IMAGE FILTER DESIGN BASED ON EVOLUTION

Jan Karásek, Radek Beneš
Doctoral Degree Programme (1), FEEC BUT
E-mail: karasek.jan@phd.feec.vutbr.cz, xbenes30@stud.feec.vutbr.cz,

Supervised by: Radim Burget
E-mail: burgetrm@feec.vutbr.cz

ABSTRACT

Image filter design is a time-consuming process and can easily get stuck in a suboptimal solution when it is designed by the human programmer. This paper proposes an evolution approach for image filter design based on grammar-guided genetic programming (GGGP) utilizing the Evolution Framework (Fig. 1) which is being developed at the Faculty of Electrical Engineering and Communication at the Brno University of Technology. The aim of this work is not to substitute the human work, but to facilitate and improve the productivity of image filter design in many scientific and engineering applications.

1. INTRODUCTION

Image processing is important in numerous applications such as multimedia, computer graphics, medical imaging, robotics, artificial intelligence and many others. Object recognition is a complex process and has to be extremely accurate. The quality of recognition depends on the quality of image and design of the algorithm. Almost all image filters are designed by conventional ways. The most common approach for a new image filter design is a static design by a human programmer. The problem of such approach is that the process is time-consuming and can easily get stuck in a suboptimal solution. Another problem is to assess objectively a significant set of samples.

The main contribution of this paper is to propose an innovative approach to the image filter design using evolutionary techniques and to show its use. The rest of the paper is organized as follows. The second chapter entitled Evolution Framework (EF) describes the fundamental principles and types of genetic programming. It also mentions the extensions of GP, which is grammar. The third chapter deals with the results of the work and describes the proposed filter for artery recognition and, finally, the work is closed with the conclusion.

2. EVOLUTION FRAMEWORK

2.1. FUNDAMENTAL PRINCIPLES OF GP

Genetic programming is a domain-independent problem-solving approach in which computer programs are evolved to solve, or approximately solve, problems. [2] Genetic programming is an extension of genetic algorithms (GA). The main difference between them is the representation of the structure they manipulate and the meanings of the representation. GAs usually operate on a population of fixed-length binary strings. GP typically operates on a population of parse trees which usually represent computer programs. [1] [3]

In applying genetic programming to a specific problem, there are five major preparatory steps. These five steps involve determining: [4]

- the set of terminals - such as numbers, constants, samples of voice or image etc.
- the set of primitive functions (non-terminals) – arithmetic and logic functions etc.
- the fitness measure
- the parameters for controlling the run
- the method for designating a result and the criterion for terminating a run

Genetic programming starts with randomly generated individuals in initialization population. The creation of the population is a blind random search of the search space of the problem represented as computer programs. Each individual in the population is measured in terms of how well it performs in the particular problem environment. This measure is called the fitness measure. After the initialization population generation and evaluation of the chromosomes, genetic operators, such as selection, crossover, mutation and replacement are executed to breed a population of trial solutions that improves over time. [4]

The primary genetic operators which are used in Evolution Framework are crossover and mutation. In principle, the crossover operation is used to create new offspring from two parents selected based on fitness. The crossover is based on selecting a crossover point in both parents and swapping the sub-trees. The mutation operation is an asexual operator, it means that it operates only with one parent. The mutation point is randomly selected and the sub-tree is replaced by a completely newly generated sub-tree. These techniques are the basic approaches and many variants of these operators exist in practice.

2.2. VARIANTS OF GENETIC PROGRAMMING

There are many variants of genetic programming. They are different in data structure representation. The chromosomes can be represented by linear, tree and graph structures. The main three ways used in contemporary literature are Tree-Based Genetic Programming (TGP), Linear Genetic Programming (LGP) [5] and Cartesian Genetic Programming (CGP) [6]. The Evolution Framework developed in FEEC BUT uses TGP, because applying of the grammar to this type of GP is the simplest.

