Evolution of Contours for Shape Recognition

Ukrit Watchareeruetai[1] and Kriangkrai Phanjan[2]
School of Engineering and Technology, International College
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang, Bangkok 10520, Thailand
E-mail: [1]ukrit@ieee.org, [2]s409005@kmitl.ac.th

Abstract: This paper proposes an evolutionary method for constructing contour feature extraction programs for shape recognition. The proposed method adopts a variant of genetic programming (GP), called linear GP, to optimize the performance of programs. Linear GP used in this paper requires two types of registers: 1) numerical registers and 2) contour registers. Consequently, the contour of an object which is stored in a contour register can be processed to produce some features and also be transformed by some primitive operators to generate another contour. During evolutionary process, an input contour is evolved and, hopefully, its useful features are then extracted. Preliminary results show that the proposed method can automatically construct a contour feature extractor for a leaf recognition problem, with an accuracy of 90%.

1. Introduction

Feature extraction is a crucial process that strongly affects the performance of a computer vision system. Generally, human experts choose a set of appropriate features to solve a given problem and design algorithms to extract such features from an input image. However, designing a feature extraction program is a difficult task and also requires a high level of expertise and problem-dependent knowledge.

To overcome the difficulties in designing a feature extraction algorithm, there are a number of research attempts to develop a system that can automatically construct a feature extraction program or directly construct a set of features[1] from a given set of training images [6-8, 12, 14-20, 22]. Most of them adopt evolutionary computation techniques such as genetic programming (GP) [5, 11]. In such a system, users only need to provide a training dataset (input images and their corresponding answers) and define objective function(s) which is used to evaluate the effectiveness of constructed programs. The system then randomly generates a population of feature extraction programs and evolves them based on the principle of natural selection.

Such an evolutionary system proposed so far can successfully extract several image features such as texture [15], local and global statistical properties [6-7, 18-20], or interest points [17]. This paper, however, focuses on shape which is an image property depicting the form of objects. As a main contribution of this paper, a new evolutionary method that can automatically construct a program for extracting shape features from a given contour is proposed. In the proposed method, a modified version of linear GP [2], which is a variant of GP, is adopted to optimize the performance of feature extraction programs.

The rest of this paper is organized as follows: Section 2 briefly explains the concept of GP and linear GP. Section 3 describes the proposed method. Section 4 presents and discusses experimental results. Section 5 summarizes the paper.

2. Linear Genetic Programming

2. 1 Genetic Programming

GP [5, 11] is a search/optimization paradigm inspired by biological evolution in nature. Until now, GP has been adopted to solve many problems effectively. The key idea of GP is to mimic the mechanism of biological evolution by means of natural selection. In GP systems, a population of computer programs is randomly generated in the first generation. These computer programs can be represented by a representation such as tree, graph, or sequence of primitive operators (POs). Each computer program will be executed to solve a given problem and its performance will be then measured in a process called fitness evaluation. A computer program with a higher performance in solving the problem will have a higher chance to be chosen and reproduced. A new population of programs, called the next generation, is reproduced from the current population by using selection, crossover, and mutation operators. However, reproduction process is not perfect as in biological reproduction. In other words, a reproduced program, often called offspring, is not exactly the same as its original program(s), called parent(s), but they are similar to each other. The next generation of population will be evaluated and reproduced again by using the same process. This will be repeated until an appropriate program is found or some termination criteria are satisfied.

2. 2 Linear Genetic Programming

Linear GP [2] is a form of GP that uses a sequence of instructions, which is called a linear program, as a program representation, which is called a linear program. Execution of a linear program is done sequentially starting from the first instruction, followed by the subsequent instructions. A set of registers is a key component for execution of a linear program. In linear GP, a register is used for three purposes: 1) to store an input of the program, 2) to store an

[1] Related approaches are to automatically construct a classifier that makes a decision from raw image data (actually, feature extraction and decision making are combined into one and are constructed simultaneously) [10] or a program that transforms an input image into a desired output image [1, 13].

intermediate result obtained during the execution, 3) to store an output of the program. An instruction in linear GP must define the index of register(s) from which input(s) will be fetched, the index of a register to which the output will be stored, and a PO that converts the input(s) into the output.

