Evolving Edge Detectors

Christopher Harris and Bernard Buxton
Department of Computer Science
University College, London
Gower Street
London
WC1E 6BT
United Kingdom
Email: C.Harris@cs.ucl.ac.uk, B.Buxton@cs.ucl.ac.uk
Telephone: Chris Harris +44 171 419 3679, Bernard Buxton +44 171 380 7294

Abstract

Edge detection is the process of detecting discontinuities in signals and images. We apply Genetic Programming techniques to the production of high-performance edge detectors for 1-D signals and image profiles. The method, which it is intended to extend to the development of practical edge detectors for use in image processing and machine vision, uses theoretical performance measures as criteria for the experimental design.

Category: Genetic Programming

Introduction

Edge detection is a classic problem in machine vision applications. It forms the first part of the process of converting a raster image to a symbolic description of the scene and for many other high-level tasks, such as stereo vision, segmentation, and object recognition and location. A large body of work exists devoted to the production of high quality detectors, and to what constitutes a good detector. We use genetic programming to produce functions that perform well on a set of synthetic and real-world test signals, using theoretical criteria to judge performance.

The purpose of this work is twofold: (i) to use existing theory as an inspiration for experimental design in Genetic Programming and (ii) to use GP as a technique for developing detectors that can give optimal (or near optimal) performance on real signals and images for which no analytic or simple numerical solution of the problem is possible. In this work we use edge detection theory to provide a fitness function for our experiments and utilise GP as a means of obtaining optimal detectors automatically, without user intervention. The development of such a facility would have important implications in machine vision applications where much human effort is often devoted to developing and tuning algorithms and techniques to particular applications.

The Edge Detection Task

Edge detection involves locating and highlighting local discontinuities in signals. This is usually achieved by convolving the signal with an operator (specified by some functional form over a given interval) to produce a response signal. For discretely sampled signals the operator is sampled to produce a filter. Most edge detectors are designed to respond strongly to areas of the signal with large first derivative, edges being marked by peaks in the response signal.

[Canny] was the first to produce a set of theoretical criteria for determining the performance of an edge detector. He proposed that a detector should have a good signal-to-noise ratio (SNR) so that it produces a strong response to an edge in the signal, and that it should have good localisation properties so that a detected edge is as close as possible to the position of

the ‘true’ edge. Finally, it should only give a single response to an edge, i.e. it should not respond to maxima in the signal due to noise.

For the first two of these criteria, Canny derived an expression in terms of integrals of the operator function and signal that would measure the performance of the operator with respect to each criterion. An optimal operator would maximise the product of the SNR and localisation criteria. Canny calculated the optimal operator for idealised step edge signals and presented the first derivative of a Gaussian function as an efficient approximation to this optimum (see Figure 10).

[Spacek] took Canny’s work and extended the theoretical performance measure to incorporate the third criterion of noise suppression. From this he derived an expression for an optimal operator as an equation depending on six parameters. Fixing two of these parameters and optimising the remaining four, Spacek found an optimal operator that performed better than Canny’s under slightly different boundary conditions. Spacek then presented a cubic spline function as a convenient approximation to the optimum.

[Petrou] presented further improvements by optimising Spacek’s expression in all six parameters. She then compared Canny’s approximation, Spacek’s approximation, Spacek’s optimum and their optimum on some real world images. Although theoretical performance was maximal for their optimal operator, in practice the slight performance advantage it had over Spacek’s cubic spline did not translate to a statistically significant improvement when applied to a real world image.

Details of the criteria used by Canny in deriving an optimal detector are presented in the section detailing the fitness function used in the GP runs.

Related Work

Applications of other automatic techniques to edge detection have produced promising results. These techniques were applied directly to images rather than the mathematics of the underlying detectors, and in two dimensions.

[Chao] used a Hopfield neural network to perform edge detection on a grey-level image. The whole image was mapped to a 2-D Hopfield net, one pixel to each neuron. Weights between neurons were defined in terms of contrast between the corresponding pixels and their distance apart in the image. The initial state of each neuron was proportional to the grey-level value of the pixel it represented. Results were found to be comparable to a Sobel [Gonzalez] operator on grey-level noisy images.

