Tree Architecture & Blockchain Integration: An off-the-shelf Experimental Approach

DIMITRIOS VARVERIS[1], ATHANASIOS STYLIADIS[1*], PANTELEIMON XOFIS[1], LEVENTE DIMEN[2]

[1]Department of Forest & Natural Environment Sciences,
International Hellenic University,
14th km Thessaloniki, Nea Moudania 570 01, Thessaloniki,
GREECE

[2]Faculty of Informatics and Engineering
"1 Decembrie 1918" University of Alba Iulia,
15-17 Uniri Street, Alba Iulia 510009, Alba Iulia,
ROMANIA

Corresponding Author

Abstract: - Temporally sensitive tree modeling and urban park spatially explicit simulation offer advantages to large-scale landscape planning and design, especially in the context of smart applications for virtual parks and forests, while Blockchain technology provides collaborative engineering, data integrity, and information confidence. A proof-of-concept 2.5D tree architecture and Blockchain integration technique (distributed Internet-of-Trees images, "IoTr-images") was presented as a low-cost metaverse case study that affects the forest monitoring and digital landscape architecture design infrastructures. At the core of the proposed feature-based parametric modeling methodology is a 2.5D tree CAD model composed of two perpendicular 2D tree frames on which recorded tree texture has been assigned. A "Batch command-line programming" technique has been implemented, as a user-defined routine at the top of a commercial CAD platform, to describe the proposed off-the-self method and to create tangible tree-image NFT tokens (Internet-of-Trees-images Blockchain). As important findings were recorded, the add-in planning intelligence, the superior data integrity, and confidence, the offline relaxed error-free CAD design, and the superiority in terms of time and cost compared to traditional 3D tree modeling methods (laser scanning, close-range photogrammetry, etc.); as well as the satisfactory tree modeling accuracy for smart forest monitoring and landscape architecture applications. The proposed 2.5D parametric tree model added new value to the CAD-Blockchain integration industry because a plain "Blockchain/Merkle hash tree" tracks tree geometry growth and texture change temporarily with simple parametric transactions (i.e. controlled hash tree magnification/scaling). So, metaverse functionality (decentralized, autonomous, coordinated, and parallel design; same-data sharing; data validation), modification and redesign ability, and planning intelligence are effectively supported by the proposed technique. Main contributions are regarded as the ability for smart forest distributed surveillance and collaborative parallel landscape architecture design, open-source Web-based educational simulations, as well as the potential for off-the-shelf contractual collaborative frameworks (smart contracts between designers and clients). Stratification based on forest types improved above-ground biomass (AGB) estimation, especially when AGB was greater than 500 Mg/ha, using the proposed "IoTr-images" technique. So, this research provides new insight into AGB modeling and monitoring. Finally, the proposed method's robustness has been validated by performance evaluation testing.

Key-Words: - Environmental modeling, geodesign, tree modeling architecture, distributed and collaborative CAD, smart forest monitoring, landscape architecture, AGB, Metaverse, Blockchain functionalities.

Received: April 29, 2023. Revised: July 17, 2023. Accepted: September 12, 2023. Published: October 10, 2023.

1 Introduction

This is a descriptive paper on a complex process regarding tree architecture and Blockchain. Here, we report the results of an experiential approach to applying spatially explicit tree modeling and environmental simulation to forests and urban parks monitoring in the design and planning process. The length of this paper does not allow for an in-depth discussion about the specifics of our approach. Instead, we focus on the application of the results in distributed and collaborative design and planning with metaverse Blockchain functionality.

3D tree modeling from mobile laser scanning and terrestrial close-range photogrammetry techniques provides the necessary data for tree architecture, forest and park monitoring, and urban transformations. In, [1], the authors declare that an adaptive urban transformation needs tree and flora visualization data, and according to, [2], tree geometry is necessary for spatial GIS analysis regarding urban parks. According to, [3], Blockchain in distributed CAD environments supports data integrity, distributed, collaborative, and validated digital design, and information confidence, and as stated by, [4], a distributed digital design needs a CAD-Blockchain integration strategy. This literature approaches the "Trees/Flora – Forest/Urban parks – collaborative design/Blockchain" problem in detail but without describing a low-cost integration solution. Similarly, in, [5], the authors discuss data for structural monitoring in asynchronous communication, while in, [6], the authors discuss the consequences of Blockchain technology on the building information modeling (BIM) process but without reference to trees as the basic representation element of an urban, green-oriented environment.

