

# Optimising Object Recognition Parameters Using a Parallel Multiobjective Genetic Algorithm

F. J. Aherne, P. I. Rockett and N. A. Thacker

Dept. of Electronic and Electrical Engineering,

Sheffield University, Mappin Street,

Sheffield, S1 3JD, UK.

## Abstract

*This paper describes application of a Multiobjective Genetic Algorithm (MOGA) to optimise the selection of parameters for an object recognition scheme. The MOGA applied uses Pareto-ranking as a means of comparing individuals over multiple objectives. In order to prevent premature convergence heuristics were added to the algorithm to encourage speciation. The population consisted of sub-populations, whose members were able to migrate to any other sub-population, thus following the 'island' population model. Prior to this work the Pairwise Geometric Histogram (PGH) object recognition paradigm required the user to manually select histogram parameters - a process involving some degree of experience with the recognition scheme. Here, through the application of a MOGA we optimise and consequently automate parameter selection. The overall result of the algorithm is to select PGH parameters giving a more compact efficient histogram representation.*

## 1.0 Introduction

One of the primary requirements of a generic object recognition system is insensitivity to large changes in the image data due to occlusions and scene clutter. Image features such as corners, and line segments provide a good basis for object recognition, because they can be recovered over a wide range of viewing conditions and object transformations. These types of features also lead to successful recognition under conditions of scene occlusion and clutter because they are defined locally. Although image features can be characterised to some extent by intrinsic attributes such as local image gradients and curvatures the context of the surrounding shape geometry provides the basis for a much more powerful descriptor. By selecting appropriate parameters and storing these measurements in the form of a frequency histogram, a concise shape descriptor which promotes robust feature classification can be produced. This histogram is referred to as a pairwise geometric histogram because it records geometric measures made between pairs of image features.

Built into the PGH paradigm are a set of parameters concerning the histogram type and the blur applied to the axes of the histograms. Presently some of these

parameters are decided using rules of thumb and trial-and-error. This paper describes the application of a MOGA in order to determine these parameters in a principled manner. The significance of this work is that it enables the process of pairwise object recognition to be fully automated so that it can reach the full potential for use on large databases.

## 2.0 Pairwise Object Recognition

Pairwise geometric histogram (PGH) based algorithms have been shown to be a robust solution for the recognition of arbitrary 2D shape in the presence of occlusion and scene clutter [1], [2], [5] and [6]. The method is both statistically founded and complete in the sense that a shape may be reconstructed from its PGH representation ([7]).

Pairwise geometric histograms (PGHs) are a representation used for the recognition of rigid shape. The complete algorithm comprises a number of stages:

- Model image data (during training) and scene image data (during recognition) are processed by an edge extraction algorithm and the edge data are approximated by line segments. (Treating individual edgels would be equivalent and line approximation is mainly done for speed).
- Model histograms (during training) and scene histograms (during recognition) are constructed for *each* line segment (*reference line*) by comparing the reference line to all other lines and making entries into a histogram according to the measured relative angles and perpendicular distances (see Figure 1). To account for errors in the measurement processes and to encode the variability in the way a shape may be segmented into lines, entries are blurred appropriately when being placed into the histogram. This representation encodes local shape geometry in a manner which is invariant to rotation and translation and is robust to missing data, line fragmentation and clutter.
- Scene line labelling is performed by finding good matches between scene histograms and model

histograms using the Bhattacharyya metric. This statistical metric has been shown to be appropriate [13] as PGHs are pdfs of local shape geometry. This similarity measure adds extra robustness for matching sparse PGHs (containing many zero entries), as the metric is multiplicative. It is important to note that each reference line corresponds to a *single* PGH.

- Object classifications are confirmed by finding consistent labelling within a scene using a probabilistic Hough transform, as described in [14].

