Genetic Programming for Image Analysis: Orientation Detection

Simon C. Roberts and **Daniel Howard**
Software Evolution Centre, Systems & Software Engineering Centre,
Room U50, Defence Evaluation and Research Agency,
Malvern, Worcestershire WR14 3PS, UK
dhoward@dera.gov.uk, Tel:(44)(1684)894480

Abstract

The problem addressed is that of detecting the orientation of objects in infrared linescan (IRLS) imagery obtained by low-flying aircraft. A novel method for orientation detection is described, based upon the minimisation of pixel-value deviation for linearly-spaced pixels. Genetic programming (GP) is used to discover the optimum manipulation of pixel information for this task. Results are compared against those produced using the conventional method of second-order central moments for principal axis detection.

1 INTRODUCTION

The principal axis (also known as major axis) of a 2-dimensional shape is the axis corresponding to the least second moment of inertia, and this axis gives an object's orientation relative to some fixed coordinate system. The principal axis is conventionally derived from the second-order central moments as follows (Xiaoqi and Baozong, 1995):

$$\theta = \frac{1}{2}\arctan\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \qquad (1)$$

where μ_{pq} denotes the $(p+q)$ order central moment which is computed as:

$$\mu_{pq} = \sum_x \sum_y (x - x_m)^p (y - y_m)^q i(x, y) \qquad (2)$$

where (x_m, y_m) is the centre of mass for the imagery $i(x, y)$.

This method assumes that an object's shape is strongly correlated to pixel brightness. However, objects in IRLS imagery often have poorly defined shape due to hot spots, cold spots, thermal shadows and noise, thus producing brightness variations across an object's regions (Howard and Roberts, 1999). Most importantly for orientation detection, an object can have uneven brightness at its boundaries, e.g. parallel sides of different brightness. An alternative method for orientation detection is described below, which is less susceptible to overall brightness variation.

The ability to detect an object's orientation is important for automatic object discrimination because it provides a reference axis and could thus avoid the need to repeat computationally-intensive rotationally-variant processes. For example, a template comparison technique could index pixel information relative to the principal axis prior to convolution as opposed to repeating the convolution at different template rotations. The impetus for investigating orientation detection was to aid an object detection system that uses GP, by eliminating the prerequisite for rotation-invariant terminal data (Roberts and Howard, 1999)(Howard *et al.*, 1999).

2 AN ALTERNATIVE METHOD FOR ORIENTATION DETECTION

The method assumes that for a regularly-shaped object, a line of pixels exists on or near the object where the deviation of pixel values is minimal, i.e. the line has near-uniform brightness. Furthermore, it is assumed that this line can be systematically related to an object's orientation.

This is illustrated by Figure 1 which shows 2 lines superimposed near a car. The line with the lowest deviation is that which lies outside the car and indeed this line indicates the car's orientation by virtue of the fact that it is parallel to the car. In general, a line will have a relatively low deviation if it does not cross an

object boundary, and within the vicinity of an object this is most likely to occur when the line is parallel to the object.

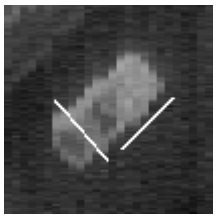

Figure 1: Example of a vehicle in IRLS imagery.

Any line that lies within a uniform region will have a low pixel-value deviation, thus a configuration of lines is required in order to discover uniform lines on or near objects. Hence a configuration is required where a line crosses an object boundary whilst another line does not, i.e. one line has a relatively high deviation whilst another line has a relatively low deviation.

The triangular configuration shown in Figure 2 was deemed advantageous for near-rectangular objects (like vehicles). When a single line has a relatively low deviation it is likely to be parallel to the object and thus the other two lines are likely to have relatively high deviations, i.e. the other two lines are unlikely to be aligned with the object's major or minor axes. Therefore, this reduces the orientation detection problem to that of formulating a process which manipulates pixel deviations along a set configuration of lines. GP was used to discover the optimum processing for various line configurations to detect the orientation of vehicles in IRLS imagery.

