International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(4): 209-222
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

Maximum Likelihood Curves for Multiple Objects Extraction: Application to Radiographic Inspection for Weld Defects Detection

Aicha Baya Goumeidane[1], Mohammed Khamadja[2] and Nafaa Nacereddine[1]
[1] Centre de Recherche Scientifique et Technique en Soudage et Contrôle (CSC), Chéraga, Algiers, Algeria
[2] SP_Lab, Departement d'Eléctronique, Université de Mentouri, Constantine, algeria
ab_goumeidane@yahoo.fr, m_khamadja@yahoo.fr, nafaa.nacereddine@enp.edu.dz

ABSTRACT

This paper presents an adaptive probabilistic region-based deformable model using an explicit representation that aims to extract automatically defects from a radiographic film. To deal with the height computation cost of such model, an adaptive polygonal representation is used and the search space for the greedy-based model evolution is reduced. Furthermore, we adapt this explicit model to handle topological changes in presence of multiple defects.

KEYWORDS

Explicit deformable model, adaptive contour representation, Maximum likelihood criterion.

1 INTRODUCTION

Radiography is one of the old and still effective Non Destructive Testing tools. X-rays penetrate welded target and produce a "shadow picture" of the internal structure of the target [1]. Automatic detection of weld defect is thus a difficult task because of the poor image quality of industrial radiographic images, the bad contrast, the noise and the low defects dimensions. Moreover, the perfect knowledge of defects shapes and their locations is critical for the appreciation of the welding quality. For that purpose, image segmentation is applied. It allows the initial separation of regions of interest which are subsequently classified.

Among the boundary extraction-based segmentation techniques, deformable model also called active contours or snakes are recognized to be one of the efficient tools for 2D/3D image segmentation [2]. Broadly speaking a snake is a curve which evolves (under the influence of internal forces going from within the curves itself and external forces computed from the image data) to match the contour of an object in the image. The bulk of the existing works in segmentation using active contours can be categorized into two basic approaches: edge-based approaches, and region-based ones. The edge-based approaches are called so because the information used to drawn the curves to the edges is strictly along the boundary. Hence, a strong edge must be detected in order to drive the snake. This obviously causes poor performance of the snake in weak gradient fields. That is, these approaches fail in the presence of noise. Several improvements have been proposed to overcome these limitations but still they fail in numerous cases [3][4][5][6][7][8][9] [10][11]. With the region-based ones [12][13][14][15] [16][17][18][19] [20], the inner and the outer region defined by the snake are considered and, thus, they are well-adapted to situation for which it is difficult to extract boundaries from the target. We can note that such methods are computationally intensive since the computations are made over a region [18][19].

This paper deals with the detection of multiple defects in radiographic films, and presents a new region based snake which exploits a statistical formulation and an adaptive active contour nodes representation are used. Section 2 details the mathematical formulation of the method. Section 3 is devoted to the development of the proposed progression strategy of our model. In section 4 we show how we adapt the model to the topology in presence of multiple defects. Results are shows in Section 5.We draw the main conclusions in section 6.

2 PROBABILISTIC DEFORMABLE MODEL

2.1 Statistical Image Model

Let $C = \{c_0, c_1, ..., c_{N-1}\}$ be the boundary of a connected image region R_1 of the plane and R_2 the points that do not belong to R_1. if x_i is the gray-level value observed at the i^{th} pixel, $X = \{x_i\}$ the pixel grey levels, p_x the grey level density, and $\phi_x = \{\phi_1, \phi_2\}$ the density parameters (i.e., $p(x_i) = p(x_i|\phi_1)$ for $i \in R_1$ and $p(x_i) = p(x_i|\phi_2)$ for $i \in R_2$). The simplest possible region based model is characterized by the following hypothesis: conditional independence (given the region contour, all the pixels are independent); and region homogeneity, i.e., all the pixels in the inner (outer) region have identical distributions characterized by the same φ_x. Thus the likelihood function can be written as done in [13] [14]

$$p(X / C, \phi_x) = \prod_{i \in R} p(x_i / \phi_1) \times \prod_{i \in R_2} p(x_i / \phi_2) \quad (1)$$

$p(x_i / \phi_1)$ and $p(x_i / \phi_2)$ stand for the pixel-wise conditional probabilities, of the inner and outer regions, respectively.

2.2 Probabilistic approach to Contour Progression

The purpose being the estimation of the contour C of the region R_1 with K snake nodes, then this can be done by exploiting the presented image model by using the MAP estimation since:

$$p(C|X) = p(C)p(X|C) \quad (2)$$

and then

$$\hat{C}_{MAP} = \arg\max_{C} p(C) p(X / C) \quad (3)$$

Since we assume there is no shape prior and no constraints are applied to the model, then $p(C)$ can be considered as uniform constant and then removed from the estimation. Moreover Model image parameters must be added in the estimation, then:

$$\hat{C}_{MAP} = \arg\max_{C} p(X / C) \quad (4)$$

$$\hat{C}_{MAP} = \arg\max_{C} p(X / C, \phi_x) = \hat{C}_{ML} \quad (5)$$

Hence the MAP estimation is reduced to ML (Maximum likelihood) one. Estimating C implies also the estimation of the parameter model ϕ_x. Under the maximum likelihood criterion, the best estimates of ϕ_x and C denoted by $\hat{\phi}_x$ and $\hat{C}$ are given by:

$$(\hat{C}, \hat{\phi}_x) = \arg\max_{C, \phi_x} \log p(X / C, \phi_x) \quad (6)$$

The log function is included as it allows some formal simplification without affecting the location of the maximum. Since solving (6) simultaneously with respect to C and ϕ_x would be computationally very difficult, then an iterative scheme is used to solve the equation:

$$\hat{C}^{t+1} = \arg\max_{C} \log p(X / C, \phi^{t}{}_{x}) \quad (7)$$

$$\hat{\phi}^{t+1} = \arg\max_{C} \log p(X / \hat{C}^{t+1}, \phi_{x}) \quad (8)$$

Where $\hat{\phi}_x^t$ and $\hat{C}^t$ are the ML estimates of C and ϕ_x respectively at the iteration t.

