Modelling Blood Vessels of the Eye with Parametric L-Systems Using Evolutionary Algorithms

Gabriella Kókai[1], Zoltán Tóth[2] and Róbert Ványi[2]

[1] Department of Computer Science, Programming Languages
Friedrich-Alexander University of Erlangen-Nürnberg
Martensstr. 3. D-91058 Erlangen, Germany
e-mail: kokai@informatik.uni-erlangen.de

[2] Institute of Informatics, József Attila University
Árpád tér 2, H-6720 Szeged, Hungary
e-mail: h531714|h531774@stud.u-szeged.hu

Abstract. In this paper the *GREDEA* system is presented. The main idea behind it is that with the help of evolutionary algorithms a grammatical description of the blood circulation of the human retina can be inferred. The system uses parametric *Lindenmayer systems* as description language. It can be applied on patients with diabetes who need to be monitored over long periods.
Keywords: Computer vision, image and signal interpretation, evolutionary algorithms

1 Introduction

In this paper the *GREDEA* (*Grammatical Retina Description with Evolutionary Algorithms*) system is presented. The main idea behind *GREDEA* was to develop patient-specific monitoring programs for examining the blood circulation of the human retina. The system can be used for on patients with diabetes who need to be monitored over long periods. The regular checkup of patients with this disease is an essential task, because the change of the vision quality is a direct consequence of the deterioration of the vascular tree caused by diabetes.

In recent years some earlier systems ([2], [5] and [13]) have been developed but these systems employed computer graphics methods. They require comparison of various images spaced over time. To make a diagnoses based on these images is a difficult task and needs a lot of expertise. Another attempts to use a syntactic approach is the system developed by Hesse et al. [4]. Here fractal modelling was applied but the authors concentrated on simulating the typical growth pattern of the retinal artery and so this work has no real diagnostic benefit.

In *GREDEA* an individual description of the blood circulation of the human retina is created for each patient. To derive it the process starts from fundus images recognized by scanning laser ophthalmoscope (*SLO*) and preprocessed. These images are referred to later as *destination images*.

Then a parametric *Lindenmayer system* (*L-system*) is picked up which creates the pattern closest to the vascular tree of the patient. This *L-system* can be stored (needs less memory than storing a picture) and used later to make comparison.

The *L-systems* are parallel grammatical rewriting systems. They contain an initial axiom and one or more rules. With the help of such a grammar complex objects can be described starting from the axiom and using the productions [9]. When choosing the representation form, the fact that *L-systems* can be evolved using evolutionary algorithms [6] was taken into consideration.

The evolutionary algorithms can be briefly described as follows: a model of natural evolution [1] is often used as a way of solving optimization problems. These algorithms are based on the collective adaptation and learning ability of *individuals*. Individuals form a *population*, and each individual represents a possible solution of the problem. At the beginning, the individuals in the population are initialized randomly, and then they can be modified by the operators *selection*, *mutation* and *recombination*. These modifications lead to the evolution of the individuals; during the process better and better individuals appear. Thus, in later generations individuals that represent better solutions are more likely to appear.

Since parametric L-systems are evolved, two kinds of evolutionary algorithms are applied: genetic algorithms [3] are used on the rewriting rules of the evolved L-system, and evolution strategies [12] are applied on the parameters of the rules.

We proceed as follows. First in Section 2 and 3 a brief introduction to the recognition of the vascular tree in the retina and some details about the L-systems is given. The evolution process is presented in Section 4. In Section 5 an example for the experiments is discussed. Finally Section 6 contains a summary and some comments on future work.

2 The retina

The retina is the light sensitive surface at back of the eye [7]. The retina contains a vast number of specialized nerve cells called rods and cones that convert light rays into nerve signals. The cones provide color vision, and are most concentrated at the fovea. The fovea can be see in Figure 1 as the region on the side without any blood vessels crossing it. Because of their dense concentration, the cones in the fovea provide the finest resolution vision. The rods are much more sensitive to light than the cones but they only provide black and white vision. All of the rods and cones connect with nerves whose signals exit from the eye via the optic nerve. The optic nerve along with the retinal artery, joins the eye at the optic disk. The optic disk is visible in Figure 2 on the right side. There are no rods or cones in the optic disk which creates a small blind spot on the retina. The vessel system of the eyebackground consists of two types of blood vessels, arteries and veins. The part where the vessels (arteries and veins) enter the eye is called papillary. Arteries and veins may have some crosspoints in the background of the eye but arteries never cross arteries and veins never cross veins, The blood vessels can be described with a binary tree because from a knot point a vessel can branch off only in two directions. The vessels starting from the papillary surround the macula. The macula does not contain any vessels

