LJMU Research Online

Natalia, F, Meidia, H, Afriliana, N, Young, JC and Sudirman, S

 Contour evolution method for precise boundary delineation of medical images

http://researchonline.ljmu.ac.uk/id/eprint/12886/

Article

For more information please contact researchonline@ljmu.ac.uk

Contour evolution method for precise boundary delineation of medical images

Friska Natalia[1], Hira Meidia[2], Nunik Afriliana[3], Julio Christian Young[4], Sud Sudirman[5]
[1,2,3,4]Universitas Multimedia Nusantara, Indonesia
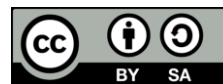[5]Liverpool John Moores University, United Kingdom

Article Info

ABSTRACT

Image segmentation is an important precursor to boundary delineation of medical images. One of the major challenges in applying automatic image segmentation in medical images is the imperfection in the imaging process which can result in inconsistent contrast and brightness levels, and low image sharpness and vanishing boundaries. Although recent advances in deep learning produce vast improvements in the quality of image segmentation, the accuracy of segmentation around object boundaries still requires improvement. We developed a new approach to contour evolution that is more intuitive but shares some common principles with the active contour model method. The method uses two concepts, namely the boundary grid and sparse boundary representation, as an implicit and explicit representation of the boundary points. We tested our method using lumbar spine MRI images of 515 patients. The experiment results show that our method performs up to 10.2 times faster and more flexible than the geodesic active contours method. Using BF-score contour-based metric, we show that our method improves the boundary accuracy from 74% to 84% as opposed to 63% by the latter method.

Corresponding Author:

Sud Sudirman,
Liverpool John Moores University
Liverpool L3 3AF, United Kingdom.
Email: s.sudirman@ljmu.ac.uk

1. INTRODUCTION

A basic problem in computer vision is to understand the high level of information contained in an image or a series of images. Many of the tasks in computer vision may look trivial when performed by a human. Tasks such as detecting and distinguishing faces and objects in images, or classifying hand gestures in videos are some of the examples of a problem that humans can do with relative ease but could be a challenge for a computer to perform. More complex tasks such as detection of anomalies in medical images pose an even greater challenge due to its high accuracy requirement and high-risk failure consequences.

Understanding medical images requires a specific set of skills and many years of experience working as a medical practitioner such as a radiologist. Thus, in building a computer-aided diagnosis (CAD) system, it is necessary to adopt the working procedure of the relevant radiologist and model it as an algorithm that computers can execute. This process is a complex task and may require an application of several image processing and computer vision techniques that as a whole complement each other. Image segmentation and boundary delineation are two important tasks in many modern image analysis processes including computer-assisted diagnosis through medical image analysis.

Image segmentation has been successfully applied in many CAD systems working on many different types of medical images. Traditionally, image segmentation is achieved using clustering methods, such as genetic algorithm [1] and fuzzy-means [2], of a number of manually designed low-level features such as pixel values distribution and histograms of gradients. Stochastic methods are also a popular approach in image segmentation. A regression segmentation framework to detect vascular anomalies in cardiac magnetic resonance imaging (MRI) images by delineating boundaries of bi-ventricle from was proposed [3]. In this framework, a regression model has been trained automatically on a deep belief network by using extracted DAISY feature [4] as input and using automatically generated boundary points as labels. The method was reported to yield high performance when tested on 2,900 images taken from 145 clinical subjects.

One of the major challenges in applying automatic image segmentation in medical images such as CT and MRI is the imperfection in the imaging process which can often result in inconsistent contrast and brightness levels, and low image sharpness and vanishing boundaries. Recent advances in deep learning produce vast improvements in the quality of image segmentation. As a result, research in this field has shifted its focus to using more contemporary techniques such as convolution neural networks [5, 6] and random forest [7]. In one of our earlier publications [5], we proposed a novel method for image segmentation and boundary delineation to assist diagnosis of lumbar spinal stenosis in MRI images using SegNet [8].

It is noted the accuracy of deep learning classifiers may also depend on the accuracy of the ground truth labels that are used to train the models. Producing label images manually is a laborious and lengthy process. It is very unlikely that medical experts' time is used to carry out this activity for hundreds or even thousands of images. Even if their time is used to label a few, they may not have the necessary skills in using digital applications to produce accurate segmentation. A practical solution to this is to have the medical experts to train several skilled image labellers on the important principles and what features to look for and assign the labellers to produce the ground truth label images for the classifier training. However, even with this approach, there is still a high level of probabilities of inaccuracies, especially along region boundaries [9].

To improve the segmentation accuracy along region boundaries, an active contour model can be applied to the label images. However, in our study of the important principles on image labeling provided by medical experts, we found that the existing active contour models have a number of drawbacks when are used to label medical images [10]. Thus, we develop a new approach to contour evolution that is more intuitive but shares some common principles with the active contour model method. Through analysis of our experimental results, we show that our method has several properties that help overcome the problems in the existing active contour models.

This paper starts by discussing the principle of active contour models which is followed by the description of the problems in applying these to medical images. The design of our solution is provided in section 3 followed by the analysis of the experimental results in section 4. We use the lumbar spine images that we have collected [11] and their associated label images [12] in our experiment. This dataset contains axial MRI scans of the last three intervertebral discs of 515 patients.

2. ACTIVE CONTOUR MODEL

Active contour model has been used to segment and delineate boundaries in different types of medical images such as echocardiographs [13], MRI [14], and Computed Tomography (CT) [15]. It is a popular class of image segmentation and boundary delineation method due to its ability to fit a curve to an object boundary by iteratively expanding or contracting its boundary estimate. The iterative process starts by defining an initial contour C_0 and at each iteration t, the contour C_t evolves from the previous iteration C_{t-1} such that $C_t = C_{t-1} + \frac{dc}{dt} \times \Delta t$, where $\frac{dc}{dt}$ is the contour evolution speed. The contour evolution speed is not a singular entity but rather a multi-value one as it is defined adaptively for each point s along the contour position at each iteration. This is illustrated in Figure 1.

