Published in final edited form as:

IEEE Trans Med Imaging. 2008 September ; 27(9): 1252–1267. doi:10.1109/TMI.2008.920612.

Computer Aided Evaluation of Ankylosing Spondylitis Using High-Resolution CT

Sovira Tan,

National Institute of Arthritis and Musculoskeletal and Skin diseases, National Institutes of Health, Clinical Center, Bethesda, MD 20892 USA (e-mail: tanso@mail.nih.gov).

Jianhua Yao,

Department of Diagnostic Radiology, National Institutes of Health, Clinical Center, Bethesda, MD 20892 USA.

Michael M. Ward,

National Institute of Arthritis and Musculoskeletal and Skin diseases, National Institutes of Health, Clinical Center, Bethesda, MD 20892 USA.

Lawrence Yao, and

Department of Diagnostic Radiology, National Institutes of Health, Clinical Center, Bethesda, MD 20892 USA.

Ronald M. Summers [Member, IEEE]

Department of Diagnostic Radiology, National Institutes of Health, Clinical Center, Bethesda, MD 20892 USA.

Abstract

Ankylosing Spondylitis is a disease characterized by abnormal bone structures (syndesmophytes) growing at intervertebral disk spaces. Because this growth is so slow as to be undetectable on plain radiographs taken over years, it is desirable to resort to computerized techniques to complement qualitative human judgment with precise quantitative measures. We developed an algorithm with minimal user intervention that provides such measures using high-resolution computed tomography (CT) images. To the best of our knowledge it is the first time that determination of the disease's status is attempted by direct measurement of the syndesmophytes. The first part of our algorithm segments the whole vertebral body using a 3-D multiscale cascade of successive level sets. The second part extracts the continuous ridgeline of the vertebral body where syndesmophytes are located. For that we designed a novel level set implementation capable of evolving on the isosurface of an object represented by a triangular mesh using curvature features. The third part of the algorithm segments the syndesmophytes from the vertebral body using local cutting planes and quantitates them. We present experimental work done with 10 patients from each of which we processed five vertebrae. The results of our algorithm were validated by comparison with a semi-quantitative evaluation made by a medical expert who visually inspected the CT scans. Correlation between the two evaluations was found to be 0.936 ($p < 10^{-18}$).

Index Terms

Level sets on nonplanar manifolds; multiscale vertebra segmentation; ridgelines/crestlines; semi-synthetic digital phantoms

I. INTRODUCTION

Ankylosing spondylitis (AS) is a relatively rare inflammatory disorder that affects the spine and joints. It occurs in about 0.2% of the general population and is more common in men than women. It has a known association with an important immunogenetic component of DNA known as HLA-B27. The majority of persons affected by AS are HLA-B27 positive. Patients with AS may develop insidious onset of low back pain accompanied by morning stiffness. They may experience limitation of spinal motion and chest expansion. They may also develop inflammation of tendon–bone junctions and the eye. Commonly affected areas of bony involvement are the spine and sacroiliac joints [1].

AS is characterized by abnormal bone structures (syndesmophytes) growing at intervertebral disk spaces, which over many years may lead to spinal fusion [2]. Existing measures of the extent of spinal fusion in AS, such as the Bath AS Radiology Index [3] are based on qualitative examination of plain radiographs and are poorly sensitive to change. Most patients do not demonstrate any change in score over a period of one or two years [3]. An accurate and sensitive measure of the rate of syndesmophyte growth is important because it would permit the evaluation of new drugs that may slow the progression of spinal fusion in AS. High-resolution computed tomography (CT) appears to be an appropriate modality for obtaining precise measurements. To the best of our knowledge, the present paper is the first attempt at measuring the syndesmophytes directly. Traditionally clinical research has sought to link the status of the disease to biomarkers. Unfortunately none has been shown to have a good correlation with radiological evidence of the disease's progress [4]. Fig. 1 shows two views of the same vertebra with its syndesmophytes.

Manual segmentation of the syndesmophytes would be prohibitively cumbersome and imprecise. In [5] we presented a first algorithm with the objective of quantitatively characterizing them in terms of volume. Although it yielded good results it unfortunately required a significant amount of human intervention to fine tune parameters. In the new version presented here we were able to keep the parameters of the algorithm identical except in a few cases that are mentioned in due course. We managed to make the algorithm more robust mostly by: 1) making our segmentation multiscale and 2) introducing a novel level set implementation capable of evolving on a triangular mesh in order to extract the ridgelines of vertebral bodies.

The first part of our algorithm (Section II) segments the whole vertebra including the syndesmophytes using a 3-D multiscale cascade of level sets. Included in this section is a systematic method for making a good choice of parameters for the level sets. Section III describes a method for extending the domain of application of level sets from regular Cartesian grids to triangular meshes. That was used to evolve a geodesic active contour (GAC) on the isosurface of a segmented vertebra in order to detect the continuous ridgeline where syndesmophytes are located. The last part of the algorithm (Section IV) separates the syndesmophytes from the vertebral body using local cutting planes. In Section V we validate our method by correlating the volumes and heights of the syndesmophytes extracted by our algorithm with the scoring of a medical expert based on visual inspection. Ten patients were examined for this study. Fig. 2 visually sums up the three stages of our algorithm. Fig. 2(a) shows a segmented vertebra with its syndesmophyte (red arrow). Fig. 2(b) shows the

ridgeline of the upper end plate extracted using our level set on the mesh and finally Fig. 2(c) shows the syndesmophyte cut from the vertebral body.

II. SEGMENTATION OF THE VERTEBRAL BODY

A. Background on Vertebra Segmentation

The segmentation of vertebral bodies presents many challenges, some of which are typical of many other segmentation applications. Gaps in the cortical bone [Fig. 3(b)] and irrelevant, internal boundaries [Fig. 3(c)], respectively, lead to over- and under-segmentation. Those two problems need to be addressed simultaneously, which leads to severely conflicting requirements. More specific to our application is the double boundary problem. The cortical bone surrounding the vertebra [shown between the two arrows in Fig. 3(a)] has a much higher density than the trabecular bone inside. That forced us to adopt a multistage strategy in which successive level sets capture the different parts of the bone.

Previous work on 3-D segmentation of vertebral bodies has mainly relied on model based methods [6], [7]. In [8] Kang *et al.* present a bone segmentation algorithm based on region growing that they validated on the European spine phantom. The Mastmeyer *et al.* algorithm consists of a deformable mesh segmentation that is fine tuned by adaptive volume growing and morphological operations [9].

Our algorithm is not model based, is 3-D, and is based on a cascade of several level sets. It can be implemented using the open library ITK [10]. For the rest of Section II we used ITK functions for the implementation of all level sets. Another distinctive feature of our method is multiscaling. Multiscaling is an attractive way to make segmentation not only faster but also more robust and accurate. Previous work on multiscale segmentation has been carried out in 2-D [11], [12]. In 3-D, work has been published on geometric flows optimized for tubular structures [13]. Model-based multiscale segmentation approaches include [14]–[16]. Multiscale 3-D segmentation was also carried out using fuzzy clustering [17].

B. Methods

1) Level Sets—Level sets are evolving contours or surfaces that can expand, contract, and even split or merge [18], [19]. For the purpose of segmentation they are designed to deform so as to match an object of interest and stop at its edges. To solve our vertebra segmentation problem we use two different level set algorithms: the GAC [20] and what we call for convenience the classical level set (CLS) [18]. The GAC evolves according to the equation

$$\frac{d\psi}{dt} = \alpha g c |\nabla \psi| + \beta g \kappa |\nabla \psi| + \gamma \nabla g \nabla \psi. \quad (1)$$

The evolving contour is encoded as the zero level set of the distance function $\psi(\vec{x}, t)$. In other words points that verify $\psi(\vec{x}, t) = 0$ form the contour. By convention the distance is negative for points inside the contour and positive for the ones outside

$$\psi(\vec{x}, t) = \pm d \quad (2)$$

where d is the distance from point $\vec{x}$ to the zero level set contour.

The first term on the right-hand side of (1) is the propagation term that makes the contour move with velocity c . The second term, the curvature term, controls the smoothness of the contour using the mean curvature κ . The third term, the advection term, was introduced by Caselles *et al.* [20] to lock the contour to the boundary. The spatial function g derived from the image $I(\vec{x})$ to be segmented, contains the information about the objects' boundaries and

is called the speed function. The parameters α , β , and γ weight the importance of each term. How to select those parameters is the subject of Section II-B-4. The CLS is equivalent to the GAC without the advection term. We use the GAC for its robustness to gaps and the CLS for its flexibility.

2) Multiscale Algorithm—Our segmentation algorithm is the multiscale version of a previous algorithm [21]. Two scales are used: the segmentation starts at half-scale and is refined at full scale. A three-scale framework was also tried but it was found that at scale 1/4 many intervertebral disk spaces disappeared leading to severe segmentation leakage between vertebral bodies.

