Interactive procedural forest in game engine environment as background for forest modelling

Marek Fabrika
Technical University in Zvolen

Abstract

The paper answer following questions: Why is the forest visualisation important in forest modelling? What kind of forest visualization is convenient? What functionality of forest visualisation is required? The contribution presents real application developed in Unity3D Game Engine Environment and integration with process-based modelling approache. The concept of a flexible simulation plot, relational database model, procedural tree trunk, procedural tree crown, interactive trees and an interface for process-based prognosis are presented. Next challenges are discussed.

Introduction

Forest visualization is an essential part of modern tools for predicting forest development (Fabrika and Pretzsch 2013). A forest is a complex system whose condition is defined by its species, diameter, height as well as spatial structure. When assessing the condition of the forest, not only production but also ecological and economic characteristics are taken into account, which often result in a more comprehensive assessment of ecosystem services. Forest visualization can replace numerical, tabular and graphical data so that the data is perceived very similarly, regardless of the type of evaluator, i.e whether it is a layman or a professional. The visualization of the forest that would meet such attributes should be three-dimensional. It should be real time. It should allow movement in the visualized forest. Ideally, it should allow interactivity, which means the ability to obtain information about the selected tree, or the ability to mark the trees. It should also be dynamic, meaning that it should support effects such as the movement of branches and assimilation organs under the influence of air flow or the casting of shadows or reflections. It should therefore support a high degree of immersiveness. Such a visualization of the forest can be achieved today using virtual reality built in the environment of game engines.

The Technical University in Zvolen currently has a tool for forest visualization, which is part of the SIBYLA software (SIBYLA Triquetra 2021). The virtual forest can be viewed on a PC, in an OCULUS Rift helmet or on a CAVE device built in a special hall (CAVE 2021). Unfortunately, the visualization of the virtual forest is still schematic. Tree trunks are composed of simple objects. The tree crowns are textured to vertical planes and do not contain separate objects of branches and assimilation organs. Visualization does not include the movement of objects in the wind, nor the penetration of light through the tree crowns, light reflections and shadows. The virtual forest was built on the basis of the VRML97 concept, which is already obsolete and does not provide all the features needed for an immersive experience. Today, there are possibilities for a photorealistic visualization of the forest. The proof is current computer games. However, the scenes of the forest landscape are prepared in advance in the form of pre-built game levels. The concept of generating a forest landscape in a real-time game engine environment based on external data on the terrain and forest cover, for example from a database, is not yet commonly available today.

Concept, requirements, and selection of the game engine

The concept of forest visualization should be based on an immersive virtual reality using a suitable game engine. The functional requirements should be as follows:

a) In the environment of a virtual forest, there should be unrestricted movement in the form of walking or running and flying. The best from the point of the first-person view, the so-called FPS mode.

b) Forest visualization should be connected to a relational database that stores information about the terrain and individual trees.

c) The terrain should be procedural. This means that it should be generated automatically based on the data structured in the database.

d) Trees should be procedural. This means that they should be created based on data stored in the database, such as tree species, position on the plot, trunk and crown dimensions, or tree quality.

e) Trees should be botanically correct and sufficiently detailed, for example with crown branching and visualization of individual leaves.

f) In addition to trees, the forest should contain other objects that are part of it: stones, litter, grass surface, microflora, shrubs, dead wood, stumps etc.

g) Visualization should support basic physical properties, such as the movement of branches and assimilation organs under the influence of air and wind movement, casting shadows based on the position of the Sun in the sky and clouds, reflections, and so on.

h) Visualization should support interactivity. The basis is to obtain information about the selected tree, marking the tree (crop tree or tree for cutting), cutting the tree, or its return, saving and loading the marked trees, etc.

i) Visualization should include an interface for the model to predict forest development, ideally on process-based principles, so that it is possible to predict forest development.

j) The visualization should include an interface for quantifying the ecosystem services based on forest structure and other data stored in a structured database, which may also be subject to forest prediction tools.

k) In a virtual forest, the state of which is predicted for the future, a so-called teleport should be allowed, i.e a shift in time.

l) Due to the recommended process-based principles for predicting the forest development, where a change in forest structure can be predicted at short intervals, such as hours, weather visualization should also be supported: sun movement across the sky, clouds, wind, precipitation, etc.

m) Additional tools such as web links or opening panoramic images at selected points in the visualized forest are also suitable.

