

A Toolkit for the Visualization of Stress and Strain Tensor Fields in Biological Tissue

Burkhard Wünsche and Richard Lobb

Department of Computer Science

The University of Auckland, Private Bag 92019

Auckland, New Zealand

Email: burkhard@cs.auckland.ac.nz, richard@cs.auckland.ac.nz

Abstract—Second-order tensors are a fundamental entity in engineering, physical sciences and biomedicine. Examples are stresses and strains in solids and viscous stresses and velocity gradients in fluid flows.

Unfortunately second-order tensors have the same complexity as matrices. Large amounts of tensor data are hence difficult to interpret. Visualizing the tensor data is important to scientists because it improves their ability to understand and interpret the data.

This paper presents a toolkit for the visualization of stress and strain tensor fields in biological tissue. The toolkit incorporates a wide range of existing visualization techniques. Data exploration is facilitated by allowing the user to derive new fields from already existing fields. We also suggest some improvements for existing visualization methods, introduces a user interface for the construction of color maps, and a new visualization method based on particle clouds.

Key Words: Visualization toolkit - tensor fields - stresses - strains - finite element models - heart model

I. INTRODUCTION

During the past few years, physically based modeling has emerged as an important new approach to computer animation and computer graphics. An important subfield is the modeling of elastic bodies as used, for example, in computer animation [Parke and Waters, 1996] and surgical simulation [Sagar et al., 1994]. A mathematical description of elastic bodies is given by the theory of *elasticity*, which is the study of the deformation of a solid body under loading together with the resulting stresses and strains.

Stresses and strains are important since they are related to material deformation and material failure. Unfortunately stresses and strains are tensors having the same complexity as matrices. Large amounts of tensor data are hence difficult to interpret. The visualization of tensor fields improves the understanding and interpretation of tensor data and is therefore important in the fields of engineering, medicine, and biomechanics.

In this paper we first introduce the concept of stress and strain tensors and then present a toolkit to visualize stresses and strains in biological tissue. We demonstrate our toolkit using finite element models of the heart.

II. NOTATIONS AND DEFINITIONS

We use a rectangular Cartesian coordinate system with the base vectors \mathbf{e}_1 , \mathbf{e}_2 , and \mathbf{e}_3 . Vectors are written in small bold Latin letters and tensors in capital bold Latin letters or small bold Greek letters. Examples of tensors are stresses and strains which are explained in the next section.

We deal only with second-order tensors which are linear transformations between vectors and can be represented by matrices.

An important property of a n -dimensional symmetric tensor \mathbf{T} is that there always exist n *eigenvalues* λ_i and n mutually perpendicular *eigenvectors* \mathbf{v}_i such that

$$\mathbf{T}\mathbf{v}_i = \lambda_i\mathbf{v}_i \quad i = 1, \dots, n \quad (1)$$

The eigenvalues and eigenvectors define the tensor completely.

III. DISPLACEMENT, STRAIN, AND STRESS

An elastic body under an applied load deforms into a new shape. We want to find a mathematical description for the displacement the body undergoes. Figure 1 shows a body before and after deformation. In order to understand the deformation we consider how two neighbouring points P and Q change in relation to each other. After deformation the points P and Q arrive in position $\mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\mathbf{x}' + d\mathbf{x}' = \mathbf{x} + d\mathbf{x} + \mathbf{u}(\mathbf{x} + d\mathbf{x})$, respectively, where \mathbf{u} is called the *displacement field*.

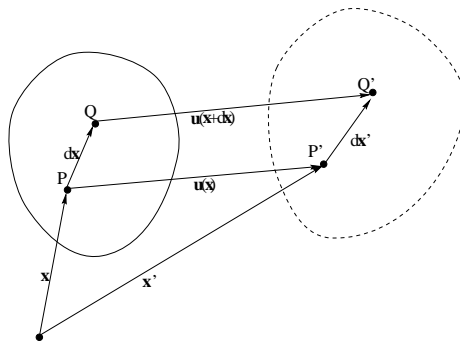


Fig. 1. A body before and after deformation.

Lai et al. [Lai et al., 1986] show that

$$d\mathbf{x}' = d\mathbf{x} + (\nabla\mathbf{u})d\mathbf{x}$$

where the second-order tensor

$$\nabla\mathbf{u} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{pmatrix}$$

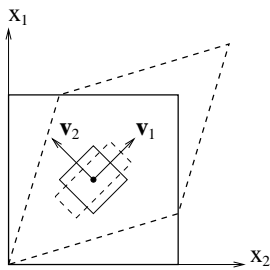


Fig. 2. Principal stretch axes of a deformed body.

is known as the **displacement gradient**.

It can be seen that if $\nabla \mathbf{u} = 0$ then $d\mathbf{x}' = d\mathbf{x}$ and the motion in the neighborhood of point P is that of a rigid body translation. The information about the material deformation around P is contained in $\nabla \mathbf{u}$. However, the displacement gradient contains also rotational information. An entity which contains information about “pure” deformation, i.e., without rigid body rotation is the *Lagrangian strain tensor*

$$\mathbf{E}^* = \frac{1}{2} ((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T + (\nabla \mathbf{u})^T (\nabla \mathbf{u}))$$

For small deformations the displacement gradients $\partial u_i / \partial x_j$ are small and the quadratic term of \mathbf{E}^* can be neglected giving the *strain tensor* ϵ , where

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2)$$

Lai et al. show [Lai et al., 1986] that for small values ϵ_{ii} can be interpreted as the unit elongation of a material element in the x_i direction and $2\epsilon_{ij}$, $i \neq j$, can be interpreted as the decrease in angle between two material vectors initially in x_i and x_j direction.

Note that by definition the strain tensor ϵ is symmetric. The eigenvectors \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 of ϵ are the *principal directions* of the strain, i.e., the directions where there is no shear strain. The eigenvalues λ_1 , λ_2 , and λ_3 are the *principal strains* and give the unit elongations in the principal directions. The maximum, medium, and minimum eigenvalue are called the *maximum*, *medium*, and *minimum principal strain*, respectively.

The meaning of the principal directions is illustrated in figure 2 which shows a 2D body before (solid lines) and after (dashed lines) a deformation called *pure shear*. The eigenvectors of the corresponding strain tensor at some arbitrary point are given by \mathbf{v}_1 and \mathbf{v}_2 . It can be seen that the small square region axis-aligned with \mathbf{v}_1 and \mathbf{v}_2 is deformed into a rectangle, i.e., it does not experience a shear strain.