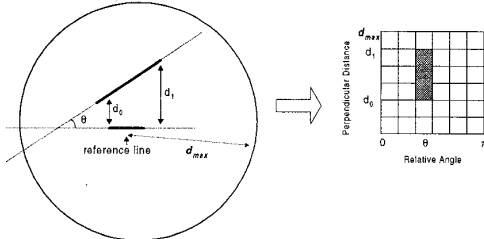


Figure 1: PGH entry for a Single Line Comparison

## 2.1 Pairwise Parameters to be Optimised

When constructing PGHs a choice is made as to the range of the relative angle  $\theta$  and perpendicular distance  $d_{\max}$  as shown in Figure 1. By virtue of the measurements recorded in PGHs, this shape representation is invariant to rotations and translations of the shape data. Depending upon exactly how angle and distance measures are defined, other invariance characteristics may also exist.

Prior to constructing geometric histograms it is necessary to decide on the histogram scale to be used. The choice of maximum perpendicular distance,  $d_{\max}$  is driven by two conflicting requirements. On one hand  $d_{\max}$  should be small enough that the PGH represents local shape and is robust to missing data and occlusion. On the other hand  $d_{\max}$  should be large enough so that shape information is present in each PGH so that they are distinct from each other. Prior to this work a rule-of-thumb used was to ensure that most of a shape is encoded into each histogram.

The simplest type of histogram is constructed by restricting angles to the range  $0$  to  $\pi$  and distances to  $0$  to  $d_{\max}$ . This histogram is invariant to reflections of the shape data about the reference line and is described as *mirror symmetric*. Mirror reflection invariance is not always desirable and can be removed by using the direction of angles (clockwise or anti-clockwise) to extend the range of angle measurements to  $-\pi$  to  $\pi$ . This doubles the area of the histogram which in turn doubles the computation needed for histogram matching but also increases the sparseness which improves robustness of matching in cluttered scenes. The area of the histogram

can be doubled again by directing the reference line towards the point where the line pair intersect and using this to define a reference frame. Measures of distance can then be signed depending upon whether they are to the left or right of the directed reference line, extending the distance range to  $-d_{\max}$  to  $d_{\max}$ .

As the histogram size increases so does the amount of information contained, and the more reliable the resulting matches. This is achieved at the expense of a loss of invariance characteristics regarding the expected behaviour of the data and/or an increase in the required computation for matching. Objects with mirror symmetry for example, should be represented with a ‘mirrored’ representation in order to reliably identify component parts of the object in mirrored examples. At present these histogram types are decided upon by the user, who is left to decide the trade-off between accuracy, suitability and speed.

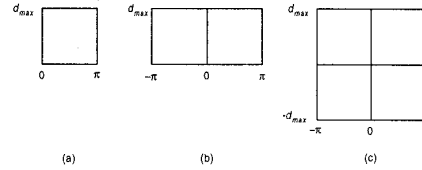


Figure 2: Illustration of Histogram Types

Typically there are three *types* of PGHs used (Figure 2):  $\{(0, d_{\max}), (0, \pi)\}$ ,  $\{(-d_{\max}, d_{\max}), (0, \pi)\}$  and  $\{(-d_{\max}, d_{\max}), (0, 2\pi)\}$  known as ‘mirrored’, ‘rotated’ and ‘directed’ histograms respectively, which encode different degrees of geometric invariance. A fourth alternative ‘contrast’  $\{(-d_{\max}, d_{\max}), (0, 4\pi)\}$  is also possible but will not be discussed further in this paper. In order to add robustness to the PGHs a local circular window is applied centred around the mid-point of the reference line. The associated histogram is constructed by making entries for each line truncated within the circular window. The radius of this window is a *continuous* variable and equivalent to the maximum perpendicular distance  $d_{\max}$  value of the associated histogram. The  $d_{\max}$  parameter is normally set in agreement with the object size so that the maximum amount of useful data is constructed during matching. Its determination is a trade-off between the local scale of the object and the amount of clutter in the window. Associated with this parameter is the size of the bin widths on the perpendicular distance axis of the PGHs. In this work we will simultaneously optimise the value of  $d_{\max}$  and the histogram *type* but bin width will remain constant.

PGHs have been shown to be applicable to extremely large databases for hundreds of thousands of objects [6]. The results in [6] are based on an extrapolation of storage capacity from a much smaller test set. In order to construct a large recognition system automatically we will

require an automated procedure for selecting the parametric representation of a line in terms of the free parameters specifying a PGH - the very nature of this problem requires us to take multiple competing objectives into account.

