WOODY PLANTS MODEL RECOGNITION BY DIFFERENTIAL EVOLUTION

Aleš Zamuda, Janez Brest, Borko Bošković, Viljem Žumer
Faculty of Electrical Engineering and Computer Science
University of Maribor, Slovenia
{ales.zamuda; janez.brest; borko.boskovic; viljem.zumer}@uni-mb.si

Abstract This paper presents an approach for recognition of procedural three-dimensional models of woody plants (trees). The used procedural tree model operates by building a three-dimensional structure of a tree by applying a fixed procedure on a given set of numerically-coded input parameters. The parameterized procedural model can later be used for computer animation. Recognition of a parameterized procedural model, from the photographic images, is done by differential evolution algorithm which evolves this model by fitting a set of its rendered images to a set of given photographic images. The comparison is done on a pixel level of the images through the integration of distances to the nearest similar pixels. The obtained results show that the presented approach is viable for modeling of woody plants for computer animation by evolution of the numerically-coded procedural model.

Keywords: Differential evolution, Numerical encoding, Procedural model, Self-adaptation, Structure recognition, Woody plant

1. Introduction

In this paper we present a new approach to design three-dimensional geometrical models for woody plants (trees). The geometrical models are expressed indirectly with the use of procedural models to reduce the enormous data storage space needed for their representation. The procedural models can also be easily animated and are suitable in computer graphics and animation. Our new approach to design of woody plant models is based on recognition of their procedural models [21], from images using evolutionary algorithms [22]. The paper [2] presents an approach for recognition of procedural models. However, the procedural models obtained in [2] were not as complex to express woody plants. Also the recognized procedural models were two-dimensional.

Therefore, we extend this approach to the domain of three-dimensional procedural models suitable to model woody plants.

In the next section, the related work is presented. In the Section 3, the proposed approach for procedural models recognition using differential evolution is described. In the Section 4, experimental results and their discussion is given, which show that the given approach is suitable for design of woody plant models. The Section 5 concludes with final remarks and propositions for future work.

2. Related Work

In this section, we present the differential evolution algorithm and one of its improvements, the jDE algorithm [4, 6]. Then, we list some of the procedural models for modeling of trees and outline the numerically-coded procedural model of the EcoMod framework [21, 23, 25].

2.1 Differential Evolution

Differential Evolution (DE) [18] is a floating-point encoding evolutionary algorithm for global optimization over continuous spaces, which can also work with discrete variables. Its main performance advantages over other evolutionary algorithms [4, 11] lie in floating-point encoding and a good combination of evolutionary operators, the mutation step size adaptation and elitistic selection. The DE algorithm has a main evolution loop in which a population of vectors is computed for each generation of the evolution loop. During one generation G, for each vector $\mathbf{x}_i$, $\forall i \in \{0, NP\}$ in the current population, DE employs evolutionary operators, namely mutation, crossover, and selection, to produce a trial vector (offspring) and to select one of the vectors with best fitness value. NP denotes population size and G the current generation step.

Mutation creates a mutant vector $\mathbf{v}_{i,G+1}$ for each corresponding population vector. One of the most popular DE mutation strategies is *'rand/1/bin'* [14, 18]:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}),$$

where the indexes r_1, r_2, and r_3 represent the random and mutually different integers generated within the range $\{1, NP\}$ and also different from index i. F is an amplification factor of the difference vector within the range $[0, 2]$, but usually less than 1. Vector at index r_1 is a base vector. The term $\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}$ denotes a difference vector which after multiplication with F, is named amplified difference vector.

After mutation the mutant vector $\mathbf{v}_{i,G+1}$ is taken into recombination process with the target vector $\mathbf{x}_{i,G}$ to create a trial vector $u_{i,j,G+1}$. The binary crossover operates as follows:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j,G} & \text{otherwise} \end{cases},$$

where $j \in \{1, D\}$ denotes the j-th search parameter of D-dimensional search space, $rand(0,1) \in [0,1]$ denotes a uniformly distributed random number, and j_{rand} denotes a uniform randomly chosen index of the search parameter, which is always exchanged to prevent cloning of target vectors. CR denotes the crossover rate.