Fig. 4 shows a flowchart of our segmentation algorithm. We first linearly subsample our data (step 1). Then the original algorithm that consists of a cascade of level sets is applied to the obtained half-scale volume. The preprocessing described in the following subsection is applied to determine the parameters of the sigmoid used to compute the speed function of the first GAC (step 2.1). The first GAC roughly segments the interior of the vertebra (step 2.2). Its seed is the result of a fast marching (FM) stage starting from a seed point roughly placed by the user in the center of the vertebral body and lasting 20 iterations. The second level set, also a GAC, refines this segmentation using a Laplacian convolution of the image as the speed function (step 2.3). The third level set, a CLS, segments the cortical bone (step 2.4). A postprocessing step fills some remaining holes using a dilation followed by an erosion (step 2.5). The resulting segmentation is then super-sampled back to full scale (step 3) and refined using a CLS (step 4). Another postprocessing is conducted afterwards using a hole-filling algorithm (step 5). From a candidate point we shoot lines in the x , y , z directions. If all the lines meet segmented points we can classify the point as inside the vertebra [21]. Dilation and erosion are also used.

3) Preprocessing—The speed function g of the first GAC should ideally have values close to 1 where there are no boundaries (so that the level sets can expand rapidly) and values close to 0 where boundaries are present (so that the level sets stop). It is crucial to get the speed function right in order to avoid enhancing irrelevant edges or missing important ones. A popular way to obtain the speed function is to remap the gradient magnitude image with a sigmoid. The sigmoid pixel intensity remapping can be written [10]

$$I' = 1 - \frac{1}{1 + \exp\left(-\frac{I - \xi}{\eta}\right)} \quad (3)$$

where I is the original pixel intensity (here representing the gradient magnitude) and I' the remapped one. The two parameters ξ and η are typically computed using the equations [10]

$$\eta = \frac{K_1 - K_2}{6} \quad \xi = \frac{K_1 + K_2}{2} \quad (4)$$

where K_1 is the minimum gradient magnitude value along the object's boundary and K_2 the average gradient magnitude inside the object where the level set is initialized. It has been shown that those parameters are crucial for the accuracy of the segmentation [21].

It is however difficult to obtain a good estimation for K_1 and K_2 in a fully automatic manner. In our application the user is asked to roughly locate the center of the vertebra using a coronal and a sagittal view. This central point is in fact the only input asked from the user. It will be used throughout the whole algorithm. It is the seed of the FM. The seeds of the ridgeline extracting level sets will also be derived from it (Section III). And it is used here for computing K_1 and K_2 . From this approximate center we can evaluate the mean gradient

magnitude in a box inside the vertebra, which is our evaluation of K_2 . We can also evaluate the gradient magnitude at the vertebra's boundaries by shooting lines from this box and taking the maximum gradient magnitude along those lines away from the center. Having made a histogram from the boundary gradient magnitudes thus obtained, we take the mean of the 10% lowest values as our estimate for K_1 . Fig. 5 shows some examples of results of the search algorithm. The red box is where K_2 is computed.

We have found that an efficient way of correcting a failed segmentation is by modifying the values of K_1 and K_2 . To correct under-segmentation we increase K_1 , which has the effect of making the sigmoid-remapped boundaries thinner. Conversely, to correct over-segmentation we decrease K_1 , which makes the remapped boundaries thicker.

4) Parameter Selection—In our previous work we observed that segmentation results varied considerably with the level set parameters used. In particular the curvature weight (CW) β and advection weight (AW) γ are essential in controlling the robustness to gaps as well as the completeness of the segmentation. In order to make an informed choice of parameters we designed an experiment involving a semi-synthetic vertebra containing real structured noise like in Fig. 3(c). The semi synthetic vertebra was created by placing a synthetic cortical bone boundary inside a real vertebra. Inhomogeneities were added to the boundary in the form of gaps or higher density areas. Details can be found in [22]. Fig. 6 shows axial, sagittal, and coronal slices of the semi synthetic vertebra. The advantage of having a synthetic vertebra with ground truth is that a quantitative evaluation of the quality of a segmentation result can be obtained. This allowed us to test a large number of parameter combinations in order to gain an insight about how they should be chosen. We varied the parameters CW and AW from 1/16 to 16 following the geometric progression 2^i , i taking integer values from -4 to 4 . The total number of possible parameter combinations is therefore $9 \times 9 = 81$. For each combination a segmentation is obtained and its mean error distance to the ground truth is computed. We first carried out the experiment for the parameters of the first GAC every other parameters being kept constant [22].

Some typical results are presented in a visual manner as 2-D functions of CW and AW [Fig. 7(a)]. Each grey level in the array is proportional to the error. The plot tells us which combinations of parameters lead to failure and which lead to good segmentation. Two areas are to be avoided. The first is high CW coupled with low AW. This combination yields under-segmentation because the curvature force is retractile for convex surfaces. The second one is low CW associated with low AW. That combination essentially makes the GAC similar to a FM and is not robust to boundary gaps. It produces over-segmentation caused by leakages. It is also interesting to note that for a large area on the right (high AW) parameter variations do not cause significant changes in error values. Those results highlight the importance of the advection force introduced by Caselles *et al.* [20]. In Section III we will also show the importance of this force for ridgeline detection on a surface mesh. In view of Fig. 7(a) a good (CW, AW) pair will have medium CW and high AW. We chose (2, 8).

Fixing the (CW, AW) parameters of the first GAC to (2, 8) we vary those of the second one. The results, shown in Fig. 7(b), are similar to the ones obtained for the first GAC except that under-segmentation areas have become much more important than leakage areas. Note that maps Fig. 7(a) and (b) have each been normalized separately. In fact if the parameters of the first GAC are well chosen the parameters of the second one do not matter much. The errors caused by the second GAC are comparatively very small. If the map of Fig. 7(b) had the same gray level to error correspondence as Fig. 7(a), it would appear to be entirely black. Although it is less crucial, we can still try to make a good choice of parameters for the 2nd GAC. From Fig. 7(b) we could choose a moderate curvature of 0.5 and high advection of 8.

We did not vary the weights of the last CLS. If the vertebral interior is well segmented the task of last CLS is greatly facilitated. All parameters are listed in Table I. For all level sets, the root mean square [(rms) distance between two consecutive fronts] stopping condition is set low at 0.001. It is mainly the number of iterations that is the controlling parameter. We found those numbers experimentally. They are suitable for our high resolution scans but could have to be tuned for differing scanning conditions. This can be a limitation to the automation of the method.

C. Segmentation Results

To segment real vertebrae we have to take into account the very different resolutions of our scans. The smallest x and y pixel size was 0.58 mm and the largest 0.94 mm. In the z direction the pixel size was either 0.7 or 0.625 mm. For higher resolution images ($px < 0.7$ mm) we increase the number of FM iterations from 20 to 30.

Using the parameters selected in Section II-B4 we used the segmentation algorithm on 50 real vertebrae, five from each of 10 patients (from T11 to L3). We visually inspected the segmentation results in both the sagittal and coronal views. We found only five noticeable errors, which means a 90% success rate. A typical mistake was leaking in the intervertebral disk space (IDS), which, when too thin, can disappear at half scale. This happened only for T11 vertebrae. We were able to correct all errors by tweaking the sigmoid parameters as explained in Section II-B3. Fig. 8 shows an example of a multiscale segmentation: the half scale segmentation result (a), which is then super-sampled back to full scale (b), and refined and postprocessed (c). We mentioned that the present segmentation algorithm is the multiscale version of a previous one. Fig. 9 shows an example where the new method managed to overcome a boundary discontinuity but not the old one.

D. Discussion

1) Seed Placement—In our algorithm seed placement is manual. We investigated the robustness of the algorithm in respect to this placement. Inside a $20 \times 20 \times 16$ (voxel) box around the initial user selected seed location, we place alternative seed locations in a regular grid-like manner. Fig. 10 shows the extent of the box (red) in relation to a vertebral body. The box is subdivided into $4 \times 4 \times 4$ boxes and the vertices of the small boxes constitute the new seed locations. There are 180 seed locations in total. The segmentations obtained with the new seed locations are compared with the reference segmentations. Those are the ones obtained with user selected seeds in Section II-C. They were visually inspected and were chosen to be used for the rest of our work. The comparison measure is the overlap similarity index (OSI) defined in the following way [23]:

$$OSI = \frac{2(V_1 \cap V_2)}{V_1 + V_2} \quad (5)$$

where V_1 and V_2 are the two volumes compared. OSI is always comprised between 0 and 1. The closer to 1 the better the similarity between the two volumes.