2.3 Greedy Progression

The implementation of the deformable model evolution (according to(7)) uses the greedy strategy, which evolves the curve parameters in an iterative manner by local neighborhood search around snake points to select new ones which maximize $\log p(X / C, \phi^{t}{}_{x})$. The used neighborhood is the set of the eight nearest pixels.

.

3 SPEEDING THE PROGRESSION

Two strategies have been associated to improve the evolution velocity of the model and make it faster then the original model. Moreover, the convergence criterion has been modified such as the convergence of each cycle is reached quickly.

3.1 Search Space Reducing and Normal Evolution

We first choose to change the search strategy of the pixels being candidates to maximize $\log p(X / C, \phi^{t}{}_{x})$. For each snake node, instead of searching the new position of this node among the 8-neighborhood positions, the space search is reduced from 1 to 1/4 by limiting the search to the two pixels laying in normal directions of snake curve at this node as shown in Figure.1.

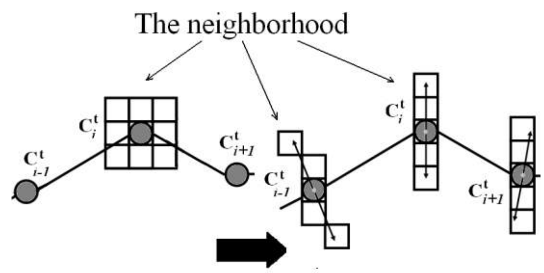

Figure 1. The new neighborhood: from the eight nearest pixels to the four nearest pixels in the normal directions

3.2 Polygonal Representation and Adaptive Segments Length

An obvious reason for choosing the polygonal representation is for the simplicity of its implementation. Another advantage of this description is when a node is moved; the deformation of the shape is local. Moreover, it could describe all shapes when a large number of nodes are used. However increase the nodes number will decrease the computation speed. To improve progression velocity, nodes number increases gradually along the snake evolution iterations through an insertion/deletion procedure. Indeed, initialization is done with few points and when the evolution stops, points are added between the existing points to launch the evolution, whereas other points are removed.

3.2.1 Deletion and Insertion Processes.

The model progression will be achieved through cycles, where the model nodes number grows with an insertion/deletion procedure. In the cycle 0, the contour initialization begins with few points. Thus, solving (7) is done quickly and permits to have an approximating segmentation of the object as this first contour converges.

In the next cycle, points are added between initial nodes and a mean length *MeanS* of obtained segments is computed. As the curve progresses towards its next final step, the maximum length allowed will be related to *MeanS* so that if two successive points c_i and c_{i+1} move away more than this length, a new point is inserted and then the segment $[c_i\ c_{i+1}]$ is divided as shown in figure 2.

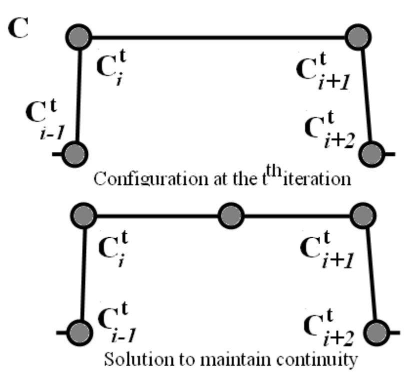

Figure 2. Regularization procedure: maintaining the continuity by adding nodes

On the other hand, if the distance of two consecutive points is less than a defined threshold (*TH*), these two points are merged into one point placed in the middle of the segment $[c_i\ c_{i+1}]$ as illustrated in figure 3. Moreover, to prevent undesired behavior of the contour, like self intersections of adjacent segments, every three consecutive points c_{i-1}, c_i, c_{i+1} are checked, and if the nodes c_{i-1} and c_{i+1} are closer than *MeanS/*2, c_i is removed (the two segments are merged) as illustrated in Figure.4. This can be assimilated to a regularization process to maintain curve continuity and prevent overshooting.

When convergence is achieved again (the progression stops) new points are added and a new *MeanS* is computed. A new cycle can begin. The process is repeated until no progression is noted after a new cycle is begun or no more points could be added. This is achieved when the distance between every two consecutive points is less then the threshold *TH*. Here, the end of the final cycle is reached.