Finally, the selection operator chooses one of the vectors with a better fitness value (for minimization problem):

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & \text{if } f(\mathbf{u}_{i,G+1}) < f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{otherwise} \end{cases}.$$

DE was proposed by Storn and Price [18] and since then, it has been modified and extended several times with new versions proposed [14, 9]. We have used the jDE algorithm [4], which adds to the original DE, a self-adaptation mechanism of F and CR control parameters. In this work, only the original jDE algorithm [4] was used, although the algorithm also has some extensions that have not been used in this work [5, 6, 7].

2.2 Woody Plants Procedural Models

The procedural modeling of trees has a thirty year tradition in computer graphics. Manual editing of a tree structure and its leaves is a tedious task, since each branch and leaf position, rotation, size, and texture must be appointed. Therefore, procedural tree models are used instead, and several techniques for procedural models are available today. Different procedural models are based on various types of branching structure construction [15]. These techniques differ in the level of detail [1, 3, 16], the flexibility, and pretentiousness of modeling [10, 19], space [13], and time complexity [13] in addition to the animation ability and representation of the built three-dimensional model. The majority of these models try to determine some visible properties of the final three-dimensional model, such as the rotation of branches around their central axes. These properties are usually biologically inspired by *phyllotaxis*, i.e. the main influence on the tree's architecture [17].

Holton [10] created trees with the use of biologically inspired strand model. An upside of this model is that, thickness of branches and proportions between branching angles are determined directly with internal rules in the model. Strands flow along branches and are divided without splitting a single strand. Branches with single strands are carrying leaves. Strand distribution determines branch thickness and their lengths. User enters the number of strands along the tree, proportions between branch lengths and branching angles to parametrize the procedural model. Certain attractors influence the branching structure, e.g. central trunk uprightness, gravimorphism, phototropism, planartropism, and phyllotaxis. A downside of the model is that user still has to enter a huge amount of numerical data, which diminishes the flexibility of the model.

Weber and Penn [19] represented the tree model with the use of simple geometry without a development of branching topology. For all branches at the same levels, they entered branching angle, branch length proportions and thickness for branches. They presented wind sway animation, branch cutting to predetermined volume, and progressive level of detail rendering.

The EcoMod framework incorporates a procedural model for woody plants, based on the Holton and Webber-Penn models. The procedural model and its modeler with woody plant models library was first introduced in [25] and is in greater detail described in [21, 23, 24]. The procedural model also helps to design the tree from a minimized set of parameters that the user must set by automatically determined positions, rotations, sizes and textures for several thousand branch segments and several thousand leaves. An individual tree species model is created by parametrizing the procedural model. It generates a three-dimensional structure [20] of a tree by recursively executing a fixed procedure over a given set of numerically coded input parameters, such as branch thickness, relative branch length and branching structure proportions. Each step of the procedure adds a building block of a tree to the geometrical model. The trees designed with this model can be foliage or coniferous trees with very different branching structures. Each branch and each leaf can be animated in real time to show the growth of a tree or its sway in the wind. By slightly modifying the parameters of procedural models, we can achieve computer animation of these models [24], thereby creating several geometrical models from a single procedural model.

Parameters of EcoMod woody plant procedural model are distinguished as vectors (local) and scalars (global). Global parameters are constant for all branch segments although local parameters vary along Gravelius (g) and Weibull (w) branch order. Vector parameters design the strand distribution, branching angles, branch segment proportions, and gravity impact to tree geometry. Scalar parameters of the model are height and thickness of base trunk, wind impact, and density and size of leaves. Using listed vector and scalar parameters, geometrical model is

Algorithm 1 Calculation of geometrical structure of the procedural model tree. Recursive procedure is called using **branchsegment**(0, 0, S, 1, $l_0^{0,0}$, $\mathbf{I}$, $\mathbf{I}$, $\mathbf{I}$), $\mathbf{I}$ denoting an identity matrix.

