On the Evolution of Interest Operators using Genetic Programming

Marc Ebner
Eberhard-Karls-Universität Tübingen
Wilhelm-Schickard-Institut für Informatik
Arbeitsbereich Rechnerarchitektur
Köstlinstraße 6, 72074 Tübingen, Germany
ebner@informatik.uni-tuebingen.de

Abstract

Interest operators play an important role in computer vision. Depending on the type of the environment some features may prove to be more advantageous than others. Thus detection of interesting features has to be made adaptive such that the best features according to some measure are extracted. We are trying to evolve such feature detectors using genetic programming. In this paper we describe our results where the desired operator, which is a Moravec interest operator, is directly specified. We show that the problem is a rather difficult one. Only an approximation to the Moravec operator could be evolved using several sets of elementary functions.

1 Motivation

Interest operators play an important role in computer vision [8]. They highlight points which can be found easily using simple correlation methods. They can be used to calculate accurate distance information and for map building [23]. However no interest operator is suitable for all types of environments. A mobile robot which may be operating in different types of environments should be able to adapt its vision system such that the robot can extract relevant information from its surroundings that can be used best according to some measure.

We are currently trying to equip a mobile robot, a RWI B21 with this type of capability. In this paper we are trying to find a simple interest operator, the Moravec interest operator [23, 22], using genetic programming [14, 15]. The Moravec operator detects points where the minimum of the sum of squared differences between adjacent pixels in four directions, horizontal, vertical and both diagonals is a local maximum. The following section gives a short summary of related work in the area of adaptive feature detection.

2 Background

Several researchers used neural nets for adaptive feature detection. Barrow [2] found weights of a neural net to converge into edge masks after the model is being trained on natural images using a Hebbian type learning rule. Linsker [18] showed that a layered self-adaptive neural network developed averaging cells, center-surround cells and orientation selective cells in successive layers using random noise as input and a Hebbian-type learning rule. Joshi and Lee [10, 11] modeled retinal responses using a neural net and showed that the weights learned with the backpropagation learning algorithm approximate the Laplacian of the Gaussian function. Thus backpropagation learns Marr's operator [11]. Lampinen and Oja [17] developed a neural network-based feature extraction and classification system for distortion tolerant pattern recognition. Kohonen [13] developed adaptive feature detectors using an adaptive-subspace self organizing map architecture.

Other researchers used evolutionary algorithms to extract image features. Lohmann [19, 20] evolved an image filter which determined the Euler number of an image using an evolution strategy [25]. Rizki et al. [26] evolved feature detectors which operate on a stack of images to which morphological operations with structuring elements at different resolutions were applied. Roth and Levine [27] extracted geometric primitives using genetic algorithms [7, 6]. Katz and Thrift [12] generated image filters for target recognition using a genetic algorithm. Bhattacharjya and Roysam [3] used evolutionary optimization for model based object recognition at low signal to noise ratios.

Tackett [29, 28] has applied genetic programming to the task of feature classification. He experimented with moment- and intensity features which are extracted from an already segmented region as well as primitive features such as the mean intensity or standard deviation. Tackett used these features in the terminal set of the algorithm, they are not subjected to an adaptive process. Koza [16] evolved detectors for letter recognition which were able to discriminate the letters "I" and "L". The detectors moved themselves over the binary pattern and could analyze the pixels in a local 3×3 neighborhood. Andre [1] used genetic programming to evolve 2-dimensional feature detectors using 3×3 hit-miss-matrixes. The task was to discriminate between one designated digit and the rest of the digits. The individuals moved themselves over the image and were able to compare their surroundings with the hit-miss-matrixes. Johnson et al. [9] used genetic programming to evolve Ullman's Visual Routines [30] for the task of determining the location of hands in the bitmap silhouette of a person. Although Johnson et al. are working on real camera data, they are using preprocessed data for the evolution, namely the bitmap silhouette which are binarized images obtained by using a blue screen to segment the person from the background.

We previously used structure evolution, developed by Lohmann [20, 21] a variant of an evolution strategy [25] to evolve hierarchical feature detectors which we applied to the task of character recognition [4]. Using one simple structure changing operator we showed that an increasingly complex detector evolved from simple filter operations. In [5] we evolved edge detectors using genetic programming by approximating the Canny edge detector. In this paper we are focusing on the task of evolving an interest operator. In contrast to the work of Johnson et al. [9] we are working with raw image data that is not preprocessed except for scaling of the pixel intensities.

