Segmentation with active contours:
a comparative study of B-Spline and Level Set techniques

DEMIAN WASSERMANN[1], MARTA E. MEJAIL[1], JULIANA GAMBINI[1], MARÍA E. BUEMI[1]

[1]UBA–Universidad de Buenos Aires, FCEyN, Departamento de Computación
Ciudad Universitaria, Pab. I, 1428 Buenos Aires, República Argentina
{dwasser,marta,jgambini,mebuemi}@dc.uba.ar

Abstract. Active contours approach is a very used family of techniques in image analysis. This work presents a comparative study between two active contour approaches for image segmentation. Level sets method and deformable contours under B-spline representation are compared. These image segmentation methods have different features and are difficult to compare in terms of performance, accuracy and initialization conditions. Both are implemented and a way to calculate the approximation error is developed. As a conclusion of this work a theoretical comparison and a comparative characterization of the approximation error for each method are presented.

1 Introduction

Image feature extraction is required for several applications such as medical image analysis, audio-visual speech analysis, computer assisted design and robotics.

Active contours approach applied to image segmentation was introduced by Kass [4] and it was a new approach in feature extraction. Active contours are curves that evolve within the image until they find the boundaries of the interest object. Based on the original work, different flavors have been developed in literature in order to improve stability, flexibility and accuracy.

Level sets´ active contour approach, originally presented in [6], is based on a curve representing an interface between two inmisible fluids (see [10]). The evolution of the interface is modeled in time by means of partial differential equations. This technique is used due to its flexibility and accuracy. On the other hand B-splines deformable models approach, originally developed by Blake (see [1]), segments an image finding object boundaries by B-spline curve fitting. This technique main advantages rely on the balance between accuracy and low computational cost.

The methods of level set and B-spline deformable contours have different features and are difficult to compare in terms of performance, accuracy and initialization conditions. In this work a comparative study between both active contour methods is presented. In the first place a theoretical introduction for these techniques is presented. Secondly both methods are compared over a set of random generated images measuring the approximation error, hence each method advantages and drawbacks are exhibited.

The organization of this paper is as follows: section 2 describes the level set model for contour detection, characterizes the curve evolution and presents the numerical implementation. Section 3 gives an introduction to B-Spline curve fitting, specifies the criterion used to determine the transition points and explains the region fitting algorithm. Section 4 shows the obtained results and finally, section 5 presents a comparative study of both techniques and the advantages and disadvantages of each method.

2 The level set approach for contour detection

In this section we present level set methods and their numerical representation (see [7]).

The mathematical front evolution model proposed by [7] and [9] is the following:
let

$$C : [0, S] \times \mathbb{R}^+ \to \mathbb{R}^2$$
$$C(s,t) = (x(s,t), y(s,t))$$

where s and t are the spatial and temporal parameters, satisfying the differential equation:

$$C_t(s,t) = \vec{F} \qquad (1)$$

where C_t is the time derivative and $\vec{F}$ is a vectorial function.

Considering the tangential $\vec{C_T}$ and normal $\vec{C_N}$ components of the curve, the equation (1) is rewritten as:

$$C_t(s,t) = \vec{C_T}(s,t)F_{\vec{T}}(s) + \vec{C_N}(s,t)F_{\vec{N}}(s)$$

and in turn, according to the parameterized curve evolution theorem (see [9]), it becomes into the equation:

$$C_t(s,t) = \vec{C_N}(s,t)F(s)$$

where $F : \mathbb{R} \to \mathbb{R}$ is the projection of $\vec{F}$ along the normal direction of the curve $\vec{C_N}$.
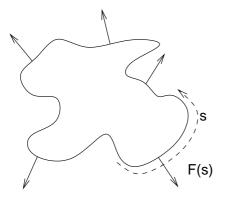
Figure 1: Front propagation according to normal velocities

An example of front propagation under normal curve evolution is shown in Figure 1 where curve parameterization is taken that the normals lines are external.

Given an initial curve $C_0(s)$, front propagation is modeled by following the equations:

$$C_t(s,t) = \vec{C_N}(s,t)F(s)$$
$$C(s,0) = C_0(s)$$

2.1 Curve evolution characterization

The speed function F of any point in the evolving curve, can be characterized as depending on three kinds of properties (see [9]), local curve properties (normal direction, curvature), global curve properties (area) and properties that are external to the curve.