procedure branchsegment(g, w, S_0, L_0, l_0, $\mathbf{M}_0$, $\mathbf{M}_{m;0}^{-1}$, $\mathbf{M}_{w;0}^{-1}$)

Require: g, w - Gravelius and Weibull index of base branch; S_0 - number of strands in base branch; L_0, l_0 - base branch relative and actual length; $\mathbf{M}_0$ - base branch coordinate system; $\mathbf{M}_{m;0}^{-1}$ - inverse matrix of rotations for gravimorphism in coordinate system for base branch; $\mathbf{M}_{w;0}^{-1}$ - inverse matrix of rotations for directed wind in coordinate system for base branch; *global* k_d, k_c, l_{type}, $k_s^{g,w}$, $M^{g,w}$, $m^{g,w}$, $k_l^{g,w}$, $\alpha_m^{g,w}$, $\alpha^{g,w}$, t, k_f, w_s, w_g

Ensure: rendered tree image

$d := k_d\sqrt{S_0}$; {thickness calculation from Mandelbrot}

render base branch($\mathbf{M}_0$, l_0, d);

if $S_0 = 1$ **then**

  render leaves(l_{type}); **return**;

end if

$S_1 := \lceil 1 + k_s^{g,w}(S_0 - 2)\rceil$, $S_2 = S_0 - S_1$; {number of strands in major and minor subbranches}

$r_1 := \max\left\{\min\left\{\sqrt{\frac{S_1}{S_0}}, M^{g,w}\right\}, m^{g,w}\right\}$ {branch length proportions dependant on strands}

$r_2 := \max\left\{\min\left\{\sqrt{\frac{S_2}{S_0}}, M^{g,w}\right\}, m^{g,w}\right\}$;

$L_1 := r_1 L_0$, $L_2 := r_2 L_0$; {relative length of subbranches}

$l_1 := k_l^{g,w} L_1$, $l_2 := k_l^{g,w} L_2$; {active subbranch length}

$\alpha_1 := k_c\sqrt{\frac{S_2}{S_0}}\alpha^{g,w}$, $\alpha_2 := \alpha^{g,w} - \alpha_1$; {branching angles}

$\alpha_x(t) := \sin(t + R_x)w_s(1 - k_f)l_0$; {animation of un-directed wind impact}

$\alpha_z(t) := \sin(t + R_z)w_s(1 - k_f)l_0$;

$\alpha_w := \frac{S_0}{S}w_g$; {animation of directed wind impact}

$\mathbf{M}_1 := \mathbf{R}_{\mathbf{w}_0}(\alpha_w)\mathbf{R}_z(\alpha_1 + \alpha_z(t))\mathbf{R}_x(\alpha_x(t))\mathbf{R}_y(\alpha_p)\mathbf{R}_{\mathbf{y}\times\mathbf{y}_m}(\alpha_m^{g,w})\mathbf{T}_y(l_0)\mathbf{M}_0$;

$\mathbf{M}_2 := \mathbf{R}_{\mathbf{w}_0}(\alpha_w)\mathbf{R}_z(\alpha_2 + \alpha_z(t))\mathbf{R}_x(\alpha_x(t))\mathbf{R}_y(\alpha_p)\mathbf{R}_{\mathbf{y}\times\mathbf{y}_m}(\alpha_m^{g,w})\mathbf{T}_y(l_0)\mathbf{M}_0$;

$\mathbf{M}_{m;1}^{-1} := \mathbf{R}_{\mathbf{y}\times\mathbf{y}_m}(-\alpha_m^{g,w})\mathbf{R}_y(-\alpha_p)\mathbf{R}_x(-\alpha_x(t))\mathbf{R}_z(-\alpha_1 - \alpha_z(t))\mathbf{M}_{m;0}^{-1}$; {refreshing inverse matrix for construction of gravimorphism vector, without considering wind impact}

