The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Generating Function of Color Information Detection using Genetic Programming

Takanori Ogawa* Naoki Oshiro** and Hiroshi Kinjo**

*Graduate School of Engineering and Science, University of the Ryukyus
**Faculty of Engineering, University of the Ryukyus
Senbaru 1, Nishihara, Okinawa 903-0213
ogawa@mibai.tec.u-ryukyu.ac.jp

Abstract: This paper proposes a function of color information detection using genetic programming (GP). In the image-processing, object detection is one of important processes. In case the object has complex color domain, the detection becomes more difficult. The authors generate the detection function of complex color domain by using GP. The detection function deals with one pixel of an input image, and it obtains an output image by processing for all pixels. We aim at the time reduction of human consideration of the image-processing system design. In this study, we detect the actual images using the detection function. The results show that the detection function has sufficiently ability for these detection.

Keyword: Genetic Programming, Color Information Detection

1 Introduction

Object detection is one of important processes in the image-processing. In case object has simple color domain, it is possible to detect the object by using basic methods such as the segmentation or the histogram. The detection becomes more difficult when the object has a complex color domain. In this case, an image-processing system is generally constructed by human's trial and error. However, its image-processing system by human consideration depends a specific process. In summery, it is said that the system by human consideration lacks of the generalization ability. Recent studies have reported the filters for the object detection generated by GP or genetic algorithm (GA) [1, 2]. Their methods automatically generates the filters by evolutional process. In the GP system, the generating filter outputs better detection results in image-processing. However, function and terminal nodes in GP are selected by human consideration from among the conventional image filters to solve the problems.

In this paper, our GP method treats the three basic operators of arithmetic and the color informations, without treating the conventional image filters. It generates a function of color information detection. The generating function receives color information from one pixel of an input image, and output a new pixel value into output image. The output image is obtained by repeating this processes for all pixels. Because our method treats only simple operators, we expect that the our proposed method facilitates the construction of GP in the image-processing. We apply this method to detect the complex domain of color images.

2 Detection function evolved by GP

2.1 Genetic Programming

Genetic programming [3] is one of the typical evolutionary method, and it is a method to search the computer programs and the equations for solve the problems. The individuals in GP system can handle tree structures, which is represented with a following S-expression.

$$x^2 + x + 2 \Rightarrow (\ +\ \ (\ *\ \ x\ \ x\ )\ \ (\ +\ \ x\ \ 2\ )\ )\ )$$

Figure 1 shows how to evolve function trees of GP. Trees are updated by repeating crossover and mutation of tree each other.

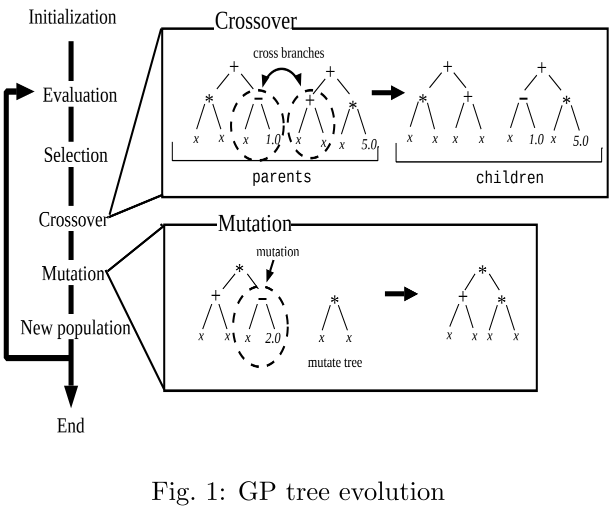


Fig. 1: GP tree evolution

2.2 Image-processing System by GP tree

In our GP system, it generate the function of color information detection while evolutional process. Figure 2 shows the image-processing system using a tree individual evolved by GP. A tree individual recieves the color information from an optional pixel $I(x, y)$ of input image (size= $W_x \times W_y$), and outputs a value O'. Afterwards, O' is segmented to a value of 0 (white) or 1 (black) according to the following threshold, which is set to new pixel $O(x, y)$.

$$O(x,y) = \begin{cases} 1 \text{ (black)} & O' \geq 0 \\ 0 \text{ (white)} & \text{otherwise} \end{cases} \quad (1)$$

Our image-processing system generates an output image by repeating the above process for all pixels.

