



Genetic contour matching

Alexander Toet^{a,*}, Walter P. Hajema^b

^a *TNO Human Factors Research Institute, Kampweg 5, 3769 DE Soesterberg, Netherlands*

^b *Ministry of Defence, DEBK/DCAKL, P.O. Box 90821, 2509 LV The Hague, Netherlands*

Received 5 January 1994; revised 3 January 1995

Abstract

Object recognition can be formulated as an optimization problem. The objective function measures for instance the evidential support for any particular projection of the parameterized object contour model onto the input image. A genetic algorithm can be used to find a set of parameters which provide an optimal interpretation of the image in terms of the model. Preliminary test results demonstrate the feasibility of the proposed approach.

Keywords: Object detection; Chamfer metric; Edge transform; Distance transform; Contour model; Contour matching; Genetic algorithms; Optimization

1. Introduction

Model-based object recognition involves the detection and classification of instantiations of predetermined spatial patterns or object models in an image. This involves the estimation of the model-to-image transformation that explains (results in) the observed image. For example, for a 2D image of a known rigid 3D object the goal is to find the (position- and viewpoint-) parameters of the model which best explain the projection observed in the image.

A fundamental problem encountered in previous approaches to model-based object recognition is the determination of a set of starting parameters (e.g. size, shape, orientation, position) which avoid recognition errors due to local extrema in the match evaluation function. In the absence of an initial approximation or cue the alternative of a “blind” search through the parameter space is highly unattractive because of its

large dimensionality. Genetic algorithms (Goldberg, 1989; Holland, 1992) focus on the global extrema of the evaluation function and are not easily trapped by local extrema. This paper reports a preliminary test of their ability to find a best match of a quantitative parameterized contour model to a grey-scale image.

An attractive feature of genetic algorithms is their ability to maintain a population of possible solutions. This makes them highly suitable for typical applications that require the recognition and localization of an object from a library of different object models. The use of multiple models allows alternative interpretations to compete with one another, the most plausible interpretation having the greatest probability of success. This introduces merely an extra dimension in the search space. The simultaneous optimization of parameters corresponding to high-level knowledge (different types of models) and low-level knowledge (the model- and viewing-parameters) results in a simultaneous top-down and bottom-up flow of information, where local evidence is actively gathered to

* Corresponding author. Email: toet@tm.tno.nl.

verify high-level assumptions and high-level knowledge is constantly updated using strictly local information. Hence, this corresponds to an effective integration of the inductive and deductive approaches to object recognition (Bhattacharjya et al., 1992; Friedland and Rosenfeld, 1992). By providing a limit to the number of iterations that are to be performed, the high-level knowledge may allow unknown objects to be identified as such (a “don’t know” result).

The contour matching algorithm presented in this paper involves 3 stages. First, a binary (cartoon-like) representation of the input image is created by thresholding the output of a simple difference operator. Second, this binary edge image is transformed into a grey-level image in which all pixels have a value that corresponds to the distance to the nearest edge pixel. Third, a genetic algorithm minimizes an average of the values of the pixels in the distance image that coincide with the projection of an instantiation of a parameterized 2D contour model. The contour model represents the hull of the star-shaped 2D projection of 3D compact objects, so that hidden edges are no problem. The result is a best fit of the parameterized contour model to the edge image. The genetic algorithm first creates an initial population of random model instances. An objective function, derived by considering local evidence for object contours (edges) within the image along profiles corresponding to projected model contours, is employed to evaluate the degree to which local image evidence supports any particular model instance. The outcome of the genetic algorithm is an instantiation of the contour model which maximally agrees with the image data.

In a similar study, Bengtsson (1990) presented a matching scheme based on a stochastic optimization method that is related to simulated annealing. It was applied to match a template contour of an object to edges in a digital image, using an edge distance measure as an evaluation function. The method was limited to a three parameter space: two translation parameters and one rotation parameter. The size and shape of the template contour were fixed. In a pilot test, the stochastic method was successful in 4 out of 8 trials, while a deterministic approach failed in all cases to find the correct match.

