World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:2, 2012

Active Contours with Prior Corner Detection

U.A.A. Niroshika and Ravinda G.N. Meegama

***Abstract*—**Deformable active contours are widely used in computer vision and image processing applications for image segmentation, especially in biomedical image analysis. The active contour or "snake" deforms towards a target object by controlling the internal, image and constraint forces. However, if the contour initialized with a lesser number of control points, there is a high probability of surpassing the sharp corners of the object during deformation of the contour. In this paper, a new technique is proposed to construct the initial contour by incorporating prior knowledge of significant corners of the object detected using the Harris operator. This new reconstructed contour begins to deform, by attracting the snake towards the targeted object, without missing the corners. Experimental results with several synthetic images show the ability of the new technique to deal with sharp corners with a high accuracy than traditional methods.

***Keywords*—**Active 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 CT, MRI and X-ray and ultrasound images [1 - 2].

The concept of deformable contours, known as snakes, was first introduced by Kaas [3] and over the past two decades, it has been enhanced by many researchers worldwide. The active contour is initialized by plotting an approximation curve using a number of control points around the desired object and it is then attracted towards the object by controlling the internal forces, image force, and constraint forces. The internal forces, such as elasticity and rigidity, serve to impose a piecewise smoothness of the snake; the image force pushes the snake towards salient image features like lines and edges while external constraint forces are responsible for placing the snake near the desired local minimum.

If a snake is defined as a parametric curve **v**(*s*) = (x(*s*), y(*s*)), the energy of the snake, E_{snake}, is defined as:

$$E_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))\,ds \tag{1}$$

$$= \int_0^1 E_{int\,ernal}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))ds.$$

where $E_{internal}$ represents the internal energy of the snake due to blending, E_{image} refers to image force and E_{con} is the external constraint force.

The internal energy of the snake is written as;

$$E_{internal} = (\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2 / 2 \tag{2}$$

where the first order term $|\mathrm{v_s(s)}|^2$ gives a measure of the elasticity and the second order term $|\mathrm{v_{ss}(s)}|^2$ gives a measure of the curvature of the deforming snake. The influence that these terms have on the overall snake is controlled by the parameters α(s) and β(s). Once the initial contour is placed, it begins to deform by trying to minimize the internal energy while converging into a local minima.

The image force E_{image} is derived from the image intensity data. It attracts the snake towards desired features in the image and settles on edges. The image energy of the snake is written as;

$$E_{image} = -\|\nabla I(x,y)\|^2 \tag{3}$$

where $\nabla I(x,y)$ denotes the gradient of the image *I* at pixel location *(x,y).*

The snake evolves over time attempting to minimize the total energy of its contour. If **v**(*s*,*t*) represents a point on a dynamic curve at a discrete time instance *t*; the point at time *t*+1, **v**(*s*,*t*+1) is given by;

$$\mathbf{v}(s,t+1) = \arg\min_{\mathbf{v}(s,t)\in V} E_{snake}\mathbf{v}(s,t) \tag{4}$$

where *V* is the set of all possible curve points **v**(*s*,*t*).

The original snake has several limitations on its performance. Thus, several new ideas such as: discrete [3], topological adaptive [4, 5], balloon [6], fast greedy [7], gradient vector flow [8], B-spline [9], and NURBS [10] have been proposed and added to the concept of the original snake.

Deformable models and active contour based segmentation are mostly claimed as one of the key successes of the computer vision community. A most fertile ground for further development and application of these ideas certainly has been within the medical image analysis community [11]. However, problems such as poor capture range, initial contour placed 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.

U.A.A. Niroshika is with the Department of Information Technology, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka (e-mail: aruni.n@sliit.lk).

Ravinda G.N. Meegama is with the Department of Statistics and Computer Science, Faculty of Applied Science, University of Sri Jayewardenepura, Nugegoda, Sri Lanka (e-mail: rgn@sjp.ac.lk).

This research addresses the problem associated with missing corner points near the sharp corners of a desired object. A new technique is proposed to overcome this problem.

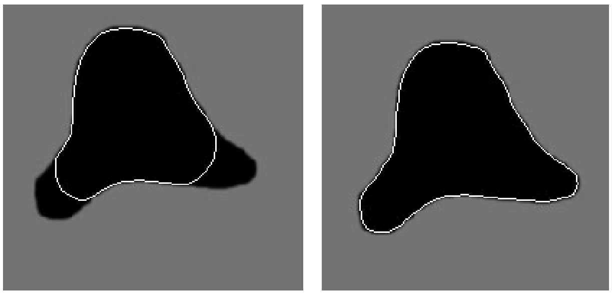
First, the corner points are checked using the Harris operator [12] and are detected as energy minimization points. Then the initial snake is reconstructed by adding the detected corner points to capture the boundary. Experiments indicate that the new technique can improve the snake's precision to capture the boundary with sharp corners.

