OBJECT DETECTION AND IMAGE SEGMENTATION USING TEXTURE PRESSURE ENERGY IN PARAMETRIC ACTIVE CONTOUR MODELS

Ali Reza Vard, Ahmad Reza Naghsh Nilchi and Payman Moallem*

ABSTRACT

In recent years, Active Contour Models (ACMs) have become powerful tools for object detection and image segmentation in computer vision and image processing applications. This paper presents a new energy function in parametric active contour models for object detection and image segmentation. In the proposed method, a new pressure energy called “texture pressure energy” is added to the energy function of the parametric active contour model to detect and segment a textured object against a textured background. In this scheme, the texture features of the contour are calculated by a moment based method. Then by comparing these features with texture features of the object, the contour curve is expanded or contracted in order to be adapted to the object boundaries. Experimental results show that the proposed method has more efficient and accurate segmenting functionality than the traditional method when both object and background have texture properties.

Key Words: parametric active contour models, energy function, image segmentation, texture features.

I. INTRODUCTION

During the last two decades, object detection and image segmentation have been fundamental tasks in computer vision and image processing research. Analysis of medical images, object detection in military applications, detection and extraction of video objects in the new standard multimedia compression methods like MPEG-4 and MPEG-7 are examples of image segmentation applications. Various methods have been proposed for image segmentation, which can be classified into two main groups: region-based approaches and boundary-based approaches. The first group tries to find partitions of the image which have homogeneous properties such as color and texture (Salembier et al., 1995; Wang, 1998). The second group segments images which rely on the information provided by object boundaries (Cheng et al., 2004).

Active contour models, more widely known as snakes, are an effective boundary-based method in segmenting an object of interest. In this method, contour is defined by user or is automatically generated around the object, then this contour is deformed by a driving energy function until it is fitted to the boundary object (Cootes et al., 1995; Yang et al., 2005). There are two types of active contour models (Li et al., 2005): parametric active contours and geometric active contours. Parametric active contours (Kass et al., 1987) are represented as parameterized curves and the snake evolution is carried out on the contour control points only. Geometric active contour models, or geodesic snakes, which were proposed by Caselles et al. (Caselles et al., 1993) and Malladi et al. (Malladi et al., 1995), are based on the theory of curve evolution and are numerically implemented via level set technique. In these models, geodesic snakes are represented implicitly as the zero-level sets
of higher dimensional surfaces, and the updating is performed on the surface function within the entire image domain. These models can automatically manage topology changes in an image. However, due to their computational complexity, their speed of convergence is slower than parametric active contours.

In this paper, we focus on parametric active contour models for object detecting and image segmentation. In the proposed method, we improve the energy function from previous models and present a new energy, called the texture pressure energy, to the energy function of the ACM, by means of which it is possible to detect and segment textured objects against textured backgrounds.

The remainder of the paper is organized as follows: Section II presents an overview of previous algorithms and various types of active contour models. A mathematical description of ACM is given in section III. The proposed method is explained in section IV. The experimental results are given in section V. Finally, the paper is concluded in section VI.

II. OVERVIEW OF PREVIOUS WORKS

Active contour models were first introduced by Kass et al. in 1987 (Kass et al., 1987). A snake is an elastic curve which is driven by the minimization of an energy function in order to move it toward the features of interest in an image such as lines, edges and corners. This method for segmentation and video object tracking was useful, however, it had some problems like optimality and convergence, for minimizing an energy function.

Amini et al. pointed out that the Kass method can be numerically unstable and there is a tendency for points to bunch up on strong portions (Amini et al., 1990). They proposed a dynamic programming approach for minimizing an energy function, that allows addition of hard constraints to obtain more desirable behavior of snakes. However it is fairly slow having a time complexity of order \(O(nm^3)\) where a contour has \(n\) points which are allowed to move to any point in a neighborhood of size \(m\) at each iteration.

Williams and Shah introduced a greedy algorithm which is much faster, having a time complexity of order \(O(nm)\), which achieves comparable performance in both image segmentation and object tracking, but it has difficulties progressing into concave boundary regions (Williams and Shah, 1992).

McInerney and Terzopoulos proposed a class of deformable contours, called topology adaptive snakes or T-snakes, that provides a mechanism for contour reparameterization, which enables T-snakes to split or merge, adapting to the topology of the object (McInerney and Terzopoulos, 2000). The T-snakes can be used to segment some of the most complex–shaped biological structures from medical images.

Gunn and Nixon suggested a dual active contour model, where one active contour is placed inside the object and expands to the object boundary and another active contour is placed outside the object and it contracts to the object boundary (Gunn and Nixon, 1997). This technique helps to reduce the sensitivity to contour initialization and allows a selective rejection of weak local minima solution.

