A Genetic Algorithm for Thermal Image Deconvolution

Marius Marcu and Mircea Vladutiu

***Abstract*—A genetic based algorithm for deconvolution of Printed Circuit Board (PCB) thermal images is presented. The deconvolution of thermal images is modeled as an optimization problem, whose cost function is to be minimized based on mechanics of natural selection and genetics. The proposed algorithm can be configured with all available a-priori information to speedup the solution computation. The paper presents the results for deconvolution using the proposed genetic algorithm and its utility in PCB infrared thermal testing.**

***Index Terms*—Blind deconvolution, image deconvolution, genetic algorithm, thermal image.**

I. Introduction

THERMAL images are very useful in the process of design and testing of Printed Circuit Boards (PCBs). Thermal image analysis relies on the power dissipation of each Integrated Circuit (IC) on the PCB. The energy dissipation associated with the passage of electrons through the junctions in a semiconductor device gives rise to thermal characteristic of each IC [1]. This thermal characteristic of electronic components is captured by an infrared camera and thus the thermal image is achieved.

There are three ways of heat dissipation: conduction, convection and radiation [1], [2], each having its own heat transfer set of equations. The overall dissipation of point-like heat source in homogeneous medium can be viewed as a Gaussian-like dispersion function [3]

$$g(x,y) = a \cdot e^{-\left(\frac{x^2}{w^2}+\frac{y^2}{h^2}\right)} + g_0 . \quad (1)$$

An ideal thermal image can be achieved, or a real thermal image can be approximated with convolution of two images f and g as shown in (2) [3]. The first image, f, which simulates the heat sources, contains spikes which are placed in the middle of each IC and have the amplitudes proportionally with the dissipated power

$$h(x,y) = f(x,y) \circ g(x,y) . \quad (2)$$

Thermal images are constrained by infrared camera position and sensitivity. Therefore, the observed image h' results from the convolution of the heat sources f and the heat dissipation function g and blurring function g', plus noise due to image acquisition process (3) [4]. Function g'' in (3) is $g''(x,y) = g(x,y) \circ g'(x,y)$. Function n is assumed to be additive noise randomly generated.

$$h'(x,y) = h(x,y) \circ g'(x,y) + n(x,y) = f(x,y) \circ g''(x,y) + n(x,y) \quad (3)$$

The convolution of two functions $f(x,y)$ and $g(x,y)$, denoted by $f(x,y) \circ g(x,y)$, is defined by

$$f(x,y) \circ g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta) \cdot g(x-\alpha, y-\beta) d\alpha d\beta . \quad (4)$$

We denote by deconvolution the process inverse to convolution. The thermal image deconvolution is to recover the original image h from the observed image h'. Also in the process of testing function f can be used as we'll see in this paper. If the heat dissipation function g and the blurring function g' are known, the function f can be easily computed using either deconvolution theorem or some linear deconvolution filters [5]-[7]. In reality the blurring function is not known therefore these methods cannot be applied in order to obtain the original image.

Blind deconvolution was first introduced to the imaging community by Ayers and Dainty [8]. Blind image deconvolution is the process of identifying both images f and g from the degraded of image h'. Using a-priori information about f and g, the speed of deconvolution process can be significantly increased.

The main objective of this article is to show the applicability of blind deconvolution to thermal images and how can be used in PCB thermal testing. There is no previous work in trying to apply blind deconvolution to thermal images of PCBs. The single article we found about thermal image reconstruction is [4]. It uses Wiener filter for image reconstruction due to infrared camera angle.

Deconvolution of thermal images is useful in eliminating camera constraints and reducing the thermal influences between integrated circuits on the PCB.

The basis of the genetic algorithm (GA) for thermal image blind deconvolution is given in the next section. Simulation results and real data results are presented in Sections III and IV. Conclusions and perspectives are presented in Section V.

II. The Genetic Algorithm

Genetic Algorithm (GA) is an iterative random search algorithm for nonlinear problem based on mechanics of natural selection and natural genetics [9], [10]. It uses probabilistic transition rules to guide the computation process toward the optimum solution. This kind of approach is particularly suited for the interpretation of poorly defined data [9]. Using a GA based algorithm, the thermal image deconvolution is modeled as an optimization problem, whose cost function is to be minimized. The typical flowchart of GA for blind deconvolution is shown in Fig. 1 [9].