To summarize the results in a compact form we plot the obtained OSIs as a function of the distance to the original seed. For each seed the distance to the original seed is computed in mm and rounded to the nearest integer. Results of seeds with the same integer distances are averaged. Standard deviation is also computed and shown as error bars. This seed variation was carried out for each of our 50 vertebrae. In total $50 \times 180 = 9000$ segmentations were performed. Fig. 11(a) shows the average OSI for the 50 vertebrae as a function of the distance to the original seed.

The OSI is 0.99 at 2 mm but drops to 0.96 at 3 mm, drop which is also accompanied by a large standard deviation. After examination of the cases where the OSI had dropped substantially at 3 mm we found that it was because the seed was trapped in an area of high gradients and consequently very low speed. The level set did not expand at all during the initial FM stage. Such cases were rare but they reduced the average OSI and increased the standard deviation.

To overcome this problem of a seed finding itself in an area of null speed we improved the algorithm by using multiple seeds instead of just one. In addition to the first seed we place eight other seeds around it. They are the vertices of a $2 \times 2 \times 2$ square centered on the first seed. To compensate for the increased number of seeds, iterations in the FM stage are reduced by 5 to avoid leakage. The purpose of the FM stage is only to produce a seed region for the first GAC. As long as there is no leakage the extent of this seed region is unimportant. It is the first GAC that ensures that interior of the vertebral body is segmented completely.

Using this new seed scheme we carried out the seed variation experiment again [Fig. 11(b)]. This time the average OSI did not drop below 0.96 before 6 mm. The standard deviations in the low distances are dramatically reduced. The reference segmentations are the same as the ones used for the single seed experiment, which explains that the OSI is not exactly 1 at distance 0 (it is 0.997 ± 0.0025).

Because of the relatively large volume of our vertebrae an OSI below 0.95 is a good indicator of failed segmentation. Conversely an OSI above 0.98 usually indicates a good segmentation. In the cases we inspected the differences were mainly due to the varying quantities of bone segmented in the pedicles, which would not affect our work because we are primarily interested in the vertebral body. Between 0.95 and 0.98 failure and success seemed to be equally likely. Setting 0.98 as a conservative limit and using the multiseed scheme, the seed displacement tolerance is about 3 mm from the user placed seed.

2) Sigmoid Parameters—As stated in Section II-B3 the preprocessing sigmoid parameters K_1 and K_2 are important in determining the final quality of the segmentation. Our method for evaluating those was described above. K_2 is the mean gradient magnitude inside a box around the user selected seed. K_1 is the mean of the 10% lowest gradient magnitudes recorded by the search algorithm on the boundary of the vertebral body. We now investigate how the performance of the algorithm is affected by using the alternative percentages: 5%, 15%, 20%, 25%, 30%, 35%, and 40% for the first sigmoid parameter K_1 . As for our previous study about seed displacement we compute the OSI between reference segmentation (where 10% is used) and the new ones obtained with the alternative percentages. How sensitive the segmentation results are to those variations depends in fact on the strength of the boundary of each individual vertebra. The cortical bone gap shown in Fig. 3(b) is the largest boundary discontinuity our algorithm had to deal with and is common to all vertebrae. It is the entrance of the basivertebral vein. However the size and appearance of the gap is very variable. We made the experiments with two cases, one vertebra with a large basivertebral vein gap and one with a small one. They can be seen on the top row of Fig. 12.

The resulting OSIs are shown in Fig. 13. It can be seen that for the vertebra with the small gap (solid blue curve) the algorithm is very robust to variations in the percentage of gradient magnitudes used to estimate K_1 . Segmentations did not vary significantly for the whole range of percentages. The smallest OSI was 0.998. However for the vertebra with the large gap (broken red curve) the OSI fell at 0.88 at 20%. The bottom row of Fig. 12 shows both segmentations at 20%. Leakage occurred at the large gap but not the small one. For both

vertebrae using 5% instead of 10% yielded good results. For vertebrae with weak boundaries the algorithm can be fairly sensitive to the percentage of gradient magnitudes used for estimating K_1 . That is because as the percentage, K_1 itself increases causing the sigmoid to attenuate more and more edges until the weak boundaries disappear from the speed function.

III. RIDGELINE DETECTION

Syndesmophytes usually start from the ridge of the vertebral body and grow into the IDS, as shown in Fig. 1. Therefore, after the segmentation of the whole vertebral body, detection of its ridgeline is the next crucial stage. Taking advantage of our previous vertebral body segmentation we carry out ridgeline detection on its isosurface, which we obtained in the form of a triangular mesh using the marching cubes algorithm [24]. In our previous work [5], we had a surface growing algorithm on the end plate that would stop at the ridge. The ridge, defined using thresholds on some curvature measures computed on the segmented volume in a Cartesian grid, was often discontinuous, noisy, and very sensitive to the values used as thresholds. The consequences were that the extracted ridgelines, defined as the contour of the last surface resulting from the growing algorithm, often suffered from over- or under-segmentation. To overcome such problems we undertook to implement a level set, in fact a GAC, on the mesh surface representation of our vertebrae. Level sets are very robust to noise and provide closed continuous contours as end results, which are desirable properties for our purposes. However level set methods are usually defined in Cartesian grids. In this section we describe how to extend the application of level sets from Cartesian grids to the more challenging domain of triangular meshes.

A. Background on Ridgeline/Crestline Detection and Level Sets on Triangulated Domains

Ridgelines/crestlines are important features of 3-D objects. They often constitute essential anatomical landmarks such as the sulcal fundi in the cerebral cortex [25] or the acetabular rim in the hip joint [26], [27]. Those landmarks have many uses. The most common are shape analysis [28]–[32], surface averaging [33] and landmark-based registration [34]–[36]. In computer graphics they have been used for surface segmentation and flattening [37], surface visualization enhancement [38], [39] and mesh simplification [40]. It should be noted their uses are not limited to those mentioned above: each application generates its own needs. For instance in hip surgery the acetabular rim is important because it is the landmark upon which certain angles used by surgeons are defined [41].

The Thirion *et al.*[42], [43] and Monga *et al.*[44] algorithms for ridgeline/crestline detection operates in the 3-D Cartesian volume without prior segmentation. For 3-D surface meshes several methods have been developed. The Stylianou *et al.* method works by growing crest vertex regions and then skeletonizing them [37]. The Yoshizawa *et al.* [40] and Ohtake *et al.* [45] algorithms are based on finding crest/ridge vertices as points where one of the principal curvatures is locally extremal and connecting those points. An additional step is then required to distinguish inessential lines from salient ones. Alternative methods for detecting ridgelines/crestlines on meshes include dynamic programming [46] and snakes [47].

For our application using level sets to extract ridgelines has several advantages. It guarantees that the final curve will be closed and continuous avoiding the fragmentation that plagues some of the previous methods. On complex surfaces noise can be important and parts of a ridgeline can have reduced saliency (equivalent of gaps in gray level boundaries in Cartesian spaces). The smoothness constraint in level sets addresses such problems. Moreover we are implementing a full GAC. We will show that the additional advection term also plays an important part here to make the algorithm robust to gaps.

Level sets on triangulated domains were pioneered by Barth *et al.* [48]. However their focus was on adaptive mesh refinement. Most of the experimental data presented in [48] is on planar meshes or simple noiseless synthetic 3-D surfaces. Xu *et al.* [49] also focused on planar mesh refinement. Their method for computing gradients on a mesh is different from Barth *et al.* and is arguably more intuitive. We extended this method to a nonplanar manifold and also to the evaluation of curvature. This extension of Xu *et al.* gradient computation method is different from Barth *et al.* curvature computation method. Barth *et al.* presented work on a level set with only a propagation term and on a curvature flow. Xu *et al.* only considered a curvature term. Such level sets can be susceptible of leakage in the presence of inhomogeneities and gaps. In this section, we provide a transposition of a full GAC to the domain of nonplanar manifolds.

B. Methods

1) Speed Function Computation—In a usual level set the speed function is based on gray level features such as the sigmoid remapped gradient magnitudes mentioned earlier. On a 3-D surface the important feature is curvature. As a measure of mean curvature at each vertex we use Curvedness (CV) defined as [50]

$$CV = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \quad (6)$$

where κ_1 and κ_2 are the principal curvatures. To compute those principal curvatures on the mesh, a patch fitting method is used [51], [52]. The method essentially fits a quadric f to a neighborhood around each vertex

$$f(x, y) = ax^2 + bxy + cy^2 + dx + ey. \quad (7)$$

The coefficients a , b , c , d , and e of the quadric are evaluated using a least square method and then used to compute the principal curvatures [51], [52].

To obtain the speed function, the curvature function (curvedness at each vertex) is remapped by the sigmoid of (3) and (4) so that the speed function is close to 1 on plane portions of a surface and close to 0 on curved ones. The values of K_1 and K_2 , the sigmoid parameters of (4), could also be determined with a search algorithm. However after extensive experimentation we found that a value of 0.2 for K_1 and 0.1 for K_2 worked for most vertebrae. As for the case of vertebral body segmentation we tweak those parameters in case of failure. We increase or decrease K_1 to correct respectively under and over-segmentation of the ridgeline. Fig. 14 shows some examples of speed functions. Vertices with low speed are marked in black. As can be seen the ridgelines have low speed but gaps exist as well as noise on the vertebral end plates.

