Evolving Edge Detectors with Genetic Programming

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.

1. 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 (GP) 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.

2. 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 1986] 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 1986] took Canny's work and extended the theoretical performance measure to incorporate the third criterion of noise suppression. An expression for an optimal operator was derived as an equation depending on six parameters. Optimising four of these, Spacek found an operator that performed better than Canny's under slightly different boundary conditions. A cubic spline function was found to be 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 the new optimum on some real world images. Although theoretical performance was maximal for the optimal operator, in practice this 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 section 4.3.

3. Related Work

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

[Chao 1994] 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. Results were found to be comparable to a Sobel [Gonzalez 1992] operator on grey-level noisy images.

[Srinivasan 1994] used a 2-stage network to detect edges. The first stage encoded information from a receptive field (an area of image around the target pixel). The second stage took this encoding and produced edge vector components as outputs. Performance matched that of Canny's operator on a single training image.

[Bhandarkar 1994] used genetic algorithm optimisation techniques to detect edges in a grey-level image. Using 2-D chromosomes to represent 'edge-images', optimisation was performed on a whole image using a complicated cost function based on region dissimilarity, and edge curvature, thickness, and fragmentation.

4. 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.

4.1 Anatomy of an Edge Detector

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] These samples are normalised to produce an operator. The operator is convolved with a training signal to produce a response signal. The candidate trees are produced from the primitives in tables 1 and 2.

Table 1. 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.

Table 2. 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 powers of negative numbers, but any power of positive numbers.
`sin`	Sine function.
`cos`	Cosine function.

4.2 Training data

Forty 1-dimensional signals of varying lengths were used for training data. Six of these were synthetic and contained idealised 'step' edges of infinite gradient, as shown in Figure 2.

In applications, edges can occur at a variety of scales and vary in width. To incorporate this property into the training set, each of the 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 (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.

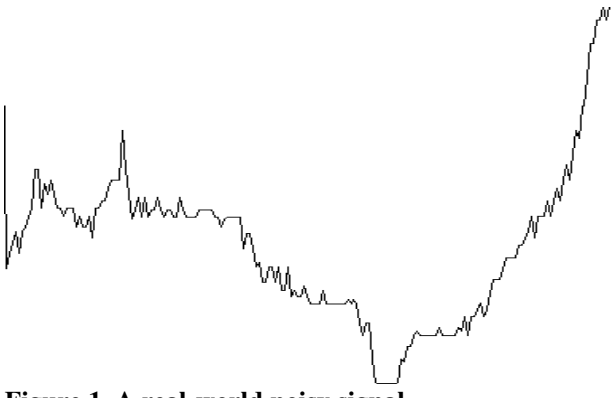


Figure 1. A real-world noisy signal

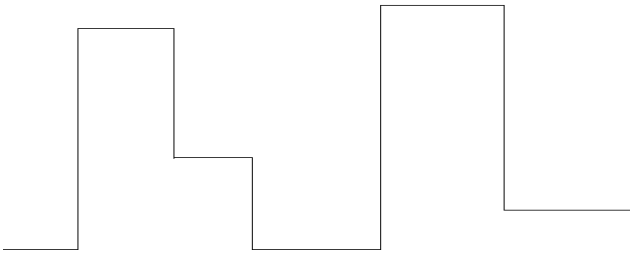


Figure 2. A signal with step edges

Each signal contained several edges, the locations of which were marked by hand and recorded. A total of 102 edges were marked over all signals. To reduce computational load, each generation was tested on a random sample of 10 of the 40 signals, which changed after each generation.

4.3 The Fitness Function

Candidate functions were assessed on three criteria for each signal, corresponding to the criteria in [Canny 1986].

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 some maximum in the response exists. 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.