3. Proposed Method

This research focuses on feature extraction process in a simple shape recognition system shown in Fig. 1. The goal of this process is to extract useful features from the contour of an object in the input image. In this paper, a new evolutionary method that can automatically construct a contour feature extraction program is proposed. It adopts linear GP to optimize the performance of feature extraction programs. However, linear GP used in this work is different from the original one as to be described in the following sub-sections.

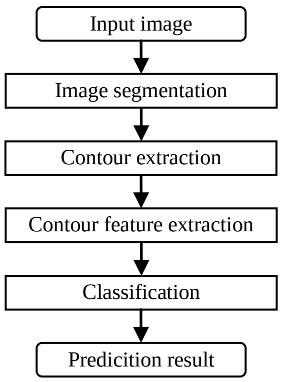

Figure 1. Flowchart of a simple shape recognition system

3. 1 Type of Registers

Two types of registers are used in this research: 1) numerical registers and 2) contour registers. A numerical register, as originally used in linear GP, stores a value which might be a parameter, an input or an output of an instruction. A feature, which is a value that describes an object, is also stored into this type of registers. On the other hand, a contour register, which is a new idea introduced in this paper, is used to store an input contour or a modified contour produced by an instruction. The use of contour registers allows linear GP to evolve an input contour into a more appropriate form for feature extraction.

3. 2 Primitive Operations

Several primitive operators (POs) used in this work are contour processing operations such as a computation of some contour descriptors or even a contour transformation. The use of these POs enables an evolution of contours and contour features. Table 1 presents all POs used in this paper, categorized by the number and type of input(s)/output. The following are descriptions of some key operators[2].

- Perimeter – the number of pixels on the contour
- Area – the number of pixels enclosed by the contour
- Width and height of minimum bounding box (MBB) – the width and height of the smallest rectangle enclosing the contour
- Major axis and minor axis – the width and length of minimum bounding ellipse (MBE), which is the smallest ellipse enclosing the contour
- Diameter – the longest distance between two points on the contour
- Equivalent diameter – the diameter of a circle whose area is equal to the area of the contour
- Circularity – a value measuring how much the contour is similar to a circle. It is defined as follows:

  $$circularity = \frac{4\pi A}{P^2},$$

  where A is the area enclosed by the contour and P is the perimeter of the contour.
- Eccentricity – a value measuring how much a cone deviates from being a circle. It is defined as follows:

  $$eccentricity = \sqrt{\frac{a^2 - b^2}{a^2}},$$

  where a and b are the length of major axis and minor axis, respectively.
- Radius of minimum bounding circle (MBC) – the radius of the smallest circle enclosing the contour
- MBC area – the area of MBC of the contour
- MBE area – the area of MBE of the contour
- Maximum contour-centroid distance – the longest distance between the centroid and a point on the contour
- Minimum contour-centroid distance – the shortest distance between the centroid and a point on the contour
- Convexity defect – the number of concaves on the contour
- Contour vertices – the number of vertices on the contour
- Contour sampling – resampling a contour
- Averaging filter – smoothing a contour by applying an averaging filter to the original contour
- Fourier descriptor – reconstructing a new contour from the first *n* frequency spectrum of the original contour
- Polygonal approximation – computing a polygon that approximates the original contour

3. 3 Genetic Operators

One-point crossover [2, 11] is used to exchange the genetic material between two selected parents. Mutation operator used in this paper consists of three sub-operations: insertion, deletion, and modification. Mutation by insertion inserts a random instruction into a linear program at a random position, while mutation by deletion randomly deletes an

[2] The details of several operators can be found in [3, 9].

instruction from a linear program. Mutation by modification, on the other hand, does not affect the number of instructions but randomly modifies an instruction in a linear program.

3. 4 Fitness Functions

During fitness evaluation process, a linear program will be executed to produce a set of features extracted from a given set of input contours. A classifier is trained and is then used to predict the class of an input contour. To assess the performance of a feature extraction program generated by linear GP, the classification accuracy, as defined in (1), is used as the fitness function.

$$acc = \frac{n_c}{N} \times 100\% \,, \tag{1}$$

where n_c is the number of contours correctly classified and N is the total number of contour images.

