# CT Image Segmentation Using FEM with Optimized Boundary Condition

**Hiroyuki Hishida[1], Hiromasa Suzuki[2], Takashi Michikawa[2], Yutaka Ohtake[1], Satoshi Oota[3]***

[1] Department of Precision Engineering, School of Engineering, The University of Tokyo, Bunkyo-ku, Tokyo, Japan, [2] Research Center for Advanced Science and Technology, The University of Tokyo, Meguro-ku, Tokyo, Japan, [3] RIKEN BioResouce Center, Tsukuba, Ibaraki, Japan

## Abstract

The authors propose a CT image segmentation method using structural analysis that is useful for objects with structural dynamic characteristics. Motivation of our research is from the area of genetic activity. In order to reveal the roles of genes, it is necessary to create mutant mice and measure differences among them by scanning their skeletons with an X-ray CT scanner. The CT image needs to be manually segmented into pieces of the bones. It is a very time consuming to manually segment many mutant mouse models in order to reveal the roles of genes. It is desirable to make this segmentation procedure automatic. Although numerous papers in the past have proposed segmentation techniques, no general segmentation method for skeletons of living creatures has been established. Against this background, the authors propose a segmentation method based on the concept of destruction analogy. To realize this concept, structural analysis is performed using the finite element method (FEM), as structurally weak areas can be expected to break under conditions of stress. The contribution of the method is its novelty, as no studies have so far used structural analysis for image segmentation. The method's implementation involves three steps. First, finite elements are created directly from the pixels of a CT image, and then candidates are also selected in areas where segmentation is thought to be appropriate. The second step involves destruction analogy to find a single candidate with high strain chosen as the segmentation target. The boundary conditions for FEM are also set automatically. Then, destruction analogy is implemented by replacing pixels with high strain as background ones, and this process is iterated until object is decomposed into two parts. Here, CT image segmentation is demonstrated using various types of CT imagery.

**Citation:** Hishida H, Suzuki H, Michikawa T, Ohtake Y, Oota S (2012) CT Image Segmentation Using FEM with Optimized Boundary Condition. PLoS ONE 7(2): e31116. doi:10.1371/journal.pone.0031116

**Editor:** Boris Rubinsky, University of California, Berkeley, United States of America

**Received** May 10, 2011; **Accepted** January 3, 2012; **Published** February 28, 2012

**Funding:** This work was funded by MEXT Global COE Program ''GlobalCenter of Excellence for Mechanical Systems Innovation'', JSPS Grant-in-Aid for Scientific Research (A), 22246018, and MEXT Grant-in-Aid for Scientific Research on Innovative Areas, 1201. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

* E-mail: hishida@den.rcast.u-tokyo.ac.jp

## Introduction

We propose a CT image segmentation method using structural analysis that is useful for objects with structural dynamic characteristics. Motivation of our research is from the area of genetic activity. In order to reveal the roles of genes, it is necessary to create mutant mice and measure differences among them, in particular, morphological difference in their skeletons. X-ray CT is used to scan a skeleton then its image is manually segmented into pieces of the bones. Considering the number of genes involved, it is desirable to make this segmentation procedure automatic. A problem specific to this segmentation is to separate bones at their joints where the gap between the bones are not necessarily clear because of their complex tissue structures. Despite the numerous image segmentation methods, none of them matches our objectives.

Accordingly, we previously proposed a method to segment CT images using structural analysis [1]. The technique is based on the assumption that the interference area (joint) between components (bones) is structurally weak. We compute strain, which tends to be large in structurally weak areas and segment the image in the region of high strain. In previous work [1], we used a commercial software VOXELCON [2], which is an image-based structural analysis system. In the approach examined, we set physical properties for every pixel and create a stiffness matrix. Then, we calculated the von Mises strain of every pixel and removed the one with the highest strain value, assuming it will be broken. We iterated strain calculation and removal until the input image is decomposed into multiple parts. The method could be used to segment low-contrast CT images - a task that was difficult with conventional methods. This technique was based on the use of structural analysis for CT image segmentation. However, it was necessary with this approach to set boundary conditions manually in structural analysis. Thus trial and error is required to apply the method.

In this paper, we expand our previous work [1] to set parameters and loading conditions in a semiautomatic way. We set regions of interest (ROIs) as candidates for weak areas, which include correct and erroneous segmentation targets. Assuming strain as an objective function, we calculate boundary conditions for each ROI using a gradient ascent method [3]. In other words, we calculate strain for all ROIs. Then, we choose the ROI with the highest strain as the correct candidate for a weak area, and these conditions are then used to segment the image.

The main advantage of the proposed approach is its reliability, as the structural analysis it involves means that appropriate object segmentation in structurally weak areas is guaranteed. As this method is designed to segment joints of a skeleton and thus prefers such objects with shapes of relatively high aspect ratio around the segmentation boundaries. The originality of this research lies in its introduction of the analogy of mechanical destruction to image processing and its optimization of boundary conditions in the finite element method (FEM).

## Related work

Numerous papers on image segmentation have been published. Most image segmentation is performed using image processing technology, but the proposed approach stands apart from conventional methods because we introduce the concept of structural analysis.

Thresholding is the simplest method of segmentation, and involves the assumption that CT values are related to an object's density. Adaptive thresholding is used in [4], but it requires parameter tuning. It is also difficult for the method to produce error-free results, since the CT values of pixels are influenced by the density of neighboring areas.

Kass et al. [5] proposed the snakes algorithm, which is an active contour method that minimizes spline energy for segmentation. This approach assumes that the target object is smooth and has strong edges. As the snakes algorithm approach is a powerful tool for CT image segmentation, there are many papers on it, such as [6,7]. However, if the edges of bones are weak or the distance between bones is minimal, the approach fails in image segmentation.

Adams et al. [8] proposed the seeded region growing method. This technique involves propagating an area of given seeds using pixel values and distances. Although the application of region growing to image segmentation has been widely reported, such as [9,10], and problems remain with regard to the difficulty of selecting the right seeds.

Vincent et al. [11] introduced the watershed approach, which uses pixel gradient values to extract regions in the same way as water flooding. The method assumes that neighboring pixels with similar values should be labeled as parts of the same cluster. It is a powerful technique on which many works have been published, such as [12,13]. However this approach has the disadvantage of over-segmentation, and also we cannot apply a cost function optimized for one dataset to other datasets, as objects in biological application are variable and not uniform.