II MISSING CORNER POINTS

The proposed technique solves a significant problem encountered in computer vision community since the introduction of active contours. The problem is that the snake cannot capture sharp corners precisely, due to inadequacy of the number of control points near the corner and the strength of the elasticity force near the sharp corner.

A. Less 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. Then, the contour gets attracted towards the targeted object during the energy minimization process. If the initial contour is drawn with fewer number of control points, there is a high probability of surpassing the sharp corners of the object during deformation of the contour.



(a) (b)

Fig. 1 Convergence of the active contour based on initial number of control points. (a): Initial contour drawn with 15 control points and (b): Initial contour drawn with 25 control points

Fig.1 illustrates the final position of the snake by running the original Kaas snake with a synthetic image, both with same number of iterations and similar values for the parameters α, β, and γ.

The output in Fig. 1 clearly indicates that, image (a) drawn using 15 control points, is not capable of capturing sharp corners and as a result, the active contour 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.

B. Strength of the Elasticity Force

The issue of surpassing corner points arises due to the strength of the elasticity force [13]. 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 as shown in Fig. 2.

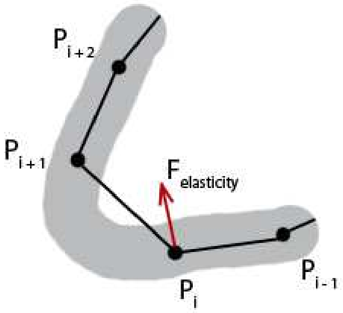


Fig. 2 The elasticity force $F_{elasticity}$ drags the control point into the object

As such, it is evident that if the number of control points in the initial contour is inadequate and if the distance between two neighboring control points is too high near a corner, there is a great chance of falsely capturing sharp corners of object of interest. As a result, the contour may shrink over the object near a sharp corner.

III. PROPOSED TECHNIQUE

A new technique to incorporate prior knowledge about the object corners into the initial contour for object boundary detection is proposed by using the Harris operator to detect corners and the original Kaas [3] algorithm to implement the behavior of the active contour.

The technique presented in this paper consists of three main steps; (A) detecting significant corners using Harris operator, (B) defining additional control points and (C) incorporating both detected corner points and additional points into the initial contour.

A. Detection of Corners using Harris Operator

Corner points are an image feature characterized by their high intensity changes in the horizontal and vertical directions. A variety of operators have been defined to detect corners where the most widely used corner detector being the Harris corner detector [11].

The Harris operator is based on the local auto-correlation function of a signal where the function measures the local change of the signal with patches N_0 shifted by a small amount in different directions. Given a shift (Δx, Δy) and a point (x, y), the auto-correlation function is defined as;

$$C(x,y) = \sum_{w} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \qquad (5)$$

A is a 2×2 matrix computed from image derivatives, defined as;

$$A = \sum_{(x,y)\in N_0} \nabla I(x,y) \nabla I(x,y)^T \qquad (6)$$

Let λ_1 and λ_2 be the eigenvalues of A. The Harris corner detector is based on the assessment that near a corner, both λ_1 and λ_2 are large and positive. Harris corner detector can be defined as;

$$H(x,y) = \lambda_1 \lambda_2 / (\lambda_1 + \lambda_2) \qquad (7)$$

During the first step, the desired object should be selected. This selection of the area is based on the prior knowledge of the desired object or an interactive manual placement by the operator. Then all the significant corners of the desired object are detected using Harris operator and the points are placed in an array as follows:

harris_points [ ] = {$\mathbf{P}_i(x,y)$| $i = 0,1,2,\ldots,n - 1$} (8)

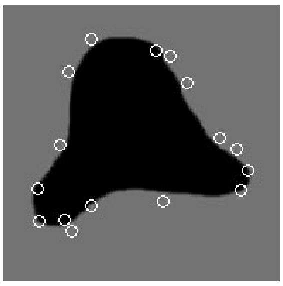


Fig. 3 Significant corners were detected using Harris operator

Fig.3 illustrates the significant corner points detected by the Harris corner detector for the same synthetic image. The Harris operator was able to capture the points near the corners (bottom two corners) successfully. In general, the Gaussian filter is applied on the image to increase the capture range.