Ivins and Porrill (Ivins and Porrill, 1994), Hamarneh et al. (Hamarneh et al., 2000), Schaub and Smith (Schaub and Smith, 2003) introduced various types of color pressure snakes. In these models, by adding region or color pressure energy (some times called color inflation energy) to the energy function of the active contour model, it is possible to detect objects with weak edges.

III. MATHEMATICAL DESCRIPTION OF ACTIVE CONTOUR MODELS

The traditional snake is a parametric curve which is defined as \(S(u) = \{x(u), y(u)\}, u \in [0, 1]\). This curve deforms shape and moves through the spatial domain of an image to minimize the energy function (Kass et al., 1987; Schaub and Smith, 2003):

\[
E = \int_0^1 E_{\text{Snake}}(S(u)) \, du, 
\]

where snake’s energy, \(E_{\text{Snake}}\), at a certain point, \(u\), is the weighted sum of internal and image energy (Almageed and Smith, 2004). Eq. (2) describes the total energy of snakes.

\[
E = \int_0^1 (E_{\text{int}}(S(u)) + E_{\text{img}}(S(u))) \, du, 
\]

where \(E_{\text{int}}(S(u))\) is an internal energy density per length and \(E_{\text{img}}(S(u))\) is an image energy density per length. The internal energy controls the contour shape and it is only related to the geometry property of the contour and is defined as:

\[
E_{\text{int}} = \frac{\alpha}{2} \left( \frac{\partial^2 S(u)}{\partial u^2} \right)^2 + \frac{\beta}{2} \left( \frac{\partial^2 S(u)}{\partial u^2} \right)^2. 
\]

The first-order derivative controls contour stretching and the second-order derivative controls contour bending. \(\alpha\) and \(\beta\) are the weighing parameters controlling the snake’s tension and rigidity respectively.

Image energy is used to drive the contour towards the desired image features, such as boundaries. This energy in traditional ACMs is most commonly estimated as the result of edge detection and is to be calculated as follows (Xu and Prince, 1997):

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

$$E_{img} = E_{edge} = -|\nabla(G_d(x, y) \ast I(x, y))^2|$$  \(5\)

where \(I(x, y)\) is the image intensity at \((x, y)\), \(G_d(x, y)\) is a 2D Gaussian kernel with standard deviation \(\sigma\). \(\nabla\) and \(\ast\) are shown gradient and convolution operator respectively. A motivation for applying Gaussian filtering in \(5\) to the underlying image is to reduce noise.

Consequently, the total energy of active contour is defined as follows:

$$E = \frac{\alpha}{2} \int \frac{\partial}{\partial u} S(u) \bigg| \frac{\partial}{\partial u} du + \frac{\beta}{2} \int \frac{\partial^2}{\partial u^2} S(u) \bigg| \frac{\partial}{\partial u} du$$

\[+ \int E_{edge}(S(u)) du. \quad (6)\]

If the salient image features (strong edges) are presented, the contour will be driven towards the object boundaries. Unfortunately in the absence of strong edges, this model cannot detect objects correctly and it tends to collapse. To overcome this limitation, a color pressure energy is used that replaces edge energy in energy functions of ACM (Ivins and Porrill, 1994; Hamarneh et al., 2000; Schaub and Smith, 2003; Almageed and Smith, 2004). This energy creates a force that pushes the contour toward the region boundaries of the object. The pressure (region) force used in (Ivins and Porrill, 1994) and (Almageed and Smith, 2004) is defined as follows:

$$f_{\text{pressure}}(S(u)) = \rho \cdot G(I(S)) \frac{3G}{\partial u} \perp,$$

where \(\rho\) is the weighting coefficient and is defined by user, \(S\) is the contour. The \(\perp\) indicates that the image force is perpendicularly applied to the tangent of the contour. \(G\) is a function that according to (Hamarneh et al., 2000) is defined:

$$G(I(S)) = \begin{cases} +1 & \text{if } I(S) \geq T \\ -1 & \text{otherwise} \end{cases}.$$

\(8\)

where \(T\) is an image intensity threshold.

The \(G\) function is also calculated from statistical characteristics of image intensity, as follows (McInerney and Terzopoulos, 2000):

$$G(I(S)) = \begin{cases} +1 & \text{if } \left(\frac{I(S) - \mu}{\sigma}\right) \leq k \\ -1 & \text{otherwise} \end{cases},$$

\(9\)

where \(\mu\) is the mean image intensity of the object, \(\sigma\) the standard deviation of the object intensity, and \(k\) is a user defined parameter. The values of \(\mu\) and \(\sigma\) are typically known a priori or computed from the image. If object and background are simple, color pressure snakes can have good results. However if we want to identify textured objects against textured backgrounds, these models can not identify boundaries of objects correctly.

