

The Integration of Automatic Segmentation and Motion Tracking for 4D Reconstruction and Visualization of Musculoskeletal Structures

José G. Tamez-Peña¹ Kevin J. Parker²
Department of Electrical Engineering,
University of Rochester,

Rochester, NY, RC-BOX 270126

¹pena@ee.rochester.edu, ²parker@ee.rochester.edu

Saara Totterman

Department of Radiology,
University of Rochester,

Rochester, NY, 14620

tot@rad.rochester.edu

Abstract

The 3D reconstruction of high contrast anatomical structures has been widely explored. However, a class of important clinical problems involves the motion of very complex musculoskeletal structures including the joints, hence a 4D reconstruction is desired. Practical difficulties with 4D reconstruction with MRI include the time required for data acquisition, the resolution required for visualization of small but critical structures, the gross inhomogeneities of field coil response, the degree of noise present with the signal and the extreme low-contrast details between some distinct anatomical structures. We present a comprehensive approach to 4D musculoskeletal imagery that address the above challenges. Specific MRI imaging protocols; segmentation, motion estimation and motion tracking algorithms are developed and applied to render complex 4D musculoskeletal systems. Applications of the approach include the analysis of the rotation of the upper arm and the knee extension.

1. Introduction

With current imaging technology it is difficult to evaluate the motion patterns of the articulating bones and the changes in the relationships between soft tissue structures of the joints during motion. Improvements in these areas would greatly increase our understanding of the normal patient and the development and staging of osteoarthritis and a wide of variety of clinical conditions. Whole-volume CT and MRI imaging which provides 3D data, would in theory, give us the ability to evaluate and also visualize static phenomena. Although there are some volume and surface rendering techniques that can be used to visualize the raw data of simple anatomical structures[17], these techniques have limited success with volumetric images of complex

anatomical structures like the joints. Image segmentation is a standard technique used to differentiate anatomical structures from the image and then use the information to get a better visualization of the volumetric data. Furthermore, segmentation can, in theory, provide a complete description of the anatomical surfaces making them suitable input for virtual reality environments. However, so far there are no unsupervised algorithms available to reliably segment soft tissue structures of a moving musculoskeletal system for their 3D and 4D visual presentation, or virtual manipulation. The purpose of this study was to develop a time adaptive 3D segmentation algorithm and 4D display of the motion of soft tissue structures and bones of joints using 3D MRI data and apply that method for the analysis of a series of MR volumes which were acquired during joint motion.

Our analysis of musculoskeletal structures is based, but not limited, on MR volumetric images. The anatomical structures of the analyzed joints are very complex. Furthermore, the small structures like tendons and cartilage need to be segmented for a full appreciation of the motion. To get the desired resolution local coils were used for image acquisition, thus field inhomogeneities were present. Besides these field inhomogeneities the tendons' properties are very complex, making them very difficult to segment. Like others volumetric image modalities, MR images suffers from low contrast between some anatomical structures; limited iter-slice resolution; noise; and spatially varying sensitivity. The segmentation of low contrast structures represents, perhaps, the greatest challenge to many segmentation algorithms. This problem has promoted the creation and use of active contour models[6] (snakes) and their 3D extension (balloons) [1]. However, these techniques have to be initialized properly and they can only segment one anatomical structure at time thus limiting their applicability. Region growing algorithms are best suited for the segmentation of slowly varying non-uniformities. They are based on the local pixel properties to segment an image[5]. Among

the more common are the edge-based pixel approaches that look for close contours to distinguish different regions. Another interesting region-based technique is based on watershed analysis which has been used lately in the segmentation of medical images [14, 4]. The advantage of searching the image locally is that they are more robust to spatially variations like contrast agents induced intensity variations or field inhomogeneities. On the other hand, they tend to over-segment the image if they are not properly adjusted.

Although there are many 2D region growing algorithms, just a few of them have been successfully applied to the segmentation of 3D structures [18, 16]. On the other hand, we followed recent advances on line-preserving and edge-preserving image restoration techniques[3, 10], to develop a novel 3D local voxel statistical parameter estimator. This voxel estimators allow the use of simple 3D region growing algorithms for the estimation of the iter-voxel connectivity, which in combination with standard region merging and relaxation labeling algorithms allows good segmentations of standard MR images.

On the other hand, the time analysis of the musculoskeletal system requires the tracking of several rigid and non-rigid structures. Although there are several successful approaches for deformable and rigid motion tracking that can be used for the tracking of single structures[8, 7], using them for tracking all the structures of the musculoskeletal system problem is not practical. Therefore, we developed a novel tracking approach based on a simple physical based model of the whole volume image. The model takes into account our *a-priori* knowledge of the human anatomy which is coded on the segmentation of the first volume of the set, as a simple finite element mesh, which deforms according to the image differences between volumes in the sequence. Our whole image approach allows a good and relatively fast motion flow estimation that is used to propagate the segmentation of the first volume to the next in the series.

In this paper, we propose an hierarchical application of our automatic region based segmentation algorithm, motion estimation techniques to the analysis of the dynamics of musculoskeletal structures. First, we perform an automatic segmentation of the first MRI volume in a 4D sequence. The segmented image and our *a-priori* knowledge of the tissue elasticity are input to our motion tracking algorithm which use this information to estimate the motion of the different anatomical components present in the image. Once the motion has been estimated we proceed to pass the segmented image from the first volume to the next. This process is repeated until we have a complete segmentation set of the whole sequence. After that, the segmented data, the raw data and some user parameters, like tissue color, are used to construct realistic surfaces of all the anatomical components. These realistic surfaces include some texture information that enhance the visualization of the different

bone-tendons attachment points.