[Srinivasan] used a 2-stage network to detect edges. The first stage acted to encode and compress information from a receptive field (an area of image around the target pixel). The second stage took the output of the first stage and produced edge vector components as outputs. Experiments using a single training image showed that the detector performed as well as Canny’s operator and outperformed other operators such as the Sobel operator and a Laplacian of Gaussian operator.

[Bhandarkar] applied Genetic Algorithm optimisation techniques to edge detection in a grey-level image. Using 2-D chromosomes to represent ‘edge-images’, optimisation was performed simultaneously on a whole image using a complicated cost function based on region dissimilarity, and edge curvature, thickness, and fragmentation. Performance was compared with optimisation by Simulated Annealing, and a local decision-tree based approach.

Experimental Design

The aim of this experiment is to use GP to evolve a good edge detector, as specified by the criteria mentioned earlier. The selection pressure comes from maximising the performance of candidate edge detectors on a mixture of synthetic and artificial signals.

An edge detector consists of a function in a variable `x` over a specified interval. For this experiment, candidate functions are generated and sampled over the interval [-1,1] to give an array of values. These values are normalised to produce an operator. The operator is convolved with each training signal to produce a response signal. The candidate function trees are produced from the following functions and terminals.

Terminal Set

`x`	The x position of the candidate function. This will range from -1 to +1.
`0.0`	A constant.
`1.0`	A constant.
`2.0`	A constant.

Function Set

`+ - * /`	Arithmetic operators with protected division. Dividing by 0 returns 0.
`rlog`	Protected ln function. Takes absolute value of parameter.
`exp`	Exponent.
`pow`	Protected power function. Raises first parameter to the power of the second parameter. Only allows integer (positive or negative) powers of negative numbers, but any power of positive numbers.
`sin`	Sine function.
`cos`	Cosine function.

Training data

Forty 1-dimensional signals, of varying lengths, were produced for training data. Six of these were synthetically produced signals containing idealised 'step' edges of infinite gradient, as shown in Figure 2.

Since, in applications, edges often occur at a variety of scales, i.e. of characteristic widths and relating to objects at some characteristic level of detail, each of these idealised signals was smoothed with a Gaussian smoothing operator to produce a further six signals. The original signals were also corrupted with normally distributed noise, to produce a total of 18 synthetic signals. A further 22 signals were extracted from real-world images taken from a video camera (see Figure 1). Edges in these images were much less obvious and the signals were often noisy. The data thus consists of roughly equal portions of synthetic and real signals. Each signal contained multiple edges, the locations of which were marked by hand and recorded. A total of 102 edges were marked over all signals.

Figure 1. A real world, noisy signal, with corrupted edges

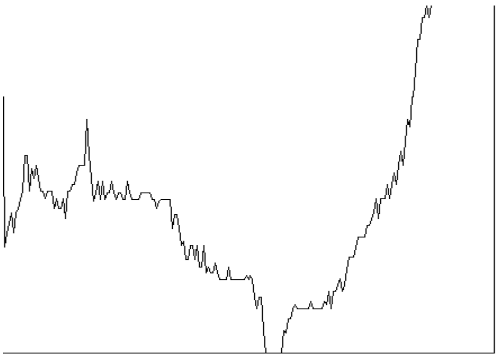
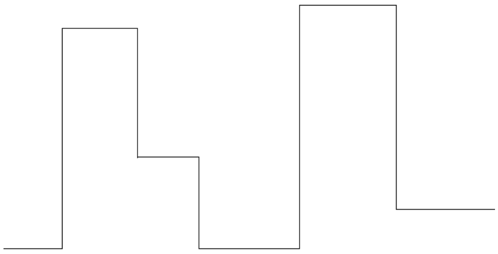


Figure 2. A signal with step edges



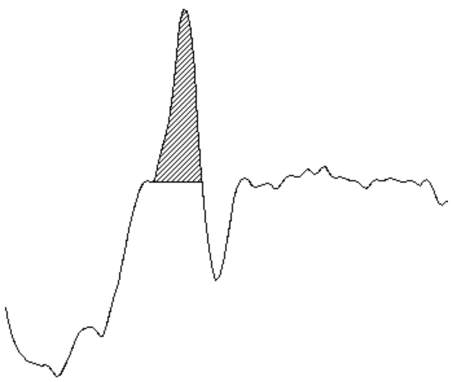
To reduce computational load, each generation was tested on a random sample of 10 of the 40 signals. This sample set changed after each generation.

