2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 399-410

An Intelligent Visual-Based System for Object Inspection and Welding, relying on Active Contour Models-Algorithms

George Georgoulas[1], George Nikolakopoulos, Yiannis Koutroulis
Anthony Tzes and Peter Groumpos
Laboratory of Automation & Robotics
Electrical & Computer Engineering Department
University of Patras
Rio 26500
[1]georgoul@ee.upatras.gr

Abstract. In this paper a vision-based integrated system, relying on Active Contour Models (ACM) will be presented. ACM is a popular technique for extracting the object's boundaries based on the a priori knowledge of the object-shape. The algorithm is implemented in industrial parts of circular cross-section with undefined radius. The objective of ACM is to define the boundaries of the incoming part. Based on the coordinates of these boundaries, a proposed algorithm based on Least Squares calculates the center of the part. An x-y transposition table is responsible for moving the recognized part in a pre-defined place, where a welding-process will take place.

1 Introduction

The theoretical improvements in the field of digital image processing and the increase in the computational capabilities are creating new opportunities in industrial control applications [1]. The new methods [2] combined with classical techniques [3] can increase the computational intelligence, create more sophisticated machines and propose non-conventional solutions to many ill-defined problems. Machine vision is extremely useful in applications with dynamical events or continuously changing processes, adding robustness and flexibility. In fuzzy-like control problems, the conventional control strategies are stiff and cannot give a proper solution, as they must be re-designed to take account of every possible case.

In the rest of the paper we will present a new approach in controlling an industrial process of welding parts using the potential of machine vision, Figure 1. The proposed control system can take advantage of the ACM technology, presented in Section 2, by performing boundary detection in the parts that are going to be welded. The algorithm development and the test case are presented in Sections 4 and 5. Finally in Section 6, general conclusions and a summary of the obtained results are presented.

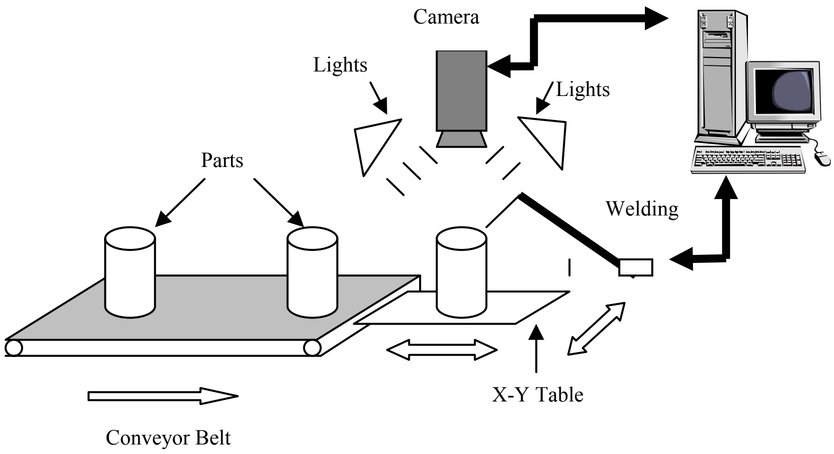

Fig. 1. The intelligent visual-based system for object inspection and welding

2 The Active Contour Models-Algorithms

Active Contour Models, also named "snakes", is a popular technique for extracting boundaries in a manner, which can surpass traditional edge detection approaches using prior knowledge of the object in search. The technique first introduced in 1987 [4] is built around an energy minimization framework. The snake's energy is the sum of internal energy E_{intern} and external energy E_{extern}. The latter consists of the energy E_{image} due to image forces and the energy E_{constr} due to external constraint forces, i.e.

$$E_{extern} = E_{image} + E_{constr}$$

A traditional snake is a curve v(s)=[x(s), y(s)], where *x(s)*, *y(s)* are *x-y* coordinates along the contour and $s \in [0,1]$ is the normalized arc length, that moves through the spatial domain of an image to minimize the energy functional:

$$E_{snake} = \int_0^1 (E_{intern}[v(s)] + E_{image}[v(s)] + E_{constr}[v(s)])ds \qquad (1)$$

The exact energy function terms come in various forms depending on the problem in hand [5].