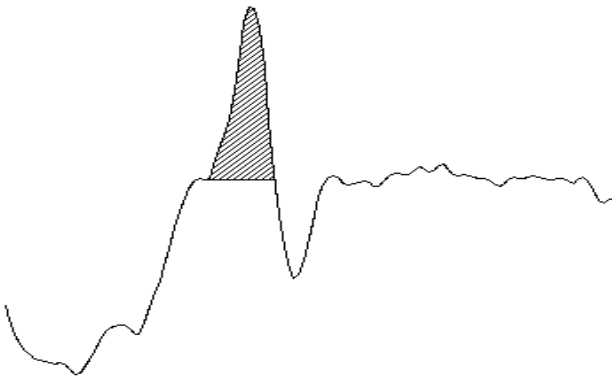


Figure 3. Area captured by a peak in signal response

Localisation. This is a measure of the accuracy in positioning of the response. A good detector will produce a peak in response, close to the position of the edge in the original signal. Performance is measured by the average displacement of the peak from the 'true' edge position, over all edges in the training set. The 'true' position is 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 peaks in the response of the operator to a signal consisting entirely of normally distributed noise. A larger peak separation means that the operator has better noise-suppression qualities.

Each candidate was sampled at resolutions of 9, 13, and 17 pixels to give 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).

4.4 Tableau of edge detection problem

Table 3. Tableau of 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 "Mutation Terminal Probability" control the average size of program trees in the population. For a fuller explanation, see [Harris, 1996].

5. Results

Six runs were made using the GP-COM system [Harris, 1996]. Each run produced an operator that performed well when compared to Canny's Gaussian first-derivative

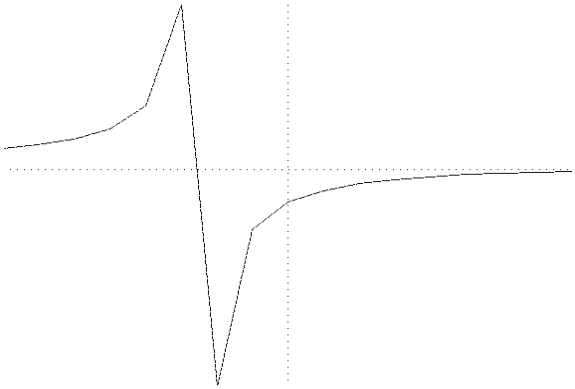
approximation. Each run was terminated when an operator was found, that exceeded the performance of Canny's operator over the sample of 10 signals.. The operators' performances are detailed in table 4, along with scores for Canny's approximation. These are raw fitness scores, high values indicating better performance, and were obtained by testing the operators over the entire set of 40 signals. This allows a fair comparison of the operators.

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.

Table 4. Results of experimental runs

Operator (f)	S(f)	L(f)	C(f)	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

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



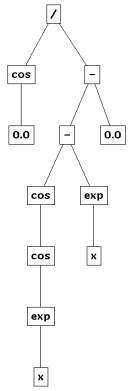
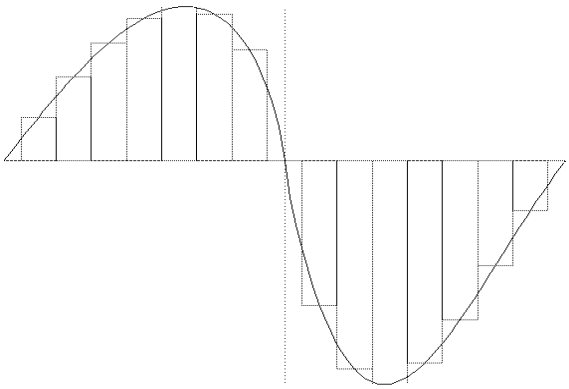