The Fitness Function

Each candidate was assessed on three criteria for each signal. The criteria correspond to the criteria in [Canny]

Signal-to-noise ratio. A measure of the response of the operator to the edge as opposed to the rest of the signal. This is defined as the area under the nearest peak to the edge location. It is assumed here that there is some maximum in the response, a lack of a proximal peak is accounted for by the localisation criterion. For this component of the fitness function, the area under the peak is found by first locating a maximum in the response of a single or two pixels width. Then, the nearest trough or zero-derivative point of inflexion to this peak is found and is used to define the base level of the response, above which the response to the edge is measured. A point to the other side of the peak is then found at which the signal crosses the base level, giving the total extent of the peak. The size of response to the edge is the sum of responses in this area, over and above the base level, as illustrated in Figure 3. A good detector will have a large response.

Figure 3. Area captured by a peak in signal response



Localisation. This is a measure of the accuracy in positioning of the response. A good detector will mark an edge by a peak in response, close to the position of the edge in the original signal. A candidate's performance is measured by the displacement of the peak from the 'true' edge position, averaged over all edges in all signals in the training set. The 'true' position is that marked by hand by a human operator. A lower value here indicates a better performance.

Single response criterion. This criterion is restated to say that the operator should respond only to real edges and not respond to features near the edge caused by noise in the signal.

This can be measured quantitatively by finding the average separation of all peaks in the response of the operator to a signal consisting entirely of normally distributed noise. A larger separation of peaks means that the operator is less susceptible to high-frequency oscillations in the signal, and therefore gives fewer spurious peaks.

Each candidate was sampled at resolutions of 9, 13, and 17 pixels to give three different width operators that would respond to edges of different width. Each operator was applied to ten signals and the responses were analysed to give average response strength S(f), localisation error L(f), and noise suppression figures C(f).The overall fitness of each candidate is calculated as $\frac{S(f)C(f)}{L(f)}$.

For step-like edges, a good edge detector will normally be a combination of a smoothing operator (noise suppression measure) and a first-derivative operator (large response to change in intensity).

Tableau of edge detection problem

Objective:	To respond strongly and accurately to edges in a signal, and not to respond to non-edge sections.
Terminal Set:	`x 0.0 1.0 2.0`
Function Set:	`+ - * / rlog exp sin cos pow`
Fitness Cases:	10 1-D signals drawn randomly from a pool of 40.
Raw Fitness:	as detailed elsewhere
Standard Fitness:	(Max raw fitness) - raw fitness
Population size:	200
Mutation rate:	10%
Crossover rate:	60%
Terminal Probability:	45%
Mutation Terminal Probability:	50%

“Terminal Probability” and “Mutate Terminal Probability” control the average size of program trees in the population. For a fuller explanation, see [Harris]

Results

Six runs were made using the GP-COM system [Harris]. Each run produced an operator that performed well when compared to Canny’s Gaussian first-derivative approximation. Each run was terminated when an operator of sufficient performance was found, i.e. when an operator had been produced that exceeded the performance of Canny’s operator over the sample of 10 signals.. The operators’ performances are detailed in the table below, along with scores for Canny’s approximation. These are raw fitness scores, high values indicating better performance, and were obtained by applying the same fitness function as used during the run, but over the entire set of 40 signals. This allows a fair comparison of the operators, but means that the performance of each operator is necessarily worse than that achieved during the runs.

Operator	Response Strength	Localisation Error	Noise Suppression	Overall Fitness
1.1	41.36	0.3583	3.938	119.9
1.2	38.37	0.4217	4.242	114.5
1.3	41.76	0.6367	3.865	98.62
1.4	35.54	0.4900	4.285	102.2
1.5	33.68	0.4733	4.786	109.4
1.6	30.66	0.5167	4.876	98.56
Canny	39.57	0.4467	3.430	93.81

The operators are detailed in Figures 4-10. Each figure contains a plot of the evolved operator over the interval [-1,1], the program tree that produced the function, and the boiled-down functional form of the operator which eliminates the superfluous parts of the program tree.