4. Experimental Results

An experiment has been conducted to test whether the proposed method can evolve a set of appropriate contour features for a given pattern recognition problem.

4. 1 Image Datasets

The proposed method has been tested to generate contour feature extraction programs for a leaf recognition problem. A leaf image dataset called Flavia [21], consisting of leaf image from 32 species, was used in the experiments. In the experiment, two sets of images were used as follows: 1) eight images from each class (totally 256 images) for training a classifier, and 2) two images per class (totally 64 images) for fitness evaluation. For each image, it must be segmented and its contour must be obtained before using the proposed method (Fig. 2).

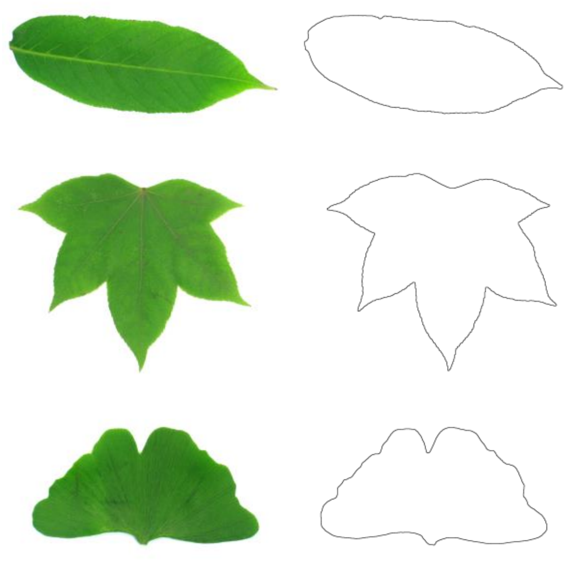

Figure 2. Example of leaf images and their contour.

Table 1. List of primitive operators

One-input Operators	Two-input Operators
Number → Number	**(*Number*, *Number*) → *Number***
sine, cosine, common logarithm, exponential, square root	addition, subtraction, multiplication, division, modulo
Contour → Number	**(*Contour*, *Number*) → *Number***
perimeter, area, width of minimum bounding box, height of minimum bounding box, major axis, minor axis, diameter, equivalent diameter, circularity, eccentricity, radius of minimum bounding circle, area of minimum bounding circle, area of minimum bounding ellipse, maximum contour-centroid distance, minimum contour-centroid distance	convexity defect, contour vertices
Contour → Contour	**(*Contour*, *Number*) → *Contour***
convex hull	contour sampling, averaging filter, Fourier descriptor, polygonal approximation

Table 2. Best-so-far fitness (classfication accuracy (%))

Trial no.	Best-so-far fitness (%)		
	10 features	*15 features*	*20 features*
1	**93.13**	**93.13**	89.38
2	90.63	**91.25**	90.00
3	**90.63**	**90.63**	90.00
4	89.38	**91.25**	88.75
5	**90.63**	90.00	89.38
6	92.50	88.75	**94.38**
7	89.38	89.38	**91.25**
8	88.75	90.00	**91.88**
9	**91.25**	**91.25**	90.00
10	90.00	**91.88**	**91.88**
Average	90.60	**90.75**	90.69

4. 2 Experiment Setup and Parameter Setting

The parameters of linear GP were set as follows. The population size was 1000. The maximum number of generations was 100. The length of an individual was variable but not longer than 50 instructions. The number of numerical registers was 20. The number of contour registers was 20. Tournament selection with tournament size of 5 and the elitist mechanism were used in the selection process. Crossover rate and mutation rate were set to 0.6 and 0.2, respectively. The number of outputs, i.e., the number of features extracted by a generated program, was varied and compared in the experiment (10, 15, and 20). A set of features extracted by a generated program was input into a decision tree [4] for classification. The proposed method has been tested independently in ten trials.

4. 3 Results and Discussion

As shown in Table 2, the classification accuracy of constructed feature extraction programs varied from 88-94%. The number of features, which was the independent variable in the experiment, did not show any effect on the performance of constructed programs as the average accuracies were very close to each other (around 90%). The results indicate the success of the proposed method in automatically constructing a contour feature extraction program for a leaf recognition problem.