Boykov et al. [14] introduced the graph-cut algorithm, which uses the max-flow min-cut theorem [15] to segment graph edges. In this approach, the cost function is set appropriately from the target in graph representation. Since graph cut is a powerful method, many related papers have been published, such as [16,17]. [17] includes a survey on segmentation for parts of living things. However, it is not easy to set a cost function appropriately and with high versatility, and a cost function optimized for one dataset cannot be applied to other datasets, as with the watershed method.

FEM is used in image processing in different contexts. First of all, in the field of bio-mechanics it is a common approach to apply FEM analysis with CT or MR images [18,19]. In the most cases they use scanned images for creating mesh models for applying 3D FEM analysis and the image segmentation was done with some conventional methods. Recently, Auer and Gasser [20] proposed FEM-based deformable models to reconstruct aneurysms from CT images. It can be considered as an active contour based segmentation. The difference from our approach is that we apply FEM analysis directly to the image to obtain the result in the form of an image with which segmentation is made. In [21], they proposed a variational approach to image segmentation with a convex variational model and use FEM to numerically solve the problem. Although their work could be considered comparable to our approach, the principle underlying their segmentation method is completely different from ours. Thus, the authors of conventional methods use FEM, however their approaches are different from ours.

## Review of FEM [22]

Our segmentation method is based on an idea that a structure can be broken along its structurally weak part which can be detected by computing *strain* distribution generated by applying some external load to the structure [22]. For instance, the top right image of Fig. 1 shows such loads in red arrows. Under such loading condition, the structure elastically deforms. The magnitude of the deformation varies place to place according to the shape of the structure and the material strength at the location. Such deformation can be measured as value called *strain* which

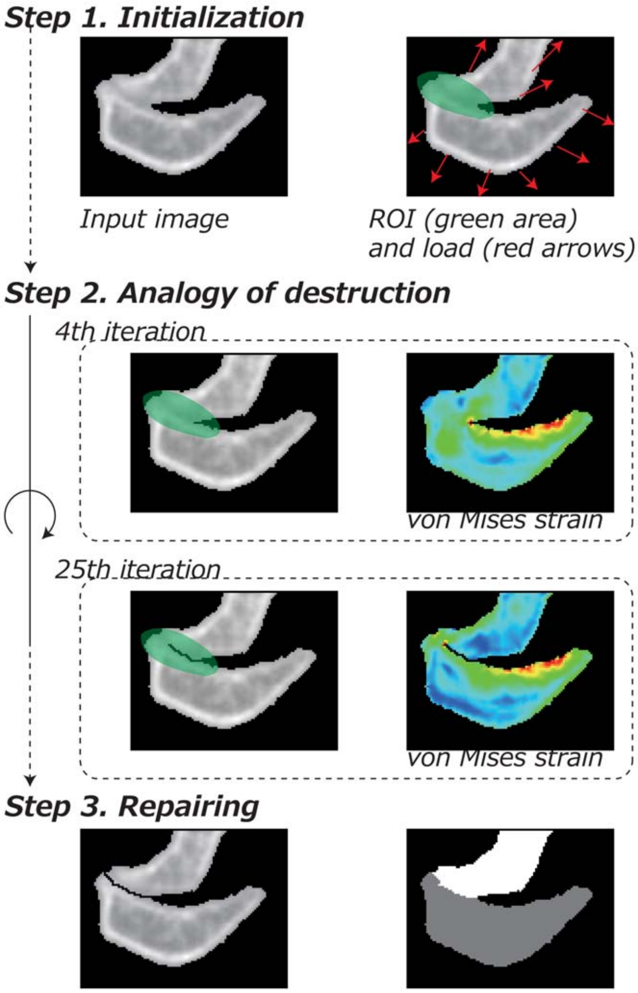

**Figure 1. Flow chart of the proposed segmentation algorithm.** Green area in the top and left middle are ROI and the red arrows show some representatives of loading forces. Images in the right and middle show distribution of von Mises strain and red pixels indicate high value. Using analogy of destruction we repeat removing these red pixels until the object gets separated along the interference region. In the bottom, the result of segmentation is shown where the removed pixels are salvaged to be included in the segments.
doi:10.1371/journal.pone.0031116.g001

represents local and net deformation (without rigid transformation) at each point of the structure. In the middle right images of Fig. 1 show the strain distribution under the loading condition. A physical structure breaks at the place with large strain which exceeds its limit of the material.

The strain distribution can be numerically computed using FEM by solving the static structural problem. In the following sections, we briefly review FEM and also introduce Pixel based FEM [23] which is an application of FEM directly to images. For more details about FEM, we refer the reader to such a text book as [22].

**Finite elements.** In FEM, the structure is decomposed into a set of *finite elements*. Among various types of the finite elements we use quadrilateral elements as shown in Fig. 2. In this figure, (a) shows an object which is decomposed into a set of quadrilateral elements. For each element, four nodes (corners) $v_1, v_2, v_3, v_4$ are defined as shown Fig. 2(b). By applying external force $\boldsymbol{f}_i$ to the node $v_i$, the deformation causes some displacement $\boldsymbol{u}_i$ to $v_i$. $\boldsymbol{f}_i$ and $\boldsymbol{u}_i$ are a two dimensional vector of force and displacement respectively. A linear equation can be derived to represent the relationship between the displacement at these nodes and the forces:

$$\boldsymbol{K}_e \boldsymbol{u}_e = \boldsymbol{f}_e, \tag{1}$$

where $\boldsymbol{u}_e$ and $\boldsymbol{f}_e$ are defined as:

$$\boldsymbol{u}_e = (\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4)^T,$$

$$\boldsymbol{f}_e = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4)^T,$$

$\boldsymbol{K}_e$ is called *elemental stiffness matrix* that is constant for the element. The derivation of $\boldsymbol{K}_e$ will be given later in this section.