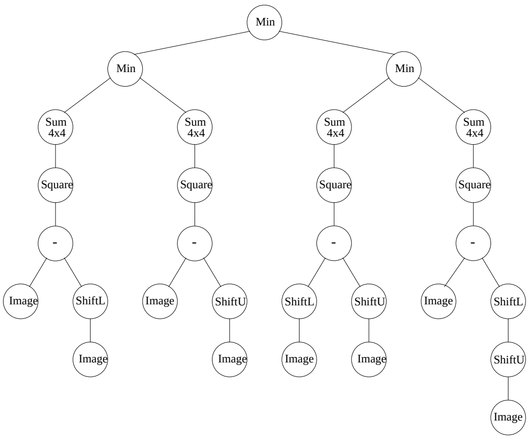

Figure 1: Structure of Moravec operator.

3 Evolution of interest operators using genetic programming

To evolve interest operators which are being optimized according to some measure, we are using genetic programming. Thus we need to specify the set of terminals, the set of elementary functions, the fitness measure, the parameters for the run and a criterion to terminate the run [14].

3.1 Set of terminals

We selected a gray scale representation of the input image as our sole terminal. The image intensities are scaled to the range [0,1]. Thus the terminal set T becomes $T = \{\text{Image}\}$. Other terminal sets can also be envisaged. For instance one could use the three color bands red, green and blue or hue, saturation and intensity or some combination of them and let evolution select the terminals which are suited best for the task at hand.

3.2 Set of primitive functions

The set of primitive functions has to be powerful enough such that the problem at hand may actually be solved. The Moravec interest operator is usually written as

$$
\begin{aligned}
I_R(x,y) = \min\{ \\
&\sum_{x-2\le x'<x+2}\ \sum_{y-2\le y'<y+2} \left(I(x',y') - I(x'+1,y')\right)^2, \\
&\sum_{x-2\le x'<x+2}\ \sum_{y-2\le y'<y+2} \left(I(x',y') - I(x',y'+1)\right)^2, \\
&\sum_{x-2\le x'<x+2}\ \sum_{y-2\le y'<y+2} \left(I(x'+1,y') - I(x',y'+1)\right)^2, \\
&\sum_{x-2\le x'<x+2}\ \sum_{y-2\le y'<y+2} \left(I(x',y') - I(x'+1,y'+1)\right)^2\}
\end{aligned}
$$

This expression operating on pixel values can be rewritten into an expression consisting entirely of elementary functions operating on whole images. The structure of the Moravec operator using the such elementary functions is shown in figure 1. The resulting image is filtered by suppressing non-local maxima and applying a thresholding operation to extract interesting points from the images.

In the following text the images used as operands are denoted by I or I_i where $i \in \{1,\ldots,4\}$ and the resulting image is denoted by I_R. The following unary functions were used:

Negation (Neg): $I_R(x,y) = -I(x,y)$. Absolute value (Abs): $I_R(x,y) = |I(x,y)|$. Square values (Square): $I_R(x,y) = I(x,y) \cdot I(x,y)$. Shift left (ShiftL): $I_R(x,y) = I(x+1,y)$. Shift right (ShiftR): $I_R(x,y) = I(x-1,y)$. Shift up (ShiftU): $I_R(x,y) = I(x,y+1)$. Shift down (ShiftD): $I_R(x,y) = I(x,y-1)$. Average in 4×4 area (Avg4x4): $I_R(x,y) = \frac{1}{16}\sum_{-2\le i,j<2} I(x+i,y+j)$. Sum in 4×4 area (Sum4x4): $I_R(x,y) = \sum_{-2\le i,j<2} I(x+i,y+j)$.

The following binary functions were used: Subtraction (-): $I_R(x,y) = I_1(x,y) - I_2(x,y)$. Division (/): $I_R(x,y) = I_1(x,y)/I_2(x,y)$. Multiplication (*): $I_R(x,y) = I_1(x,y) \cdot I_2(x,y)$. Addition (+): $I_R(x,y) = I_1(x,y) + I_2(x,y)$. Minimum (Min): $I_R(x,y) = \min\{I_1(x,y), I_2(x,y)\}$. Maximum (Max): $I_R(x,y) = \max\{I_1(x,y), I_2(x,y)\}$.