### 3.0 Multiobjective Genetic Algorithms

Several methods for adapting GAs to cope with the simultaneous optimisation of a problem over a number of dimensions have been proposed, including the use of Pareto-ranking, [10]. The MOGA applied in this work uses Pareto-ranking as a means of comparing solutions across multiple objectives. The Pareto-optimal set of a multiobjective optimisation problem consists of all those vectors for which their components cannot be all simultaneously improved without having a detrimental effect on at least one of the remaining components. This is known as the concept of *Pareto optimality* [10], and the solution set is known as the *Pareto-optimal set* (or *non-dominated set*).

In the sense that they are able to solve NP-complete search problems, genetic algorithms (GAs) have been recognised to be well-suited to multiobjective optimisation as described in [11] and [12]. MOGAs do not impose an ill-informed weighting process on the task of selecting a single optimal solution but instead the concept of pareto-ranking can be applied in order to deliver a set of candidate solutions optimised for different combinations of criteria. If, however, one particular solution can be found which dominates all others then this solution would have been the optimal choice for any combined optimisation procedure with arbitrary weighting. In the absence of a dominant solution all candidate solutions are ranked in terms of pareto dominance.

### 3.1 Justification for Using a MOGA

Advantages of GAs have already been used by computer vision researchers examples of which are described in [8] and [9]. GAs are particularly applicable to the problem addressed in this paper, as our search space is noisy, discontinuous, impossible to search exhaustively and contains no derivative information. In order to explain the selection of our objective functions it is useful to recall the motivation for application of MOGAs. For a given scene or model line we have a choice as to the parameters for the associated PGH. Our algorithm will optimise these parameters producing *optimal* sets of lines relative to all others in the database.

### 3.2 The Objective Functions

In what follows  $B$  denotes the Bhattacharyya match score between two histograms and is defined as:

$$B = \sum_{x=0}^{n_\theta} \sum_{y=0}^{n_d} \sqrt{H_1(x, y)} \sqrt{H_2(x, y)}$$

where  $H_1$  and  $H_2$  are two PGHs of the *same type*,  $n_\theta$  the number of bins on the  $\theta$  axis and  $n_d$  the number of bins on the perpendicular distance axis.  $B$  is used as a measure of similarity between two histograms - two identical histograms yield  $B=1$ , (assuming the histograms are normalised) and two completely different histograms give  $B=0$ .

Suppose we have a set of PGHs, we want a principled technique for determining the relative distinctiveness of each of the histograms in the set. For a given PGH belonging to such a set this can be achieved by calculating the match scores between our given PGH and *all* others in the set. The distance between the identical match  $B=1$  and the mean of all the other matches ( $B_\mu$ ) will achieve our aim, as the greater this quantity, the more distinct our given PGH relative to the rest of the set, hence our first objective,  $f_1$ , is:

$$f_1 = \max(1 - B_\mu)$$

(other descriptive statistics such as the median were tried and gave similar results to the mean). In addition, we want a measure that reflects histogram consistency across different examples of a given object subject to variable segmentation/fragmentation.

We will consider different sets of data defined as follows:

- Sets of objects consisting of a fixed known object together with variants (for example different levels of segmentation): a *variant set*.

So for a given histogram this measure will tell us which of the variant sets that particular PGH was most likely to have come from. We can achieve this by calculating the matches between our given histogram and all PGHs in each variant set (these PGHs are all constructed with the same *fixed* parameters, see section 3.5). Recall, each variant set consists of a set of objects. The best match between our given PGH and all the PGHs for *each* object in *each* variant set is stored (the inner summation of  $f_2$ ) and then these best matches are summed for each variant set. The variant set with the highest sum is the winner and the given histogram is deemed to have come from the winning variant set. This avoids the difficult problem of labelling corresponding lines in the database. Thus:

$$f_2 = \sum_{j=1}^N \max_j \left( \sum_{i=1}^{n_j} \max(B_i) \right)$$

(where  $n_j$  is the number of objects in variant set  $j$  and  $N$  is the number of variant sets).  $f_2$  is computationally

intensive as it involves computing the matches between all histograms in the *variant* sets for every member of our GA population.