According to the current study, monitoring in nearly real-time 3D tree geometry growth and tracking tree texture change temporarily is an ideal off-the-shelf distributed and collaborative environment but means time, cost, and an exhausting "Blockchain/Merkle hash tree". In, [7], the authors describe in detail a method for linking image-based metrics to 3D model-based ones for the assessment of visual landscape quality, but no information is given about a potential Blockchain integration. Finally, according to, [8], decision-support models, for sustainable urban investment optimization, could be useful in urban transformation processes, but no implementation details for collaborative and distributed environments are given.

The proposed technique uses batch commands and event-driven routines (design level/layer, color, weight, style, reference point, height, and width) for 2.5D parametric tree CAD modeling relative to a *ground reference point* (GRP) for tree CAD geometry deployment (geo-referenced tree-CAD frame's). Thus, with the proposed technique we save time, greatly reduce costs, and achieve the operation and maintenance of a simple and functional Blockchain with all that this implies at the level of distributed and collaborative design. Certainly, the proposed technique produces less accurate tree models, but this disadvantage does not affect smart forest or landscape architecture applications, [1], [2].

For the proposed technique the *research questions* are described as follows: • Describe parametric and relative 2.5D easy-to-design tree CAD modeling methodology based on simple tree images (e.g. jpg format); • Design Blockchain data structures for tangible tree-image and tree-model NFTs respectively; and • Code batch command-line programming for hooking user-defined *commands* (simple English phrases) to CAD domain-dependent software routines (system key-ins). The *commands* refer to both 2.5D tree modeling, as well as building and maintaining the Blockchain structure, [9].

The article's main *aim* is to develop an experimental technique for integrating tree images in mutually perpendicular 2D CAD frames and connecting them into a dedicated "Internet-of-Tree" Blockchain. So, after describing the general research questions, this article's specific research *objectives* are defined as (1) The implementation details of the tree CAD modeling methodology; (2) The implementation details of the Metaverse/Blockchain "tree display file" structure/metadata, in DXF ASCII format, for tangible tree-image and tree-model NFT tokens; (3) The smart and user-friendly "Batch command-line programming" implementation technique with command-line *commands* (ASCII text) hooked to CAD-domain dependent (Bentley's MicroStation CAD platform was selected) KEY-Ins; and (4) The implementation of the "Internet-of-Tree" Blockchain app case study.

The 2.5D models are decentralized among several nodes that hold identical information, and at the same time, none (designer or client) holds the complete authority (digitally distributed consensus). This enables transparency of design activity and enhancement of 2.5D data security.

Novelties: (i) The proposed 2.5D tree model added new value to the CAD-Blockchain integration industry, thanks to a very plain "Merkle hash Tree" that tracks tree geometry growth and texture change temporarily with simple parametric transactions. So, smart forest surveillance; decentralized, autonomous, coordinated and parallel design; same-data sharing; tree modeling data validation; design files transaction management with analytics functionality; and contractual frameworks (e.g. smart and performance-based contracts) are effectively supported; (ii) Smart forest add-in app with simple Blockchain transactions (tracking tree growth in the course of time); (iii) The innovative "2.5D tree CAD modeling – Blockchain integration" methodology for parametric, relative, and untagged geo-referenced tree modeling; (iv) The smart "Batch command-line programming" implementation technique, associated with key-ins (domain-dependent dedicated CAD software tools) for 2.5D generic and parametric tree modeling by simply writing -in an online text editor (Notepad)- commands with phrases from the English language (plain ASCII text data files for job control); and (v) The "tree display file" Blockchain structure storing the metadata of a DXF/ASCII representation of generic tangible tree-image NFTs, [7], [9].