In addition we used the following N-ary functions ($N \in \{3,4\}$). Multiplication (PiN): $I_R(x,y) = \Pi_{i=1}^{i=N} I_i(x,y)$. Addition (AddN): $I_R(x,y) = \sum_{i=1}^{i=N} I_i(x,y)$. Minimum (MinN): $I_R(x,y) = \min\{I_i(x,y) | i \in \{1,\ldots,N\}\}$. Maximum (MaxN): $I_R(x,y) = \max\{I_i(x,y) | i \in \{1,\ldots,N\}\}$.

1	BASE
2	BASE ∪ { Avg4x4 }
3	BASE ∪ { Sum4x4 }
4	BASE ∪ { Sum4x4, Pi3, Add3, Max3, Min3 }
5	BASE ∪ { Sum4x4, Pi3, Add3, Max3, Min3, Pi4, Add4, Max4, Min4 }

Table 1: Different sets of elementary functions used for the experiments.

3.3 Fitness measure

As raw fitness measure to be minimized we selected the squared pixel differences between the actual and the desired output of the operator. For our problem raw fitness equals standardized fitness.

$$
\text{fitness}_{\text{raw}}(\text{Ind}) = \sum_{i=1}^{5}\Big(U(\text{Ind}(\text{I}_i)) + \frac{1}{n}\sum_{p\in \text{I}_i}\left((\text{Ind}(\text{I}_i))(p) - (\text{Moravec}(\text{I}_i))(p)\right)^2\Big)
$$

where the five images for the different fitness cases are given as $\{\text{I}_1,\ldots,\text{I}_5\}$, p is a point from the image and n is the number of points in the image. The evolved operator is denoted by Ind and the desired operator is denoted by Moravec. The term $U(\text{Ind}(I_i))$ evaluates to a large value for a uniform image and to zero otherwise.

4 Experiments

We performed five experiments with a population size of 4000 individuals to evolve feature detectors which approximate the response of the Moravec operator. Crossover probability has been set to 85%, reproduction rate has been set to 10% and the mutation rate has been set to 5%. We used ramped half and half initialization and fitness proportionate selection with over-selection. Five fitness cases are evaluated. The five pictures used during the evolution are shown in figure 4. Each run was aborted after 50 generations. For each experiment we performed three different runs. For the experiments we used different sets of elementary functions. The following base set of elementary functions was used for all experiments:

$$\text{BASE} = \{\text{Neg, Abs, Square, ShiftL, ShiftR, ShiftU, ShiftD, -, /, *, +, Max, Min}\}$$

The different sets of elementary functions used for the experiments are shown in table 1.

The base set of elementary functions is sufficient to evolve a Moravec interest operator. There seems to be one elementary function missing, namely Sum4x4. However this elementary function can be constructed using the shift operations and the add operation. The absence of the Sum4x4 elementary function complicates the search for a correct interest operator considerably, because the Sum4x4 elementary function is used 4 times in the correct individual. In each of these places a subroutine performing the desired summation would have to be evolved. This would be a task where automatically defined functions [15] might simplify the problem. We wanted to see what solutions are found if no automatically defined functions are used.

Next we augmented the base set with the operator Avg4x4. Now genetic programming has the possibility of evolving the constant 16 to produce the required elementary function Sum4x4 (e.g. $\text{Sum4x4} = 16 \cdot \text{Avg4x4}$ where $16 = \text{Square}(\text{Square}(\frac{I}{I} + \frac{I}{I}))$. Then we added the required function to the base set. Finally we added the min, max, +, * operators with arity 3 and arities 3 and 4 to the base set.

The best individual of all runs was found in generation 50 using set 5: (Min3 (Sum4x4 (Square (- (Min Image Image) (ShiftU Image)))) (Sum4x4 (Square (- (Neg (- (Min (Abs (Min Image Image)) Image) (ShiftL Image))) (/ (ShiftD (ShiftR (- (Max3 Image Image Image) Image))) (Square (Max3 Image Image Image)))))) (Sum4x4 (Square (- (Neg (- (Min (Abs (Min Image Image)) Image) (ShiftL Image))) (/ (ShiftD (ShiftR (- (Max4 Image Image Image Image) (ShiftR Image)))) (Sum4x4 (ShiftL Image))))))).

