An artificial immune approach for optical image based vision inspection

Hong Zheng (郑 宏)[1], Nanfeng Xiao (肖南峰)[2], and Jinhui Lan (蓝金辉)[3]

[1]*School of Electronic Information, Wuhan University, Wuhan 430079*

[2]*School of Computer Science & Engineering, South China University of Technology, Guangzhou 510641*

[3]*Department of Precision Instruments, Tsinghua University, Beijing 100084*

Received September 6, 2002

This paper presents a novel approach of visual inspection for texture surface defects. The approach uses artificial immune theory in learning the detection of texture defects. In this paper, texture defects are regards as non-self, and normal textures are regarded as self. Defect filters and segmentation thresholds used for defect detection are regarded as antibodies. The clonal selection algorithm stemmed from the natural immune system is employed to learn antibodies. Experimental results on textile image inspection are presented to illustrate the merit and feasibility of the proposed method.

OCIS codes: 100.5010, 150.3040.

Quality inspection is an important problem in many production processes. There is a great number of manufacturing processes where inspection for surface finishing or surface defects is attempted, such as textile, paper, and steel surfaces. The most difficult task of inspection is that of inspecting for visual appearance. As human visual inspection is slow, expensive and erratic, optical image based automated visual inspection has become the alternative to the human inspector[1].

This paper is concerned with the problem of vision inspection of texture surfaces. In the past years numerous approaches have been developed for texture inspection[2]. However, textures in the real world are often not uniform, due to changes in orientation, scale or other visual appearance. How to extract robust texture features has become a key issue in the field of texture inspection. In recent years, some researchers presented approaches based on Gabor filters, autoregressive random field model and local binary patterns to extract texture features which are not invariant to changes in rotation or scale[3]. Unfortunately, the degree of computational complexity of these proposed texture measure is high. In order to solve the problem, this paper presents a novel approach to extract robust defect texture features. A clonal selection algorithm stemmed from biology immune system is employed to rapidly learn texture defect filters and segmentation thresholds. These learned filters and thresholds can be used to detect various defects on the same type texture surfaces. Due to computational simplicity, the approach is very suitable for industrial application.

The nature immune system is one of the most intricate bodily systems and its complexity can be compared to that of the brain. With the advances in the biology and molecular genetics, the comprehension for the immune system behaves is increasing very rapidly. The knowledge about the immune system functions has unravelled several of its main mechanisms. These mechanisms have demonstrated to be very interesting not only from a biological standpoint, but also under a computational perspective. Similarly to the way the nervous system stemmed the development of artificial neural networks, the immune system has now led to the emergence of artificial immune systems (AIS) as a novel computational intelligence paradigm.

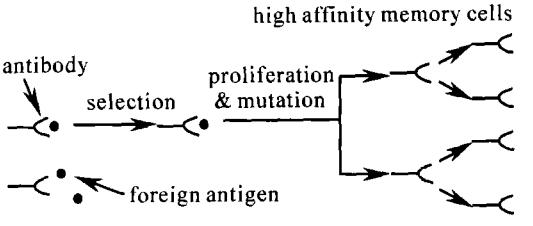

Fig. 1. The clonal selection procedure.

Clone selection algorithm (CSA) is one of main computational models developed by nature immunological mechanisms. It is inspired from self-nonself discrimination in the immune system. It is achieved by B-cells, which produce memory antibodies with high affinity. The whole selection procedure includes proliferation and mutation, which are known as the maturation of the immune response and is analogous to the natural selection of species[4]. Figure 1 illustrates the clonal selection, proliferation and affinity maturation processes.

In this paper, texture defects are regarded as antigens. Defect detection parameters are regarded as antibodies. The clonal selection principle is employed to evolve antibodies. The algorithm details are described as follows.

Firstly, the immunological terms are defined as

Antigen: any of training texture images.

Antibody: a float string encoded by filter parameters and a segmentation threshold. Figure 2 illustrates an antibody architecture.

