Automatic Test Image Generation by Genetic Algorithms for Testing Halftoning Methods - Comparing Results using Wavelet Filtering

Timo Mantere and *Jarmo T. Alander*

University of Vaasa
Department of Information Technology and Production Economics
P.O. Box 700, FIN-65101 Vaasa
FINLAND
E-mail: *firstname.lastname*@uwasa.fi

ABSTRACT

This work introduces automatic test image generation by genetic algorithms for testing different halftoning methods. In general, the proposed method has potential in software test data generation. The goal was to reveal, if genetic algorithm is able to generate images that are difficult for the object software to halftone, in other words to find if some prominent characteristics of the original image disappear or ghost features appear due to the halftoning process. Using a Haar wavelet based fitness function did image comparison between the input image and the corresponding halftoned image.

1. INTRODUCTION

There does not seem to be much research in the field of test image evaluation. How to determine or create a good test image. What are the essential characteristics of a good test image? How to determine that a particular image is good for testing some specific image processing algorithm? More often than not researchers rely on commonly used but very limited test image sets. We encountered this problem, when we wanted to test the image-processing system we implemented for an ink jet marking machine [1, 2]. In our other study [3, 4, 5] we used genetic algorithms (GA) for software testing purposes. In this work, we try to combine the knowledge of these two previous studies in order to use GA for generating test images for halftoning methods. This study concentrates on finding how wavelets can be adapted to image comparison as essential component of the fitness function.

1.1. Genetic algorithms

Genetic algorithms [6] are optimization methods that mimic evolution in nature. They are simplified computational models of evolutionary biology. A GA forms a kind of electronic population, the members of which fight for survival, adapting as well as possible to the environment, which is actually an optimization problem. GAs use genetic operations, such as selection, crossover, and mutation in order to generate solutions that meet the given optimization constrains ever better and better. Surviving and crossbreeding possibilities depend on how well individuals fulfill the target function. The set of the best solutions is usually kept in an array called population. GAs do not require the optimized function to be continuous or derivable, or even be a mathematical formula, and that is perhaps the most important factor why they are gaining more and more popularity in practical technical optimization. The genetic algorithm (GA) in this study was written in Java. One of the advantages of Java is its easiness to use image handling procedures. However, the execution speed of Java programs may not be the best possible.

1.2. Dithering

Digital halftoning [7], or dithering, is a method used to convert continuous tone images into images with a limited number of tones, usually only two: black and white. The main problem is to do the halftoning, so that the bi-level output image does not contain artifacts, such as alias, moiré, lines or clusters, caused by dot placement [8]. The average density of the halftoned dot pattern should interpolate as precisely the original image pixel values as possible. Dithering methods include static methods, where each pixel is compared to a threshold value that is obtained e.g. from a threshold matrix, generated randomly or is a static median value. Depending on matrix this method can create both frequency or amplitude modulated halftones. There are also error diffusion methods, such as Floyd-Steinberg and Jarvis-Judge-Ninke coefficients. In these methods the rounding error of the current pixel is spread on those neighboring pixels, the bi-level value of which is not yet determined.

This study concentrates only on frequency modulated halftoning methods. The halftoning methods used here were Floyd-Steinberg (`FS`), and Jarvis-Judge-Ninke (`JJN`) error diffusions and thresholding with 16×16 ordered threshold matrix [7] (`THO`), and with GA optimized 16×16 threshold matrix [2] (`THG`). Also rounding (`NEAR`) the nearest bi-level tone (black/white) was used to compare results with, since it should lead rather poor halftone result and therefore lead worst results with each target function.

1.3. Haar Wavelet

The wavelet transforms [9] are signal processing operations that decompose signals into components at different frequency scales. A wavelet transform represents a sum of wavelets on different locations and scales. It is based on multiresolution analysis. The most well known and simplest wavelet is Haar function (filter). The characteristic property of Haar function is sharp edges. Haar filter is special case of Daubechies filter family; it is actually order one Daubechies filter, and the only one of that orthonormal filter family that has explicit expression. Decomposition in the Haar basis eliminates high frequency terms when the input sequence is constant. Haar function is often used for images with high contrast of black and white; therefore, we can assume Haar function well suitable when applied to halftoned images.