$\mathbf{M}_{m;2}^{-1} := \mathbf{R}_{\mathbf{y}\times\mathbf{y}_m}(-\alpha_m^{g,w})\mathbf{R}_y(-\alpha_p)\mathbf{R}_x(-\alpha_x(t))\mathbf{R}_z(-\alpha_2 - \alpha_z(t))\mathbf{M}_{m;0}^{-1}$;

$\mathbf{M}_{w;1}^{-1} := \mathbf{R}_{\mathbf{y}\times\mathbf{y}_m}(-\alpha_m^{g,w})\mathbf{R}_y(-\alpha_p)\mathbf{R}_x(-\alpha_x(t))\mathbf{R}_z(-\alpha_1 - \alpha_z(t))\mathbf{R}_{\mathbf{w}_0}(-\alpha_w)\mathbf{M}_{w;0}^{-1}$; {refreshing inverse matrix for construction of directed wind vector}

$\mathbf{M}_{w;2}^{-1} := \mathbf{R}_{\mathbf{y}\times\mathbf{y}_m}(-\alpha_m^{g,w})\mathbf{R}_y(-\alpha_p)\mathbf{R}_x(-\alpha_x(t))\mathbf{R}_z(-\alpha_2 - \alpha_z(t))\mathbf{R}_{\mathbf{w}_0}(-\alpha_w)\mathbf{M}_{w;0}^{-1}$;

branchsegment($g+1$, $w+1$, S_2, L_2, l_2, $\mathbf{M}_2$, $\mathbf{M}_{m;2}^{-1}$, $\mathbf{M}_{w;2}^{-1}$); {minor branch development}

branchsegment(g, $w+1$, S_1, L_1, l_1, $\mathbf{M}_1$, $\mathbf{M}_{m;1}^{-1}$, $\mathbf{M}_{w;1}^{-1}$); {major branch development} built recursively. From a procedural model for a tree, a geometry model is calculated using the briefly denoted Algorithm 1. Geometrical model is rendered using photo textures for final look of a tree. This model differs from many other models [1, 3, 10, 12, 16] since all of its parameters are fully numerically encoded and are fixed dimensionality. It is therefore especially suitable for parameter estimation using differential evolution.

2.3 Image-based Approaches to Modeling

Image-based approaches have the best potential to produce realistically looking plants, since they rely on images of real plants [8]. Also, little work has been done to design trees with the use of a general recognition from images without user interaction. In [2] an approach for recognition of procedural models is presented. However, the procedural models used in [2] were two-dimensional. Therefore, we extended their approach to the domain of three-dimensional procedural models suitable to model woody plants without user interaction.

3. Woody Plants Recognition by Differential Evolution

We have combined the jDE algorithm [4] and the numerically coded procedural model of woody plants from EcoMod framework [21, 23, 25]. Thereby, we recognize woody plant models from images by evolving the parameters of the procedural model. The fitness computation is based on the comparison of two-dimensionally rendered images. The fitness is better (i.e. takes smaller values) for images with greater similarity. The recognition method operates by encoding the parameters of the procedural model in genotype of the individual vector of jDE population. In the following, parts of the optimization procedure are described, i.e. the genotype encoding, genotype-phenotype mapping, and its fitness evaluation.

3.1 Genotype Encoding

An individual genotype vector $\mathbf{x}$ of jDE population represents the set of procedural model parameters, used in Algorithm 1. The dimensionality of evolved floating-point encoded parameters is $D = 4509$. Each parameter $x_{i,j} \in [0,1]$ for all $i \in \{1..NP\}$ and $j \in \{1..D\}$ encodes the following parameters (for more explicit formulation of the parameters see [21]):