Affinity: the percentage of correct detection of an antibody. It is defined by

$$p = \frac{\text{Number of correctly recognized images}}{\text{Total number of training texture images}} \times 100\%. \quad (1)$$

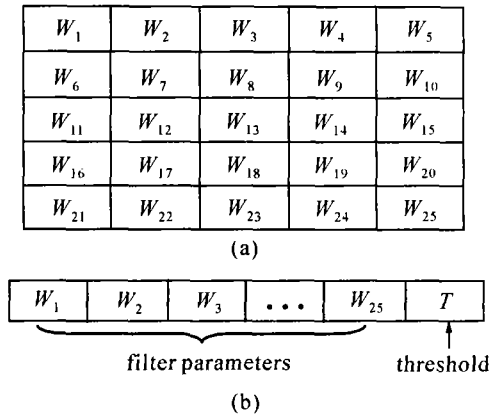

Fig. 2. Antibody encoding scheme. (a) A 5 by 5 filter architecture, $W_i \in [-2,2], i = 1,2,\cdots,25$. (b) An antibody architecture, $T \in [0,512]$.

The greater the value of p, the higher the antibody affinity.

For each antibody, the procedure to recognize texture images consists of following three steps:

1) Decode an antibody and get a filter and a threshold.

2) Convolve the all training images by the filter. The 2D convolution of the image $I(i,j)$ and filter $A(i,j)$ with size $2a+1$ by $2a+1$ is given by the relation

$$F(i,j) = A(i,j) * I(i,j) = \sum_{k=-a}^{a} \sum_{l=-a}^{a} A(k,l) I(i+k, j+l). \quad (2)$$

For a 5 by 5 filter, a is 2.

3) Calculate the standard deviation of each convolved training image. If a standard deviation is greater than the threshold, the image corresponding to the mean is recognized as a defect texture, otherwise, it is recognized as a defect-free texture.

Secondly, proper immune operations should be decided. Basically, the CSA includes three immune evolutionary operators, which are described as follows.

Clone: This operation is to generate copies of every individual in an antibody population proportionally to its affinity with the antigen. All individuals are sorted in descending order firstly. The amount of clones of an antibody is given by

$$n_i = \text{round}\left(N \times \frac{f_i}{\sum_{i=1}^{N} f_i} \right), \quad (3)$$

where N is the number of all individuals in an antibody population. f_i is the affinity value of the ith antibody. Obviously, the higher the affinity, the greater the number of copies, and vice-versa.

Mutation: The mutation operation creates a new antibody by randomly changing one or more of the unit values in the antibody with a probability based on its affinity. The mutation probability is given by

$$P_i = \frac{f_{\max} - f_i}{f_{\max} - f_{\min}}, i = 1,2,\cdots,N, \quad (4)$$

where $f_{\max}$ is the highest affinity value, and $f_{\min}$ is the lowest affinity value.

From Eq. (3) and Eq. (4), we can see that antibodies with high affinities in a population always have more clone copies and less mutation probabilities than that with low affinities. This is correspond to nature evolutionary mechanism.

Reselection: This operation sorts all individuals in descending order, and replaces the m lowest affinity antibodies with m new randomly generated antibodies.

The whole learning algorithm is described as follows:

1) Randomly generate an antibody population (M) which represent a set of candidate filters and segmentation thresholds;

2) Evaluate the affinity of each antibody in the population with Eq. (1);

3) Generate clone copies of all individuals with Eq. (3);

4) Mutate all these copies with Eq. (4);

5) All individuals are sorted in descending order, and replace the m lowest affinity antibodies in M with m new randomly generated antibodies;

6) Repeat steps 2 to 5 until a given iteration times is met.

After learning procedure, the acquired antibody (the texture filter and threshold) is used to detect antigens (texture images).

Firstly, according to Eq. (2), convolve the test image by the learned filter. Secondly, calculate the standard deviation within a $2n+1$ by $2n+1$ window at each pixel in the image. The standard deviation is defined as the texture feature TE at the pixel. It is given by

$$\text{TE}(i,j) = \frac{1}{(2n+1)^2} \times \sqrt{\sum_{k=i+n}^{i+n} \sum_{l=j+n}^{j+n} |F(k,l) - \overline{F(i,j)}|}, \quad (5)$$

where $n = 9$, $1 \le i \le H$, $1 \le j \le W$, W and H represent the width and height of an image, respectively. Thirdly, compare the TE value at every pixel with the learned threshold. If the TE is greater than the threshold, the pixel belongs to a defect texture region. Otherwise, it belongs to a defect-free texture region.