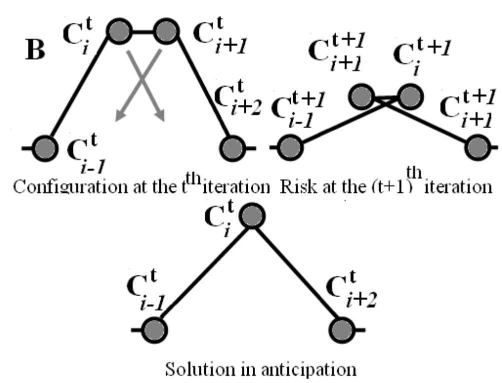

Figure 3. Regularization procedure: Avoiding overshooting by merging nodes

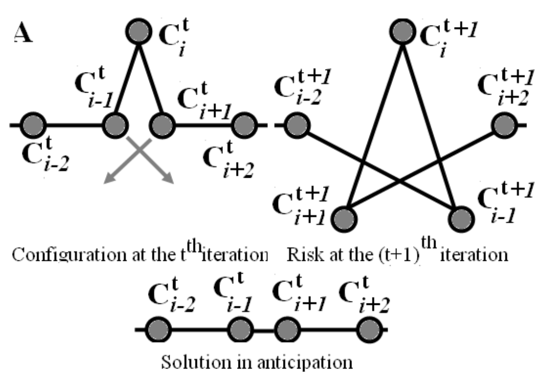

Figure 4. Regularization procedure: Avoiding overshooting by merging segments

3.3 Convergence criterion

In the fast greedy algorithm presented above, a cycle's iteration will be stopped if the convergence criterion of the process is achieved. This criterion consists of a combination of two criterions. Suppose that $A^{(k,S)}$ is the area delimited by the model at the k^{th} iteration of a cycle S. Then, the criterion $AC_{(k,S)}$ is defined as follows:

$$AC_{(k,S)} = \frac{\left|A^{(k-1,S)} - A^{(k,S)}\right|}{A^{(k,S)}} \tag{9}$$

Working alone, this criterion is not self-satisfied thus we add to it an other one which is related to the length of the snake cord length Suppose that $L^{(k,S)}$ is the length of the snake cord at the k^{th} iteration of the S^{th} cycle, then second criterion $LC_{(k,S)}$ is defined as follows:

$$LC_{(k,S)} = \frac{\left|L^{(k-1,S)} - L^{(k,S)}\right|}{L^{(k,S)}} \tag{10}$$

In this method, theses two criterions are used instead of the ML value's one because they have exhibit quickest convergence than the ML variation which continues to grow even if the snake is doing very little jumping around its real final contours. This is mainly due to the digital nature of the image.

3.4 Algorithms

Since the method kernel is the Maximum Likelihood (ML) estimation of the model nodes by optimizing the search strategy (reducing the neighborhood), we begin by presenting the algorithm related to the ML criterion, we have named **AlgotithmML**. Next to this algorithm we present the algorithm of the regularization we have just named **Regularization**. These two algorithms will be used by the algorithm which describes the evolution of the snake over a cycle. We have called this algorithm **AlgorithmCycle**. The overall method algorithm named **OverallAlgo** is given after the three quoted algorithms. For all these algorithms *MeanS* and *TH* are the mean segment length and the threshold shown in the section 3.2, α is a constant related to the continuity maintenance of the snake model. ε is the convergence threshold, A_C and L_C are the area delimited by the polygon C and the cord length of this polygon respectively

3.3.1 Algorithm 1. *AlgorithmML*

input: M nodes $C = [c0, c1, . . . cM-1]$,

output: C_{ML}, A_{CML}, L_{CML}

Begin;

Step0: Estimate $\phi_x(\phi_1, \phi_2)$ inside and outside C;

Step1: Update the polygon according to:

$$C_j^{ML} = \underset{nj \in N(c_j)}{\arg\max} \log p(X / [c_1, c_2, .., n_j, .., c_M], \phi_x)$$

$N(c_j)$ is the set of the four nearest pixels laying in the normal direction of c_j. This will be repeated for all the polygon points;

Step2: Estimate ϕ_x^{ML} for C_{ML} and A_{CML} *and* L_{CML} *as:*
A_{CML} *is* the pixel number inside the polygon C_{ML} *and* A_{CML} *is the polygon* (C_{ML}) *cord length of the*

End

3.3.2 Algorithm 2. *Regularization*

input : M nodes $C = [c0, c1, . . . , c_{M-1}]$, *MeanS*, *TH*, α

output: C^{Reg}

Begin;

Step0: Compute the M segments length: *S lenght*(*i*) ;

Step1:
for *all i (i=1,...,M)* **do**
 if *S length*(*i*) < *TH* **then**
  Remove *ci* and *ci*+1 and replace them by a new one in the middle of $[c_i c_{i+1}]$
 end
 if *S length*(*i*) > $\alpha *$ *MeanS* **then**
  insert a node in the middle of $[c_i c_{i+1}]$
 end
end

Step 2 :
for *all triplet*(c_{i-1}, c_i, c_{i+1}) **do**
 if *ci*−1 *and ci*+1 *are closer than MeanS/*2 **then**
  Remove c_i
 end
end
End

3.3.3 Algorithm 3. *AlgorithmCycle*

input : Initial nodes $C_{cy}^{0} = \left[C_{cy,0}^{0}, C_{cy,2}^{0}, ..., C_{cy,N-1}^{0}\right]$, *MeanS*, *TH*, α, ε

output: The estimate $\hat{C}_{cy}$, and $A_{\hat{C}_{cy}}$ and $L_{\hat{C}_{cy}}$ of the current cycle

Begin

Step0: Set t = 0 (iteration counter) and $C_{cy}^{t} = C_{cy}^{0}$
Compute *MeanS* of the *N* initial segments

Step1: Estimate $\phi_{x,cy}^{t}(\phi_1, \phi_2)$ inside and outside C_{cy}^{t}
Compute $A_{\hat{C}_{cy}^{t}}$ and $L_{\hat{C}_{cy}^{t}}$ of C_{cy}^{t}
$L_{C1}=L_{\hat{C}_{cy}^{t}}$ and $A_{C1}=A_{\hat{C}_{cy}^{t}}$
Perform **AlgorithmML**(C_{cy}^{t})