Figure 4. Operator 1.1

$$\frac{1}{\cos(\cos(e^x)) - e^x}$$

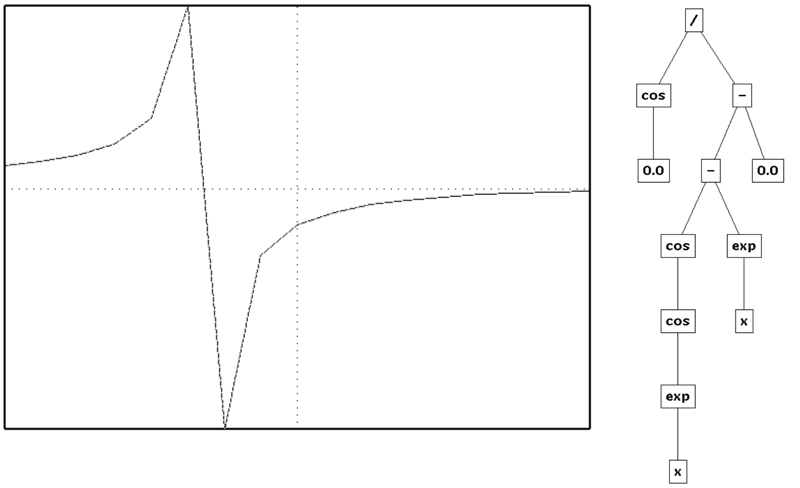


Figure 5. Operator 1.2

$$\frac{\sin(x)\,\mathrm{rlog}(x)}{x^x}$$

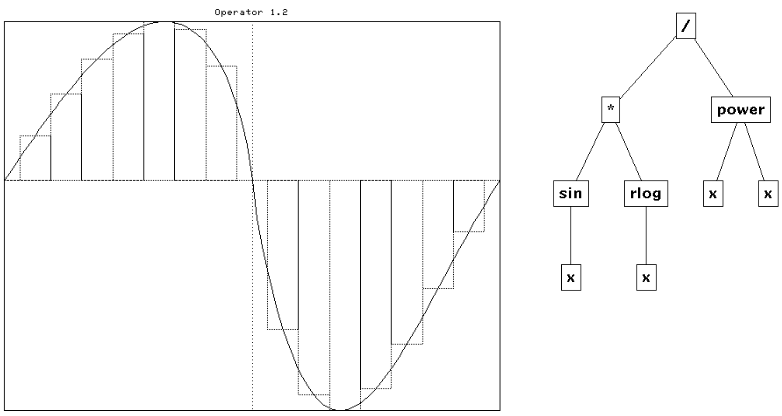


Figure 6. Operator 1.3

$$\frac{\sin(2)}{-x-\dfrac{\mathrm{rlog}(x+2)}{2-x}}$$

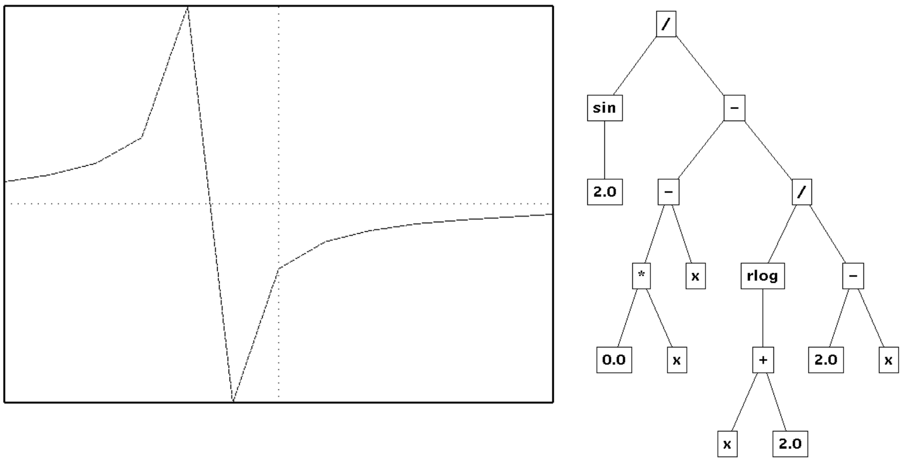