**Strain-displacement relationship.** The application of external forces at those four nodes causes displacement $(\alpha, \beta)$ also at each point inside the element. And the strain $\boldsymbol{\epsilon} = (\epsilon_x, \epsilon_y, \gamma_{xy})^T$ at this point is defined by derivatives of $(\alpha, \beta)$ w.r.t. $x$ and $y$ as follows:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{2}$$

where $\epsilon_x$ denotes the strain in the $x$ direction, $\epsilon_y$ is that in the $y$ direction, and $\gamma_{xy}$ is the shear of strain.

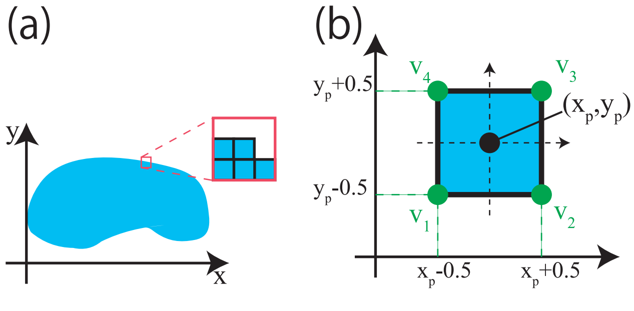

**Figure 2. Node definition.** (a) is Finite elements and (b) is Nodes of an finite element. The blue area in (a) is an object represented with a set of quadrilateral elements. (b) shows four corner nodes of an element.
doi:10.1371/journal.pone.0031116.g002

The displacement $(\alpha, \beta)$ can be estimated by interpolating the displacements at the four nodes $\boldsymbol{u}_e = (\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4)^T$. By using a bi-linear interpolation to estimate $(\alpha, \beta)$ from $\boldsymbol{u}_e$, the strain can be represented by the following linear equation.

$$\boldsymbol{\epsilon} = \boldsymbol{B}\boldsymbol{u}_e. \tag{3}$$

The matrix $\boldsymbol{B}$ is called *strain-displacement matrix.*

**Stress-strain relationship.** The relationship between stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\epsilon}$ is obtained using the widely known Hooke's Law as follows:

$$\mathbf{s} = \boldsymbol{D}\boldsymbol{\epsilon}, \tag{4}$$

$$\boldsymbol{D} = \kappa \begin{bmatrix} 1-v & v & 0 \\ v & 1-v & 0 \\ 0 & 0 & \frac{1-2v}{2} \end{bmatrix},$$

$$\kappa = \frac{E}{(1+v)(1-2v)}, \tag{5}$$

where $E$ and $v$ denote Young's modulus and Poisson's ratio, respectively.

**Load-displacement relationship.** Now we derive relationship between the external forces and the displacement by use of the principle of virtual work, that is, when a structure is in equilibrium, the internal work $\delta W_i$ consorts with the outer work $\delta W_o$ generated by applying arbitrary infinitesimal displacement to the structure. $\delta W_o$ is represented by

$$\delta W_o = \delta \boldsymbol{u}^T \boldsymbol{f}, \tag{6}$$

where $\delta \boldsymbol{u}$ denotes virtual displacement, which is infinitesimal so as not to affect the external forces. The internal work $\delta W_i$ is given by integrating the product of virtual strain and stress over the finite element.

$$\delta W_i = \int_A \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dA, \tag{7}$$

where $\delta\boldsymbol{\epsilon}$ is virtual strain given by $\delta\boldsymbol{\epsilon} = \boldsymbol{B}\delta\boldsymbol{u}$. $A$ denotes the area of the finite element.

By the principle of virtual work $\delta W_o = \delta W_i$, we obtain

$$\delta \boldsymbol{u}^T \boldsymbol{f}_e = \int_A \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dA.$$

By substituting Eq. 3 and 4 to this equation, we obtain

$$\begin{aligned} \delta \boldsymbol{u}^T \boldsymbol{f}_e &= \delta \boldsymbol{u}^T \int_A (\boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \boldsymbol{u}_e) dA \\ &= \delta \boldsymbol{u}^T \int_A (\boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B}) dA \boldsymbol{u}_e. \end{aligned}$$

Since this equation holds for arbitrary virtual displacement and strain, the following equation of equilibrium is derived. It defines the relationship between the external forces and displacement at the nodes.

$$\begin{aligned} \boldsymbol{f}_e &= \int_A (\boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B}) dA \boldsymbol{u}_e \\ &= \boldsymbol{K}_e \boldsymbol{u}_e. \end{aligned} \tag{8}$$

This matrix $\boldsymbol{K}_e$ is equal to $\boldsymbol{K}_e$ in Eq. 1.

**System equation.** By assembling the Eq. 8 for all the elements, we obtain a linear system equation to solve a static structural problem to find displacement of nodes for given external forces. We define node numbers 1,2,3,...,$N_v$ for the set of all the nodes $\{v_i\}$, where $N_v$ denotes the total number of nodes. We also define the node displacement vector $\boldsymbol{u}_i$ and external force vector $\boldsymbol{f}_i$ for the node $v_i$. Then the total node displacement vector $\boldsymbol{u}$ is defined by arranging $\boldsymbol{u}_i$ in this form:

$$\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{N_v})^T.$$

The total external force vector $\boldsymbol{f}$ is also defined in the same way:

$$\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_{N_v})^T.$$

By using those vectors the system equation for the static structural problem can be represented by

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{9}$$

where $\boldsymbol{K}$ is called the *system stiffness matrix*.

**von Mises strain.** From the node displacement $\boldsymbol{u}_e$, we can estimate strain vector (deformation ratio) $\boldsymbol{\epsilon}$ using Eq. 3. As the measure for the total amount of the strain, von Mises strain $\epsilon_m$ is used as a scalar expression of strain, which can be defined as

$$\epsilon_m = \sqrt{\frac{2}{9}\left((\epsilon_x - \epsilon_y)^2 + \epsilon_x^2 + \epsilon_y^2\right) + \frac{1}{3}\left(\gamma_{xy}^2\right)}. \tag{10}$$

We rewrite this equation using

$$\boldsymbol{P} \equiv \frac{1}{3}\begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ \sqrt{2} & -\sqrt{2} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix}. \tag{11}$$

Then we obtain

$$\begin{aligned} \epsilon_m &= \sqrt{(\boldsymbol{P}\boldsymbol{\epsilon})^T(\boldsymbol{P}\boldsymbol{\epsilon})} \\ &= \sqrt{(\boldsymbol{P}\boldsymbol{B}\boldsymbol{u}_e)^T(\boldsymbol{P}\boldsymbol{B}\boldsymbol{u}_e)}. \end{aligned} \tag{12}$$

Note that $\epsilon_m$ is defined for each element.