The internal energy maintains the "smoothness" of the spline. It has the following expression

$$E_{intern} = \frac{1}{2}(a|v'(s)|^2 + \beta|v''(s)|^2) \qquad (2)$$

The α and β are weighting parameters that control the snake's tension and rigidity, respectively (α and β can be functions of the contour parameter s). The functions $v'(s)$ and $v''(s)$ denote the first and second derivatives of $v(s)$ with respect to s. The image energy function E_{image} is derived from the image in a matter that minimizes its value at the features of interest, such as boundaries [5,6].

Given a gray level image $I(x,y)$ which is viewed as a function of continuous position variables (x,y), the typical image energies designed to lead an active contour towards step edges are

$$E^1_{image} = -|\nabla I(x,y)|^2 \tag{3}$$

$$E^2_{image} = -|\nabla(G_\sigma(x,y) * I(x,y))|^2 \tag{4}$$

Where $*$ stands for convolution, $G_\sigma(x,y)$ is a two-dimensional Gaussian function with standard deviation σ and ∇ is the gradient operator. If the image is a line drawing (black or white), then the appropriate image energies include

$$E^3_{image} = I(x,y) \tag{5}$$

$$E^4_{image} = G_\sigma(x,y) * I(x,y) \tag{6}$$

Obviously in these definitions larger σ's will cause the boundaries to become blurry and distorted. Such large σ's are often necessary, in order to make the effect of the boundary "felt" at some distance from the boundary, such as to increase the "capture range" of the active contour [7].

The external constraint forces can be applied both automatic or with manual supervision. To insert a spring-like force between a snake element and a point p in an image, the energy term E_{constr} can be expressed as a tensile energy $c|p-v|^2$, according to the sign of parameter c, the force can be either attractive or repulsive [5]. An ACM that minimizes its overall energy (E) must satisfy the Euler equation

$$\alpha v''(s) - \beta v''''(s) - \nabla E_{ext} = 0 \tag{7}$$

This can be viewed as a force balance equation

$$F_{\text{int}} + F^1_{ext} = 0 \tag{8}$$

Where $F_{\text{int}} = \alpha v''(s) - \beta v''''(s)$ and $F^1_{ext} = -\nabla E_{ext}$. The internal force F_{int} discourages stretching and bending while the external potential force F^1_{ext} pulls the ACM towards the desired image contour.

In recent years many different approaches have been proposed in order to overcome some inherent disadvantages concerning the initial ACM-algorithm. In this paper the methodology named dual active contour models is adopted [8] and is used for the specific problem under consideration.

3 Dual Active Contour Models

A discrete contour over a digitized image is defined by

$$v_i = (x_i, y_i) \ , i\text{=0..N-1} \tag{9}$$

The discrete active contour model is defined as

$$E_{snake}(v) = \sum_{i=0}^{N-1} E_{\text{int}}(v_i) + E_{image}(v_i) \tag{10}$$

The discretisation of the model introduces an approximation and special attention must be paid in the selection of the step size and the number of points, defining the snake and other parameters. In here the contour is restricted to lie on the pixel grid. The advantage of a search based discrete space is that the whole search space is finite and consequently minimization techniques exploit that property. A search-based technique does not require a driving force or the calculation of the gradient of the snake's energy function and therefore the solutions are not discounted.