Figure 7. Operator 1.4

$$x\,\mathrm{rlog}(x)$$

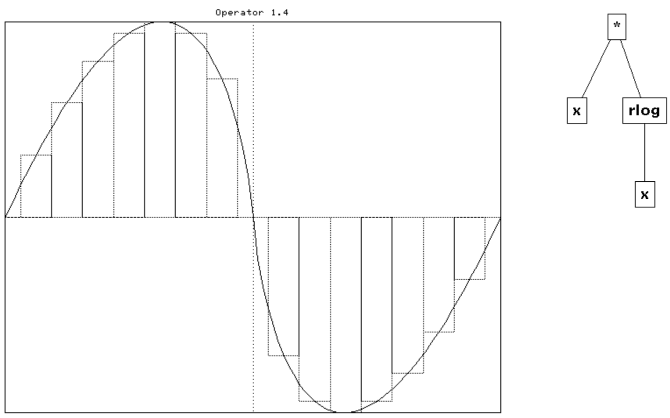


Figure 8. Operator 1.5

$$x\sin(\mathrm{rlog}(x))$$

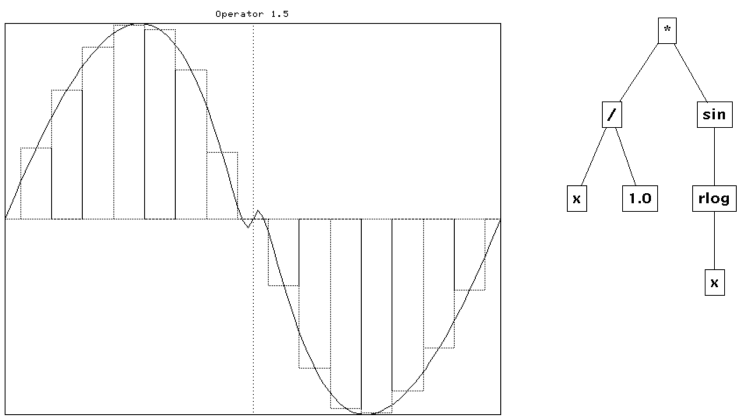


Figure 9. Operator 1.6

$$x - 2x\cos(x)$$

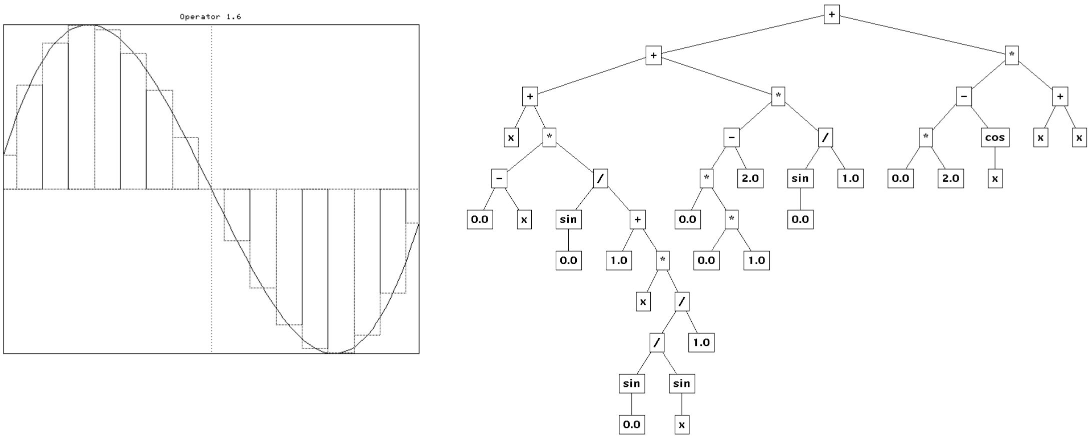
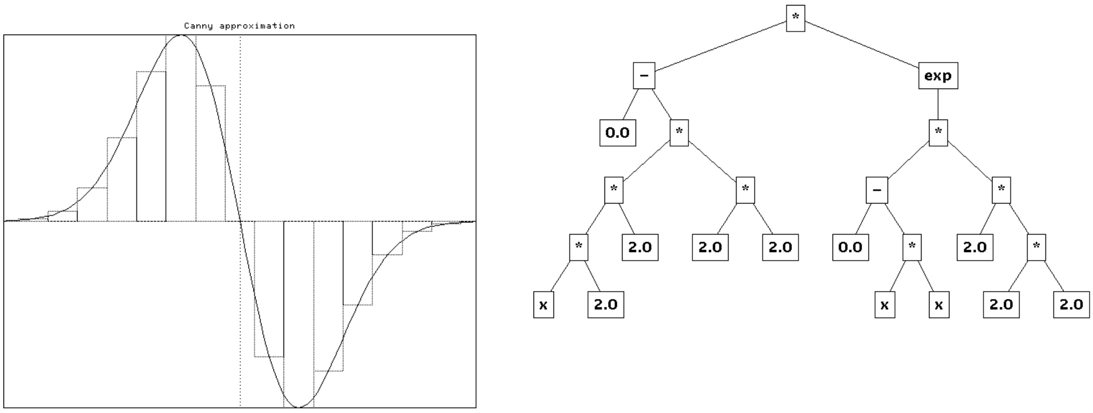