### Pixel based FEM

We apply FEM to CT image (a two-dimensional image). This approach was first proposed by Bendsoe and Kikuchi [23] as "Pixel based FEM." The pixels of the image $D$ are squares aligned with an *xy* coordinate system in a reticular pattern. In this paper, we express pixels as $p$ and its value as $V(p)$. We consider a set of foreground pixels $D_f(\subset D)$ represents a structure. In this paper, we use a simple thresholding method to extract the foreground with a conservative threshold value so as not to miss vague foreground pixels. Although more sophisticated methods can be used here, we regard this problem out of scope of this paper as the choice of the method does not affect our method.

For each pixel of $D_f$ we define a quadrilateral element of FEM. The four corners of $p$ will be nodes of the element. From this point, pixels and elements are regarded as the same.

## Methods

Here we propose a CT image region segmentation method using FEM in which boundary conditions are optimized to achieve superior image segmentation. We first discuss two-fraction partitioning problem. Fig. 3 illustrates the proposed approach to image segmentation. We assume that the target object has an *interference region* between *regions*. Since we usually use the word "interface" as the boundary between foreground and background. In order to avoid confusion, we named the area between regions as interference region.

The method is based on the intuitive observation that a structure can be broken along this interference region by applying appropriate external forces because it is structurally weaker than other regions and carries a higher level of strain. We thus find pixels with a higher level of strain and remove them to break the object. Strain can be computed using the FEM.

Fig. 1 gives a flow chart of the proposed segmentation method including three main steps. In the figure, a CT image of mouse's shoulder is shown as an example. We will discuss these steps in the following sections.

### Initialization

For applying FEM, we need to define material properties of Young's modulus $E$ and Poisson's ratio $v$. In the case of bone, $E$ is related to the cube of its density [24]. Accordingly we assume that the Young's modulus $E(p)$ of pixel $p$ is proportional to $V(p)^3$, in this paper. Poisson's ratio $v$ is set constantly as 0.3.

Eq. 9 is a system of simultaneous linear equations. However, we cannot solve this equation, since **K** is a singular matrix. In general cases, to solve Eq. 9, a displacement constraint must be set to the nodes. This operation is equivalent to fix the nodes to the ground. However, selection of fixing nodes is not a trivial problem. Instead, we set weak springs to all nodes, and these springs are connected to the ground. Setting the spring coefficient as $\delta$, the stiffness matrix is modified as:

$$\boldsymbol{K} := \boldsymbol{K} + \delta \boldsymbol{I}, \tag{13}$$

where $\boldsymbol{I}$ denotes the identify matrix. $\delta$ is not sensitive to computation, so we determine $\delta$ experimentally, which is sufficiently smaller than the largest element of $\boldsymbol{K}$.

We assume the user roughly knows the location of the interference region. The user must define a region of interest (ROI) $R(\subset D_f)$ such that it includes the interference region.

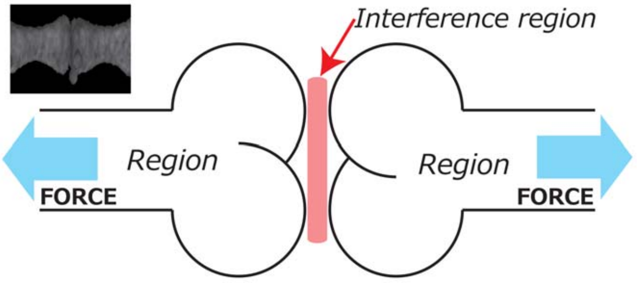

**Figure 3. Schematic overview of the proposed segmentation method.** On the upper left is a CT image of a joint of a mouse's tail which consists of bones and cartilage. We assume the bone as a region and the cartilage as an interference region and separate the regions in the interference region. By applying forces to the bones so as to generate high strain to destroy the interference region.
doi:10.1371/journal.pone.0031116.g003

Fig. 4(a) shows an image $D_f$, which is a 2D foreground CT image of a mouse's backbone. We want to segment along the interference region, which is highlighted as red in Fig. 4(b).

**Loading condition setting.** In order to find pixels to be removed with a higher level of strain, we evaluate von Mises strain $\epsilon_m(p)$ of each pixel $p$ with the displacement of nodes $\boldsymbol{u}$, which is obtained by solving Eq. 9. For this purpose, we need to set the loading condition of external forces $\boldsymbol{f}$. One method was to do it manually [1], but this is not a realistic way particularly for complex structures. In this paper we propose a method to automatically find some appropriate $\boldsymbol{f}$ by casting this problem to an optimization problem.

A desirable loading condition creates high strain in the ROI. As von Mises strain is a non-negative scalar value, we set the objective function as

$$F(\boldsymbol{f}) = \sum_{p \in R} \epsilon_m(p)^2 \rightarrow max, \tag{14}$$

where $\epsilon_m(p)$ denotes the von Mises strain of element $p$.

**Optimization.** Gradient ascent method is used to calculate loading condition vector $\boldsymbol{f}$.

$$\boldsymbol{f} = \boldsymbol{f} + \lambda \nabla F(\boldsymbol{f})$$

$$\nabla F(\boldsymbol{f}) = 2\boldsymbol{K}^{-1} \sum_{p \in R} \boldsymbol{w}(p)$$

$$\boldsymbol{w}(p) = \begin{array}{cl} \left(\begin{array}{c} \vdots \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_3 \\ \vdots \\ \boldsymbol{w}_4 \\ \vdots \end{array}\right) & \begin{array}{l} \\ (i1 \\ \\ (i2 \\ \\ (i3 \\ \\ (i4 \\ \\ \end{array} \end{array}$$

$$(\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4)^T = (\boldsymbol{PB})^T (\boldsymbol{PB}) \boldsymbol{u}(p), \tag{15}$$

where $(i1, i2, i3, i4)$ denote the four node numbers of element $p$ and $\boldsymbol{u}(p)$ denotes the displacement vector of $p$ which is equal to $\boldsymbol{u}_e$ defined by Eq. 1. The other elements of $\boldsymbol{w}(p)$ are all zero. This $\nabla F(\boldsymbol{f})$ can be derived using Eqs. 9, 11, and 13. $\lambda$ is the user input coefficient. Since $\lambda$ is sensitive to convergence speed, we use the Armijo rule [3] to set $\lambda$.

