Active Contour Model to Extract Boundaries of Teeth in Dental X-Ray Images

A. A. Niroshika  
Department of Information Technology  
Sri Lanka Institute of Information Technology  
Malabe, Sri Lanka  
aruni.n@sliit.lk

R. G. N. Meegama, T.G.I. Fernando  
Department of Statistics and Computer Science  
University of Sri Jayewardenepura  
Nugegoda, Sri Lanka  
{rgn,gishantha}@dscs.sjp.ac.lk

Abstract— Extracting the actual boundary of a tooth is useful in order to obtain an accurate shape to design a false tooth. A tooth in a dental X-ray image, however, exhibits sharp corners at its root making the boundary extraction task difficult. In this paper, a new technique based on active contours is presented along with a novel modification to detect sharp corners of objects of interest. The proposed technique has the ability to incorporate prior knowledge of significant corners of teeth into the deforming contour and is able to deform towards the boundaries of the object without surpassing corner points. Experimental results with several synthetic and real medical images show the ability of the new technique to extract features of interest from images consisting of sharp corners with increased accuracy.

Index Terms—Contours, Image Segmentation, Harris Operator, Snakes

I. INTRODUCTION

Deformable active contour models are widely used in computer vision in image segmentation and boundary detection. It has become more popular in extracting useful information from medical images; especially in medical image analysis that involves Computer Tomography (CT), Magnetic Resonance Imaging (MRI) and X-ray and ultrasound images [1 - 2].

Extracting boundary elements belonging to the same structure and assimilating these elements into a rational and consistent model of structures is a significant challenge in medical imaging [3]. However, this segmentation task, whereby the contours of the structures of interest are determined, is both difficult and time consuming to perform manually. Due to common image features such as texture, noise, image blur as well as non-uniform scene illuminations edge detection techniques frequently fail in producing accurate results. Therefore, in order to analyze large volumes of medical image data effectively, high-throughput automated tools are required. Unfortunately, fully automated segmentation techniques fail in producing confident results on biological images due to several issues such as; high noise levels, low local contrast and numerous structures surrounding the object of interest [4].

This expresses the fact that semi-automated tools are required to segment medical images with high accuracy and with more flexibility. In the past few decades, many effective algorithms have been developed under three main namely; thresholding techniques, pattern recognition and deformable models to perform computer assisted segmentation.

Deformable models can be classified into parametric or geometric models according to the way the contour is defined. Parametric deformable models represent the curves and surfaces explicitly in their parameter forms during deformation. These models have received a significant attention due to high computational efficiency and the ability to incorporate prior knowledge to the procedures.

There are two types of formulations for parametric deformable models: an energy minimizing formulation and dynamic force formulation. Among those two formulations, the energy minimizing formulation has the benefit that its solution satisfies a minimum principle while the dynamic force formulation has the flexibility of permitting the use of more general types of external forces.

The concept of parametric deformable contours, known as snakes, was first introduced by Kass [5] and over the past two decades, has been enhanced by many researchers worldwide. The original snake has several limitations on its performance. Thus, several new ideas such as: discrete snake [5], topological adaptive [6], balloon [7], fast greedy [8], gradient vector flow [9], B-spline [10], and NURBS [11] have been proposed and added to the concept of the original snake.

Deformable models and active contour based segmentation are considered as milestones in computer vision and image processing research [12]. However, problems such as poor capture range, placing the initial contour far from the desired object, problems with the concavities, high user interaction and problems with sharp corners have not been addressed fully in the literature. Moreover, almost all snake models given in literature are concentrated on maintaining a smooth continuous curve and do not address the issue of extracting boundaries of objects having sharp corners with discontinuities.

II. PARAMETRIC DEFORMABLE MODEL

In image analysis context, an active contour is a parametric curve that can dynamically change its shape in order to match with the geometry of salient image regions such as edges or boundaries. The main principle of energy minimizing
formulation of deformable models is to find a parameterized curve that minimizes the weighted sum of internal energy, image energy and constraint energy. The internal energy is defined within the curve itself and the image energy is computed from image data.