Due to the possibility of progressive development, it is wise to use a game engine that is freely available. Today it is mainly Unreal Engine and Unity 3D. The Unreal Engine is more complex in terms of integrated tools and has more flexible options for creating virtual reality. However, it has a more complicated user interface. In terms of the number of supporting assets, it is comparable to Unity 3D. It also has comparable community support. However, there are fewer tutorials and examples, and it supports fewer platforms for final products. If the Unreal engine uses the C++ language, the Unity 3D is the basic C# language. Due to the above features, we chose the Unity 3D game engine for the development of the virtual reality of the forest.

Development methods

a) The concept of a flexible simulation plot

Existing forest development simulation tools are often limited to a representative simulation plot, such as 50 x 50 m or 100 x 100 m (Nagel 1996, Pretzsch et al. 2002, Fabrika 2005). With such selected plots, it is necessary to solve the problem with the edge effect in modeling the competition of trees and there are also complications in the need to visualize the whole forest landscape, respectively in modeling of regeneration procedures on larger areas. From this point of view, we chose the concept of a flexible simulation surface (Fig. 1). It is a regular square network that divides the forest area into equally large parts. Regular square plots are marked with the PLOT identifier. However, these plots contain trees, the affiliation of which is registered not only with respect to the square network, but also with respect to the forest stand. Forest stands are marked with the STAND identifier. Thus, the boundaries of forest stands can pass through individual simulation plots, and each tree carries information about the affiliation to the plot as well as to the forest stand. The simulation of forest development in each simulation plot is then done by considering the condition of trees not only in the plot but also in the surrounding neighboring plots (8 plots). This minimizes the impact of the edge effect on competitive relations and allows the visualization of the entire forest landscape as well as considering the morphology of the forest landscape terrain or simulating regeneration procedures in the forest complex. This system is also backward compatible with classical procedures, it means a representative simulation plot. In this case, one plot coincides with one forest that the plot represents.