Figure 10. Approximation to Canny optimal operator with first derivative of Gaussian. For these experiments, σ was set to 0.25.

$$-\frac{x}{\sigma^2}e^{\frac{-x}{2\sigma^2}}$$



Operators 1.1 and 1.3 do not look like good edge detectors (i.e. not like the derivatives of Gaussian smoothing functions) when sampled at high resolutions. When sampled at lower rates however (as in Figures 4 and 6), they do approximate the shape of a Canny-like edge detector reasonably. The zero-crossing in the operator function occurs at about -0.3 in these two operators, which should indicate a poor localisation score (ideally it would be at 0). This is indeed the case for 1.3, but not so for 1.1 which has the best overall localisation score! This localisation error accounts for almost all the difference in total fitness between the two operators. The coarse-grained nature of the localisation criterion, with a resolution of a whole pixel and only around 100 edges in the entire training set, combined with a larger spread of influence in operator 1.3 over the interval [-0.3,1], is sufficient to explain the different performance of the two operators with regards to this criterion. These results indicate that it is worth emphasising the effect of sampling on the operators. Only when sampled coarsely do the operators become 'sensible'. These operators both have large discontinuities around their zero-crossings near -0.3, which are very evident at higher sampling rates. These discontinuities have the potential to dominate the make-up of the operator but the low sampling rate of the filters seems to reduce this effect considerably. Thus a high fitness should be taken in context - what is found is a function that makes a good detector *at the specified sample rates*.

The other four evolved operators (1.2, 1.4, 1.5, 1.6) look fairly similar. They have the shape associated with good edge detectors - a smooth curve, zero-crossing at 0, and low values at the extrema. The differences in response strength correspond quite well to the shape of the operators at their extrema. A good detector will go smoothly to 0 at -1 and 1, and stay there. This avoids a phenomenon called Gibb's ringing, a distortion of the resulting signal. Those operators which are flatter at the extrema, such as 1.1, 1.2, 1.3, and Canny, have the best response strength. This is because more of the influence of the operator is located toward the centre of the edge, and the peripheries are de-emphasised. A larger response to sharp edges is the result. The highest scores in this criterion were for operators 1.1 and 1.3. These operators are both asymmetric, and these high scores may be due to some corresponding asymmetry in the edges in the training set.

These four evolved operators with smoother profiles performed better with respect to noise suppression. The two criteria of good response and few false responses are to some extent mutually exclusive. A good response to first derivative involves an operator with high first derivative at its zero-crossing. This in turn implies a high-frequency aspect to the operator which is contrary to the requirements of a smoothing operator. This mutual exclusion principle was noted by Canny and means that maximising one property invariably involves reduced performance in the other. Hence the overall fitness is not based on maximising either value but their product. The Canny operator performs reasonably well on response strength but falls down when it comes to noise suppression. Increasing the value of σ here would increase noise suppression performance, and reduce response strength, by dilating the operator. The reduced performance of the Canny operator here can also be attributed to the combination of types of edge in the training data. Canny's approximation was to an optimal detector for step edges, yet step edges and smoothed step edges made up only half of the edges in the training set. It is thus to be expected that this function is not optimal for this data set.