We limit $\boldsymbol{f}$ to the surface of $\|\boldsymbol{f}\| = 1$. W need to modify $\nabla F(\boldsymbol{f})$ to be $\nabla_M F(\boldsymbol{f})$ as shown in Fig. 5. Instead of Eq. 14, we use the following equations:

$$\begin{aligned} \boldsymbol{f} :&= \boldsymbol{f} + \lambda \nabla_M F(\boldsymbol{f}), \\ \nabla_M F(\boldsymbol{f}) &= \nabla F(\boldsymbol{f}) - ((\boldsymbol{v} \cdot \nabla F(\boldsymbol{f})) \cdot \boldsymbol{v}) \\ &= \nabla F(\boldsymbol{f}) - ((\boldsymbol{f} \cdot \nabla F(\boldsymbol{f})) \cdot \boldsymbol{f}), \end{aligned} \tag{16}$$

where $\boldsymbol{v}$ denotes the normal vector of $F(\boldsymbol{f})$ and is equal to $\boldsymbol{f}$.

We set external force only to the boundary nodes of $D_f$. We set the initial loading condition vector $\boldsymbol{f}^0$ to obtain boundary nodes away from $R$ (Fig. 3) because the level of strain will be large when nodes around pixels are moved in different orientations. By calculating the center point $(g_x, g_y)$ of $R$ assuming the CT value as the weight, we set $\boldsymbol{f}^0$ for the $j$-th node $(x_j, y_j)$ as $(x_j - g_x, y_j - g_y)$. By aligning all the initial loading conditions of all boundary nodes and normalizing them, we make $\boldsymbol{f}^0$.

Fig. 6 shows the optimized loading condition vector for the image in Fig. 4(a) with ROI of Fig. 4(b). In Fig. 6, the black points are boundary nodes, and the red arrows are the loading condition vectors of the black points.

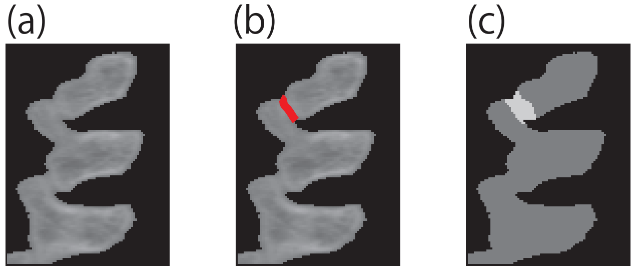

**Figure 4. ROI setting.** (a) is Input image $D_f$, (b) is Interference region, and (c) is ROI $R$. (a) is the input foreground image. The red line in (b) indicates an interference region. The white area in (c) is user specified ROI $R(\subset D_f)$.
doi:10.1371/journal.pone.0031116.g004

## Analogy of destruction

We imitate destruction by iterating the two steps of strain calculation and pixel removal. As previously outlined, $\boldsymbol{K}$ is taken as the foreground pixel set $D_f$. With $\boldsymbol{f}$, we calculate $\boldsymbol{u}$ by solving Eq. 1, and calculate $\epsilon_m(p)$ using Eq. 11.

The pixel $\hat{p} \in R$ with the highest von Mises strain value is removed from $D_f$ by setting it as a background pixel in this form:

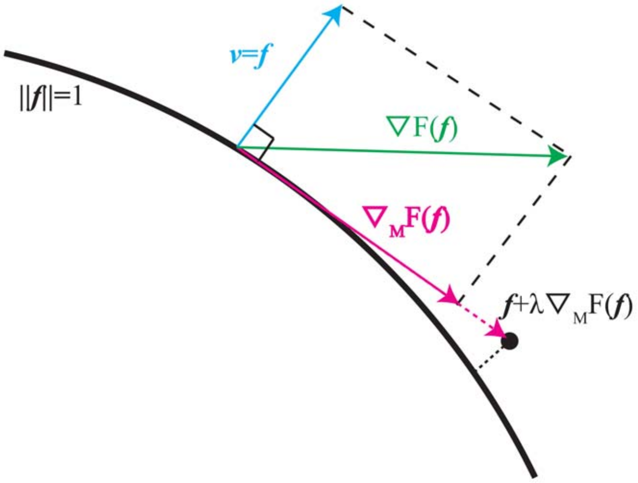

**Figure 5. Loading condition on manifold surface.** $\mathbf{v}$ is a normal vector of $\|\mathbf{f}\| = 1$ and is equal to $\mathbf{f}$.
doi:10.1371/journal.pone.0031116.g005

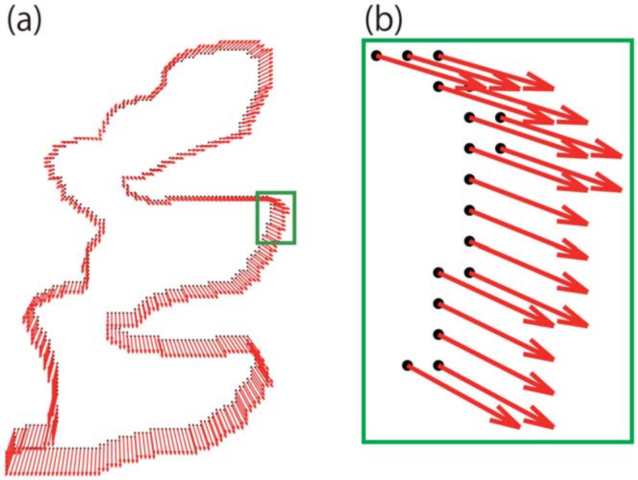

**Figure 6. Results of loading condition optimization.** (a) is Loading condition and (b) is Close up view of (a). The black points are boundary nodes, and the red arrows are loading condition vectors for each node. (b) is a close-up of (a) from the green area.
doi:10.1371/journal.pone.0031116.g006

$$D_f := D_f \backslash \{\hat{p}\},$$

where\denotes set subtraction operator.