Friedland and Rosenfeld (1992) present a compact object recognition scheme based on a cyclic 1D Markov Random Field polar model representation.

The evaluation function consists of (i) a low-level part that restricts the model’s shape (contour smoothness) and measures the local edge strength in the image (i.e. the evidence from the image data), and (ii) a high-level part that matches a model instantiation to a library of existing object configurations. Simulated annealing was used to find a best match. The center of the model must be initialized at a location within the object that is to be classified. To avoid detection errors due to incorrect initialization, it was suggested to divide the image into many overlapping regions, and start the recognition scheme in each region.

The main goal of this study was to investigate the feasibility of a model based matching scheme based on a genetic algorithm. In this scheme, the size and shape of the model contour adapt to the local image evidence. Also, the method is less likely to suffer from initialization sensitivity, since the genetic algorithm generates model instances at all possible locations in the image plane, while avoiding being trapped in local minima.

The organization of this paper is as follows. Sections 2 and 3 discuss respectively the edge- and distance-transforms. Section 4 describes the contour model. Section 5 presents the evaluation function. Section 6 introduces the proposed genetic matching algorithm. Section 7 gives some results of a pilot test. Finally, Section 8 summarizes the main findings of this study.

2. Edge transform

Local discontinuities in image luminance are called *luminance edges*. Edges can be detected by applying a local difference or gradient operator to the image luminance function and thresholding the output (Rosenfeld and Kak, 1982). The result of this transformation is a cartoon-like image, representing the locations in the original image (Fig. 1) where the luminance gradient is strong (Fig. 2).

A wide range of edge detectors has been defined. Each of these detectors has its own characteristic performance (e.g. Pratt, 1991). Since the nature of the edge detector is irrelevant in this study, a simple Sobel difference operator is applied (Pratt, 1991; Rosenfeld and Kak, 1982).

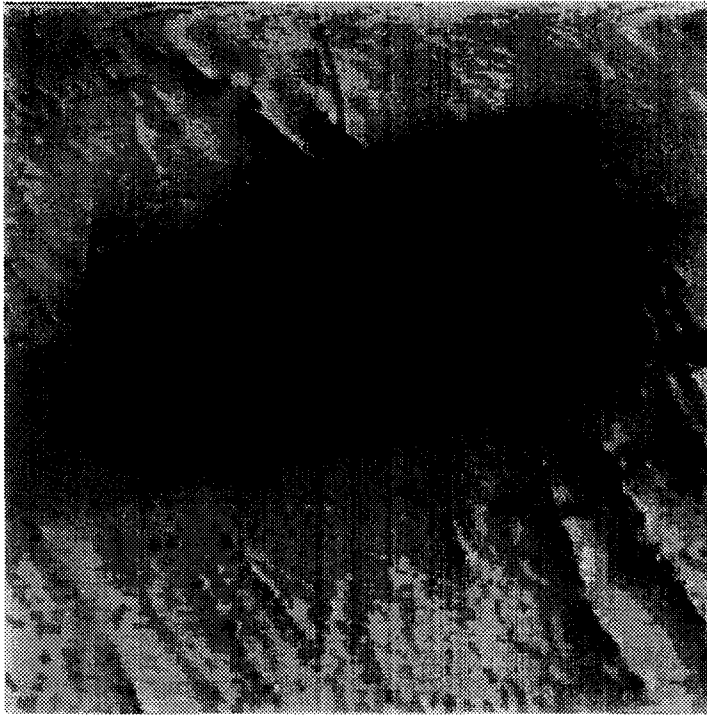


Fig. 1. Original jeep image.



Fig. 2. The thresholded Sobel edge transform of Fig. 1.