Manuscript received December 12, 2003; revised June 18, 2004.

The authors are with the Computer Software and Engineering Department, "Politehnica" University of Timisoara, Romania (e-mail: mmarcu@cs.utt.ro, mvlad@cs.utt.ro).

Publisher Item Identifier S 1682-0053(04)0248

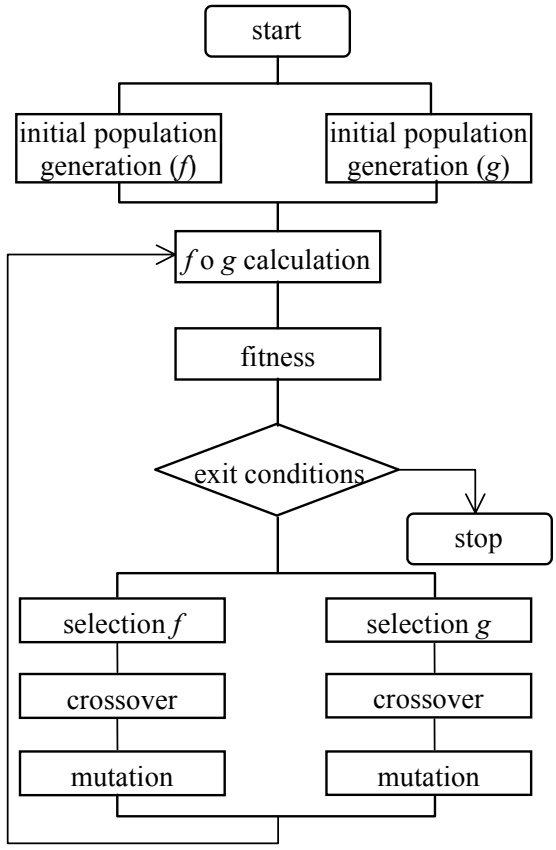


Fig. 1. GA for blind deconvolution [9].

A. Image Coding

Function f represents the original image with spikes placed in the middle of each integrated circuit (IC) on the PBC as shown in Fig. 2(a). The first a-priori information used by the algorithm is circuits' coordinates on the PCB. This algorithm can be configured to search also the spikes' locations, but with high increasing of processing time. The image f is encoded as a matrix $\mathbf{A}[m \times n]$ of real values, each element in the matrix containing the corresponding spike amplitude in the image. There is one element in the matrix $\mathbf{A}$ for each IC on the PCB, element which is proportional with power consumed by the IC.

Function g describes both the Gaussian-like heat dissipation function and process acquisition blurring function as depicted in Fig. 2(b). This function is encoded by w and h parameters in (1), considering $a = 1$ and $g_0 = 0$. Therefore image g will be encoded as a vector $\mathbf{G}[2]$ or for convenience as a matrix $\mathbf{G}[1 \times 2]$.

B. Initial Population

First of all the initial f and g populations are to be created. The encodings of these two functions (the chromosomes) are randomly created in the initialization step. The population size is another parameter of the deconvolution algorithm.

The second a-priori information available to use in the algorithm is about function g. Significant speedup can be achieved when w, h or w/h are known.

C. Fitness Measure

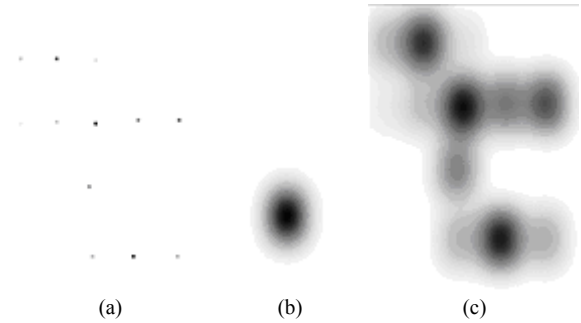
In the second step of the GA algorithm, convolution of each pair of chromosomes is calculated. The fitness of each pair of chromosomes is evaluated by comparing the calculated convolution image with the image h. The cost function of two chromosomes f_i and g_j is given by (5):

$$E_{i,j} = \left| g - f_i \circ g_j \right| \tag{5}$$

The optimum solution can be obtained by minimizing the costs of f and g. The costs of f_i and g_j are given by

$$\begin{aligned} Ef_i &= \min_{j=1,p}(E_{i,j}) \\ Eg_j &= \min_{i=1,p}(E_{i,j}) \end{aligned} \tag{6}$$

where p is the population size.



