, 1994.
- [11] C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Computational steering for patient-specific implant planning in orthopedics. In *Proc. Eurographics Workshop on Visual Computing for Biomedicine*, pages 83–92, 2008.
- [12] J. Georgii and R. Westermann. A multigrid framework for real-time simulation of deformable bodies. *Computer & Graphics*, 30(3):408–415, 2006.
- [13] H. Handels, J. Ehrhardt, W. Plötz, and S. J. Pöppel. Simulation of hip operations and design of custom-made endoprostheses using virtual reality techniques. *Methods of Information in Medicine*, 40(2):74–77, 2001.
- [14] K. M. Hasan, P. J. Bassar, D. L. Parker, and A. L. Alexander. Analytical computation of the eigenvalues and eigenvectors in DT-MRI. *Journal of Magnetic Resonance*, 152(1):41–47, 2001.
- [15] L. Hesselink, Y. Levy, and Y. Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE TVCG*, 3(1):1–11, 1997.
- [16] I. Hotz, L. Feng, H. Hagen, B. Hamann, K. Joy, and B. Jeremic. Physically based methods for tensor field visualization. In *Proc. IEEE Visualization*, pages 123–130, 2004.
- [17] K. Kawate, Y. Ohneda, T. Ohmura, H. Yajima, K. Sugimoto, and Y. Takakura. Computed tomography-based custom-made stem for dysplastic hips in Japanese patients. *Journal of Arthroplasty*, 24(1):65–70, 2009.
- [18] G. Kindlmann. Superquadric tensor glyphs. In *Proc. Joint Eurographics - IEEE TVCG Symposium on Visualization*, pages 147–154, 2004.
- [19] G. Kindlmann, D. Weinstein, and D. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE TVCG*, 6(2):124–138, 2000.
- [20] P. Kondratieva, J. Krüger, and R. Westermann. The application of GPU particle tracing to diffusion tensor field visualization. In *Proc. IEEE Visualization*, pages 73–78, 2005.
- [21] J. Krüger, P. Kipfer, P. Kondratieva, and R. Westermann. A particle system for interactive visualization of 3D flows. *IEEE TVCG*, 11(6):744–756, 2005.
- [22] J. Krüger, J. Schneider, and R. Westermann. ClearView: An interactive context preserving hotspot visualization technique. *IEEE TVCG*, 12(5):941–948, 2006.
- [23] B. Kummer. *Biomechanik: Form und Funktion des Bewegungsapparates*. Deutscher Ärzte-Verlag, 2005.
- [24] D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *Proc. IEEE Visualization*, pages 127–134, 1998.
- [25] D. Merhof, M. Sonntag, F. Enders, C. Nimsky, P. Hastreiter, and G. Greiner. Hybrid visualization for white matter tracts using triangle strips and point sprites. *IEEE TVCG*, 12(5):1181–1188, 2006.
- [26] K. Myers and L. Bavoil. Stencil routed A-buffer. In *Proc. ACM SIGGRAPH Technical Sketch Program*, page 21, 2007.
- [27] I. Oh and W. H. Harris. Proximal strain distribution in the loaded femur. *Journal of Bone and Joint Surgery, American Volume*, 60(1):75–85, 1978.
- [28] F. Pauwels. *Atlas zur Biomechanik der gesunden und kranken Hüfte*. Springer, 1973.
- [29] V. Petrovic, J. Fallon, and F. Kuester. Visualizing whole-brain DTI tractography with GPU-based tuboids and LoD management. *IEEE TVCG*, 13(6):1488–1495, 2007.
- [30] G. Reina, K. Bidmon, F. Enders, P. Hastreiter, and T. Ertl. GPU-based hyperstreamlines for diffusion tensor imaging. In *Proc. Eurographics/IEEE-VGTC Symposium on Visualization*, pages 35–42, 2006.
- [31] T. Renkawitz, F. S. Santori, J. Grifka, C. Valverde, M. M. Morlock, and I. D. Learmonth. A new short uncemented, proximally fixed anatomic femoral implant with a prominent lateral flare: design rationals and study design of an international clinical trial. *BMC Musculoskeletal Disorders*, 9:147, 2008.
- [32] A. Sigfridsson, T. Ebbers, E. Heiberg, and L. Wigström. Tensor field visualisation using adaptive filtering of noise fields combined with glyph rendering. In *Proc. IEEE Visualization*, pages 371–378, 2002.
- [33] J. A. Simões and M. A. Vaz. The influence on strain shielding of material stiffness of press-fit femoral components. *Journal of Engineering in Medicine*, 216(5):341–346, 2002.
- [34] M. Starker, P. Thümler, A. Weipert, and S. Hanusek. Computer-assisted prosthesis selection and implantation control. *Orthopäde*, 29(7):627–635, 2000.
- [35] F. Taddei, E. Schileo, B. Helgason, L. Cristofolini, and M. Viceconti. The material mapping strategy influences the accuracy of CT-based finite element models of bones: An evaluation against experimental measurements. *Medical Engineering and Physics*, 29(9):973–979, 2007.
- [36] M. Viceconti, A. Chiarini, D. Testi, F. Taddei, B. Bordini, F. Traina, and A. Toni. New aspects and approaches in pre-operative planning of hip reconstruction: a computer simulation. *Langenbeck's Archives of Surgery*, 389(5):400–404, 2004.
- [37] A. Wenger, D. F. Keefe, S. Zhang, and D. H. Laidlaw. Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE TVCG*, 10(6):664–672, 2004.
- [38] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6(2):93–108, 2002.
- [39] K.-H. Widmer and P. A. Grützner. Joint replacement—total hip replacement with CT-based navigation. *Injury*, 35(1):S-A84–S-A89, 2004.
- [40] A. Wilson and R. Brannon. Exploring 2D tensor fields using stress nets. In *Proc. IEEE Visualization*, pages 11–18, 2005.
- [41] R. L. Wixson. Computer-assisted total hip navigation. *Instructional Course Lectures*, 57:707–720, 2008.
- [42] J. Wolff. *Das Gesetz der Transformation der Knochen*. Hirschwald, 1892.
- [43] Z. Yosibash, R. Padan, L. Joskowicz, and C. Milgrom. A CT-based high-order finite element analysis of the human proximal femur compared to in-vitro experiments. *Journal of Biomechanical Engineering*, 129(3):297–309, 2007.
- [44] S. Zhang, Ç. Demiralp, and D. H. Laidlaw. Visualizing diffusion tensor MR images using streamtubes and streamsurfaces. *IEEE TVCG*, 9(4):454–462, 2003.
- [45] X. Zheng and A. Pang. Volume deformation for tensor visualization. In *Proc. IEEE Visualization*, pages 379–386, 2002.
- [46] X. Zheng and A. Pang. HyperLIC. In *Proc. IEEE Visualization*, pages 249–256, 2003.
- [47] X. Zheng, B. N. Parlett, and A. Pang. Topological lines in 3D tensor fields and discriminant Hessian factorization. *IEEE TVCG*, 11(4):395–407, 2005.
- [48] L. Zhukov and A. H. Barr. Oriented tensor reconstruction: Tracing neural pathways from diffusion tensor MRI. In *Proc. IEEE Visualization*, pages 387–394, 2002.