

A New Heuristic Search for Boundary Detection

Marc Salotti and Mostafa Hatimi

Groupe VISIA, Centre de Mathématiques et Calculs Scientifiques,
Faculté des Sciences, BP 52, 20250 Corté (FRANCE)
e-mail: salotti@univ-corse.fr, hatimi@univ-corse.fr

Abstract

Using the A* algorithm, we propose a new heuristic search strategy to find object boundaries. We show that a cost function with a Gaussian curvature is more appropriate to develop only the best paths. An application to the detection of cells boundaries is presented.

Keywords

Boundary detection, Edge following, Heuristic search

1 Introduction

Heuristic search strategies have been developed for many applications of artificial intelligence. Martelli applied the A* algorithm to boundary detection in 1972 [6]. Other similar approaches have been proposed, most of them for the detection of boundaries in biomedical applications. Encouraging results have been presented, but some difficulties were not successfully overcome.

In the first part, we propose a short overview of edge following techniques using the A* algorithm or variations of it. Then, we present new ideas to improve the search and we give the implementation details for the detection of an arbitrary number of cells with specific region properties. Some results are presented.

2 Heuristic Search Strategies

2.1 The Basic Algorithm

In 1972, Martelli showed that the problem of boundary detection can be brought back to the problem of finding the minimal cost path in a weighted and directed graph, with positive costs [6]. Although different modelisations can be made, we consider that nodes of the graph correspond to pixels, and arcs exist between two connected pixels P_i , P_j if the direction of vector (P_i, P_j) is the same as the tangent to the boundary given by vector (P_{i-1}, P_i) , with a maximal deviation of one pixel. Then, the key of the problem is the choice of the evaluation function used to define the quality of the path. This function is called f . The main stages of the A algorithm are described below [8], [2] :

1. Expand the start node (put the successors on a list called OPEN with pointers back to the start node).
2. Remove the node P_i of minimum f from OPEN. If P_i is a goal node, then stop. Trace back through pointers to find the optimal path. If OPEN is empty, then fail.

3. Else expand node P_i , putting successors on OPEN, or eventually updating them, with pointers back to P_i . Go to step 2.

The nodes are selected at each step according to an evaluation function $f(P_i)$, which is an estimate of the cost of a minimal cost path from s the starting node to a goal node constrained to go through node P_i . This function $f(P_i)$ can be expressed as :

$$f(P_i) = g(P_i) + h(P_i)$$

$g(P_i)$ is the cost of the path from s to P_i given by summing the arc costs encountered while following the pointers from P_i to s (this path is the lowest cost path from s to P_i found so far). $h(P_i)$ is an estimate of the cost of a minimal cost path from P_i to a goal node. If no information is available on the shape to be found, it is preferable to avoid estimating the cost from the current node to a goal node and therefore to set $h(P_i) = 0$ for all P_i . In this case, algorithm A^* coincides with Dijkstra's algorithm [7], [9].

2.2 Choice of Heuristics

Martelli suggested to use an expression roughly equivalent to the following :

$$\text{cost}(P_i, P_{i+1}) = M - \text{gradient}(P_{i+1}), \text{ where } M = \max_x \{ \text{gradient}(x) \} \quad (1)$$

The problem of the heuristic search is that it must keep track of a set of current best paths, and this set may become very large if using heuristics like expression (1): a good but long path may eventually look expensive compared to small undesirable paths. Several ideas have been proposed to solve the problem. The selection of the next node in the OPEN list can be made with a depth-first strategy or using a rating function [3], [10], [11]. Lester and *al* suggest to take the maximum cost arc of the path as the value of g . The advantage is that g does not build up continuously with depth, so that good paths can be followed for a long time [5]. Since the value of g necessarily increases with depth if the costs are positive, Ashkar and Modestino proposed a cost function that takes negative values if the edge has a good evaluation [1].

However, if some interesting ideas have been proposed, it seems that it is not possible to keep looking for the path with minimum cost and at the same time to have a depth first strategy, avoiding small undesirable paths.