(a) (b) (c)

Fig. 2. Thermal image convolution sample (a) function f, (b) function g, and (c) function $h = f \circ g$.

$$\mathbf{A} = \begin{bmatrix} 34.8 & 110 & 56.5 \\ 0 & 47.9 & 35.1 \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} 12 \\ 15 \end{bmatrix}$$

Fig. 3. Chromosomal encoding samples.

D. Selection

In the selection process the best chromosomes are selected for the new generation of population. The others are randomly selected for the new population according with their cost functions. Some chromosomes with low cost functions will be selected more than once for the new population, therefore the best chromosomes get more copies, the average ones stay the same and the worst ones die.

The selection step of the GA algorithm was improved with a local iterative adaptive search for the best chromosomes. This local search algorithm applied to the best chromosomes also copies its neighbors to the new population.

E. Crossover

Crossover combines the values of two parent chromosomes to form two similar offspring by applying some swapping rules. The intention of crossover step of GA algorithm is information exchange between different potential solutions [9]. There are three types of crossover rules implemented by the algorithm, (see Fig. 4):

- Row switching - two selected parents exchange a randomly selected line (Fig. 4(a));
- Column switching - two selected parents exchange a randomly selected column (Fig. 4(b));
- Pivot or point switching - two selected parents randomly exchange the neighbors of a randomly selected point (Fig. 4(c)).

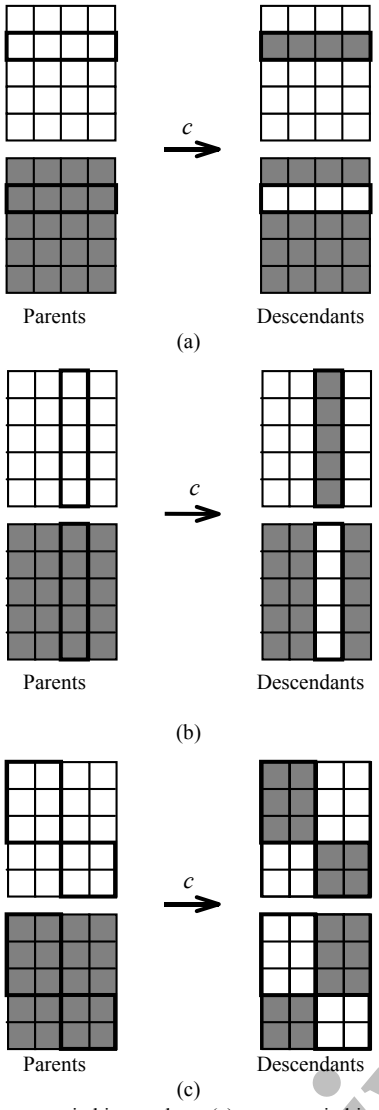


Fig. 4. Crossover switching rules, (a) row switching, (b) column switching, and (c) pivot switching.

In order to improve the search speed, the crossover step of the algorithm presented in [9] was improved with introduction of a switching function c. In an original GA algorithm c is the copy (identity) function, which randomly exchanges selected parts of two parents' chromosomes. Four functions were implemented in the algorithm: identity, average, minimum and maximum. The applied function is also randomly selected.

F. Mutation

The mutation process adds some extra variability into the new population. In order to add new type of solutions, a randomly selected chromosome item is increased or decreased by a random value.

G. Exit Conditions

The algorithm in Fig. 1 exits when a certain configurable error is reached or when the error is no longer changed in a certain number of generations or a certain programmable number of generations has elapsed.

III. Simulation Results

These kind of simulation tests have been carried out in order to validate this deconvolution algorithm: in the first test deconvolution was applied to the original distortion free image in order to show the convergence of the algorithm to the desired solution; in the second test the algorithm was applied to images of different sizes in order to emphasize its computing time; the last simulation test tries to show the algorithm recover capacity and how a-priory information influence the speed of calculations.