The particular significance of this study lies in the plain "Blockchain/Merkle hash tree" structure supporting distributed and collaborative tree architecture for monitoring landscape architecture, smart forest, and digital documentation apps where there are no requirements on the accuracy of the tree representation, [10], [11], [12].

The rest of the paper is organized as follows. In Section 2 ("Method and Technique") the proposed low-cost "*2.5D tree CAD modeling – Blockchain integration*" method and the "*Batch command-line programming*" implementation technique are introduced. In Section 3 ("Results") the outline design of an "Internet-of-Tree images" Blockchain case study (experimental approach) is presented followed by a comparative validation analysis as a "IoTr-images" usability test. Finally, in Section 4 ("Discussion and Conclusion") the results, major findings, article's main significance and contributions, as well as spotted limitations and suggestions for improvements and further study are discussed, followed by concluded remarks.

2 Method and Technique

Parametric modeling lets designers modify the entire shape of the design at once, not just individual dimensions one at a time. Also, a feature-based parametric modeling CAD software design tool saves time as it eliminates the need for a design engineer to constantly redraw a design every time one of the design's dimensions changes.

MicroStation, AutoCAD, Pro/ENGINEER, and SolidWorks offer direct modeling CAD platforms on top of existing feature-based parametric modeling, [12].

2.1 2.5D Tree CAD Modeling – Blockchain Integration Method

For the so-called 2.5D generic tree modeling, the relative design of two rectangular tree-frames, perpendicular to each other, starting with a GRP with coordinates 0,0,0 is preceded (geo-referenced functionality for relative to *ground reference point* modeling deployment) (Figure 1).

Next is the assignment of low-cost tree images (e.g. smartphone's jpg images) in these frames (Figure 2), and the storage of this generic tree format as a 2.5D tree CAD model (at a local hard disk), and as a tangible tree image NFT token as well ("Internet-of-Trees image"/IoTr-images chain at the metaverse cloud). For the implementation of the "2.5D tree CAD modeling – Blockchain integration" method, a smart "Batch command-line programming" technique is used, [1], [9], [12], [13], [14].

In more detail, the proposed 2.5D tree CAD modeling procedure takes as parameters the CAD current drawing level (LV), frame's line color (CO), line weight (WT), line style (LC), frame's geo-referencing point coordinates (GRP), and tree's height (H) and width (W), and is developed as follows:

(a) The tree-frame's CAD parameters LV, CO, WT, and LC are predefined.
(b) In the CAD platform (FRONT view) a tree frame is designed according to the tree's height parameter (H) and for a generic geo-referencing, the GRP is assigned to coordinates 0,0,0 (CAD universe space) (Figure 1, bottom-left window).
(c) In the CAD platform (RIGHT view) another tree-frame is designed according to the tree's width parameter (W) (Figure 1, bottom-right window).
(d) The last tree frame is relocated in such a way that both frames cross each other perpendicularly (Figure 1, top-left window).

In this way, a compound tree frame is designed starting from the GRP (0,0,0) (Figure 1, top-right window).

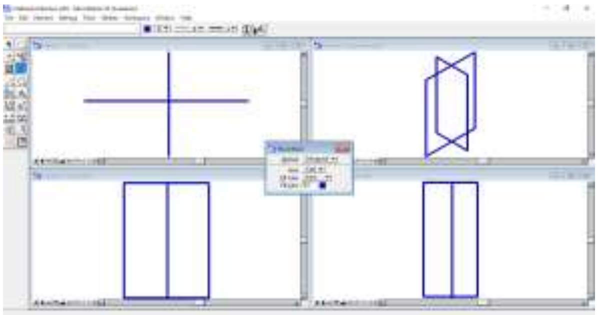

Fig. 1: The two orthogonal tree frames cantered perpendicularly (FRONT and RIGHT view)

(e) A tree's scalable raster image (jpg), acquired from (i) the CAD platform's system palette with flora images (e.g. "Flora.pal"), or (ii) the user's noise-free photography (e.g. a smartphone jpg image), or (iii) the proposed IoTr-images ecosystem, is assigned to both tree-frames (as the unique geometry located in level LV and colored with color CO), utilizing a user-defined GUI's dialog setting box, according to the predefined level (LV) and color (CO) parameters (Figure 2).