If a snake is defined as a parametric curve \( v(s) = (x(s), y(s))^T \), where \( x \) and \( y \) are the coordinate functions, and the energy of the snake, \( E_{\text{snake}} \), is defined as:

\[
E_{\text{snake}} = \int_0^1 E_{\text{internal}}(v(s)) + E_{\text{image}}(v(s)) + E_{\text{con}}(v(s)) \, ds.
\]

where \( E_{\text{internal}} \) represents the internal energy of the snake due to bending, \( E_{\text{image}} \) refers to image energy and \( E_{\text{con}} \) is the external constraint energy. The sum of the image energy \( E_{\text{image}} \) and the external constraint energy \( E_{\text{con}} \) is also known as the external energy. The internal energy of the snake is written as;

\[
E_{\text{internal}} = (\alpha(s) |v_s(s)|^2 + \beta(s) |v_{ss}(s)|^2) / 2
\]

where the first order term \( |v_s(s)|^2 \) gives a measure of the elasticity and the second order term \( |v_{ss}(s)|^2 \) gives a measure of the curvature of the deforming snake. The parameters \( \alpha(s) \) controls the “tension” of the contour while \( \beta(s) \) controls the “rigidity”. Therefore, the internal force holds the curve together (first order term) and keeps it from bending too much (second order term).

The image energy \( E_{\text{image}} \) (sometimes known as potential energy) is derived from the image intensity data. It attracts the snake towards desired features in the image and settles on local minima at the image intensity edges occurring at object boundary. The image energy of the snake is written as;

\[
E_{\text{image}} = -|\nabla I(x,y)|^2
\]

where \( \nabla \) denotes the gradient operator performed at \((x,y)\) in image \( I \). In order to remove noise from the image and to increase the capture range of the snake, the image can be convolved with a Gaussian kernel before computing the gradient.

III. RECENT WORK

The first parametric model known as “snake” was first proposed by Kass [5] that was very influential and sparked much research. This model could only attract the model towards the boundary when it is initialized near the desired object, or else it will converge into an undesired region. As an example; if the initial snake is placed far away from the subject boundary, the snake might deform randomly due to lack of gradient forces or it might get attracted towards undesired edges, lines or noisy areas. Consequently, it tends to be difficult for this model to progress into boundary concavities. In order to increase the efficiency and performance of the traditional snake, numerous new energy terms are added to the energy function later.

Topology adaptive snake (T-snake) was proposed by McInerny and Terzopoulos [6], by introducing topological flexibility among other features. This model employs an Affine Cell Decomposition (ACID) of the image domain. After every deformation step, the boundary of the object can be determined unambiguously by keeping track of the vertices inside the grid. This model has the ability to segment some of the most geometrically complex objects in an efficient and highly automated manner. It is capable of modifying its topology as required in order to fit the appropriate image data.

Cohen [7, 14] introduced a new model based on pressure forces also known as “balloons”. The pressure force is capable of either inflating or deflating the contour and therefore the initial contour can be placed either inside or outside the targeted object. This model eliminates the requirement to initialize the contour near the desired object boundary and is also capable of pushing the contour into boundary concavities. This model has several limitations such as overwhelmed weak edges and the tendency to form loops during deformation.

Xu and Prince [9, 15] have proposed a new deformable model called Gradient Vector Flow (GVF) snake. Instead of using image gradient as the external force, it uses spatial diffusion of the gradient of an edge map of the image. The amount of diffusion adapts according to the strength of edges to avoid distorting object boundaries. This model solves both problems associated with initialization and poor convergence towards boundary concavities. It is revealed that external forces derived from GVF snakes has a larger capture range than traditional snakes.

The majority of the above mentioned snake models suffer from several limitations such as: slow convergence speed (many control points), and difficult to adjust weighting factors for energy terms. Menet [10] introduced the perception of B-Spline snakes, emphasizing the advantages of local control, compact representation, and the possibility to include corners. This model, while addressing many of the problems associated with the former models, showed it could offer improved convergence speed and stability. However, B-Spline snakes are difficult to fit into sharp corners of objects completely, especially when there are discontinuities because B-Spline curves are inherently smooth.

The snake models discussed above are incapable of increasing the local flexibility of the contour without adding more control points. A more flexible model called Dynamic NURBS (or D-NURBS), a physics based generalization of Non-Uniform Rational B-Splines was introduced by Terzopoulos [11] for computer aided geometric design. This model has a capability to adjust the weight near a desired region interactively by the designer.

As seen above, many aspects of the original model have been modified and extended by many researchers. The latest snake models have a larger capture range and stronger convergence ability towards boundary concavities than traditional snakes. However, none of the above implementations provide a satisfactory solution to the complex object with sharp corners; so the active contours can still tend
to surpass the sharp corners of the object during deformation process.