3 EVOLVING AN ORIENTATION DETECTOR

Given a rectangular image of $n \times n$ pixels containing a single vehicle, the task was to construct a function of pixel data which could indicate the vehicle's orientation. Typically, $n \times n$ was $O(10^3)$ pixels but the actual value depended on altitude scaling. A vehicle width (VW) was typically 23 pixels at a reference altitude of 300ft.

3.1 Pixel data

The pixel data comprised the following elements:

- AA and AS are the average and standard deviation pixel values over an area of approximately 15VW square.
- PA_d and PS_d are the average and standard deviation pixel values over a single-pixel width ring of diameter d defined in terms of VW. (See (Roberts and Howard, 1999) for more details.)
- TA_{DLRN} and TS_{DLRN} are the average and standard deviation pixel values over one line in a triangular configuration, where D refers to the triangle size, L refers to the line length, R is a rotation index and N is a line index as described below.

The line configuration was based on an equilateral triangle as shown in Figure 2. The triangle had an internal circle of diameter D defined in terms of VW (i.e. the centre point of each side lay on the circumference). The length of each line was a factor L of the triangle side where $0.5 \leq L \leq 1.0$. In theory, lower values of L should more readily yield near-uniform lines (i.e. the line was less likely to cross an object boundary or shadow boundary). The values D and L were varied for different GP runs.

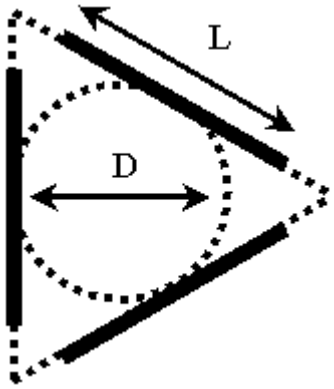

Figure 2: Configuration where lines are equal to or shorter than the sides of an equilateral triangle.

The line configuration was rotated clockwise about its centre in 5 degree steps thus giving 24 positions before the configuration returned to its initial position, although a given line would now cover the pixels initially covered by the line immediately clockwise. This is depicted in Figure 3. Thus each vehicle was associated with 72 values of TA and TS for a given GP run (i.e. a value for each of the 3 lines at 24 rotations).

The lines in the configuration were indexed in the clockwise direction to suit the reference coordinate system for orientation measurements. For each rotation step, the lines were ordered according to ascending pixel-value deviation, such that N_0 gave the index of the line with the minimum deviation and N_2 gave the index of the line with the maximum deviation. This is illustrated by Figure 4 which shows the indexed line

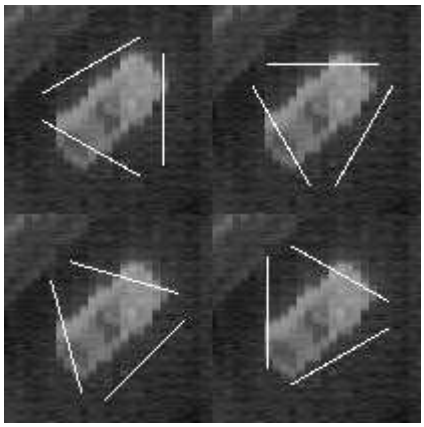

Figure 3: Line configuration rotated clockwise about the geometric centre of a vehicle. Top left is at 0 degrees; top right is at 30 degrees; bottom left is at 45 degrees and bottom right is at 60 degrees.

configuration at four rotation steps and Table 1 which gives a possible line order for each rotation step, assuming that a pixel deviation is lower when the associated line lies mostly outside the vehicle.

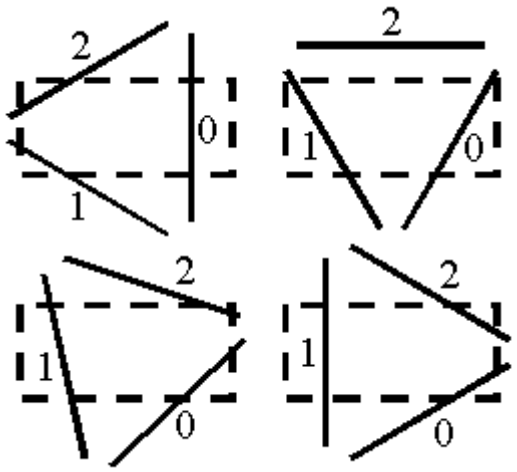

Figure 4: Line indices at four rotation steps. Top left is at 0 degrees; top right is at 30 degrees; bottom left is at 45 degrees and bottom right is at 60 degrees.