Step2 : Recover A_{CML}, L_{CML} and C_{ML}
$A_{C2} = A_{CML}$, $L_{C2}=L_{CML}$ $C_{cy}^{t+1} = C_{ML}$
Perform **Regularization**(C_{cy}^{t+1}, *MeanS*, *TH*, α)
Recover C^{Reg}
If($|AC1 - AC2|)/AC1 > \varepsilon$ *and* $(|LC1 - LC2|)/LC1 > \varepsilon$ **then**
 $C_{cy}^{t} = C^{Reg}$
  go to **step 1**
else
 $\hat{C}_{cy} = C_{cy}^{t}$, $A_{\hat{C}_{cy}} = A_{C2}$, $L_{\hat{C}_{cy}} = L_{C2}$
  go to end
end
End

3.3.4 Algorithm 4. *OverallAlgo*

input : Initial nodes C^0, *MeanS*, *TH*, α, ε

output: Final contour $\hat{C}$

Begin

Step0: Compute *MeanS* of the all segments of C^0

Step1:Perform **AlgorithmCycle**(C^0, ε, *TH*, α, *MeanS*)

Step2: Recover $A_{\hat{C}_{cy}}$, $L_{\hat{C}_{cy}}$ and the model nodes $\hat{C}_{cy}$

Step3:Insert new nodes to launch the evolution
 if *no node can be inserted* **then**
  $\hat{C} = \hat{C}_{cy}$
  Go to **End**
 end

Step4:Creation of C^{New} because of the step 3

Step5:Perform **AlgorithmML**(C^{New})
Recover A_{CML}, L_{CML}, Recover C_{ML}
 If $(|A_{CML} - A_{\hat{C}_{cy}}|)/A_{\hat{C}_{cy}} < \varepsilon$ *and* $(|L_{CML} - L_{\hat{C}_{cy}}|)/L_{\hat{C}_{cy}} < \varepsilon$ **then**
  $\hat{C} = \hat{C}_{cy}$

```
  go to End
 end

Step6:C⁰ = C^ML
Go to step 1
End
```

4. HANDELING THE TOPOLOGYCAL CHANGES

The proposed adaptive deformable model can be used to represent the contour of a single defect. However, if there is more than one defect in the image, this model can behave so that it splits, handles the topological changes and determines the corresponding contour of each defect. We will describe here the determination of critical points where the model is split for multiple defect representation.

The validity of each contour will be verified so that invalid contour will be removed.

4.1 The Model Behavior in the Presence of Multiple Defects

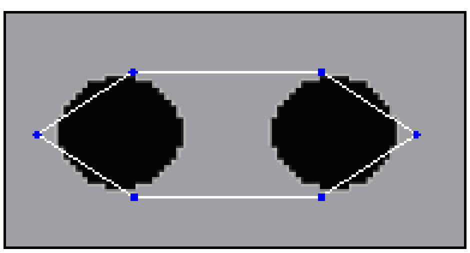

Figure 5. The model initialization for two objects

In presence of multiple defects as in figure 5, the model curve will try to surround all these defects. From this will result one or more self intersections of the curve, depending of the number of the defects and their positions with respect to the initial contour. The critical points where the curve is split are the self intersection points. The apparition of self intersection implies the creation of loops which are considered as valid if they are not empty. It is known that an explicit deformable model is represented by a chain of ordered points. Then, if self intersections occur, their points are inserted in the snake nodes chain first and then, are stored in a vector named *Vip* in the order they appear by running through the nodes chain. Obviously each intersection point will appear twice in this new chain. For convenience, we define a loop as a points chain which starts and finishes with the same intersection point without encountering another intersection point. After a loop is detected, isolated and its validity is checked, then, the corresponding intersection point is removed from *Vip* and thus can be considered as an ordinary point in the remaining curve. This will permit to detect loops born from two or more self intersections.

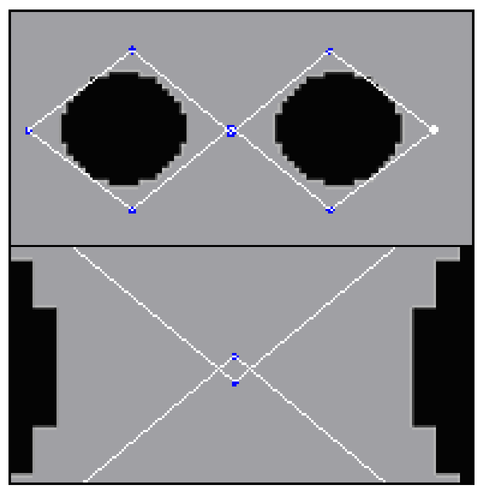

Figure 6. At the top self intersection of the polygonal curve, at down zoomed self intersections

This can be explained from an example: Let $C_n = \{c_1, c_2, ..., c_n\}$, with n=8, be the nodes chain of the curve shown in the Figure 6, with c_1 as the first node (in white in the figure). These nodes are taken in the clock-wise order in the figure. This curve, which represents our snake model, has undergone two self

intersections, represented by the points we named c_{int1} and c_{int2}, when it tries to surround the two disks. These two points are inserted in the chain nodes representing the model to form the new model points as following:
$C^{new}_n = \{c^{new}_1, c^{new}_2, ..., c^{new}_n\}$, with n=12 and $c^{new}_3=c_{int1}$, $c^{new}_5=c_{int2}$, $c^{new}_9=c_{int2}$, $c^{new}_{11}=c_{int1}$. After this modification, the vector *Vip* is formed as: $V\,ip=[c_{int1} c_{int2}\ c_{int2}\ c_{int1}]=[c^{new}_3\ c^{new}_5\ c^{new}_9\ c^{new}_{11}]$. Thus, by running through the snake nodes chain in the clock-wise sense, we will encounter *Vip*(1) then *Vip*(2) and so on...By applying the loop definition we have given, and just by examining *Vip*, the loops can be detected. Hence, the first detected loop is the one consisting of the nodes between *Vip*(2) and *Vip*(3). ie. $\{c^{new}_5, c^{new}_6\ c^{new}_8\}$, ($c^{new}_5$ being equal to c^{new}_9).