The dual active contour employs a different energy function and optimization criterion to those of the original snake. Accordingly, this technique eases problems associated with initialization and improves its overall efficiency. The original initialization is replaced with a region-based version, enabling the computation of robust solution. However the solution must lie within the specified Region of Interest (ROI). To address the parameterization problem, this technique uses parameterization of the snake's energy based around regularization

$$E_{snake}(\mathbf{v}(s)) = \int_{s=0}^{1} \lambda E_{int}(\mathbf{v}(s)) + E_{image}(1-\lambda)(\mathbf{v}(s))ds \tag{11}$$

where $\lambda \in [0,1]$ is the regularization parameter. Setting λ = 1 completely regularizes the contour and the solution is entirely dependent upon the prior information. Setting λ = 0 provides no regularization and the contour is entirely dependent upon the image data. Somewhere between there is a more appropriate trade-off, combining image data and the a priori information.

The search-based approach employs a dynamic programming technique [9] to search the contour space and the problem of minimizing a function of the form:

$$E(v_1, v_2, ..., v_n) = E_1(v_1, v_2) + E_2(v_2, v_3) + ...E_{n-1}(v_{n-1}, v_n) \qquad (12)$$

Since the snake is a local-based approach with points dependent only on their neighbors, the dynamic programming technique is particularly suitable and reduces the problem of exhaustive search to a feasible calculation using the following “principle of optimality”: stating that

"All points on an optimal path are possible initial points for that path" [10].

4 Construction of the search space

Assuming that the part will be located near the center of the picture a search space is created around that center. The search space is generated from two initial discrete contours (circles) by connecting the corresponding points with lines.

A discrete contour is defined by equation (9), where subscripted arithmetic is modulo N. Each line is discretized into M points for the dynamic programming search. The energy of an open contour is

$$E_{snake}(\mathbf{v}) = E_0(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) + E_1(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) + \cdots + E_{N-3}(\mathbf{v}_{N-3}, \mathbf{v}_{N-2}, \mathbf{v}_{N-1}) \qquad (13)$$

emphasizing the local dependencies; a point is dependent only on its immediate neighbors for its energy. The energy at each snake point $\mathbf{v}_i$ is given by

$$E_{i-1}(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}) = \lambda_i E_{\text{int}}(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}) + (1 - \lambda_i) E_{ext}(\mathbf{v}_i) \qquad (14)$$

where $\lambda_i \in [0,1]$ is again the regularization parameter. In order to apply dynamic programming a two-element vector of state variables ($\mathbf{v}_{i+1}$, $\mathbf{v}_i$), is calculated at each stage. The optimal value function, S_i, is a function of two adjacent points on the contour and is calculated as:

$$S_i(\mathbf{v}_{i-1}, \mathbf{v}_i) = \min_{\mathbf{v}_{i-1}} \left[S_{i-1}(\mathbf{v}_i, \mathbf{v}_{i-1}) + \lambda_i E_{\text{int}}(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}) + (1 - \lambda_i) E_{ext}(\mathbf{v}_i) \right] \qquad (15)$$

given the initial condition $S_0(\mathbf{v}_1, \mathbf{v}_0)=0$. In addition to the energy matrix corresponding to the optimal value function, a position matrix is also required. Each entry of the position matrix at stage i stores the value of $\mathbf{v}_{i-1}$ that minimizes the optimality function, equation (15). This is evaluated for i = 1…N-2. The result is obtained by back tracking through the position matrix. The internal energy is given by:

$$E_{\text{int}}(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}) = \left(\frac{|\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}|}{|\mathbf{v}_{i+1} - \mathbf{v}_{i-1}|} \right)^2 \qquad (16)$$

The numerator is the discrete curvature term from the original snake. The continuity term is less important because constraining the points to lie on specific lines controls the point spacing. The denominator ensures that the internal energy is scale-invariant making no preference for large or small contours nut rather smooth ones.