2) Gradient Computation—It can be seen from (1) that we need to compute the gradient of the distance and speed functions. The key to computing gradients and curvature accurately on a nonplanar manifold is to do it in local coordinate systems. If we consider a neighborhood small enough we can hope that it can reasonably be considered plane. Second, because on a mesh the points are no longer disposed along a regular orthogonal grid, finite difference methods for computing derivatives need to be replaced by least square estimates.

In the following sections it is important to distinguish the global orthonormal frame (GOF) associated with the scanner and in which the coordinates of the vertices are initially defined from the local orthonormal frame (LOF) associated with each individual vertex. Coordinates

in the GOF and LOF are respectively denoted with upper (X, Y, Z) and lower (x, y, z) case subscripts. The meaning of superscripts will be explicated in each occurrence.

We can form the LOF ($\vec{u}^x, \vec{u}^y, \vec{u}^z$) at each vertex V by taking V as the origin and the normal to the surface at V as the z axis. Let us note $\vec{u}^z = (u_x^z, u_y^z, u_z^z)^T$ the z vector of the LOF (T indicates transposition). It is obtained by averaging the normals of all faces incident on V and normalizing the resulting vector. The x and y vectors can be chosen arbitrarily as long as $(\vec{u}^x, \vec{u}^y, \vec{u}^z)$ form an orthonormal system. For instance we can take

$$\begin{aligned} u_x^x &= -\frac{u_y^z}{\sqrt{(u_x^z)^2 + (u_y^z)^2}} \\ u_y^x &= \frac{u_x^z}{\sqrt{(u_x^z)^2 + (u_y^z)^2}} \\ u_z^x &= 0 \end{aligned} \quad (8)$$

and

$$\vec{u}^y = \vec{u}^z \times \vec{u}^x. \quad (9)$$

which, explicated in terms of the components of $\vec{u}^z$ gives

$$u_x^y = -\frac{u_z^z u_x^z}{\sqrt{(u_x^z)^2 + (u_y^z)^2}} \quad (9a)$$

$$u_y^y = -\frac{u_z^z u_y^z}{\sqrt{(u_x^z)^2 + (u_y^z)^2}} \quad (9b)$$

$$u_z^y = \sqrt{(u_x^z)^2 + (u_y^z)^2} \quad (9c)$$

This choice of basis vectors is valid as long as either u_x^z or u_y^z is not zero. If they both are zero, there is no need for rotation.

The rotation matrix from global to local orthonormal frames can then be written

$$R_{\text{GOF} \rightarrow \text{LOF}} = \begin{pmatrix} u_x^x & u_y^x & u_z^x \\ u_x^y & u_y^y & u_z^y \\ u_x^z & u_y^z & u_z^z \end{pmatrix}. \quad (10)$$

Conversely the rotation matrix from local to global orthonormal frames is the transpose of $R_{\text{GOF} \rightarrow \text{LOF}}$. Note that to transform global coordinates to local coordinates a translation bringing V as the origin is performed before the rotation. Therefore, if (X, Y, Z) and (x, y, z) are respectively the coordinates of a point in the GOF and the LOF defined at V , the complete transformation equations from global to local coordinates are

$$\begin{aligned} x &= u_x^x(X - V_x) + u_y^x(Y - V_y) + u_z^x(Z - V_z) \\ y &= u_x^y(X - V_x) + u_y^y(Y - V_y) + u_z^y(Z - V_z) \\ z &= u_x^z(X - V_x) + u_y^z(Y - V_y) + u_z^z(Z - V_z). \end{aligned} \quad (11)$$

We note $\psi(V)$ the distance function at vertex V and V^i the i th immediate neighbor of V . We can now provide a step by step breakdown of the gradient computation method for the distance function.

1. For each vertex V where the distance function is defined we determine a LOF taking V as the origin and the normal to the surface at V as the z axis and using (8) and (9) for the x and y axes.
2. We make a list of the N immediate neighbors V^i of V and determine their coordinates in the LOF using (11). In our planar approximation we set all z components to 0. Fig. 15 shows a vertex with its N immediate neighbors in N nonorthogonal directions.
3. Although we cannot take finite differences in two orthogonal directions we can take the finite differences in the N nonorthogonal directions. We note $\nabla\psi(V)^i$ the finite difference of the distance function at vertex V in the direction of its neighbor V^i . It can be computed using

$$\nabla\psi(V)^i = \frac{\psi(V^i) - \psi(V)}{\|VV^i\|}. \quad (12)$$

4. We consider that $\nabla\psi(V)^i$ is the projection of gradient vector $\nabla\psi(V)$ (that we are looking for) on the unit directional vector $\vec{u}^i$ pointing from V to V^i . To find the x and y components of $\nabla\psi(V)$ in the LOF we minimize the error function E

$$E = \sum_{i=1}^N \left(\nabla\psi(V) \cdot \vec{u}^i - \nabla\psi(V)^i \right)^2. \quad (13)$$

5. We transform the components of $\nabla\psi(V)$ back to the GOF and record them. This last step will be used later for the curvature computation of the distance function.

For the gradient of the speed function the procedure is identical. In the steps above we just replace the distance function $\psi(V)$ by the speed function $g(V)$. Because the gradient of $g(V)$ needs to be computed only once it is not time consuming to do it on all vertices instead of just the narrow band. This calculation can be performed before the level set evolution. Fig. 16 show some gradients of the speed function at the ridge of a vertebral body. The black dots represent vertices with low speed values. The gray lines represent the gradient vectors of the speed function at each of those vertices. Fig. 16 permits the visualization of the opposing fields of force at each side of the ridge. Those are typical of the advection force introduced by Caselles *et al.* [20]. The forces on the side of the end plate attract the contour toward the ridge. The forces outside the end plate prevent the contour from going further. Those two opposite forces lock the contour on the ridge and make the GAC robust to inhomogeneities and gaps.

3) Curvature Computation—To compute the curvature of the distance function we use the usual equation

$$\kappa = \frac{\psi_{xx}\psi_y^2 - 2\psi_x\psi_y\psi_{xy} + \psi_{yy}\psi_x^2}{(\psi_x^2 + \psi_y^2)^{\frac{3}{2}}}. \quad (14)$$

However in our case the derivatives have to be performed in the LOF of each vertex. From the procedure above we already have $\nabla\psi(V)$ that we recorded in the GOF for each vertex V (Section III-B2, step 5). The first step here is to derive the local components $\psi_x(V)$ and

$\psi_y(V)$ from the vector $\nabla\psi(V)$ using $R_{\text{GOF} \rightarrow \text{LOF}}$. Then, in order to compute ψ_{xx} , ψ_{xy} , and ψ_{yy} , we can repeat the same gradient computing procedure (Section III-B2) replacing $\psi(V)$ by the new functions $\psi_x(V)$ and $\psi_y(V)$ that simply contain the values of ψ_x and ψ_y at each vertex V . A detailed breakdown of our curvature computation algorithm is given below.

1. For each vertex V where the distance function is defined we determine a LOF system taking V as the origin and the normal to the surface at V as the z axis.
2. We make a list of the N immediate neighbors V^i of V and determine their coordinates in the LOF.
3. We transform the gradients $\nabla\psi(V)$ and $\nabla\psi(V^i)$ from GOF to LOF using $R_{\text{GOF} \rightarrow \text{LOF}}$.
4. We write the finite differences in the N nonorthogonal directions

$$\nabla\psi_x(V)^i = \frac{\psi_x(V^i) - \psi_x(V)}{\|VV^i\|} \quad (15a)$$

$$\nabla\psi_y(V)^i = \frac{\psi_y(V^i) - \psi_y(V)}{\|VV^i\|}. \quad (15b)$$

5. We minimize the two functions

$$E_x = \sum_{i=1}^N \left(\nabla\psi_x(V) \cdot \vec{u}^i - \nabla\psi_x(V)^i \right)^2 \quad (16a)$$

$$E_y = \sum_{i=1}^N \left(\nabla\psi_y(V) \cdot \vec{u}^i - \nabla\psi_y(V)^i \right)^2. \quad (16b)$$

In the LOF the two components of $\nabla\psi_x(V)$ and $\nabla\psi_y(V)$ are respectively ψ_{xx} , ψ_{xy} , ψ_{yx} , and ψ_{yy} . These are our four unknowns. The vector $\vec{u}^i$ is the unit vector pointing from V to V^i .

6. We compute the curvature κ using (14). On an usual orthogonal grid we would have $\psi_{xy} = \psi_{yx}$. This is not the case here. In (14) we replace ψ_{xy} by the mean of ψ_{yx} and ψ_{xy} .