Conclusions

We have shown an application of Genetic Programming to the evolution of operators designed to detect edges in 1-dimensional signals. The design of the experiment was based upon theoretical analysis of the ideal edge detector and its properties as, for example, first carried out in [Canny]. The fitness function was adapted from a set of criteria that a good edge detector would fulfil, and stated in terms of the responses of those detectors to a series of signals, in the digital domain. A number of detectors were found that out-performed Canny's approximation to an optimal detector, on a mixture of synthetic and real-world signals.

Future Work

It is intended to extend this work to the production of automatic detectors for a number of image features. For this purpose, a primitive set consisting of arithmetic operations is not particularly useful when considering operations on raster images. It is proposed to build a set of strongly-typed primitives that correspond closely to the domain of 2-dimensional images and allow genetic programs to be built out of sequences of image operations. Some work on images has already been done in [Johnson], [Teller], [Tackett], [Andre], but we intend to take a closer look at the issue of image representation. The chance of success in a GP application depends on the nature of the search space created, which in turn depends on the representations used to build the function and terminal set. It is crucial therefore to have a powerful and appropriate set of primitives available to GP. Representations to be considered include wavelet representations and the representational basis proposed in [Köenderink].

References

[Andre] Andre D. “Automatically Defined Features: The Simultaneous Evolution of 2-Dimensional Feature Detectors and an Algorithm for Using Them.” in *Advances in Genetic Programming,* Kenneth E Kinnear (ed).MIT Press. 1994.

[Bhandarkar] Suchendra M Bhandarkar, Yiqing Zhang, Walter D Potter. “An edge detection technique using Genetic Algorithm-based optimisation.” Pattern Recognition 27(9). 1994. pp 1159-1180.

[Canny] Canny J. “A Computational Approach to Edge Detection.” IEEE PAMI 8(6). 1986.

[Chao] Chih-Ho Chao, Atam P. Dhawan. “Edge detection using a Hopfield neural network.” Optical Engineering 33(11). 1994.

[Gonzalez] Gonzalez R, Woods R. “Digital Image Processing”, Addison-Wesley, Reading, Mass. 1992.

[Harris] Harris CP, Buxton BF. “GP-COM: A Distributed, Component-Based Genetic Programming System in C++” submitted to GP-96.

[Johnson] Johnson M P, Maes P, Darrell T. “Evolving Visual Routines” in *Artificial Life IV: Proceedings of the fourth International Workshop on the Synthesis and Simulation of Living Systems*, Rodney Brooks and Pattie Maes, ed. MIT Press. 1994.

[Köenderink] Köenderink JJ, van Doorn AJ. “Generic Neighbourhood Operators.” IEEE PAMI 14(6). 1992.

[Koza] Koza JR. “Genetic Programming: On the Programming of Computers by Natural Selection”. MIT Press. 1992.

[Petrou] Petrou M, Kittler J. “On the optimal edge detector.”

[Srinivasan] Srinivasan V, Byatia P, Ong SH. Edge detection using a neural network. Pattern Recognition 27(12). 1994. pp 1653-1662.

[Spacek] Spacek L A. “Edge detection and motion detection.” Image Vision Computing 4. 1986. pp 43-55.

[Tackett] Tackett WA. “Genetic Programming for Feature Discovery and Image Discrimination” in *Proc 5th International Conference on Genetic Algorithms,* ICGA ‘93. Morgan Kauffman. 1993.

[Teller] Astro Teller and Manuela Veloso. “PADO: A New Learning Architecture for Object Recognition.” in *Symbolic Visual Learning*, Katsushi Ikeuchi and Manuela Veloso, ed. Oxford University Press. 1995

[Ullman] Ullman S. “Visual Routines”. Cognition(18). 1984. pp 97-156.