The energy of a closed snake contains two extra energy terms arising from points where the open snake is joined and is given by

$$\begin{aligned} E_{snake}(\mathbf{v}) = E_0(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) + E_1(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) + \cdots E_{N-3}(\mathbf{v}_{N-3}, \mathbf{v}_{N-2}, \mathbf{v}_{N-1}) \\ + E_{N-2}(\mathbf{v}_{N-2}, \mathbf{v}_{N-1}, \mathbf{v}_0) + E_{N-1}(\mathbf{v}_{N-1}, \mathbf{v}_0, \mathbf{v}_1) \end{aligned} \quad (17)$$

The efficiency of dynamic programming is compromised when applied to closed contour problems. To guarantee a global minimum, using the method of [11] requires a separate optimization to be calculated for all values v_0 and v_1, incurring an M^2 increase over the open contour optimization. To avoid this increase an approximate solution using a two stage technique, transforms the problem into two open contour optimizations. First an open contour solution is found, which does not apply any continuity or smoothness constraints at the ends. The two points at the middle point of this contour are then taken as the start and end points for the closed contour. A second optimization of the energy function given in equation (17), is computed with fixed v_0 and v_1. The optimality function is evaluated for i=1...N. By fixing the two points v_0 and v_1 the closed contour optimization can be achieved.

5 The application of the Active Contour Models

The scope of the utilized algorithm is the detection of the center of the incoming cylindrical part with unknown radius. The specification of the center will determine the basic transpositions of the part in a 2D place. This is necessary for the next process of welding that is going to take place.

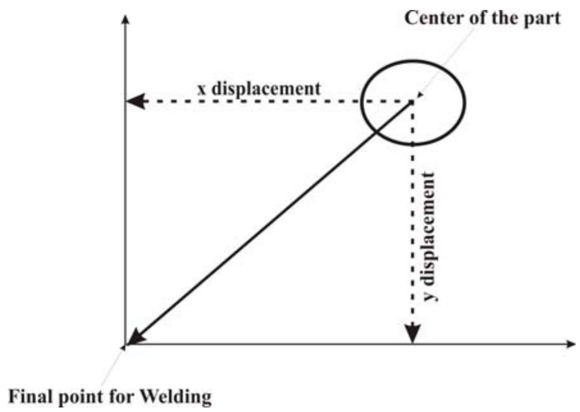

Fig. 2. The problem of area localization

If the procedure of the image processing misses the center, then the welding fails and the part should be rejected, having bad quality. The problem is of area localization is presented in Figure 2. The transpositions of the part are executed by a couple of step motors, driven by the calculated x and y displacements. The image acquisition, the algorithms for the active contour, the mathematical calculations and the D/A conversion of the signal for driving the step motors, are all developed in Labview 6.0. The Image acquisition is used in an open loop control form, which is presented in Figure 3.

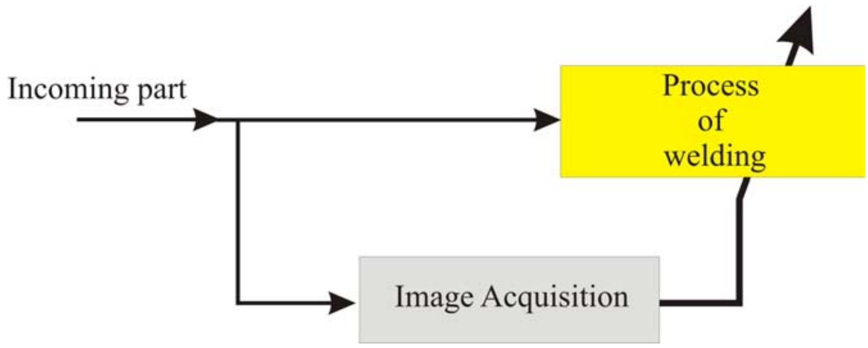

Fig. 3. The control open loop structure

The incoming parts are placed at the welding position on an X-Y table and are properly lighted in order to avoid the noise in the acquired photograph. After the shape and center definition the X-Y table moves the part to a pre-defined point for the welding process. This point is the center of the x-y axes and all the transpositions are calculated with respect to this point of reference. A typical acquired frame of the incoming part is presented in Figure 4 with a captured resolution of 300x300 dpi (RGB format).