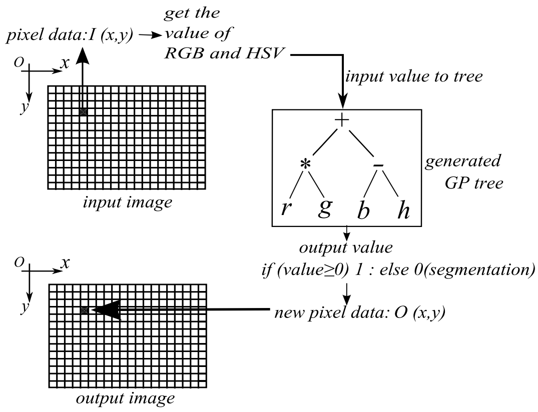


Fig. 2: Image-processing system using GP tree

2.3 Function and terminal nodes

To generate the desired tree structure, we set the function nodes and the terminal nodes as shown in Table 1. The function nodes are the three basic operators of arithmetic, such as addition (+), subtraction (−), multiplication (∗). The terminal nodes are color informations from the optional pixel $I(x, y)$ of the input image, such as values of Red, Green, Blue, Hue, Saturation , Brightness, Luminosity [1]. And moreover, we set the random number which are drawn from the range [-1, 1].

2.4 Evaluation Method

At the begining of evolutional process, we provide the input image, and a target image for the GP (for example: Figure 3). Target image is a mask image of a target object domain. During the evolving, evaluation value E of tree individuals is computed using following [1], and its value is within the range [0, 1].

[1]We consider that the system using GP can detect with an only information of RGB or HSV, whereas we have not achieved its detection yet.

Table 1: Funtion nodes and terminal nodes

node	arity	description
+	2	sum of the branches
−	2	subtract 1 from 2 brances
∗	2	multiply of the branches
r	0	value of Red
g	0	value of Green
b	0	value of Blue
h	0	value of Hue
s	0	value of Saturation
v	0	value of Brightness
y	0	value of Luminosity
rand	0	random number [-1,1]

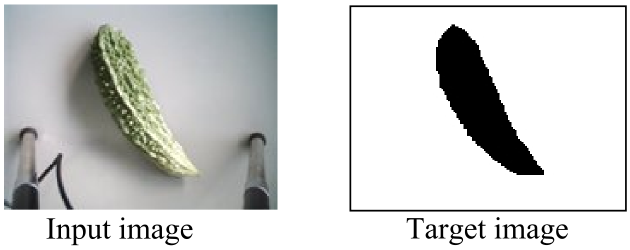


Fig. 3: Training image set

$$E = 1 - \frac{\sum_{x=1}^{W_x}\sum_{y=1}^{W_y} w(x,y)|O(x,y) - T(x,y)|}{\sum_{x=1}^{W_x}\sum_{y=1}^{W_y} w(x,y) \cdot V_{\max}} \quad (2)$$

$$\begin{cases} O(x,y) & : \text{pixel value in the output image} \\ T(x,y) & : \text{pixel value in the target image} \\ w(x,y) & : \text{evaluation weight of each pixel} \\ V_{\max} & : \text{max gradation value} \end{cases}$$

where, evaluation weight value set to $w(x, y) = 1.0$, and gradation value is $V_{\max} = 1.0$.

3 Simulation

To examine the effectiveness of the proposed method, we prepare the samples shown in Table 2. Parameter of GP is shown in Table 3.

Table 2: Sample images

Sample No.	Image name
1	Balsam pear
2	Hand
3	Sea

Table 3: GP parameter set

Number of population	100
Probability of crossover	0.8
Probability of mutation	0.2
Maximum depth after crossover	14

3.1 Evolutionary Result

Figure 4 shows the evaluation results of GP evolving. It can be seen that all processes acquired good evaluation values until the 100th generation. At the end of evolution, evaluation values E became 0.995 (Sample 1), 0.988 (Sample 2), and 0.984 (Sample 3). Figure 5 shows the input images, and output images using detection functions generated by GP, respectively. Sample 1 and 2 are taken the image in indoor environment. The balsam pear of Sample 1 compounds the complex color information which includes many tones of green, yellow, and near to white. In spite of these difficulties, from Figure 5 (a), it observed that generating detection function could detect the target object. Similarly, the hand of Sample 2 has the complex color domain and complex background, and, from Figure 5 (b), the detection function can detect the target object. Moreover, image of Sample 3 is a landscape image. This image compounds the complex color information both target object and background, and it includes color domain of the sky which is near to color information of target object of the sea. Although we considered that detection of this image is difficult, the sea is detected by the detection function. From these results, we confirmed that it is possible for the detection function evolved by GP to obtain sufficient ability for these detection. Figure 6 shows the detection result of an untrained image using detection function generated by Example 1. From this figure, it can be seen that the detection function could detect not only trained images, but also color domain of the balsam pear in the untrained image. Therefore, it is clear that the generated detection function has the generalization ability.