The contour evolution speed, i.e., the red arrow illustrated in Figure 1, (which is often also referred to as *Force* denoted as F), is the movement of each point along the contour. The force has direction perpendicular to the tangent of the curve. The value of F depends on local properties such as curvature, global properties such as shape, and other independent properties [16]. One of the earliest and most popular active contour models is Snake [17]. It can arguably be considered as the basis of almost all subsequent active contour models to date. In this model, the force is calculated to minimize an energy function E that is a function of the curve's internal factors e.g., the first and second-order derivatives of the curve as well as external factors (e.g., image dependent features such as image gradient or edges). The value of E is the integral of three components expressed in the (1):

$$E = \alpha \int_0^1 |C'|^2 ds + \beta \int_0^1 |C''|^2 ds + \gamma \int_0^1 |\text{feature}(C)|^2 ds \qquad (1)$$

where $C' = \frac{dC}{ds}$, $C'' = \frac{d^2C}{ds^2}$, and $\text{feature}(C)$ is the image feature value calculated at the contour location. The model uses three parameters to control the effect of each component on the contour evolution. Parameters α and β control the contribution of the internal factors whereas γ controls the contribution of the external factors. Almost all of the subsequently proposed active contour models in the literature are based on this principle. Differences amongst them are mainly on the different ways of calculating and determining each of the three components and their parameters.
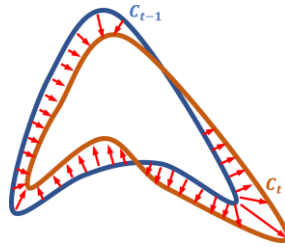

Figure 1. The evolution of the contour C_t (brown line) at time t from the previous contour C_{t-1} (blue line) with the red arrows marking the contour evolution speed at each position on the contour

The implementation of the Snake active contour algorithm uses the zero level set (ZLS) framework [18]. This framework propagates a shape boundary perpendicularly to the contour by means of an Eulerian formulation. Instead of representing the boundary and solve the evolution parametrically, ZLS represents 2D contours implicitly as a set of points of where the value of a function crosses zero (either from positive to negative, or vice versa). This function is an everywhere-differentiable image function $\phi: \Omega \to \Re$ with domain $\Omega \subset \Re^2$. The function ϕ is a special image function that is derived from a label image. Given a binary image denoting a closed set of labelled pixels with two classes namely A and $\neg A$, where A is the class of the object of interest in the image and $\neg A$ is its complement class, the function ϕ has a negative value at pixels that belong to A and a positive value at pixels which do not. Since ϕ is differentiable everywhere, its value should be zero, or close to zero, at boundary locations between A and $\neg A$ regions. The function is often illustrated in 3D with the x and y axes making up the image plane and the z for the value of the function. The intersection of the function with the $z = 0$ plane will create an outline of the boundary of the two regions in a segmented image. This set of points is called zero level set and is denoted as Γ. Contour evolution using the ZLS method is achieved by evolving the ϕ function, hence the state of the function at iteration t is denoted specifically as ϕ_t. Therefore, at any iteration t, the zero-level set $\Gamma_t = \{(x,y)|\phi_t = 0\}$ contains the intermediate solution to the boundary delineation problem. The process to construct ϕ from a binary label image is illustrated in Figure 2.

The initial value of ϕ, i.e., the state of the function at the start of the iteration, denoted as ϕ_0 can be calculated from the initial segmented image. The initial segmented image is a binary image with pixel values of 1 for every pixel that is inside the shape and 0 otherwise as shown in Figure 2 (a). One popular method to construct ϕ from a segmented image is by calculating the signed distance function (sdf) [19]. For each pixel, ϕ is calculated as the distance between that pixel to the closest point on Γ. This may sound like an egg-and-chicken problem with regards to ϕ and Γ, but in practice, the sdf can be calculated using the Euclidian distance transform (Edt) of the binary segmented image. More precisely, the sdf is calculated as the difference of the Euclidian distance transform of the binary segmented image and the Euclidian distance transform of the inverse of the binary segmented image, i.e.,

$$sdf = Edt(B) - Edt(\neg B) \qquad (2)$$

where B is the binary segmented image and $\neg B$ is its inverse. A distance transform is an operator that is applied to every pixel in its input binary image to calculate the distance between the pixel to the nearest non-zero pixel [20]. Figure 2 (b) and Figure 2 (c) show the corresponding sdf of the binary image in 2D and 3D, respectively.

Using ZLS, the evolution of the contour is achieved through an evolution of this signed distance function. The first sdf, ϕ_0 is required as the initial value to start with. The relationship between the new sdf and the contour evolution speed F is given as the level set equation [16]:

$$\phi_t = -F \times |\nabla \phi_{t-1}| \tag{3}$$

since its introduction, there are a number of variations of the ZLS framework. Some methods apply different energy minimization procedures whereas others use different image features when calculating F. One of the most popular is the edge-based Geodesic Active Contours (GAC) [21]. In this approach, the force F acting on the signed distance function is calculated as:

$$F = \alpha g_I + \beta \kappa + \gamma(\nabla g_I \vec{N}) \tag{4}$$

where g_I is the speed function component derived from the feature of the input image I. When a positive value of α is used, this component always acts outward to drive the contour to expand. The second component of the speed function, κ, is the mean curvature of the contour. The parameter β hence controls the smoothness of the evolving contour. The third component, $\nabla g_I \vec{N}$ is the advection force that draws the contour towards the nearest edges along the normal vector of the contour.
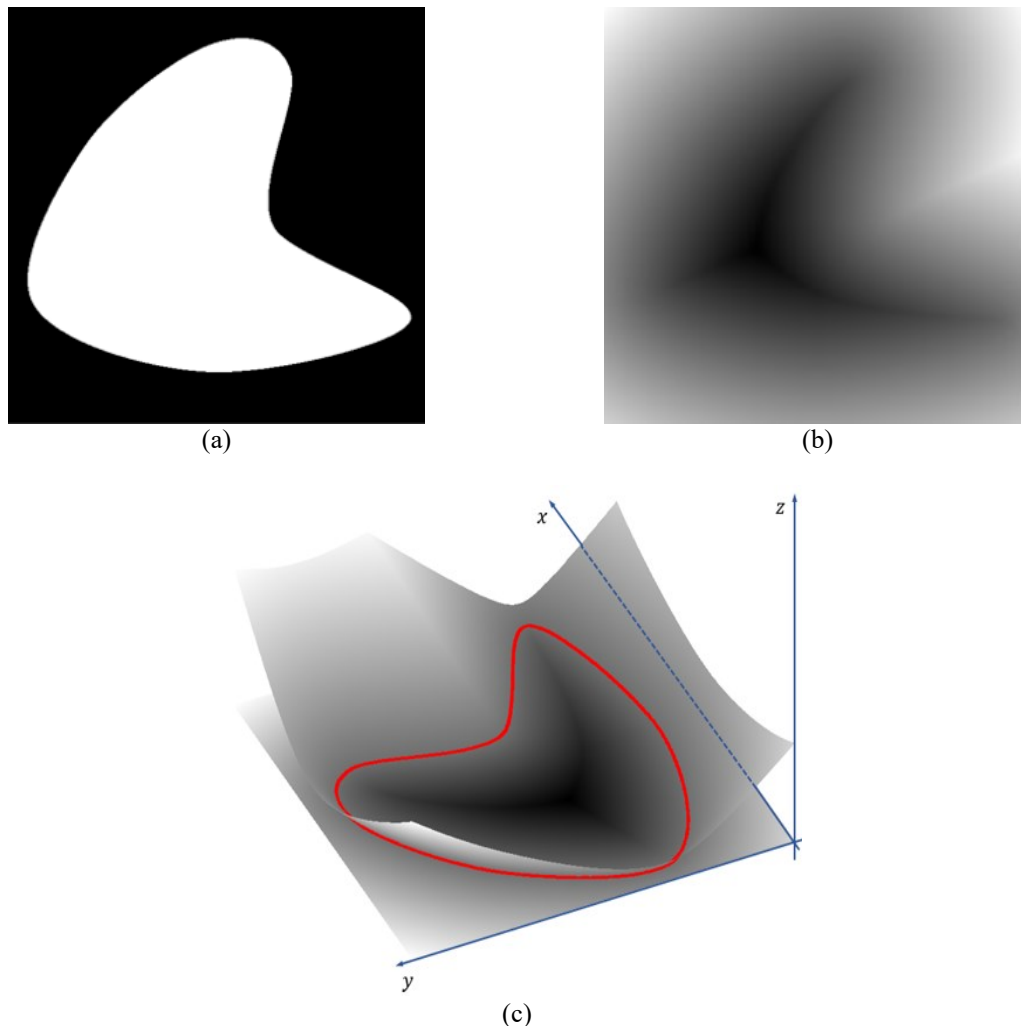

(a)   (b)

(c)

Figure 2. An illustration of the construction of the signed distance function from a binary label image;
(a) input label image is used to develop it, (b) corresponding sdf,
(c) shows the same sdf as viewed in 3D with its zero level set marked in red

As the name suggests, the GAC method calculates the feature g_I using the gradient magnitude information of the input image. In particular, it is calculated per pixel as:

$$g_I = \frac{1}{1 + |f|^2}$$ (5)

where the feature used is

$$f = \frac{\|\nabla(G_\sigma * I)\|}{max\|\nabla(G_\sigma * I)\|}$$ (6)

this feature is more popularly known as the normalized gradient magnitude of the input image I. The operation $(G_\sigma * I)$ is a convolution of the image I with a Gaussian kernel G_σ with width σ.

2.1. Drawbacks in using active contour models to segment medical images

Accurate segmentation of images, including medical images, with significant intensity inhomogeneity within the same region, is often difficult to achieve [22]. In our study of several medical image types, topology, and geometry, and while guided by the relevant medical experts, we found that the existing active contour models have a number of drawbacks when they are used to segment medical images. Although in this paper we only focus on the segmentation of lumbar spine MRI images as a case study, our rationale and findings should apply generally.

Borrowing from the approach we took in our previous study when delineating this type of images [5] to detect lumbar spinal stenosis [23], there are five regions of interests in each image. They are the intervertebral discs (IVD), posterior element (PE), thecal sac (TS), area between anterior and posterior (AAP) vertebrae elements, and Other (OT). The drawbacks that we found are summarized as follow:

− The true boundary of an object region may not be found by just using a single type of image features because it could depend on several different ones. For example, the boundary between the IVD with AAP regions is characterized by strong pixel intensity differences however the boundary between IVD and TS is not. Therefore, if the image gradient feature alone is used to improve the boundary of IVD it may result in its deterioration instead. While many different models of active contour algorithms have been proposed in the literature which use different types of image features, there is none, to the best of our knowledge, which allows multiple different ones to be adaptively applied along the curvature.

− The inclusion of all three components of speed functions in one calculation makes it harder to find the right combination to yield the best results. The effect of adjusting one control parameter may counter the effect of adjusting the others. In GAC, for example, generally using a large γ value pushes the contour harder towards an edge resulting in a contour evolution that moves strongly towards hard edges which will make it fits these edges rigidly. However, increasing the value of β could cancel that effect out because it penalizes high curvature. As a result, finding the right combination of parameter values is very tricky and can be application dependent. This phenomenon is illustrated in Figure 3, that shows the delineation of an unmodified label image (Figure 3 (a)) and the effects of applying the GAC algorithm using different combinations of γ and β values to the label image (Figure 3 (b)). The boundary of each region pair is color-coded to allow easier inspection. The figure shows that having low values of γ and β produces the worst results (top row and first four columns). The results are better when a large value of β is used (top row–fifth column). While fixing this β value, the effect of changing γ values does not produce significantly different results (last column). However, when the largest γ value is used the effect of changing β appears to reverse. This time lowering β value shows an improvement in the quality of the delineation.