2.3. Proposition for a Suitable Cost Function

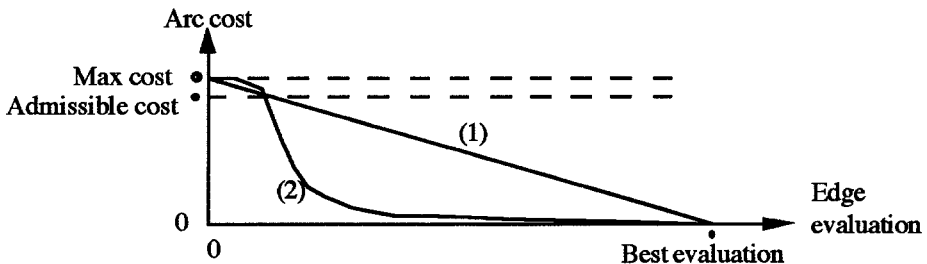


Fig. 1. Examples of cost functions.

The problem is that the sum of arc costs of a good and long path may be equal to the sum of arc costs of a small and undesirable path. In order to overcome this difficulty, it is necessary to assign a very high cost when the evaluation is bad and a very small cost if the evaluation is good. Two cost functions are displayed figure 1. Clearly, a linear expression like $\text{cost}(P_i, P_j) = M - \text{gradient}(P_i)$ is not appropriate. Function (2) is better, it could be generated by $\text{cost}(P_i, P_j) = e^{-\text{gradient}^2(P_i)}$. In a practical standpoint, if all gradient values of the best path are significant and little superior to the gradient values of other paths, this best path is developed first. On the contrary, if low gradient values are present, the path is not sure, the cost is high and other paths are explored. Since a Gaussian curvature is suitable to favor a depth first strategy, we propose a generalization, taking simultaneously into account several features:

$$\text{cost}(P_i, P_{i+1}) = \sum_k \alpha_k \cdot e^{-x_k^2} \quad (3)$$

For k features, x_k is an expression of feature k , and α_k determines the weight of each feature, such that $\sum_k \alpha_k = 1$.

3 Application

We present an application of heuristic search to determine the boundaries of specific objects. We assume that regions corresponding to these objects have homogeneous properties. Then we propose to specify the cost function (3) by the following expressions:

- Gradient criterion:

$$x_1 = \sigma_1 \cdot \text{gradient}(P_{i+1})$$

σ_1 specifies the shape of the function. A gradient map is obtained with Deriche's operator at the beginning of the program [4].

- Region criterion:

The goal is to find a cycle by turning around the object. We assume that the object is on the right side of the contour when expanding the nodes. In order to control the stability of the region properties, we propose to observe the gray level variations between the pixels located close to P_{i+1} and the pixels located close to the edge pixels $P_i, P_{i-1} \dots P_{i-n}$:

Let P_{i-1}, P_i, P_{i+1} be 3 connected pixels of the contour, if θ is the angle determined by:

$$\theta = \text{angle}(\text{vector } P_{i-1}, P_{i+1}) - \pi/2$$

and P_j is the pixel determined by:

$$X_{pj} = X_{pi} + \text{round}(2 \cdot \cos(\theta))$$

$$Y_{pj} = Y_{pi} + \text{round}(2 \cdot \sin(\theta))$$

then we can reasonably assume that P_j belongs to the region corresponding to the object. Moreover, in order to avoid errors due to noise, a 3x3 median filter is applied to P_j .

Finally, with $\text{Gray1} = \text{Gray_Level_Average}(P_{j-10}, P_{j-9}, \dots, P_j)$

and $\text{Gray2} = \text{Gray_Level}(P_{j+1})$

we propose to characterize the region criterion with the following expression:

if ($| \text{Gray2} - \text{Gray1} | > \text{maxdiff}$) then $x_2 = 0$
 else $x_2 = \sigma_2 \cdot (\text{maxdiff} - | \text{Gray2} - \text{Gray1} |)$

where maxdiff is a threshold for the maximum difference of the Gray level variations.