- number of strands of a tree
  $S = 400x_{i,0} + 10$ ($S \in [10, 410]$),
- height of base trunk
  $l_0^{0,0} = x_{i,1} 10$ m ($l_0^{0,0} \in [0 \text{ m}, 10 \text{ m}]$),
- coefficient of branch thickness
  $k_d = 0.05x_{i,2}$ ($k_d \in [0, 0.05]$),
- phyllotaxis angle
  $\alpha_p = 360° x_{i,3}$ ($\alpha_p \in [0°, 360°]$),
- branching ratio of subbranch strands distribution
  $k_s^{g,w} = 0.5x_{i,j} + 0.5, \forall j \in \{4, 753\}$ ($k_s^{g,w} \in [0.5, 1]$),
- branching angle between dividing subbranches
  $\alpha^{g,w} = 180° x_{i,j}, \forall j \in \{754, 1503\}$ ($\alpha^{g,w} \in [0°, 180°]$),
- maximum relative subbranch to base branch length
  $M^{g,w} = 20x_{i,j}, \forall j \in \{1504, 2253\}$ ($M^{g,w} \in [0, 20]$),
- minimum relative subbranch to base branch length
  $m^{g,w} = 20x_{i,j}, \forall j \in \{2254, 3003\}$ ($m^{g,w} \in [0, 20]$),
- branch length scaling factor
  $k_l^{g,w} = 20x_{i,j}, \forall j \in \{3004, 3753\}$ ($k_l^{g,w} \in [0, 20]$),
- gravicentralism impact
  $k_c = x_{i,3754}$ ($k_c \in [0, 1]$),
- gravimorphism impact (i.e. gravitational bending of branches)
  $\alpha_m^{g,w} = 360° x_{i,j} - 180°, \forall j \in \{3755, 4504\}$ ($\alpha_m^{g,w} \in [-180°, 180°]$),
- enabling leaves display on a tree
  $B_l = \lfloor x_{i,4505} + 0.5 \rfloor$ ($B_l \in \{0, 1\}$),
- density of leaves
  $\rho_l = 30x_{i,4507}$ ($\rho_l \in \{0, 30\}$),
- size of leaves
  $l_l = 0.3x_{i,4506}$ ($l_l \in [0, 0.3]$), and
- leaf distribution type
  $l_{type} = 5x_{i,4508}$ ($l_{type} \in \{Spiral, Stacked, Staggered, Bunched, Coniferous\}$),

where $g \in \{0, 15\}$, $w \in \{0, 50\}$, and each 750 real-coded parameters encode one matrix of a Gravelius and Weibull ordered parameter.

3.2 Genotype-phenotype Mapping

Our recognition method is based on recognition of two-dimensional images of woody plants $\mathbf{z}^*$, possibly taken by a digital camera. To compare the three-dimensional tree evolved with the use of genotype $\mathbf{x}$ to the reference image $\mathbf{z}^*$, the encoded D-dimensional genotype $\mathbf{x}$ must be transformed to its phenotype first. Phenotype is a rendered two-dimensional image $\mathbf{z}$ of a genotype $\mathbf{x}$ with the use of Algorithm 1. Images $\mathbf{z}^*$ and $\mathbf{z}$ are all of dimensionality $X \times Y$ pixels, where the reference image is scaled to the given resolution, if necessary. Both images are converted to black and white, where white (0) pixels mark background and black (1) pixels mark the material, e.g. wood. With the use of the conversion, the evolved procedural model is compared twice to the reference images, differing by camera view angle of $\beta = 90°$ along the trunk base. The latter is done to favor three-dimensional procedural models generation. If we denote the Algorithm 1 as function $\mathbf{g}$ then $\mathbf{z} = \mathbf{g}(\mathbf{x}, \beta)$.

3.3 Phenotype and Reference Image Comparison

The recognition success is measured by similarity of the reference original images and the generated rendered images of evolved parametrized procedural models. To measure similarity of these images we chose to compare the images pixel-wise as follows. For each pixel rendered as non-background in the evolved image, we compute the Manhattan distance to the nearest non-background pixel in the reference image, and vice-versa [2]. The sum of these distances accounts for fitness evaluation of each phenotype:

$$f(\mathbf{x}) = f(\mathbf{g}(\mathbf{x}, 0°), \mathbf{g}(\mathbf{x}, 90°)) = h(\mathbf{z}_1) + h(\mathbf{z}_2),$$

$$h(\mathbf{z}) = \sum_{x,y} m_1(z_{x,y}, z^*_{x,y}) + \sum_{x,y} m_1(z^*_{x,y}, z_{x,y}),$$

where m_1 denotes a function which computes a Manhattan distance to the nearest pixel in an image $\mathbf{z}^*$, being set to 1 (i.e. black, wood material).