Deformation and strain give a purely kinematic description of the motion and deformation of an elastic body without considering the internal and external forces causing it. Internal forces are body forces acting throughout the body (e.g., gravity) and external forces are surface forces acting on a real or imagined surface within the body. The surface

force at a point on such a surface is described by a *stress vector*.

Consider a plane S with normal \mathbf{n} at a point P of the elastic body as shown in figure 3. Let $\Delta \mathbf{f}$ be the force acting on a small area ΔA containing P . The stress vector \mathbf{t}_n at P is defined as

$$\mathbf{t}_n = \lim_{\Delta A \rightarrow 0} \frac{\Delta \mathbf{f}}{\Delta A}$$

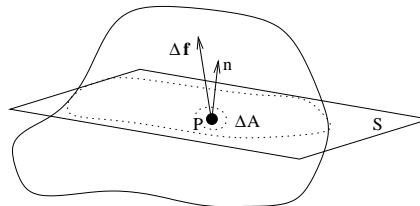


Fig. 3. Definition of a stress vector.

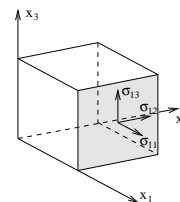


Fig. 4. Stress components of a tensor.

In classical continuum theory the resulting stress vector is the same for all surfaces through point P with a tangent plane S in P , i.e., it is independent of the surface curvature. It can be shown ([Lai et al., 1986]) that the stress vector acting on any plane with normal \mathbf{n} through P can be expressed as

$$\mathbf{t}_n = \boldsymbol{\sigma} \mathbf{n}$$

where the linear operator $\boldsymbol{\sigma}$ defines the *stress tensor* in P .

To interpret the components of the stress tensor $\boldsymbol{\sigma}$ consider an infinitesimal axis-aligned cube as shown in figure 4. The stress tensor components σ_{11} , σ_{12} , and σ_{13} are the components of the stress vector $\mathbf{t}_{\mathbf{e}_1}$. The other components of $\boldsymbol{\sigma}$ are interpreted similarly. We call the diagonal elements σ_{11} , σ_{22} , and σ_{33} the *normal stresses* and the off-diagonal elements σ_{12} , σ_{13} , σ_{23} , σ_{21} , σ_{31} , and σ_{32} the *shear stresses*. By using the conservation of momentum equation it can be shown that $\boldsymbol{\sigma}$ is symmetric for most materials [Henwood and Bonet, 1996].

As for the strain tensor the three eigenvectors of the symmetric stress tensor $\boldsymbol{\sigma}$ give the *principal directions* of the stress. The eigenvalues of $\boldsymbol{\sigma}$ give the *principal stresses*. Each principal direction gives the normal direction of a plane on which the shear stresses are zero and the normal stress is the *principal stress*.

A. Overview

A three-dimensional finite element model of the mechanical and electrical behavior of the heart has been developed by the Department of Engineering Science and the Physiology Department of the University of Auckland in collaboration with the University at San Diego, U.S., and the McGill University, Canada.

The model is based on the theory of large deformation elasticity and is solved using Galerkin and collocation techniques. Electrical activation is described by the FitzHugh-Nagumo equations and the mechanical behavior is governed by an orthotropic “pole-zero” law and a Wiener cascade model for the passive and active properties of the heart muscle (*myocardium*), respectively. Since the behavior of the myocardium is highly anisotropic the model incorporates the orientation of the muscle fibers and the fiber sheet normals. Details are described in [Hunter et al., 1993].

For our visualization purposes we use the data computed with the full heart model and don't attempt to solve the model ourselves. We therefore need to implement only a part of the model. The geometry is described by tricubic Hermite elements and the muscle fibers and sheet normal directions in the myocardium are interpolated using linear-bicubic basis functions. The use of different-order basis functions for dependent and independent variables depends on their spatial variation and continuity requirements. Note that the muscle fiber and sheet normal directions are specified with respect to the material coordinate axis (see below) so that their orientation changes consistent with the deformation of the model. An image of our model is shown in figure 5.

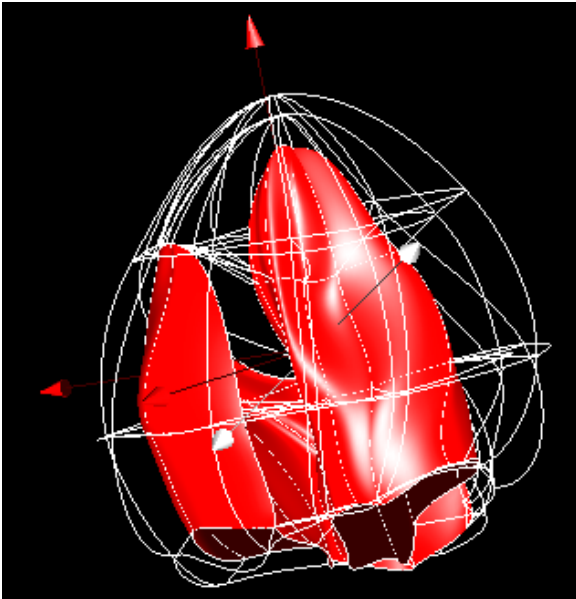


Fig. 5. A wireframe representation of the finite element model of the full heart during contraction. The element faces at the left and right ventricles are rendered as surfaces. The heart is rotated upside down.

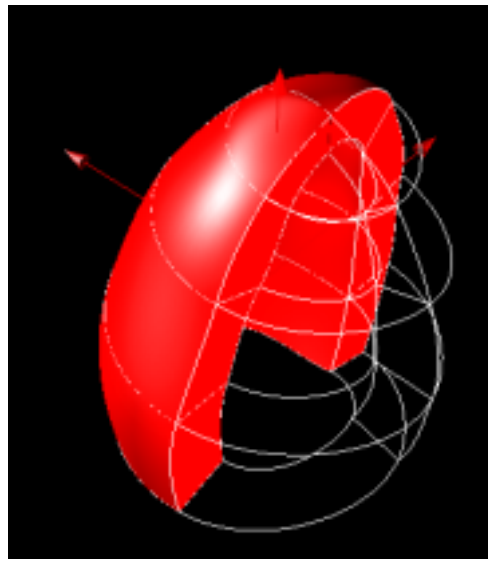


Fig. 6. The finite element model of the left ventricle at end-diastole.

Additionally we use a model of the left ventricle developed by the Physiology Department of the University of Auckland. The geometry is represented by linear-bicubic elements and was computed from tagged MRI images using B-spline snakes and surface fitting methods. An image of our model is depicted in figure 6.