The variables associated with the above pixel data features were fixed for each GP run, as opposed to randomising the variables during evolution, thus allowing all the pixel data to be pre-computed for each vehicle prior to any GP runs. This approach reduces both the size of the search space and the computing time requirement (Poli, 1996); concerns which have been reported by other investigators researching GP for image analysis (Tackett, 1993)(Winkeler and Manjunath, 1997)(Daida *et al.*, 1996).

Table 1: Example of line ordering based on pixel deviation at four rotation steps.

rotation (degrees)	N_0	N_1	N_2
0	1	2	0
30	2	1	0
45	2	0	1
60	0	2	1

Table 2: GP parameters

parameter	setting
functions	+, -, *, / ($x/0 = 1$),
	$\min(A, B)$, $\max(A, B)$,
	if $(A < B)$ then C else D,
	TriCmp (see below)
terminals	pixel data: AA, AS,
	PA_d, PS_d,
	TA_{DLRN}, TS_{DLRN},
	altitude in feet (250 to 650)
population	1000
mate radius	1000
kill tournament	size 2 for steady-state GP
breed tournament	size 4 for steady-state GP
regeneration	90% x-over, 5% clone,
	5% truncation mutation
max generations	20
max tree size	1000 nodes
fitness measure	av. orientation error (see below)

3.2 GP specifications

Each individual was represented as a tree consisting of function and terminal nodes. The population was initialised using the ramped-half-and-half technique (Koza, 1992) with a maximum initial tree size of 10 nodes. The GP parameters are given in Table 2.

Four concentric pixel rings (PA_d and PS_d) were used for each vehicle with $d = 0.5, 1.0, 1.5$ and 2.0VW as displayed in Figure 5. The limits of the line configuration data were $1.0 \leq D \leq 2.0\text{VW}$, $0.5 \leq L \leq 1.0$, $0 \leq R < 24$ and $0 \leq N < 3$.

The motivation for using truncation mutation was to counteract the well-known 'bloating' behaviour of GP without severely restricting GP's generality and search power, e.g. as opposed to including parsimony pressure in the fitness measure (Langdon and Poli, 1998)(Rosca, 1996)(Soule and Foster, 1997).

For each vehicle, the fitness measure was formulated to

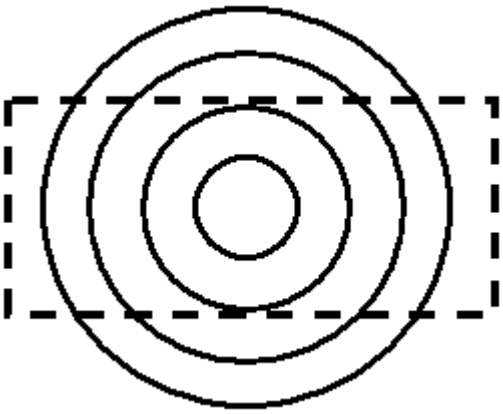

Figure 5: Four concentric pixel rings centred on a vehicle with diameters: 0.5, 1.0, 1.5 and 2.0VW.

train a GP tree to flag when the line configuration was rotated to the same orientation as the vehicle. The line configuration was rotated clockwise through the 24 possible positions, and the tree was evaluated for each position. Note that, at a given rotation step, the tree could exploit all of the 72 TA and TS values due to the rotation index R. When $R = 0$ the values corresponded to the current rotation. When $R = n$ the values corresponded to n clockwise rotations in advance, which included wrap-around such that $R = 23$ was actually a single anti-clockwise rotation.