B. Defining Additional Control Points

The required additional control points are placed manually by the operator somewhat closer to the desired object and these additional control points are placed in a separate array as follows:

additional_points [ ] = {$\mathbf{P}_i(x,y)$| $i = 0,1,2,\ldots, m$-1} (9)

C. Reconstruction of the Initial Contour

At the next step, both arrays are merged together and a new array is constructed as follows:

all_points [ ] = {$\mathbf{P}_i(x,y)$| $i = 0,1,2,\ldots, n$+m-1} (10)

This new array contains both the corner points detected by the Harris operator and the additional control points given by the operator in an unordered manner.

Before the contour deforms, in order to reconstruct the initial contour, it rearranges all points according to the correct order by considering the angle clockwise as in the following algorithm;

```
// n is the total number of corner points detected by the Harris detector
// m is the total number of additional points defined by the operator
mid_point = mid point of the desired object
reference_point = a point corresponds to the mid point
θ = angle corresponds to the reference point, middle point and current point

Load (image)

// Detection of corners using Harris operator
harris_points [ ] = getCornetPoints (execute Harris operator)
// Detection of additional control points
additional_points [  ] = getCotrolPoints (mouse input)

// merge two arrays
all_points [ ] = harris_points [ ] + additional_points [  ]

// Reconstruction of the initial contour
for (i = 0 to (n+m)-1) {
   current_point = $i^{th}$ point

   $A^2$ = $distance^2$ (middle_point , reference_point)
   $B^2$ = $distance^2$ (middle_point , current_point)
   $C^2$ = $distance^2$ (reference_point , current_point)
   θ = arccos (( $A^2 + B^2 - C^2$) / 2AC)

   if (θ > 180°)
      θ = 360° - θ
   end if
   theta_array  [  ] = {$\mathbf{P}_i$ (θ angle) , (x , y)}
end for

//sort the array by the θ angle according to the ascending  order
final_points [ ] = Sort (theta_array [  ], ascending)
//final_points [ ] contains all the control points and corner points according to the correct order

//reconstructed initial contour will be the inputs to the snake algorithm
Contour = Snake Algorithm (final_points [ ])
```

As seen in Fig.4, the proposed algorithm is capable of reconstructing the initial contour by concatenating both pre-detected corner points and additional control points defined by the operator.

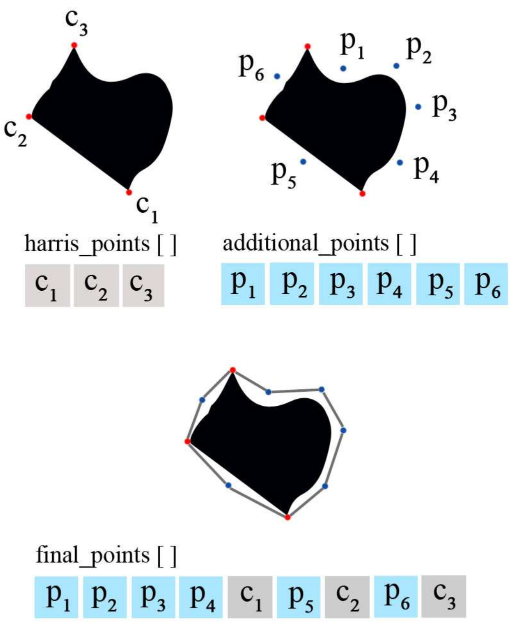


Fig. 4 Reconstruction process of the initial contour. (top row left): Corners detected using Harris operator, (top row right): Additional control points and (bottom row): New initial contour

IV. Experimental Results

For the experimental purpose, the new technique was applied to original Kaas algorithm. The convergence of the original Kaas algorithm and proposed algorithm (incorporating prior detected corners using Harris operator) on several synthetic images including objects with sharp corners are illustrated in Fig. 5.

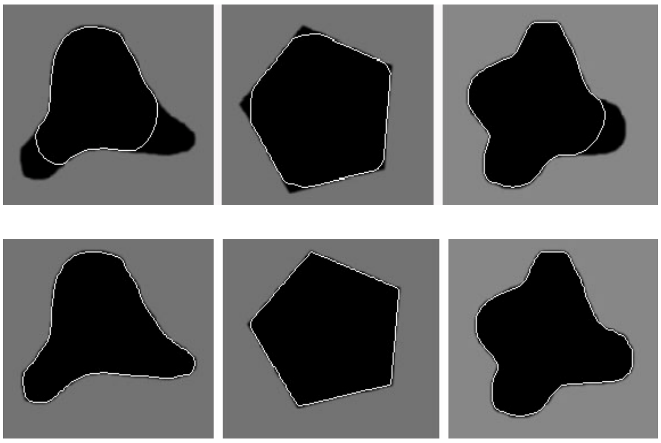


Image 1 Image 2 Image 3

Fig. 5 Convergence of the active contour on synthetic images (top row): Kaas algorithm and (bottom row): Proposed algorithm

Parameters (α, β, and γ) are chosen to be (0.40, 0.20, and 1.00) for all scenarios. Initial contour was first plotted using 20 control points for all the cases. Additional numbers of control points were added to places on the object where sharp corners were automatically detected. For all the images, the Gaussian filter is applied on the image to boost the captured range.