Other constraints have been added to the search:

- 1) if all objects are convex, the convexity of the shape can be controlled by measuring the curvature of the contour;
- 2) small closed contours with length inferior to a threshold are rejected;
- 3) since the exploration should stop if the quality of any feature is too weak (for example, the gradient value is 0), we propose avoiding the creation of a new arc if $\text{cost}(P_i, P_{i+1})$ is superior to a threshold admcost (admcost corresponds to the admissible cost of figure 1);
- 4) only the 3 first pixels of the search are declared goal nodes.

In order to find all objects, the image is first split into 10×10 square windows and a list of edge seeds is determined by selecting the pixels with maximum gradient value in each window.

4 Results

We propose to evaluate our method with a synthetic image and a real image. The same thresholds have been used in both cases:

$\alpha_1 = \alpha_2 = 0.5$, $\sigma_1 = 0.01$, $\sigma_2 = 0.1$, $\text{maxdiff} = 25$, $\text{minlength} = 30$, $\text{admcost} = 0.45$.

Each time a contour is closed, all pixels are marked and the corresponding region is filled. It takes approximately 2 minutes on a Sparc 2 to process a 256×256 image.

In the first image "test1", the Gray value of the regions corresponding to the objects is 100 and the Gray value of the background is 130. In order to add some difficulties, a small part of each boundary (a 10×10 window) has been blurred. All objects have been correctly detected. The second image is a noisy version of "test1": Gaussian noise with $\sigma = 10$ has been added. All objects have been found, but the boundaries do not always correspond to the exact frontiers.

In the second image "test2", we propose to try finding the contours of black cells uniquely. An edge map obtained with Deriche's detector and hysteresis thresholding (15,8) is presented to illustrate the difficulties: these cells are lowly contrasted and sometimes merged with other dark cells. The convexity criterion has been used to constrain our heuristic search. All cells detected have been correctly located without any error. Few cells too close from the border of the image have not been completely detected (these missing contours should not be considered as errors) and very few are missed, their contour being indistinguishable. Note that a constraint on the Gray level of regions has been used to avoid the search around light Gray cells.

5 Conclusion

Our conclusion is that heuristic search strategies are powerful tools in pattern recognition: few errors are made and good results can be obtained even with complex images. The advantage of the heuristic search approach holds in the possibility of adding many contextual constraints to the detection, thus making the search adaptive to the application. The major problem is to find a suitable expression of the different criteria and our proposition is finally a step toward an acceptable solution.

References

- [1] Ashkar,G.P. and Modestino,J.W. "The contour extraction problem with biomedical applications", CGIP, vol. 7, 331-355, 1978.
- [2] Ballard, D. H. and Brown, C. "Computer Vision", Prentice Hall Inc., Englewood Cliffs, New Jersey, USA, 1982.
- [3] Ballard,D.H. and Slansky,J. "A ladder-structured decision tree for recognizing tumors in chest radiographs", IEEE Trans. Computers, vol. 25, 503-513, 1976.
- [4] Deriche, R. "Using Canny's criteria to derive a recursively implemented optimal edge detector", International Journal of Computer Vision, vol. 1, 167-187, 1987.
- [5] Lester, J. M., Williams, H.A., Weintraub, B.A., and Brenner J.F., "Two graph searching techniques for boundary finding in white blood cell images", Computers in Biology and Medicine, col. 8, 293-308, 1978.
- [6] Martelli, A. "Edge detection using heuristic search methods", CGIP, vol. 1, n°2, 169-182, 1972.
- [7] Martelli, A. "An application of heuristic search methods to edge and contour detection", Comm. of the ACM, vol. 19, 73-83, 1976.
- [8] Nilsson,N.J. "Principles of artificial intelligence", Palo Alto, CA:Tioga, 1980.
- [9] Pearl, J. "Heuristics: intelligent search strategies for computer problem solving", Addison-Wesley, 1984.
- [10] Persoon, E. "A new edge detection algorithm and its applications in picture processing", CGIP, vol. 5, n°4, 425-446, 1976.
- [11] Weschler, H. and Sklansky J. "Finding the rib cage in chest radiographs", Pattern Recognition, vol. 9, 21-30, 1977.