1.1. Image model

A temporal volumetric digital image $g(x, y, z, k)$ is modeled as a sampled version of a continuous 3D temporal image $g_c(x_c, y_c, z_c, t)$, where a discrete point $\mathbf{x} = (x, y, z)$ in the volumetric digital image is called voxel. This image can be assumed to be created after illuminating a 3D complex dynamical object $f(\mathbf{x}_c, t)$ with some sort of time invariant illumination field $I(\mathbf{x})$. f is assumed to be composed of M distinct regions R_1, R_2, \dots, R_M , $R_i \cap R_j = \emptyset$, for $i \neq j$. If we assume that the image $g(\mathbf{x}_c, t)$ is corrupted by additive independent Gaussian noise η , then the observed discrete value at the voxel $\mathbf{x} \in g$ at time $t = kT$ is given by:

$$g(\mathbf{x}, t) = I(x \Delta x, y \Delta y, z \Delta z) f(x \Delta x, y \Delta y, z \Delta z, kT) + \eta(x \Delta x, y \Delta y, z \Delta z, kT), \quad (1)$$

where $(\Delta x, \Delta y, \Delta z)$ are the sampling spacing intervals. Our specific goal is to analyze the motion of the M regions that compromise f . To address this problem, we have to be able to segment and track those regions from the observed image g .

2. 3D segmentation

The 4D analysis of the temporal volume data sets is done by a hierarchical implementation of volume segmentation and motion estimation and tracking. In these section we present how we addressed the 3D segmentation problem

2.1. Region growing

Our 3D segmentation is based in a hybrid linkage region growing scheme, where every voxel in the volume is regarded as a node in a graph. Neighboring voxels whose properties are similar enough are joined by an arc. The image segments R_i are maximal sets of voxels all belonging to the same connected components [5]. If we assume that our image is composed of stochastic regions, then we can assign to every voxel (\mathbf{x}) in the image two local statistical properties: the local region mean value $\mu(\mathbf{x})$ and local region variance $\sigma^2(\mathbf{x})$. Thus according to this scheme two voxels are connected by an arc if their mean value and variance are similar enough. If these two properties were known, then one can test the voxel similarity according their Gaussian cumulative distribution function. On the other hand, if we can only estimate them with certain confidence, then we can use comparative statistical tests. Before continuing, let us state the definitions of a neighborhood system and a clique in a three dimensional space. Let $\mathcal{G}_i(\mathbf{x}) = \mathcal{G}(\mathbf{x}) \forall \mathbf{x} \in R_i$ be a neighborhood system on

R_i , meaning any collection of subsets of R_i for which 1) $\mathbf{x} \notin \mathcal{G}_i(\mathbf{x})$ and 2) $\mathbf{x}_k \in \mathcal{G}_i(\mathbf{x}_j), \Leftrightarrow \mathbf{x}_j \in \mathcal{G}_i(\mathbf{x}_k)$. Thus $\mathcal{G}_i(\mathbf{x})$ is a set of neighbors of \mathbf{x} . A subset $C_i(\mathbf{x}) \subseteq R_i$ is a clique if every pair of sites in $C_i(\mathbf{x})$ are neighbors, and $\mathcal{C}_i(\mathbf{x})$ is the set of cliques [3]. Notice that voxels at or near the boundary of a region R_i have fewer neighbors than do interior voxels, this is very important in the formulation of the segmentation algorithm. Figure 1a) shows the 3D neighborhood system adopted in our research.

2.2. Parameter estimation

Now, our problem is to derive estimations of the local region mean and local variance in the presence of noise. Theoretically, the mean and variance can be estimated using standard image restoration techniques; but these don't properly address the regions' boundaries. If the estimated parameters are function of the local connected voxel values, then they can be modeled by a Markov random field, where a Gibbs Random Field (GRF) is special case. Assume a Gibbs Random Field model for the the joint distribution of the estimated mean value image, μ , and the input image, g , therefore the probability function for the input output relationship is in the form[10]:

$$P(\mu, g) = k \exp[-U(\mu, g)] \quad (2)$$

where k is just a normalization constant.(GRF details can be found in [3, 10]). The exponent $U(\mu, g)$, is called the energy of the input-output image pair under the specified model. It is known that the energy function associated with the input-output pair can be decomposed and written as the sum of clique potentials, as:

$$U(\mu, g) = \sum_{\forall C \in \mathcal{G}} V_C \quad (3)$$

where C is a clique in the neighborhood, and the summation is over the set of all cliques in the neighborhood \mathcal{G} . For simplicity, many of the possible clique potentials are set to zero. Therefore, the energy U is a sum of a small set of cliques with nonzero potentials.

A general estimator based on the GRF approach is an operator which maximizes the probability function with respect to μ given a fixed input g

$$\mu_o = \max_{\mu} P(\mu|g) = \max_{\mu} \frac{P(\mu, g)}{P(g)} \quad (4)$$

But for a given input image g , $P(g)$ is constant, therefore, this maximization problem reduces to the minimization of the energy function:

$$\mu_o = \min_{\mu} U(\mu, g). \quad (5)$$

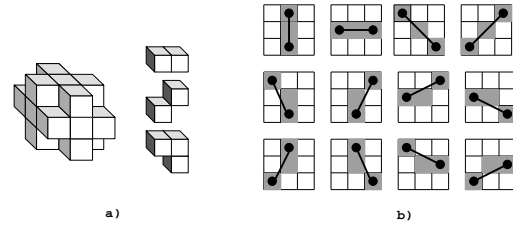


Figure 1. a) 3D 18 neighborhood system b) 2D lines segments in a 2D 3x3 region

Let the energy function be given as:

$$U(\mu, g) = \sum_{\forall \mathbf{x}_i \in \mathcal{G}} \sum_{\forall \mathbf{x}_j \in \mathcal{G}(\mathbf{x}_i)} (\mu(\mathbf{x}_i) - a_{i,j}g(\mathbf{x}_j))^2 \quad (6)$$

In this case, the solution to the minimization becomes,

$$\mu_o(\mathbf{x}_i) = E[\mathbf{x}_i] = \sum_{\forall \mathbf{x}_j \in \mathcal{G}(\mathbf{x}_i)} a_{i,j}g(\mathbf{x}_j), \quad (7)$$

where

$$a_{i,j} = \begin{cases} \frac{\hat{a}_{i,j}}{N} & \mathbf{x}_j \in \mathcal{G}_k(\mathbf{x}_i) \\ 0 & \mathbf{x}_j \notin \mathcal{G}_k(\mathbf{x}_i) \end{cases}, \quad (8)$$

and N is the number of points in the neighborhood $\mathcal{G}_k(\mathbf{x}_i)$. This is simply a linear estimator. This filtering operation is equivalent to the optimal estimation problem where the observation g is corrupted version of the original f with Gaussian noise, such that $\mu(\mathbf{x}) - g(\mathbf{x})$ is a zero-mean Gaussian random variable for all voxels in the image. It is well known that the best (ML) estimate of μ given the observations is the linear combination of the observations.