Then, we check if the topology (connectivity) of the foreground $D_f$ changes or not. When $D_f$ decomposes into multi sets, the iteration halts and the process progresses to the next section. Otherwise, we repeat the above strain calculation.

### Repairing

We label the remaining pixel sets in $D_f$ using neighbor relationships. And we apply the region growing method until all removed voxels are classified into one of the segmented regions.

### Expansion to multi-segmentation

Here, we expand the proposed segmentation method to form a multi-segmentation problem. Assuming that structurally weak areas should be segmented, a decomposition priority is calculated using the ROI strain.

In line with the above discussion on two-fraction partitioning, one ROI is set. Here, we consider a case with $N_d$ interference regions. The user sets $N_d$ ROIs. Fig. 7(a) is the same sample as Fig. 4(a), however two interference regions (red lines) are considered and two ROIs (white regions) are specified in Fig. 7(b).

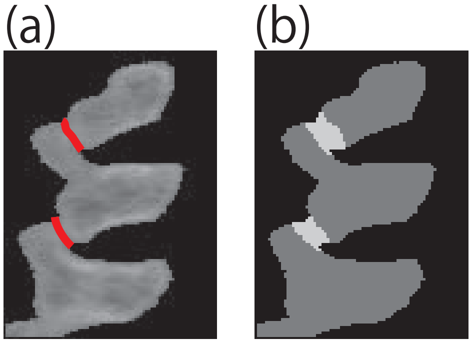

**Figure 7. Expanding to multi segmentation.** (a) is Interference regions and (b) is ROIs. In (a), red lines are the interference regions. In (b), two ROIs are shown in white.
doi:10.1371/journal.pone.0031116.g007

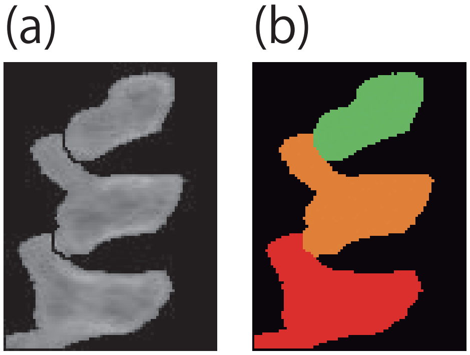

**Figure 8. Result of Fig. 4.** (a) is the result of destruction. (b) is the repaired version of (a).
doi:10.1371/journal.pone.0031116.g008

We segment ROI one by one. In order to determine the iteration order, we calculate values of Eq. 13 for all ROIs and divide them by their area to evaluate their structural weakness. We iterate region segmentation for each of ROIs with the order of their structural weakness. The total number of iterations is $N_d$. The result for the case of Fig. 7 is shown in Fig. 8.

### Doubtful ROI specification

The ROIs must be given by a user, however in some cases he/she makes a mistake to specify ROIs to regions that only look like interference but are not real ones. It is worth mentioning that we can call off the segmentation for those incorrect ROIs by using the structural weakness values as they are very low for such incorrect ROIs. It is demonstrated using the sample of Fig. 9 in the next section.

## Results

We applied the proposed algorithm to a number of CT images. Figs. 9(a) 10(a), and 11(a) are CT images of parts of an orange, a mouse's backbone, and a mouse's front teeth, respectively. The ROIs are highlighted in white in Figs. 9(b), 10(b) and 11(b).

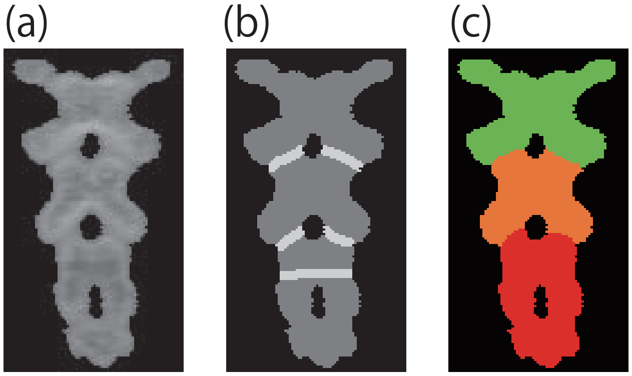

**Figure 9. Backbone of a mouse (a) is the input CT image.** The white areas in (b) are the ROIs. (c) is the result of the proposal method. Note that segmentation is not made along the lowest ROI in (b). Such ROIs are sometimes mistakenly defined for the region without interference. Our method can skip such incorrect ROIs.
doi:10.1371/journal.pone.0031116.g009

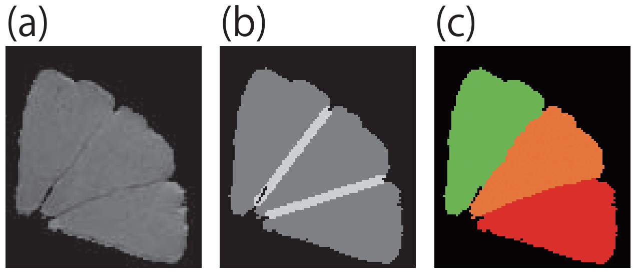

**Figure 10. Sections of orange (a) is the input CT image.** The white areas in (b) are the ROIs. (c) is the result of the proposed method.
doi:10.1371/journal.pone.0031116.g010

Table 1 shows the conditions and parameters of the experiments. We implemented our algorithm using C++ and performed experiments using a Windows 7 64-bit computer with an Intel(R) Core i7 CPU 920 @ 2.67 GHz and 12 G RAM.

Figs. 9(c), 10(c) and 11(c) show the segmentation results of Figs. 9(a), 10(a) and 11(a), respectively. Table 2 shows the computation time necessary to obtain the experimental results.

## Discussion

### Comparison with graph cut-based method

The graph cut method was originally proposed by [14] for image segmentation. A user defines hard constraints for segmentation by specifying two sets of pixels (seeds) that must be a part of the two segments. The other pixels are segmented into either one of these segments automatically by solving a global optimization problem. By regarding the image as a graph of pixels, the pixels are connected by arcs to their neighboring ones and also to two terminal nodes representing the two segments. Costs are defined for these arcs according to the similarities between their incident pixels. Then the max flow between the terminal nodes is computed so that the graph is cut into two parts along the max cut. These two graphs constitute the segmentation.