On a further note, we notice that most active contour models are applied mainly to an image in its original size. This limits the precision of boundary points to half a pixel. Increasing the size by up sampling the image and its label k number of times, where $k \in \mathbb{N}$, can increase the precision to the nearest $2^{-(k+1)}$. However, this approach significantly increases the number of pixels to consider and our experience shows that active contour models can scale poorly with the number of pixels. Based on these factors, we specify three requirements that must be met for a suitable contour evolution method. They are 1) it must allow an application of evolution on a subset of contour as opposed to the entire contour. This will make it possible to tailor the evolution process using different image feature type for different parts of the contour and 2) it should decouple the different components of evolution function so that the boundary fitting can be optimized independently in each component space. Lastly, as the third requirement 3) the method must scale well with the increase in pixel numbers due to up sampling to increase the delineation precision.
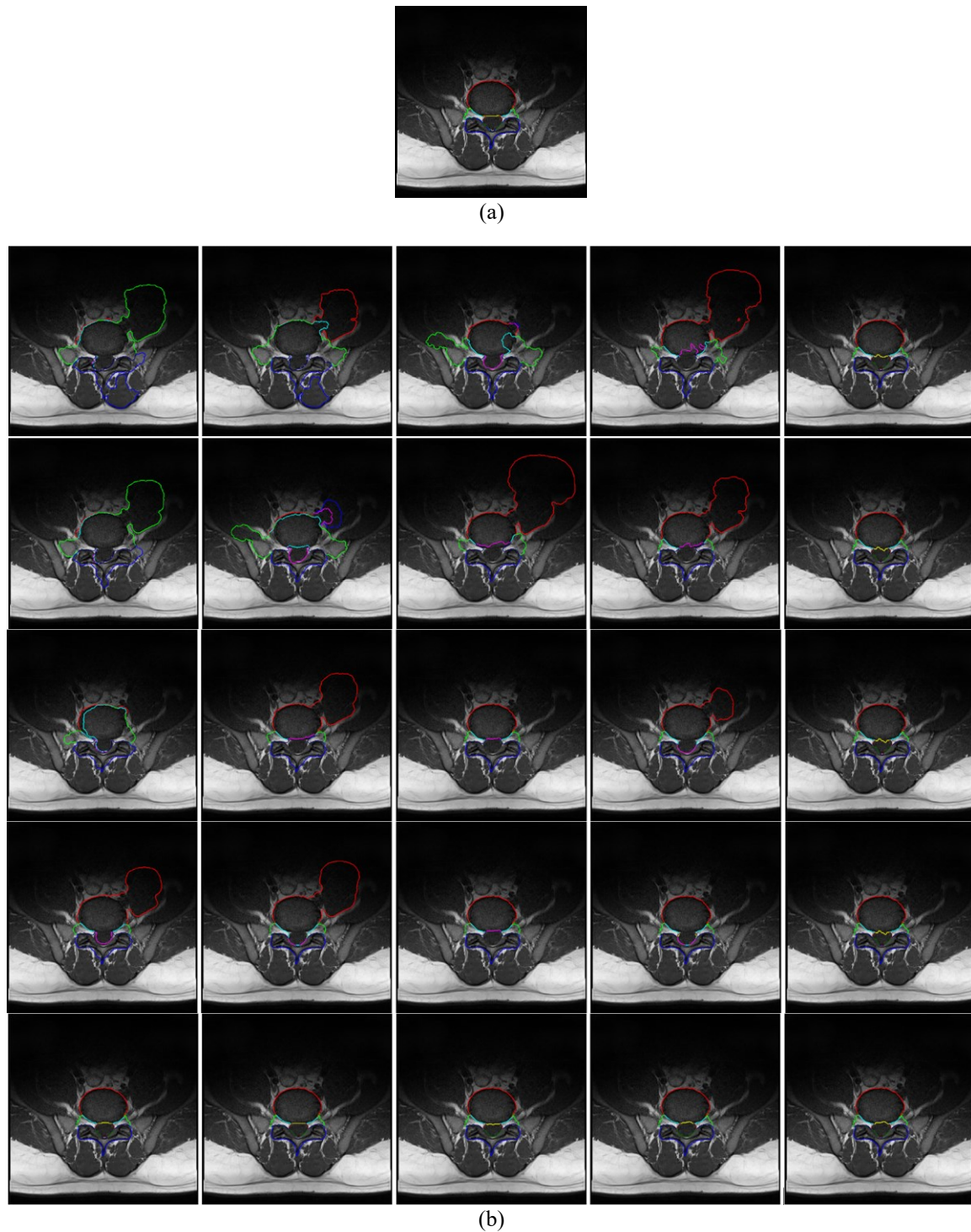
(a)

(b)

Figure 3. (a) Delineation of the original label image of a lumbar spine MRI image,
(b) The result of applying the GAC algorithm using different combinations of γ and β values;
From left to right, the γ values used are 0.1, 0.5, 1.0, 1.2, and 5.0.
From top to bottom the β values used are 0.1, 0.5, 1.0, and 1.2. The value of α is fixed at 1 throughout

3.   THE PROPOSED CONTOUR EVOLUTION METHOD

Image segmentation is a process that clusters pixels into multiple regions. The most straightforward way to represent the regions in an image is by using a matrix of labels with the same size as the original image. As an example, an instance where a 4x4 image that is segmented into three regions is illustrated in Figure 4. The figure depicts the region boundaries as red lines between pixels that have different labels.
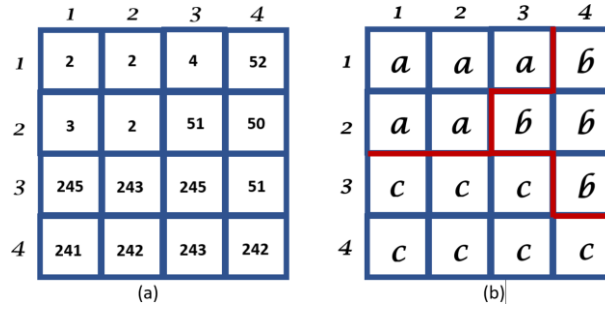
Figure 4. (a) An example of a4x4 image, (b) The image segmented into three regions

3.1. Boundary information representations

Boundary locations are naturally sub-pixels because it exists between adjacent pixels. They can be stored as a *boundary grid, B* which is a matrix whose size is twice that of the label image minus one. A version of the Boundary Grid that is used in [24] using the label image shown in Figure 4 (b) is shown in Figure 5.
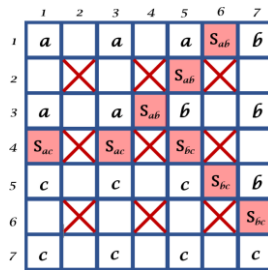

Figure 5. Boundary Grid of the label image shown in Figure 4 (b)

In the above example, cells with boundary information are shaded in red. Not all cells in a boundary grid can be boundary cells and in general only cells that have either 1) an even row number and an odd column number or 2) an even column number and an odd row number can be boundary cells. Furthermore, boundary cells in category (1) mark only horizontal boundaries whereas those in category (2) mark only vertical boundaries. Any of the cells that meet the criteria but are not boundary cells are illustrated as blank squares.