Fig. 4. The acquired pre-processed photo of the part

In our approach we created a search space consisted of 75 lines with 60 evenly distributed points on them, on which the nodes of the snake can reside. To be more specific this is accomplished in a three-step process

1. The center of the image is tracked
2. Two circles are constructed having the same center (which comes from the first step) Figure 5

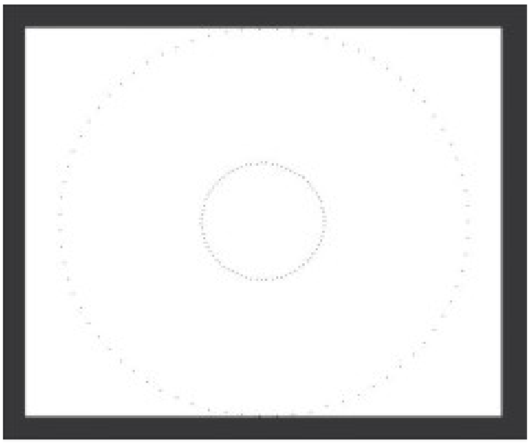

Fig. 5: The formulation of the 2 circles

The corresponding points on these 2 circles are connected and evenly distributed points on these lines are defined. This way a “star-like” region of interest is constructed Figure 6. Thus the snake can only move along these directions.

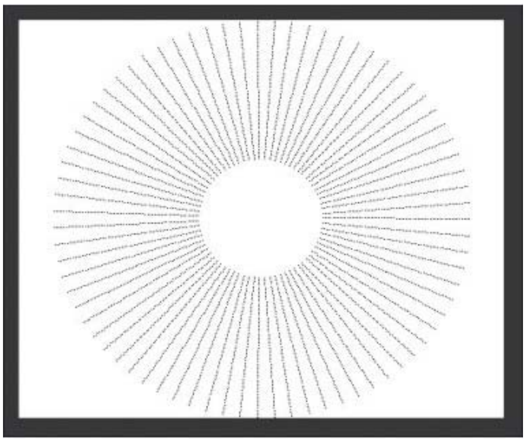

Fig. 6. The search space

After the construction of the search space the algorithm just described in the above section is used for the minimization of the energy's functional. The functional $\boldsymbol{S_i}$ is a non-acceding function taking its smallest value in the final iteration of the algorithm as it is illustrated in Figure 7.

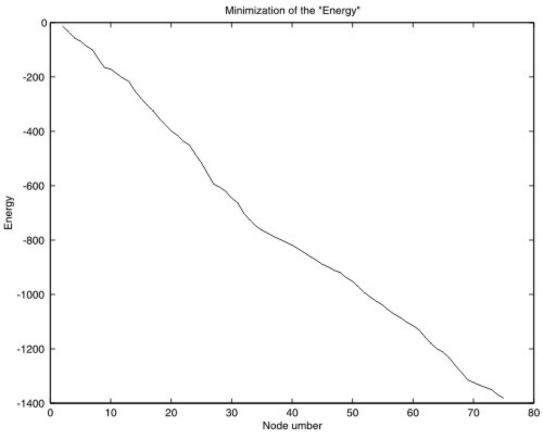

Fig. 7. The energy minimization

Having minimized the energy of the snake, the final position of the nodes is used to locate the center of the part. Taking advantage of the fact that the perpendicular bisector of the line segment which connects two points that belong in a circle and that the inner-product of perpendicular vectors equals zero, we create a linear system of the form

$$(x_i - x_j)(x_{centre} - \frac{x_i + x_j}{2}) + (y_i - y_j)(y_{centre} - \frac{y_i + y_j}{2}) = 0$$

or

$$(x_i - x_j)x_{centre} + (y_i - y_j)y_{centre} = (x_i - x_j)(\frac{x_i + x_j}{2}) + (y_i - y_j)(\frac{y_i + y_j}{2}), i \neq j \quad (18)$$

or in a more compact form

$$A\bar{x} = b \quad (19)$$

where $\bar{x} = [x_{center}, y_{center}]^T$ (20)

The least squares solution to this inconsistent system is $\bar{x} = (A^T A)^{-1} A^T b$ Consequently having the center of the part the radius is calculated solving the equation

$$(x_i - x_{center})^2 + (y_i - y_{center})^2 = r^2 \quad (21)$$

and averaging over the whole population of the points on the contour. The final outcome of the algorithm is illustrated in Figure 8 where the center and the circumference are denoted with white dots. The algorithm is presented in Table 1.