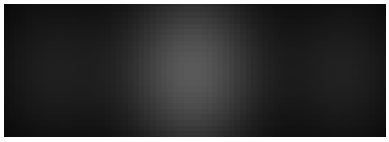


Fig. 5. Convolution image ($f \circ g$).

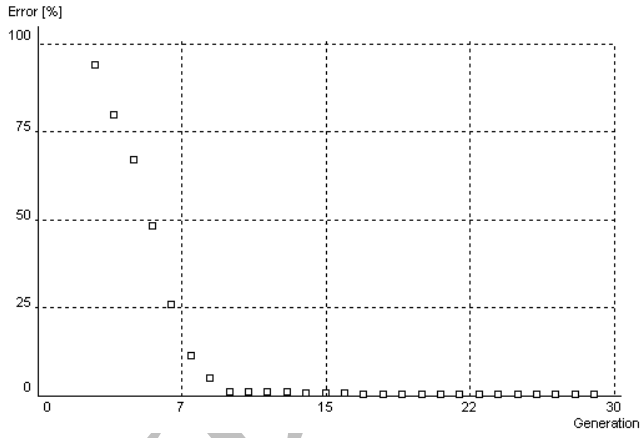


Fig. 6. Error vs. generation.

TABLE I
ALGORITHM EVOLUTION

Generation	A	Error [%]	Time [s]
1	51.28, 69.98, 43.21	1.02318	36
5	43.63, 99.42, 40.61	0.66938	188
10	31.28, 89.98, 33.21	0.01051	355
15	31.10, 89.98, 33.15	0.00663	535
20	30.10, 89.22, 32.58	0.00125	729
25	30.10, 89.22, 32.54	0.00109	895
30	30.10, 89.34, 32.58	0.00085	1000
40	30.10, 89.36, 32.60	0.00069	1304

A. Ideal Image Deconvolution Convergence

First of all, the deconvolution algorithm was applied on an image obtained by convolution of two images (f and g). Neither blurring nor noise was applied. The original image was obtained by convolution between images f and g (Fig. 5). Image f contains three spikes of heights randomly generated: 30.10, 89.38 and 32.64. Image g is a Gaussian-like dispersion function (1) with $a = 1$, $w = 12$, $h = 15$, and $g_0 = 0$.

The population size used in all tests was 100. Probability of mutation was 20% and all crossover modes (Fig. 4) had the same probability.

The evolution of the deconvolution algorithm for the original image in Fig. 5 is presented in Table I. The error variation with population generation is shown in Fig. 6. Error in Table I and Fig. 6 is computed using (6) and is defined as difference between original image and the computed one. Matrix **A** in Table I represents the codification of function f (Fig. 3). It can be seen that both functions f and g are improved as the generation increases and the error decreases below 1% in 15 generations.

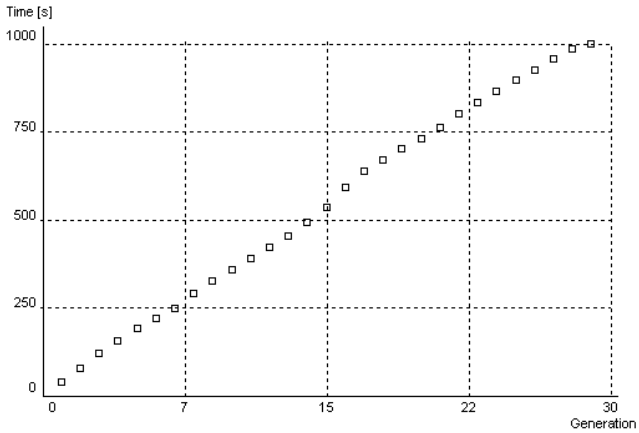


Fig. 7. Time vs. generation.