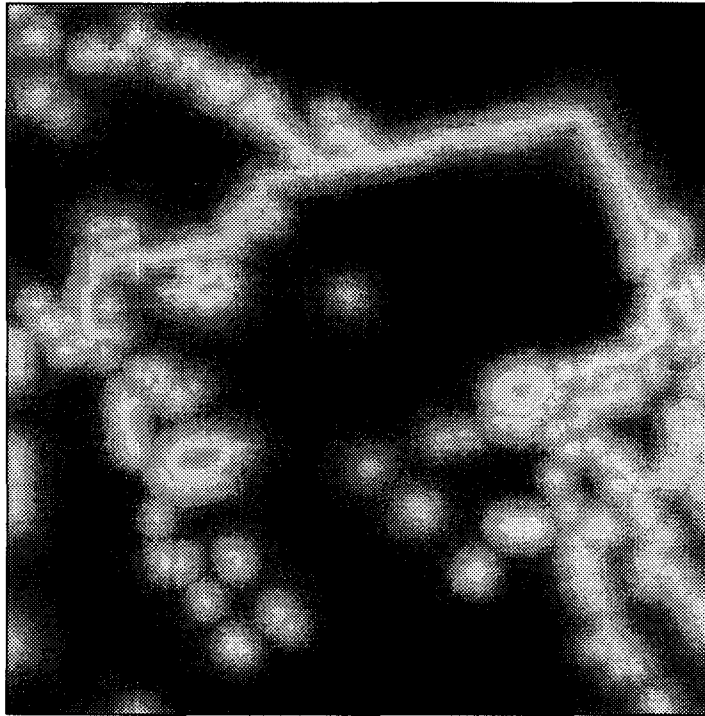


Fig. 3. The distance transform of Fig. 2. The grey-levels shown are inversely proportional to the calculated Chamfer distance; i.e. distance zero corresponds to a grey-value of 255, and progressively larger distances correspond to progressively lower grey-values.

3. Distance transform

A distance transformation converts a binary digital image, consisting of feature and non-feature pixels, into a grey-level image in which each pixel has a value that corresponds to the distance to the nearest feature pixel (Fig. 3). Sequential algorithms that approximate global distances in the image by propagating local distances (i.e. distances between neighbouring pixels) over the image plane are also known as “chamfer” distance algorithms.

Because of the discrete nature of digital images and the influence of noise on the location of edge points, it is an unnecessary waste of effort to compute exact Euclidian distances (Danielsson, 1980) from the inexact boundary pixels. In most digital image processing applications it is therefore preferable to use integers to represent distances. The two different local distances in a 3×3 neighbourhood are the distance between the horizontal/vertical neighbours and between diagonal neighbours. The chamfer-3/4 distance uses respectively the values 3 and 4 to represent these distances. This distance has a maximum difference of 8% from

the true Euclidian distance (Borgefors, 1986; Verwer, 1991).

4. Contour model

The simple generic contour model presented here is restricted to compact 2D image shapes. A compact 2D object has a star-shaped boundary, i.e. a boundary which is entirely visible from an interior point (Friedland and Rosenfeld, 1992; Rosenfeld and Kak, 1982). A 2D shape S is called *star-shaped* from a point P if every line through P meets S only once. If a 2D shape S is star-shaped from P , every point of S , and in particular every point of the boundary of S , is visible from P . In this case, the boundary of S has a single-valued one-dimensional cyclic polar coordinate representation (R, θ) , where $R = \{r_1, \dots, r_n\}$ and $\theta = \{\theta_1, \dots, \theta_n\}$ are vectors or 1D arrays of length n of discrete variables that represent respectively the distances of the boundary points from the center P , and the angular orientation of the radii emanating from P . If S is convex, this representation can be constructed