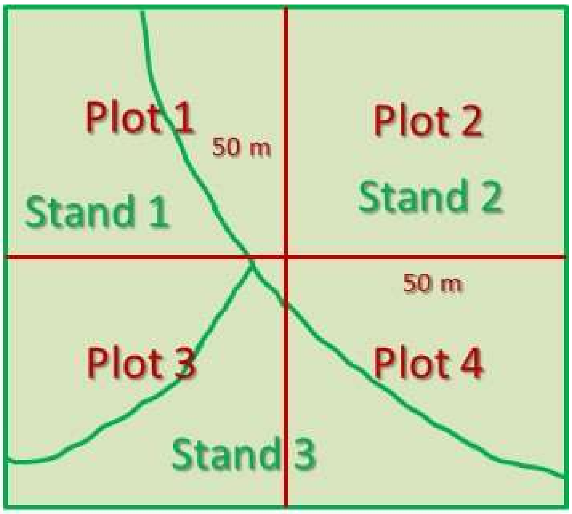

Fig. 1: The concept of a flexible simulation plot

b) Relational database model

The relevant relational database model is also connected to the concept of a flexible simulation plot (Fig. 2). The basis is the PLOTS table, which can be connected in a 1: 1 relationship to the TERRAIN table (one plot can have only one terrain defined) and is always connected to the TREES table in a 1: n relationship (there can be several trees on one plot). TERRAIN describes the character of the terrain detailed by the coordinates of the plate model in the SURFACE table (relation 1: n). TREES describes individual trees, and the MARKS table in a 1: 1 relationship can contain information about the marking of the tree (target or for harvesting). In addition, there is the STANDS table, which contains a list and characteristics of forest stands. The tables CLIMATE, SOIL and WEATHER are connected to the table STANDS in the relation 1: n. The given tables describe a series of climatic characteristics, soil characteristics and weather characteristics over time (hence 1: n). Each tree in the TREES table then has an identifier for the corresponding forest in the STANDS table. The overall characteristics of the quantification of ecosystem services are then summarized for individual stands within the plots (OUTPUT_PLOT_STAND) and within them also for individual trees in the forest stands on the given plots (OUTPUT_PLOT_SPECIES). In this way, it is then possible to aggregate total data for entire forest stands. The simulation and thus also the visualization of the forest can then be done according to individual plots or even for the whole forest landscape. Due to the possibility of re-generating the forest structure and repeating the prediction, data for the identified structures and prognosis are stored. Information about the change of data over time using time identifiers (year, day, hour) is also stored.