In our experiments with the graph cut we use the same cost functions for all the images. The cost function between neighboring pixels is defined by using an exponential value of the difference between the CT values of these pixels. We conducted trials of segmentation to tune parameters involved in the cost function to give good results. As for the seeds of the graph cut, we also tested their several patterns and Figures 12(a), (b), and (c) are the ones resulting in the best segmentations.

In the case shown in Fig. 10(a), interference regions are thin and clear. It is easy to segment this type as shown in Fig. 12(a). Although this is a CT image of an orange rather than a skeleton, Fig. 10(c) also shows successful segmentation. From these results, our algorithm works well in the case of clear CT image, which could be segmented by using conventional method.

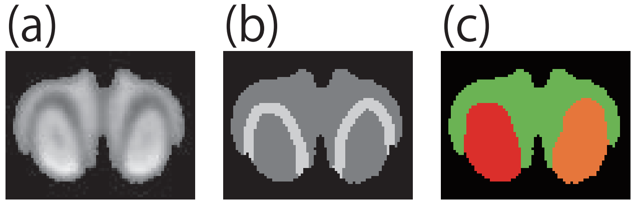

**Figure 11. Front teeth of mouse.** (a) is the input CT image. The white areas in (b) are the ROIs. (c) is the result of the proposed method.
doi:10.1371/journal.pone.0031116.g011

**Table 1.** Conditions of experiment $N_u$ is the number of ROIs and $N_d$ is the number of iteration.

| | Size | $\lambda$ | $N_d$ | $N_u$ |
|---|---|---|---|---|
| Fig. 1 | $105\times 81$ | $5.0E-10$ | 1 | 1 |
| Fig. 4 | $80\times 108$ | $5.0E-10$ | 2 | 3 |
| Fig. 10 | $91\times 103$ | $5.0E-10$ | 2 | 2 |
| Fig. 9 | $80\times 144$ | $5.0E-10$ | 4 | 5 |
| Fig. 11 | $73\times 57$ | $5.0E-10$ | 2 | 2 |

doi:10.1371/journal.pone.0031116.t001

Here we also discuss the cases of Fig. 9(a) with multiple interference regions. As these interference regions are not ambiguous, the graph cut method produces successful results as shown in Fig. 12(b). The proposed algorithm also works well. Additionally, although Fig. 9(b) contains erroneous ROI, the proposed algorithm avoids segmentation in this region. From these results, our algorithm can avoid wrong segmentation when the user knows the number of the regions.

In the case of Fig. 11(a), whose interference region is clear but comparatively large. Because of its size, the energy of the region is the same as that of the interference region, and Fig. 12(c) shows the drawback of using the graph cut method in this case. From these results, our algorithm also works in the case of CT image with ambiguous interference region or wide interference region.

### Influence of ROI

We tested a number of ROI patterns as shown in Fig. 13, which is an image of a human knee (The Volume Library [25])($41\times 56$ pixels). Fig. 13(a) shows the input image with a standard ROI (highlighted in red), and Fig. 13(e) is the segmentation result for Fig. 13(a). Figs. 13(b), (c), and (d) are also input images with a large ROI, a very large ROI and an irregularly shaped ROI, respectively. Figs. 13(f), (g), and (h) are the results of segmentation for Figs. 13(b), (c), and (d), respectively.

Figs. 13(b) and (d) are similar to Fig. 13(a). This indicates that the area and shape of the ROIs are not dominant with the proposed algorithm. However, failed segmentation is shown in Fig. 13(g), whose ROI is almost the same size as the entire foreground of the input image. These experiments indicate that it is necessary to set an appropriate ROI because images generally contain more than one structurally weak areas. In other words, if the user sets ROIs based on a certain level of prior knowledge, successful segmentation can be guaranteed.

**Table 2.** Computational times for step 1 and step 2.

| | Step 1 | Step 2 | Total |
|---|---|---|---|
| Fig. 1 | 0.10 | 2.53 | 2.63 |
| Fig. 4 | 0.14 | 0.62 | 1.20 |
| Fig. 10 | 0.22 | 5.24 | 9.07 |
| Fig. 9 | 0.32 | 1.36 | 6.32 |
| Fig. 11 | 0.05 | 1.00 | 1.76 |

The values are computational times (min) for each step. Step 1 is for loading condition optimization, and step 2 is for destruction and repairing.
doi:10.1371/journal.pone.0031116.t002

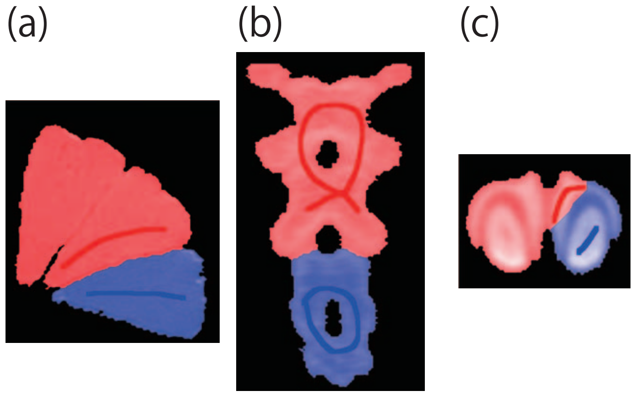

**Figure 12. Results of the graph cut method.** (a), (b) and (c) are the results of the graph cut method for the images of Fig. 10, Fig. 9, and Fig. 11, respectively. (a) and (b) are considered fine, but (c) is not an appropriate result.
doi:10.1371/journal.pone.0031116.g012

## Computational time

Table 2 shows computational times required for segmentation. The proposed algorithm is comparatively slow because FEM was used for structural analysis and iterated. The computational time taken for FEM depends on the size of $\boldsymbol{K}$. The total iteration time is similar to that of area $M$ in the interference region. Thus, the proposed algorithm's total time complexity can be estimated as

$$T \approx O(N_v^2 \times M), \tag{17}$$

where $N_v$ is the number of foreground vertices (nodes).

## Shape limitation

Since our algorithm is designed to segment joints of a skeleton, it prefers such objects with relatively high aspect ratio around the joints as shown in Figs. 9(c), 10(c) and 11(c). However, it is not suitable for objects with isotropic shapes which tend to have uniform distribution of strain.