When diabetes affects the eyes, it often affects the retina lining the back of the eye, specifically the small blood vessels (see in Figure 3) which nourish this membrane. This condition is called diabetic retinopathy. Diabetic retinopathy often has no symptoms in its early, most treatable stages. Improving awareness of the importance of early detection and treatment is imperative if one is to reduce visual loss from this disease. A minimal evaluation of the diabetic retina consists of regular examinations by a qualified practitioner though dilated pupils.

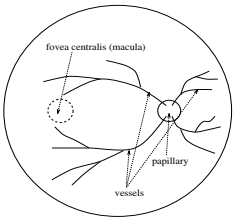

Fig. 1. The structure of the retina

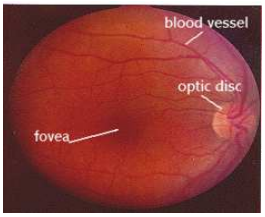

Fig. 2. Fundus image of the retina

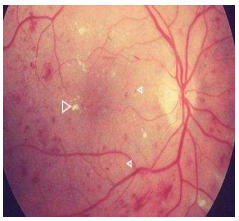

Fig. 3. Diabetic retinopathy

3 The Lindenmayer systems

Lindenmayer introduced special kinds of descriptive grammars in order to describe natural development processes [8]. These are called Lindenmayer grammars or *Lindenmayer system*s (*L-system*s). The most important characteristic of *L-system*s compared with Chomsky grammars is *parallel rewriting*. In Lindenmayer systems rewriting rules are applied in parallel and simultaneously replace all letters in a given word.

Different *L-system* classes are introduced, such as deterministic or nondeterministic *L-systems*. The distincion can be made between context free and context sensitive *L-systems*. The definition of these classes is similar to the definition of Chomsky language classes. In this paper only *deterministic context free L-systems*, (*D0L-systems*) are used. To enlarge the number of described objects *L-systems* can also be associated with parameters.

3.1 Parametric D0L-systems

With the idea of parametric *L-systems* the concept of parallel rewriting to parametric words instead of strings of symbols can be applied. Following the notation of Prusinkiewicz [10] the word "module" as a synonym for "letter with associated parameters" is used.

The alphabet is denoted by Σ, and the set of parameters is the set of real numbers $\boldsymbol{R}$. A module with letter $a \in \Sigma$ and parameters $p_1, p_2, \ldots, p_n \in \boldsymbol{R}$ is denoted by $a(p_1, p_2, \ldots, p_n)$.

The real-valued *actual* parameters appearing in words have a counterpart in the *formal* parameters which may occur in the specification of *L-system* productions. Let Π be the set of formal parameters. The combination of formal parameters and numeric

constants using the arithmetic operators $(+, -, *, /)$, the relational operators $(<, <=, >, >=, ==)$, the logical operators $(!, \&\&, ||)$ and parentheses $(())$ generates the set of all correctly constructed logical and arithmetic expressions with parameters from Π. There are noted $\mathcal{C}(\Pi)$ and $\mathcal{E}(\Pi)$.

Definition 1. A *parametric D0L-system* an ordered quadruple $G = (\Sigma, \Pi, \alpha, P)$, where

- $\Sigma = s_1, s_2, \ldots, s_n$ is the alphabet of the system,
- Π is the set of formal parameters,
- $\alpha \in (\Sigma \times \boldsymbol{R}^*)^+$ is a nonempty parametric word called the axiom,
- $P \subset (\Sigma \times \Pi^*) \times \mathcal{C}(\Pi) \times (\Sigma \times \mathcal{E}(\Pi)^*)^*$ is a finite set of productions.