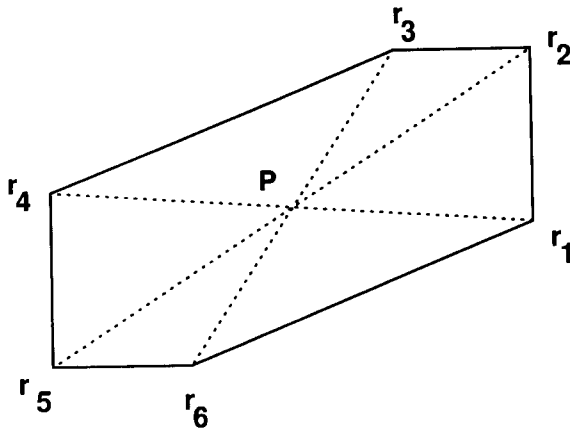


Fig. 4. Projection of the 2D contour model, with 3 independent axes r_1, r_2, r_3 .

for any $P \in S$. For simplicity, the neighbourhood N_i of a boundary point i is defined as $\{i-1, i, i+1\}$. For closed contours, $r_{n+1} = r_1$, $\theta_{n+1} = \theta_1$, and the neighbourhoods are defined modulo n .

In aerial imagery, the silhouette of the 2D projection of the main body of most vehicles and buildings can be approximated by the convex hull of the end-points of minimally 2 and at most 3 axes with coincident midpoints. Each of these axes corresponds to a pair of collinear radii of equal length and with an angular interval of π radians (Fig. 4). For 3 independent axes, the simple generic contour model is therefore completely characterized by 6 parameters: $R = \{r_1, r_2, r_3\}$ and $\theta = \{\theta_1, \theta_2, \theta_3\}$. The polar coordinates of the other 3 contour points are $(r_1, \theta_1 + \pi)$, $(r_2, \theta_2 + \pi)$ and $(r_3, \theta_3 + \pi)$. The location of the center of the contour is determined by 2 coordinates in the image plane. Finding the best match of a model instance to the image data can in this case be formulated as an optimization problem with 8 parameters.

5. Evaluation function

After a model has been instantiated into the image, its fit to the image must be evaluated. No single, universal evaluation function exists for all matching problems. The objective function used in this study is based on the observation that a good match is one where every element of the projected model contour is spatially near an image contour. It is formulated as the cumulative sum of the distance value along the

projection of the model contour in the image plane.

In the present application the features that are to be matched are the luminance edges in the image and the contours of the projection of a model instance. The distance transformation is highly suitable for matching edge images (Borgefors, 1988; Liu and Srinath, 1990). A distance or match between two sets of image contours can be computed by (i) projecting the first set of edges onto the distance transform of the second set and by (ii) computing an average of the values of the pixels in the distance image that coincide with the edge projections. The average value thus found is called the *edge distance*. A perfect fit between two sets of edges results in an edge distance zero, since each pixel from of the projected model contour will be at zero distance from an image contour.

The edge distance depends both on the distance transformation used and on the type of average that is taken. It is computed by traversing the projection of the model edges in the image plane pixel by pixel while computing the cumulative sum of the distance value at each location (Borgefors, 1988). Extensive tests (Borgefors, 1984) have shown that the root mean square (r.m.s.) average is a good choice for the distance measure D between the edge image E and the projected model contours M :

$$D(M, E) = \frac{1}{3} \sqrt{\frac{1}{n} \sum_{i=1}^n v_i^2}$$

where v_i are the distance values and n the number of pixels in M . The average is divided by three to compensate the unit distance in the chamfer-3/4 distance transform. Note that the number actually used in the matching algorithm can be the sum of squares, which is integer. Other possible distance measures are for instance the median of the distance value of the individual contour points, or the fraction of the number of model edge points that is within a certain threshold distance from the image edges (the *edge coverage*: Mundy and Heller, 1990).

6. Genetic matching

Contour matching is generally computationally expensive, since it involves the generation and evaluation of a large number of different model instances.