Finally we desire as compact a representation as possible and therefore want to minimise the area of the histograms. For this work the bin width of the histograms is fixed and the only variable parameters for area are the perpendicular distance,  $d_{\max}$  (which is also the radius of the circular window used for constructing the PGH) and the type. So the final objective function minimises the histogram area. A scale factor of 1, 2 and 4 for mirrored, rotated and directed histograms respectively (Figure 2) was used reflecting the associated area, (since rotated histograms have twice the area of mirrored and directed twice the area of rotated). Hence we have:

$$f_3 = \min(\text{histogram area})$$

### 3.3 The Representation

In our encoding scheme an individual is represented as shown below in Figure 3. The **HISTOGRAM** in the above encoding is a PGH constructed from a reference line belonging to some object in a variant set. The **TYPE** and  $d_{\max}$  refer to the histogram, being the histogram type and maximum perpendicular distance respectively.

HISTOGRAM	TYPE	$d_{\max}$
-----------	------	------------

Figure 3: Encoding Scheme for an Individual

### 3.4 Algorithm Description

The population consisted of sub-populations each of which was stored as an *array* of individuals. Figure 4 below, shows pseudo code for the MOGA. The number of individuals in each sub-population was equal and did not change. Initially the sub-populations were randomly generated and each individual assigned a fitness based on its pareto-rank. Fitness was then assigned by taking the compliment of the pareto-rank with the sub-population size - for example, suppose our sub-population was of size 10 and we had one-non-dominated individual (assigned Pareto-rank 1) then its fitness would be 9, and so on. By summing the fitness across all individuals within a sub-population we could assign a fitness that sub-population. The sub-populations were then *randomly* paired for crossover and mutation to be performed, producing two new sub-populations. The individuals in the new sub-populations were evaluated and similarly the new sub-populations were assigned a fitness. Initially Stochastic Universal Sampling, (as described in [15]) was used for deciding which sub-populations to keep. It was found that a more efficient technique for selection was simply to compare the fitness of the old sub-populations (prior to crossover and mutation) and the newly created sub-populations. If the new sub-population was fitter, it

was kept and the old one disregarded (and vice-versa). Finally for each sub-population the individuals were sorted in descending order according to their fitness (Grouping() in Figure 4), the reason for which is now described as we discuss the crossover and mutation operators.

```

for(run=0;run<max_runs;run++)
{
    Initpop();
    for(gen=0;gen<max_gen;gen++)
    {
        Crossover();
        Mutation();
        Pareto();
        Fitness();
        Grouping();
        Selection();
    }
}

```

Figure 4: Pseudo Code for MOGA

### 3.5 Crossover and Mutation Operators

In order for crossover to function as desired we needed to ensure that two good sub-populations produced fit/fitter offspring, the mating restrictions required for this are now explained. Crossover was performed by randomly selecting a pair of sub-populations and two crossover points (corresponding to rows of the arrays). Suppose we had  $N$  individuals in each sub-population. Two crossover points were used, the first was  $N/2$  and the second was generated randomly. The range of the random crossover point was restricted to lie in  $(N/2, N)$ , due to the Grouping() function of Figure 4, this ensured at least 50% of the best individuals were always kept. Two new sub-populations were then produced by replacing the bottom 50% of sub-population 2 with the top 50% of sub-population 1, and similarly for the random crossover point. Different crossover points were used to increase the diversity of the sub-populations.

Mutation consisted of creating a sup-population of randomly generated individuals and calculating the Bhattacharyya score between each member of that random set with every member of the sub-population (Figure 6).

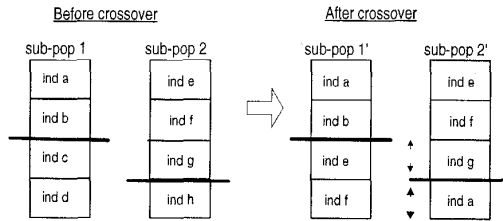


Figure 5: Crossover Operator used

For a given individual of the random set, the match score with every member of the sub-population was calculated and if *all* scores were less than the corresponding individuals fitness on the first objective then the random individual replaces the bottom-most individual in the sub-population (and the replacement continues upwards). This ensures that the population diversity is maintained and that premature convergence did not occur. This process could also be considered as goal-attainment on the first objective.