1.4. Comparing the images

Comparing a dithered image with the original one is obviously a challenging problem. One cannot simply use pixel by pixel comparison, since dithered images usually have only two tones. The minimum difference by that measure would be achieved if every gray tone were rounded to the nearest tone (black or white), which in practice usually results in poor images. Better image comparison methods have been developed [7, 10].

In addition, a set of methods called inverse halftoning [7] has been developed. From these the perhaps most common is the low pass filtering method. In this method, images are first low pass filtered and the resulting images are then compared pixel by pixel. The problem with low-pass filtering is that the high frequencies will disappear and the images get a somewhat blurred overall appearance. However, this method is easy to implement and it enables pixel by pixel comparison. In a way the blurring by low pass filtering also resembles human eye perception: when we look the image from a distance the small details disappear and the visual observation of larger objects is averaged out from the small details.

If the images are not compared properly, the received evaluated difference between images may as well depend on the comparison method used as the actual difference between the images, i.e. the dithering methods used. Several fitness functions i.e. image comparison methods were tested in refs. [11, 12].

This study concentrates on using Haar wavelet in the comparison. Since the original and halftoned image represents the same image, Haar wavelet coefficients of them should be related. The lower frequency coefficients should be quite similar, since the average gray for both images should be approximately the same. The higher the frequency the more the coefficients are likely to diverge. A certain weight coefficient for each frequency scale is used to determine the significance of different level wavelet coefficient difference.

1.5. Related work

Wavelets have been applied for finding similarities on images i.e. image comparison in refs [13, 14]. In the previous studies on finding optimum halftone patterns the human eye modulation transform function [15] is considered the best method, while especially optimization speed may favor more simple methods.

Genetic algorithm were previously adapted to the dithering problem [16, 17]. For further references of GAs in image processing see e.g. bibliography [18] or book [19]. Image generation with GA is used at least in ref. [20]. Image generation for algorithm validation is represented in ref. [21]. GAs has previously been adapted to automatic software test data generation in several studies, see refs. [3–5] and references therein.

2. THE PROPOSED METHOD

This work is a continuation to that given in refs. [11, 12]. The image comparison in those papers were done using a) pixel by pixel comparison using low pass filtered images, b) tone difference between consecutive pixels in each image, c) the average density at the corresponding image areas, d) a hybrid of the three previous methods, and e) edge detector and comparing edge locations.

The GA runs as an independent program and optimizes parameter vectors which are used by an image generator to create images, which are further sent to the object software, that halftones it and returns the resulting image. The pixelgrapper reads pixels from both the test image and its halftoned transformation image and transmits 8-bit pixel arrays of both images to the fitness function evaluator. The difference between these images is used as the fitness function. GA generates new parameter vectors by using crossover and mutation, favoring those parent chromosomes that previously had gotten a high fitness value. Test images in this study were created by optimizing parameters, such as place, size and color of elementary graphical objects, like lines, rectangles, circles and ASCII characters, together with the background tiles and colors all encoded as one GA chromosome.

2.1. Implementation

The implementation used integer coded GA, where the chromosome consisted of total 79 parameters. From those the first seven parameters were for background, three of them break background into four segments and the other parameters determine the tone b of each background segment. This way one parameter does not dominate optimization. However, the background might still become monotone if one segments takes the whole space or the tone parameters b_n are equal. Next 70 parameters where divided into 10 groups of 7 parameters, each 7 parameter long group defines one elementary image object as follows:

1. Image object (line, rectangle, oval, ASCII character); for characters also the font style.
2. Object color.
3. Object starting point; x coordinate.
4. Object starting point; y coordinate.
5. Object length in x coordinate direction or character font size.
6. Object length in y coordinate direction or character font type.
7. Not used or the character value (only printable ASCII characters were used).