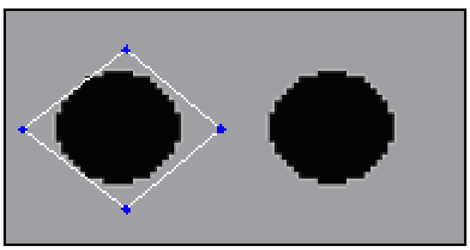

Figure 7. First detected loop

This first loop, shown on the Figure 7, is separated from the initial curve, its validity is checked (not empty) and c^{new}_5 , c^{new}_9 are deleted from *Vip* and then considered as ordinary nodes in the remaining curve. Now, *Vip* equals $[c^{new}_3\ c^{new}_{11}]$. Therefore, the next loop to be detected is made up of the nodes that are between c^{new}_3 and c^{new}_{11}. It should be noted that we have to choose the loop which do not contain previous detected loops nodes (except self-intersection's points).
In this case the new loop consists of the node's sequence $\{c^{new}_{11}, c^{new}_{12}, c^{new}_1, ..., c^{new}_2\}$ (c^{new}_3 being equal to c^{new}_{11}). This loop, which is also separated from the remaining snake curve, is illustrated in the Figure 8.

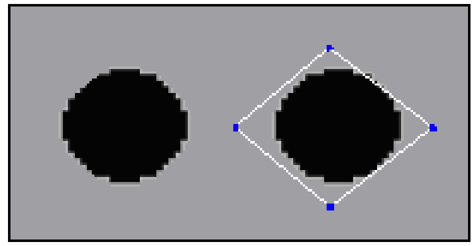

Figure 8. Second detected loop

Once *Vip* is empty, we check the remaining nodes in the remaining snake curve. These nodes constitute also a loop as shown in Figure 9.

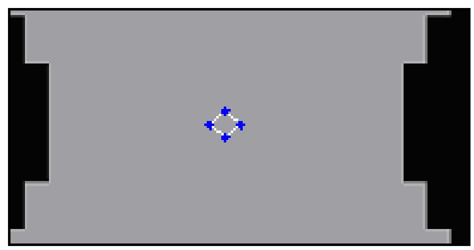

Figure 9. Third detected loop, it is empty and then it is an invalid one

To check the validity of a loop, we had just to see the characteristics of the outer region of the snake model at the first self intersections, like for example the mean or(and) the variance. If the inside region of the current loop have similar characteristics of the outside region of the overall polygonal curve at the first intersection (same characteristics of the background) then this loop is not valid and, it will be rejected. On the other hand, a loop which holds few pixels (a valid loop must contain a minimum number of pixels we have named *MinSize*) is also rejected because there are no weld defects that have such little sizes. The new obtained curves (detected valid loops) will be treated as independent ones, i.e. the algorithms quoted before are applied separately on each detected loop. Indeed, their progressions depend only on the object they contain.

5 RESULTS

The explicit deformable model we proposed is tested first on a synthetic image consisting of one complex object (Figure 10). This image is corrupted with a Gaussian distributed noise.

Fig. 10. The first synthetic test image

The image pixels grey levels are then modeled with a Gaussian distribution with mean and variance μ and σ respectively. The estimates of ϕ_x with x=1, 2 are the mean and the variance of pixels grey levels inside and outside the polygon representing the snake. The Gaussian noise parameters of this image are $\{\mu 1, \sigma 1\}$ = {70, 20} for the object and $\{\mu 2, \sigma 2\}$ = {140, 15} for the background.
First, we begin by focusing on the model behavior without regularization. Figure 11 gives an example of the absence of regularization procedures effect. Indeed, the creation of undesirable loops is then inescapable.
We show then, the evolution time we have gained when using both of a normal evolution for Maximum likelihood criterion evolution instead of the original one (choosing the next position among the eight nearest neighbors) and an adaptive model nodes number.

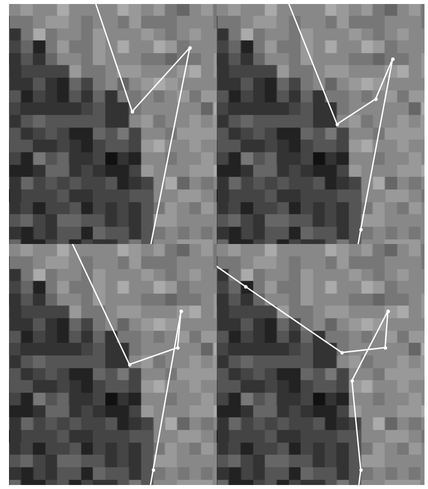

Figure. 11 Undesirable loops creation without regularization

To do this test, we begun by choose the model nodes as the regularly sampled points of the model (The nodes number depends only on the model cord length). In the figure below the initialization is done with a circle crossing the shape (Figure 12. A) where the re-sampling step equals to two pixels. The two models final results are shown after the initialization in figures 12.B and 12.C