When the tree returned a negative value the current rotation was deemed insignificant, otherwise the vehicle orientation was calculated in degrees as follows.

$$\theta_c = 5R_c + 120N_0 \tag{3}$$

where R_c is the index of the current rotation step (0 to 23) and N_0 is the index of the line with the minimum pixel-value deviation at the current position (0 to 2). The actual orientation of each vehicle was known thus allowing the absolute orientation error θ_e to be calculated. Note that the maximum θ_e was 90 degrees because the known orientation did not discriminate the front of the vehicle from the back, as it was the orientation of the vehicle's shape which was deemed important. In other words, rotating a vehicle about 180 degrees had no consequence. The fitness measure was assigned to the average θ_e for all orientation calculations over all vehicles:

$$\text{fitness} = \frac{1}{n_v} \sum_{v=1}^{n_v} \frac{1}{n_c(v)} \sum_{c=1}^{n_c(v)} \theta_e(v, c) \tag{4}$$

where n_v is the number of vehicles and $n_c(v)$ is the number of orientation calculations for a given vehicle.

Table 3: TriCmp conditions dependent on variable j.

j	condition data
0	$R = 0$ and $N = 1$
> 0	$R = j$ and $N = 0$

It was possible for a tree to return a negative value for all rotation steps when processing a single vehicle, thus never spawning an orientation calculation. In this case, the tree was punished by setting θ_e to 180 degrees. Conversely, it was possible for a tree to return a non-negative value for all rotation steps thus accumulating a large average θ_e.

The TriCmp function was associated with a randomised integer $0 \leq j \leq 23$ to compare specific TS_{DLRN} values to a single argument α as follows. If $TS > \alpha$ for $R = 0$ and $N = 0$ (i.e. the minimum deviation at the current rotation step) then the return was $-\alpha$. Otherwise the return depended on an additional $TS > \alpha$ condition using different R and N as shown in Table 3. Therefore, when $j = 0$ TriCmp depended on the second lowest TS at the current rotation step, and when $j > 0$ TriCmp depended on the minimum TS at j clockwise rotations from the current step (with wrap-around). If this $TS > \alpha$ condition was true, the function returned α otherwise it returned $-\alpha$.

Hence, the TriCmp function typically returned a negative when the minimum deviation at the current rotation was relatively high or when deviations on other lines were relatively low. This function, therefore, aided the fitness measure to drive the evolution towards discovering when the line configuration comprised a single relatively low deviation.

3.3 GP results

GP was trained on pixel data for the geometric centres of 598 vehicles that were taken from 28 IRLS images that corresponded to various altitudes. The vehicles varied greatly in appearance, e.g. vehicles in low altitude images contained the inner detail of windscreens, roofs, etc. whereas vehicles in high altitude images appeared more uniform; some vehicles appeared as bright objects on dark backgrounds whereas others appeared as dark objects on bright backgrounds. Furthermore, the images comprised a diverse set of environments: industrial, docklands, residential, rural, etc.

GP runs were conducted for various line configurations corresponding to different combinations of D and L. Twenty different randomiser seeds were used for each

Table 4: Distribution of θ_e (degrees) and n_c for the training set.

D	L	θ_e			n_c	
		av.(runs)	av.	s.d.	av.	s.d.
1.4	0.6	32.5	28.5	17.4	6	3
1.4	0.7	29.6	25.1	15.5	6	2
1.4	0.8	24.9	23.8	18.7	3	1
1.4	0.9	28.3	24.0	16.2	6	2
1.6	0.6	23.4	20.5	16.4	4	2
1.6	0.7	23.0	19.3	14.2	3	1
1.6	0.8	23.9	20.2	12.4	7	3
1.6	0.9	24.1	20.8	15.9	4	1
1.8	0.6	23.7	20.8	17.1	4	1
1.8	0.7	23.2	19.5	19.0	3	1
1.8	0.8	24.0	20.9	13.5	8	5
1.8	0.9	24.5	22.8	16.3	4	2
2.0	0.6	28.9	26.2	15.9	7	3
2.0	0.7	29.3	25.9	14.3	9	5
2.0	0.8	30.3	23.9	14.5	6	2
2.0	0.9	32.8	25.5	12.3	9	3

Table 5: Distribution of θ_e (degrees) and n_c for the test set.