All objects are opaque and may cover partly or totally earlier created objects. Background is created first and then the other objects on it.

The generated image as such was still quite monotonous. Normal image usually has more variation between neighboring pixels. Our test image was further diversified by adding chaotic data with Verhulst [22] logistic equation: $x_i = ax_{i-1} \times (1 - x_{i-1})$. The chaotic data was used rather than random noise in order to control diversity and to keep the added noise repeatable. The last two parameters of the chromosome forms 16-bit value *a* for Verhulst function that was scaled to be a decimal number in range [2, 4]. The optimization process usually favored such chaos parameters that generated striped patterns rather than patterns that resemble uniform random noise.

The size of the generated image was selected to be 256×256 pixels, so that the values of most parameters would fit into eight bits. Population size was 50, elitism 40%, total 3050 evaluations (initial population + 100 generations) were done, uniform crossovers was used, and the mutation probability was 1%.

3. EXPERIMENTAL RESULTS

These experiments concentrated on finding a target function based on Haar wavelet coefficients. Target function 1 is given by equation (1). The notations used are the following: *A* = original image, *B* = halftoned image, *S* = wavelet coefficients, *K* represents different scales (levels), *i* and *j* are indices to the wavelet coefficients of particular level, *W* is weight coefficient for difference of wavelet coefficients, and *X* represents the threshold against which the wavelet coefficient are compared with. This target function compares images by counting the amounth of wavelet coefficients that differs from each other over some threshold value, which may be different for different frequency scales. In practice if the sum is small the images are similar to each other.

$$\sum_K \left(W_K \times \left\{ \sum_{i,j} \left| [A.S_K(i,j) - B.S_K(i,j)] \right| > X_K \right\} \right) \quad (1)$$

Haar wavelet coefficients for the original and halftoned images should be quite similar at large scales, however the difference between them tends to increase the higher the frequency scale. However, when comparing images the coefficient similarity for the lower frequency scale is more important.

Table 1 represents the maximal fitness values obtained using formula 1 for each test image set. Standard test image set (STD) contains 13 test images [23] {Airplane, Barbara, Bird, Boat, Bridge, Camera, Frog, Goldhill, Lenna, Mandrill, Peppers, Washsat, and Zelda} that are often used for expressing image processing systems. Random noise images (RN) contained 10 gray noise images generated by a random generator. One tone images (BCR) contains 256 possible 8-bit one tone gray images. Test images (GI) generated by genetic algorithm contained best values obtained from five different GA optimization runs.

Table 1. Best fitness values for different test image sets with target function 1.

	FS	JJN	THO	THG	NEAR
STD	280827	396177	324519	331283	518184
RN	268107	286279	371756	400798	402969
BCR	301234	348930	262144	259072	16386
GI	369576	471069	500874	465104	525869
Max(STD)	Peppers	Goldhill	Mandrill	Bridge	Frog
Max(BCR)	30, 225	60, 195	∈ [86, 170]	162	∈ [1, 254]

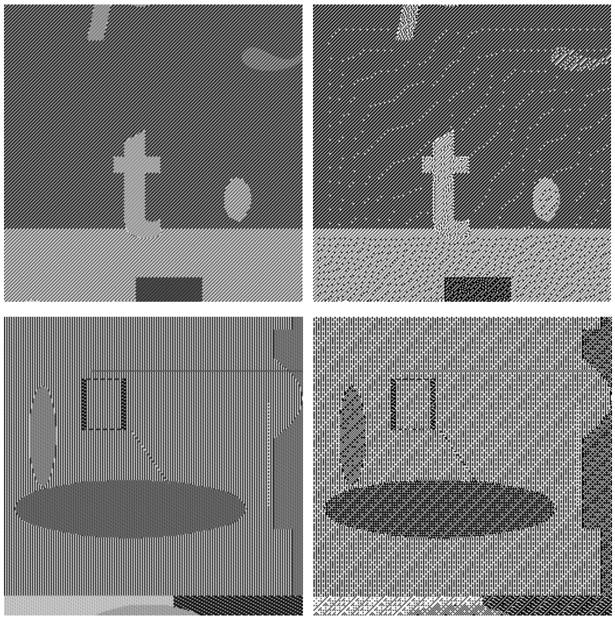

Fig. 1. An example of a GA generated test images and the corresponding halftoned images
a) Test image for FS. b) Dithered a.
c) Test image for THO. b) Dithered c.