Set 3 had the highest average adjusted fitness at generation 50. The structure of the best individuals found with the sets 5 and 3 are shown in figure 2. To make sure that random search by itself did not already produce an interest operator we examined generation 0 of the runs which produced these individuals. The structure found by random search in generation 0 both only emphasize vertical edges. The best individual found with set 5 approximates the Moravec operator quite closely. However, it is not a 100% correct individual. The fitness is especially good for 4 pictures where the sum of the squared pixel differences is less than 0.0005. The result that an interest operator with a seemingly simple structure is rather difficult to evolve is quite surprising. Figure 3 shows the average adjusted fitness of best-of-generation individual for all of the base sets. The features detected by the best evolved interest operator can be seen in figure 4. The first two rows in figure 4 show the result of the Moravec operator. The next two rows show the response of the Moravec operator before a suppression of non-local maxima has been applied. Next, the best individual of generation 0 (found by random search) in the run that produced the closest approximation to the Moravec interest operator is shown. The following two rows show the best approximation to the Moravec interest operator found during the experiments. The final two rows show the features extracted by the best evolved individual superimposed on the original images after the non-local maxima suppression and thresholding operator has been applied.

Note that although the response of the evolved operator approximates the response of the Moravec operator very closely,

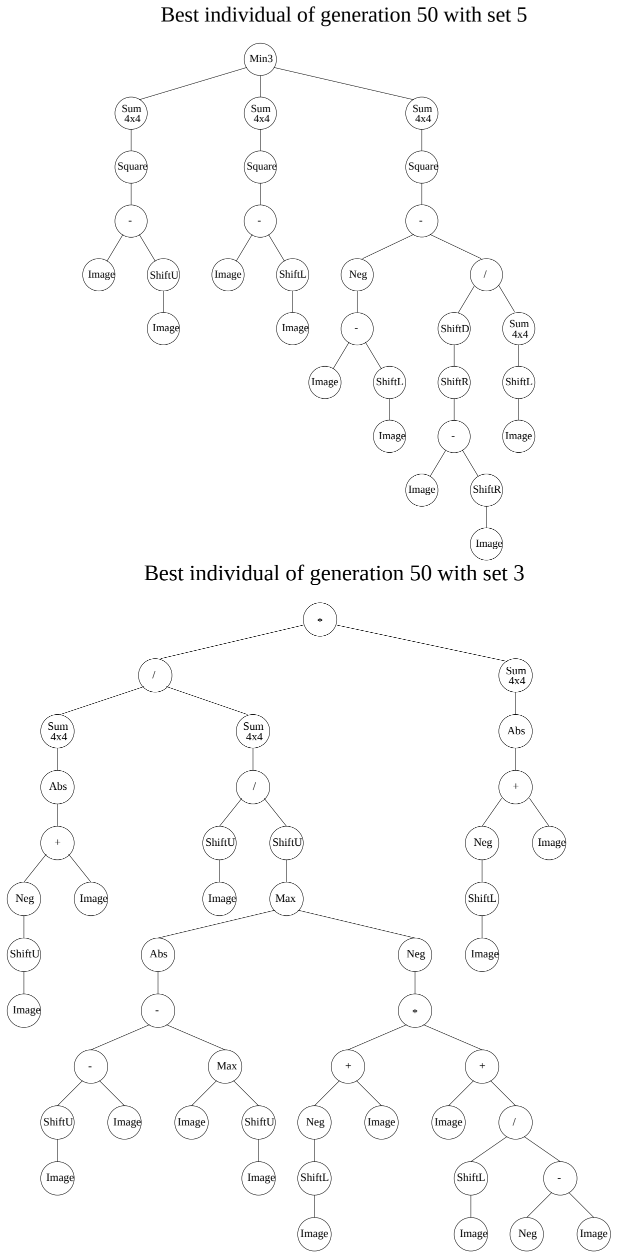

Figure 2: Two best individuals which approximate the Moravec interest operator (the one on top was found using set 5, the one below was found using set 3). Both individuals have been manually simplified.