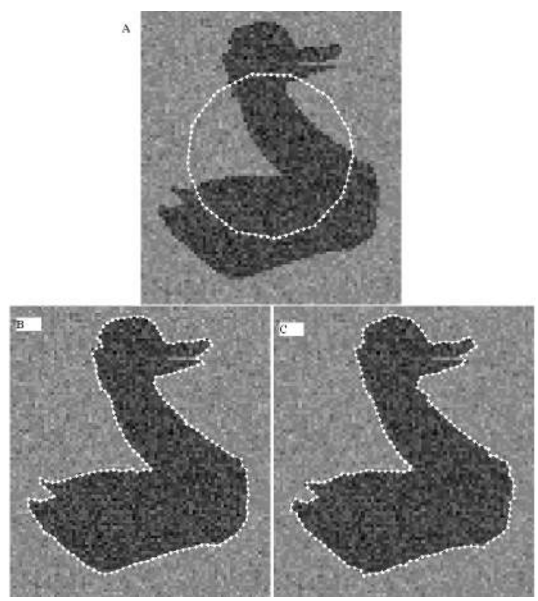

Figure. 12 A: initialization, B: final contour for the original evolution (choosing the next position among the eight nearest neighbors), C: final contour for a normal evolution

From the final contour point of view of, the two evolutions give approximately the same results. However, when we checked the execution time, we saw that the normal evolution algorithm permits to speed up the model progression. Indeed, beginning from the same initialization, the progression time to our proposed method final contour is four times less than the original one. Nevertheless, even the normal evolution has reduced the progression time, the progression remains slow because of the big nodes number of a regular re-sampled region-based model. We try to let the nodes number unchanged to speed up the progression. The execution time has dropped and unfortunately, the results quality also as shown in the figure 13.

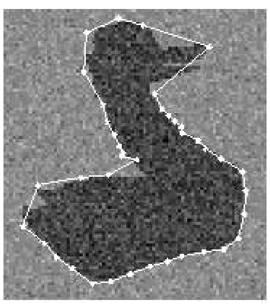

Figure. 13 Results of the normal evolution model with an unchanged model nodes number.

Now we show in figure 14 the behavior of the proposed model and what brings the association of the algorithms **AlgorithmML**, **AlgorithmCycle**, **Regularization** and **Algorithm** to the evolution on the same synthetic test image with the same initialization. We have noticed that the earlier iterations are the quickest ones because of low nodes number and it get slower in the latest iterations because of the need of a lot of points to describe the shape. The model can track concavities and although the noisy considered image, the object contour is correctly estimated and the execution time was 45% less then the case of regular sampled deformable model. In this application we choose α = 1.5, TH = 2, ε = 10−4. The computation time is

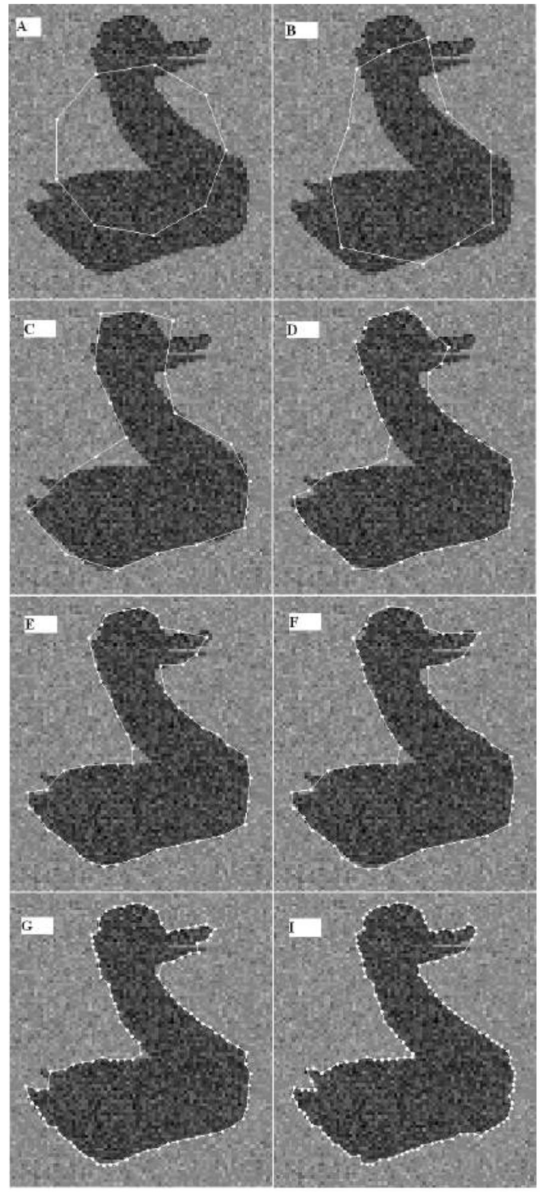

Figure 14. The proposed model progression in case of synthetic images, A: initialization, B, C, D, E, F, G, H: different intermediate results in the chronological order. I: final result

Furthermore, the model is tested on weld defect radiographic images containing one defect as shown in Figure 15. Because the industrial or medical radiographic images follow, in general, Gaussian distribution and that is due mainly to the differential absorption principle which governs the formation process of such images. The initial contours are sets of eight points describing circles crossing the defect in each image, the final ones match perfectly the defects boundaries.