## Conclusion

We proposed a CT image segmentation method using structural analysis aiming at segmenting a CT image of a skeleton into pieces of the bones at their joints where the contrast of the image is usually weak. In our novel approach, we compute strain distribution of a structure in the image to find those pixels with high level of the strain then segment the image along these pixels. To compute the strain distribution, we use Pixel based FEM method. In order to automatize the procedure, we introduce an optimization method to derive loading condition (external forces) so as to bring high strain to the pixels in user defined ROI. From experimental results, it can be concluded that the proposed algorithm works well with CT images with large or ambiguous interference regions. Our future work includes extension to three dimensional problems and improvement of computational efficiency.

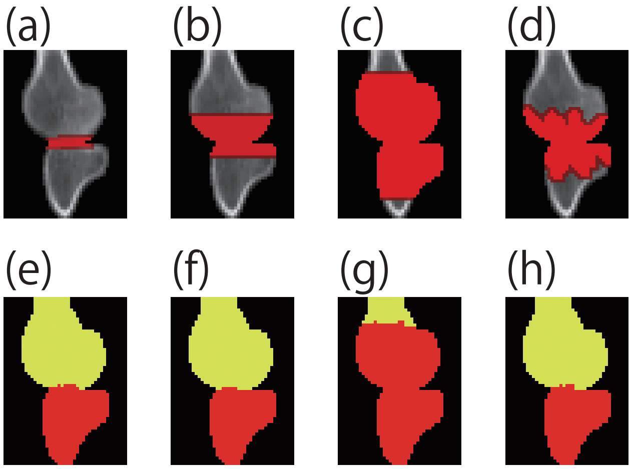

**Figure 13. Results of ROI test.** Images in the top row are input CT images with ROIs (red regions). Images in the bottom row are segmented results for corresponding images in the top row.
doi:10.1371/journal.pone.0031116.g013

## Acknowledgments

The authors thank Dr. Keizo Ishii, Mr. Makoto Tsukino, and Mr. Hiroyuki Eiyama at Quint Corp for their technical support to use FEM software.

## Author Contributions

Conceived and designed the experiments: HH HS TM YO SO. Performed the experiments: HH HS TM YO SO. Analyzed the data: HH HS TM YO SO. Contributed reagents/materials/analysis tools: HH HS TM YO SO. Wrote the paper: HH HS TM YO SO.

## References

1. Hishida H, Michikawa T, Ohtake Y, Suzuki H, Oota S (2010) CT image segmentation using structural analysis. Advances in Visual Computing. pp 39–48.
2. Quint Corporation. Voxelcon. Available: http://www.quint.co.jp/jp/pro/vox/index.htm. Accessed 2012 Jan 8.
3. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C (2nd ed.): the art of scientific computing, ISBN:0-521-43108-5. New York: Cambridge University Press.
4. Zhang J, Yan C, Chui C, Ong S (2010) Fast segmentation of bone in CT images using 3D adaptive thresholding. Computers in Biology and Medicine 40: 231–236.
5. Kass M, Witkin A, Terzopoulos D (1988) Snakes: Active contour models. International journal of computer vision 1: 321–331.
6. Pardo X, Carreira M, Mosquera A, Cabello D (2001) A snake for CT image segmentation integrating region and edge information. Image and Vision Computing 19: 461–475.
7. Sebastian T, Tek H, Crisco J, Kimia B (2003) Segmentation of carpal bones from CT images using skeletally coupled deformable models. Medical Image Analysis 7: 21–45.
8. Adams R, Bischof L (1994) Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence 16: 641–647.
9. Joliot M, Mazoyer B (1993) Three-dimensional segmentation and interpolation of magnetic resonance brain images. IEEE Transactions on Medical Imaging 12: 269–277.
10. Mat-Isa N, Mashor M, Othman N (2005) Seeded region growing features extraction algorithm; its potential use in improving screening for cervical cancer. International Journal of the Computer, the Internet and Management (ISSN No: 0858-7027) 13.
11. Vincent L, Soille P (1991) Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE transactions on pattern analysis and machine intelligence 13(6): 583–598.
12. Grau V, Mewes A, Alcaniz M, Kikinis R, Warfield S (2004) Improved watershed transform for medical image segmentation using prior information. IEEE Transactions on Medical Imaging 23: 447–458.
13. Hahn H, Wenzel M, Konrad-Verse O, Peitgen H (2006) A minimally-interactive watershed algorithm designed for efficient CTA bone removal. Computer Vision Approaches to Medical Image Analysis. pp 178–189.
14. Boykov Y, Funka-Lea G (2006) Graph cuts and efficient N-D image segmentation. International Journal of Computer Vision 70: 109–131.
15. Elias P, Feinstein A, Shannon C (1956) A note on the maximum flow through a network. Information Theory, IRE Transactions on 2: 117–119.

16. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22: 888–905.
17. Campadelli P, Casiraghi E, Esposito A (2009) Liver segmentation from computed tomography scans: A survey and a new algorithm. Artificial Intelligence in Medicine 45: 185–196.
18. Müller R, Rüegsegger P (1995) Three-dimensional finite element modelling of non-invasively assessed trabecular bone structures. Medical engineering & physics 17: 126–133.
19. Azar F, Metaxas D, Schnall M (2002) Methods for modeling and predicting mechanical deformations of the breast under external perturbations. Medical Image Analysis 6: 1–27.
20. Auer M, Gasser T (2010) Reconstruction and finite element mesh generation of abdominal aortic aneurysms from computerized tomography angiography data with minimal user interactions. IEEE Transactions on Medical Imaging 29: 1022–1028.
21. Schnörr C (1998) A study of a convex variational diffusion approach for image segmentation and feature extraction. Journal of Mathematical Imaging and Vision 8: 271–292.
22. RAO SS (2005) The finite element method in engineering (4th ed.), ISBN: 978-0-7506-7828-5. Oxford: Elsevier Butterworth Heinemann.
23. Bendsoe M, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Computer methods in applied mechanics and engineering 71: 197–224.
24. Carter D, Hayes W (1977) The compressive behavior of bone as a two-phase porous structure. The Journal of bone and joint surgery 59: 954–962.
25. Roettger S. Volume library. Available: http://www9.informatik.uni-erlangen.de/External/vollib/. Accessed 2012 Jan. 8.