D	L	θ_e		n_c	
		av.	s.d.	av.	s.d.
1.4	0.6	19.9	17.6	6	2
1.4	0.7	18.5	15.8	5	3
1.4	0.8	16.6	18.1	3	1
1.4	0.9	17.1	16.8	5	2
1.6	0.6	14.1	16.1	4	1
1.6	0.7	13.8	16.6	3	1
1.6	0.8	14.7	13.8	6	2
1.6	0.9	15.7	14.8	5	2
1.8	0.6	14.0	14.2	6	2
1.8	0.7	13.9	15.9	4	2
1.8	0.8	14.6	14.1	9	3
1.8	0.9	15.8	15.6	5	2
2.0	0.6	20.3	15.1	5	2
2.0	0.7	19.2	12.9	6	2
2.0	0.8	18.9	15.2	4	2
2.0	0.9	19.0	13.1	8	4

configuration and the resulting average θ_e are shown in the "av.(runs)" column of Table 4. These averages were computed from the lowest θ_e for each GP run for a given line configuration. It can be seen that the average θ_e was lowest when $D = 1.6$ or 1.8VW and $L = 0.7$. Larger values may have caused the lines to intersect boundaries of neighbouring objects (particularly because vehicles tend to be parked in rows or close to buildings), but smaller values may have undesirably prevented lines from crossing vehicle boundaries. Note that preliminary experiments investigated a broader range of line configurations and found the best range to be that shown in the table.

The results for the trees which gave the minimum θ_e for each line configuration are also shown in Table 4. The θ_e column gives the average and standard deviation of the orientation error over all vehicles. The n_c column gives the average and standard deviation of the number of orientation calculations that were spawned per vehicle, i.e. the number of rotation steps at which the tree returned a non-negative value for a single vehicle. These results confirm the aforementioned optimal line configurations.

The best trees for the training set were applied to a test set of 128 vehicles taken from 18 further IRLS images. These images corresponded to different altitudes and flight missions. Table 5 shows the performance of these trees, giving the average and standard deviation of θ_e and n_c over all vehicles in the test set. The θ_e values in Table 5 are lower than those in Table 4 for the following reasons. The θ_e value for most vehicles on training and testing was well below the average θ_e, but some vehicles (typically 10%) yielded θ_e greater than twice the average. Observation showed that these vehicles included inner detail (e.g. strong roof boundaries) which caused the method to prefer the minor axis to the major axis, thus producing a θ_e value which tended towards 90 degrees. This explains the considerable standard deviations in θ_e in the tables. Furthermore, the training set was considerably larger than the test set and it comprised a greater variety of vehicles and vehicle contexts.

Although there was no explicit restriction on the number of rotation steps at which a tree could spawn an orientation calculation, Tables 4 and 5 show that trees seldom spawned more than 6 calculations per vehicle thus minimising an accumulative orientation error. Pixel uniformity along the lines in the triangular configuration was likely to occur at more rotation steps when D was relatively high (i.e. the lines were further from the vehicles) or when L was relatively low (i.e. the lines were short). There is slight evidence for this in the tables where n_c tends to be greater for higher D and lower L.

The second-order central moments method described in Section 1 was applied to the training and test sets and the results are presented in Tables 6 and 7 respectively. The moments were calculated over a disc centred on each vehicle and the average and standard deviation of the resulting θ_e over all vehicles were com-

Table 6: Second-order central moment results using various disc diameters (× VW): average and standard deviation of θ_e (degrees) across all vehicles in the training set.

diameter	av. θ_e	s.d. θ_e
1.0	39.1	30.4
1.2	33.2	30.7
1.4	28.3	31.0
1.6	26.4	31.6
1.8	26.4	32.4
2.0	28.5	33.2
2.2	31.3	33.5
2.4	34.1	33.8
2.6	38.0	33.6
2.8	41.4	33.3
3.0	44.9	33.1

Table 7: Second-order central moment results using various disc diameters (× VW): average and standard deviation of θ_e (degrees) across all vehicles in the test set.

diameter	av. θ_e	s.d. θ_e
1.0	53.4	34.9
1.2	46.6	35.6
1.4	27.4	31.1
1.6	21.7	29.4
1.8	17.7	25.8
2.0	15.1	23.0
2.2	14.9	22.1
2.4	15.3	22.5
2.6	16.4	23.1
2.8	18.4	24.9
3.0	23.1	26.2

puted. This procedure was repeated using different disc diameters. The tables show that the disc diameter had to exceed 1.2VW in order to capture substantial information about the vehicles' shape. Comparing these tables with Tables 4 and 5 shows that the average orientation error was greater for the moments method (with an optimal disc diameter of approximately 2VW) than for the GP method (with an optimal line configuration of $D = 1.6$VW and $L = 0.7$). Furthermore, the GP trees gave a lower standard deviation in θ_e across the diverse sets of vehicles thus the GP method generalised better than the moments method.