Using the estimator linear property, a variance estimator is just:

$$\sigma^2(\mathbf{x}_i) = E[(\mathbf{x}_i - \mu(\mathbf{x}_i))^2] = \sum_{\forall \mathbf{x}_j \in \mathcal{G}(\mathbf{x}_i)} a_{i,j}g(\mathbf{x}_j^2) - \mu(\mathbf{x}_i)^2. \quad (9)$$

Equations (7) and (9) are not simple to compute. The weights $a_{i,j}$ are unknown. If we assume that the image is a single region, then our problem is reduced to find the weights that meet the following requirement: to provide a reliable parameter estimations even at the true region's boundaries. To accomplish this, we propose an adaptive weight estimator that adjust the weights values in such a way to preserve the regions' boundaries.

From geometry we know that a discrete region R is a finite set of lines segments. Thus, if we had a line segment detector then we can use it to estimate region parameters even at the boundaries. We say that two "opposite" neighbor voxels of \mathbf{x} belong to the same local line segment if they share the same property, where "opposite", in a 3x3x3 neighborhood system, can be written as

$(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{G}(\mathbf{x})$ but $\mathbf{x}_i \notin \mathcal{G}(\mathbf{x}_j)$. Figure 1 shows simple 2D local line segments in a 3x3 neighborhood system. Although there are different ways to test this similarity, we opted for this simple test: A neighbor voxel property $p(\mathbf{x}_i)$ is similar to $p(\mathbf{x})$ if $|p(\mathbf{x}_i) - p(\mathbf{x})| < T_o \sigma_p(\mathbf{x})$, $\mathbf{x}_i \in \mathcal{G}(\mathbf{x})$, where T_o is a constant greater than zero and $\sigma_p(\mathbf{x})$ is a local dispersion measure given by:

$$\sigma_p(\mathbf{x}) = \sigma_o + \frac{1}{N} \sum_{\forall \mathbf{x}_k \in \mathcal{G}(\mathbf{x})} (|p(\mathbf{x}_k) - p(\mathbf{x})|), \quad (10)$$

where N is the number of neighbor points, and σ_o the current estimated variance from (9). From the former definitions, our line segment detector can be defined as:

$$\mathcal{L}_p\{\mathbf{x}, \mathbf{x}_i, \mathbf{x}_j\} = \begin{cases} 1 & \text{if } |p(\mathbf{x}) - p(\mathbf{x}_i)| < T_o \sigma_p(\mathbf{x}), \\ & \mathbf{x}_i \in \mathcal{G}(\mathbf{x}), \mathbf{x}_i \notin \mathcal{G}(\mathbf{x}_j) \\ & \text{and} \\ & |p(\mathbf{x}) - p(\mathbf{x}_j)| < T_o \sigma_p(\mathbf{x}), \\ & \mathbf{x}_j \in \mathcal{G}(\mathbf{x}), \mathbf{x}_j \notin \mathcal{G}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Which says that $\{\mathbf{x}, \mathbf{x}_i, \mathbf{x}_j\}$ belong to the same line segment if $\mathcal{L}_p\{\mathbf{x}, \mathbf{x}_i, \mathbf{x}_j\} = 1$.

For most medical images the estimated voxel mean value and the estimated variance can be used as voxel properties in the line segment detector. The only problem is that these properties are uncertain at the beginning; but they can be estimated iteratively, using (7) and (9) and the initial conditions are $\mu_0(\mathbf{x}_i) = g(\mathbf{x})$, the voxel value, and $s_0(\mathbf{x}_i) = g^2(\mathbf{x}_i)$, the voxel square value.

Voxels belonging to the same line segment are connected, thus those voxels can be used in the parameter estimation. Using the line detector we have a way to estimate a connectivity. Therefore, we use this estimated connectivity information to enhance the weights $a_{i,j}^n$, of those voxels connected to \mathbf{x}_i :

$$a_{i,j}^n = k_i^n \left(a_o + \sum_{\forall \mathbf{x}_k \in \mathcal{G}(\mathbf{x}_i)} \left(\frac{w_\mu \mathcal{L}_{\mu_n}(x_i, x_j, x_k)}{w_\sigma \mathcal{L}_{\sigma_n^2}(x_i, x_j, x_k)} \right) \right) \quad i \neq j \quad (12)$$

$$a_{i,i}^n = k_i^n \left(a_o + \sum_{\forall \mathbf{x}_j \in \mathcal{G}(\mathbf{x}_i)} \sum_{\forall \mathbf{x}_k \in \mathcal{G}(\mathbf{x}_i)} \left(\frac{w_\mu \mathcal{L}_{\mu_n}(x_i, x_j, x_k)}{w_\sigma \mathcal{L}_{\sigma_n^2}(x_i, x_j, x_k)} \right) \right)$$

where a_o is the initial weight, w_μ, w_σ are the weight gains related to the local dispersion $w_\mu = 1.0 / (|f(x_i) - f(x_j)| + \sigma_p(\mathbf{x}))$, and k_i the normalization constant such that $k_i^n = 1 / \sum_{j=1}^N a_{i,j}$, and N the number of points in the neighborhood. In other words, we increase the weight $a_{i,j}^n$ if \mathbf{x}_i and \mathbf{x}_j where estimated connected by the line segment detector for each parameter. This process is repeated several times until the local dispersion (10) is close to zero.