Curve evolution in time is formulated using the *fundamental theorem of plane curves* ([3]), which states that two plane curves which have the same curvature differ only by an Euclidean motion, i.e. translation and rotation. The equation (2) can be written as:

$$C_t(s,t) = F(k(s,t))C_{\vec{N}}(s,t)$$

where $k(s,t)$ is the curvature at any point, in a moment t.

2.2 Numerical implementation

A natural approach is to consider a Lagrangian formulation of the problem, obtaining the motion equations for $x(s,t)$ and $y(s,t)$. In this approach the parameterization is discretized into a set of marker particles over the domain parameterization. These particles are updated in time approximating spatial derivatives and thus changing their locations. Finally, propagating front will then be reconstructed using these particles. Nevertheless, this approach has several flaws discussed in [9]. Firstly small errors in the computed location of the particles are deeply amplified in the curvature estimation thus leading the model to numerical instability. Secondly if there is no smoothing term for the curvature (viscosity) in the differential equation, singularities will develop on the propagating front and an entropy condition must be applied in order to obtain a weak solution of the differential equation. Finally topology changes on the curve are hard to handle.

As an alternative, the level set approach proposed by Sethian and Osher in [10] and [7] represents the front $C(s,t)$ as a level set of a function ϕ which is one dimension higher than $C(s,t)$. Thus, given a closed curve $C(s,t)$ which evolves in time, the given front in a moment t will be given by the set $\Gamma(t) = \{\vec{x} \in \mathbb{R}^2 : \phi(\vec{x},t) = c\}$.

Therefore an Eulerian mathematical model for front evolution is stated: let $\gamma(t) = C(s,t)$ be the front evolving along its normals with speed F, and let $\phi(\vec{x},t=0), \vec{x} \in \mathbb{R}^2$, satisfying

$$\phi(\vec{x},t=0) = \pm d$$

where d is the signed distance from s to $\gamma(t=0)$ and the sign is chosen according to the point position inside or outside of the curve.

Thus, the evolution equation for $\phi(\vec{x},t)$ is obtained, ([6]):

$$\phi_t + F|\nabla\phi| = 0 \qquad (2)$$

where F can have several parameters including, among others, curvature or normal direction.

An advantage to this formulation is that ϕ will remain a function as long as F is smooth, hence the level set $\Gamma(t)$ can change its topology, merge, split and develop sharp edges as ϕ evolves, see [7].

A second advantage concerns numerical approximation. Because ϕ remains a function as it evolves, a discrete grid in the domain and finite differences numerical approximations to the temporal and spatial derivatives can be used.

However, there are cases where the solution to the differential equation can lose its differentiability in a moment t_p and the solution curves for $t > t_p$ lack interpretation as a propagating front. An example of this problem is presented (see [9]) for the initial curve with a constant evolution speed $F(k(s,t))$.

$$\begin{cases} C(s,0) & = & \left(1-s, \frac{1+cos(2\pi s)}{2}\right) \\ F(k(s,t)) & = & 1 \end{cases}$$

The evolution of this curve for $t = 0; 0.05; \ldots; 0.20$ is shown in Figure 2. There a "sharp corner" can be devised in $t = 0.05$, the curvature and the normal cease to exist at $(s,t) = (0.5, 0.05)$. Beyond, for $t > 0.05$ the curve passes through itself and its interpretation as an interface becomes uncertain.
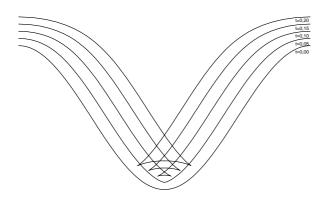
Figure 2: Curve evolution(2.2), $s \in [0, 1]$

In order to solve this problem an entropy condition is invoked, i.e. a property based in the knowledge of the "order" that the particles in the front should have:

Definition 1 (Entropy condition [5]) *In the process of inward deformation, once a point is dislodged from the inner part of the curve, it remains disjoint from it forever. Similarly, in the process of outward deformation, once a point becomes part of the interior of the curve, it remains a part of it forever.*