5. Summary

This paper proposed a linear GP based method for automatically constructing contour feature extraction programs. Linear GP used in this work uses two types of registers: 1) numerical registers and 2) contour registers which enable the method to evolve an input contour into a more appropriate form as well as to extract contour features. A preliminary experiment has been conducted to evaluate the performance of this method. For a leaf recognition problem, the proposed method could automatically construct a contour feature extraction program with an average accuracy of 90%.

References

[1] A. Aoki and T. Nagao, “Automatic construction of tree-structural image transformation using genetic programming,” Proc. ICAIP-99, Venezia, Italy, pp.136-141, 1999.

[2] M. Brameier and W. Banzhaf, Linear Genetic Programming, Springer, 2007.

[3] L.F. Costa and R.M. Cesar Jr., Shape Analysis and Classficiation: Theory and Practice, CRC Press, 2001.

[4] R.O. Duda et al., Pattern Classification (2nd Edition), Wiley, 2001.

[5] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.

[6] K. Krawiec and B. Bhanu, “Visual learning by coevolutionary feature synthesis,” IEEE Trans. System, Man, and Cybernetics, Part B, vol.35, no.3, pp.409-425, 2005.

[7] K. Krawiec and B. Bhanu, “Visual learning by evolutionary and coevolutionary feature synthesis,” IEEE Trans. Evolutionary Computation, vol. 11. No.5, pp. 635-650, 2007.

[8] K. Lilywhite et al., “A feature construction method for general object recognition,” Pattern Recognition, vol.46, pp.3300-3314, 2013.

[9] F.B. Neal and J.C. Russ, Measuring Shape, CRC Press, 2012.

[10] M.L. Nguyen et al., “Rice leaf detection with genetic programming,” Proc. 2013 IEEE Congress on Evolutionary Computation, pp.1146-1153, 2013.

[11] R. Poli et al., A Field Guide to Genetic Programming, published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk, 2008.

[12] L. Shao et al., “Feature learning for image classification via multiobjective genetic programming,” IEEE Trans. Neural Networks and Learning Systems, vol.25, no.7, July, 2014.

[13] S. Shirakawa and T. Nagao, “Genetic image network (GIN): Automatically construction of image processing,” Proc. IWAIT 2007, pp.643-648, 2007.

[14] A. Song, “Fast video analysis by genetic programming,” Proc. 2008 IEEE Congress on Evolutionary Computation, Hong Kong, China, pp. 3237-3243, 2008.

[15] A. Song and V. Ciesielski, “Texture segmentation by genetic programming,” Evolutionary Computation, vol.16, no.4, pp.461-481, 2008.

[16] A. Teller and M. Veloso, “PADO: a new learning architecture for object recognition,” in Symbolic Visual Learning, K. Ikeuchi and M. Veloso (Eds), Oxford Univ. Press, pp.77-112, 1997.

[17] L. Trujillo and G. Olague, “Automated design of image operators that detect interest points,” Evolutionary Computation, vol.16, no.4, pp.483-507, 2008.

[18] U. Watchareeruetai et al., “Acceleration of genetic programming by hierarchical structure learning: a case study on image recognition program synthesis,” IEICE Trans. Information and Systems, vol.E92-D, no.10, pp.2094-2102, 2009.

[19] U. Watchareeruetai et al., “Multi-objective genetic programming with redundancy-regulations for automatic construction of image feature extractors,” IEICE Trans. Information and Systems, vol.E93-D, no.9, pp.2614-2625, 2010.

[20] U. Watchareeruetai et al., “Redundancies in linear GP, canonical transformation, and its exploitation: A demonstration on image feature synthesis,” Genetic Programming and Evolvable Machines, vol.12, no.1, pp.49-77, 2011.

[21] S.G. Wu et al., “A leaf recognition algorithm for plant classification using probabilistic neural network,” Proc. IEEE ISSPIT 2007, pp.11-16, 2007.

[22] M. Zhang et al., “A domain independent window approach to multiclass object detection using genetic programming,” EURASIP Journal on Applied Signal Processing, vol.8, pp.841-859, 2003.