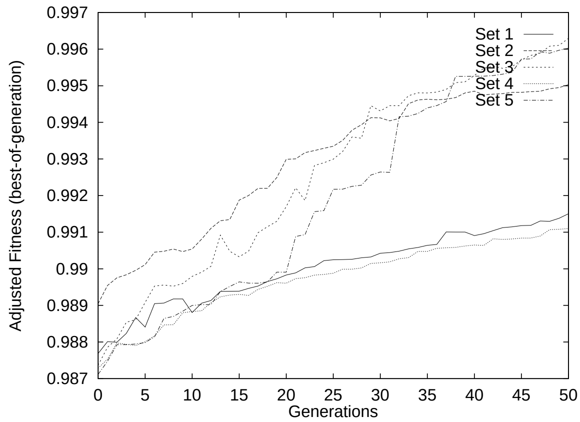

Figure 3: Fitness statistics for all experiments. Each curve has been computed as the average of three runs. Following the standard definition, adjusted fitness is to be maximized.

the actually detected features may still differ. This is due to the fact that a non-local maxima suppression and thresholding operation has been applied that was not included in the fitness function. The task was to approximate the operator response and not to extract the same features. The best evolved interest operator has also been applied to a set of five previously unseen images. The results are shown in Figure 5. The features in the top two rows were extracted with a Moravec interest operator. The next two rows show the response of the Moravec interest operator. The following two rows show the response of the best evolved individual. The final two rows show the features detected by the evolved detector after a non-local maxima suppression and a thresholding operation has been applied.

5 Conclusion and ongoing research

We have shown that genetic programming evolved feature detectors which approximate the Moravec interest operator. However a 100% correct individual has not been found using a population size of 4000 and terminating the evolution after 50 generations. This could be due to the particular structure of the operator at the top of the tree which could be difficult to find.

We are currently experimenting with fitness functions that are not based on any existing operator. Such a fitness measure only describes the desired characteristics of the interest operator. In addition we are experimenting with high level operators such as edge detection, Gaussian smoothing and Gabor filters which augment the set of elementary functions.

6 Acknowledgements

The author is currently supported by a scholarship according to the Landesgraduiertenförderungsgesetz.

For our experiments we used the lil-gp Programming System, version 1.01 [31]. For image processing we used the Vista software environment [24].

References

[1] David Andre. Automatically defined features: The simultaneous evolution of 2-dimensional feature detectors and an algorithm for using them. In Jr. Kenneth E. Kinnear, editor, *Advances in Genetic Programming*, pages 477–494, Cambridge, Massachusetts, 1994. The MIT Press.

[2] Harry. G. Barrow. Learning receptive fields. In *Proceedings of the 1st International Conference on Neural Networks*, volume 4, pages 115–121. IEEE, 1987.

[3] Anoop K. Bhattacharjya and Badrinath Roysam. Joint Solution of Low, Intermediate, and High-Level Vision Tasks by Evolutionary Optimization: Application to Computer Vision at Low SNR. *IEEE Transactions on Neural Networks*, 5(1):83–95, January 1994.

[4] Marc Ebner. Evolution of hierarchical translation-invariant feature detectors with an application to character recognition. In Erwin Paulus and Friedrich M. Wahl, editors, *Mustererkennung 1997, 19. DAGM-Symposium Braunschweig, 15.-17. September 1997*, pages 456–463, Berlin, 1997. Springer-Verlag.

[5] Marc Ebner. On the evolution of edge detectors for robot vision using genetic programming. In Horst-Michael Groß, editor, *Workshop SOAVE '97 - Selbstorganisation von Adaptivem Verhalten 1997*, pages 127–134, Düsseldorf, 1997. VDI Verlag.