Following [10] choosing a speed function $F(k) = 1 - \epsilon k$, $\epsilon \in \mathbb{R}$, $\epsilon > 0$ (viscosity solution) makes the curve evolution respect the *entropy condition*, thus the curve will remain smooth and the constant speed solution is obtained when $\epsilon \to 0$. Replacing this function in the equation (2):

$$\phi_t + (1 - \epsilon k)|\nabla \phi| = 0$$

then

$$\phi_t + H_0(\phi_x, \phi_y, t) =$$
$$= \epsilon H_1(\phi_{xx}, \phi_{yy}, \phi_{xy}, \phi_x, \phi_y, t)$$

where

$$H_0(\phi_x, \phi_y, t) = \sqrt{\phi_x^2 + \phi_y^2}$$
$$H_1(\phi_{xx}, \phi_{yy}, \phi_{xy}, \phi_x, \phi_y, t) =$$
$$\frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{\frac{3}{2}}}\sqrt{\phi_x^2 + \phi_y^2}$$

This is a Hamilton-Jacobi second order equation whose numerical solution can be found in [7].

Taking a curvature linear dependent velocity $F(k) = F_0 - F_1 k$, $F_0, F_1 \in \mathbb{R}$, the equation 2 can be rewritten:

$$\phi_t + (F_0 - F_1 k)|\nabla \phi| = 0$$

$$\phi_t + F_0|\nabla \phi| - F_1 k|\nabla \phi| = 0 \qquad (3)$$

The second term of the equation (3) is known as advection or constant movement term, whereas the thirds is known as diffusion or curve smoothing term (see [10]).

Conversely a discretization, where Δt is the temporal step, ϕ_{ij}^n is the value of the function ϕ at (i, j), in the moment n, is taken, namely.

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} + (F_0 \nabla_{ij} \phi_{ij}^n) - F_1 k(\nabla_{ij} \phi_{ij}^n) = 0.$$

The discretized advection term, $F_0(\nabla_{ij} \phi_{ij}^n)$ is computed with an *upwind scheme*

$$\left\{ (max(F_0, 0)\nabla_{ij}^+)^2 + (min(F_0, 0)\nabla_{ij}^-)^2 \right\}^{\frac{1}{2}}$$

where

$$\nabla_{ij}^+ = [max(D_{ij}^{-x}, 0)^2 + min(D_{ij}^{+x}, 0)^2$$
$$+ max(D_{ij}^{-y}, 0)^2 + min(D_{ij}^{+y}, 0)^2]$$
$$\nabla_{ij}^- = [max(D_{ij}^{+x}, 0)^2 + min(D_{ij}^{-x}, 0)^2$$
$$+ max(D_{ij}^{+y}, 0)^2 + min(D_{ij}^{-y}, 0)^2].$$

The foward and backward finite difference approximation to the spatial derivatives over x, D_{ij}^{+x} and D_{ij}^{-x} are stated as

$$D_{ij}^{+x} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}$$
$$D_{ij}^{-x} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}$$

where Δx is the spatial step over x; the derivatives over y are analogously defined.

The diffusion term $F_1 k|\nabla \phi|$ is discretized using central differences

$$F_1 k\sqrt{(D_{ij}^x)^2 + (D_{ij}^x)^2}$$

where the approximation to the derivative of ϕ over x is stated as:

$$D_{ij}^x = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}$$

and the derivative over y is analogously defined. Finally the curvature k defined using the mean curvature:

$$k = \frac{\phi_{xx}\phi_x^2 - \phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_y^2}{(\phi_x^2 + \phi_y^2)^{\frac{3}{2}}}$$

is discretized using central finite differences for first and second derivatives:

$$k_{ij} = \frac{D_{ij}^{xx}(D_{ij}^x)^2 - D_{ij}^{xy}D_{ij}^x D_{ij}^y + D_{ij}^{yy}(D_{ij}^y)^2}{((D_{ij}^x)^2 + (D_{ij}^y)^2)^{\frac{3}{2}}}.$$

where the central approximation to the second derivative over x is defined as

$$D_{ij}^{xx} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta x^2}$$

and the approximations to the second derivatives over y and over x and y are defined analogously.

The third advantage of this formulation is that the intrinsic geometric properties of the front may be easily determined from the level set function ϕ, (i.e, curvature, enclosed area, etc.).

Finally the fourth advantage is the easiness to generalize this approach to higher dimensional models by changing ϕ's domain.