We made a small modification to this idea by including image feature values into these cells. The type of image feature chosen is a design decision and it is important to the accuracy and suitability of the algorithm to a given problem. However, in this paper, we use a similar type of image feature as GAC to ensure its direct relevance when comparing to the latter. We refer to this modification as the modified boundary grid, denoted as B'. The image function of B' is given as:

$$B' : \begin{cases} \Omega \to \mathcal{L} \mid (\Omega_1 \bmod 2) \neq 0 \wedge (\Omega_2 \bmod 2) \neq 0 \\ \Omega \to \mathfrak{R} \mid (\Omega_1 \bmod 2) = 0 \oplus (\Omega_2 \bmod 2) = 0 \\ \qquad \Omega \to \emptyset \mid otherwise \end{cases} \tag{7}$$

where Ω_i is the ith element of Ω and $\mathcal{L} = \{1, ..., N\}$ where N is the maximum number of regions in the image. The algorithm to construct our B' is as follows: Let $I : \Omega \to \mathfrak{R}$ be the input image function and $\chi : \Omega \to \mathcal{L}$ be its associated label image function, both with domain $\Omega \subset \mathfrak{R}^2$. At locations where $(\Omega_1 \bmod 2) \neq 0 \wedge (\Omega_2 \bmod 2) \neq 0$, B' contains the label information as prescribed in χ. At locations where $(\Omega_1 \bmod 2) = 0 \wedge (\Omega_2 \bmod 2) \neq 0$, B' contains vertical gradients and at locations where $(\Omega_1 \bmod 2) \neq 0 \wedge (\Omega_2 \bmod 2) = 0$, B' contains horizontal gradients.

In practice, the gradient values are calculated using the convolution of the input image I with a Gaussian kernel G_σ of width σ to minimize high-frequency elements in the derivative operation. Hence the horizontal and vertical gradient image functions $g_h : \Omega \to \mathfrak{R}$ and $g_v : \Omega \to \mathfrak{R}$ are defined as $g_h = \nabla_h(G_\sigma * I)$ and $g_v = \nabla_v(G_\sigma * I)$, respectively. The size of the horizontal and vertical gradient images are $H \times (W - 1)$ and $(H - 1) \times W$, respectively. To illustrate the process to construct our B' let's assume that the horizontal and vertical gradient images of the input images we used previously are shown in Figure 6 (a) and Figure 6 (b), respectively. The resulting B' is shown in Figure 6 (c).
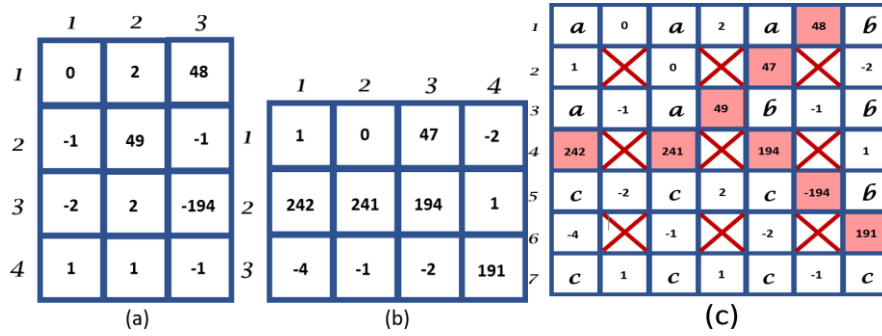
Figure 6. Gradient images of the input image shown in Figure 5 (a),
(a) horizontal, (b) vertical, (c) the result of modified boundary grid

This representation is, however, neither efficient nor compact if we want to use it in an iteration to search for the coordinates of a specific region's boundary. To compensate this shortcoming, we also use a *sparse boundary representation*, denoted as B_S, as a look-up table to index pairs of neighbouring regions. This is a much more compact and efficient representation of the boundary information because any search or information retrieval operations that are carried out in this representation are carried out faster due to them being dependent only on the number of boundary pixels rather than the number of pixels. We can construct B_S from B' by parsing the latter once, to gather all edge coordinates for each neighboring region pairs. Table 1 shows the sparse boundary representation of the Boundary Grid example we used earlier. Note that matrix subscripts are expressed as row and column order.

Table 1. Sparse boundary representation

Neighbors	Edge matrix subscripts (S)
a,b	(1,6) (2,5) (3,4)
a,c	(4,1) (4,3)
b,c	(4,5) (5,6) (6,7)

3.2. The contour evolution algorithm

In this section, we will discuss the algorithm to evolve the region boundaries using the two boundary representations described previously. Let S be a set containing the matrix subscript pairs of all the boundary points that we want to evolve using our chosen image feature. We construct S directly by querying B_S. For each element in S, we find its location in B', and based on the values of the coordinate we can ascertain the type of edge it is. A horizontal edge is evolved horizontally by evaluating the feature vectors in the horizontal direction. Likewise, vertical edges are evolved vertically. We need to decide the value of a parameter w, where $w \in \mathbb{N}$ and $w \neq 0$, which is the search width (in pixel) of the evolution. The value of w affects the speed and accuracy of the evolution and typically we want to use a small number. Large w value may result in a label that is very different from the original estimate. The evolution of the boundary contour will be based directly on S and is done by altering the label contents of B' as described in Algorithm 1 shown in Figure 7. Our new contour is then obtained directly by parsing the evolved B' while generating the new B_S. The algorithm is executed iteratively until the number of pixels in the Boundary Grid that change from one iteration to the next, averaged out over n number of iterations, converges below a specified threshold τ.

3.3. Subpixel boundary location

To further improve the precision of our method, we upsample our input and label images at the end of each complete evolution. We can repeat this for k number of times, where $k \in \mathbb{N}$, to increase the boundary precision to the nearest $2^{-(k+1)}$ of a pixel as described in algorithm 2 shown in Figure 7.