B. Finite Elements

As an example of a finite element, we consider a cubic-linear element in two dimensions. We first specify a parent element (figure 7 (a)), which is a square in ξ -parameter space. The coordinates ξ_i ($0 \leq \xi_1, \xi_2 \leq 1$) are called the element or *material coordinates*. The value of some variable u (e.g., temperature) at the material coordinates ξ is then specified by interpolating the variables u_i and its derivatives in ξ_1 -direction $\left(\frac{\partial u}{\partial \xi_1}\right)_2$ ($i = 1, \dots, 4$) at the element nodes.

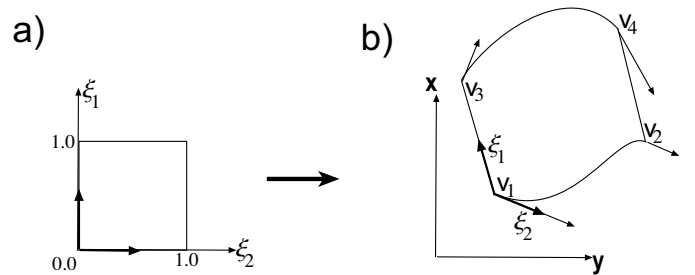


Fig. 7. The finite element model of the full heart.

The cubic-linear interpolation of u over the parameter space is then given by

$$\begin{aligned}
 u(\xi_1, \xi_2) = & \\
 & H_1^0(\xi_1)L_1(\xi_2)u_1 + H_2^0(\xi_1)L_1(\xi_2)u_2 \\
 & + H_1^0(\xi_1)L_2(\xi_2)u_3 + H_2^0(\xi_1)L_2(\xi_2)u_4
 \end{aligned} \tag{3}$$

$$\begin{aligned}
& + H_1^1(\xi_1)L_1(\xi_2) \left(\frac{\partial u}{\partial \xi_1} \right)_1 + H_2^1(\xi_1)L_1(\xi_2) \left(\frac{\partial u}{\partial \xi_1} \right)_2 \\
& + H_1^1(\xi_1)L_2(\xi_2) \left(\frac{\partial u}{\partial \xi_1} \right)_3 + H_2^1(\xi_1)L_2(\xi_2) \left(\frac{\partial u}{\partial \xi_1} \right)_4
\end{aligned}$$

where

$$L_1(\xi) = 1 - \xi, \quad \text{and} \quad L_2(\xi) = \xi \quad (4)$$

are the one-dimensional linear Lagrange basis functions, and

$$\begin{aligned}
H_1^0(\xi) &= 1 - 3\xi^2 + 2\xi^3, & H_1^1(\xi) &= \xi(\xi - 1)^2 \\
H_2^0(\xi) &= \xi^2(3 - 2\xi), & H_2^1(\xi) &= \xi^2(\xi - 1)
\end{aligned} \quad (5)$$

are the one-dimensional cubic Hermite basis functions.

The geometry of an element in world coordinates (figure 7 (b)) is then obtained by specifying the world-coordinates v_i and the ξ_1 -tangents $\left(\frac{\partial v}{\partial \xi_1} \right)_i$ ($i = 0, \dots, 4$) of the element vertices and interpolating them as above.

V. VISUALIZATION OF TENSOR FIELDS

The major sources of material failure in structural mechanics are stresses and strains. For an idealized isotropic and homogeneous material failure will occur if the maximum principal stress reaches a material dependent critical value. Biological tissue, however, is usually highly anisotropic and inhomogeneous. Complete knowledge of both the stress and the strain tensor fields is therefore necessary to understand the material behavior. For example, Aspden and Hukins [Yettram, 1989, chapter 8] relate shear strains to fatigue syndromes in bone material and Knight reports about recent research that suggests that very low or turbulent wall shear stress might be responsible for atherosclerosis [Knight, 1999].

The difficulty of interpreting tensor data arises not only from the usually large size of the data sets but also from the fact that each tensor is represented by a matrix. The aim of tensor field visualization is therefore to transform these large amount of data into a single image which can be easily understood and interpreted by the user.

A. Data Transformation and Reduction

Two useful tools in visualizing data are *data transformation* and *data reduction* (e.g., [Schroeder et al., 1996]). Data transformation retains all the information in the data but presents it in a different form. We have already presented the transformation of a tensor into its eigenvectors and eigenvalues. In 3D the tensor field is then represented by three orthogonal vector fields and three scalar fields. Another popular transformation is given by a change of coordinate systems. For example, if we represent a strain tensor with respect to the material coordinate system of our heart model its diagonal components give us the strain in the circumferential, longitudinal and radial directions.

Data reduction on the other hand extracts only partial information from the original tensor data and so gives an incomplete representation. We distinguish by the type of the reduced data, e.g., vector or scalar. Examples of *vector data* obtained by reducing tensor data are the major

or minor principal stress or strain together with the corresponding principal direction. Other useful vector quantities are the *surface traction vector* obtained by multiplying the tensor with the surface normal and the direction of the maximum shear stress [Lai et al., 1986] and shear strain [Fung, 1990].

Scalar data obtained by reducing tensor data includes: any principal stress or strain, the angle between any principal direction and a material coordinate axis or muscle fiber direction, the maximum shear stress or strain, and the strain energy per unit volume [Fung, 1990].

B. Classification of Tensor Icons

Having determined what form of data transformation or reduction is appropriate for a given tensor field, the problem is now to visualize the selected data. The visualization is by means of geometric objects with visual attributes, so-called *tensor icons*. We first present a classification of tensor icons before giving a more detailed explanation.

Our classification is a modification and extension of a framework originally introduced in [Hesslink and Delmarcelle, 1994]. Tensor icons are classified according to the covered *spatial domain* and the embodied *information level*, which we call the *information scope*. We additionally classify by the inherent *information type*. As a result we get a three-dimensional classification matrix as shown in table I. In three dimensions the *spatial domain* is either

Info. Level	Spatial Domain	Information Scope		
		Elementary	Local	Global
Full	Point	tensor ellipsoid, tensor cross, tensor glyph	degenerate point	topology
	Line	hyperstreamline		
	Surface			
	Volume			
Vector	Point	tensor line		
	Line	tensor streamline		
	Surface	spot noise		
	Volume			
Scalar	Point			
	Line			
	Surface	color mapping,		isosurface
	Volume	volume visualization		

TABLE I

TENSOR ICONS CONTAINING FULL TENSOR INFORMATION, VECTOR INFORMATION, AND SCALAR INFORMATION.

a point, a line, a surface, or a volume. The embodied *information scope* is *elementary* if the icon represents the data only across the extent of its spatial domain; *local* if in addition derivative information (e.g., gradient, curl) is displayed; and *global* if the icon represents the structure of the whole tensor field. The *information type* gives the type of information displayed after data reduction, in our case scalar, vector, or tensor.