**IV. THE PROPOSED METHOD**

In this section, we introduce a new pressure energy, which is called “texture pressure energy”, to detect and segment textured objects against textured backgrounds. At first in sub section I, we describe the method of computing texture features and then in sub section II we present the new pressure energy based on texture features.

1. **Texture Feature Computation**

Various approaches have been proposed for computation and extraction of texture features. Based on kind of application (accuracy and speed), each one has advantages and disadvantages (Tuceryan and Jain, 1993).

In this paper, we apply the approach based on moments to extract texture features (Tuceryan, 1994). The \((p + q)\)th order moments \(m_{pq}\) of a function of two variables \(f(x, y)\) with respect to the origin \((0, 0)\) are defined as:

$$m_{pq} = \iint_{W} f(x, y)x^p y^q dx dy \quad p + q = 0, 1, 2, \cdots$$

\(10\)

In the proposed approach, as in Tuceryan’s work (Tuceryan, 1994), we regard the intensity image as a function of two variables, \(f(x, y)\). We compute a fixed number of the lower order moments for each pixel in the input image (we use \(p + q \leq 2\)). The moments are computed within small local windows around each pixel. Given a window size \(W\), the coordinates are normalized to the range \([-1, 1]\) and the pixel is located at the center. The moments are computed with respect to this normalized coordinate system. This permits us to compare the set of moments computed for each pixel. Let \((i, j)\) be the pixel coordinates for which the moments are computed. For a pixel with coordinates \((m, n)\) which falls within the window, the normalized coordinates \((x_m, y_n)\) are given by:

$$x_m = \frac{m-i}{w/2}, \quad y_n = \frac{n-j}{w/2}.$$  \(11\)

Then the moments within a window centered at pixel \((i, j)\) are computed by a discrete sum approximation of Eq. (10) that uses the normalized coordinates \((x_m, y_n)\):

$$m_{pq} = \sum_{-w/2}^{w/2} \sum_{-w/2}^{w/2} f(m, n)x_m^p y_n^q.$$  \(12\)
convert the moment images into features describing the textures present in the image.

(3) Construct the feature vectors $F_i$ from the transformed images for all contour points.

Next we modify pressure force, Eq. (7), and combine it with texture feature. Then we define texture pressure force as follows:

$$f_{\text{feature}}(S(u)) = \rho T(I(S)) (\frac{\partial S}{\partial u})^\perp, \quad (14)$$

where $\rho$ is the weighting coefficient, $S$ is a snake curve. The $\perp$ indicates that the texture pressure is perpendicularly applied to the tangent of the contour. $T$ is a function that is defined in binary form:

$$T(I(S)) = \begin{cases} +1 & \text{if } \sum_{i=1}^{6} \left( \frac{F_i(I(S)) - \mu_i}{\sigma_i} \right)^2 \leq k \\ -1 & \text{otherwise} \end{cases}, \quad (15)$$

where $F$ is texture feature vector of contour, $\mu_i$ and $\sigma_i$ are mean and standard deviation of the texture feature vector of the object which exist as prior knowledge or are calculated from the image. $k$ is a parameter that is defined by user.

The seed region or object appearance identifies positive vs. negative pressure regions. When a portion of the contour is in a positive region, it expands. When a portion of the contour is in a negative region, it contracts. It follows that the minimum energy of the contour lies on the pressure boundary between the positive and negative regions.

Unfortunately, using binary pressure on an active contour model cannot achieve equilibrium because the pressure is always non-zero; rather, it will oscillate around object boundaries. To solve this problem, we define the $T$ function as follows:

$$T(I(S)) = 1 - \frac{1}{k} \sqrt{\sum_{i=1}^{6} \left( \frac{F_i(I(S)) - \mu_i}{\sigma_i} \right)^2}. \quad (16)$$

This function makes the model expand rapidly when $||F(I(s)) - \mu|| \equiv 0$. Conversely, when $||F(I(s)) - \mu||$ is very large the model will contract very quickly. Of course, when $||F(I(s)) - \mu|| \equiv k\sigma$ the pressure force will be zero, allowing the model to stabilize.