[6] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

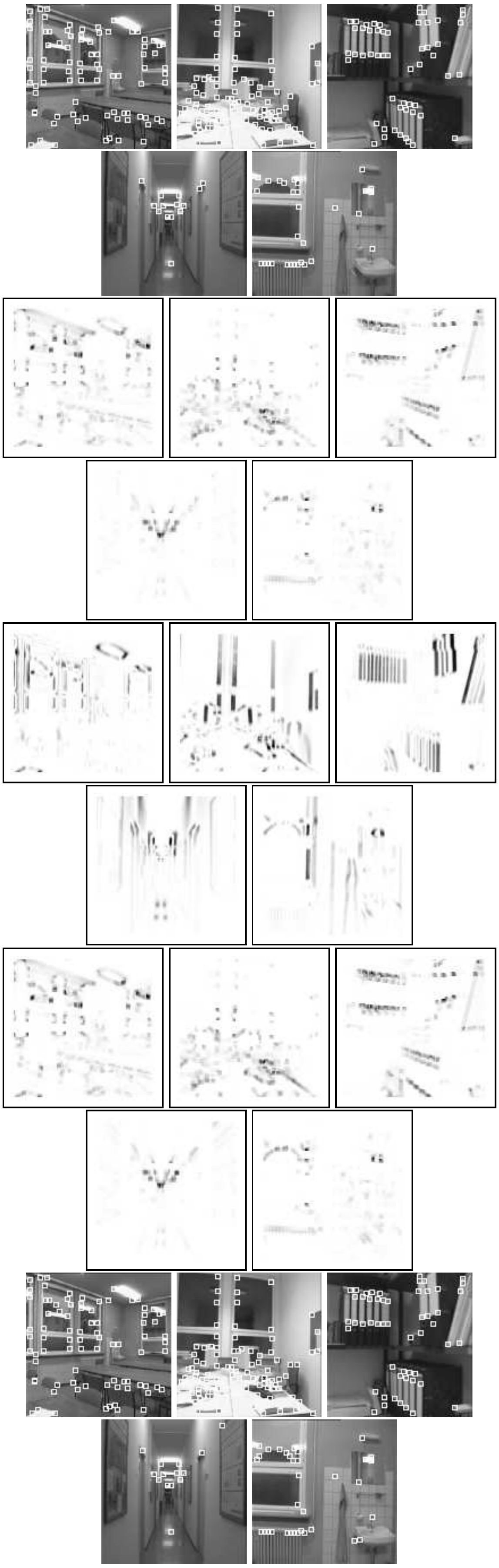

Figure 4: The top two rows show five different images were interesting features have been located which a Moravec operator. The final two rows show the features of the best evolved individual superimposed on the original images after the non-local maxima suppression and thresholding operator has been applied. See text for an explanation of the other images.

[7] John H. Holland. *Adaptation in natural and artifical systems: an inroductory analysis with applications to biology, control, and artificial intelligence*. The MIT Press, Cambridge, Massachusetts, 1992.

[8] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck. *Machine Vision*. McGraw-Hill, Inc., New York, 1995.

[9] Michael Patrick Johnson, Pattie Maes, and Trevor Darrell. Evolving visual routines. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 198–209, Cambridge, Massachusetts, 1994. The MIT Press.

[10] Anupam Joshi and Chia-Hoang Lee. On modelling the retina using neural networks. *Proc. Int. Joint Conf. on Neural Networks 91, Singapore*, pages 2343–2348, 1991.

[11] Anupam Joshi and Chia-Hoang Lee. Backpropagation learns Marr's operator. *Biological Cybernetics*, 70:65–73, 1993.

[12] A. J. Katz and P. R. Thrift. Generating image filters for target recognition by genetic learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):906–910, Sept. 1994.

[13] Teuvo Kohonen. Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. *Biological Cybernetics*, 75:281–291, 1996.

[14] John R. Koza. *Genetic Programming, On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, Massachusetts, 1992.

[15] John R. Koza. *Genetic Programming II, Automatic Discovery of Reusable Programs*. The MIT Press, Cambridge, Massachusetts, 1994.

[16] John R. Koza. *Automatic Discovery of Detectors for Letter Recognition*. In *Genetic Programming II, Automatic Discovery of Reusable Programs*, pages 389–416. The MIT Press, Cambridge, Massachusetts, 1994.

[17] Jouko Lampinen and Erkki Oja. Distortion tolerant pattern recognition based on self-organizing feature extraction. *IEEE Transactions on Neural Networks*, 6(3):539–547, May 1995.

[18] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21:105–117, 1988.

[19] Reinhard Lohmann. Selforganization by evolution strategy in visual systems. In Hans-Michael Voigt, Heinz Mühlenbein, and Hans-Paul-Schwefel, editors, *Evolution and Optimization '89*, pages 61–68. Akademie-Verlag, 1990.

[20] Reinhard Lohmann. *Bionische Verfahren zur Entwicklung visueller Systeme*. PhD thesis, Technische Universität Berlin, Fachbereich 10 Verfahrenstechnik und Energietechnik, 1991.

[21] R. Lohmann. Structure evolution and incomplete induction. *Biological Cybernetics*, 69:319–326, 1993.

[22] Hans P. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Vision–1: p. 584, 1977.

[23] Hans P. Moravec. *Ostacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Computer Science Department, Stanford University, No. STAN-CS-80-813 and AIM-340, September 1980.

[24] Arthur R. Pope and David G. Lowe. Vista: A software environment for computer vision research. In *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 768–772. IEEE, 1994.