3.2 Characteristic of Tree Structure

In this paper, we examine the characteristic of tree structure evolved by GP. S-expression in Figure 7 shows the tree for Sample 1. Tree depth is 13, and number of nodes is 77. We set an examination process as follows First, we set an individual node-number to each node like this.

$$(+\ \ r\ (*\ \ g\ \ b)) \Rightarrow \text{Node-number:}(0\ \ 1\ (2\ \ 3\ \ 4))$$

Next, we exclude an optional branch according to each node-number. Finally, after exclusion, it computes evaluation value of its tree. In this process, we considered that the branch which decreases evaluation value largely is important in the tree.

Figure 8 shows the change of the evaluation value when tree excludes optional branch. From this figure, evaluation value decreased to the smallest value at the node-number 1 ($E = 0.0049$), and next was node-number 16 ($E = 0.1412$). Because the evalua-

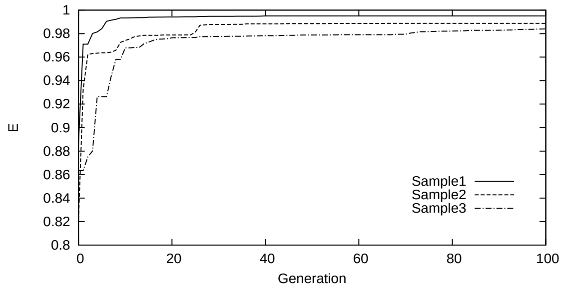
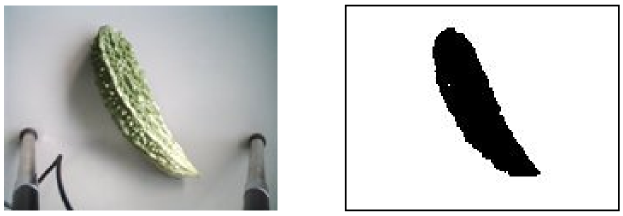
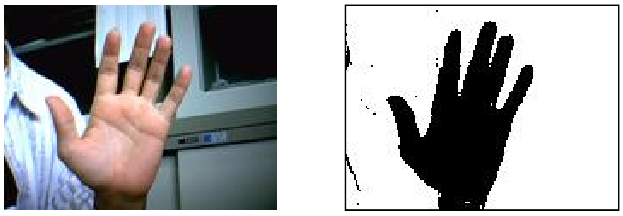


Fig. 4: Evaluation value of GP evolving



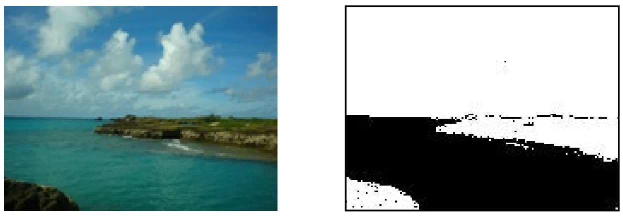
Input image Output image

(a) Sample 1 : a balsam pear ($E = 0.995$)



Input image Output image

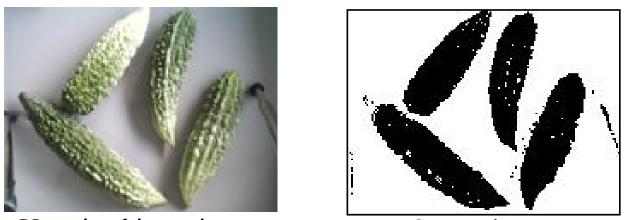
(b) Sample 2 : a hand ($E = 0.988$)



Input image Output image

(c) Sample 3 : the sea ($E = 0.984$)