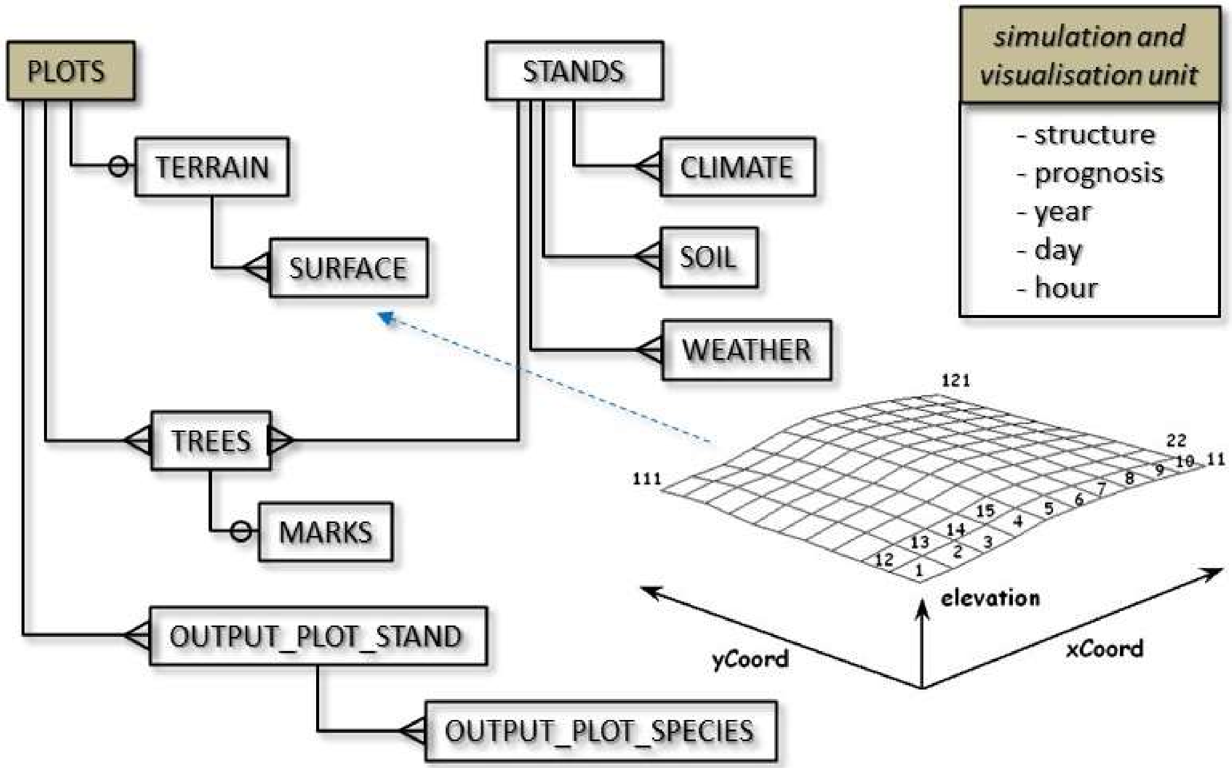

Fig. 2: Relational database model

c) Procedural tree trunk

Trunk models of individual tree species were created for forest visualization. The construction principle of the model is shown in Fig.3. The models were created using the principles of close-range photogrammetry. Representative trees were selected for selected species. The trees had a botanically typical appearance. They were in flat terrain (due to the need for a symmetrical trunk foot). There were no obstacles that would prevent the view of the trunk from every direction. Trees with a continuous trunk with a diameter of at least 30 cm and a trunk height of at least 8 m were selected. When taking images, we changed the horizontal direction around the trunk along the slide circle and changed the vertical direction from the base of the tree to the beginning of the tree crown. The images were taken so that they had sufficient overlap (min. 30%). An average of 150 to 300 images were taken for each selected tree. A point cloud was created from the images thus taken in the environment of Agisoft Photoscan software, which was subsequently replaced by vector polygons (meshes). Triple level of detail (LOD) models have been created: LOD0 with about 5000 triangles and texture with a resolution of 4096 pixels, LOD1 with about 2500 triangles and texture with a resolution of 4096 pixels, and LOD2 with about 1000 triangles and texture with a resolution of 2048 pixels. The models were created for a living trunk as well as for two types of dead trunks (freshly dead trunk - up to approx. 5 years and rotten - over 5 years). Stump models were also made. Models were created in OBJ format. An initial reference point with coordinates (0, 0, 0) was chosen at the trunk foot. The point is used to place the tree in the terrain of the forest stand. An ANCHOR node in the form of a sphere was placed at a height of 1.3 m, the diameter of which coincides with the diameter of the tree at this height. At the thinner end of the trunk, another KNOT node was placed, which serves as a reference point for connecting the tree crown to the tree trunk. The scaling of the trunk to the real diameter of the tree and the length of the trunk after the crown is placed is ensured by the created algorithm, which considers the size and position of the nodes as well as the algorithms of taper curves of the stem according to Petráš (1989, 1990). The taper curves were chosen with respect to the absolute height of the ANCHOR node location (1.3 m), which changes with the scale in the vertical direction. The tree trunks are then randomly rotated in the vertical direction. This rotation and resizing of the trunks create the impression of trunk diversity. Thus, 24 prototypes of tree trunks were created: spruce, fir, pine, larch, Douglass, beech, oak, hornbeam, maple (1), maple (2), sycamore, ash, elm, lime, locust, birch, poplar, aspen, willow, alder, cherry, rowan, walnut, and plane.