Fig. 17 shows a test we devised to verify that the thus computed curvature had the basic properties required to be a smoothing in-influence on the evolving contour. Fig. 17(a) shows the initial contour which is a circle augmented by a rectangle. This contour is composed of vertices (yellow) belonging to the mesh representation of a 3-D sphere. Curvature is computed on the distance function defined around this initial contour. In Fig. 17(b) the points with the highest curvatures in absolute value are marked in blue. As expected the corners of the rectangles have high curvatures. In Fig. 17(c) the points with negative curvatures are marked in black. As expected concave portions of the curve have negative curvatures. Those two properties combine to create a smoothing field of forces on the contour. Fig. 17(d) shows the updated zero-level contour (in blue) after the first iteration. As can be seen the parts of the contour that moved most are the most curved ones (where the rectangle joins the circle). The new contour is smoother than the original one.

4) Complete Algorithm—We now provide a complete step by step breakdown of our level set algorithm. Detail about distancing and other procedures will be given below.

Initialization

1. compute the gradient of the speed function;
2. put the initial contour on the mesh (in our case it is a simple circle);
3. do the initial distancing.

Level set loop: While the number of iterations and the rms difference between the new and old front have not reached some chosen thresholds:

1. blur the distance function;
2. compute the gradient of the distance function;
3. compute the curvature of the distance function;
4. update the distance function using a Runge–Kutta procedure [53];
5. find the new zero-level front;
6. compute the rms difference between the new front and the old one;
7. compute the distance function.

The distancing is performed inside a narrow band (NB). The list of vertices belonging to the NB is obtained in the following way: for every vertex on the zero-level contour (ZLC) we consider a fourth-degree neighborhood. The first-degree neighborhood of a vertex is composed of all its immediate neighbors linked to it by an edge. To obtain the second-degree neighborhood we add all the first-degree neighborhoods of the vertices in the first-degree neighborhood. We iterate the process until the fourth degree neighborhood. All the vertices in this neighborhood that have not yet been added to the NB list are added. Then for every vertex V in the NB we compute the nearest distance to the ZLC. We again consider a fourth-degree neighborhood around V . If a vertex V_0 in this neighborhood belongs to the ZLC we compute the Euclidian distance $\|VV_0\|$. We keep the smallest distance found. Because the narrow band is small enough we found the Euclidian distance could be used with satisfactory results. The need for a more accurate geodesic distance did not arise.

For the initial distancing, determining the sign of the distance is done in the following way: for each vertex V in the NB we look for its nearest vertex V_0 in the ZLC. We respectively note V_+ and V_- the nearest neighbors of V_0 in the ZLC in the counter-clockwise and clockwise directions. If the vector product $(V_- - V) \times (V_+ - V)$ is positive, V is inside the contour and the distance is negative. Otherwise the distance is positive. Note that such an algorithm is possible only because the initial contour is a simple circle of known center.

For subsequent distancing the signing algorithm is different. The distancing comes after updating of the distance function and the determination of the new ZLC. Signing needs to be done only for vertices that have been newly added to the NB. Old vertices already have a sign (that might have changed after updating). For a newly added vertex we determine a path on the mesh from it to its nearest neighbor on the ZLC. If on this path we meet more vertices with positive than negative distance, the vertex receives a positive sign. Otherwise the sign is negative.

Blurring of the distance function is performed by averaging the distances in a first degree neighborhood around each vertex.

We update the distance function using a first-order Runge–Kutta algorithm, which is a discretization of (1). For the propagation, curvature and advection weights we respectively use the values of 1, 1.5, and 2. We use 0.8 for the timestep.

To find the new ZLC after updating we examine every pair of vertices in the NB. If one has a positive distance and the other a negative one we set to 0 the one with the smallest absolute value.

To compute the rms between the old and new ZLC we start by taking a small neighborhood around each vertex of the old ZLC. In that neighborhood we look for the nearest new ZLC vertex and record its distance. If none is found in the neighborhood an arbitrary large distance of 5 mm is recorded. The mean of the distances form the rms. The maximum number of iterations was 80 and the threshold rms 0.0075. All the parameters of the GAC are gathered in Table II.

Our algorithm was implemented using C++ and the mesh format OFF (object file format). Any mesh format where neighbor connectivity information is available for each vertex should be suitable.

C. Experimental Results

We tested our algorithm on synthetic cylinders whose surfaces we perturbed with noise as shown in Fig. 18. Those cylinders were obtained in the following way. First we generate a noiseless volume cylinder of 30 pixel radius and 35 pixel height (pixel size are respectively 0.74, 0.74, 0.625 mm in the x , y , z directions). The ground truth ridgeline can easily be extracted as the highest of all circles forming the cylinder. The mesh representing the surface is then extracted using the marching cubes algorithm. This mesh is deformed. Every vertex is visited and modified once only. We go through the list of all vertices belonging to the mesh. If a vertex has not yet been modified it is taken as the center of a fourth-degree neighborhood that is moved in the direction of the center's normal. The sign of the direction (inwards or outwards) is random. The neighborhood is moved to assume a Gaussian shape with a standard deviation of 1.5 mm. The maximum displacement (at the center) is either 1 mm [Fig. 18(a)] or 1.5 mm [Fig. 18(b)]. Note that if the neighborhood contains vertices that were previously moved they are not moved again. The mesh is then smoothed with a Laplacian using a first degree neighborhood and a maximum deformation value of 5 mm. In Fig. 18 the ground truth ridgelines are shown in black, the resulting contour of our level set in yellow. We quantified the rms between the two contours in the same way as we did the rms between old and new ZLC in the subsection above, ground truth replacing old ZLC. For the cylinder on the left [Fig. 18(a)] rms was 0.16 mm and for the one on the right [Fig. 18(b)] 0.4 mm.

We then applied the method to real vertebrae. In the following section we extract the syndesmophyte volumes from four intervertebral disk spaces (T11-T12, T12-L1, L1-L2 and L2-L3) from each of 10 patients. Each intervertebral disk space (IDS) requires the segmentation of two ridgelines, one for the upper vertebra and one for the lower one. In total we therefore extracted 80 ridgelines using the parameters in Table II. After visual inspection only two segmentations had to be redone. Both errors were leakages on the upper end plate near the junction with the pedicles where the curvature is the least pronounced. Both errors were corrected by incrementally lowering the value of K_1 by 0.1 until the boundary was strong enough to stop the level set. The initial seed contour was a 10-mm radius circle. To determine the center of this circle we make use again of the user-provided point roughly in the center of the vertebral body (if using a multiple seed point strategy for the vertebral body segmentation, this point is the same as the central seed). From this point we shoot a vertical line. The points where this line meets the surface mesh are taken as the centers of the seed

circles. Fig. 19 shows an example of contour evolution on the upper end plate of a vertebra. Fig. 20 shows some other examples of successful ridgeline segmentation in yellow. In black are the vertices with low values in the speed function.

To justify that a full GAC was necessary we carried out some tests with the advection weight set to zero, which basically makes the GAC equivalent to the CLS. Gaps are usually to be found near the junction between the vertebral body and its pedicles. Fig. 21 shows two cases where the CLS (top row) leaked through the gap but not the GAC (bottom row).

D. Discussion (Sigmoid Parameters)

As for the level set segmentation of vertebrae (Section II), obtaining suitable parameters for the speed function sigmoid is also important here. As stated in Section III-B-1 we found through experimentation that values of 0.2 for K_1 and 0.1 for K_2 worked well in the present situation. The range of possible values for K_1 and K_2 is constrained by the geometry of the vertebrae. K_1 and K_2 are related to CV values on the end plate and the rim. To investigate why 0.2 and 0.1 worked well we looked at the distributions of CV values on the end plates and rims of our vertebrae. Our level set method described above permits to record vertices belonging to the end plates. As the contour expands we accumulate the new vertices reached by it (their distances change from positive to negative). The vertices of the last contour, which is taken as the rim, are also recorded. Fig. 22(a) and (b) respectively show CV values histograms for our 80 end plates and rims.

It can be seen that CV values peak at around 0.1 and 0.3–0.35 for the end plates and ridgelines respectively. In analogy with the Cartesian volume level sets, that is, replacing gradient magnitudes by CV values, K_2 should be an estimate of the mean CV on the end plate. 0.1 being the frequency peak of the end plate histogram is therefore a good estimate for K_2 . Meanwhile K_1 should be larger than K_2 and should represent an estimate of the weakest CV on the rim. As can be seen from Fig. 22(b), 0.2 is roughly located at the start of the rim CV histogram. The histogram curve suggests that a value between 0.15 and 0.25 could be suitable for K_1 .