3.4. Adaptive curve smoothing

We design our method to allow the decoupling of the different components of the contour evolution speed. To this effect, we apply an adaptive curve smoothing function at the end of the above contour evolution process. The rationale for this is to allow us to adjust the tightness of the curve to the feature and

only concern about smoothing the results afterward. The curve smoothing is adaptive, in such a way that in areas where strong image features are present it applies a weaker smoothing function to preserve the edge than in areas where the image features are weak. We achieve this by means of variable width moving average method. Let $A = \langle p_n | n \in \mathbb{N}$ and $n < \text{card}(A) \rangle$ where $p_n \in \Re^2$ be a sorted sequence of set S. We define the sort operation such that $\|p_n - p_{n-1}\|$ for $1 \le n < \text{card}(A)$ is minimized. We then apply a moving average on A at every point p along its curvature with variable half-width $w_p \in \mathbb{N}$ such that:

$$p_i = \frac{\sum_{j=-w_p}^{w_p} p_{i+j}}{2 \times w_p + 1} \tag{8}$$

for $\forall p$ that meets the $w_p < p < \text{card}(A) - w_p$ requirement. The value of w_p is set between $w_{\min}$ and $w_{\max}$ and tied to the image feature at the location of the boundary points. In our experiment, we use the second derivative of the image function I'' as the image feature and set w_p to $w_{\min}$ when the I'' is at its lowest value, to $w_{\max}$ when the I'' is at its highest value and linearly interpolated and rounded to the nearest integer in between.

```
Algorithm 1. EvolveBoundaries
   Inputs:          Boundary Grid B'
                    Search width w
                    Union of all boundary points sets S
   Output:          Evolved Boundary Grid B'_out

let w_s = 2^w
let h = -w_s:2:w_s // h = [-ws, -ws+2, -ws+4, … 0, 2, 4, … ws]
let h_1 = -w_s:2:0     // h1 = [-ws, -ws+2, -ws+4, … 0]
let h_2 = 0:2:w_s      // h2 = [0, 2, 4, … ws]
let i_c = w + 1;
// w+1 is the center of the index vector marking current pixel
// hence the index vector is [1, 2, … w+1, … 2w+1]

for ∀(r,c) where (r,c) ∈ S
   let a = sign(B'(r,c));
   if a == 0 then a = 1
   if (r mod 2) == 0          // row is an even number hence vertical edge

      let g = B'(r + h,c) × a;
      // g is a vector with the same length as h
      let i = arg max g         // returns the index of max element
      let d = i_c − i           // distance to the highest gradient

      if d > 0                  // max gradient is above the current pixel
         let d_v = 1:2:2d       // all cells from current to max position
         B'(r − d_v,c) = B'(r + 1,c);
      else if d < 0             // max gradient is below the current pixel
         B'(r + d_v,c) = B'(r − 1,c);
   else if (c mod 2) == 0      // col is an even number hence horizontal edge
      let g = B'(r,c + h) × a;
      // g is a vector with the same length as h
      let i = arg max g         // returns the index of max element
      let d = i_c − i           // distance to the highest gradient

      if d > 0                  // max gradient is above the current pixel
         let d_v = 1:2:2d       // all cells from current to max position
         B'(r,c − d_v) = B'(r,c + 1);
      else if d < 0             // max gradient is below the current pixel
         B'(r,c + d_v) = B'(r,c − 1);
   B'_out = B';
```

```
Algorithm 2. SubPixelBoundaryEvolution
   Inputs:          Input Image I
                    Width of Gaussian smoothing kernel σ
                    Initial Label Image χ_o
                    Search width w
                    Iteration averaging window n
                    Standard Deviation threshold τ
                    Number of up samplings  k

   Output:          Evolved Label Image χ

let χ = χ_o
   construct G_σ      // a Gaussian kernel with width σ
   let D = ∅          // an empty sequence to store difference values,

   for i = 1:k
      let g_h = ∇_h(G_σ ∗ I)   // Horizontal gradient image
      let g_v = ∇_v(G_σ ∗ I)   // Vertical gradient image
      construct B' from χ,g_h and g_v           // use Eq.1
      construct B_S from B'
      select S from B'   // a set of important boundary points to evolve

      for t=1:t_max   // we put a hard cap on the number of iterations
         B'_out = EvolveBoundaries(B',w,S);
         let ∂ = |B'_out ≠ B'|
         // number of pixel changes between B'_out and B'
         append ∂ to D   // appending ∂ to the end of D sequence

         if card(D) ≥ n   // length of D exceeds n-1
            let v = √(Σ|D−D̄| / n)
            // use standard deviation as a measure of convergence
            if v < τ
               break;          // converges, so break from the loop
            else
               D[1] = ∅
               // remove the oldest value in D and ensuring card(D) < n

   χ = downsample(B',2)
   // we retrieve the label image from B' simply by
   // downsampling it at its every odd pixel location
   if i ≠ k
      // up sample the input image using cubic interpolation
      I = upsample(I,2)
      // up sample the label image using nearest interpolation
      χ = upsample(χ,2)
```

(a)   (b)

Figure 7. (a) Algorithm 1 to evolve the boundaries stored in a Boundary Grid and a Sparse Boundary Representation and (b) Algorithm 2 that iteratively up-samples the input and label images and makes calls to Algorithm 1 to achieve sub-pixel boundary evolution.

4. EXPERIMENTAL RESULTS AND ANALYSIS

We tested our method on a dataset of lumbar spine MRI images that we have collected [11] and their associated label images [12]. The dataset contains axial MRI scans of the last three intervertebral discs of 515 patients. We present in this section, one–in our view the best–example case because it contains all of the problems to solve as described in the last part of section 2. This example case is described and illustrated in Figure 8. The figure shows the color-coded boundaries of a manually created label image. The figure shows a number of problems with the boundaries. Areas labelled (a)-(d) are examples of gaps between regions that were meant to be neighbours. Area labelled (e) illustrates inaccurate boundaries that are a few pixels away from the true edge. And lastly, we note that these boundaries in general have high curvature which can be seen from its jaggedness such as those labelled (f).

The parameters that we use are as follows: image features $f = \nabla(G_\sigma * I)$, the search width $w = 2$, the number of upsampling levels $k = 2$, the iteration averaging window $n = 10$, the standard deviation threshold

$\tau = 2$, the min and max smoothing window half-widths $w_{min} = 5$ and $w_{max} = 20$, respectively. We applied our method to only evolve boundaries that we have prior knowledge to be gradient dependent while also improving the smoothness of the others. For comparison, we also apply the GAC algorithm [21] to the label image, sequentially for all regions except OT. We experimented with several combinations of α, β and γ values to produce an acceptable compromise between accuracy and smoothness, and we decided to use $\alpha = 1$, $\beta = 160$ and $\gamma = 500$ to be the best. We also apply a morphological closing on the input label image prior to applying each technique to remove any small holes and gaps in the image.
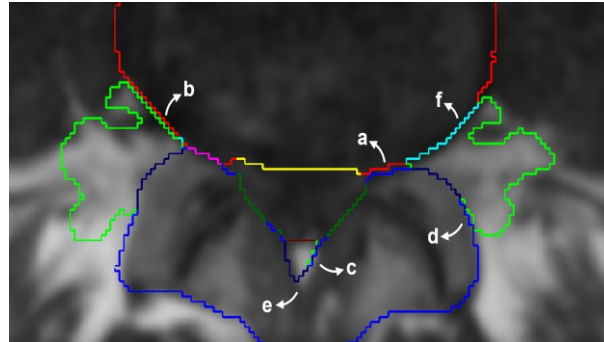