Figure 4. Operator 1.1

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



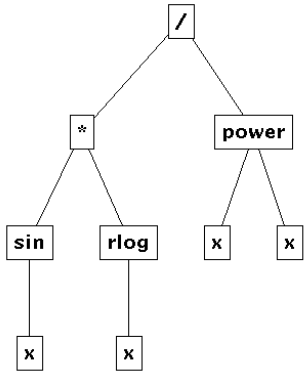
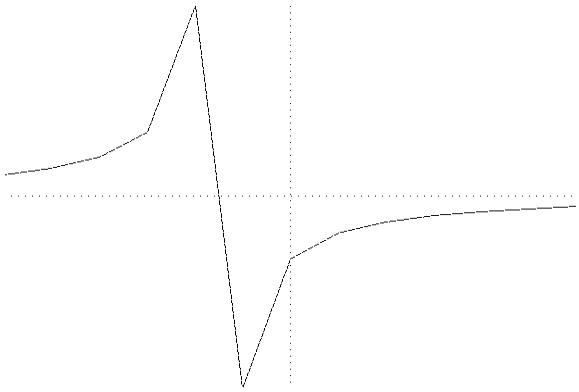


Figure 5. Operator 1.2

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



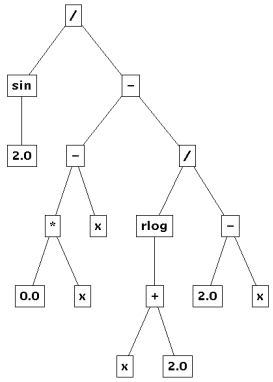
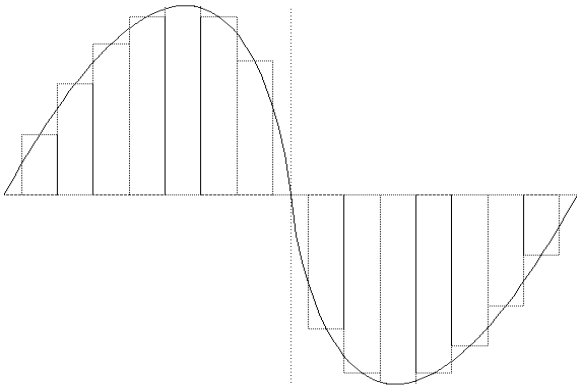


Figure 6. Operator 1.3

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



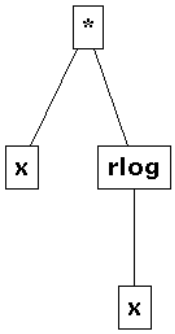
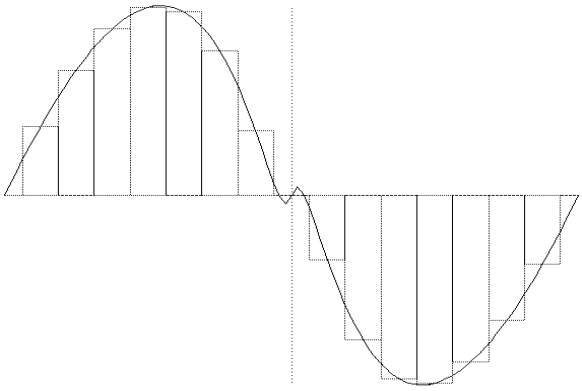


Figure 7. Operator 1.4

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



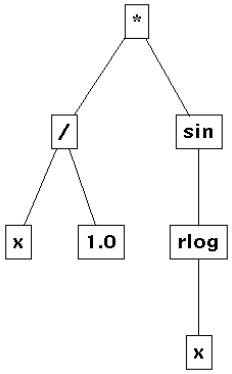
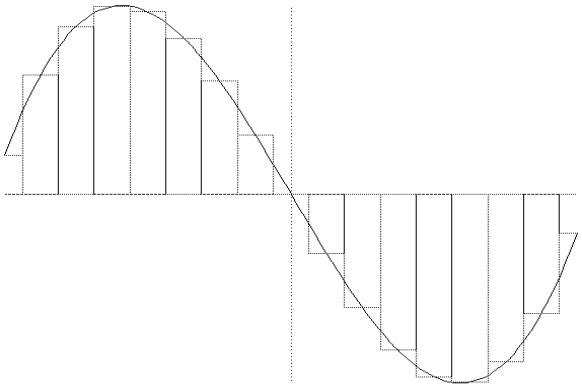