VI. THE VISUALIZATION TOOLKIT

We programmed our visualization toolkit in OpenGL using FLTK, a LGPL'd C++ graphical user interface toolkit

for X (UNIX), OpenGL, and WIN32 (Microsoft Windows NT 4.0, 95, or 98) [Spitzak, 1999]. Our visualization toolkit has four essential components explained in the following subsections: a unified field data structure to achieve optimal data exploration, 3D user interaction, a user interface to design specialized color maps, and finally a variety of visualization methods for tensor, vector, and scalar fields.

A. The Field Data Structure

The previous section suggested that a tensor field can be visualized by transforming it or by reducing it to a vector field or a scalar field. We are also interested in other fields not derived from the tensor field such as the fiber field, the displacement vector field, and the electrical potential field. A unified approach to represent, manipulate, and compare these diverse field structures is therefore desired.

We achieve this goal by using an object-oriented approach. The field class is first subclassed into a tensor field, a vector field, a scalar field, and a fiber field class. The first three of these fields can be either *defined fields* or *derived fields*. With each defined field we associate a sampling grid and a set of interpolation functions. As mentioned in section IV the interpolation functions chosen depend on the spatial variation and continuity requirements of the field.

Usually the field is defined at the nodes of the underlying finite element model. However, a field can also be defined over a different grid. For example, the strain field for the left ventricle is defined at a regular subsampling grid over each finite element and is interpolated trilinearly over that grid.

A derived field, on the other hand, is associated with a parent field and contains a function defining how a field value is derived from the corresponding parent field. The advantage of this construction is twofold:

- we eliminate problems with the interpolation of derived values. For example, directly interpolating the eigenvalues of a tensor over a finite element gives usually the wrong results. Instead we rather interpolate the tensor and compute the eigenvalues from the resulting tensor.
- we can combine arbitrary fields through arithmetic functions (e.g., the difference between two scalar fields) even if they are defined over different grids. Similarly, we can interactively derive new fields by choosing a parent field for a derived fields.

Examples of derived fields are eigenvalue fields and eigen-vector fields (in both cases we can specify whether we want the major, medium, or minor eigenvalue or eigenvector), vector length fields, vector angle fields (specifying the angle with any of the material axis), gradient fields, and vector and tensor component fields. We are usually interested in the components of a tensor with respect to the material coordinate system so that a basis transformation is performed if the tensor is defined with respect to a different coordinate system.

The only disadvantage of our field structure is that the computation of a derived field value is slower than if the field were directly defined at the nodes of the finite element

mesh.

This class hierarchy could also be extended by analytic fields, which are directly defined as a analytic functions over the domain of the finite element model. Up to now, however, we haven't found any application of such a field for our visualization task.

B. User Interaction in 3D

Two types of user interaction are essential in a visualization toolkit. Firstly the user wants to rotate, translate, and zoom into the model. This is achieved in OpenGL using a standard camera model. Rotation is performed using a virtual trackball. Movement of the mouse cursor in the window is interpreted as movement on the surface of a virtual sphere enclosing the window. Zooming can be performed by either changing the distance of the camera to the origin or by selecting a subwindow. We found that in the latter case the user prefers the trackball origin to be shifted in a way such that its projection lies in the center of the new window.

The second important type of user interaction is the selection and movement of objects in the 3D domain. Objects are selected using the standard OpenGL mechanism [Woo et al., 1997]. We then move them by translating the mouse movements on the screen into an object movement parallel to the view plane. This is achieved by determining the z-buffer value of the selected object. The new mouse position is then projected back into the 3D domain so that the z-buffer value stays constant.

Many people are used to moving objects in 3D by interacting with the orthogonal projections of the model into the three coordinate planes. Our method is a generalization of this. However, a similar effect is obtained by aligning the yz-, xz-, and xy-plane of the coordinate system with the view plane which is achieved by pressing the x,y, and z-key, respectively.

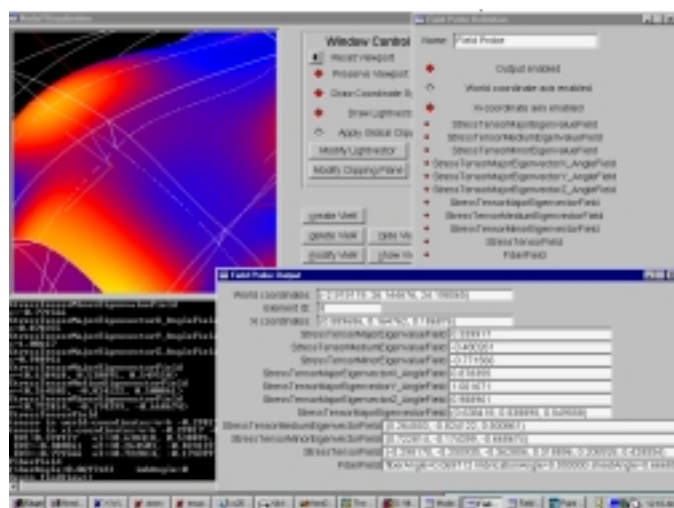


Fig. 8. Zoom into the full heart model showing the tensor probe and the values of a variety of fields at that point.

We use these principles to construct a field probe. The

user can move the field probe through the 3D domain. At any point the probe shows the current world coordinates, the current element it is in (if any), the associated material coordinates, and the current ξ -coordinate system axis. Additionally, the values of all enabled fields at that point are shown (figure 8).

Note, that since the probe gets moved in world coordinate a multi-dimensional Newton method [Press et al., 1992] must be employed to find the corresponding material coordinates. The start point of the search is given by the element number and ξ -coordinate of the last position of the probe. If the returned ξ -coordinate lies outside the element we determine the face intersected by the line connecting the old point and new point in parameter space. By using a precomputed adjacency matrix we then obtain the neighboring element and continue the search in it using the intersection point with the face as the start point of the Newton search. We found that on average only three iterations are necessary to determine the material coordinates of the probe when moving it through the model. The Newton method fails close to the apex of our models since the finite elements are degenerate at that location. For this special case we employ a simple hack.

C. Color Mapping

A popular method to visualize a scalar field over a one-dimensional or two-dimensional domain is color mapping (e.g., [Cox, 1988]). In this technique the range of scalar field values is associated with a color spectrum. Figure 9 shows two popular spectra. The top spectrum interpolates linearly between saturated yellow, saturated red, and saturated blue. We found that the visualizations obtained with this spectrum often had too much red content which made the interpretation of the results difficult. An improved spectrum is shown at the bottom of figure 9. This spectrum is the same as the previous one, except that on the left hand side it is extended by a linear interpolation between saturated yellow and yellow-white and on the right side the spectrum is extended by a linear interpolation between saturated blue and blue-black.