Figure 8. Color-coded boundaries of a manually created label image illustrating (a-d) gaps between supposedly neighbouring regions, (e) inaccurate boundary and (f) high curvature

On average, the execution speed of our method compared to GAC at up sampling levels $k = 0$, 1 and 2 are 10.2, 8.6, and 5.2 times faster, respectively. We observed that the main reason why our method is significantly faster is that it carries out a shorter number of iteration before it converges. As can be seen in Figure 9, our method solves the bulk of the pixel changes in the first iteration resulting in a much quicker convergence than GAC.
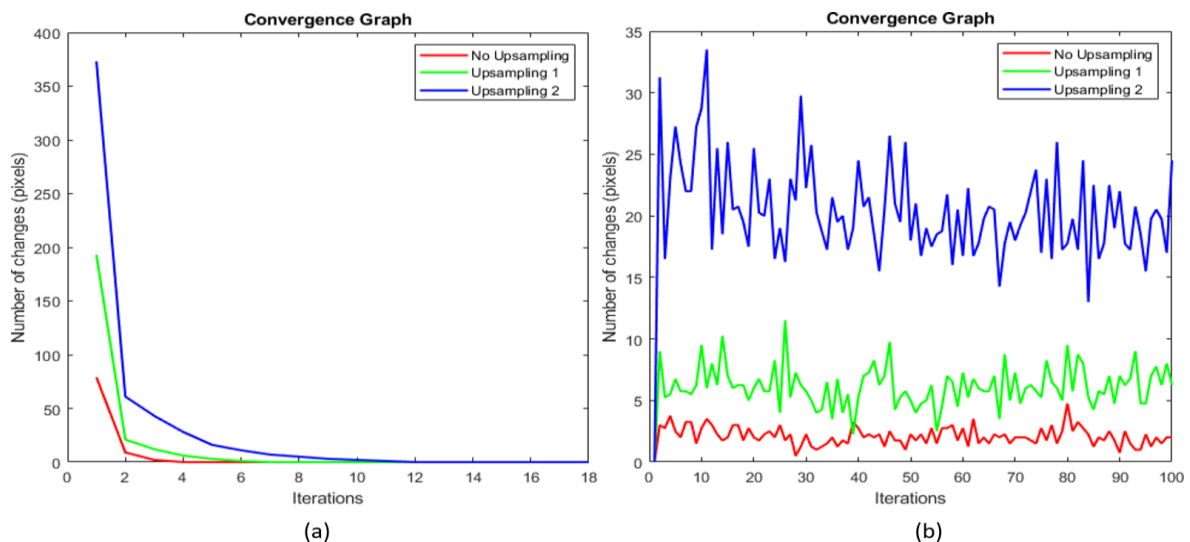

Figure 9. A typical number of pixel changes between successive iterations,
(a) proposed, (b) geodesic active contours

The contour evolution results of the two methods are shown in Figure 10. There are two notable improvements of our method compared to GAC. Our method primarily produces boundaries that are more tightly aligned with the true edge-marked as (e) in Figure 8 while at the same time produces a generally smoother contour than the latter. Note also the fact that GAC worsens the boundary between IVD and TS since this type of boundary is not characterized by strong intensity differences.
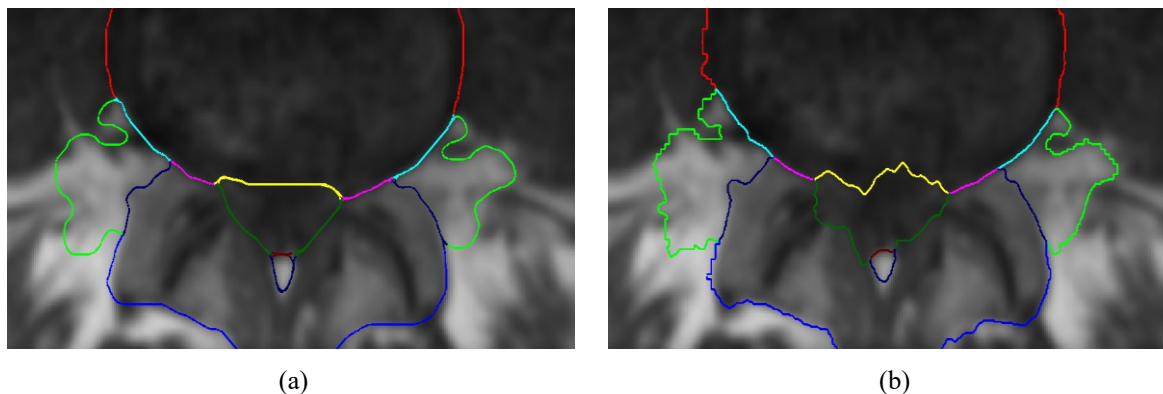
Figure 10. (a) The contour evolution results using our method, (b) The contour evolution results using GAC

To quantitatively measure the method's accuracy, the boundary-improved manually-labeled images were used to train a SegNet image segmentation model [5] which is then used to automatically label lumbar spine MRI images. The method was then subsequently applied to the segmented images to further improve their boundary. The method's performance in improving region boundaries can be quantitatively measured by BF-score semantic contour-based metric [25] of the boundary-improved automatically segmented label images with respect to its boundary-improved manually-labeled images. We used the with distance threshold values of 1, 2, and 3 pixels for this purpose. The results are shown in Table 2. For comparison, the same metric is calculated when GAC was used instead and when no modification was applied. The result clearly shows that our method is better at improving the boundary accuracy of the label images than GAC and when no modification was applied.

Table 2. BF-Score of the automatically segmented label images

Method	$d_T = 1$	$d_T = 2$	$d_T = 3$
Unmodified	0.74	0.89	0.92
Proposed	0.84	0.92	0.95
GAC	0.63	0.80	0.89

It is worth noting that the implementation of our method can be parallelized hence its effectiveness can be further optimized by utilizing GPU computation. Using the criteria set in [26], our method is suited for GPU computation due to its high data parallelism and low memory usage and branch divergence. The only drawback in this respect is the iterative nature of the process which requires synchronization at the end of each iteration.