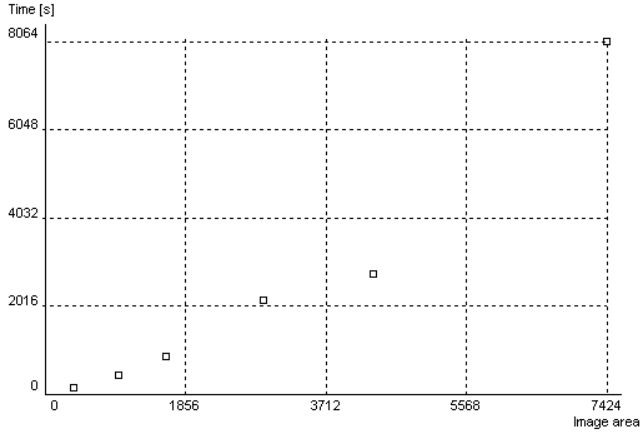


Fig. 8. Time vs. image size.

TABLE II
TIME VS. IMAGE SIZE

ICs	Generations	Time[min]	Error[%]	Image size
1	20	1:58	0.1613	16×24
2	20	6:43	0.1320	41×24
3	20	13:53	0.0667	67×27
4	20	45:45	0.2880	67×65
5	20	35:43	0.4061	116×25
8	20	2:14:24	0.4900	116×64

TABLE III
TIME AND ERROR VS. IMAGE QUALITY

Test description	Generations	Time [min]	Error [%]
Function g is known ($w=12$, $h=15$)	20	00:08	0.066
Function g is partially known ($h=5w/4$)	26	17:57	0.067
Convolution image	40	21:44	0.069
Convolution image with 5% noise	40	22:48	2.247
Convolution image with 10% noise	40	22:44	3.386
Convolution image with 10% noise and blur	40	22:46	4.842

On the other hand, time is the main drawback of this algorithm (Fig. 7). Each generation takes a constant amount of time. Usually the number of generations needed to achieve the goal is around 100, so the calculation time is around three hours for a 67×27 image.

B. Ideal Image Deconvolution Time

The second test tried to emphasis the relation between image size and deconvolution time and error. Different thermal images, each having its own size, were used in order to show how image size influence the deconvolution time. The deconvolution results are presented in Table II.

For each test case, the algorithm was let to evolve a constant number of generations. The evolution time and error are presented in Table II. Error in Table II is computed using (6) and is defined as difference between original image and the computed one.

As easily can be observed (Fig. 8), the evolution time increase with image size and also with the number of circuits composing the image.

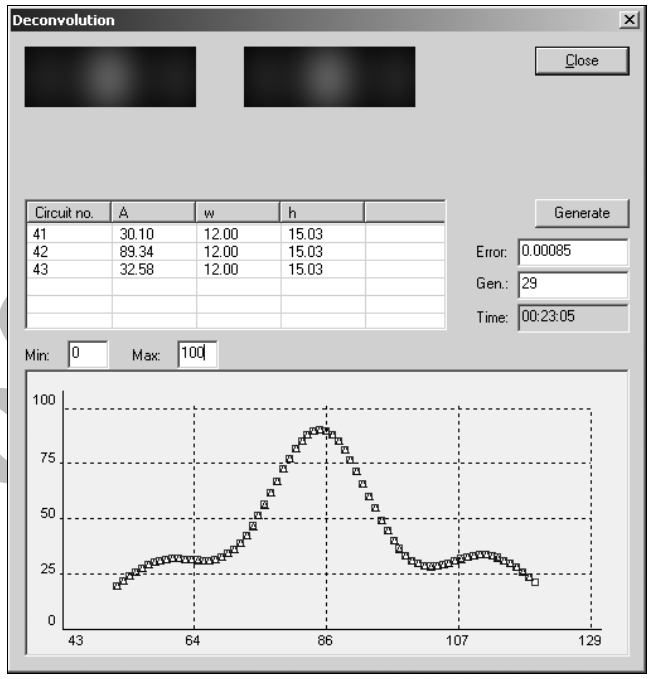


Fig. 9. Deconvolution application screenshot.

C. Ideal Image Deconvolution Time

Further, the third test was designed for image quality influence over the deconvolution algorithm. Different kinds of image distortions were applied in order to show how the algorithm is able to restore the image.

Deconvolution time is constant with image quality but error increase with the image degradation. A-priori information about original images can decrease dramatically the evolution time, but the accuracy of the results is not influenced very much. Error in Table III is computed using (6) and is defined as difference between original image and the computed one. We assumed that the noise is additive and randomly generated.