Table 1. The proposed algorithm for center definition

The Intelligent Algorithm for Center Computation
Step1: Acquire the stand-still image of the incoming part
Step2: Conversion of the image from RGB to Grayscale
Step3: Definition of the Region Of Interest (ROI)
Step4: Minimization of the Energy
Step5: Backtracking for computing the Optimal points
Step6: Center calculation
Step7: Return to *Step1*

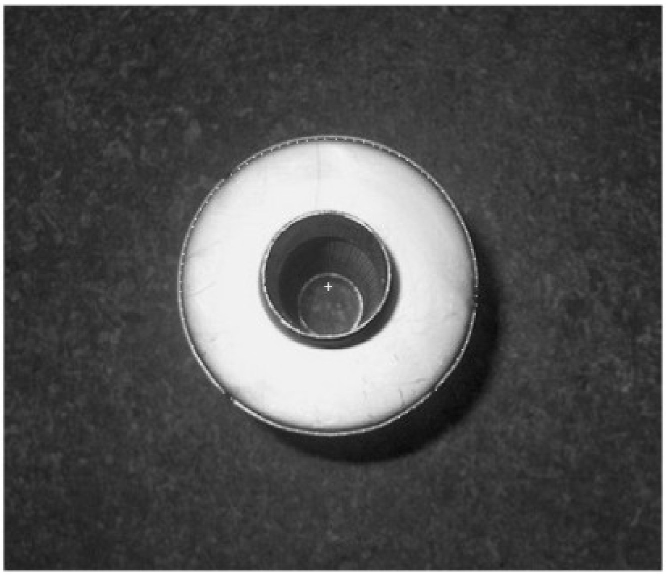

Fig. 8. The outcome of the algorithm

6 Conclusions

In this paper we have proposed a new algorithm, based on digital image processing, for the definition of the center of unknown incoming cylindrical parts. The algorithm was occupied in a general framework of an industrial process of welding. The applied

ACM utilized the a priori knowledge about the shape of the incoming part and was found to be robust and stable against bad lighting conditions and noise injection.

References

1. Horn, B.: Robot Vision MIT Electrical Engineering and Computer Science (1986)
2. Blake, A., Isard, M., : Active Contours, Springer-Verlag Berlin Heidelberg New York (1998)
3. Gonzales, R., Wintz, P.: Digital Image Processing, Addilson Welsey (1987)
4. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. Int. J. Computer Vision, 1(4) (1987) 321-331
5. Yang, Q., Marchant , J. A.: Accurate blemish detection with active contour models. Computers and Electronics in Agriculture. 14 (1996) 77-89
6. Lam,C. L., Yuen, S. Y.: An unbiased active contour algorithm for object tracking. Pattern Recognition Letters 19 (1998) 491-498
7. Xu, C. and Prince, J.L.: Gradient Vector Flow: A New External Force for Snakes. IEEE Proc. Conf. On Comp. Vis. Patt. Recog. (CVPR'97)
8. Gunn, S.: Dual Active Contour Models for Image Feature Extraction. PhD thesis University of Southampton, 1996
9. Amini, A., Weymouth, A., Jain, R. C.: Using Dynamic Programming for Solving Variational Problems in Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(9) (1990) 855-867
10. Bryson, A. : Dynamic Optimization Addison Wesley 1999.
11. Geiger, D., Gupta, A., Costa, L., Vlontzos, A., J.: Dynamical Programming for Detecting, Tracking and Matching Deformable Contours. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(3)(1995) 294-302