A production in a *D0L-system* is given in the form $pred \mid cond \rightarrow succ$, where $pred \in \Sigma \times \Pi^*$ denotes the *predecessor*, $succ \in ((\Sigma \times \mathcal{E}(\Pi)^*)^*$ is the *successor* and $cond \in \mathcal{C}(\Pi)$ describes the *conditions*. A production *matches* a module if the following conditions are met:

- the letter in the module and the letter in the production predecessor are the same
- the number of actual parameters in the module is equal to the number of formal parameters in the production predecessor
- the condition evaluates to true if the actual parameter values are substituted for the formal parameters in the production.

If a production is matching a module then it can be *applied* in order to create a string of modules specified by the production successor. The actual parameter values are substituted for the formal parameters according to their position.

There is no essential difference between *D0L-systems* that operate on strings with or without brackets. In our implementation brackets are used: However, in this case productions involving brackets are restricted to the following forms:

- $pred \mid cond \rightarrow succ$, where $pred$ and $cond$ are as before, but $succ \in ((\Sigma \times \mathcal{E}(\Pi)^* \cup \{[,]\})^*$, and $succ$ is well nested;
- $[\rightarrow [$, or $] \rightarrow]$.

A string S is called well-nested if and only if for all S' substring of S the number of '[' symbols is not less than the number of ']' symbols, and the number of these symbols is equal in S. More details and examples of parametric *D0L-systems* can be found in [11].

3.2 Displaying images defined by L-systems

To display an image defined by an *L-system* the well-known turtle graphics is used. The turtle is an object in space such that its position is defined by a vector and its orientation is defined by a coordinate-system. The coordinate-system determines the heading (h), the left(l) and the up (u) directions. Using these vectors the turtle can be moved in space. There are several turtle operations, also known as turtle commands. The most important ones are drawing ('F'), and turning left and right ('+' and '−'). There are special turtle operations, called stack operations. These are denoted by '[' and ']', and used to save and restore turtle position and orientation. Using stack operations branching structures can be produced.

To use turtle graphics with *L-systems* turtle commands as the elements of the alphabet are applied.

4 The evolution process

If an image is given and one wants to find the grammar that describes it, this is called the inverse problem of L-systems. Evolutionary algorithms are used to solve this problem are descriebed here in detail

4.1 Initial population

The first step is define the structure of the initial population and the individuals themselves. For the first examination the initial population is randomly created, but if the patient returns for a check-up, the previously generated L-system is used to create the initial population. This makes the convergence speed greater and makes comparison easier.

4.1.1 The structure of the population

As it was mentioned the individuals are represented by *L-systems*. It is easier to handle *L-systems* if it is assumed that the axiom is constant and contains only a single symbol. Let the axiom be *S*. It can be proved that every *L-system* can be transformed into such form, without changing the described image. It is also assumed that the number of the rules is constant.

Because the shape of the generated image depends both on the rewriting rules and on the parameters, the representation can be divided into two important parts. The right sides of the rules are represented as strings while the parameters are described as a vector of real values. Usually it is more important to find the appropriate right sides. Changing the parameters is used for the fine-tuning of the image. *L-systems* with the same rules are considered as an *L-system group*. In an *L-system group* the rules are constant, but the parameters can be variated. To find the best parameters *evolution strategies* (*ES*) are used inside the *L-system group*. Therefore an *L-system* group works as a *subpopulation*. A subpopulation can be represented by rewriting rules. Because the number of the rules, and also the left sides are constant, an *L-system group* can be represented by an array of strings. To find the rewriting rules, genetic operators are applied on the string arrays representing an *L-system group*. So the *L-system groups* make up the population. The structure of the population can be seen in Figure 4.

To calculate the fitness value, the images described by the L-systems are created, then the distance between these generated images and the destination image is calculated with a suitable function. This function will be discussed in section 4.2. Using this fitness value *evolution strategies* are applied on the parameter vectors several times. The number of the evolution steps (denoted by S_{ES}) is defined at the very beginning of the process. When the ES steps are finished, the best individuals in each subpopulation are selected. The parameters and the fitness value of these *L-systems* will act as the parameters and the fitness value for the whole subpopulation. Using this fitness value, selection can be applied to the subpopulations to generate new subpopulations using *genetic algorithms* (*GA*). It is important that selection applied on subpopulation is a *GA-selection*, which is *fitness proportional*. Thus choosing the best individual in the subpopulation as the representing individual does not mean that the L-system group

with the best L-system will be selected. The detailed description of the applied ES and GA operators can be found later in section 4.3. Earlier we mentioned that ES is usually applied on real valued vectors, however GA is used with words over a finite alphabet. Therefore in this process both of them are used.