2.3 Boundary detection using front propagation

In this section a description of the level set technique application for boundary detection is given. In the first place the interface is taken as representing the boundary of a shape that evolves in time. Given that the goal is to detect several objects´ boundary in an image, the evolution of the interface must be forced to a halt when it reaches the boundary. Finally the contour of the object is defined as the set of points that belong to the interface when all of them have come to a halt.

The definition of a curve evolution speed function F such that the evolution stops when the curve reaches the boundary is the objective now. Thus the equation (3) is restated:

$$\phi_t(\vec{x}, t) + g(\vec{x})(F_0|\nabla\phi(\vec{x}, t)| - F_1 k(\vec{x}, t)|\nabla\phi(\vec{x}, t)|) = 0 \tag{4}$$

where $g(\vec{x})$ is a function that becomes zero as long as the interface approaches a border and is at the most one at any other point of the domain. If I is the image a classical choice for g is ([6])

$$g(\vec{x}) = \frac{1}{1 + |\nabla(G_\sigma * I(\vec{x}))|^2}$$

where $G_\sigma * I$ is the convolution of the image I with a filter G_σ which removes the image noise.

Following the revision stated in [11], the equation (4), has two segmentation problems: firstly g does not suffice to stop completely the interface at the boundary, and secondly if the interface crosses the border there is no term which can pull it back with enough strength for it to came back to the boundary. In [2] a new term is added which makes the algorithm more robust and solves this problems:

$$\phi_t(\vec{x}, t) + g(\vec{x})(F_0|\nabla\phi(\vec{x}, t)| - F_1 k(\vec{x}, t)|\nabla\phi(\vec{x}, t)|) + \nabla g(\vec{x}) \cdot \nabla\phi(\vec{x}, t) = 0$$

3 Boundary detection under Bspline Deformable Contours

In this section a contour extraction B-spline curve based method is developed. A brief theoretical review of B-spline representation and a boundary detection method are presented.

3.1 Bspline Representation

A brief theoretical review of B-spline representation of contours is presented; for more details see [1, 8].

Let $\{Q_0, ..., Q_{N_B-1}\}$ be a set of control points, where $Q_n = (x_n, y_n)^t \in \mathbb{R}^2$, $0 \leq n \leq N_B - 1$, and let $\{s_0 < s_1 < s_2 < \cdots < s_{L-1}\} \subset \mathbb{R}$ be a set of L knots. A B-spline curve of order d is defined as a weighted sum of N_B polynomial basis functions $\mathrm{B}_{n,d}(s)$ of degree $d-1$, within each interval $[s_i, s_{i+1}]$ with $0 \leq i \leq L-1$. The constructed spline function is $r(s) = (x(s), y(s))^t$, $0 \leq s \leq L - 1$, being

$$r(s) = \sum_{n=0}^{N_B-1} \mathrm{B}_{n,d}(s)Q_n,$$

and

$$x(s) = \mathrm{B}^t(s).Q^x \tag{5}$$
$$y(s) = \mathrm{B}^t(s).Q^y \tag{6}$$

where the basis functions vector $\mathrm{B}(s)$ of N_B components is given by $\mathrm{B}(s) = (\mathrm{B}_{0,d}(s), \ldots, \mathrm{B}_{N_B-1,d}(s))^t$. The weight vectors Q^x and Q^y give the first and second components of the Q_n, respectively.

The curves used in this work are closed of order $d = 3$ or $d = 4$ specified by periodic B-spline basis functions.

The problem of determining a polygon that generates a fitting B-spline curve with known number of control points, N_B, was studied by [8]. We now present a brief review of this subject.

A set of k data points in the image plane is given by $\{D_0, D_1, \ldots, D_{k-1}\}$, where $D_i = (x_i, y_i)^t$, $i = 0, \ldots, k-1$, and the spline curve that best-fits them is sought. Then, by equations 5 and 6, the components D_i must satisfy

$$x_i = \mathrm{B}^t(t_i)Q^x,$$
$$y_i = \mathrm{B}^t(t_i)Q^y,$$

for certain values of t_i, where $i = 0, \ldots, k - 1$, and $N_B \leq k$.