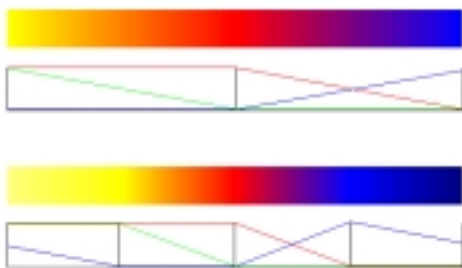


Fig. 9. Color spectra for the visualization of scalar fields.

The most common use of colour mapping is to visualize

the scalar field on a surface intersecting the domain. We prefer taking a surface orthogonal to one of the material coordinate axis. Color mapping involves subdividing the surface into polygons and computing the scalar field values at the vertices of the polygons. These values are then mapped into the colour spectrum. The resulting polygons are rendered using standard graphics hardware. An example is given in figure 10.

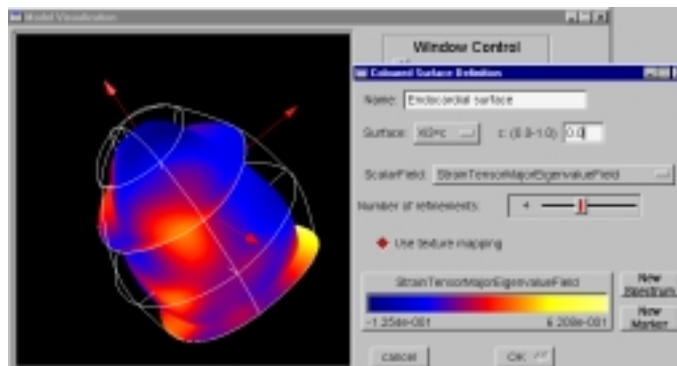


Fig. 10. A predefined colour map is used to visualize the maximum principal stress on the endocardial surface of the left ventricle and on two surfaces orthogonal to the longitudinal material axis.

The user has the choice to enable or disable lighting. For most application disabling of the lighting is preferred since otherwise it would be hard to determine the mapped colour. However, lighting is important for perception of the 3D geometry of an object. A lit colour mapped surface is used for the construction of hyperstreamlines, which are explained in subsection VI-D.2.

Defining colour mapped objects by specifying vertex colours results in the graphics hardware using linear interpolation (Gouraud shading) to render the polygon. This can result in shading artifacts as shown in figure 11 (a) or, worse, in omitted colors. The problem becomes especially obvious when using cyclical colour maps which are introduced in the next paragraph. The problem can be reduced by subdividing the object surface into more polygons. This, however, is very inefficient. Rather we define a colour map as a one-dimensional texture map. For each polygon vertex we define a texture coordinate and render the polygons using texture mapping hardware. We found that 1024 texels, each consisting of three floats for the RGB colour values, were sufficient in all our applications. Using an OpenGL graphics card the rendering times using texture mapping did not change. The resulting improvement is demonstrated in figure 11 (b). Note that texture mapped polygons can still be lit in OpenGL by specifying them as white for the shading algorithm and then blending them with the texture colour.

A further advantage of using 1D texture maps is that we can insert markers into them. The markers are then visible as isocontour-like structures on the colour mapped surface as illustrated in figure 12.

Often, predefined colour maps are not suited for a particular visualization task, e.g., because their color interferes

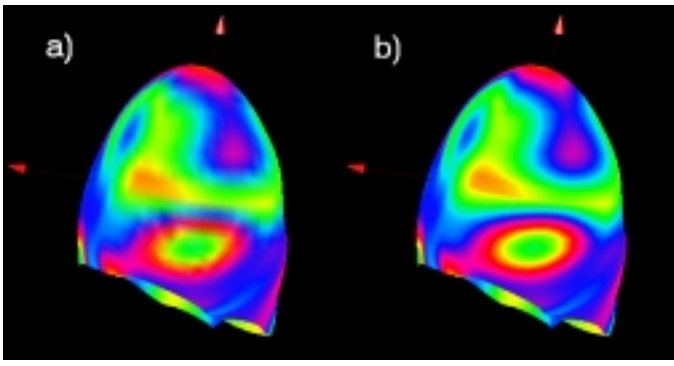


Fig. 11. A colour mapped surface using Gouraud shading (a) and texture mapping (b).

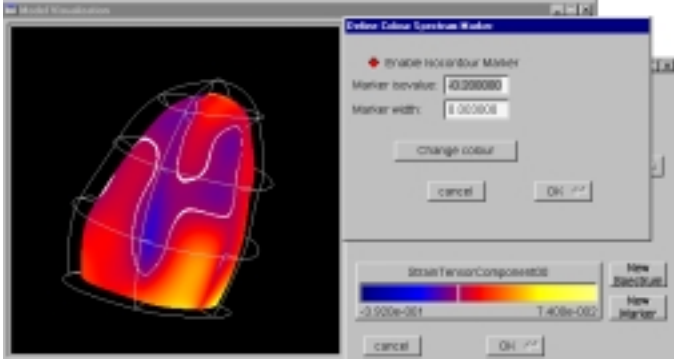


Fig. 12. The circumferential strain at a surface inside the myocardium is visualized using a colour map and a marker.

with other visualization icons. Our toolkit therefore allows the specification of arbitrary piecewise linear colour maps (the right-hand side window in figure 13). It is also possible to construct cyclical colour maps, where several cycles of the colour spectrum are mapped over the range of a scalar field. We have found cyclical colour maps especially useful when trying to understand the fine structure of a scalar field. Figure 13 visualizes the maximum principal stress using a colour mapped surface and additionally indicates the fibre direction and fiber sheet normal direction by red and blue arrows, respectively. Another feature of our toolkit is to allow a logarithmic mapping into the scalar field range. This proved useful in structural mechanics, e.g., crack propagation, where extreme stress values occur resulting in a visualization with large surface areas of the same color.