We investigated the consequences of varying K_1 between 0.15 and 0.3 by intervals of 0.05. We performed the ridgeline detection with the alternative K_1 values and compared the results with reference ridgelines. Those are the ridgelines obtained in Section III-C, that is, with the parameters in Table II except for a few exceptions. Those reference ridgelines are the ones used in the rest of presented work. We compute the rms and maximum distance (MD) in mm between ridgelines in the following way. For each vertex on a new ridgeline we look for the closest corresponding vertex on the reference ridgeline through an exhaustive search. The MD is the maximum of all recorded minimum distances and the rms is their mean. This is done for the 80 ridgelines and the average rms and MD out of 80 is shown in Fig. 23.

Although rms and MD follow the same trend, MD provides a more sensitive measure. When 0.15 is used for K_1 , under-segmentation occurs as the level set contour is blocked by internal ridges not attenuated by the sigmoid. It can be seen in Fig. 22(a) that although the end plate CV values distribution peaks at 0.1, the number of vertices with a CV around 0.15 is far from negligible. Fig. 24(a) shows an example of under-segmentation occurring at an inferior end plate. Conversely when K_1 is set too high, leakages will occur as weak portions of the correct ridgeline are attenuated in the speed function. Fig. 24(b) shows an example of leakage occurring for K_1 set at 0.25. How sensitive the algorithm is to sigmoid parameter variation mainly depends on the geometry of each vertebral surface. The larger the difference between CV values on the actual ridgeline and CV values on internal ridges, the more robust the level set will be.

IV. SEGMENTATION OF THE SYNDESMOPHYTES

The level set described in Section III allowed us to find the ridgeline of the vertebral end plate. We also used it to make a list of vertices belonging to the end plate, as explained in Section III-D. We fit a plane through those points using a least square error method, which allows us to determine the normal to each end plate. Then at each vertex of the ridgeline we define a local plane centered around the vertex and orthogonal to the end plate's normal. Fig. 25(a) shows how a syndesmophyte can be separated from the end plate by a local plane. At this stage of the algorithm we consider the parts of the vertebra above all the local planes of the superior end plate (or below the local planes of the inferior end plate) as syndesmophytes. Fig. 25(b) shows the output of this stage.

It can be seen from Fig. 25(b) that the output of this simple plane cutting is noisy. Large parts of rim of the vertebra have been marked as syndesmophyte. The next step of our algorithm distinguishes real syndesmophytes from parts of the rim that are just slightly above the local planes. The main difficulty when using a threshold is to discard the noise without discarding the lower part of real syndesmophytes. For this reason we need to consider a neighborhood around each possible syndesmophyte point. In this neighborhood we take the maximum syndesmophyte height. This maximum height is compared to the local height of the end plate. To robustly find this local end plate height we use an iterative path growing on the mesh representing the surface of the vertebra. The start of the path is the approximate middle of the plate (which is also the center of the seed circle for the level set on mesh of Section III). The path is directed towards the candidate syndesmophyte point. The candidate syndesmophyte point and the middle of the end plate form a directing vector. The middle of the end plate is the first point of the path. In a neighborhood around it, the point that best matches the directing vector is taken as the next point in the path growing. We iterate this process considering the neighborhood of the last added point until we reach the contour of the end plate (determined in Section III). Fig. 25(b) shows an example of such a path. The height of the last point in the path is taken as the local end plate height. We compared this height with the maximum syndesmophyte height around the candidate syndesmophyte. Note that all heights are corrected by projection on the end plate's normal to take into account the orientation of each vertebral end plate. If the corrected height difference is above a threshold of 2 mm, the candidate syndesmophyte is classified as a genuine one. Fig. 25(c) and (d) shows the result of our processing of candidate syndesmophytes. Fig. 26 shows other examples of segmented syndesmophytes.

V. PERFORMANCE AND VALIDATION

A. Complexities and Times

In Table III we give the average durations of the three parts of our algorithm. The computer used had a processor speed of 2.8 GHz. Our algorithms were not optimized for speed. For the vertebral body segmentation (Part 1) the average is out of 50 vertebrae. For the ridgeline detection (Part 2) durations for 80 ridgelines were averaged. The mean syndesmophyte detection time (Part 3) was from 40 IDSs.

The longer duration for the level set on mesh compared to Cartesian volume level set mainly reflects the increased complexity of computing derivatives (gradient and curvature) using the least mean square algorithm rather than finite differences and the transformation from global to local coordinates at each vertex. Our algorithm could also be speeded up by updating the partial differential function using Runge–Kutta several times before recomputing the distance function. The third part of the method, a simple search algorithm, is relatively fast and mainly depends on the number of candidate syndesmophyte voxels.

B. Volume/Score Correlation

To validate our algorithm we quantitated the segmented syndesmophytes in terms of volume. Those results are compared with the scoring made by a medical expert from visual inspection of the CT scans. A 4-level scoring system was used. The scores from 0 to 3 measure the extension of the syndesmophytes in the IDSs. The meanings of the scores are:

- 0: no or dot-like syndesmophytes;
- 1: syndesmophytes extending less than 50% of the IDS;
- 2: syndesmophytes extending more than 50% of the IDS.
- 3: bridging (which occurs when the syndesmophytes continuously link the two end plates).

For this study, ten patients were examined, five of whom are in an early stage of the disease and the five others in a more advanced stage. Five vertebrae were segmented for each of them (from T11 to L3). Syndesmophyte volumes were extracted for the corresponding four IDSs. The same disk spaces were examined by the medical expert and scored.

Correlation analysis [Fig. 27(a)] showed that correlation between syndesmophyte volume and expert score is 0.802 ($p < 10^{-9}$). Most of the scores for the patients in an early stage of the disease were 0. The corresponding volumes were also 0 or very small. On the graph those points can only be shown as a concentration of points around (0,0). This concentration of points can constitute a bias for the correlation estimation. Therefore we carried out another correlation leaving out the IDSs which received an expert score of 0. The new regression line is shown in Fig. 27(b). Correlation was still 0.758 ($p < 3 \times 10^{-4}$).

However the dispersion of volume values for score 2 and 3 seem to indicate an important discrepancy between the algorithm's and the physician's evaluations. That is mainly because the score only takes into account the height of syndesmophytes in respect to the width of the IDS. For example in Fig. 25(c) it can be seen that the IDS has five syndesmophytes. If it had only one the score would be the same although the volume would change considerably. IDS widths also vary from vertebra to vertebra and from patient to patient. The visual score does not consider the volume of individual syndesmophytes.

C. Height/Score Correlation

To further validate our method we also provide a correlation between the scores and the heights of the syndesmophytes. We compare the height of each individual syndesmophyte to the local width of the IDS where it is located. To obtain an estimate of this local width we again make use of the ridgelines extracted in Section III. As shown in Fig. 28, we first locate the extremal point S of the syndesmophyte (yellow). We then locate the two points, R_1 and R_2 , on the upper and lower ridgelines that are closest to S . Those two points determine the local width of the IDS nearest to the extremal point of the syndesmophyte. We define the height of the syndesmophytes as a proportion of the IDS width using

$$h = (\vec{S} - \vec{R}_1) \cdot (\vec{R}_2 - \vec{R}_1) \quad (17)$$

where $\vec{R}_1$ belongs to the ridgeline closest to the centroid of the syndesmophyte. Most IDS have several syndesmophytes. We take the largest height as the measure we compare to the physician's score.

Fig. 29 shows the new regression lines. The new correlation values are respectively 0.936 ($p < 10^{-8}$) and 0.825 ($p < 3 \times 10^{-5}$) with and without the zero score. Correlation has improved because the new measure, syndesmophyte height, is closer to the physician's scoring

criteria. For score 0, all heights except one were also 0. For score 3, all heights were close to 1, the lowest being 0.98. From the definition of score 1, it is intuitively expected that heights should range between 0 and 0.5. In practice we found the range [0.12,0.43]. Score 2 had the largest dispersion. About half of the IDSs in that category had heights close to 1. That is because they had syndesmophytes that were is near bridging, like in Fig. 30. But because bridging was not complete the score was only 2. This highlights the problem of trying to correlate a continuous variable (height) and a discrete score.

D. Kruskal–Wallis Test

We also conducted a Kruskal–Wallis test on the volumes and heights of the syndesmophytes. Kruskal–Wallis is a nonparametric analysis of variance test. One major difference with the previous regression analysis is that here a linear relationship is no longer assumed. The test's objective is to evaluate how different the groups of volume/height values associated with the 4 four scores are. Table IV and Table V show the results of the test. Range is the minimum and maximum values inside a group. In Table IV all volumes are in mm^3 . It can be seen that the chi-squares for the volumes and heights are very similar. It was respectively 34.2 and 35.1 for the volumes and heights. Both correspond to $p < 0.0001$. That indicates a good probability that the sets corresponding to the different scores are different. The Kruskal–Wallis analysis was conducted using SAS version 9.3.

In Table V it can be seen that the median height for score 2 was 1. That is because of the problem highlighted in Section V-C. In four out of seven IDSs, near bridging syndesmophytes were present.