2.3. GRAMMAR-GUIDED GENETIC PROGRAMMING

The GGGP is an extension of traditional GP systems. The GGGP goals are to provide knowledge about the problem to be solved to simplify the search space and to solve the closure problem and to always facilitate generating of valid individuals. This extension of genetic programming concerns the crossover operation, the mutation and also the initialization method, which has to be adapted. The Evolution Framework uses the grammar-based crossover (GBX) and the grammar-based mutation (GBM) based on the research in [7] [8].

A context-free grammar G is defined as 4-tuple $G = (T, F, S, P)$, where T is the set of terminals, F is the set of non-terminal symbols (functions) and $T \cap F = 0$, S is the start symbol and P represents the set of production rules, written in BNF (Backus-Naur Form). Based on this grammar, the individuals are defined as derivation trees with the root S, the internal nodes from the set F and the external nodes (leaves) from the set T. [7], [8]

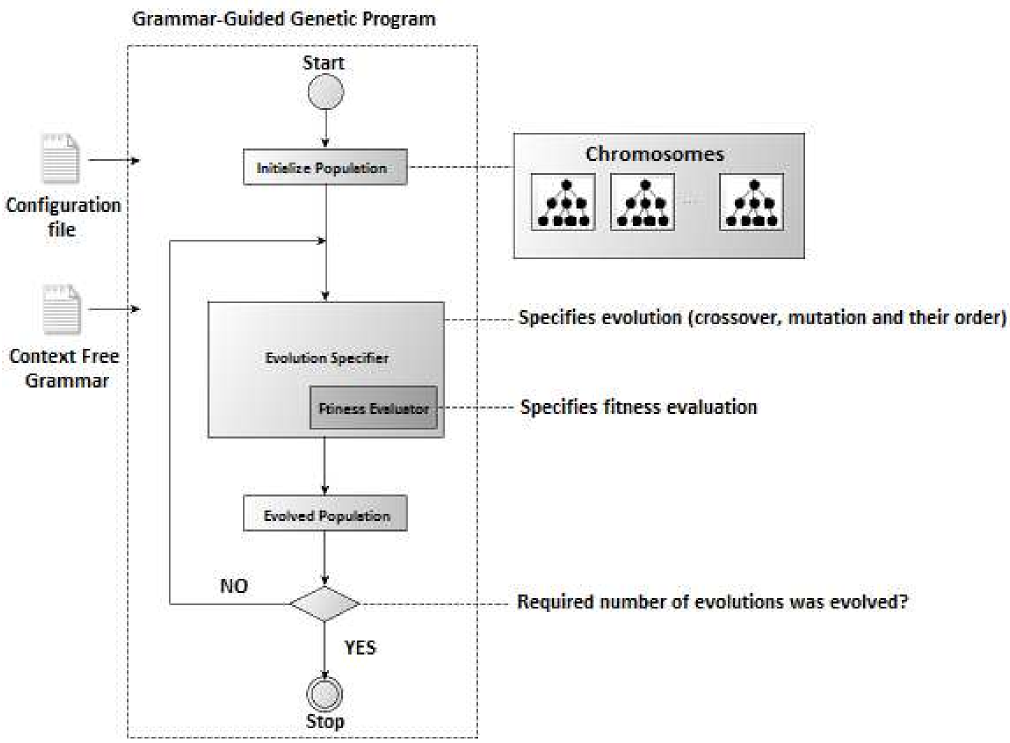

Figure 1: Scheme of Evolution Framework

3. RESULTS

The following text presents an illustrative example and its results achieved by the proposed evolution method. In this example, the filter for artery recognition from the medical screening has been designed. Robustness design of the resulting filter is dependent on the number of plug-ins. In this project, a limited set of plug-ins is used, nevertheless it is possible to trace the good functionality of the proposed program. In these examples, the following grammar *G* was used:

- *T* = (Image, Integer)
- *F* = (Hough Transform, Erode, Dilate, Open, Close, Threshold, Entropy Treshold, Histogram Equalization, Watershed, Hessian, Blur, Logarithm, Sobel, Laplace)
- *S* = Root
- *P* = { Root ::= HoughTransformation

  HoughTransformation ::= Erode Integer Integer Integer | Dilate Integer Integer Integer | Open Integer Integer Integer | Close Integer Integer Integer

  Open | Close | Erode | Dilate ::= Treshold | EntropyTreshold

  Treshold | EntropyTreshold ::= Sobel | Laplace | HistogramEqual. | Watershed

  HistogramEqual. ::= Hessian

  Sobel | Laplace | Hessian | Watershed ::= Blur | Logarithm

  Blur ::= Logarithm Integer | Image Integer, Logarithm ::= Image }

Fig. 2a shows the filter proposed by Evolution Framework and Fig. 2b-2g shows outputs of the individual filters in the generated program. The filter was designed in 40 evolutions with initialization population which contained 100 chromosomes, with crossover rate 80% and mutation rate 30%. The total time of the final filter design was 12h 11m. The detection of artery by an individual chromosome (filter) takes from 1.2s to 1.8s in individual images.

The fitness function was designed for the evaluation of individual chromosomes based on calculation of the accuracy of detection of centres in all the arteries from the image training set containing 9 frames. The resulting filter was tested on a set of test images, comprising 147 images. The success of the artery detection on images of the testing set was approximately 75%. The success at this stage of development is very good and it may be even higher in the future, when additional plug-ins will be implemented.

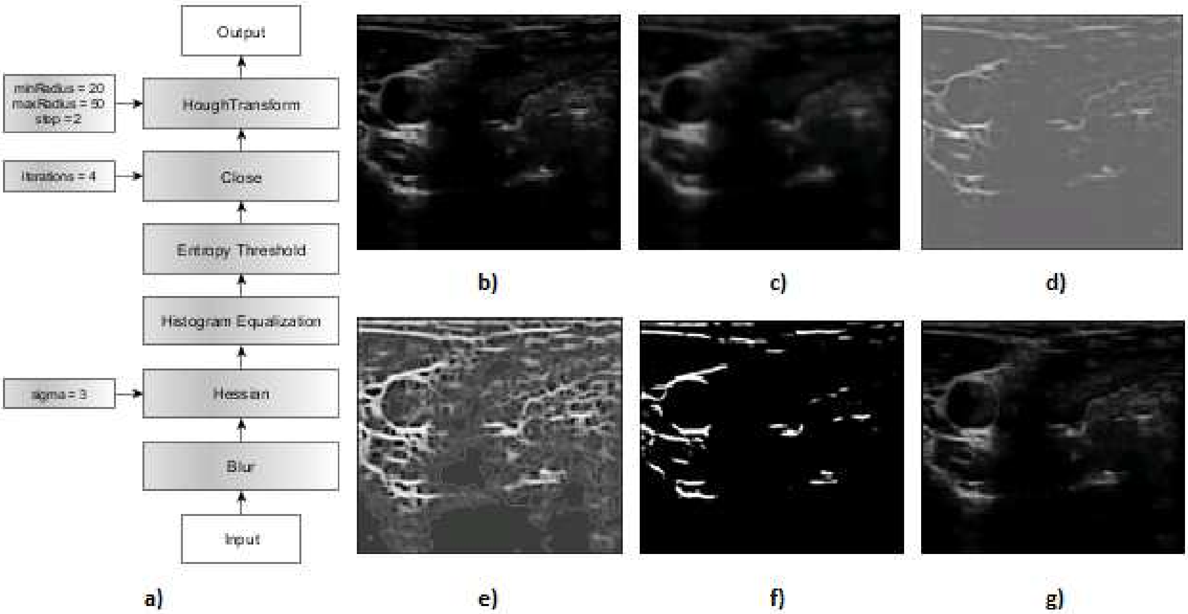

Figure 2: a) Filter proposed by Evolution Framework, b) Input image, c) Output of Gaussian smooth, d) Output of Hessian, e) Output of Histogram Equalization, f) Output of Threshold, g) Final output - recognized artery.