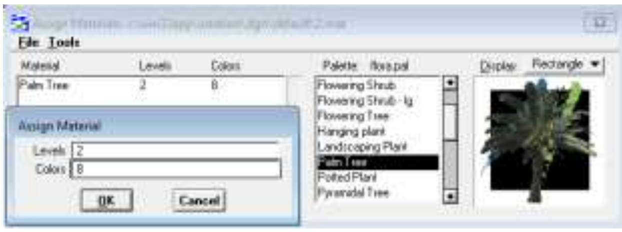

Fig. 2: Tree texture image assignment to FRONT and RIGHT view tree-frames.

In this way, the modeling is referred to as "2.5D tree CAD modeling" and the visualization accuracy is adequate, sufficient, and satisfactory for digital documentation and visualization purposes regarding tree landscapes, forests, monument landscapes rich in trees, and landscape architecture applications.

Figure 3 presents a 2.5D tree CAD model (ISOMETRIC view). The modeling accuracy is adequate for projects and applications without special tree-shape accuracy and visualization requirements.

Fig. 3: A 2.5D tree CAD model.

2.2 CAD Software (Event-Driven Procedures & Key-Ins)

The parametric modeling routines assigned to Icon tools (user-defined event-driven PROCEDURES) were implemented in MDL (MicroStation Development Library). MDL code can be compiled using Microsoft Visual C++ as a native-code DLL. This both enhances programmer productivity with C++ object-oriented concepts and provides better performance. The command-line BATCH COMMANDS were implemented as plain text in an online text editor (Notepad). For batch job control, these commands are organized in batch files and use phrases from the English language (plain ASCII text data). The CAD platform MicroStation was used as the hosting software environment for the event-driven procedures and the command-line programming of the proposed framework. MicroStation® is Bentley Systems' CAD product and one of its many strengths is its adaptability. Inherent to that adaptability are tools to customize and extend MicroStation.

User customization and task-specific tools: MicroStation lets an administrator modify its user interface and create custom menus, palettes (toolboxes), and icon tools (icon buttons) that provide a fast track to commands and functions used frequently. The icon tools are grouped into modeling thematic palettes, and they are assigned to event-driven procedures (user-defined CAD s/w) or key-ins (CAD system s/w / Bentley's propriety MDL source code). In the presented research, several event-driven procedures were written in MDL/C++ as .mc source code, linked to appropriate libraries, compiled to new .ma executable routines (dedicated to smart tree landscape framework user-defined procedures) and then they are grouped into the "TREE.ma" exe file.

Figure 4 presents the development process for the graphic representation of the new icon tool "PLACE Tree Frame". The modeling duty of this icon tool is to design a rectangular parametric and relative tree frame in any CAD view (Front, Right, Left, etc.).

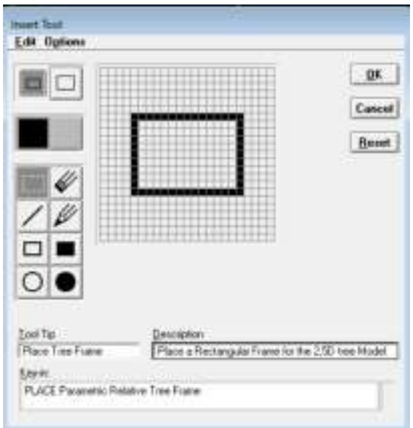

Fig. 4: The "PLACE Tree Frame" icon tool: Graphic design and assignment to user-defined key-in "PLACE Parametric Relative Tree Frame".