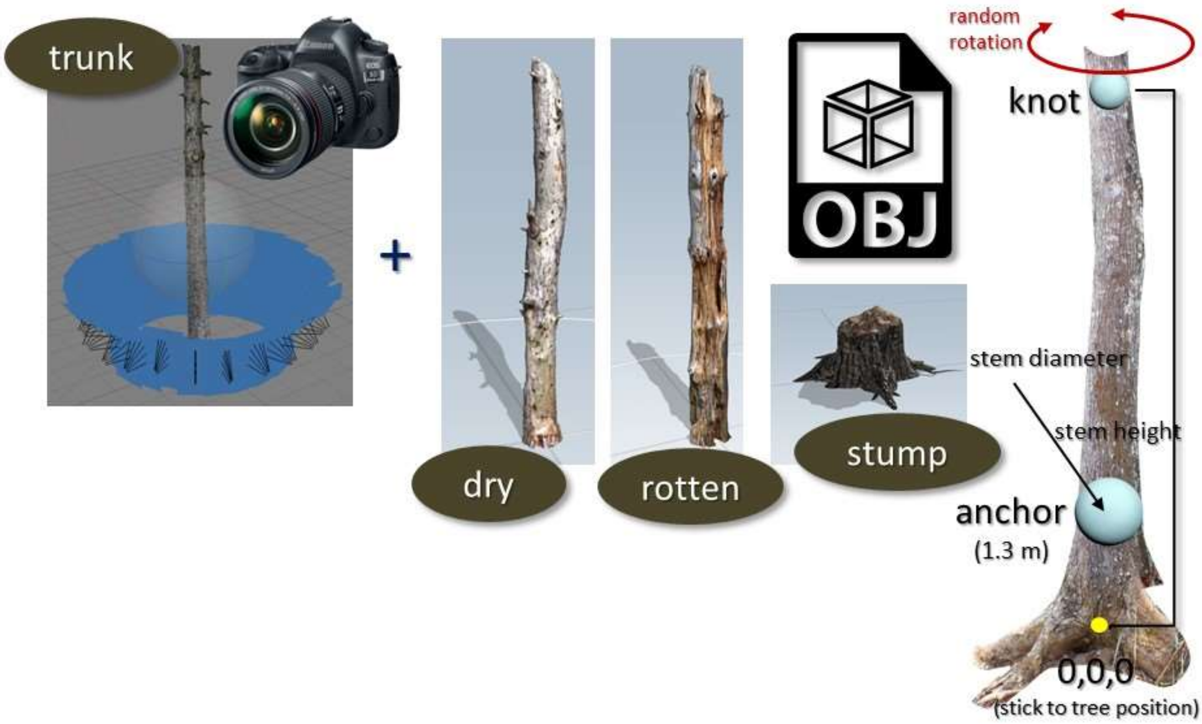

Fig. 3: The construction principle of a tree trunk procedural model

d) Procedural tree crown

Tree crown models were created in the SpeedTree Modeller environment. Models offered within commercially available libraries of tree species were used. The crowns were adjusted in the environment of the instrument so that the crown of the tree covered the entire axis of the trunk. The parameters of the branching and dimensions of the crown objects were modified as needed in the environment of the software tool. Branches and assimilation organs were replaced, if necessary, by photographed branches of collected empirical material. The principle of textures with a transparent background was used. Image keying methods were used to filter the background. The tree crown model was supplemented by the AMBIT node, which is located at the beginning of the crown axis with reference coordinates (0, 0, 0). The node has a spherical shape, and its diameter is identical to the widest diameter of the crown model. It is used to connect the tree crown to the KNOT node on the trunk model and to scale the crown in the horizontal direction. The APEX node is located at the top of the tree crown, which is used to scale the tree crown in the vertical direction based on its distance from the AMBIT node. The crown model is randomly rotated in the direction of the vertical axis when placed on the tree trunk. Amount of crown prototypes is identical to number of trunk prototypes (24). The principle of construction of the procedural crown is shown in Figure 4.