This linear system is more compactly written in matrix form as $D = K(Q^x \quad Q^y)$, where the $k \times N_B$ elements of the real matrix K are given by $K_{ij} = B_{j,d}(t_i)$, with $i = 0, \ldots, k-1$, $j = 0, \ldots, N_B - 1$, and $D = (D_0, D_1, \ldots, D_k)^t$. In the most general case $N_B < k$ and, therefore, K is not a square matrix. In this case, the pseudo-inverse matrix form $(Q^x \quad Q^y) = K^+ D$ is used to find the
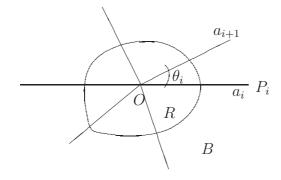
Figure 3: Radial straight lines a_i separated an angle θ_i.

B-spline fitting curve. A useful set of values for the parameters $\{t_0, \ldots, t_{k-1}\}$ is given by

$$t_0 = 0, \; ; t_\ell = \frac{\sum_{i=1}^{\ell} \|D_i - D_{i-1}\|}{\sum_{i=1}^{k-1} \|D_i - D_{i-1}\|}, \ell \geq 1.$$

The knot set to build the B-spline basis functions is arbitrarily chosen.

3.2 Boundary detection

In this section we describe an algorithm developed for boundary detection using B-spline deformable contours. Let E be a scene made up by the background B and a region R with its respective boundaries ∂R. We want to find the curve C_B that fits boundary ∂R in the image. We define an initial search area, which is specified by polygons, the vertexes of which are control points that generate a B-spline curve. Once the initial search zone is determined the centroid is calculated.

If a point belongs to the object boundary, then a sample taken from the neighborhood of that point exhibits a change in the intensity level of image and it is considered to be a transition point. Then N segments a_i, $i \in \{1, ..., N\}$ with the form $a_i = \overline{OP_i}$ are considered. Here O is the centroid of the initial region, the extreme P_i is a point outside of the region and $\theta_i = ang(a_i, a_{i+1}) \, \forall i$ is the angle between two consecutive segments as it can be seen in Figure 3. The segment a_i is an array of m elements coming from a discretization of the straight line on the image. In this work all the experimental results are binary images, so the border point on each segment is found convolving the data of the segment with a mask given by $[-1, 1]$. Once the set of border points $A = \{b_1, ..., b_N\}$ is found, the method builds the interpolating B-spline curve as its explained in section 3.1. This method can generate redundant contour points, and generally it does not need to use all of them for obtaining the interpolating curve. An algorithm to find significant boundary points in the description of the same

curve, is now presented. Given the set of border points A, the points with maximum curvature (critical points) are chosen. This method was proposed by P. Zhu (see [12]) in order to detect critical points in A. Let D be the set of critical points of A and E the set of redundant points of A, then

$$D \cap E = \emptyset$$
$$D \cup E = A$$

We consider the polar coordinates of each point $b_i \in A$:

$$b_i = (\rho_i, \alpha_i)$$

where ρ_i and α_i are the length and the orientation of b_i, respectively. A is an ordering sequence, then each point b_i has only two geometrical neighbors b_{i-1} and b_{i+1}.

If b_{i-1}, b_i and b_{i+1} are consecutive points then there is a critical point if

$$sign(\rho_i - \rho_{i-1}) \neq sign(\rho_{i+1} - \rho_i)$$

or

$$sign(\alpha_i - \alpha_{i-1}) \neq sign(\alpha_{i+1} - \alpha_i)$$

hold. In this manner, a subset

$$M = \{b_{c_1}, \ldots, b_{c_m}\}$$

of critical points is obtained. Then, for each pair of adyacent points $b_{c_j}, b_{c_{j+1}}$ of M, an interpolating straight line S_j is considered. Then the distances d_i between $b_i \in A - M$ and S_j are computed. The points with maximum and minimum distances that verify the following condition:

$$sign(d_i - d_{i-1}) \neq sign(d_{i+1} - d_i)$$

for $c_j < i < c_{j+1}$, are selected.

We define $\Gamma = \left\{\tilde{b}_0, \tilde{b}_1, ..., \tilde{b}_{r-1}\right\}$ the set of r points that verify equation (3.2).