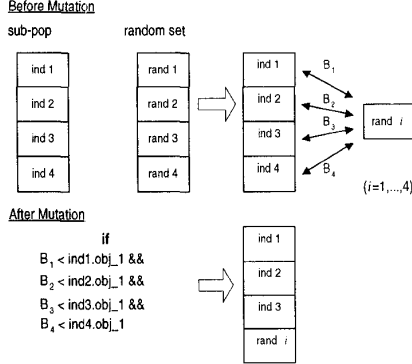


Figure 6: Mutation Operator used

### 3.6 Algorithm Modifications

The second objective function,  $f_2$ , is computationally intensive as it involves computing the matches between *all* histograms in the variant sets. In order to speed up the algorithm an exhaustive match for a given histogram with all histograms (constructed using fixed parameters, here we used  $d_{\max} = 50.0$  and 'rotated' histograms) in the variant sets was only calculated on the *initial* run; thereafter a random histogram is selected from a given variant set, an the sum of match scores is calculated. If for any given model the match score is greater than that calculated from the initial exhaustive search, that histogram replaces the existing one. Thus we relied on the iterative nature of the MOGA to perform a stochastic search.

### 4.0 The Experiments

The four data sets used are shown in Figure 8. In each of the sets the leftmost object is that shown in Figure 7. The first set consists of variants having the original line length shortened or lengthened by a different scale factor.



Figure 7: Test Objects Used

The second set consists of six objects the first is the original stegosaurus, the following two are the first with increasing levels of Gaussian noise added. The last three stegosaurus objects are the first three mirrored around the midpoint of the objects length. The third data set consists of objects having increasing levels of Gaussian noise added to the lines. Finally the triceratops set contains objects all having different levels of segmentation, giving a polygonisation effect.

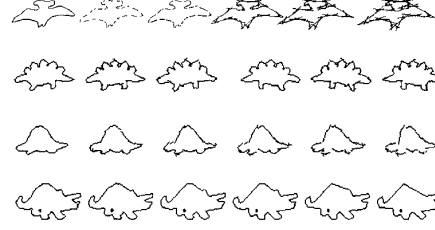


Figure 8: Data Sets 1-4

### 5.0 Results and Discussion

The MOGA applied consisted of 10 sets of 58 individuals and was run for 100 generations. Figure 9(a) shows a distribution of match scores for a randomly generated initial population of PGHs. This data was generated by calculating the match scores between all histograms having a similar type and then binning the results. The data distribution shown in Figure 9(a) has mean = 0.415 and variance = 0.013. Similarly, the histogram shown in Figure 9(b) shows a typical distribution of match scores for a final population, this distribution has mean = 0.275 and variance = 0.015.

By comparing Figure 9(a) and Figure 9(b) we can see that the effect of the MOGA is to produce histograms that have, on average, a lower match score across their associated populations, confirmed by the lower mean of Figure 9(b). This is due to the first objective function optimising on histogram distinctiveness, and thus the MOGA has produced sets of PGHs containing less redundant information than the initial populations. It can also be seen by comparing Figures 9(a) and (b) that the MOGA has reduced the number of PGHs having high match scores, this is another indication of how the amount of redundancy in the representation has been reduced and the distinctiveness increased. The overall effect of the MOGA is that it produces sets of PGHs, (consisting of individuals with associated pairwise parameters), that represent the line data more efficiently.