VI. CONCLUSION

We have devised a multistage algorithm with the aim of measuring syndesmophyte volumes for the evaluation of the disease Ankylosing Spondylitis. We have sought to make the algorithm as automated as possible. For the segmentation part we introduced a 3-D multiscale cascade of successive level sets that addresses the often conflicting requirements created by the complex structure of vertebral bodies. The algorithm was found to be robust. Out of 50 vertebrae we were able to segment 90% with the same set of parameters. We hope to extend the technique to the segmentation of other bones.

We also introduced a novel level set implementation with the aim of performing segmentation on the surface of an object represented by a triangular mesh using curvature features. That helped us extract the ridgelines of vertebral bodies. We implemented a full GAC and showed how the advection force prevented leakage on the weakest parts of a vertebra's ridgeline. Out of 80 ridgelines we only had to change the level set parameters for 2. Although we devised the method for our particular problem it is general and can be used in other applications. Ridgelines or crestlines are essential landmarks of 3-D surfaces.

We have also devised a reliable method to segment the syndesmophytes from the vertebral body using local cutting planes. Our algorithm was tested on 40 intervertebral disk spaces from 10 patients and validated by comparing its results with the scoring of a medical expert. Correlation between expert scores and syndesmophyte heights was 0.936 ($p < 10^{-18}$).

Acknowledgments

The authors would like to thank A. Huang for his implementation of the patch fitting method.

This work was supported by the Intramural Research Program of the National Institutes of Health, Clinical Center and the National Institute of Arthritis and Musculoskeletal and Skin diseases.

REFERENCES

1. Inman, RD. The spondyloarthropathies. In: Goldman, L.; Ausiello, D., editors. Cecil Textbook of Medicine. 22nd ed. Vol. ch. 279. Philadelphia, PA: Saunders; 2003. p. 1655-1657.
2. Arnett, FC. Ankylosing spondylitis. In: Koopman, WJ., editor. Arthritis and Allied Conditions. 13th ed. Baltimore, MD: Williams & Wilkins; 1997. p. 1197-1208.
3. MacKay K, Mack C, Brophy S, Calin A. The bath ankylosing spondylitis radiology index (BASRI). A new, validated approach to disease assessment. Arthritis Rheum. 1998; vol. 41:2263–2270. [PubMed: 9870884]
4. Ward MM. Outcomes in ankylosing spondylitis: What makes the assessment of treatment effects in outcomes in ankylosing spondylitis different? Ann. Rheum. Diseases. 2006; vol. 65(no.):iii25–iii28. [PubMed: 17038467]
5. Tan, S.; Yao, J.; Ward, MM.; Yao, L.; Summers, RM. Computer aided evaluation of ankylosing spondylitis; Proc. IEEE Int. Symp. Biomed. Imag; Arlington, VA. 2006. p. 339-342.
6. McInerney T, Terzopoulos D. Topology adaptive deformable surfaces for medical image volume segmentation. IEEE Trans. Med. Imag. 1999 Oct.vol. 18(no. 10):840–850.
7. Ghebreab S, Smeulders AWM. Combining strings and necklaces for interactive three-dimensional segmentation of spinal images using an integral deformable spine model. IEEE Trans. Biomed. Eng. 2004 Oct.vol. 51(no. 10):1821–1829. [PubMed: 15490829]
8. Kang Y, Engelke K, Kalender WA. A new accurate and precise 3-D segmentation method for skeletal structures in volumetric ct data. IEEE Trans. Med. Imag. 2003 May; vol. 22(no. 5):586–598.
9. Mastmeyer A, Engelke K, Fuchs C, Kalender WA. A hierarchical 3-D segmentation method and the definition of vertebral body coordinate systems for QCT of the lumbar spine. Med. Image Anal. 2006; vol. 10:560–577. [PubMed: 16828329]
10. Ibanez, L.; Schroeder, W.; Ng, L.; Cates, J. The ITK Software Guide. Clifton Park, NY: Kitware; 2003.
11. Lin N, Yu W, Duncan JS. Combinative multiscale framework for echocardiographic image segmentation. Proc. MICCAI. 2002; (no. 1):682–689.
12. Liang Q, Wendelhag I, Wikstrand J, Gustavsson T. A multiscale dynamic programming procedure for boundary detection in ultrasonic artery images. IEEE Trans. Med. Imag. 2000 Feb.vol. 19(no. 2):127–142.
13. Descoteaux M, Collins DL, Siddiqi K. Geometric flows for segmenting vasculature in MRI: Theory and validation. Proc. MICCAI. 2004; (no. 1):500–507.
14. Krissian, K.; Ellsmere, J.; Vosburgh, K.; Kikinis, R.; Westin, CF. Multiscale segmentation of the aorta in 3-D ultrasound images. 25th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.; 2003. p. 638-641.
15. Okada K, Comaniciu D, Krishnan A. Robust anisotropic gaussian fitting for colometric characterization of pulmonary nodules in multislice ct. IEEE Trans. Med. Imag. 2005 Mar.vol. 24(no. 3):409–423.
16. Joshi S, Pizer S, Fletcher PT, Yushkevich P, Thall A, Marron JS. Multiscale deformable model segmentation and statistical shape analysis using medial descriptions. IEEE Trans. Med. Imag. 2002 May; vol. 21(no. 5):538–550.
17. Sanchez-Ortiz GI, Wright GJT, Clarke N, Declerk J, Banning AP, Noble JA. Automated 3-D echardiography analysis compared with manual delineations and SPECT MUGA. IEEE Trans. Med. Imag. 2002 Sep.vol. 21(no. 9):1069–1076.
18. Sethian, JA. Level Set Methods. Cambridge, U.K: Cambridge Univ. Press; 1999.
19. Caselles V, Catte F, Coll T, Dibos F. A geometric model for active contours. Numer. Math. 1993; vol. 66:1–31.
20. Caselles V, Kimmel R, Sapiro G. Geodesic active contours. Int. J. Comput. Vis. 1997; vol. 22:61–79.
21. Tan, S.; Yao, J.; Ward, MM.; Yao, L.; Summers, RM. Level set based vertebra segmentation algorithm for the evaluation of ankylosing spondylitis; Proc. SPIE Med. Imag; San Diego, CA. 2006. p. 58-67.

22. Tan, S.; Yao, J.; Ward, MM.; Yao, L.; Summers, RM. 3-D multiscale level set segmentation of vertebrae. 2007 4th IEEE Int. Symp. Biomed. Imag.: Macro Nano; Arlington, VA: 2007. p. 896-899.
23. Dawant BM, Hartmann SL, Thirion JP, Maes F, Vandermeulen D, Demaerel P. Automatic 3-D segmentation of internal structures of the head in MR images using a combination of similarity and free-form transformations: Part 1, methodology and validation on normal subjects. *IEEE Trans. Med. Imag.* 1999 Oct.vol. 18(no. 10):909-916.
24. Lorensen W, Cline H. Marching cubes: A high resolution 3-D surface construction algorithm. *Proc. SIGGRAPH'98, Comput. Graph.* 1987; vol. 21:163-169.
25. Kao CY, Hofer M, Sapiro G, Stern J, Rehm K, Rottenberg DA. A geometric method for automatic extraction of sulcal fundi. *IEEE Trans. Med. Imag.* 2007 Apr.vol. 26(no. 4):530-540.
26. Reijman M, Hazes JMW, Pols HAP, Koes BW, Bierma-Zeinstra SMA. Acetabular dysplasia predicts incident osteoarthritis of the hip. *Arthritis Rheum.* 2005; vol. 52:787-793. [PubMed: 15751071]
27. Haddad FS, Garbuz DS, Duncan CP, Janzen DL, Munk PL. CT evaluation of periacetabular osteotomies. *J. Bone Joint Surg. Br.* 2000; vol. 82:526-531. [PubMed: 10855875]
28. Gueziec A, Ayache N. Smoothing and matching of 3-D space curves. *Int. J. Comput. Vis.* 1994; vol. 12:79-104.
29. Thirion JP. The extremal mesh and the understanding of 3-D surfaces. *Int. J. Comput. Vis.* 1996; vol. 19:115-128.
30. Monga O, Armande N, Montesinos P. Thin nets and crest lines: Application to satellite data and medical images. *Comput. Vis. Image Understand.* 1997 Sep.vol. 67(no. 3):285-295.
31. Subsol G, Thirion JP, Ayache N. A scheme for automatically building 3-D morphometric anatomical atlases: Application to a skull atlas. *Med. Image Anal.* 1998; vol. 2(no. 1):37-60. [PubMed: 10638852]
32. Little JJ, Shi P. Structural lines, tins, and dems. *Algorithmica.* 2001; vol. 30:243-263.
33. Subramanyan R, Dean D. A procedure to average 3-D anatomical structures. *Med. Image Anal.* 2000; vol. 4:317-334.
34. Audette MA, Ferrie FP, Peters TM. An algorithmic overview of surface registration techniques for medical imaging. *Med Image Anal.* 2000; vol. 4:201-217. [PubMed: 11145309]
35. Thirion, JP.; Monga, O.; Benayoun, S.; Gueziec, A.; Ayache, N. Automatic registration of 3-D images using surface curvature; *Proc. SPIE Med. Imag;* 1992. p. 206-216.
36. Maintz JBA, van den Elsen PA, Viergever MA. Comparison of edge-based and ridge-based registration of CT and MR brain images. *Med. Image Anal.* 1996 Jun.vol. 1:151-161. [PubMed: 9873926]
37. Stylianou G, Farin G. Crest lines for surface segmentation and flattening. *IEEE Trans. Vis. Comput. Graph.* 2004 Sep-Oct;vol. 10(no. 5):536-544. [PubMed: 15794136]
38. Interrante, V.; Fuchs, H.; Pizer, S. Enhancing transparent skin surfaces with ridge and valley lines; *Proc. IEEE Visualization;* Atlanta, GA. 1995. p. 52-59.
39. DeCarlo D, Finkelstein A, Rusinkiewicz S, Santella A. Suggestive contours for conveying shape. *ACM Trans. Graph.* 2003:848-855.
40. Yoshizawa, S.; Belayev, A.; Seidel, HP. Fast and robust detection of crest lines on meshes; *Proc. ACM Symp. Solid Phys. Model;* 2005. p. 227-232.
41. Armiger RS, Armand M, Lepisto J, Minhas D, Tallroth K, Mears SC, Waites MD, Taylor RH. Evaluation of a computerized measurement technique for joint alignment before and during periacetabular osteotomy. *Comput. Aided Surg.* 2007; vol. 12:215-224. [PubMed: 17786597]
42. Thirion J, Gourdon A. Computing the differential characteristics of isointensity surfaces. *Comput. Vision Image Understand.* 1995; vol. 61(no. 2):190-202.
43. Thirion JP, Gourdon A. The 3-D marching lines algorithm. *Graph. Models Image Process.* 1996; vol. 58(no. 6):503-509.
44. Monga O, Benayoun S. Using partial derivatives of 3-D images to extract typical surface features. *Comput. Vision Image Understand.* 1995; vol. 61:171-189.