4. Experimental Results

We have assessed the algorithm for tree recognition on an example tree, seen in Fig. 1 on the far right. The sampling rate dimension of the rendered parametrized procedural model was set to 250x250, the maximal number of strands in the tree to $S = 410$, and the maximal number of fitness evaluations (FEs) for jDE algorithm to $FEs = 10000$. The remaining parameters were kept as defaults in original algorithms from their literature.

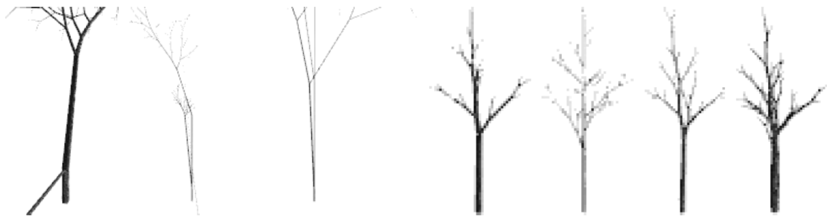

Figure 1. Rendered evolved parameterized procedural models at $FEs = \{1, 8, 18, 1992, 2727, 3230\}$ ($NP = 100$, seed 1) and fifth, the reference image.

Final best evaluations obtained over 30 runs for different settings of NP in the evolutionary algorithm are seen in Fig. 2. The best average final best was obtained using $NP = 100$ with fitness of 1828.3. For population size of $NP = 100$, the jDE algorithm in 30 runs obtained the best fitness value of 1806, the worst being 1870, and the average of 1828.3 with standard deviation of 84.4. The sampled procedural models for run 1, with seed 1, for this test are seen in Fig. 1. The tree on the image is 2.5 m tall, 1 m for the first branch segment, therefore it only extends to a part of the image's canvas which is 25 m tall in total. Therefore, the images in Fig. 1 were zoomed to fit. Since we can design woody plants with a reliability, seen in Fig. 2 and obtain such models as seen in Fig. 1, we can conclude that the presented approach is viable for modeling of woody plants for computer animation by evolution of the numerically-coded procedural model.

5. Conclusions

We presented an approach to design woody plant geometrical models. To construct a geometrical model, we have used a parameterized procedural model. The parameters of the procedural model were evolved through the jDE differential evolution algorithm. The sampled procedural models were rendered with the use of the EcoMod framework. Rendered images were then compared to the reference source images, for recognition, to guide the optimization process. After the description of proposed approach, we demonstrated its experimental results by recognition of a sample woody plant model and statistical analysis of the obtained results.

In the future research, we would like to improve metrics for comparison of rendered and reference images. Multiple metrics could be used

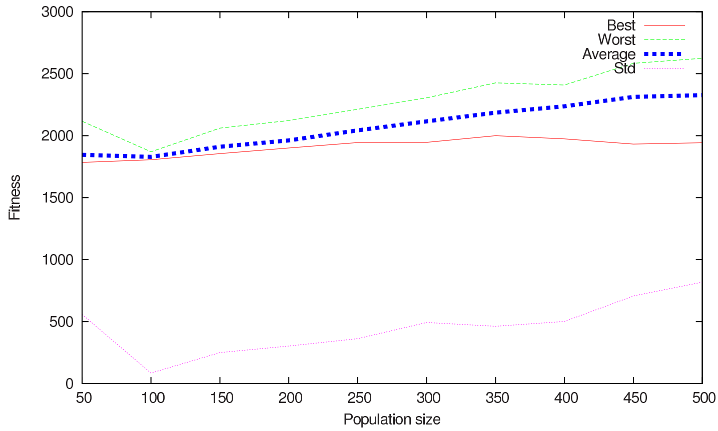

Figure 2. Algorithm performance, dependent on population size.

and combined with the use of multi-objective search [22], and possibly combined with interactive methods for optimization.