Let $C_0 = sort(M \cup \Gamma)$ be the ordering set of pseudo critical points. In order to select the critical points from C_0, we define a critical level l for each point b_i:

$$l(b_i) = area(b_{i-1}, \overset{\triangle}{b_i}, b_{i+1}).$$

Finally, taking the critical point set

$$D = \{b_i \in C_0 / l(b_i) \geq L\},$$

where L is a threshold defined as $L = \min\left\{\frac{l(b_i)}{3}, b_i \in A\right\}$, a new interpolating B-spline curve is obtained.
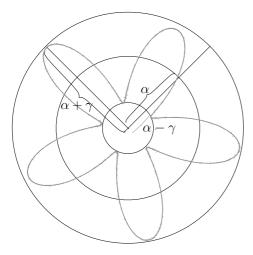
Figure 4: Parameters α and γ as seen on the flower equation (7)

4 Results

This section presents the obtained results of the segmentation methods applied to images which are randomly generated. The approximation error for both methods, B-spline and level set approach is characterized and compared.

The algorithms are applied to 500 synthetic images (flowers) generated by means of the parametric curve in polar coordinates:

$$f(s; \alpha, \beta, \gamma) = (\theta(s), \rho(s; \alpha, \beta, \gamma)), \ s \in [0, S]$$
$$\theta(s) = s \frac{2\pi}{S} \qquad (7)$$
$$\rho(s; \alpha, \beta, \gamma) = \alpha - \gamma cos(\theta(s)\beta)$$

where α is the mean radius of the flower, β is the number of petals and 2γ is the height of the petals, see Figure 4. α, β and γ are considered independent random variables with uniform distribution. The variation ranges for each variable are given by: $\alpha \in [5, 20]$, $\beta \in [15, 50]$, $\gamma \in [2, 10]$.

In order to compute the approximation error, ∂R the boundary of the region of interest and C the approximating curve are considered. Let a_j be the jth radial straight line, see Figure 3, which equation is given by:

$$a_j : \lambda \vec{u}_j + O \quad j = 1, \ldots, N$$

where $\vec{u}_j$ is the straight line unit vector and O is the centroid of the region R. Let $\vec{V}_j$ the intersection point between the curve C and the straight line a_j, and $\vec{W}_j$ the intersection point between the boundary ∂R and the straight line a_j:

$$a_j \cap C \quad = \vec{V}_j$$
$$a_j \cap \partial R \quad = \vec{W}_j$$

Then, a distance between C and ∂R is defined as

$$d(\partial R, C) = \frac{1}{N} \sqrt{\sum_{j=1}^{N} \left\| \vec{V}_j - \vec{W}_j \right\|^2} \qquad (8)$$

Figure 5 (bs1) to (bs4) shows boundary detection with B-spline method and (ls1) to (ls4) boundary detection with level set method, in both cases as a white curve. The two methods applied to four synthetic images generated from the equation (7). Shapes from (bs1) and (ls1) generated with $\alpha = 10$, $\beta = 38$ and $\gamma = 5$, (bs2) and (ls2) with $\alpha = 6$, $\beta = 50$ and $\gamma = 10$, (bs3) and (ls3) with $\alpha = 8$, $\beta = 38$ and $\gamma = 9$ and (bs4) and (ls4) with $\alpha = 12$, $\beta = 15$ and $\gamma = 3$. Figure 6 shows the histograms of the B-spline and level set approximation error.

5 Conclusions and Further Work

Comparing the exhibited results in Section 4 from a visual perspective the segmentation achieved with level sets method is more accurate than the one achieved with B-splines. Furthermore, using the measure defined in (8) over 500 generated samples, we noticed that more than 50% of the segmented shapes using the level sets method, presented an error less or equal than 0.05 and more than 85% an error less or equal than 0.1. On the other hand, less than 20% of the samples segmented using the B-spline technique presented an error lower than 0.1 and less than 84% were segmented with an error lower than 0.3. These results can be seen in table 5, corresponding to the accumulated frequencies of the approximation errors of the B-spline and level set methods, respectively. This fact can also be noticed on the error histograms shown in Figure 6.

In this work we established a comparing method for segmentation techniques. Finally, as a conclusion of this work we present in a comparative manner the advantages and drawbacks of each segmentation method compared:

- Level set method

  1. Boundary fitting accuracy depends only on the ratio between the advection and diffusion parameters. Furthermore, Eulerian evolution instead of Lagrangian implies higher accuracy on the differential equation resolution, hence better fitting results are achieved.

  2. Front topology can change during the evolution, thus supervised initialization is not required.

  3. The result is a discretization over the domain of a function whose zero level set is the boundary. Therefore, in order to obtain a continuous representation of the boundary it is necessary to extract the isocontour and interpolate it.