As we can see the image under consideration is given to the entry of Gaussian blur. This filter is one of the linear smoothing filters and its main task is to reduce noise. Blur must not be chosen too large to prevent the filter to remove some important properties of the picture. The blurred image then enters the block which analyzes the degree of curvature in the image using the Hessian. The operator indicates with a light color the areas where a certain curvature is apparent. This can be used with advantage to find circular shapes as required by assignment. The proposed filter then performs histogram equalization which enhances the image contrast. This is followed by binary thresholding. Specifically, thresholding based on entropy is used. [9] The accepted method comes out of the assumption that the input image is generated from two signals: the foreground and background signal. An ideal thresholding occurs at the moment when the sum of entropies of the two signals reaches the maximum. Based on this premise, a formula is established to determine the ideal threshold, which is then used for binary thresholding. Over the binary images, the morphological operations of erosion and dilation, constituting together the operation Close, were performed. They ensure improving of images for further processing.

The last step, strictly prescribed by grammar, is Hough transformation. Besides the input image, other parameters enter the plug-in: the minimum and maximum size of the circle and the step to analyze this range. The method will search the specified number of best-rated circuits and will return their parameters. The principle of Hough transformation lies in the transformation of the original image into a new multi-dimensional space. The maximum, which coordinates correspond to the parameters of sought element, is localized in the new space. There are modifications to find lines, circles, as well as methods for other shapes. The proposed structure of the resulting plug-in achieves the best fitness function across a training set and gains a certain optimality of found solution. The resulting filter can then be applied to many other similar images.

4. CONCLUSION

In this paper a new approach to image processing filter design was proposed, implemented and basically evaluated. It is clear that the evolution cannot substitute the human in the process of a new image filter design; however it can definitely significantly facilitate and improve the productivity of the human work.

ACKNOWLEDGEMENTS

This work was prepared with the support of the Czech Ministry of Education Youth and Sports project No. 2B06111 and ME10123.

The access to the METACentrum (super)computing facilities provided under the research intent MSM6383917201 is highly appreciated.

REFERENCES

[1] Koza, John R. *Genetic Programming : On the Programming of Computers by Means of Natural Selection*. Cambridge: MIT Press, 1992. 840 s. ISBN 0-262-11170-5.

[2] Koza, John R. *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, Massachusetts: MIT Press, 1994. 768 s. ISBN 0-262-11189-6

[3] Koza, John R., Andre, D., Bennett III, F., a Keane, M. *Genetic Programming 3: Darwinian Invention and Problem Solving*. Morgan Kaufman, 1999. 1154 s.

[4] Koza, John R. *Survey of genetic algorithms and genetic programming*. *WESCON Conference*. 1995, IEEE. 589 - 594. [cit. 2010-01-25].

[5] Brameier, M. *Linear Genetic Programming*, Springer, New York. 2006. 323 s.

[6] Miller, J. F., Thompson, P., *Cartesian Genetic Programming, Proceedings of the European Conference on Genetic Programming*. Springer, Berlin. 2000. Str. 121 – 132.

[7] García-Arnau, M., Manrique, D., Ríos, J., Rodríguez-Patón, A. *Initialization method for grammar-guided genetic programming*. ScienceDirect. 2006. 7 s.

[8] Couchet, J., Manrique, D., Ríos, J., Rodríguez-Patón, A. *Crossover and mutation operators for grammar-guided genetic programming.* Springer. 2006. 12 s.

[9] Sahoo, P. K., Soltani, S., Wang, A. K. C. and Chen, Y. C. *A. Survey of The Thresholding Techniques.* Computing Vision Graphics Image Process. 1988. Str. 233-260