References

[1] M. Aono and T. Kunii. Botanical tree image generation. *IEEE Comput. Graph.*, 4(5):10–34, 1984.

[2] D. Beaumont and S. Stepney. Grammatical Evolution of L-systems. In *Proc. IEEE Congress on Evolutionary Computation*, pages 2446–2453, 2009.

[3] J. Bloomenthal. Modeling the mighty maple. In *Proc. 12st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 305–311, 1985.

[4] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE T. Evol. Comput.*, 10(6):646–657, 2006.

[5] J. Brest and M. Sepesy Maučec. Population size reduction for the differential evolution algorithm. *Appl. Intell.*, 29(3):228–247, 2008.

[6] J. Brest, A. Zamuda, B. Bošković, M. S. Maučec, and V. Žumer. Dynamic optimization using self-adaptive differential evolution. In *Proc. IEEE Congress on Evolutionary Computation*, pages 415–422, 2009.

[7] J. Brest, A. Zamuda, B. Bošković, and V. Žumer. An analysis of the control parameters adaptation in DE. In *Advances in Differential Evolution, Studies in Computational Intelligence*, 143:89–110, 2008.

[8] X. Chen, B. Neubert, Y. Q. Xu, O. Deussen, and S. B. Chang. Sketch-based Tree Modeling Using Markov Random Field. ACM T. Graphic., 27(5):109, 2008.

[9] V. Feoktistov. *Differential Evolution: In Search of Solutions*. Springer, 2006.

[10] M. Holton. Strands, gravity, and botanical tree imagery. *Comput. Graph. Forum*, 13(1):57–67, 1994.

[11] E. Mezura-Montes and B. C. Lopez-Ramirez. Comparing bio-inspired algorithms in constrained optimization problems. In *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pages 662–669, 2007.

[12] B. Neubert, T. Franken, and O. Deussen. Approximate image-based tree-modeling using particle flows. *ACM T. Graphic.*, 26(3):88, 2007.

[13] P. E. Oppenheimer. Real time design and animation of fractal plants and trees. *Comp. Graph.*, 20(4):55–64, 1986.

[14] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer, 2005.

[15] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, 1990.

[16] W. Reeves. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proc. 12st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 313–322, 1985.

[17] D. Reinhardt, E.-R. Pesce, P. Stieger, T. Mandel, K. Baltensperger, M. Bennett, J. Traas, J. Friml, and C. Kuhlemeier. Regulation of phyllotaxis by polar auxin transport. *Nature*, 426:255–260, 2003.

[18] R. Storn and K. Price. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11:341–359, 1997.

[19] J. Weber and J. Penn. Creation and rendering of realistic trees. In *Proc. 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 119–128, 1995.

[20] O. Yogev, A. A. Shapiro, and E. K. Antonsson. Computational evolutionary embryogeny. *IEEE T. Evol. Comput.*, 14(2):301–325, 2010.

[21] A. Zamuda. Modeling, Simulation and Rendering of Tree Ecosystems. Diploma thesis, Faculty of Electrical Engineering and Computer Science, University of Maribor, 2006.

[22] A. Zamuda. Self-adaptation of Control Parameters in Differential Evolution for Multiobjective Optimization Algorithm. MSc Thesis, Faculty of Electrical Engineering and Computer Science, University of Maribor, 2008.

[23] A. Zamuda, J. Brest, N. Guid, and V. Žumer. Construction of virtual trees within ecosystems with EcoMod tool. In *Proc. International Conference on Advances in the Internet, Processing, Systems, and Interdisciplinary Research (IPSI)*, 2006.

[24] A. Zamuda, J. Brest, N. Guid, and V. Žumer. Modelling, simulation, and visualization of forest ecosystems. In *Proc. IEEE Region 8 EUROCON 2007: International Conference on "Computer as a Tool"*, pages 2600–2606, 2007.

[25] A. Zamuda and D. Strnad. Interactive tool for modelling realistic animated trees. In *Proc. Thirteenth International Electrotechnical and Computer Science Conference (ERK)*, pages 11–14, 2004.