In order to automatically determine the value of $k$, we estimate it by statically calculating as follows:

$$k = \left| \frac{B\mu - O\mu}{O\sigma} \right|. \quad (17)$$

where $B\mu$ is mean of texture feature vector of background (Fig. 2). If changes of background texture are small, a constant $k$ will be sufficient but if back-
color pressure snake based on color pressure energy and traditional active contour based on edge energy in MATLAB-7 software. In order to evaluate efficiency and accuracy, we tested our method on both simple objects and backgrounds as well as with textured objects and backgrounds. In the following we investigate the simulation results. All the experiments described here were conducted on a Pentium (M) machine running at 1.86 GHz with 512 MB of memory and the Windows XP operating system. In all experiments, ACM points are initialized by user and, during the energy minimization process at each iteration, adding points to contour or removing points from contour, according to distances and angles among neighbor points is done dynamically.

The first experiment (Fig. 2) indicates estimation of k parameter and detection of object boundaries by using texture pressure energy. As in Fig. 2(a), user specifies some points around the object that belong to background [B1, B2, B3 and B4]. Texture feature vectors of these points are calculated by using Eq. (13) and mean vector (BO) is obtained. Then, the center of those points (O point), that belongs to the object, and a number of points around the O point are selected. Texture feature vectors of these points are also calculated. Their mean vector (OU) and standard deviation vector (OS) are obtained. After that, k is determined by using (17) and contour by applying Eq. (18) deforms to fit object boundaries (Fig. 2(b)).

The second experiment (Fig. 3) was carried out to compare three ACMs for segmenting a concave object against a simple background. As in Fig. 3.1, the traditional active contour, using the edge energy, cannot detect object boundaries correctly but the color pressure snake, using the color pressure energy, (Fig. 3.2) and our proposed method, using the texture pressure energy, (Fig. 3.3) detect the object with high accuracy. In order to measure the accuracy of segmentation, we defined error measure, according to (Seo et al., 2006), as follows:

\[ E_{SCB} = \frac{SCB(n)}{n} \]  

(20)

where \( n \) is the number of snake pixels (snaxels) and \( SCB(\cdot) \) refers to the number of snaxels on the correct boundary. This measure converges to 100% when correct boundary detection is done. Three methods for segmentation of two shapes (starfish and cat) are summarized in Table 1.

In order to compare the convergence speed of color pressure snake (Fig. 3.2) and our proposed method (Fig. 3.3), we compare run times and number of iterations for the two methods. Table 2 shows that the run time of our method is slower than color pressure snake. This is related to texture computation, which is a time consuming operation. But our method...
Table 1 Comparative result of segmentation for three methods

<table>
<thead>
<tr>
<th>Image</th>
<th>Segmentation method (%)</th>
<th>$E_{SCB}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starfish</td>
<td>Traditional active contour</td>
<td>9.7143</td>
</tr>
<tr>
<td></td>
<td>Color pressure snake</td>
<td>94.8837</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>96.3158</td>
</tr>
<tr>
<td>Cat</td>
<td>Traditional active contour</td>
<td>82.1429</td>
</tr>
<tr>
<td></td>
<td>Color pressure snake</td>
<td>98.6111</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2 Comparison of the convergence speed for the color pressure snake and the proposed method

<table>
<thead>
<tr>
<th>Image</th>
<th>Image size</th>
<th>Segmentation method</th>
<th>Iteration number</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starfish</td>
<td>800 × 600</td>
<td>Color pressure snake</td>
<td>40</td>
<td>5.1870 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed method</td>
<td>35</td>
<td>14.9380 sec</td>
</tr>
<tr>
<td>Cat</td>
<td>320 × 240</td>
<td>Color pressure snake</td>
<td>28</td>
<td>1.8750 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed method</td>
<td>22</td>
<td>4.5150 sec</td>
</tr>
</tbody>
</table>

Fig. 3 Comparative result of three methods for segmentation of two shapes ((a) Starfish and (b) Cat).

3.1 Segmentation using the traditional active contour
3.2 Segmentation using the color pressure snake
3.3 Segmentation using the proposed method

Fig. 4 Segmentation of a textured object against a textured background. As is shown in Fig. 4.1, color pressure snake cannot detect a textured object and loses it, but our proposed method is successful in finding object boundaries and correctly segmenting.
In the fourth experiment (Fig. 5), we compared the performance of our proposed method with color pressure snake for image segmentation in the presence of noise. In this experiment we added Gaussian noise with zero mean and different standard deviations ($\sigma = 0.1, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4$) to the image and evaluated the resistance against noise of the proposed method with color pressure snake. As is shown in Fig. 5.1, color pressure snake is more sensitive to noise and by increasing the noise, correct object detection decreases greatly. But as is shown in Fig. 5.2 the proposed method is more resistant against noise and by increasing noise correct object detection is made less difficult.