The "PLACE Tree Frame" icon tool is assigned to a dedicated user-defined event-driven procedure embedded into the new 2.5DTREE.ma MDL application, and has been grouped, as the 3rd tool on the 2nd row, into the thematic palette "Icon Tools for 2.5D TREE" (menu: Smart Tree Landscapes (Customized GUI technique) / Trees / 2.5D tree CAD geometry/setup FRONT Frame) and on a "left-button" event (cursor hit activation) the dedicated user-defined software procedure "PLACE Parametric Relative Tree Frame" is called (Figure 5).

Fig. 5: The pull-down menus: "Smart Tree Landscapes (Customized GUI)", "Setup NFT Wallet", "create NFT", and "Add NFT" for adding a tangible tree-image NFT to MetaMak WAX Blockchain.

2.3 Batch Command-Line Programming Technique

At the heart of the batch command-line programming implementation technique is a NotePad or WordPad ASCII text file located outside of the CAD platform, e.g., the batch command file "*TreeModeling.bat*" located at the local Hard Disk (Figure 6). The "2.5D tree CAD modeling – Blockchain integration" methodology is performed with a batch top-down job control (Figure 6).

So, initially, a design file segmentation is performed by allocating discrete tree model parts to design session layers/levels (e.g. in our experiment level 22 was chosen to host the GRP and level 11 to host the tree CAD model). With that "segmentation", we gain design independence, modification functionality, and redesign flexibility, because by withdrawing some levels we can denominate a design unit and focus on it. Following, the *ground reference point* and the *tree modeling settings* are defined by level, line color, line weight, and the 0,0,0 coordinates for the GRP.

Subsequently, in segmentation level (LV) 11, with line color (CO) 0, line weight (WT) 1, and line style (LC) 0, the 2.5D CAD geometry of the tree, the tree imagery and texture assignment, and the tangible tree-image NFT (cloud wallet, token, "IoTr-images" Blockchain) were deployed step by step (Figure 6).

```
// The code interpreting the proposed "Internet-of-Tree Images" Blockchain (IoTr-images)
        // The Ground Reference Point (GRP)
LV=22; CO=4; WT=5; PLACE GRP; XYZ=0,0,0; RESET
        // Tree model settings (level, color, weight)
LV=11; CO=0;  WT=1;  LC=0
        // Tree Geometry
PLACE Tree Corner (XYZ coordinates)
PLACE BLOCK (Tree "Front" frame)
PLACE BLOCK (Tree "Right" frame)
        // Hooking Tree Image to 2.5D Tree Geometry
ASSIGN Image to tree frames          // Imagery from e.g. Flora.ma library
ASSIGN Image to tree frames          // Imagery from a user-defined library
ASSIGN Image to tree frames          // Imagery from IoTr WAX blockchain
        // Tree Settings (size, shape)
Tree-HEIGHT; Tree-WIDTH
        // Tree Texture
ATTACH Texture
        // Tree-Tangible NFT (Metaverse ecosystem / Blockchain structure)
SETUP Tree-Image NFT Cloud Wallet
CREATE Tangible Tree-Image NFT token      // Token for the "Tree-Image"
ADD Tangible Tree-Image NFT to "IoTr-images" Blockchain
```

Fig. 6: The batch command file *TreeModeling.bat* (ASCII text file with command-line COMMANDS).

The "*TreeModeling.bat*" batch file can be executed from a CAD software platform prompt (key-in dialog box) by typing a link string. E.g.
@c://PhD-Research/Smart-TreeLandscapes/TreeModeling.bat
The introduced "2.5D tree CAD modeling – Blockchain integration" method, implemented with the "Batch command-line programming" technique, is a "*smart and distributed CAD modeling*" procedure because operates in near real-time with planning intelligence (modeling and design process modification ability in near real-time) and Internet of Things (IoT) metaverse efficiency. Also, it is flexible because it is performed offline in a friendly safety and relaxed way, with simple phrases from the English language as ASCII CAD-platform-offline coding "Commands" hooked to domain-dependent key-in routines.

3 Results

For the demonstration of the introduced distributed and collaborative smart tree landscape framework (method and implementation technique), an “Internet-of-Tree images” Blockchain outline design, for an urban landscape architecture design project, is presented in Fig 7. Follows a comparative analysis’s usability test for validation purposes.