[25] Ingo Rechenberg. *Evolutionsstrategie '94*. frommann-holzboog, Stuttgart, 1994.

[26] Mateen M. Rizki, Louis A. Tamburino, and Michael A. Zmuda. Evolving multi-resolution feature-detectors. In David B. Fogel and W. Atmar, editors, *Proceedings of the Second American Conference on Evolutionary Programming*, pages 108–118. Evolutionary Programming Society, 1993.

[27] Gerhard Roth and Martin D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):901–905, Sept. 1994.

[28] Walter Alden Tackett. Genetic generation of "dendritic" trees for image classification. In *Proceedings of the World Conference on Neural Networks, Portland, Oregon, July, 1993*, pages IV–646–IV–649. IEEE, 1993.

[29] Walter Alden Tackett. Genetic programming for feature discovery and image discrimination. In S. Forrest, editor, *Proceedings of the Fifth International Conerence on Genetic Algorithms*, pages 303–309. Morgan Kaufmann, 1993.

[30] Shimon Ullman. Visual routines. In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 298–328, Los Altos, California, 1987. Morgan Kaufmann Publishers.

[31] Douglas Zongker and Bill Punch. *lil-gp 1.01 User's Manual (support and enhancements Bill Rand)*. Michigan State University, March 1996.

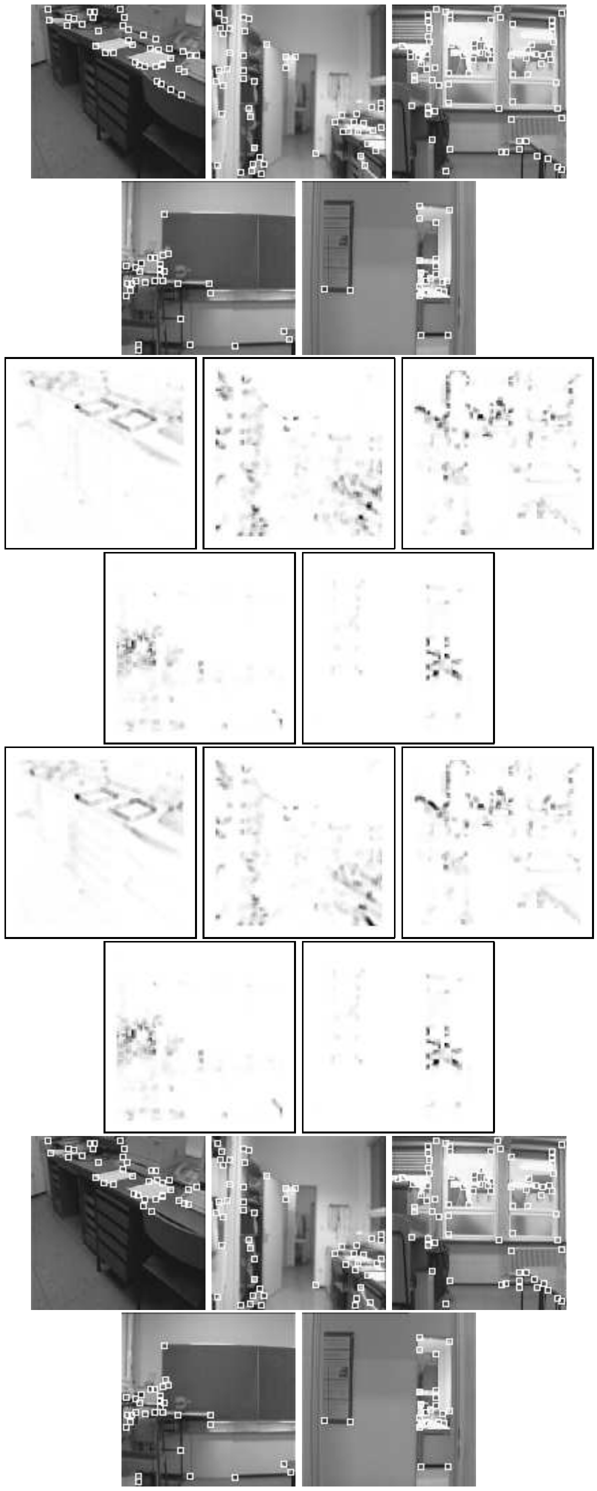

Figure 5: Five images were used to test the evolved interest operators. See text for further explanation.