Fig. 5: Results of detection



Untrained input image Output image

Fig. 6: Detection of the untrained image

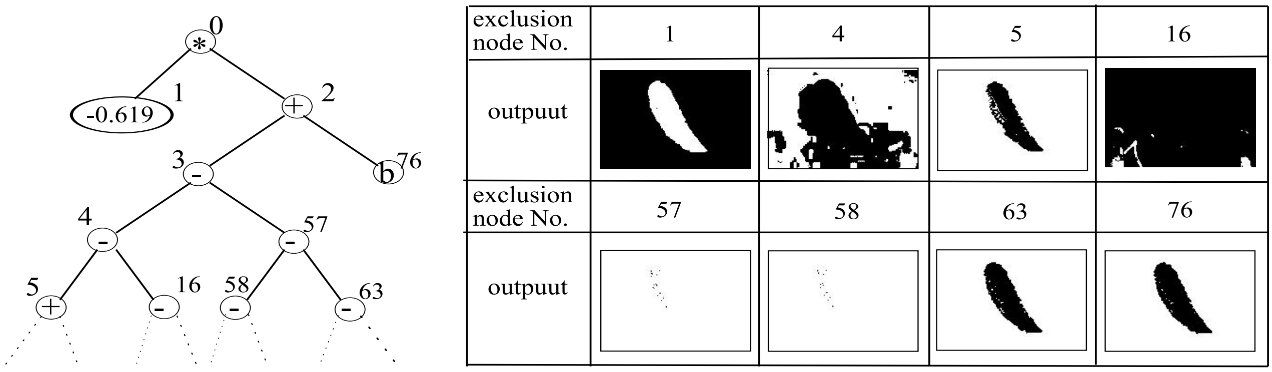


Fig. 9: Examine the tree structure

(* -0.6190547(+(-(-(+(+(-(- r s) v)(- r y)) v)(-(- y(+(* y(*(+(+ s r) s)(-(- r (+ v v) v))(-(-(- g s) v) y)))) 0.5004493))(+ b(*(- r g)(* r(- g(- b g)))))))(- (- g(- b g))(-(+(+(* -0.6292867 v) h) b)(+ s(+ g v))))) b))

Fig. 7: Generation tree of Sample 1

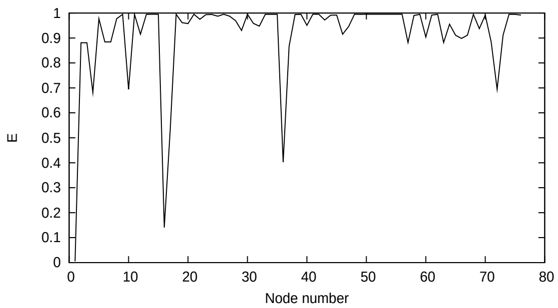


Fig. 8: Change of evaluation value

Table 4: Exclusion branch

node-No.	exclusion branch
1	-0.6190547
16	(-(- y(+(* y(*(+(+ s r) s)(-(- r(* (+ v v) v))(-(-(- g s) v) y)))) 0.5 004493))(+ b(*(- r g)(* r(- g(- b g))))))

tion value decreased largely by excluding their branches, we consider that their branches are important in the tree. Their branches are shown in Table 4. Furthermore, we investigated output images when tree excludes its important branches. Figure 9 shows a part of the tree structure, and shows output images after excluding branches. This tree structure includes the branches of node-number 1 and 16. From output images in the figure, at the excluding branch of node-number 1, tree does not detect the target object, whereas detect the background. Its output image is an inversed image of target image. It can be seen that this excluding branch has a role which reverses a sign of output value from tree. At the excluding branch of Node-number 16, tree not only detect the target object but also background of excepting color domain near to black in input image. Namely, we considered that its branch processes the background domain. Also, from other output images in this figure, it can be confirmed that each branches has various roles in the tree. Therefore, it is clear that generating tree by GP is composed of each branches which has various roles.

4 Conclusion

In this paper, we proposed a method of function generation for color information detection using GP. We set the function nodes and terminal nodes that contain three basic operations of arithmetic and color informations. The evolutionary results showed that GP could generates the function to detect the target domain. Moreover, we confirmed that the branch in generated tree structure influences each other, and its branches compose one detection function for image-processing system.

References

[1] T. Nagao, "Evolutionary image processing", Syouko-do, 2002 (in Japanese).

[2] M. Maezono, S. Ono, S. Nakayama, "Automatic Parameter Tuning and Bloat Restriction in Image Processing Filter Generation Using Genetic Programming", Trans. of JSCES, Paper No.20060021, 2006.

[3] J. Koza, "Genetic Programming, On the Programming of Computers by means of Natural Selection", MIT Press, 1992.