5. CONCLUSION

We have proposed a new method of contour evolution that is suited to improving manually segmented medical images. Our method solves two main problems in Active Contour Models namely 1) their inability to use different image features at different segment of the contour and 2) the interdependency of the parameters of the contour evolution speed which makes finding good and suitable combination parameter values a difficult task. We tested our method on lumbar spine MRI images and our experimental results show that it can, not only, improve the accuracy of the boundary delineation of the manually segmented images, by 10 percentage points as measured using the BF-Score semantic contour-based metric, but also produce visually accurate and yet smoother contour than the traditional GAC method. Ours finding should apply generally even though we only focus on the segmentation of lumbar spine MRI images as a case study.

REFERENCES

[1] V. Jaiswal, V. Sharma, and S. Varma, "An implementation of novel genetic based clustering algorithm for color image segmentation," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 2, pp. 1461-1467, 2019.

[2] B. N. Li, C. K. Chui, S. Chang, and S. H. Ong, "Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation," *Comput. Biol. Med.*, vol. 41, no. 1, pp. 1-10, 2011.

[3] Z. Gao, W. K. Hau, M. Lu, W. Huang, H. Zhang, W. Wu, X. Liu, and Y.-T. Zhang, "Automated framework for detecting lumen and media-adventitia borders in intravascular ultrasound images," *Ultrasound Med. Biol.*, vol. 41, no. 7, pp. 2001-2021, 2015.

[4] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815-830, 2010.

[5] A. S. Al-Kafri, S. Sudirman, A. Hussain, D. Al-Jumeily, F. Natalia, H. Meidia, N. Afriliana, W. Al-Rashdan, M. Bashtawi, and M. Al-Jumaily, "Boundary delineation of MRI images for lumbar spinal stenosis detection through semantic segmentation using deep neural networks," *IEEE Access*, vol. 7, pp. 43487-43501, 2019.

[6] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. De Marvao, T. Dawes, D. P. O'Regan, and others, "Anatomically constrained neural networks (ACNNs): application to cardiac image enhancement and segmentation," *IEEE Trans. Med. Imaging*, vol. 37, no. 2, pp. 384-395, 2017.

[7] C. Lindner, S. Thiagarajah, J. M. Wilkinson, G. A. Wallis, T. F. Cootes, arcOGEN Consortium, *et al.*, "Fully automatic segmentation of the proximal femur using random forest regression voting," *IEEE Trans. Med. Imaging*, vol. 32, no. 8, pp. 1462-1472, 2013.

[8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481-2495, 2017.

[9] F. Natalia, H. Meidia, N. Afriliana, A. S. Al-Kafri, S. Sudirman, A. Simpson, A. Sophian, M. Al-Jumaily, W. Al-Rashdan, and M. Bashtawi, "Development of ground truth data for automatic lumbar spine MRI image segmentation," *IEEE 20th Int. Conf. on High Performance Computing and Communications; IEEE 16th Int. Conf. on Smart City; IEEE 4th Int.l Conf. on Data Science and Systems*, pp. 1449-1454, 2018.

[10] J. C. Young, N. Afriliana, F. Natalia, H. Meidia, and S. Sudirman, "A study on the suitability of applying active contour evolution models in segmenting and delineating boundaries in medical images," *5th International Conference on New Media Studies (CONMEDIA)*, pp. 232-237, 2019.

[11] S. Sudirman, A. Al Kafri, F. Natalia, H. Meidia, N. Afriliana, W. Al-Rashdan, M. Bashtawi, and M. Al-Jumaily, "Lumbar Spine MRI Dataset," *Mendeley Data*, 2019. [Online]. Available: https://data.mendeley.com/datasets/k57fr854j2/2. Accessed: 13-May-2019.

[12] S. Sudirman, A. Al Kafri, F. Natalia, H. Meidia, N. Afriliana, W. Al-Rashdan, M. Bashtawi, and M. Al-Jumaily, "Label image ground truth data for lumbar spine MRI dataset," *Mendeley Data*, 2019. [Online]. Available: https://data.mendeley.com/datasets/zbf6b4pttk/2.

[13] R. Sigit, C. A. Roji, T. Harsono, and S. Kuswadi, "Improved echocardiography segmentation using active shape model and optical flow," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 2, pp. 809-818, 2019.

[14] S. Bendazzoli, I. Brusini, P. Damberg, Ö. Smedby, L. Andersson, and C. Wang, "Automatic rat brain segmentation from MRI using statistical shape models and random forest," *Medical Imaging: Image Processing*, vol. 10949, 2019.

[15] E. E. Nithila and S. S. Kumar, "Segmentation of lung from CT using various active contour models," *Biomed. Signal Process. Control*, vol. 47, pp. 57-62, 2019.

[16] J. A. Sethian, "Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science vol. 3," *Cambridge University Press*, 1999.

[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321-331, 1988.

[18] M. H. Siddiqi, S. Lee, and Y.-K. Lee, "Object segmentation by comparison of active contour snake and level set in biomedical applications," *IEEE Int. Conf. on Bioinformatics and Biomedicine*, pp. 414-417, 2011.

[19] T. Chan and W. Zhu, "Level set based shape prior segmentation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 1164-1170, 2005.

[20] R. C. Gonzalez, R. E. Woods, *et al.*, "Digital image processing [M]," *Publ. House Electron. Ind.*, vol. 141, no. 7, 2002.

[21] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 6-79, 1997.

[22] K. Zhang, L. Zhang, K. M. Lam, and D. Zhang, "A level set approach to image segmentation with intensity inhomogeneity," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 546-557, 2016.

[23] F. Natalia, H. Meidia, N. Afriliana, A. Al-Kafri, and S. Sudirman, "Methodology to determine important-points location for automated lumbar spine stenosis diagnosis procedure," *International Conference on Intelligent Medicine and Health (ICIMH)*, pp. 53-57, 2019.

[24] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, "Convolutional oriented boundaries: From image segmentation to high-level tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 819-833, 2017.

[25] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37-63, 2011.

[26] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth, "Medical image segmentation on GPUs-A comprehensive review," *Medical Image Analysis*, vol. 20, no. 1, pp. 1-18, 2015.