We are also planning to define colour maps by letting the user define a spline curve in a 2D or 3D colour space (RGB, HSV, or CIE colour space). The spline curve is then parameterized with its arclength [Hoschek and Lasser, 1992] and sampled at regular intervals resulting in a 1D texture map usable with our color mapping algorithm. Initial consultation with the intended user group also showed that it is desirable to use the same colour map for different scalar fields. For example, in our case the user wanted to compare the strain values in longitudinal, circumferential, and

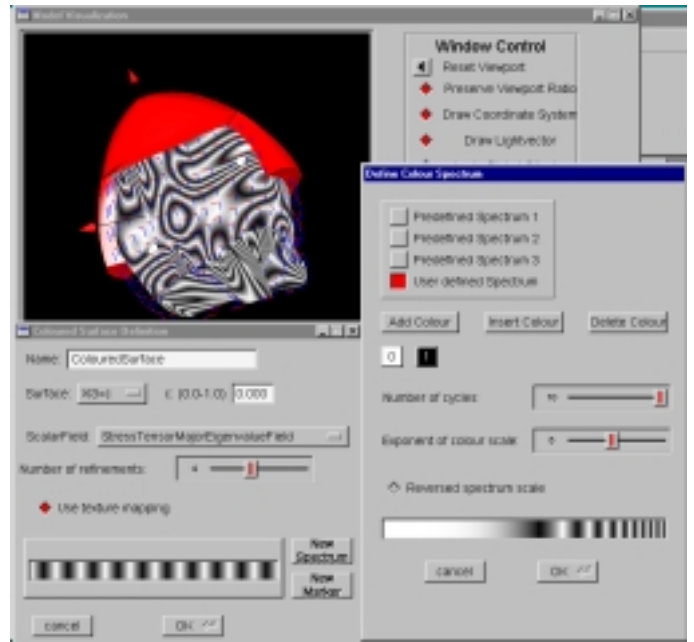


Fig. 13. The image shows some of the finite elements representing the outer layer of the heart model in red. The direction of the muscle fibers and the direction of the fiber sheet normals is indicated by red and blue arrows, respectively. The maximum principal stress in the middle of the myocardium is visualized using a cyclical colour map.

radial direction defined over different surfaces.

D. Visualization Icons

In the previous sections we introduced a classification of tensor icons, a unified field structure, and colour maps. In this section we show how our visualization toolkit incorporates these principles in order to visualize tensor fields.

D.1 Tensor Icons Containing Scalar Information

Several examples of colour mapped surfaces visualizing derived scalar fields, such as the maximum principal stress and the strain in circumferential direction, have already been given in the previous section.

The *c-isosurface* of a scalar field s is defined as all points \mathbf{x} for which $s(\mathbf{x}) = c$. Isosurfaces give a rough idea of the overall distribution of the scalar field and are best combined with other visualization tools in order to achieve optimal results. Figure 14 shows the 0.23-isosurface of the radial strain together with the radial strain on the endocardial surface. The isosurface encloses the area in the heart muscle with wall thickening rates higher than 23%.

D.2 Tensor Icons Containing Vector Information

When visualizing vector quantities we must distinguish between signed and unsigned vector fields. Most traditional vector fields, such as the velocity and the displacement vector field are signed, and we can therefore visualize them by small arrows.

The sign of an eigenvector, on the other hand, is indeterminate with equation 1. We therefore visualize an

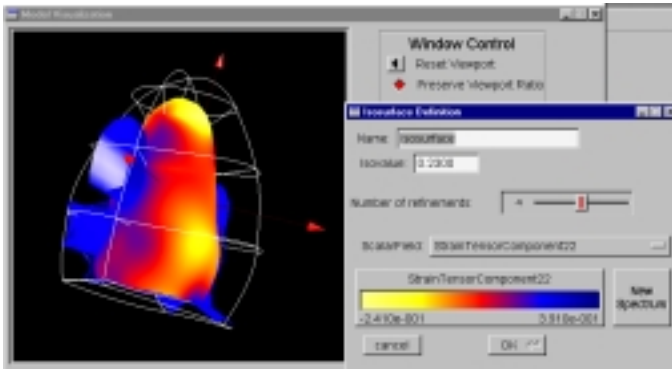


Fig. 14. The 0.23-isosurface of the radial strain together with the radial strain on the endocardial surface.

eigenvector by a simple line. The direction and length of this line is given by the eigenvector and the corresponding eigenvalue, respectively. The same icon can be used for other derived vector data with indeterminate sign, such as the direction of the maximum shear stress.

The obvious disadvantage of using vector icons at discrete points is that using too few might fail to reveal interesting areas of the tensor field, but using too many leads to visual cluttering which makes interpretation difficult. A continuous representation of a vector field \mathbf{v} is achieved by a streamline which is an integral curve $\mathbf{x}(s)$ satisfying

$$\frac{d\mathbf{x}}{ds} = \mathbf{v}(\mathbf{x}) \quad (6)$$

An unsigned vector field, such as an eigenvector field, is visualized in the same way, except that we have to create a streamline for the positive and negative direction of the unsigned vector field. The resulting streamline is color coded with an appropriate scalar field: for example, if using an eigenvector field we would use the corresponding eigenvalue field. An example is given in figure 15 which shows streamlines in the direction of the medium principal stress.

One problem with streamlines is the difficulty in placing them. If they are too close together they merge and the directional information is obscured. If the tensor streamlines are too far apart information is missing for large areas. One solution is *spot noise* which is generated by distorting a noise texture with the derived vector field [de Leeuw and van Wijk, 1995]. An example in two-dimensions is shown in figure 16.

D.3 Tensor Icons Containing Full Tensor Information

Tensor ellipsoids are constructed by defining the directions and lengths of the principal axes of an ellipsoid as the eigenvectors and eigenvalues of the tensor, respectively. We additionally visualize the sign of each eigenvalue by using blue for a negative value and red for a positive value. An example is given in figure 17. The image illustrates clearly that the strain in the myocardium is usually compressive in circumferential and longitudinal direction and it is usually

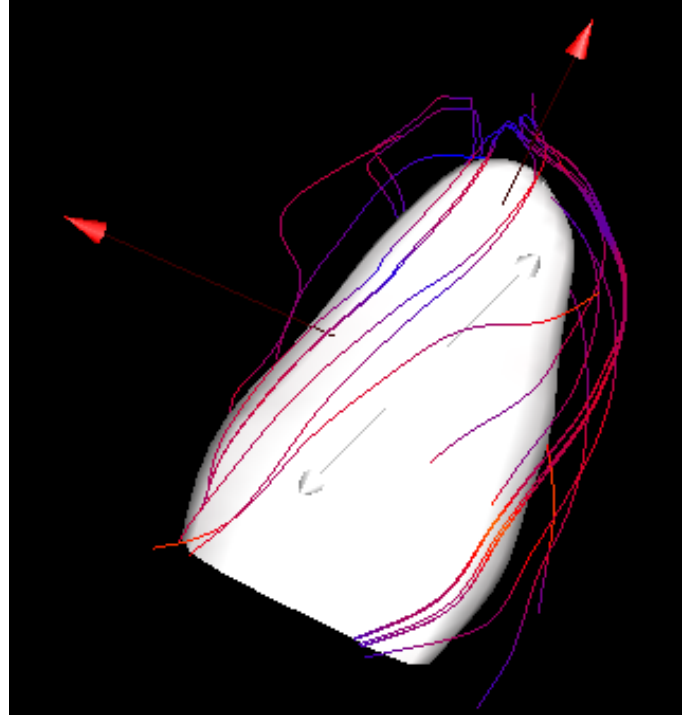


Fig. 15. Direction and value of the medium principal strain visualized by color mapped streamlines.