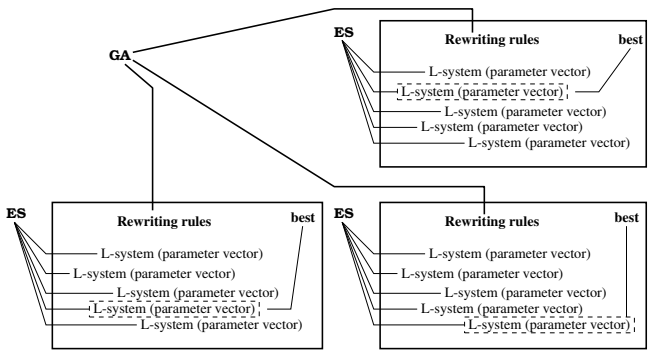

Fig. 4. The structure of the population

4.1.2 The form of an individual

The base *L-system* used as an individual is the following:

alphabet: $\Sigma = \{S, F, +, -, [,]\}$
axiom: $\alpha = S$
rewriting rule: $S \to u$, where $u \in \Sigma^*$ is a well-nested string.

Using this approach it can be noticed that this type of *L-system* sometimes generates *unnecessarily long word*s (*ul-words*). For example the image described by the word $F+--++F-+$ can be also generated by the word $F+F$ which is much shorter. To solve this problem, sequences of symbols can be defined, such that these sequences always start with a drawing symbol and do not contain unnecessary turtle movements (e.g. $+--+$). If *regulated evolution* is used to *preserve sequences* those ul-words can be avoided. During regulated evolution sequences cannot be broken, so genetic operators must be applied on substrings and not single symbols. But this makes the genetic algorithm more difficult. The other solution employs the specialties of the problem. Examining the blood vessel in the eye, a better *L-system* can be defined which itself includes the sequences:

$\Sigma = \{S, F, +, -, [,], B, Y, T\}$
$\alpha = S$
and rewriting rules:
$S \to v$, where $v \in \{S, B, Y, T, [,]\}^*$ is a well-nested string.
$B \to F + S$
$Y \to F[+S][-S]$
$T \to F[+S]S$

Note that the vascular tree can contain only the following parts: *bending* (B), *branching* (Y) and *forking* (T). Using this *L-system*, only the first rule will be modified during the evolution process. The initial F symbols of the other three rules ensure

that ul-words will not be created. So this *L-system* will produce better words, just like the *L-system* with sequences, without making the use of genetic operators difficult.

4.2 The fitness function: the distance of two images

There are many possibilities to define the distance between two images. A very simple method to calculate the distance d_q between two images I_1 and I_2 is the quadratic error of the images. During the definition we assume that the sizes of both pictures are the same: $N \times M$.
The distance is the sum of quadratic errors of two corresponding pixels:

$$d_q(I_1, I_2) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (I_1(x,y) - I_2(x,y))^2$$

For binary images, where $I_1^B(x,y)$ and $I_2^B(x,y)$ are two pixels the quadratic error is:

$$d_p(I_1^B(x,y), I_2^B(x,y)) = \begin{cases} 1 & \text{if } I_1^B(x,y) \neq I_2^B(x,y) \\ 0 & \text{otherwise} \end{cases}$$

So let the distance between two binary pictures be defined like this:

$$d_q(I_1, I_2) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} d_p(x,y)$$

In this paper an extended version of this definition is used because in our evolutionary process we wanted to make a distinction between the following four cases for an arbitrary (x,y) point:

- The point (x,y) is 0 on both images, it is not important for the fitness calculating.
- The point (x,y) is 1 on both images. In this case a match is found; this means positive fitness.
- The point (x,y) is 0 on the destination image, but 1 on the generated image. This means point that is not part of the vascular tree is covered.
- The point (x,y) is 1 on the destination image, but 0 on the generated image. So there is a missing point.