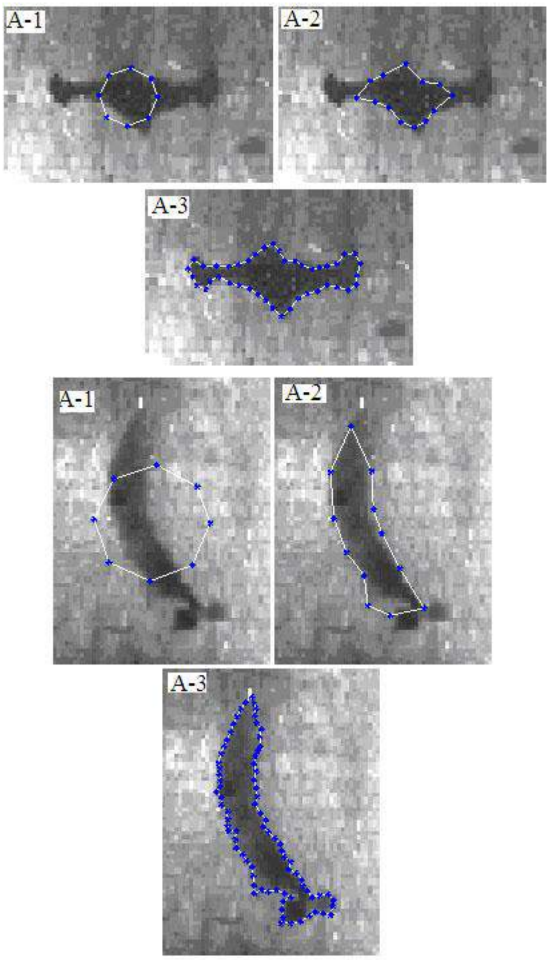

Figure 15: The proposed model progression in case of radiographic images: A1 initial contours, A2 intermediate contours, A3 final contours

After having tested the model behavior in presence of one defect, we show in the next three figures its capacity of handling topological changes in presence of multiple objects in the first next image (figure 16: Same characteristics as the figure 10) and multiple defects in the image (figure.17, figure.18).

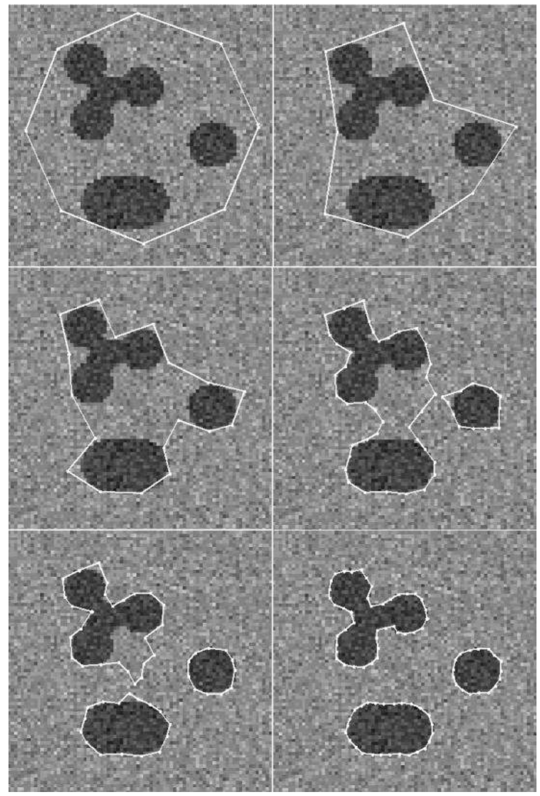

Figure 16 The model progression in presence of multiple different object in a synthetic image