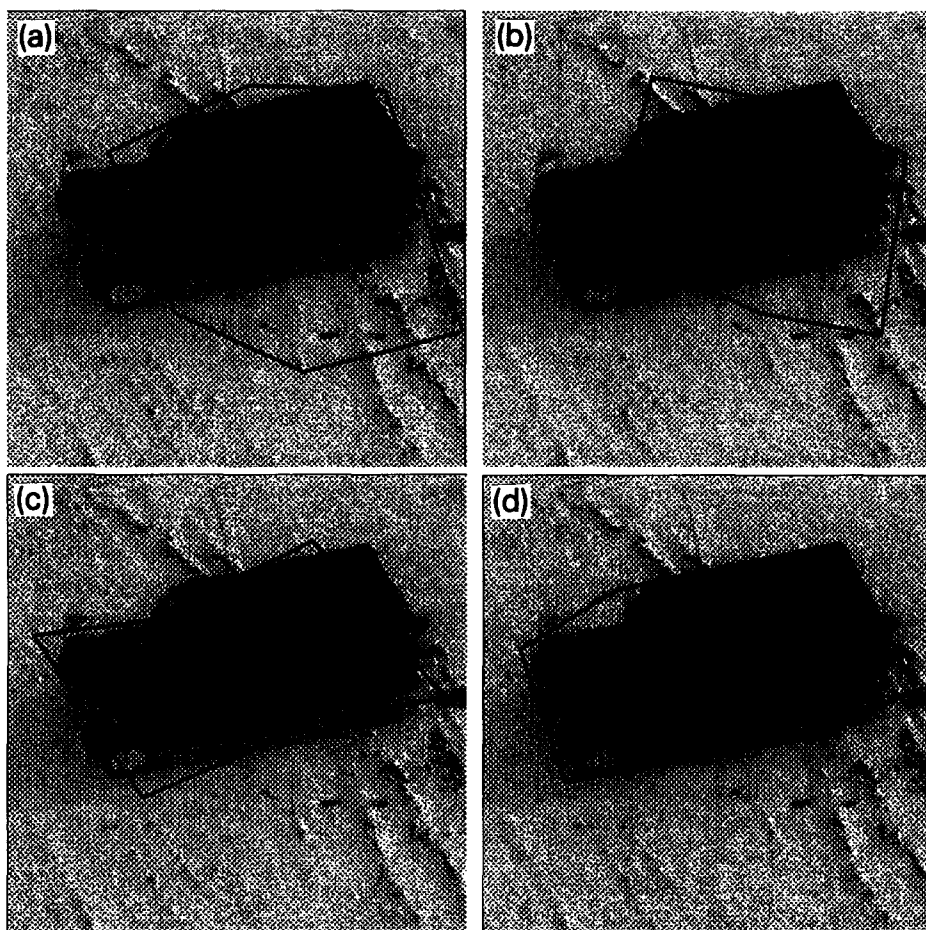


Fig. 5. Some steps in the 2D genetic contour matching process. The best fitting contour is shown after respectively 250 (a), 500 (b), 750 (c) and 1250 (d) iterations of the genetic algorithm.

Finding the best match between two image contours involves minimizing the evaluation function. Unless the matching situation is very simple, this function will have many local minima apart from the global one. To find the global minimum, the search has to be started close to the optimal match position. Unless this position is known, searches must be started in a large number of different positions, so that at least one is close to the optimal location. Each search is rather costly in computing resources.

This genetic contour matching approach alleviates the abovementioned problem by (i) sampling the objective function in many regions of the parameter space simultaneously and by (ii) focussing the search process on those parts of the solution space that are most

likely to contain a global extremum of the objective function.

In this study Grefenstette's (1987) GENESIS system for genetic function optimization is used to find the model parameters corresponding to a global optimum of the objective function. The default parameter values are used, i.e. the population size is 50, the crossover rate is 0.60, and the mutation rate is 0.001. In general, the maximum number of objective function evaluations is fixed at 5000. The initial population from which the search progresses is purely random.