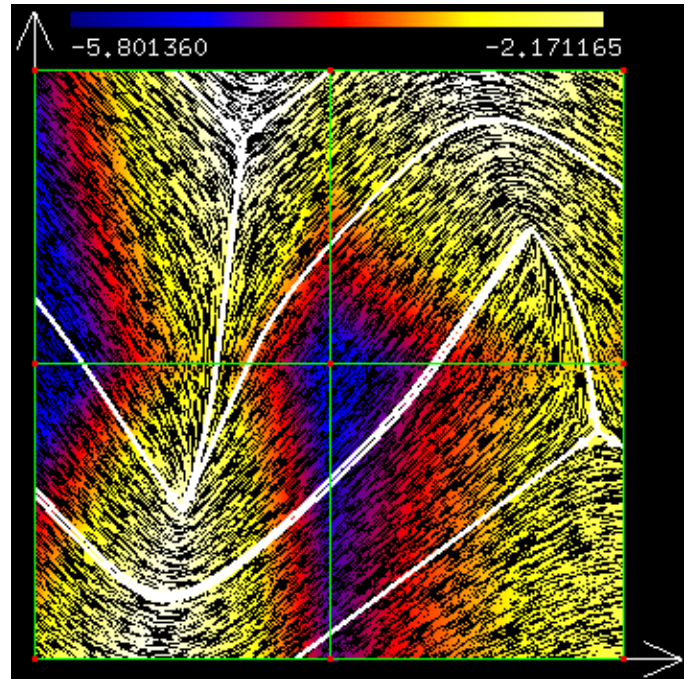


Fig. 16. Topology of a stress field with maximum principal stress visualized by spot noise.

tensile in radial direction. More sophisticated tensor icons are given by Haber [Haber, 1990].

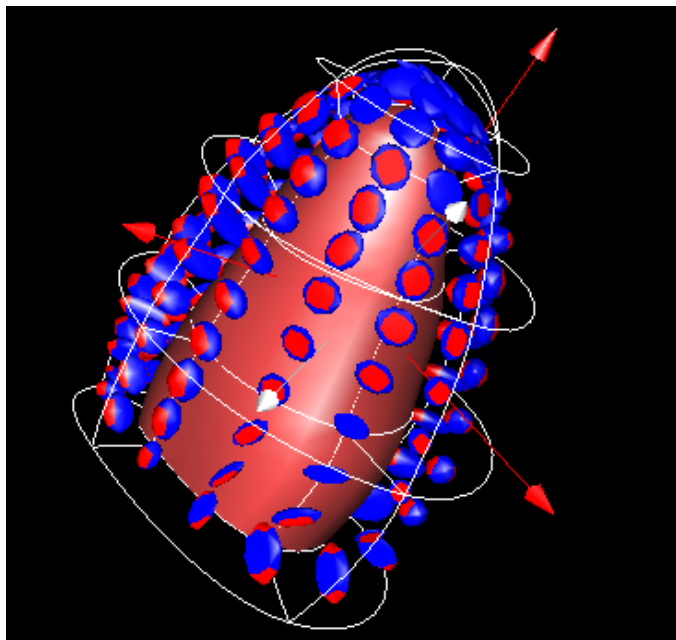


Fig. 17. Tensor ellipsoids at regular sampling points over each element.

The visual clutter of the ellipsoidal tensor icons can be reduced by using a *hyperstreamline* [Hesselink and Delmarcelle, 1994]. The trajectory of a hyperstreamline is a tensor streamline of an eigenvector field. The other two eigenvectors and corresponding eigenvalues define the axes and lengths of the ellipsoidal cross section of the hyperstreamline. The eigenvalue corresponding to the eigenvector defining the trajectory is color mapped onto the surface of the hyperstreamline. Figure 18 shows that the complete information about the eigenvector fields along the trajectory is revealed.

Degenerate points are points for which at least two eigenvalues of the tensor are equal. It can be shown that in this case all vectors in a plane are eigenvectors. Degenerate points are therefore the only points where the trajectories of an eigenvector field can cross. Delmarcelle shows that the derivatives of the tensor components can be used to derive the direction in which *separatrices* emanate from the degenerate point [Delmarcelle, 1994]. The separatrices are trajectories of the major eigenvector field separating it into regions of similar behavior and therefore define the *topology* of the tensor field. A two-dimensional example is given in figure 16. The separatrices are drawn in white and the directions in which separatrices emanate from a degenerate point are indicated by white arrows. The major eigenvector field is additionally visualized by spot noise. Information about the minor eigenvector field can be obtained by using hyperstreamlines as separatrices.

As a new visualization method we present tensor particles. Distributing the particles randomly over the domain and color coding them gives a good impression of a scalar

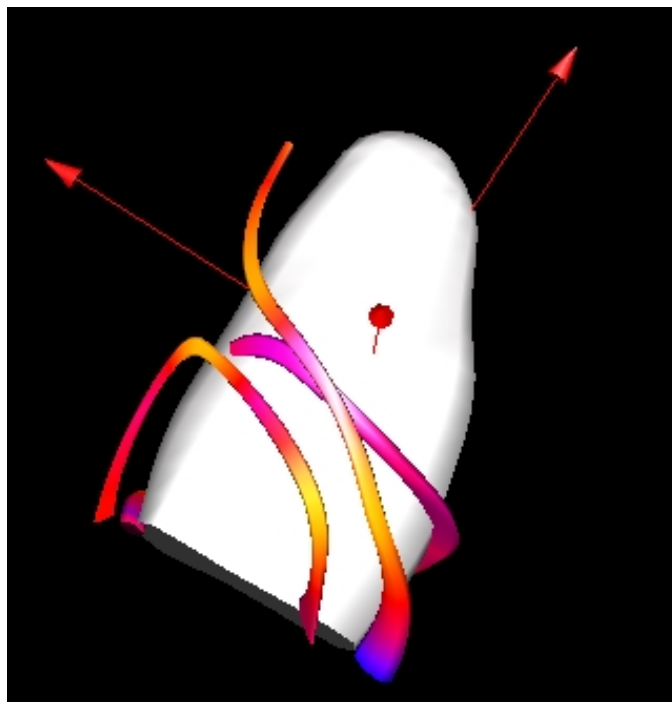


Fig. 18.

field. By using tensor particles, figure 19 reveals that the maximum strain in the circumferential direction occurs at the bottom and on the outside of the ventricular wall.