2.3. Connectivity computation

At this point we have associated an estimate of the local mean value and local variance to each voxel in the image with special considerations of boundaries to avoid blurring across them. Using these estimates the voxel connectivity computation is just a simple detection problem, where we want to establish for every voxel in the image if their neighbor voxels are connected to them. Assuming Gaussian textures and noise, the maximum likelihood level detector can be written as a quadratic Mahalanobis distance criterion[2]. The quadratic Mahalanobis distance is given by:

$$d_i^2(\mu(\mathbf{x}_j)) = [\mu(\mathbf{x}_i) - \mu(\mathbf{x}_j)]^T \mathbf{C}_i^{-1} [\mu(\mathbf{x}_i) - \mu(\mathbf{x}_j)], \quad (13)$$

where \mathbf{C}_i is the correlation matrix, where for the scalar case $\mathbf{C}_i = \sigma^2(\mathbf{x}_i)$. For uniform intensity illuminated regions, we establish connectivity between neighbor voxels if the observed Mahalanobis distance between them is less than some threshold d_{min} . This criterion is sensitive to illuminations variations; but for illuminated degraded regions, like those common in medical images, a simple criterion can be established for the scalar case. Let the observed scalar Mahalanobis distance between voxels be written as:

$$d_i(g(\mathbf{x}_j)) = \frac{|\mu(\mathbf{x}_i) - g(\mathbf{x}_j)|}{\sigma(\mathbf{x}_i)}. \quad (14)$$

Therefore, the connectivity estimation can be more accurate if we are able to express the the Mahalanobis distance between the non-degraded voxels $d_i(f(\mathbf{x}_i))$. Assume that the local intensity field function is given by

$$I(\mathbf{x}_i + \Delta_x) = (1 - \alpha_x) I(\mathbf{x}_i), \quad (15)$$

where Δ_x is just a small displacement, and α_x the field correction in that direction, which models a locally exponential field distribution. Therefore from our image model (1),(14) and (15); two neighbor voxels are connected if:

$$d_i(g(\mathbf{x}_j)) - \alpha_x \frac{\mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} \simeq d_i(f(\mathbf{x}_j)) \leq d_{min} \quad (16)$$

Although α_x is directionally dependent; in this study we are assuming that α is isotropic at neighbor voxels.

One independent statistical test that can be performed at textured regions is the ratio of the variances

$$F_{i,j} = \frac{\sigma_g^2(\mathbf{x}_i)}{\sigma_g^2(\mathbf{x}_j)} \simeq \frac{\sigma_f^2(\mathbf{x}_i) + (\sigma_\eta^2)/I^2(\mathbf{x}_i)}{\sigma_f^2(\mathbf{x}_j) + (\sigma_\eta^2)/I^2(\mathbf{x}_j)}, \quad (17)$$

This ratio follows a f distribution, and is robust to the noise η , if the square field intensity function is greater than the noise variance, $I^2(\mathbf{x}) \gg \sigma_\eta^2$. If we use similar number of sample points we say that both $\sigma^2(\mathbf{x}_i)$ and $\sigma^2(\mathbf{x}_j)$ estimate

the same variance if this ratio is less than a given threshold, t_h . That is, if $F_{i,j} < t_h$, where $t_h \geq 1$ then there is an arc that joins the neighbor voxel $\mathbf{x}_j \in \mathcal{G}(\mathbf{x}_i)$ to the voxel $\mathbf{x}_i \in \mathcal{G}(\mathbf{x}_j)$. In this study we decided to implement both tests over the images. This allows us to discriminate between regions with similar mean value but different texture properties, like some structures present in the musculoskeletal system.

Once we have built the connectivity graph, the segmented image is just the maximal sets of voxels all belonging to the same connected components; but there can be some leakage among the segmented regions, due to noise and some inaccurate mean and variance estimations. To avoid that, a simple 3D morphological closing operation can be applied to connectivity graph which will help to split loose connected regions by removing all isolated joining arcs. A better solution is the use of hysteresis with the selection of two different thresholds. After that the final connectivity map is formed by the connectivity of the high threshold plus the neighbor components from the lower threshold.

2.4. Relaxation labeling and merging

Real images do not follow our basic image model, thus the segmentation resulting from the former approach is not free from misclassification of the volume image. The problem of isolated miss-segmented voxels and noisy regions' boundaries can be addressed by assuming a Markov random field model of the labeled image, where the voxel label probability is a function of the neighbor labels. Following the work of Geman and Geman [3] we adopted a Gibbs random field for the estimation of the voxels probabilities given the distribution of the neighbor labels then we relax the labeling to correct errors in the segmentation by minimizing the *a posteriori* probability. Although finding a global minimum of it is possible, this process is very time consuming. A usual approach is to rely on the iterated conditional mode (ICM)[12]. This technique iteratively minimizes the function by using the previous solution to estimate the Gaussian PDF and the probabilities at every step. Although this is no guarantee it will reach a global minimum, given good initial conditions it will converge to a mean-optimal local minimum.

Furthermore an incorrect selection of the threshold value (d_{min} or t_h) will cause an incorrect segmentation of the image. A high threshold will cause some regions to be merged and a low threshold will result in multiple broken regions. If an image region has been broken into several contiguous patches, then the problem can be addressed by merging those patches. Following our image model, we determine that neighboring image patches whose means are equal belong to the same region. A simple statistical *t*-test can be used to compare the two patches' means and decide if they

are equal. Thus, two patches P_i, P_j are merged if the *t*-test confirms that their local means at their boundaries are the same with a confidence of α . This simple criterion is able to merge most of the small, but with similar mean, into the big segments.

3. Motion estimation and tracking

Our goal is the segmentation and motion analysis of a series of MRI scan of the joints. There are several anatomical components in the joints whose motion is very important to analyze. Joint motion is very complex due to the non-rigid motion of the soft tissues involved, like tendons, ligaments, muscle, etc. These deformations cause topological changes that make the application of automatic 4D segmentation difficult. Although the segmentation of the next volume in the sequence can be done by applying the same segmentation algorithm, there are certain problems in using this approach. First, the boundary between low contrast structures is not well defined thus noise and sampling makes the boundaries found by the segmentation to change between series. The second problem resides in the region labeling of the image. There are so many structures that a consistent labeling at each volume in the sequence is difficult. Third, the segmentation algorithm is computationally demanding thus segmenting each volume using this process is not practical. Giving these three objections we decided to use an alternative approach for the segmentation of the remaining volumes in the series.

A motion estimation and tracking algorithm will provide the information needed to pass the segmented image from one frame to the other. Since the regions have both rigid and nonrigid components, this has to be taken into account when performing the estimation of the 3D motion. Basically our algorithm estimates the motion vector of each voxel after the registration of selected feature points in the image.