45. Ohtake Y, Belayev A, Seidel HP. Ridge-Valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.* 2004; vol. 23:609–612.
46. Khaneja N, Miller MI, Grenander U. Dynamic programming generation of curves on brain surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* 1998 Nov.vol. 20(no. 11):1260–1265.
47. Lee Y, Lee S. Geometric snakes for triangular meshes. *Comput. Graph. Forum.* 2002; vol. 21:229–238.
48. Barth TJ, Sethian JA. Numerical schemes for the Hamilton-Jacobi and level set equations on triangulation domains. *J. Comput. Phys.* 1998; vol. 145:1–40.
49. Xu M, Thompson PM, Toga AW. An adaptive level set segmentation on a triangulated mesh. *IEEE Trans. Med. Imag.* 2004 Feb.vol. 23(no. 2):191–201.
50. Koenderink, JJ. *Solid Shape*. Cambridge, MA: MIT Press; 1990.
51. Huang, A.; Summers, RM.; Hara, AK. Surface curvature estimation for automatic colonic polyp detection; *Proc. SPIE Med. Imag*; San Diego, CA: 2005. p. 393-402.
52. Petitjean S. A survey of methods for recovering quadrics in triangle meshes. *ACM Comput. Surv.* 2002 Jun.vol. 34:211–262.
53. Press, WH.; Teukolsky, SA.; Vetterling, WT.; Flannery, BP. *Numerical Recipes in C*. Cambridge, U.K: Cambridge Univ. Press; 1992.

Fig. 1. Two views of syndesmophytes (red arrows) from the same vertebra: (a) 2-D and (b) 3-D views.

Fig. 2. Results of the three stages of our algorithm. (a) Segmentation of the vertebral body. (b) Detection of the end plate's ridgeline. (c) Segmentation of the syndesmophyte.

Fig. 3. Typical difficulties for vertebra segmentation. (a) Double boundary. (b) Discontinuity. (c) Inner boundary.

Original segmentation

Multiscale segmentation

Fig. 4.
Flowcharts of the original and multiscale algorithms.

Fig. 5.
Results of the search algorithm for estimating gradient magnitudes inside and at the boundary of the vertebral body.

Fig. 6.
Axial, sagittal, and coronal view of the semi-synthetic vertebra.

Fig. 7. Error as a function of curvature and advection weights for the parameters of (a) the first GAC and (b) the second GAC. The lighter the gray level the larger the error.

Fig. 8. Example of multiscale segmentation. (a) Result of half scale segmentation. (b) Half scale segmentation super-sampled back to full scale. (c) Final segmentation.

Fig. 9. Segmentation of a vertebra (left) by the multiscale (middle) and singlescale (right) algorithms.

Fig. 10.
Axial, sagittal and coronal view of the box (red) inside which alternative seeds are placed.

Fig. 11. Variation of the OSI with seed displacement relative to the user selected seed. (a) One seed scheme (b) Multiseed scheme.

Fig. 12. Two vertebral bodies (top row) and their respective segmentations (bottom row) when 20% of gradient magnitudes are used to estimate the first sigmoid parameter. Arrows indicate the boundary gap caused by the basivertebral vein.

Fig. 13.

Variation of the OSI with the percentage of gradient magnitudes used to estimate the first sigmoid parameter for the vertebral body on the left (red broken curve) and right (blue solid curve) of Fig. 12.

Fig. 14.
Speed function of two vertebrae with low values marked in black.

Fig. 15.
Example of a vertex with six immediate neighbors in six nonorthogonal directions.

Fig. 16.
Gradient of the speed function at the ridge of a vertebra.

Fig. 17.

Computation of the curvature of the level set function on a synthetic sphere. (a) Original zero level set contour. (b) Vertices with highest curvature in absolute value in blue. (c) Vertices with negative curvature in black. (d) New zero level set contour after one iteration in blue.

Fig. 18. Ridgeline detection on synthetic surfaces of maximum noise amplitude (a) 1 mm (b) 1.5 mm.

Fig. 19.
Example of contour evolution on a real vertebral end plate.

Fig. 20.
Examples of ridge line segmentations. Black dots are vertices with high curvedness.

Fig. 21.
Comparison between the segmentation results of the CLS (top) and GAC (bottom) in the presence of gaps.

Fig. 22.
Curvedness value distribution on the 80 (a) end plates (b) ridgelines.

Fig. 23. Variation of ridgeline rms (blue solid curve) and MD (red broken curve) with first sigmoid parameter. Average out of 80 ridgelines is shown.

Fig. 24.
Ridgeline segmentation errors due respectively to (a) too low (b) too high first sigmoid parameter.

Fig. 25. Syndesmophyte segmentation. (a) Local plane cutting a syndesmophyte. (b) Path (black) towards a possible syndesmophyte point (white). (c) Syndesmophytes after thresholding. (d) Two-dimensional view of syndesmophytes. The arrows point to corresponding syndesmophytes in (c) and (d).

Fig. 26.
Examples of segmented syndesmophytes in red.

Fig. 27.
Correlation between syndesmophyte volumes and expert scores (a) with all scores and (b) excluding 0 scores.

Fig. 28.
Determination of the height of a syndesmophyte relative to the local IDS width. Ridgelines are indicated in black.

Fig. 29. Correlation between syndesmophyte heights and expert scores (a) with all scores (b) excluding 0 scores.

Fig. 30.
Near bridging syndesmophytes. Height ratios are close to 1 while physician's score is 2.

TABLE I

Level Set Parameters For Vertebral Body Segmentation

	half scale			full scale
	1 st GAC	2 nd GAC	CLS	CLS
propagation	1	1	1	0.1
curvature	2	0.5	1	1
advection	8	8	N/A	N/A
iterations	200	100	200	250
rms	0.001	0.001	0.001	0.001

TABLE II

Level Set Parameters For Ridgeline Extraction

propagation	1	timestep	0.8
curvature	1.5	iterations	80
advection	2	RMS	0.0075

TABLE III

Mean Durations of the Three Parts of the Algorithm

Part1	Part2	Part3
55ms	138ms	17ms

TABLE IV

Kruskal–Wallis for the Volumes

Score	0	1	2	3
Frequency	22	6	7	5
Range	0–90	24–178	44–1123	414–2050
Median	0	70	280	1262
chi=34.2			p<0.0001	

TABLE V

Kruskal–Wallis for the Heights

Score	0	1	2	3
Frequency	22	6	7	5
Range	00.41	0.12–0.43	0.27–1.1	0.98–1.1
Median	0	0.325	1	1.1
chi=35.1			p<0.0001	