3.1 “Internet-of-Tree images” Blockchain: The Outline Design

The outline design of the “Internet-of-Tree images” Blockchain case study (metaverse application for an urban park rich in trees) is displayed in Figure 7. Each tree, as a block in the IoTr-images chain, is referred to by a hash value created by the SHA256 cryptographic algorithm. Hence, the “IoTr-images” Blockchain is composed of a linked list of blocks of transactions (tree-frame dimensions and raster image data in autonomous collaborative design) tracking tree geometry growth and AGB texture change over time, [3], [4]. The “Root of Hash Tree” points to a “Merkle hash tree” chain of transactions, [6].

It is important to know that, in the discussed Blockchain the “Merkle hash tree” is relatively compact in its extension (small-scale Blockchain functionality), plain, and easily manageable (controlled magnification), [4], [6], [13], [15], [16] (Figure 7).

The jpg tree-photography “Tree-image.0”, “Tree-image.1”, “Tree-image.2”, and “Tree-image.3” together with relative tree geometry measures, are regarded as tree texture and tree geometry “Data Blocks” respectively (i.e. transactions for the “IoTr-images” Blockchain).

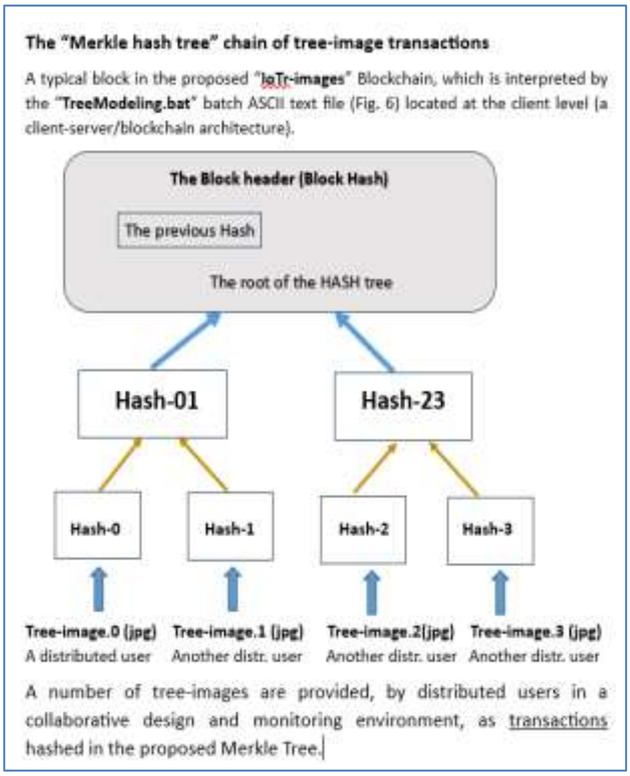

Fig. 7: The proposed “Blockchain/Merkle hash tree” (The “*Root of Hash Tree*” is pointing to a “Merkle hash tree” chain of transactions).

The “IoTr-images/Merkle hash tree”, for the rich-in-trees urban park, is constructed with a bottom-up approach. Hence, every leaf node (the Hash-0, Hash-1, Hash-2, and Hash-3 in Figure 7) is a hash of transactional data, i.e. the periodically recorded tree geometry growth (tree dimensions), and the AGB texture change (tree images), and the non-leaf node (the Hash-01 and Hash-23 in Figure 7) is a hash of its previous hashes. The proposed Merkle hash tree is a binary one, so it always requires an even number of leaf nodes.