Although there have been some research in this area of deformable motion tracking and analysis [8, 13, 15, 7, 9], most of them concentrate in the time evolution of a single structure in the image or a simple parametric deformable models. The analysis of the joint kinematic involve the motion of many structures making the current approaches unsuitable. Therefore we decided to develop a more general approach. This takes into account the local deformations of the soft tissues by using the *a-priori* knowledge of the material properties of the different structures found at the segmentation step. Also, this knowledge allows us to apply two different strategies; one for recovering the rigid motion and other for the soft tissues. Once the selected points have been registered, the motion vectors of every voxel in the image are computed by interpolating the motion vector of these selected points. Once the motion vector for each voxel has been estimated the segmentation of the second volume

is just the propagation of the segmented data of the former volume. These process is repeated until all the volumes in the sequence are segmented.

3.1. Motion estimation

Researchers have proposed the use of finite element models (FEM) for the analysis of images [6] as well its application to the time evolution analysis [8, 15, 9]. Following a similar approach, our algorithm recovers the point correspondence by minimizing the "total energy" of a mesh of mass and springs system which models the physical properties of the anatomy. The main difference is that our mesh is not constrained to a single structure in the image, but it models the whole volumetric image which topology properties are driven by the original data and its physical properties by our *a-priori* knowledge and the segmented data. Basically our motion estimation approach is a FEM-based point correspondence recovery algorithm between two consecutive frames in the sequence. In our model each node of the mesh is an automatically selected feature point of the image we want to track, and the spring stiffness is computed using the segmented image and our *a priori* knowledge of the human anatomy.

Many deformable models assume that a vector force field can be extracted from the image which drives spring attached point masses. Most of them use this approach to build semi-automatic feature extraction algorithms. Using a similar approach, let us assume that the image sampled at $t = n$ is a set of three dynamical scalar fields: $\Phi(\mathbf{x}, t) = \{g_n(\mathbf{x}), \|\nabla g_n(\mathbf{x})\|, \nabla^2 g_n(\mathbf{x})\}$ (The gray value, the magnitude of the image gradient, and a Laplacian of the image), so that a change in the field $\Phi(\mathbf{x})$ causes a quadratic change in the scalar field energy $U_\Phi(\mathbf{x}) \propto (\Delta\Phi(\mathbf{x}))^2$. Furthermore, assume that the image underlying structures are modeled as a mesh of spring attached point masses in a state of equilibrium with these scalar fields. Although equilibrium implies that there is an external force field, its shape is not important in our analysis. Let us assume that the point distribution changes in time, thus the total energy change in (Δt) , is:

$$\Delta U_n(\Delta \mathbf{X}) = \sum_{\forall \mathbf{X} \in g_n} \left[\begin{array}{l} \alpha(g_n(\mathbf{x}) - g_{n+1}(\mathbf{x} + \Delta \mathbf{x}))^2 \\ + \beta(\|\nabla g_n(\mathbf{x})\| - \|\nabla g_{n+1}(\mathbf{x} + \Delta \mathbf{x})\|)^2 \\ + \gamma(\nabla^2 g_n(\mathbf{x}) + \nabla^2 g_{n+1}(\mathbf{x} + \Delta \mathbf{x}))^2 \\ + \eta \frac{1}{2} \Delta \mathbf{X}^T \mathbf{K} \Delta \mathbf{X} \end{array} \right] \quad (18)$$

where α, β, γ weights the contribution of every individual field change, and η weights the gain in the strain energy. \mathbf{K} represent the FEM stiffness matrix, and $\Delta \mathbf{X}$ the FEM node displacement matrix. Analyzing (18) we see that any change in the image fields or in the mesh point distribution will increase the system total energy. Therefore, using the above energy definition we state without proof that the point correspondence form g_n to g_{n+1} is given by the mesh configuration whose total energy variation is minimum. That

means, the point correspondence is given by:

$$\hat{\mathbf{X}} = \mathbf{X} + \Delta \hat{\mathbf{X}} \quad (19)$$

where

$$\Delta \hat{\mathbf{X}} = \min_{\Delta \mathbf{X}} \Delta U_n(\Delta \mathbf{X}) \quad (20)$$

3.2. Feature points extraction

Although (19) can be used to estimate the motion (point correspondence) of every voxel in the image, the number of voxel points $\geq 10^6$ and the complex nature of the equation makes its global minimization difficult. To simplify the problem we opted to construct a coarse FEM mesh using selected points from the image. Therefore, the energy minimization will give us the estimated point correspondence of them.

The selection of those points, is not trivial. First, for practical purposes the number of points have to be very small $\simeq 10^4$, therefore the selected points must be enough to describe the whole image motion. Region boundaries are very important features because, for our motion tracking proposes, boundary tracking is enough for an accurate region motion description. Furthermore, at regions boundaries the magnitude of the gradient is high and the Laplacian is a zero crossing point, making them easy features to track. These facts, compels the selection segmented boundary points in the construction of our FEM.

Although boundary points represent a small subset of the image points, they are still too many for practical proposes. To reduce the number of points we decided to use constrained random sampling of the boundary points for the point extraction step. The constraint consists of avoiding the selection of a point too close to the ones already selected. This constraint allow a more uniform selection of the points around the boundaries. Finally, to reduce the motion estimation error at region internal points, we randomly selected a few more points from the image using the same distance constraint. Experimental results show us that between 5000 and 10000 points are enough to estimate and describe the motion on a typical volumetric image of 256x256x34 voxels, from where we arbitrary choose 75 % of the total as boundary points and the remaining 25 % as interior points.

3.3. FEM mesh construction

Once we have selected an appropriate set of feature points to track, the next step is the construction of a FEM mesh from those points. This mesh constrains the kind of motion allowed by coding the material properties and the different regions iterations in its topology. The first step in the mesh construction is to find for every nodal point, the neighboring nodal point. This corresponds to build the

$k^{l,m}$	Humeral head	muscle	tendon	cartilage
Humeral head	10^4	0.15	0.7	0.01
muscle	0.15	0.1	0.7	0.6
tendon	0.7	0.7	10^1	0.01
cartilage	0.01	0.6	0.01	10^2

Table 1. Relative stiffness and bounding stiffness of selected anatomical structures in the shoulder.