With this target function GA was able to generate test image that resulted proportionally highest fitness value for each of the tested halftoning methods. This implies that our GA based image generator can optimize test images according to the given function.

From table 1 we can further see that according to the target function FS is best dithering method for all other test images, except for one images (BCR). What is notable here is that random noise is not the worst case for any halftoning method. If we want to analyze the test images generated by GA and evaluate what properties caused the high difference between the compared images we want them to have some properties that a human eye can observe. If our target function considered random noise the worst case we would not be able to generate any sensible test images. Random noise is still random noise after halftoning and human eye does not see that much difference when comparing original noise and halftoned one.

Figure 1 shows examples of GA generated test image. With FS and JJN the typical test images shows a kind of ghost worms c.f. (fig. 1b). With the ordered threshold matrix method the test images caused the kind of behavior where the vertical stripes caused by certain chaos parameter in the original image are changed into crosswise stripes when halftoned (fig. 1d).

4. CONCLUSIONS AND DISCUSSION

The results got seem to confirm that our genetic algorithm based image generator is capable of generating test images for testing different halftoning methods according to a given target function. With most cases, GA was able to generate test images that resulted highest difference score. In the cases were GA did not generated highest value it still reached very close to the highest value obtained with our static test image set.

In most cases, the halftoned images show some properties that evidently differ from the original images. This supports the proposal that wavelet based image comparison methods are worth considering.

4.1. Future

However there may not exist only one universal “right” way to compare halftoned images with originals. Different target functions may discover different kinds of dissimilarities between images. One future research alternative is to find a good set of comparison functions that together discovers all possible different types of dissimilarities.

The use of wavelets as a hybrid with other methods could be studied. The use of also other than Haar wavelets in image comparison could be studied. The possibilities of applying fuzzy logic to image comparison is under study.

GA coding can be improved. Integer coded GA may not be the most suitable for this problem. It is planned that future version will more freely create desired objects. More massive test runs may eliminate the bias of background tone dictation. The significance of other objects and their position in the image may be identified if we use static background tones and let other features settle.

However, so far the work is been mostly experimental, the goal has been to solve what this kind of optimization approach results in software testing, and how the method could be further improved.

After a satisfying fitness function has been found, the obvious application of the above testing method is automatic dithering method design. One GA generates halftone filters while the other GA tries to create the hardest test image for each filter. The best filter being the one where the hardest test image is closest to the original after dithering. In general, this kind of differential evolution based approach could be used in the design and testing of demanding software.

REFERENCES

[1] J. T. Alander, T. Mantere, and T. Pyylampi, Threshold matrix generation for digital halftoning by genetic algorithm optimization. In D. P. Casasent, ed., *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, volume SPIE-3522*, Boston, MA, 1.-6. November 1998. SPIE, 1998, pp. 204-212.

[2] J. T. Alander, T. Mantere, and T. Pyylampi, Digital halftoning optimization via genetic algorithms for ink jet machine. In B. Topping, ed., *Developments in Computational Mechanics with High Performance Computing*, CIVIL-COMP Press, Edinburg, UK, 1999, pp. 211-216.

[3] J. T. Alander, T. Mantere, G. Moghadampour and J. Matila, Searching protection relay response time extremes using genetic algorithm – software quality by optimization. *Electric Power Systems Research 46*, 1998, pp. 229-233.