This research addresses the problem associated with missing corner points near the sharp corners of a desired object. First, the significant corners of the desired object are selected manually by the operator or automatically by the Harris Corner Detector [13]. By re-parameterizing the model at user-specified iterations of the deformation process, it incorporates those pre-defined corner points into the deforming contour. Therefore, the proposed technique does not let the deforming contour to surpass such sharp corners. Experiments indicate that the new technique can improve the snake’s precision to capture boundaries with sharp corners.

IV. PROBLEMS WITH SHARP CORNERS

Although the concept of snake has been enhanced by many researchers during the past two decades, it still has several limitations on its performance and convergence. The proposed technique solves a significant problem encountered in the computer vision community since the introduction of active contours. The problem is that the snake cannot capture sharp corners precisely due to the inadequacy of the number of control points and the strength of the elasticity force near such sharp corners.

A. Inadequate Number of Control Points

The operator must manually define the initial contour with several control points located quite closer to the object. Thus, the initial contour depends on the user defined points that are placed around the desired object. If the initial contour is drawn with a fewer number of control points or if two adjacent control points are situated at a large distance apart near a sharp corner; there is a high probability of surpassing the sharp corners of the object during deformation of the contour.

Figure 1 illustrates the final position of the snake by running the original Kass snake with an image of a tree leaf, both with the same number of iterations and similar values for the parameters $\alpha$, $\beta$, and $\gamma$. In both cases, all the control points were placed quite close to the desired object manually by the operator. In Fig. 1(a), the initial contour was plotted using 15 control points which are spaced equally apart whereas in Fig. 1(b), the initial contour was plotted using 25 control points, and a sufficient numbers of control points were consciously placed near the left sharp peak of the tree leaf.

The output in Fig. 1 clearly indicates that the image drawn using 15 control points is not capable of capturing the sharp corner and as a result, the active contour had penetrated into the object. Though the initial contour is plotted closer to the object, it cannot cover the sharp corners due to a fewer number of points near the corner. On the other hand, it was able to extract the nearest boundary of the desired object including the left sharp corner when the number of control points was 25.

B. Strength of the Elasticity Force

The issue of surpassing corner points arises due to the strength of the elasticity force [16] as well. The moving equation for the parametric deformable models has been derived through the energy function which is a composition of elasticity force, bending force, image force and constraint force. The elasticity force tries to minimize the distance between the control points by keeping them equidistant along the contour. However, it has the effect of causing the contour to shrink. If the distance between two adjacent control points is too high, especially near a corner, the elasticity force tries to drag the control point away from the corner. Therefore, the accurate segmentation results are highly dependent on initial control points that are used to define the initial contour. Images with sharp corners may give rise to spurious convergence that surpasses the corner before it reaches the desired solution.

V. METHODOLOGY

A new technique to incorporate prior knowledge about the object corners into the deforming contour for object boundary detection is proposed and the original Kass [5] algorithm is used to implement the behavior of the active contour. The proposed technique introduces a novel method of contour construction which improves the capabilities of the active contour in detecting sharp corners. The proposed technique can be described in two phases as follows.

A. Definition of Corner Points and Initialization

The proposed Algorithm starts with manual placement of significant corner points at the boundary of the targeted object by the operator and it requires prior knowledge about the shape of the object and the target boundary. Although human intervention is needed to place the set of initial corner points, it can be automatically detected using an algorithm such as the Harris corner detector [13].

Then, all the significant corners of the desired object are placed in an array called corner_points [ ] which is defined as:

$$\text{corner\_points} [ ] = \{C(x,y) | i = 0,1,2,...,n - 1\}$$

(4)
The initial contour is plotted closer to the desired object using a set of control points known as “snaxels” and these snaxels are placed in a separate array defined as:

\[
\text{snaxel\_points} = \{ V_i(x, y) | i = 0, 1, 2, \ldots, m-1 \}
\]  

In order to achieve better results, it is advised to keep adjacent snaxels not too close to each other, so that the computation of the perpendicular distance to the nearest curve segments does not become tedious.

Before the deformation commences, the algorithm finds the nearest four adjacent snaxel points to each corner point \(C_i\) by considering the distance between the corner point and snaxels. A small distance between the corner point and the concerned snaxel point indicates that the snaxel point is closer to the corner point. By detecting the nearest four adjacent snaxel points for each corner point, the nearest three adjacent curve segments for each corner point can be detected. Therefore, the computation effort required to find the nearest curve segment at each re-parameterization step can be greatly reduced by selecting them at the beginning. These three nearest curve segments associated with each corner point are stored and tracked along with the corner point, and will be recalculated and modified after each re-parameterization step.