Voronoi diagram of the mesh. Its dual, the Delaunay triangulation, represents the best possible tetrahedra finite element for a given nodal configuration[11]. Although we can compute the Voronoi map for a N nodes mesh with a $O(N \log(N))$ algorithm, we also need to know the closest node point for every voxel in the image. Therefore we used instead a simple dilation approach for the construction of the Voronoi digram. The dilatation approach dilates every nodal point in the discrete volume and testing where it contacts other dilated feature, this procedure at the same time labels every voxel in the image with the corresponding closest nodal point of the finite element mesh. Once every point \mathbf{x}_i has found its neighbor \mathbf{x}_j a spring its attached to the point pair with constant $k_{i,j}^{l,m}$, which defines and average iteration strength between material l and material m . The spring stiffness $k_{i,j}^{l,m}$ is defined by the material iteration properties of the connected points that are predefined by the user. If the connected points belong to the same region; then the stiffness constant $k_{i,j}^{l,l}$ is related to the material elasticity properties; while $k_{i,j}^{l,m}$ for different material codes the average iteration force between both at boundaries. Table I shows a typical input matrix which defines the iteration between components. Although in theory we must define all the iteration between all adjacent regions, in practice we just define the interaction between the major anatomical components in the image, and the rest are left as an arbitrary constant. This arbitrary selection, makes the motion estimation less accurate; but the introduced error is not significant compared to others induced errors in our assumptions.

3.4. Energy minimization

The problem of minimizing U can be addressed by dividing the problem in two: Rigid motion estimation and deformable motion estimation. Both use the same energy function; but we use different strategies. The rigid motion estimation minimization strategy uses the fact that the mesh deformation energy $(\Delta \mathbf{X}^T \mathbf{K} \Delta \mathbf{X})/2$ is very close to zero. From the segmentation, and our *a-priori* knowledge of the

anatomy we know which points belong a rigid body. If we select those points for every individual rigid region then the rigid motion energy minimization problem is equal to finding the rigid motion rotation \mathbf{R}_i and translation \mathbf{T}_i that minimize every region own energy:

$$\begin{aligned} \Delta \hat{\mathbf{X}}_{rigid} &= \min_{\Delta \mathbf{X}} U_{rigid} \\ &= \sum_{\forall i \in rigid} (\Delta \hat{\mathbf{X}}_i = \min_{\Delta \mathbf{X}_i} U_n(\Delta \mathbf{X}_i)), \end{aligned} \quad (21)$$

where

$$\Delta \mathbf{X}_i = \mathbf{R}_i \mathbf{X}_i + \mathbf{T}_i - \mathbf{X}_i \quad (22)$$

and $\Delta \hat{\mathbf{X}}_i$ is the optimum displacement matrix of the points that belong to a rigid body R_i . This minimization problem has only 6 degrees of freedom for every rigid body: 3 in the rotation matrix and 3 in the translation matrix. Therefore, the 12 coefficients (9 rotation parameters, 3 translations) can be found via a 6 dimensional steepest decent technique if the difference between every two sequences is small enough.

Once we have found the rigid motion parameters, we proceed to estimate deformable motion throughout the minimization of the total system energy U . This problem can not be simplified as much as the rigid motion case, and without further considerations the number of degrees of freedom in a 3D deformable object is $M = 3 * N$ where N is the number of node points in all the mesh. The nature of the problem allows the use of a simple gradient decent technique for each node in the mesh, which is derived from the physics of the problem. From the potential and kinetic energies the Lagrangian of the system can be used to found the Euler-Lagrange equations for every node of the system where the driving local force is just the gradient of the energy field. For every node in the mesh its local energy is given by:

$$\begin{aligned} U_{(\mathbf{x}_i),n}(\Delta \mathbf{x}) &= [\alpha(g_n(\mathbf{x}_i + \Delta \mathbf{x}) - g_{n+1}(\mathbf{x}_i))^2 \\ &\quad + \beta(\|\nabla g_n(\mathbf{x}_i + \Delta \mathbf{x})\| - \|\nabla g_{n+1}(\mathbf{x}_i)\|)^2 \\ &\quad + \gamma(\nabla^2 g_n(\mathbf{x}_i + \Delta \mathbf{x}) - \nabla^2 g_{n+1}(\mathbf{x}_i))^2 \\ &\quad + \eta \frac{1}{2} \sum_{\mathbf{x}_j \in \mathcal{G}_m(\mathbf{x}_i)} k_{i,j}^{l,m} (\mathbf{x}_j - \mathbf{x}_i - \Delta \mathbf{x})^2] \end{aligned} \quad (23)$$

where \mathcal{G}_m represents a neighborhood in the Voronoi diagram.

Therefore, every process handles a 3-degrees-of-freedom problem whose minimization is performed using a simple gradient decent technique that iteratively reduces the local node energy. The local node gradient decent equation is

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) - \nu \nabla U_{(\mathbf{x}_i),n}(\Delta \mathbf{x}), \quad (24)$$

where the gradient of the mesh energy is analytically computable and the gradient of the field energy is numerically estimated from the image at two different resolutions, and $\mathbf{x}(n+1)$ is the next node position, and ν weights the gradient contribution.

At every step in the minimization, each process takes into account the neighboring nodes former displacement found for every other node in the mesh. The process is repeated until the total energy gets into a local minimum, which for small deformations is close to (or is) the global minima. The displacement vector found represents the estimated motion at the node points.

3.5. Dense motion field estimation and motion tracking

The sampled displacement field ΔX found in the minimization process is used to estimate the dense motion field needed to track the segmentation from one frame to the next. The dense motion estimation is found by weighting the contribution of every neighbor node in the mesh. Assuming a constant velocity model, let us call $\mathbf{v}(\mathbf{x}, t) = \Delta \mathbf{x}(t)/\Delta t$ the estimated velocity at voxel \mathbf{x} at time t , then the dense motion field is estimated by:

$$\mathbf{v}(\mathbf{x}, t) = \frac{c(\mathbf{x})}{\Delta t} \sum_{\forall \Delta \mathbf{x}_j \in \mathcal{G}_m(x_i)} \frac{k^{l,m} \Delta \mathbf{x}_j}{|\mathbf{x} - \mathbf{x}_j|} \quad (25)$$

where

$$c(\mathbf{x}) = \left[\sum_{\forall \Delta \mathbf{x}_j \in \mathcal{G}_m(x_i)} \frac{k^{l,m}}{|\mathbf{x} - \mathbf{x}_j|} \right]^{-1}, \quad (26)$$

$k^{l,m}$ the stiffness between material l and m associated to voxel \mathbf{x} and voxel \mathbf{x}_j respectively, ΔT the time between successive samples, $|\cdot|$ is a simple Euclidean distance, and the interpolation is done using the neighbor nodes of the closest node to the voxel \mathbf{x} , where $\mathbf{x} \in \mathcal{V}_m(x_i)$. This interpolation weights the contribution of every neighbor node by its material property $k_{i,j}^{l,m}$ thus the estimate voxel motion is similar for every homogeneous region even at the region's boundary.