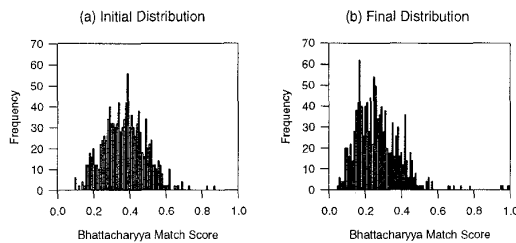


Figure 9: Match Score Distributions

## 6.0 Conclusion

In this work we have applied a Pareto-based MOGA to parameter selection for the PGH object recognition paradigm. Heuristics were applied in the crossover and mutation operators to avoid speciation and the need for sharing. By carefully selecting our data sets we have shown that the optimisation technique works irrespective of many common problems such as noise, missing data and segmentation. The overall effect of applying the MOGA is the optimisation of a set of pairwise parameters which are at present decided by the user. We are therefore making the construction of PGHs a fully automatic, optimised process. The algorithm produces distinct, efficient histogram definitions for subsets of lines in our database. In future work we intend to investigate the incremental learning capacity of the system, by successively adding new objects to the database. This is not a trivial problem as the set of optimal histograms is defined recursively by all others in the database. Finally we will examine the effect of adding clutter to the objects in the database.

## References

1. A. C. EVANS, N. A. THACKER, J. E. W. MAYHEM, The Use of Geometric Histograms for Model-Based Object Recognition, *Porch. 4<sup>th</sup> BMVC*, Guildford, pp 429-438, September 1993.
2. A. C. EVANS, N. A. THACKER & J. E. W. MAYHEM, Pairwise Representation of Shape, *Proceedings 11<sup>th</sup> Int. Conf. Pattern Recognition (ICPR92)*, The Hague, Netherlands, vol. 1, pp 133-136, 1992.
3. D. E. GOLDBERG, *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley, 1989.
4. L. DAVIS, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
5. A. P. ASHBROOK, N. A. THACKER & P. I. ROCKETT, Multiple Shape Recognition Using Pairwise Geometric Histogram Based Algorithms, *Proceedings 5<sup>th</sup> IEE Int. Conf. Image Processin & Applications (IPA95)*, Edinburgh, 1995.
6. A. P. ASHBROOK, N. A. THACKER & P. I. ROCKETT, Pairwise Geometric Histograms: A Scalable Solution for Recognition of 2D Rigid Shapes, *Proceedings 9<sup>th</sup> Scandinavian Conf. Image Analysis (SCIA95)*, Uppsala, Sweden, vol. 1, pp 271-278, 1995.
7. N. A. THACKER, P. A. RIOCREUX & R. B. YATES, Assessing the Completeness Properties of Pairwise Geometric Histograms, *Image & Vision Computing*, Vol. 13, no. 5, pp 423-9, June 1995.
8. G. ROTH & M. D. LEVINE, Geometric Primitive Extraction using a Genetic Algorithm, *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 16(9):901-905, September 1994.
9. A. HILL, & C. J. TAYLOR, Model-based Image Interpretation using Genetic Algorithms, *Image & Vision Computing*, 10(5):295-300, June 1992.
10. C. M. FONSECA & P. J. FLEMING, An Overview of Evolutionary Algorithms in Multiobjective Optimisation, *Evolutionary Computing*, 3(1):1-16, 1995.
11. C. M. FONSECA & P. J. FLEMING, Genetic Algorithms for Multiobjective Optimisation: Formulation, Discussion and Generalisation, *Porch. of 5<sup>th</sup> Int. Conf. on Genetic Algorithms*, Sam Mateo, CA: Morgan Kaufmann, July 1993.
12. J. HORN, N. NAFPLIOTIS & D. E. GOLDBERG, A niched Pareto Genetic Algorithm for Multiobjective Optimisation, *IEEE Conf. on Evolutionary Computation*, Vol. 1, pp 82-87, 1994.
13. N. A. THACKER, F. J. AHERNE & P. I. ROCKETT, *The Bhattacharyya Metric as an Absolute Similarity Measure*, Submitted to Pattern Recognition, April 1996.
14. R. S. STEPHENS, A Probabilistic Approach to the Hough Transform, *Porch. BMVC 1990*, Oxford Univ., pp 55-60, September 1990.
15. J. E. BAKER, Reducing Bias and Inefficiency in the Selection Algorithm, *Porch. of the 2<sup>nd</sup> Int. Conf. on Genetic Algorithms*, Lawrence Erlbaum, 1987.
16. A. C. EVANS, Geometric Feature Distributions for Shape Representation and Recognition, Ph.D Thesis, AI Vision Research Unit, Sheffield University, January 1994.