As illustrated in Fig. 2; \(V_{i-1} - V_i\), \(V_i - V_{i+1}\) and \(V_{i+1} - V_{i+2}\) are the nearest three adjacent curve segments with respect to user defined corner point \(C_i\).

**B. Deformation and Re-parameterization**

The proposed model is re-parameterized every after “\(r\)” time steps of the numerical time integration (referred to as a deformation step size), where “\(r\)” is user-controllable and typically set between 5 and 10. Consequently “\(T_d\)” is another new parameter referred to the threshold values for the distance between a specific corner point and its nearest curve segment. The value of “\(T_d\)” should be set according to the nature of the targeted object. Parameters \(a, \beta, \gamma, n\) and \(T_d\) should be tuned manually by the operator. After carefully setting these parameters, the initial contour deforms towards the desired object by minimizing the energy terms.

After each deformation step size “\(r\)”, it performs an efficient local search and computes the absolute perpendicular
distance “\(d\)” from each corner point to its nearest candidate contour segments as follows;

\[
d(ax + by + c = 0, (x_0, y_0)) = \frac{\left| ax_0 + bx_0 + c_0 \right|}{\sqrt{a^2 + b^2}}
\]  

If the perpendicular distance is lower than the specified threshold “\(T_d\)”, the corner point \(C_i\) is inserted between the respective curve segment and declares it as a new snaxel point. Subsequently, that particular curve segment is pulled towards the corner point. If there are more than one distance that is less than the \(T_d\), the algorithm selects the minimum distance. Now, the corner point has been merged with the deforming contour and becomes a new snaxel of the updated contour. The movement of the newly added corner point is restricted so that it remains stable without any movement during the deformation. However, the rest of the snaxels can move the contour towards the targeted object by effectively minimizing the energy over all deformation steps.

After adding such a new corner point, the new set of candidate contour segments should be detected again from the beginning for the remaining corner points. Consequently, the particular corner point is removed from the snaxel_points [ ], and inserted to the correct position in snaxel_points [ ].

**Fig. 2.** Position of the initial contour around the targeted object.

**Fig. 3.** Example of proposed re-parameterization steps. (a) At this point, the distance is less that \(T_d\). (b) Corner point \(C_i\) is added between \(V_i\) and \(V_{i+1}\), and the curve segment is pulled towards the corner.

**Fig. 4.** Example of proposed re-parameterization steps. (a) At this point, the distance is less that \(T_d\). (b) Corner point \(C_i\) is added between \(V_{i+1}\) and \(V_{i+2}\), and the curve segment is pushed backwards the corner.
As an example; after five iterations (where \( n = 5 \)), if \( d_i < T_d \), the corner point \( C_i \) is added in between \( V_i \) and \( V_{i+1} \) as shown in Fig. 3 and there onwards \( C_i \) is also considered as a snaxel point of the deforming contour.

Figure 3(b) depicts the new connection after successfully adding the corner point \( C_i \) into the deforming contour. The small arrows represent the direction of movement of the curve segment towards the object boundary. Therefore, it makes the contour converge faster and precisely in the regions where the image has corners.

However, at certain times, after several iterations, the contour may step over the corner point as well. Notice in Fig. 4 that the curve segment \( V_{i+1} - V_{i+2} \) has already stepped over the corner point \( C_i \). Even in such a scenario, during the next re-parameterization step, it will push the curve segment backwards to reach the corner point already surpassed in order to recapture it. In other words, it acts as a force that pulls (or pushes) the curve segment towards a desired corner point.

The proposed re-parameterization process provides an automatic and unambiguous mechanism to reconnect critical corner points into the deforming contour. Hence, the energy \( E_{snake} \) is minimized by moving the user defined snaxels towards the targeted object while keeping the merged corner points stable.

The following algorithm explains the mechanism behind the proposed technique.