To fill the next volume in the series with the segmented data at time $(t + \Delta t)$, we need the estimated velocity for every voxel in the next volume. That means that we need a reverse mapping of the estimated motion, which is given by:

$$v(\mathbf{x}, t + \Delta t) = \frac{1}{H} \sum_{\forall [\mathbf{x}_j + v(\mathbf{x}_j, t)] \in \mathcal{S}(\mathbf{x})} v(\mathbf{x}_j, t). \quad (27)$$

where H is the number of points that fall into the same voxel space $\mathcal{S}(\mathbf{x})$ in the next volume. This mapping, does not fill all the space at $t + \Delta t$; but a simple interpolation between mapped neighbor voxels can be used to fill out that space. Once we have an estimation of the velocity for every voxel in the next frame, its segmentation is simply:

$$L(\mathbf{x}, t + \Delta t) = L(\mathbf{x} - v(\mathbf{x}, t + \Delta t), t) \quad (28)$$

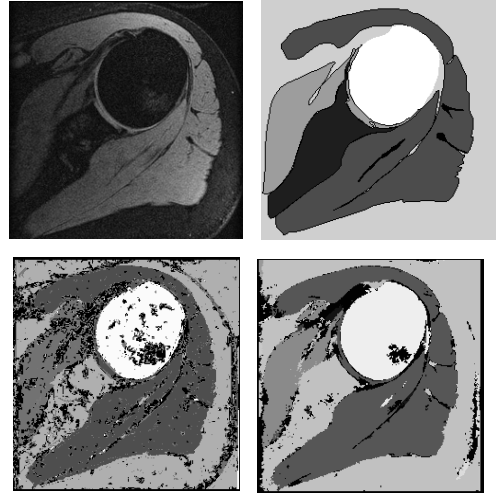


Figure 2. Single slice of segmentation of the 3D shoulder image. Left to right, top to bottom: original raw data, target segmentation, statistical region growing result, final result

where $L(\mathbf{x}, t)$ is the label at voxel \mathbf{x} from the previous image, and $L(\mathbf{x}, t + \Delta t)$ is the voxel label at the new segmented image.

4. Results

Our 4D analysis approach has been tested in several MRI volumetric images sequences. In these section we present the segmentation and motion tracking results for a rotating shoulder sequence and a knee extension sequence. In both cases the MRI image parameters for 3D GRE and 3D PS-GRE sequences were optimized for contrast, noise and field coil placement using five volunteers.

4.1. 4D shoulder

Using the optimized imaging sequence, a volunteer was imaged when she rotated her arm stepwise from the neutral position to the maximal external rotation. A total of eight useful image sequences were acquired. Each image size were 256x256x28 with a spatial resolution of 0.43x0.43x1.7 millimeters for each voxel. The left image in figure 2, shows a single slice of the shoulder image. A manually segmentation of selected structures in the anatomy in this slice is shown in the next figure, which represents the target segmentation at that slice of four anatomical structures: the humeral head, two different muscle groups and the scapula. The next image shows the segmentation after the connectivity estimation. This image shows that our parameter es-

timization algorithm is able to estimate the local statistical parameters in real images and useful for our simple region growing algorithm. The third image shows the final segmentation results, where the different gray scales represents different components of the shoulder anatomy. As we can see our segmentation approach did a very good job on the humeral head and muscle tissue. Although the tendons and the cartilage were not uniformly segmented; they were not confused with other major anatomical components. The scapula and fat were classified by our algorithm as a single structure due the lack of contrast on the original image. The segmentation error was measured using the relative volumes differences between the manually extracted humeral head and it corresponding automatic segmentation. This relative difference was close to 1.0%. The complete volume segmentation was evaluated by an expert radiologist. In the future we will use knowledge-based techniques to handle this segmentation abnormalities.

The segmented image was used as an input for the motion estimation and tracking algorithm. The relative material properties were defined for each one of the major components in the image like bone, muscle. Once these parameters and the $\alpha, \beta, \gamma, \eta$ coefficients of the energy term were defined the algorithm was run on the image sequence, where a set of 5000 points were automatically selected for the motion estimation algorithm: 4000 at boundaries and 1000 elsewhere. The algorithm was able to estimate the rotation of the humeral head, as well the displacement of all the soft tissues that surround it. Finally, figure 3 shows a lateral view of the surfaces renderings of three frames of the sequence. The humeral head is in gray, the muscle tissue is a transparent light gray. Notice the difference in position of the humeral head throughout the sequence.

4.2. 4D knee

The knee extension sequence was analyzed in a similar way; but we in this example used a relatively higher resolution image as an input for the segmentation algorithm. The image sequence consisted on a set of six $60 \times 256 \times 256$ volumetric frames with a spatial resolution of $0.47 \times 0.47 \times 1.4$ cubic millimeters. Figure 4 shows at the left a single slice of the first volume of the knee sequence. The next image is the segmentation after connectivity estimation, while the third shows the segmentation results of the region of interest, which was a subset of the original image. The segmentation of this image was not perfect due its lack of contrast among bone tissue and fat, but it was good enough for our purposes. Again the segmentation evaluation was carried out by our expert radiologist.

The estimated motion field for frame 1 to 2 is shown on the both right image on figure 4, where 3D motion vectors show how the tibia has a large displacement to the left,

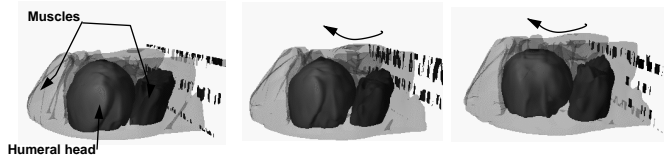


Figure 3. Rotating shoulder sequence. From left to right: surfaces renderings of frames 1,4 and 8.