Hash-01 = hash (Hash-0 & Hash-1)
Hash-23 = hash (Hash-2 & Hash-3)
Hash-0 = hash (tree-geometry.0 & tree-image.0)
Hash-1 = hash (tree-geometry.1 & tree-image.1)
Hash-2 = hash (tree-geometry.2 & tree-image.2)
Hash-3 = hash (tree-geometry.3 & tree-image.3)

Notes

1. For the *hashing* (hash values Hash-0, Hash-1, Hash-2, Hash-3, Hash-01, and Hash-23) the hash function SHA256 (cryptographic algorithm) has been used.
2. The very first “IoTr-image” block upon which additional blocks, in the proposed chain, have been added is called “*block 0*” or “*genesis block*”. This block represents the starting point of the “IoTr-images” Blockchain (ledger). It is hardcoded into

the Blockchain/Bitcoin software as the so-called foundational block.
3. On January 3, 2009, pseudonymous Bitcoin creator "Satoshi Nakamoto" mined the *genesis Bitcoin*, which led to the mining of the first 50 Bitcoins.
4. The reference to the *genesis Bitcoin* is made because, in self-executed agreements (smart contracts under the proposed "IoTr-images" Blockchain) between management (as urban planning project managers), landscape architects (as designers), and local authorities (as clients), the payments decided to be made with Bitcoin (BTC).

3.2 Comparative Validation Analysis ("IoTr-images" usability test)

This research conducted a comparative analysis of different tree datasets and modeling algorithms, between the proposed 2.5D tree modeling technique and traditional 3D modeling methods (laser scanning, terrestrial close-range photogrammetry for Above-Ground Biomass/AGB monitoring), [16]. Tree and forest type, as well as AGB range, influence tree modeling and they are important factors in comparative analysis.

The results show the following:
(i) Laser scanning imagery provides more accurate AGB estimates (RMSE values in about 28 Mg/ha) than the proposed "IoTr-images" technique (about 95 Mg/ha).
(ii) Overestimation for small AGB values (<50 Mg/ha) and underestimation for large AGB values (>300 Mg/ha) are major problems when using terrestrial close-range photogrammetry.
(iii) Stratification based on forest types improved AGB estimation, when AGB>500 Mg/ha, using the proposed "IoTr-images" technique. So, this research provides new insight into AGB modeling, [17].
(iv) The off-the-shelf "Batch command-line programming" technique is cheaper (expenditure), faster (time), user-friendly, and more flexible (design process modification ability and redesign functionality), [14].
(v) The proposed technique provides metaverse functionality (data validation), collaborative risk management analytics, planning intelligence (smart forest monitoring, landscape architecture design), and IoT efficiency in (nearly) real-time, [15], and
(vi) The end-user, with the interpretation batch file at a client level, can modify the offline and outside of the CAD environment the whole process by using a NotePad for editing the batch file (Fig. 6). The modification was performed with AI functionality, in a simple way (text editor), offline (CAD platform), and without stress/risk of a design accident, [10], [12].

4 Discussion and Conclusion

Results: (a) A simple 2.5D parametric and relative tree CAD modeling methodology has been described for tangible tree images as NFT tokens (IoTr-images Blockchain) facilitating trees and smart forests distributed design and collaborative monitoring; and
(b) A smart, safe, relaxed, and error-free "Batch command-line programming" tree modeling approach has been implemented, with simple English language phrases as command-line commands for dedicated key-ins hooking, and the IoTr-images Metaverse/Blockchain.
Also, code has been written in MDL, an event-driven CAD programming language, for hooking Commands (simple English phrases) to CAD-domain dependent s/w (system Key-ins).
Findings: (i) Experimental data (findings) proved the satisfactory time performance, in tree shape modeling, of the proposed technique compared to manual terrestrial laser scanning and close-range photogrammetry methods.
(ii) A batch file, as an interpretation tool for the IoTr-images Blockchain case study, with simple ASCII plain-text commands, can support near real-time 2.5D tree modeling operations in a safe, relaxed, and error-free offline environment, with redesign flexibility, planning, and design functionality; and
(iii) The proposed technique supports coordinated design, same-data sharing, and parallel design (i.e. characterized by decentralized and autonomous design efficiency).
Significance, contributions, and limitations: The simple, plain, controlled magnification/scaling "Blockchain/Merkle hash tree" is considered the most important significance of the technique. Also, the proposed "2.5D tree CAD modeling - Blockchain integration" method could be studied as an open-source, web-based simulation, that provides students with virtual experiences of the impact of global natural disasters, such as unplanned, uncontrolled, and unpredictable forest fires, [18], [19].
Also, the most important contributions are considered the ability for low-cost smart forest monitoring and collaborative landscape architecture design, the superior data integrity and confidence (cryptographic data for tree geometry and AGB texture), as well as the potential for contractual frameworks for self-executing agreements (smart