The matching process starts with the generation of an initial population of 2D star-shaped candidate contours, by randomly selecting the parameter values from

their respective domains, using a uniform distribution function. Each contour S_i is represented by a vector or parameter list of length 8:

$$S_i = [r_{1i}, \theta_{1i}, r_{2i}, \theta_{2i}, r_{3i}, \theta_{3i}, x_i, y_i],$$

where the 3 pairs r, θ represent the polar coordinates of the 3 axes and x_i, y_i are the coordinates of the center of the contour in the 2D image plane. Both an upper bound (128) and a lower bound (10) are adopted for the radii of these contours. The upper bound reflects a priori information about the maximal object size. The lower bound serves to prevent a large number of false alarms (extremely small contours will fit almost any noise speckle). Similarly, the angular interval is restricted (the angle between 2 neighbouring radii should be (i) larger than 5 deg and (ii) positive) to prevent degenerate contours corresponding to (i) self-intersections of the contour, caused by order-reversal of the radii, or to (ii) collinearity of the radii. These restrictions reduce the searchspace and thereby the amount of computation needed to find a best match. The cycle of (i) population evaluation and (ii) generation of a new population is repeated until either an acceptable match has been found or an upper bound has been reached on the number of optimization iterations. By providing a limit to the number of iterations performed, the absence of contour evidence can be signalled and a “no match” classification can be obtained.

7. Results

The 2D genetic contour matching scheme was tested on aerial reconnaissance images of military vehicles. Fig. 1 shows an image of a jeep. The thresholded Sobel-edge transform of Fig. 1 is shown in Fig. 2. Fig. 3 shows the chamfer-3/4 distance transform of Fig. 2. The best fitting contour after a number of generations is shown in Figs. 5(a)–(d). On average, the algorithm converged after about 1000–1500 iterations of the genetic contour matching procedure. A number of 1000 iterations (corresponding to 50,000 evaluations) takes about one minute on a SUN Sparc LX. The best match obtained closely approximates the original image contours.

8. Concluding remarks

This paper proposes a 2D genetic contour matching scheme. In a preliminary test, the feasibility of this scheme was proven on aerial reconnaissance images of military vehicles. The results show that the best match found from the genetic contour matching procedure approximates the original image contours.

Currently the algorithm is being tested on artificial rendered images. In this case the location of the object boundaries is known exactly and can be used to evaluate the quality of the match.

Possible application areas include aerial recognition (updating road and vegetation maps) and surveillance (military target recognition).

References

- Bengtsson, M. (1990). Stochastic optimization algorithms — an application to pattern matching. *Pattern Recognition Lett.* 11, 717–724.
- Bhattacharjya, A.K., D.E. Becker and B. Roysam (1992). A genetic algorithm for intelligent imaging from quantum-limited data. *Signal Processing* 28, 335–348.
- Borgefors, G. (1984). An improved version of the chamfer matching algorithm. In: *Proc. 7th Internat. Conf. Pattern Recognition*, 1175–1177.
- Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing* 34, 344–371.
- Borgefors, G. (1988). Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* 10, 849–865.
- Danielsson, P.E. (1980). Euclidian distance mapping. *Computer Vision, Graphics, and Image Processing* 14, 227–248.
- Friedland, N.S. and A. Rosenfeld (1992). Compact object recognition using energy-function-based optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 770–777.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York.
- Goldberg, R.R. (1993). Pose determination of parameterized object models from a monocular image. *Image and Vision Computing* 11, 49–62.
- Grefenstette, J.J. (1987). *A User's Guide to GENESIS*. Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC.
- Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA.
- Liu, H.C. and M.D. Srinath (1990). Partial shape classification using contour matching in distance transformation. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 1072–1079.

- Mundy, J.L. and A.J. Heller (1990). The evolution and testing of a model-based object recognition system. In: *IEEE Proc. 3rd Internat. Conf. on Computer Vision*.
- Pratt, W.K. (1991). *Digital Image Processing*, 2nd edition. Wiley, New York.
- Rosenfeld, A. and A.C. Kak (1982). *Digital Picture Processing*, Vol. 2, 2nd edition. Academic Press, New York.
- Verwer, B.J.H. (1991). Local distances for distance transformations in two and three dimensions. *Pattern Recognition Lett.* 12, 671–682.