Figure 8. Operator 1.5

$x - 2x\cos(x)$



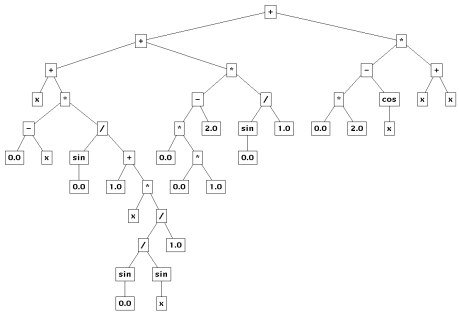
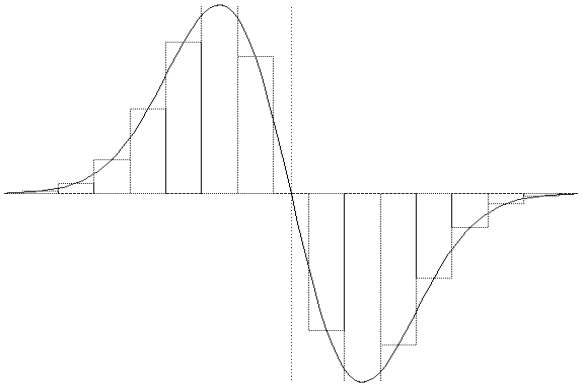


Figure 9. Operator 1.6

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



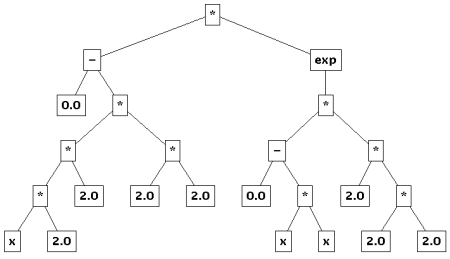


Figure 10. Approximation to Canny optimal operator with first derivative of Gaussian. Here, σ was set to 0.25.

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. 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 used in the experiments reduces this effect considerably. At these low sampling rates (as in Figures 4 and 6), they reasonably approximate the shape of a Canny-like edge detector. A high fitness should therefore be taken in context - genetic search found 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 and performed well with respect to noise suppression. 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 for noise suppression. 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 differences in response strength correspond quite well to the shape of the operators at their extrema. 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, reducing the influence of the peripheries. A larger response to sharp edges is the result.

The Canny operator performs reasonably well on response strength but falls down when it comes to noise suppression. Increasing 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 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.

7. 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, 1986]. 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.

Acknowledgements

This work was carried out under the Postgraduate Training Partnership, between SIRA Ltd and University College London.

Bibliography

Andre, D. 1994. 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.

Bhandarkar, S.M., Yiqing Zhang, Potter, W.D. 1994. An edge detection technique using Genetic Algorithm based optimisation. *Pattern Recognition.* 27(9). pp 1159-1180.

Canny, J. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 8(6).

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

Gonzalez, R., Woods, R. 1992. *Digital Image Processing*. Addison-Wesley, Reading, Mass.

Harris, C.P., Buxton, B.F. 1996. *GP-COM: A Distributed, Component-Based Genetic Programming System in C++*. Department of Computer Science Research Note RN/96/2, University College London.

Johnson, M.P., Maes, P., Darrell, T. 1994. 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.

Koza, J.R. 1992. *Genetic Programming: On the Programming of Computers by Natural Selection*. MIT Press.

Petrou, M., Kittler, J. *On the optimal edge detector.*

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

Spacek, L.A. 1986. Edge detection and motion detection. *Image Vision Computing* 4. pp 43-55.

Tackett, W.A. 1993. Genetic Programming for Feature Discovery and Image Discrimination. In *Proc 5th International Conference on Genetic Algorithms,* ICGA '93. Morgan Kauffman.

Teller, E., Veloso, M. 1995. PADO: A New Learning Architecture for Object Recognition. In *Symbolic Visual Learning*, Katsushi Ikeuchi and Manuela Veloso, ed. Oxford University Press.