D. Testing Environment

The hardware configuration of the machine used for tests is: processor 1.20 GHz AMD Athlon, 256 MB memory, running Windows XP operating system.

A Windows application was implemented in order to test the deconvolution algorithm. The user can select an area in the original image for deconvolution. The dialog window in Fig. 9 is used to track the evolution of the deconvolution algorithm. The application was implemented in Microsoft

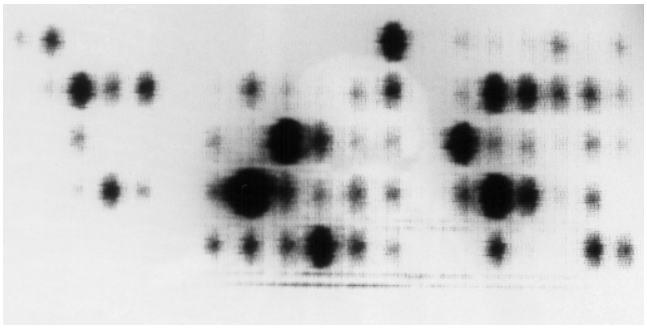


Fig. 10. PCB infrared image.

Fig. 11. Processed PCB infrared image.

Visual C++ 6.0 and contains around 50 classes and 16000 lines of code.

IV. Real Data Results

Image degradation is the main source of temperature inaccuracies in infrared testing. Temperature noise and spatial image degradation due to optical diffraction and lens influence; are the main ways of thermal image distortion [4].

The deconvolution algorithm was further used for real infrared PCB images. The image used in deconvolution is an infrared picture of a PCB with 66 ICs (Fig. 10). [2], [3], [11]. The images in Figs. 10 and 11 are printed with inversed colors. The processed image is printed in Fig. 11.

The image in Fig. 11 can be further used in the process of thermal testing. Table IV contains the values of matrix **A** (see Fig. 3) for the 12 ICs in the left of the image. These values are plotted related to measured ICs' current (Fig. 12).

Drawing the results in Table IV the plot in Fig. 12 is obtained. The error level of this solution is:

Average error: 5.72 %

Maximum error: 14.96 %

The other method used to recover the original image and to achieve a direct relation between image and power consumption of ICs was through image processing. We tried different filters in order to remove noise and enhance the image. The best relation we could obtain through image processing method was:

Average error: 10.10 %

Maximum error: 25.72 %

This algorithm can be further used in thermal testing of PCBs. In the process of testing, matrix A is obtained for the PCB under test and is compared with PCB defect free matrix. In case there are differences between the two matrixes the PCB is considered to be defect, also the defected circuit is detected.

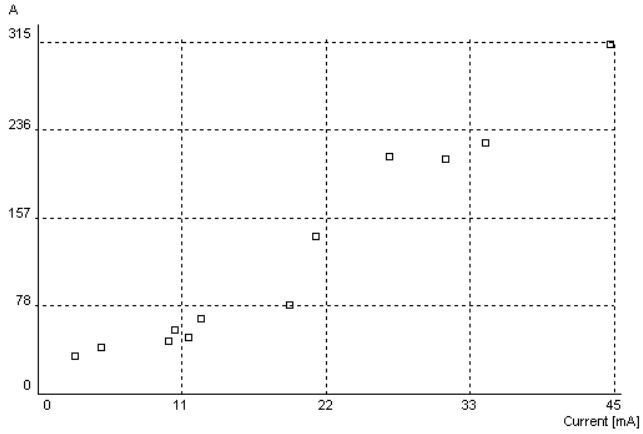


Fig. 12. Real infrared image deconvolution results.

TABLE IV
REAL INFRARED IMAGE DECONVOLUTION RESULTS

Circuit No.	Current [mA]	A
1	10.80	55.98
2	31.90	209.82
3	5.00	41.15
11	3.00	33.11
12	10.25	46.61
13	44.80	312.04
14	21.75	140.40
15	27.50	212.00
28	19.75	79.22
41	11.80	49.43
42	35.05	223.86
43	12.80	66.52

V. Conclusion

Blind deconvolution is not a new issue in the image processing community. It was used in image restoration since 1988, mainly in astronomy. This article introduced for the first time blind deconvolution for thermal images. Deconvolution of thermal images is useful in eliminating infrared image acquisition errors and reducing the influences between integrated circuits on the PCB.