Let us denote the pixels with $p_d = I_d^B(x,y)$ and $p_g = I_g^B(x,y)$, where p_d means the pixel value on the destination image I_d and p_g means the value on the pixel in the generated image I_g. In this case the weighted distance of two pixels can be defined:

$$d'_p(x,y) = \begin{cases} 0 & \text{, if } p_d = p_g = 0 \\ W_1 & \text{, if } p_d = p_g = 1 \\ -W_2 & \text{, if } p_d < p_g \\ -W_3 & \text{, if } p_d > p_g \end{cases}$$

So the fitness function φ is the weighted distance between two images:

$$\varphi = d'(I_1, I_2) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} d'_p(x,y)$$

Using this definition a missing point will cost more than an unnecessary one if W_2 is set to greater than W_3. These parameters can also be used to fine-tune the fitness function.

4.3 The used operators

Since parametric L-systems are evolved, two kinds of evolutionary operators are applied; evolution strategies are applied on the parameters of the rules and genetic algorithms are used on the rewriting rules of the evolved L-system. In this section these are described.

4.3.1 The ES operators

Let us denote two parameter vectors with $u = u_1u_2 \dots u_n$ and $v = v_1v_2 \dots v_n$. During the evolution of parameter vectors two ES operators are used:

- *ES mutation.* If the u vector is mutated, the resulting vector will be u' where $u'_k = u_k$ if $1 \leq k < i$ or $i < k \leq n$ and $u'_i = u_i + \xi(R_{ES})$ for a randomly selected i. $\xi(R_{ES})$ represents the Gaussian derivation with deviation R_{ES}. This R_{ES} parameter is called the *radius* of the mutation, and is defined before the evolution process.
- *ES recombination.* If the u and v vectors are recombined, the resulting vector will be w, where $w_i = (u_i + v_i)/2$.

4.3.2 The GA operators

Let us denote two strings with $u = u_1u_2 \dots u_n$ and $v = v_1v_2 \dots v_m$. To evolve these strings two GA operators are used which are similar to the operators used for ES. These operators are:

- *GA mutation.* If the u vector is mutated, the resulting vector will be u' where $u'_k = u_k$ if $1 \leq k < i$ or $i < k \leq n$ and $u'_i = T$ for a randomly selected i position, and a randomly selected T character.
- *GA recombination.* If the u and v vectors are recombined, the resulting vector will be w, where $w_i = u_i$, if $i \leq k$ and $w_i = v_i$, if $k < i$ for a randomly selected k.

4.4 Halting criteria

Because our image is a finite set of discrete points there is an upper bound for the fitness value: $\varphi_{max} = (N \times M) \cdot W_1$. Therefore after a finite number of steps there will be a population such that its fitness is the same as of the previous one. The process can be stopped if the maximum fitness is reached, or nearly reached, or the fitness value was unchanged during the previous S_{GA} steps.

4.5 Choice of the evolution parameters

There are several parameters that can be changed before the process and also during the evolution. The *GREDEA* system can currently be used to find L-systems for simpler retina images. An example for it is presented in section 5. To find L-systems for more complex retina images, the starting parameters of the process must be set more carefully. In order to do, more retina images must be analyzed. This analysis is currently being carried out in our project.

- *General parameters.* The user can change the number of the subpopulation and the number of individuals in a population.
- *EA parameters.* To control the evolution strategy parameters are defined as previously mentioned: S_{ES}, W_1, W_2, W_3 and R_{ES}.
- *GA parameters.* The genetic algorithm has only one parameter: S_{GA}.

5 Evolution result

Here we present an example of finding a Lindenmayer-system that describes a given retina image. In Figure 5 an *SLO* recognized and preprocessed retina image can be seen with the structure created by the L-system described below. The preprocessing part is discussed in the literature (see for example [13]) therefore we do not go into detail. The L-system describes the structure of the top-right quarter of the retina image.

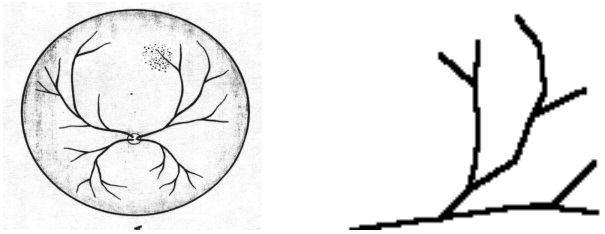

Fig. 5. An example for a retina image and an L-system that describes it

The rules of the above L-system are:

- S→;(2)B(37,0) [B(0,38) B(16,0) [B(0,27) B(11,14) B(12,6) B(19,0) [B(0,47) B(19,0)] B(0,-6) B(17,0)] B(0,-14) B(22,37) B(19,0) [B(0,-6) B(12,34) B(18,32) B(14,0)] B(0,-43) B(24,0)] B(37,0) [B(0,-7) B(8,43) B(24,0)] B(0,-14) B(26,0)
- B(a,b)→ F(a)+(b)

The turtle commands ($B[BB[BBBB[B]\ldots$) determine the main shape of the structure, but the segment lengths and angles depend on the parameters $(37,0,0,38,\ldots)$. (The ; (2) turtle command sets the initial line width.) Later this evolved description can be used to establish the change in a patient's eye: The patient comes again for a checkup and pictures are recorded again. From these *L-systems* are also generated and the previous grammars can be compared automatically with the newly evolved grammars. For example the shortening of the segments or the growing of the angles can be detected easily from the parameter vector. For the comparison different algorithms may be used but the detailed description of this process is beyond the subject of this paper.

To find the appropriate L-systems for more complex images needs more evolution steps. However more steps mean more computing. The evolution steps are time-consuming, because the operators work with parametrized strings and the fitnes function works with a matrix. For a complex image the resolution also has to be increased which means a greater matrix, so the computational time grows quadratically.

The convergence speed can be increased by choosing better starting parameters. They can be determined by statistical methods after analyzing more images. For the

time being we do not have the adequate number of pictures, and getting new medical images has legal connection also. Our experiments showed that using evolutionary algorithms, grammatical description of the retina can be determined, but perfecting the method needs more experiments. This is our current project.

6 Conclusion

The program *GREDEA* developed by us can be applied in diabetic retinopathy for diabetic retinal disease. It solves the problem of describing the blood vessels of the eye using parametric *Lindenmayer system*s (*L-system*s). To find the most suitable description evolutionary algorithms are applied.

A further possibility of the application is that if the picture generated from the *SLO* was imperfect the incomplete part of the vascular system can be reconstructed using the description grammar.

References

1. Darwin, C.: *On the Origin of Species* Murray, London, 1859.
2. C. Georgiadis, *Segmentation of the Vascular Tree in Retinal Images*, Department of Computer Science, University of Crete at Heraklion, Greece, September 1995.
3. Goldberg, D.E.: *The Genetic Algorithm Approach Why, How and What Next* In: Narenda 1986
4. Hesse, L., Chofflet, J. and Le Mer, Y.: *Simulating of the growth pattern of the central retinal artery using fractal modeling technique* In German Journal of Opthalmology 2:116-118 1993
5. John Hipwell: *Digital Angiography in Ophthalmology* Ph.D research at the Department of BioMedical Physics at Aberdeen University
6. Jacob, C.: *Genetic L-system Programming* PPSN III - Parallel Problem Solving from Nature, International Conference on Evolutionary Computation, Lecture Notes in Computer Science 866, Springer-Verlag, Berlin, 1994, pp. 334 - 343.
7. L'Esperance, F. A. Jr: *Ophthalmic Lasers: Photocoagulation, Photoradiation and Surgery.* 2nd ed. The C. V. Mosby Company St. Louis. 1983
8. Lindenmayer, A.: *Mathematical models for cellular interaction in development* In: Journal of Theoretical Biology 18: 280-315 1968.
9. Rozenberg. G., Salomaa A.:*The book of* $\mathcal{L}$ Springer Verlag 1985
10. Prusinkiewicz, P., Hammel, M., Hanan, J., Měch, R.: *Visual models of plant development* From G. Rozenberg and A. Salomaa, editors, handbook of formal languages Springer-Verlag, 1996.
11. Prusinkiewicz, P., Hanan. J.: *Lindenmayer systems, fractals and plants* Lecture Notes in Biomathemathics vol 79. Springer-Verlag, 1989.
12. Rechenberg, I.: *Evolutionsstrategie '94, Werkstatt Bionik und Evolutionstechnik* Band 1, Fromman-Holzboog, Stuttgart, 1994.
13. Zana, F.: *Registration of Retinal Images with Occlusion under a Weak Affine Hypothesis, Multimodal and/or Temporal Registration.* In the Proc. The 13th International Conference on Digital Signal Processing (DSP97), on the island of Santorini, Greece, July 2-4, 1997