Moving the particles along the gradient of a principal stress field and constraining them to user defined isovalues defines isosurfaces. Currently we are implementing interactive methods, similar to that defined for the tensor probe, to select particles and to expand them into tensor ellipsoids or, by tracing, into tensor streamlines and hyperstreamline.

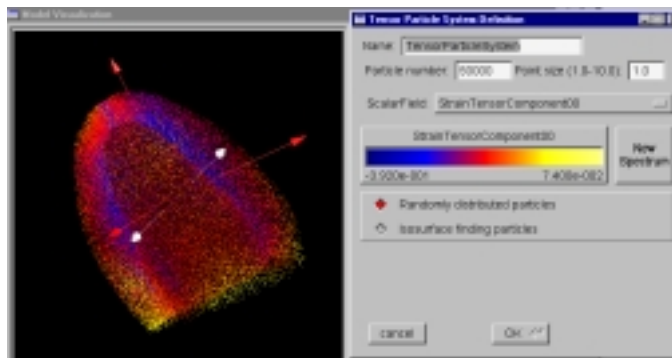


Fig. 19. Circumferential strain throughout the myocardial wall visualized by tensor particles.

VII. CONCLUSION

We have presented a simple introduction to stress and strain fields as they occur in elastic bodies. An understanding of the stresses and strains in a body is important in medicine and biomechanics since they are a major source

of organ failure and are necessary for the understanding of organ function [Fung, 1990], [Knight, 1999].

Interpreting large amounts of tensor data is difficult, though, since a second-order tensor is equivalent to a matrix. As a solution we have presented a visualization toolkit which contains a wide variety of visualization techniques, allows interactive data exploration and the derivation of new data fields, and provides a flexible color mapping.

We have demonstrated our work by visualizing the stress in a model of the heart and the strain in a model of the left ventricle. Visualizing the stress and strain fields in the heart can be used for diagnosis, surgical planning, and the evaluation of the success of surgery.

REFERENCES

- [Cox, 1988] Cox, Donna J. (1988). Using the supercomputer to visualize higher dimensions: An artist's contribution to scientific visualization. *Leonardo*, 22:233 – 242.
- [de Leeuw and van Wijk, 1995] de Leeuw, Willem C. and van Wijk, Jarke J. (1995). Enhanced spot noise for vector field visualization. In Nielson, Gregory M. and Silver, Deborah, editors, *Proceedings of Visualization '95*, pages 233 – 239, Los Alamitos, California. IEEE, Computer Society Press.
- [Delmarcelle, 1994] Delmarcelle, Thierry (1994). *The Visualization of Second-Order Tensor Fields*. PhD thesis, Stanford University. URL: <http://www.nas.nasa.gov/NAS/TechReports/RelatedPapers/StanfordTensorFieldVis/DelmarcelleThesis/abstract.html>.
- [Fung, 1990] Fung, Y. C. (1990). *Biomechanics - Motion, Flow, Stress, and Growth*. Springer Verlag New York Inc.
- [Haber, 1990] Haber, R. B. (1990). Visualization techniques for engineering mechanics. *Computing Systems in Engineering*, 1(1):37 – 50.
- [Henwood and Bonet, 1996] Henwood, David and Bonet, Javier (1996). *Finite Elements - A Gentle Introduction*. MacMillan Press Ltd.
- [Hesselink and Delmarcelle, 1994] Hesselink, Lambertus and Delmarcelle, Thierry (1994). Visualization of vector and tensor data sets. In et. al., L. J. Rosenblum, editor, *Scientific Visualization: advances and challenges*, pages 419 – 433. Academic Press, London.
- [Hoschek and Lasser, 1992] Hoschek, Josef and Lasser, Dieter (1992). *Fundamentals of Computer Aided Geometric Design*, chapter 14, pages 572 – 601. AK Peters Ltd., Wellesley, MA 02181, second edition.
- [Hunter et al., 1993] Hunter, Peter J., Nielsen, Poul M. F., Smaill, Bruce H., LeGrice, Ian J., and Hunter, Ian W. (1993). An anatomical heart model with applications to myocardial activation and ventricular mechanics. In Pilkington, Theo C., editor, *High-Performance Computing in Biomedical Research*, chapter 1, pages 3–26. CRC Press.
- [Knight, 1999] Knight, Jonathan (1999). Cunning plumbing. *New Scientist*, (2172):32–37.
- [Lai et al., 1986] Lai, W. Michael, Rubin, David, and Krempl, Erhard (1986). *Introduction to Continuum Mechanics*, volume 17 of *Pergamon Unified Engineering Series*. Pergamon Press, Headington Hill Hall, Oxford OX3 0BW, England, revised edition in si/metric units edition.
- [Parke and Waters, 1996] Parke, Frederic I. and Waters, Keith (1996). *Computer Facial Animation*. A K Peters Ltd., Wellesley, Massachusetts.
- [Press et al., 1992] Press, William H., Vetterling, William T., Teukolsky, Saul A., and Flannery, Brian P. (1992). *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, second edition.
- [Sagar et al., 1994] Sagar, Mark A., Bullivant, David, Mallinson, Gordon D., Hunter, Peter J., and Hunter, Ian W. (1994). A virtual environment and model of the eye for surgical simulation. In Glassner, Andrew, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 205–213. ACM SIGGRAPH, ACM Press. ISBN 0-89791-667-0.
- [Schroeder et al., 1996] Schroeder, Will, Martin, Ken, and Lorensen, Bill (1996). *The Visualization Toolkit: An Object-Oriented Ap-*

proach to 3D Graphics. Prentice Hall PTR, Upper Saddle River, New Jersey 07458.

[Spitzak, 1999] Spitzak, Bill (1999). The FLTK home page. URL: <http://www.ftk.org>.

[Woo et al., 1997] Woo, Mason, Neider, Jackie, and Davis, Tom (1997). *OpenGL Programming Guide*. Addison-Wesley Publication Company Inc, 3 edition.

[Yettram, 1989] Yettram, A. L., editor (1989). *Material properties and stress analysis in biomechanics*. Manchester University Press.