Infrared PCB images deconvolution can be used in the process of PCB testing and power consumption estimation for ICs mounted on the PCB.

The main drawback of this approach is the time of solution achievement, therefore further work will try to reduce the time and cost of calculations.

References

[1] A. D. Krauss and A. Bar-Cohen, *Thermal Analysis and Control of Electronic Equipment*, McGraw-Hill, 1987.

[2] M. Vladutiu, H. Moldovan, and M. Marcu, "Current calculation from thermogram using a bi-directional model for heat transfer," *Trans. on Automatic Control and Computer Science,* vol. 41 (55), no. 1-2, pp. 122-133, Oct. 1996.

[3] M. Vladutiu, H. Moldovan, and M. Marcu, "Ideal thermal image generation," *Trans. on Automatic Control and Computer Science,* vol. 41 (55), no. 1-2, pp. 122-133, Oct. 1996.

[4] E. R. Meinders, G. M. P. van Kempen, L. J. van Vliet, and T. H. van der Meer, "Measurement and application of an infrared image restoration filter to improve the accuracy of surface temperature measurements of cubes," *Experiments in Fluids*, vol. 26, pp. 86-96, 1999.

[5] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison Wesley, 1987.

[6] A. Rosenfeld and A. C. Kak, *Digital Image Processing*, Academic Press, 1982.
[7] T. Holmes, *Background of Deconvolution*, Technical Note, AutoQuant Imaging, Inc. Feb. 2002.
[8] G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," *Optics Letter*, vol. 13, no. 3, pp. 547-549, Jul. 1988.
[9] Y. Chen, Z. Nakao, K. Arakaki, and S. Tamura, "Blind deconvolution based on genetic algorithms," *IEICE Transaction Fundamentals*, vol. 80-A, no. 12, pp. 2603-2607, Dec. 1997.
[10] B. Liang and S. U. Pillai, "Two-dimensional blind deconvolution using a robust GCD approach," in *Proc. Int. Conf. on Image Processing*, *ICIP '97*, vol. 1, pp. 424-427, 1997.
[11] M. Vladutiu, H. Moldovan and M. Marcu, "A new method of testing using infrared images," in *Proc. Fourth Int. Conf. on Technical Informatics, CONTI'2000*, pp. 78-83, Timisoara, Romania, Oct. 12-13, 2000.

Marius Marcu was born in Romania in 1972. He received the B.Sc. and M.Sc. degrees in Computer Science and Engineering from "Politehnica" University of Timisoara, Romania, in 1995 and 1996, respectively.

From 1995 to 1998 he was an Assistant Lecturer at Computer Science and Engineering Department of "Politehnica" University of Timisoara, and was promoted to Lecturer.

His research interests include microprocessors, thermal management, drivers development. He also works as software engineering consultant for Lasting Software, a company in Timisoara.

Mircea Vladutiu was graduated from Electronic Computers Department, Politehnica University of Timisoara, Romania, in 1967, in Electromechanical Engineering. He received his Ph.D. in Computer Reliability and Testing from the same university in 1982.

He has been a Professor at the Computer Engineering Department of the Politehnica University of Timisoara since 1990. Prof. Vladutiu is a Ph.D. advisor since 1990.

Prof. Vladutiu is involved in research and teaching activity at the Computer Engineering Department of the Politehnica University of Timisoara. He has a major contribution to the university curricula, where he introduced new courses: computer engineering, reliability of computer systems, computer systems testing, fault tolerant computing systems, and error control in computing systems. He is also responsible for computer architecture course.

He is the founder and director of the ACSA Lab, as well as of the Department of Lifelong Learning, inaugurated at the Politehnica University of Timisoara in 1998.

Prof. Vladutiu's main research interests are: computer architectures, computer arithmetics, computer reliability and testing, quantum computing, bio-inspired computing systems.

He is also topic administrator for CONTI conferences organized by the Politehnica University of Timisoara.