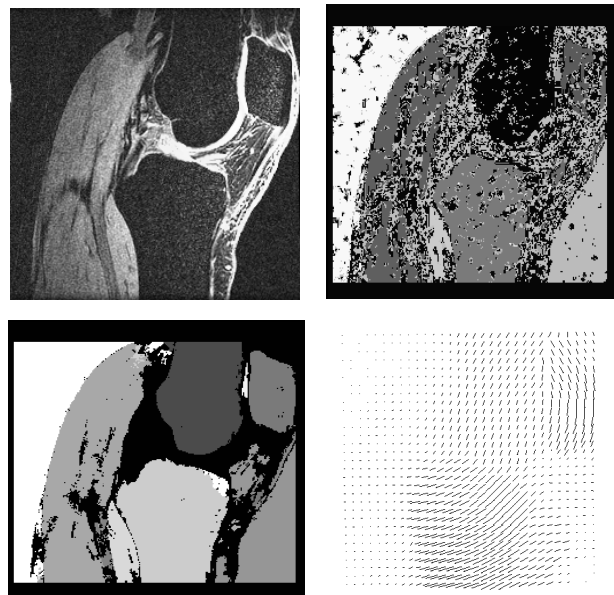


Figure 4. Single slice of segmentation of the 3D knee image. Left to right, top to bottom: original raw data, initial segmentation, final segmentation, motion flow vectors from frame 1 to 2.

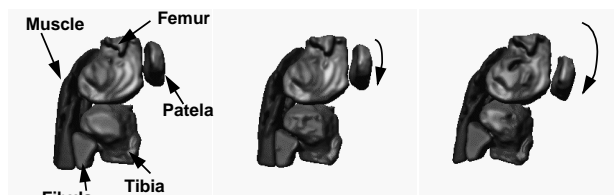


Figure 5. Surface renderings of the knee extension sequence. From left to right, side views of frames 1,4 and 7.

while most of the leg was moving down, which corresponds to the real motion present. The energy coefficients and the rest of the motion parameters were the same as the ones used for the shoulder motion. Finally, figure 5 shows the complete knee extension sequence which is visualized as a set of surfaces. Notice the different position on the patella (knee cap) throughout the sequence.

The visualization of the volumetric data included textured surfaces renderings extracted from the segmented data. The texture information was obtained from the original MR data by projecting the surrounding tissue density values into the surface. This approach enhances the visualization of the tendon-bone attachments and bone-cartilage contact points. The motion visualization was done by generating a series of movies which allowed a more vivid visualization of the iterations between different joint structures, where figures 5 and 3 show some frames of two of the generated movies.

5. Conclusion

We have presented a new approach to 3D segmentation and motion tracking which allows an automatic segmentation and tracking of the soft tissue and rigid structures in a sequence of volumetric images. Our approach was used for the 4D visual analysis of two major joints: the shoulder and knee. These were imaged using an optimal MRI pulse sequence and acquired during stepwise motion. Our approach requires a minimum of interactivity with the user, eliminating the need for exhaustive tracings and editing of the image data. This 4D analysis is based on the hierarchical application of the hierarchical application of three unsupervised segmentation techniques and a solid finite element model as base for the motion tracking algorithm. This approach allowed us to present and visualize the interaction of muscle-tendon-bone which can be very useful in understanding the role of each structure in the kinematics of the joints, and will enable, in the future, the development of a complete kinetic, bio-mechanical models of the human joints.

6. Acknowledgment

This work is supported in part by the NSF/NYS Grant to the Center for Electronic Imaging Systems at the University of Rochester, the University of Rochester departments of Electrical Engineering and Radiology and by a Fulbright Scholarship from the U.S.-Mexico Fulbright-CONACYT Commission, Mexico.

References

[1] E. A. Ashton, K. J. Parker, M. J. Berg, and C. W. Chen. A novel volumetric feature extraction technique with applica-

- tions to MR images. *IEEE Transactions on Medical Imaging*, 16:365–371, 1997.
- [2] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.
- [3] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images. *IEEE Trans. Patt Anal. Mach. Intel*, 6:721–741, 1984.
- [4] M. W. Hansen and W. Higgins. Relaxation methods for supervised image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):949–962, 1997.
- [5] R. M. Haralik and L. Shapiro. Image segmentation techniques. *Computer Vision Graphics and Image Processing*, 29:100–132, 1985.
- [6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, pages 32–331, 1988.
- [7] J. C. McEachena and J. S. Duncan. Shape-based tracking of left ventricular wall motion. *IEEE Transactions on Medical Imaging*, 17(3):270–283, 1997.
- [8] T. McInerney and D. Terzopoulos. Dynamic finite element surface model for segmentation and tracking in multidimensional medical images with applications to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, 1995.
- [9] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis: application to four dimensional medical images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1067–1079, 1996.
- [10] L. Onural, M. B. Alp, and M. I. Gurelli. Gibbs random field model based weight selection for the 2-D adaptive weighted median filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:831–837, 1994.
- [11] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
- [12] T. N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Trans. Signal Proc.*, SP-40:901–914, 1992.
- [13] J. Park, D. Metaxas, A. Young, A. Alistair, and L. Leon. Deformable models with parametric functions for cardiac motion analysis from tagged MRI data. *IEEE Transactions on Medical Imaging*, 15(3):278–289, 1996.
- [14] C. E. Priebe, D. J. Marchette, and G. W. Rogers. Segmentations of random fields via borrowed strength density estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:494–498, 1997.
- [15] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.
- [16] J. Thiran, V. Warscotte, and B. Macq. Queue-based region growing algorithm for accurate segmentation of multidimensional digital images. *Signal Processing*, pages 1–10, 1997.
- [17] U. Tiede, K. H. Hoehne, M. Bomans, A. Pommert, M. Riemer, and G. Wiebecke. Investigation of medical 3D-rendering algorithms. *IEEE Computer Graphics and Applications*, 10(2):41–52, 1990.
- [18] K. L. Vincken, A. S. E. Koster, and M. A. Viergever. Probabilistic multiscale image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):109–120, 1997.