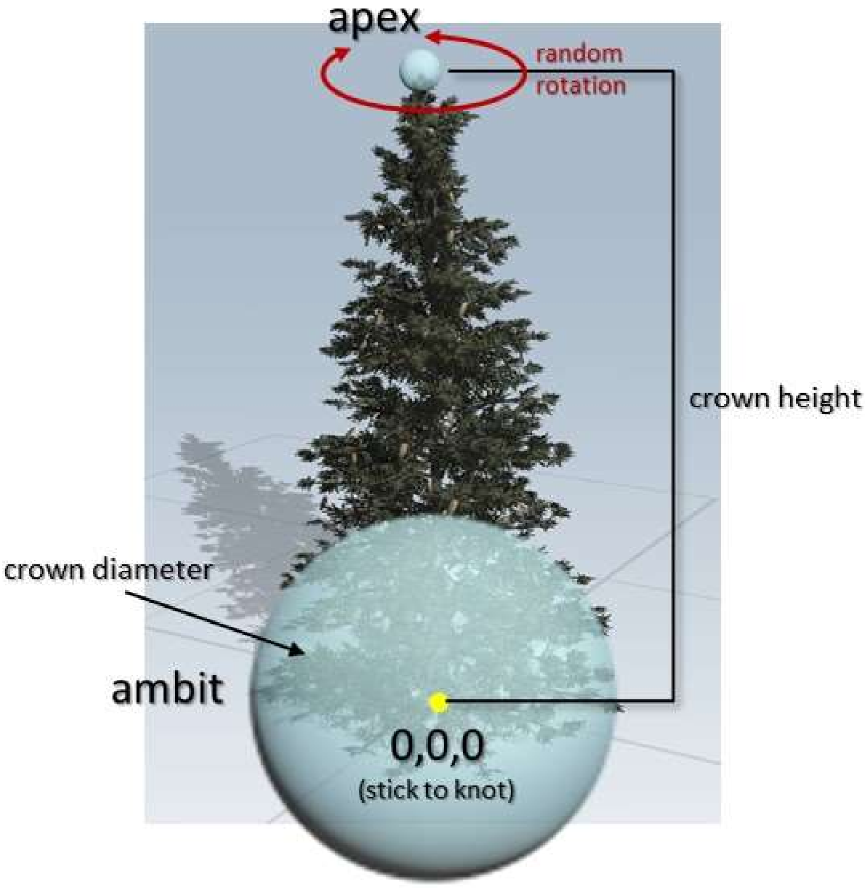

Fig. 4: The construction principle of a tree crown procedural model

e) Interactive trees

Tree models are created in the form of prefabricated parts, which are associated with the BIOMETRY script. The scripts encapsulate the tree parameters (identifier, year, stand, tree species, age, diameter, height, crown diameter, crown placement height, tree coordinates on the plot, tree quality parameters, etc.). One of the encapsulated methods is to display tree information in the info window. The second method is the method for placing the tree on the plot as well as the method for updating the tree parameters during the time frame change of the visualization. Another method encapsulated in the BIOMETRY object is to change the tree marking (red circle = tree intended for cutting, green circle = crop tree), representation of the tree growth area (white circle in the trunk axis with the width of the tree crown), tree cutting and undo of this action. Examples of interaction are shown in Figure 5. The methods are linked to an event that expresses the location of the cursor on a tree object. The event is solved by tracing the beam from the eye height of the FPS object (player or user) towards the front.

Fig. 5: Examples of forest visualization and interactions with trees

f) Interface to process-based prognosis

Forest visualization is connected via a defined database structure and configuration file to external software modules that enable forest development prognosis. The structure of the modules is adapted to process-based models in a standardized form. However, the defined interface allows you to connect other types of models. The connection of software modules uses so-called slots. The slots are marked with the module serial number and a letter that expresses the type of information associated with the slot: A = slot name, B = module type (explicit / implicit), C = tool name, D = disk path to the module. There are 14 slots. Table 1 lists the standardized modules.

Tab. 1: List of standardized modules for forest development prediction connected to forest visualization via defined slots.

slot number	slot name	tool name	tool description
1	ATLANTIS - **A**utomated **T**ree **L**ocalization **A**nd **N**ecessary **T**ree **I**nformation **S**etup	structure generator	generating the initial forest structure from various data sources
2	TARTARUS - **Tar**get **T**rees **Ar**bitrary for **Us**age	plot selector	plot selection for simulations of forest development
3	CAMELOT - **C**limate **A**nd **Me**teorological **Lo**ng-term **T**rends	climate predictor	prediction of the development of climatic characteristics of the forest