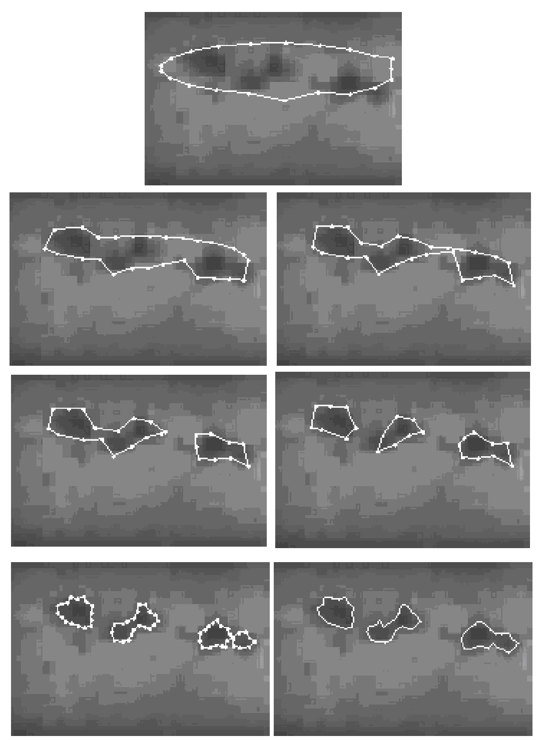

Figure 17 The proposed model progression in presence of multiple defects

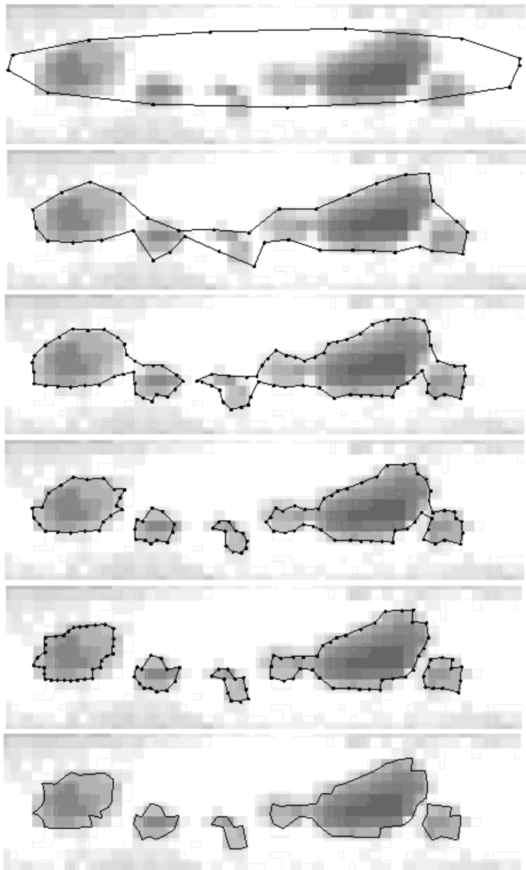

Figure 16 The proposed model progression in presence of multiple defects

The minimal size of an obect/defect is chosen to be equal to three pixels (*MinSize* = 3). The snake surrounds the objects/defects, splits and fits successfully their contours.

6. CONCLUSION

We have described a new approach of boundary extraction of weld defects in radiographic images. This approach is based on statistical formulation of contour estimation improved with the use of the combination of four algorithms to speed up the progression and increase in an adaptive way the model nodes number.
Moreover the proposed snake model can split successfully in presence of multiple contours and handle the topological changes. The performance of this method is confirmed by the experiments, on synthetic and radiographic images, which show the ability of the proposed method to give quickly a good estimation of the contours by fitting almost boundaries.

REFERENCES

1. Halmshaw, R.: The Grid: Introduction to the Non-Destructive Testing in Welded Joints. Woodhead Publishing, Cambridge (1996)
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. International Journal of Computer Vision, 321–331 (1988)
3. Xu, C., Prince, J.: Snakes, Shapes, and gradient vector flow. IEEE Transactions on Images Processing 7(3), 359–369 (1998)
4. Jacob, M., Blu, T., Unser, M.: Efficient energies and algorithms for parametric snakes. IEEE Trans. on Image Proc. 13(9), 1231–1244 (2004)
5. Tauber, C., Batatia, H., Morin, G., Ayache, A.: Robust b-spline snakes for ultrasound image segmentation. IEEE Computers in Cardiology 31, 25–28 (2004)
6. Zimmer, C., Olivo-Marin, J.C.: Coupled parametric active contours. IEEE Trans. Pattern Anal. Mach. Intell. 27(11), 1838–1842 (2005)
7. Srikrishnan, V., Chaudhuri, S., Roy, S.D., Sevcovic, D.: On Stabilisation of Parametric Active Contours. In: CVPR 2007, pp. 1–6 (2007)
8. Li, B., Acton, S.T.: Active Contour External Force Using Vector Field Convolution for Image Segmentation. IEEE Trans. on Image Processing 16(8), 2096–2106 (2007)
9. Li, B., Acton, S.T.: Automatic Active Model Initialization via Poisson Inverse Gradient. IEEE Trans. on Image Processing 17(8), 1406–1420 (2008)
10. Collewet, C.: Polar snakes: A fast and robust parametric active contour model. In: IEEE Int. Conf. on Image Processing, pp. 3013–3016 (2009)
11. Wang, Y., Liu, L., Zhang, H., Cao, Z., Lu, S.: Image Segmentation Using Active Contours With Normally Biased GVF External Force. IEEE signal Processing 17(10), 875–878 (2010)

12. Ronfard, R.: Region based strategies for active contour models. IJCV 13(2), 229–251 (1994)
13. Dias, J.M.B.: Adaptive bayesian contour estimation: A vector space representation approach. In: Hancock, E.R., Pelillo, M. (eds.) EMMCVPR 1999. LNCS, vol. 1654, pp. 157–173. Springer, Heidelberg (1999)
14. Jardim, S.M.G.V.B., Figuerido, M.A.T.: Segmentation of Fetal Ultrasound Images. Ultrasound in Med. & Biol. 31(2), 243–250 (2005)
15. Ivins, J., Porrill, J.: Active region models for segmenting medical images. In: Proceedings of the IEEE Internation Conference on Image Processing (1994)
16. Abd-Almageed, W., Smith, C.E.: Mixture models for dynamic statistical pressure snakes. In: IEEE International Conference on Pattern Recognition (2002)
17. Abd-Almageed, W., Ramadan, S., Smith, C.E.: Kernel Snakes: Non-parametric Active Contour Models. In: IEEE International Conference on Systems, Man and Cybernetics (2003)
18. Goumeidane, A.B., Khamadja, M., Naceredine, N.: Bayesian Pressure Snake for Weld Defect Detection. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 309–319. Springer, Heidelberg (2009)
19. Chesnaud, C., Réfrégier, P., Boulet, V.: Statistical Region Snake-Based Segmentation Adapted to Different Physical Noise Models. IEEE Transaction on PAMI 21(11), 1145–1157 (1999)
20. Nacereddine, N., Hammami, L., Ziou, D., Goumeidane, A.B.: Region-based active contour with adaptive B-spline. Application in radiographic weld inspection. Image Processing & Communications 15(1), 35–45 (2010)