[4] J. T. Alander, and T. Mantere. Automatic software testing by genetic algorithm optimization, a case study. In C. Ryan and J. Buckley (eds.) *SCASE'99 - Soft Computing Applied to Software Engineering*, 11.-14.4.1999, Limerick, Ireland, 1999, pp. 1-10.

[5] T. Mantere, and J. T. Alander. Automatic Software Testing by Optimization with Genetic Algorithms Introduction to the Method and Consideration of the Possible Pitfalls. Submitted to *MENDEL 2001*, June 6-8, 2001, Brno, Check Republic, 5 pages..

[6] J. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor, MI, Reissued by The MIT Press, 1992.

[7] H. R. Kang, *Digital Color Halftoning*. SPIE Optical Engineering Press, Bellingham, Washington, & IEEE Press, New York, 1999.

[8] P. G. J. Barten, *Contrast Sensitivity of the Human Eye and Its Effects on Image Quality*. SPIE Optical Engineering Press, Bellingham, Washington, USA, 1999.

[9] J. C. Goswami and A. K. Chan. Fundamentals of Wavelets – Theory, Algorithms, and Applications. John Wiley & Sons.

[10] F. Nilsson, Objective quality measures for halftoned images. *Optics, Image, Science and Vision*, Volume 16, Number 9, September 1999, pp. 2151-2162.

[11] T. Mantere, and J. Alander. Automatic test image generation by genetic algorithms for testing halftoning methods. In D. P. Casasent, editor, *Intelligent Systems and Advanced Manufacturing: Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision, volume SPIE-4197*, Boston, MA, 5.-8. November 2000, SPIE, Bellingham, Washington, pp. 297-308.

[12] T. Mantere, and J. Alander. Testing Halftoning Methods by Images Generated by Genetic Algorithms. Submitted to *Arpakannus*, 2001.

[13] M. K. Mandal, T. Aboulnasr, and S. Panchanathan. Image indexing using moments and wavelets. *IEEE Transaction on Consumer Electronics*, Vol. 42, No. 3, August 1996.

[14] J. Ze Wang, G. Wiederhold, O. Firschein, and S. Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. IEEE 1997.

[15] J. Sullivan, L. Ray and R. Miller (1991). Design of Minimum Visual Modulation Halftone Patterns. In IEEE Transactions on Systems, Man, and Cybernetics, *Vol. 21, No 1*, January/February 1991.

[16] N. Kobayashi, and H. Saito, Halftoning Technique Using Genetic Algorithms. In *Systems and Computers in Japan* ***27***, 1996, pp. 89-97.

[17] J. Newbern, and V. M. Bove, Jr., Generation of blue noise arrays using genetic algorithm. In *Human Vision and Electronic Imaging II*, B.E. Rogowitz and T.N. Pappas, eds., vol SPIE-3016, SPIE, Bellingham, San Jose, CA, 10.-13. Feb. 1997, pp. 401-450.

[18] J. T. Alander, *An Indexed Bibliography of Genetic Algorithms in Optics and Image Processing*. Department of Information Technology and Production Economics. University of Vaasa. Report Series No. 94-1-OPTICS, 2000, Available ftp://garbo.uwasa.fi /cs/report94-1/gaOPTICSbib.ps.Z

[19] S. K. Pal, A. Ghosh, and M. K. Kundu, *Soft Computing for Image Processing*. Physica-Verlag, Heidelberg, New York, 1999.

[20] K. Sims, Artificial Evolution for Computer Graphics. *Computer Graphics (Siggraph '91 Proceedings)*, July 1991, pp.319-328.

[21] M. Miyojim, and H. Cheng, Synthesized images for pattern recognition. In *Pattern Recognition*, Vol. 28, No. 4, 1995, pp. 595-610.

[22] P. S. Addison, *Fractals and chaos: An illustrated course*. Bristol, Philadelphia, Institute of Physics Publishing, 1997.

[23] Michael Frydrych. Test images. Available http://www.it.lut.fi /research/ip/images/Standard/index.html