Figure 6 illustrates the error measure $E_{SCB}$, which is calculated by Eq. (20), vs. different values of noise standard deviations (Noise STD) for the proposed method and color pressure snake.

In the last experiment, we considered the proposed method for segmenting textured objects against textured backgrounds where noise is added to the image. In this experiment we added Gaussian noise with zero mean and different standard deviations to the image and evaluated the resistance against noise of the proposed method. The result of this experiment for $\sigma = 0.1, 2.5, 4$ is shown on Fig. 7.

Figure 8 illustrates the error measure $E_{SCB}$ vs. different value of noise standard deviations (Noise STD) for proposed method.

VI. CONCLUSIONS

Contour curves in the traditional ACMs lead to object boundaries by edge energy, where strong edges are not present, those models cannot detect objects correctly. To solve this problem color pressure snakes are presented. In these models, a contour curve towards an object is expanded or contracted by applying pressure energy. However these models cannot identify textured objects against textured backgrounds. In this paper,
we modify the pressure energy of previous models and introduce texture pressure energy. In the proposed approach, by applying texture features of object and background, which are calculated based on moment method, we are able to segment texture object in texture background. The experimental results indicate which proposed method is superior both in accuracy and resistance against noise over previous snake algorithms.

**NOMENCLATURE**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B\mu$</td>
<td>mean of texture feature vector of a background</td>
</tr>
<tr>
<td>$E$</td>
<td>total energy of active contour model</td>
</tr>
<tr>
<td>$E_{\text{edge}}$</td>
<td>edge energy</td>
</tr>
<tr>
<td>$E_{\text{img}}(S(u))$</td>
<td>image energy density per length</td>
</tr>
<tr>
<td>$E_{\text{int}}(S(u))$</td>
<td>internal energy density per length</td>
</tr>
<tr>
<td>$E_{\text{SCB}}$</td>
<td>error measure</td>
</tr>
<tr>
<td>$E_{\text{Snake}}$</td>
<td>snake’s energy</td>
</tr>
<tr>
<td>$F$</td>
<td>texture feature vector of contour</td>
</tr>
<tr>
<td>$F_t$</td>
<td>texture feature image</td>
</tr>
<tr>
<td>$f_{\text{pressure}}$</td>
<td>pressure (region) force</td>
</tr>
<tr>
<td>$f_{\text{texture}}$</td>
<td>texture pressure force</td>
</tr>
<tr>
<td>$G_\sigma$</td>
<td>2D Gaussian kernel with standard deviation $\sigma$</td>
</tr>
<tr>
<td>$I(x, y)$</td>
<td>image intensity at $(x, y)$</td>
</tr>
<tr>
<td>$k$</td>
<td>specified parameter</td>
</tr>
<tr>
<td>$M_t$</td>
<td>moment image</td>
</tr>
<tr>
<td>$m_{pq}$</td>
<td>$(p + q)^6$ order moments</td>
</tr>
<tr>
<td>$n$</td>
<td>number of snake pixels (snaxels)</td>
</tr>
<tr>
<td>$O\mu$</td>
<td>mean of texture feature vector of an object</td>
</tr>
<tr>
<td>$O\sigma$</td>
<td>standard deviation of texture feature vector of an object</td>
</tr>
<tr>
<td>$S(u)$</td>
<td>snake or parametric active contour</td>
</tr>
<tr>
<td>$SCB(\cdot)$</td>
<td>number of snaxels on a correct boundary</td>
</tr>
<tr>
<td>$u$</td>
<td>parameter of active contour</td>
</tr>
<tr>
<td>$(x_m, y_n)$</td>
<td>normalized coordinates</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>weighing parameters controlling the snake’s tension</td>
</tr>
<tr>
<td>$\beta$</td>
<td>weighing parameters controlling the snake’s rigidity</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>parameter that controls the shape of the logistic function</td>
</tr>
<tr>
<td>$\mu$</td>
<td>mean of image intensity of an object</td>
</tr>
<tr>
<td>$\rho$</td>
<td>weighting coefficient</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>standard deviation of image intensity of an object</td>
</tr>
<tr>
<td>$\nabla$</td>
<td>gradient operator</td>
</tr>
<tr>
<td>$*$</td>
<td>convolution operator</td>
</tr>
<tr>
<td>$||$</td>
<td>norm operator</td>
</tr>
</tbody>
</table>

**REFERENCES**


Caselles, V., Catte, F., Coll, T., and Dibos, F., 1993,


*Manuscript Received: Apr. 27, 2007*

*Revision Received: Jan. 30, 2008*

*and Accepted: Mar. 19, 2008*