We do experiments on textile images. The training textile images used in the experiment are from the TILDA textile texture image database created at the University of Freiburg, Germany. In TILDA, textile samples are grouped as defect-free or having a certain type of defect. For each defect type, defect-free samples are provided. In our experiment, we learned a defect filter and segment threshold for each texture type. For each texture type, a total of 40 different size sample images with different defects are selected for learning purpose. These images are collected from the same type textile images, and they are divided into two texture classes. One of them belongs to defect-free texture samples, the other belongs to defect texture samples. Figures 3(a) and (c) show two type defect texture samples under different orientations, scales and shapes. Figures 3(b) and (d) show two type defect-free texture samples under different orientations, scales and shapes.

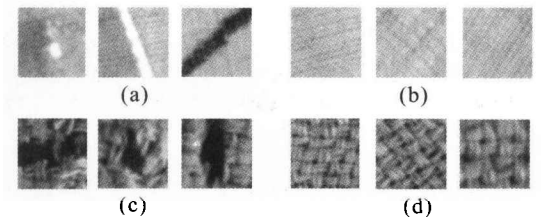

Fig. 3. (a) Defect texture samples from textile set C1R1. (b) Defect-free texture samples from textile set C1R1. (c) Defect texture sample from textile set C2R2. (d) Defect-free texture samples from textile set C2R2.

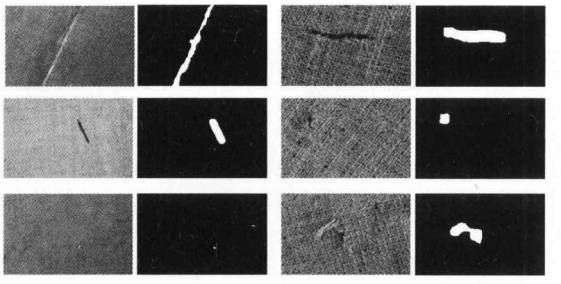

Fig. 4. (a) Tested images from textile set C1R1. (b) Inspection results for images in (a) (white indicates defects). (c) Tested images from textile set C2R2. (d) Inspection results for images in (c) (white indicates defects).

Table 1. Performance Comparison of GA and CSA

Algorithm	Convergence Generations	Processing Time (s)
GA	23	58
CSA	8	19

The experiment of texture defect detection is performed in two stages. The first stage is the training phase where a defect filter and a segmentation threshold are acquired through learning the training texture samples using CSA. By learning sample image set C1R1 and C2R2, we can get an optimal filter and threshold. In our experiments, CSA is compared with standard genetic algorithms (GA). Table 1 shows the comparison results of average convergence generations and processing time of 50 learning experiments. It can be seen that CSA can find the optimal solution only after 8 generations, while GA cannot find the optimal solution until the 23rd generation. In addition, CSA can save much more processing time than GA. This illustrates that the searching ability of CSA is stronger than that of standard GA. It is because selection and reproduction mechanism adopted by CSA maintains solutions' diversity so that CSA is easy to jump out of local optima to explore better solutions. The second stage is to detect defects using learned filters and thresholds. Figures 4(b) and (d) show the results of extracting defects from images shown in Figs. 4(a) and (c), respectively. From Fig. 4, it can be seen that, although defect appearances on different textile surfaces are quite different, they can be extracted correctly by proposed method. We tested our algorithms on 200 textile images. The correct rate can reach 95.4%.

In this paper we have described an approach for defect detection using learning techniques with clonal selection principle. Experimental results show that nature immune mechanism can be used to build novel computational tools to solve problems in the field of visual inspection.

We plan to further explore the same approach and find general principles to detect various surface defects in industrial images.

This project was partially supported by the National Natural Science Foundation under grant No. 40271094. H. Zheng's e-mail address is zhenghong@21cn.com or hzheng@deakin.edu.au.

References

1. T. S. Newman and A. K. Jain, Computer Vision and Image Understanding **61,** 231 (1995).
2. K. Y. Song, M. Petrou, and J. Kittler, Proc. SPIE **1708,** 99 (1992).
3. D. M. Tsai and B. Hsiao, Pattern Recognition **34,** 1285 (2001).
4. L. N. De Castro and F. J. Von Zuben, Technical Report, RT DCA 01/99 (Brazil, 1999).