(bs1)   (ls1)

(bs2)   (ls2)

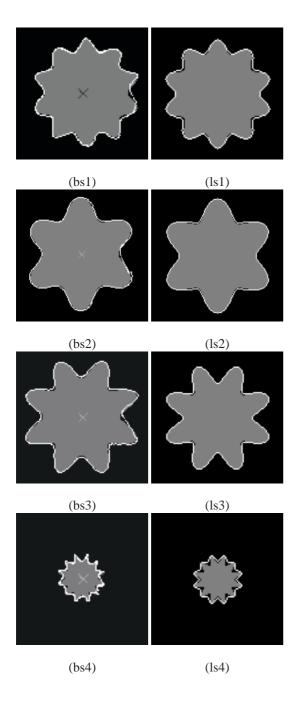(bs3)   (ls3)

(bs4)   (ls4)

Figure 5: The B-spline approximation curve (bs1) to (bs4). Level set approximation curve (ls1) to (ls4). (bs1) and (ls1) generated with $\alpha = 10$, $\beta = 38$ and $\gamma = 5$, (bs2) and (ls2) with $\alpha = 6$, $\beta = 50$ and $\gamma = 10$, (bs3) and (ls3) with $\alpha = 8$, $\beta = 38$ and $\gamma = 9$ and (bs4) and (ls4) with $\alpha = 12$, $\beta = 15$ and $\gamma = 3$

4. The smoothing level of corners depends on the diffusion parameter and can be omitted. Then, this approach can handle sharp edges effortlessly.
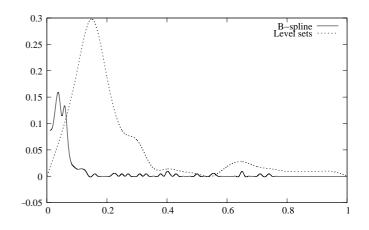

Figure 6: Histogram of the approximation errors for the B-spline and level set methods.

5. High computational cost.

6. Easy to generalize to 3D and more dimensional models. The mathematical formulation has a direct implementation.

- B-spline

1. The number of control point and the B-spline order have to be known a priori and determines the accuracy of the results.

2. It is mandatory to specify an initial area of interest, often in a supervised manner, for each contour in the image. In addition, the topology of the solution must be known beforehand.

3. The result is a parametric equation of the boundary determinated by a set of control points.

4. Smoothes out corners.

5. Low computational cost.

Finally, as it was exposed previously in this section, the level set method is more accurate than the B-spline method.

As further work we propose to improve the error characterization of both methods by performing this analysis on samples of synthetic and real textured images, hence partitioning the sample sets according to the geometrical properties of the shapes. In order to decide which method is more convenient for a particular problem, a statistical description of the approximation error for each method will be studied.

References

[1] A. Blake and M. Isard. *Active Contours*. Springer Verlag, 1998.

d	method	
	B-spline	Level set
0.05	0	0.592
0.1	0.197	0.853
0.15	0.495	0.890
0.2	0.683	0.895
0.25	0.770	0.908
0.3	0.839	0.913
0.35	0.858	0.922
0.4	0.872	0.936
0.45	0.881	0.945
0.5	0.885	0.950
0.55	0.885	0.954
0.6	0.904	0.959
0.65	0.931	0.968
0.7	0.950	0.968
0.75	0.963	0.977
0.8	0.972	0.977
0.85	0.982	0.977
0.9	0.991	0.977
0.95	1	0.977
≥ 1	1	1

Table 1: algo

[2] Vincent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *ICCV*, pages 694–699, 1995.

[3] do Carmo. *Geometría diferencial de curvas y superficies*. Alianza Editorial, 1990.

[4] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. of IEEE Conference on Computer Vision*, pages 259–268, London, England, 8-11 1987.

[5] B. Kimia, A. Tannebaum, and S. Zucker. Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space, 1995.

[6] Ravi Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[7] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[8] D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, New York, USA, 2 edition, 1990.

[9] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, second edition, 1999.

[10] J.A. Sethian. Curvature and the evolution of fronts. *Comm. in Math. Phys.*, (101):487–499, 1985.

[11] J. Suri, K. Liu, S. Singh, S. Laxminarayana, and L. Reden. Shape recovery algorithms using level sets in 2-d/3-d medical imagery: A state-of-the-art review, 2001.

[12] P. Zhu and P.M. Chirlian. On critical point detection of digital shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:737–748, 1995.