As seen in Fig. 5, the final output shows that the boundary detection was achieved best with the proposed algorithm even when the snake was initialized with a fewer number of control points for an object with sharp corners.

Standard distance errors [14], with respect to the actual boundary of the object, were calculated to find the best approximates for the perimeter of the object and tabulated in Table I where the results clearly indicate the increased capability of the proposed algorithm to extract the nearest boundary of the desired object.

TABLE I
Comparison Using Standard Distance Errors

Image	Standard Distance Error with Kaas Algorithm	Standard Distance Error with Proposed Algorithm
Image 1	3.8756	1.1532
Image 2	1.6444	1.1475
Image 3	2.8777	1.1950

The algorithm is capable of reconstructing the initial contour by adding the detected corner points after giving careful attention to the order and position of points. Subsequently, the newly reconstructed initial contour starts to deform as a snake guided by defined forces. When it deforms, the snake behaves in a quite different manner. The detected corner points tend to stay in the same location without any movement since the image energy is minimum at those places.

V Conclusion

Though active contours are useful in many applications, most of the models are not capable to capture the boundary of complex objects with sharp corners precisely. In this paper, a new technique is proposed to reconstruct the initial contour by incorporating prior knowledge of the significant corners of the object of interest detected using the Harris operator. The computations are straight forward; new initial contour is reconstructed using pre-detected corner points. The results obtained after applying the technique on several synthetic images shows the increased performance of the proposed method compared with traditional active contour models even with fewer numbers of initial control points.

References

[1] T.McInerney and D.Terzopoulos. “Deformable models in medical image analysis”: A survey. *Medical Image Analysis*, vol. 1, no. 2, pp. 91-108, 1996.

[2] A. Singh, D. Goldgof, and D. Terzopoulos, editors. “Deformable models in Medical Image Analysis”. *IEEE Computer Society Press*, 1998.

[3] M. Kaas, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models”, *Int. Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331, 1988.

[4] T. McInerny, D. Terzopoulos, “T-sankes: Topology adaptive snakes”, in Proc. *International Conference on Computer Vision*, pp. 840-845, 1995.

[5] Wai-Pak Choi, Kin-Man Lam an Wan-Chi Siu, “An adaptive active contour model for highly irregular boundaries”, *Pattern Recognition*, vol. 34, pp. 323-331, 2001.

[6] L. D. Cohen, “On active contour models and balloons”, *CVGIP: Image Understanding*, 53(2), pp. 221-218, 1991.

[7] K. M. Lam and H. Yan, “Fast greedy algorithm for active contours”, *Electronic Letters*, vol. 30, no.1, pp. 21-23, 1994.

[8] C. Xu and J. L. Prince, “Gradient Vector Flow: A New External Force for Snakes”, in *Proc. IEEE Conf. on Computer Vision and pattern Recognition (CVPR)*, Los Alamitos: Comp. Soc. Press, pp. 66-71, June 1997.

[9] S. Menet, P. Saint-Marc, and G. Medioni. “B-snakes: Implementation and application to stereo”. *In proceedings DARPA*, pp. 720-726, 1990.

[10] R. G. N. Meegama, J. C. Rajapakse, “NURBS snakes”, *Image and Vision Computing*, vol. 21, no. 6, pp. 551-562, 2003.

[11] J. S. Duncan, N. Ayache, “Medical Image Analysis: Progress over Two Decades and the Challenges Ahead”, in *Proc. IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 90-92, 2000.

[12] C. Harris and M. Stephens, “A combined corner and edge detector”, in *Proc. 4th Alvey Vision Conference*, pp. 147-151, 1988.

[13] J. Ahlberg, “An Active Contour in Three Dimensions”, *Thesis project at Computer Vision Laboratory, Linkoping University*, pp. 17, 1996.

[14] L. Staib, J. Duncan, “Model-based deformable surface for medical images”, *IEEE Transactions on Medical Imaging*, 15 (6), pp. 859-870, 1996.