4	WALHALLA - **W**eather **A**pproximation **L**inked to **H**ourly-**A**nnual **L**ikelihood **L**imited **A**lgorithms	weather generator	generating weather characteristics from climate characteristics for hourly simulation intervals
5	ASGARD - **As**tronomical & **G**eographical **A**lgorithms of **R**adiation **D**istribution	radiation emulator	derivation of solar radiation characteristics with an hourly interval
6	SHANGRI-LA - **S**oil **H**ydrology **A**nd **N**utrients **G**eneral **R**egime **I**mpacts -**L**aunching **A**spects	soil generator	generation of soil characteristics
7	XIBALBA - Extended **I**nformation **Bal**ancing **B**ehavioural **A**spects	parameter adjustor	process model parameters adjusting
8	SHAMBALA - **S**imulation & **H**ierarchical **A**lgorithms of **M**atter **Bal**ance and **A**llocation	process-based calculator	calculation of the development of primary wood production with an hourly interval based on process models
9	THEBES - **Th**innings & **E**comanagement **B**uilding **E**cosystem **S**ervices	management controller	forest management settings
10	PAITITI - **P**rognosis **A**pplication for **I**ndividual **T**ree **I**n **T**ime **I**ncrementing	forest predictor	forest development prediction
11	AVALON - **A**dvanced **Val**uation **o**f **N**ature	ecosystem services valuator	assessment of forest ecosystem services based on quantification
12	AKATOR - **A**nalytical **K**it for **A**chieving **T**rends & **O**bserving **R**elations	forest analyser	analysis of forest development in time and space
13	ANGKOR - **A**dditional **N**ative **G**eneral **K**ind **o**f **R**eferences	reference models	yield tables, distribution, and biome model as reference models
14	AGHARTHA - **Ag**gregated **H**elp **A**bout **R**elations, **Th**eory & **A**pplications	model guide	growth model principles and algorithms description

Conclusion

The presented methodology and application for forest visualization uses the potential of the Unity 3D game engine. It has a high degree of immersiveness and is integrable with the growth model. The potential of the application can be used in the educational process, research, public relation and in selected practical applications. Another challenge for improving the application seems to be the addition of several tree variants due to age, quality and damage, the possibility of integrating the visualization of scanned point clouds in addition to forest visualization as well as migrating the application to HMD (Head Mounted Display) devices such as OCULUS RIFT or HTC Vive. However, the most important task is the development of growth simulator modules, as presented in Table 1. Their completion will create a robust tool for forest development prediction and its sophisticated visualization.

Acknowledgement

Presented research was supported by project KEGA No. 011TU Z-4/2019, titled 'Forest visualization by Unity 3D game engine for e-learning' and project No. APVV-19-0035, titled 'Simulation and visualization analytical tool for forest planning (SAVANT)'.

References

CAVE (2021): web link: http://etools.tuzvo.sk/cave/

FABRIKA, M., 2005: Simulátor biodynamiky lesa SIBYLA. Koncepcia, konštrukcia a programové riešenie. (SIBYLA Forest biodynamics simulator. Conception, construction, and programming solution.) Habilitačná práca (Habilitation Thesis), Technická univerzita vo Zvolene (Technical university in Zvolen), 238 p.

Fabrika, M., Pretzsch, H. (2013): Forest ecosystem analysis and modelling. Technical university in Zvolen, 620 p.

Nagel, J. (1996): Anwendungsprogramm zur Bestandesbewertung und zur Prognose der Bestandesentwicklung. Forst und Holz, 3, p. 76-78.

Petráš, R. (1989): Matematický model tvaru kmeňa ihličnatých drevín (Mathematical model of the stem taper curve for coniferous species). Lesnictví, 35, Nr. 10, p. 867-878.

Petráš, R. (1990): Matematický model tvaru kmeňa listnatých drevín (Mathematical model of the stem taper curve for deciduos species). Lesnícky časopis, 36, Nr. 3, p. 231-241.

Pretzsch, H., Biber, P., Ďurský, J. (2002): The single tree-based stand simulator SILVA: construction, application and evaluation, Forest Ecology and Management 162, p. 3-21.

SIBYLA Triquetra (2021): web link: http://sibyla.tuzvo.sk/