**ALGORITHM: Deformation and Re-parameterization**

1. **Step 1:**
   - Placement of the significant Corner Points
   - corner_points = getCornetPoints (mouse input)
   - Placement of the Initial Contour
   - snaxel_points = getControlPoints (mouse input)

2. **Step 2:**
   - Finding the nearest three adjacent curve segments
   1. Calculate the distance “\( d \)” from the corner point \( C_i \) to each snaxel point
   2. Select four nearest adjacent control points
   3. Select three nearest adjacent contour segments
   //Start deformation
   **Step 3:**
   - Deform the initial contour according to the energy minimizing formulation
   1. After each deformation step size “\( n \)”, for each corner point
      1.1. Compute the perpendicular distance “\( d \)” from the corner point to each candidate contour segment

3. **Step 4:**
   - Merge the corner point into the deforming contour
   //If \( d < T_d \):
   - Add the corner point \( C_i \) in-between into the corresponding candidate contour segment
   - Remove the corner point \( C_i \) from corner_points
   - Push the corner point \( C_i \) into snaxel_points
   - Push/Pull the curve segment towards \( C_i \)
   - Go back to Step 2

**VI. EXPERIMENTAL RESULTS**

The proposed algorithm was implemented in Matlab version R2012a without code optimization, and executed in an Intel Core 2 Duo 2.66 GHz standard desktop computer. For the experimental purpose, the new technique was applied to the original Kass algorithm. Results are presented which demonstrate performance of the proposed technique on a number of synthetic test images and on real images.

**A. Experiments with synthetic images**

The convergence of the algorithm was tested on many synthetic images, where the region of interest consists of several sharp corners having both convex and concave shapes. Figure 5 shows the extracted boundary using the proposed algorithm on a synthetic image.

We have analyzed the convergence process by using standard distance error (SDE) proposed in [17]. This method has also been widely used to quantify segmentation results obtained from the active contours. Standard distance errors, with respect to the actual boundary of the object, were calculated to find the best approximates for the perimeter of the object.

**Fig. 5.** Illustrations of algorithm converging to the boundary of the foreground object in the synthetic image: (a) corner points detected using Harris operator, (b) additional control points defined by the operator, (c) the location of the contour after 50 iterations, (d) the location of the contour after 100 iterations and (e) final boundary extracted after 200 iterations.

Results yield an SDE of 1.37 with the proposed algorithm compared to 1.97 using Kaas algorithm. It is evident that the proposed algorithm gave the nearest actual boundary of the
object (while capturing all the sharp corners accurately) compared with the former method.

![Images of teeth](image1.png) ![Images of teeth](image2.png) ![Images of teeth](image3.png)

Fig. 6. Convergence of the active contour on dental images. (top row): Kass algorithm and (bottom row): Proposed algorithm.

<table>
<thead>
<tr>
<th>Image</th>
<th>Standard Distance Error with Kass Algorithm</th>
<th>Standard Distance Error with Proposed Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>2.7097</td>
<td>1.2317</td>
</tr>
<tr>
<td>Image 2</td>
<td>2.9530</td>
<td>1.0762</td>
</tr>
<tr>
<td>Image 3</td>
<td>1.7492</td>
<td>1.1729</td>
</tr>
</tbody>
</table>

**Table 1. Standard Distance Errors – Comparison**

**B. Experiments with Dental X-Ray Images**

The proposed algorithm was further tested on several dental X-Ray images as in Fig. 6 which shows the comparison of results for extracting the boundary by the traditional Kass [5] algorithm and the proposed algorithm. Parameters ($\alpha$, $\beta$, and $\gamma$) are chosen to be (0.40, 0.20, and 1.00) for all scenarios. As seen in Table 1, the proposed algorithm is capable of extracting the actual boundary of a tooth with increased accuracy compared to the traditional Kass algorithm.

Detecting the actual boundary of a tooth from an X-ray image is useful for a dentist to design a false tooth for a person whose current tooth is to be extracted. Such an accurate detection of the boundary enables the dentist to design the false tooth before the actual tooth is extracted from the patient.

**VII. CONCLUSION**

Though active contours are useful in many applications, most of the models are not capable of capturing the boundary of complex objects with sharp corners. In this paper, a new technique is proposed to re-parameterize the deforming contour by adding new corner points. Unlike traditional active contours, the proposed model can capture sharp corners of the targeted object accurately while it deforms. The results obtained after applying the technique on several synthetic and real images shows the increased performance of the proposed method compared with traditional active contour models. Extending this method to 3D to deform a surface onto a tooth is straightforward and such a 3D surface model can be transferred to a computer aided design (CAD) system to automatically design duplicate tooth with increased accuracy.

**REFERENCES**