4 CONCLUSIONS

A novel method for orientation detection is described, based upon the analysis of pixel-value deviations along a set configuration of lines. GP was used to discover the optimal processing of various line configurations in order to detect the orientation of vehicles in IRLS imagery. The associated error was lower than that obtained when using second-order central moments to derive the principal axis. Furthermore, the GP method generalised better across the diverse vehicles. This suggests that the shape of objects as defined by their boundaries is more important for the current task than the shape as defined by overall brightness.

Note that the new method does not require edge or line detection explicitly. This is an advantage for analysing objects in IRLS imagery due to the blurring of edges as a result of noise, jitter and inadequate resolution. Furthermore, the method does not require an object's precise centre of mass or geometric centre to be known as it simply depends on the relative uniformity of lines; whether a particular line falls on or outside an object is not important.

Further work will investigate the method's performance when applied to multiple points per object. This work is expected to involve a point clustering scheme in order to prioritise orientation hypotheses for a single object.

References

M. Daida, T. F. Bersano-Begey, S. J. Ross and J. F. Vesecky (1996). Computer-assisted design of image classification algorithms: dynamic and static fitness evaluations in a scaffolded Genetic Programming environment. In John R. Koza, David E. Goldberg, David B. Fogel and Rick L. Riolo (ed.). *Genetic Programming 1996: Proceedings of the First Annual Conference*, pp. 279-284, Cambridge, MA: The MIT Press.

D. Howard and S. C. Roberts (1999). A Staged Genetic Programming Strategy for Image Analysis. *Proceedings of the Genetic and Evolutionary Computation Conference*, pp 1047-1052, Orlando, Florida, July 1999: Morgan Kaufmann.

D. Howard, S. C. Roberts and R. Brankin (1999). Evolution of Ship Detectors for Satellite SAR Imagery. *Proceedings of the Second European Workshop in Genetic Programming*, pp 135-148, Göteborg, Sweden, May 1999: Springer LNCS.

J. R. Koza (1992). *Genetic Programming.* Cambridge, MA: The MIT Press.

W. B. Langdon and R. Poli (1998). Fitness Causes Bloat: Mutation. *Proceedings of the First European Workshop on Genetic Programming*, pp 37-48.

R. Poli (1996). Genetic Programming for Image Analysis. In John R. Koza, David E. Goldberg, David B. Fogel and Rick L. Riolo (ed.). *Genetic Programming 1996: Proceedings of the First Annual Conference*, Cambridge, MA: The MIT Press.

S. C. Roberts and D. Howard (1999). Evolution of Vehicle Detectors for Infrared Line Scan Imagery. *Joint Proceedings of the European Workshop on Evolutionary Image Analysis and Signal Processing and Telecommunications*, pp 111-125, Göteborg, Sweden, May 1999: Springer LNCS.

J. P. Rosca (1996). Generalty versus size in Genetic Programming. In John R. Koza, David E. Goldberg, David B. Fogel and Rick L. Riolo (ed.). *Genetic Programming 1996: Proceedings of the First Annual Conference*, pp 381-387. Cambridge, MA: The MIT Press.

T. Soule and J. A. Foster (1997). Code size and depth flows in genetic programming. In Koza, Deb, Dorigo, Fogel, Garzon, Iba and Riolo (ed.). *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pp 313-320.

W. A. Tackett (1993). Genetic Programming for feature discovery and image discrimination. *Proceedings of the Fifth International Conference on Genetic Algorithms*: Morgan Kaufmann.

J. F. Winkeler and B. S. Manjunath (1997). Genetic Programming for Object Detection. In Koza, Deb, Dorigo, Fogel, Garzon, Iba and Riolo (ed.). *Genetic Programming 1997: Proceedings of the Second Annual Conference*.

Z. Xiaoqi and Y. Baozong (1995) Shape Description and Recognition using the High Order Morphological Pattern Spectrum. *Pattern Recognition* 28(9), pp 1333-1340.