contracts) between manager, designers, and clients to automate and control design and AGB monitoring, and to document and validate contract transactions in a secure, low-cost, and transparent manner without the need for superintendence by a central authority (i.e. track the design process and verify the tree and AGB data authenticity and ownership, create and manage decentralized AGB identity and authorization, verify ownership of AGB data as a digital asset, and support applications that run at the top of the "IoTr-images" decentralized Blockchain environment). An obvious limitation of the proposed "Batch command-line programming" technique is the poor tree modeling accuracy and visualization. Also, a limitation regards the CAD platform dependency of the user-defined source code (domain-dependent key-in procedures). However, feature-based parametric CAD routine adaptation, for compatibility reasons, is not a major problem.

Suggestions for improvements and further study: An improvement (open research issue) is an IoTr-images Blockchain with spatial analysis functionalities in near real-time for a secure and decentralized autonomous GIS. For this case, we need to incorporate georeferenced data into the tangible raster tree-image NFTs (ISO/TC 211 series of standards for geoinformation compliance).
Also, future research should study distributed and collaborative "tree architecture" reconstructed from rough videometry (scanned) 3D tree data instead of raster tree images. So, we need a video-driven tree architecture that implements video sequences to 3D model transformations using a flexible and fully configurable template. Nowadays, terrestrial videometry can provide the necessary point clouds for 3D tree geometry growth and 3D texture change.
Conclusion. The described conceptual case study (method and implementation technique) is a low-cost metaverse application that supports smart forest monitoring, coordinated and parallel design, plain and controlled hash tree scaling, same-data sharing, and a trustworthy collaborative design process (digitally distributed consensus); enabling coordinated activity, transparency, data security enhancement, and contractual framework functionality for smart contracts (self-executed agreements between designers and clients).

Acknowledgement:
We would like to acknowledge the support of the Department of Forest & Natural Environment Sciences, International Hellenic University (Greece).

References:
[1] Nijhuis, S., Sun, Y., Lange, E., *Adaptive Urban Transformation*, Springer 2023, DOI: 10.1007/978-3-030-89828-1.
[2] Ma, Y., Brindley, P., Lange, E., A Comparison of GIS-based Methods for Modelling Walking Accessibility of Parks in Guangzhou Considering Different Population Groups, *Journal of Digital Landscape Architecture*, 7, pp.269-279, 2022, DOI: 10.14627/537724026.
[3] Lemeš, S., Lemeš, L., Blockchain in Distributed CAD Environments. In Karabegovic, I. (ed.) *New Technologies, Development and Application II*, Springer Nature, pp.25-32, 2020, DOI: 10.1007/978-3-030-18072-0_3.
[4] Dounas, T., Lombardi, D., A CAD-blockchain integration strategy for distributed validated digital design. In Kępczyńska-Walczak, A. and Bialkowski, S. (eds.) *Computing for a Better Tomorrow* proceedings of the 36th International Conference on Education and Research, eCAADe, pp.223-230, 2018.
[5] Sabba, M.-F., Lerna, M., Diaferio, M., Foti, D., Satellite Data for Structural Monitoring of Historical Building: The Temple of Minerva Medica in Rome, *WSEAS Transactions on Environment and Development*, 17, pp.1284-1289, 2021, DOI: 10.37394/232015.2021.17.117.
[6] Nawari, N.O., Ravindran, S., Blockchain technology and BIM process: review and potential applications, *ITcon*, 24(1), pp. 209-238, 2019, https://www.itcon.org/2019/12.
[7] Xudong, Z., Lin, S., Qi, J., Yok, T.P., Linking Image-based Metrics to 3D Model-based Metrics for Assessment of Visual Landscape Quality, *Journal of Digital Landscape Architecture*, 8, pp.167-177, 2023, DOI: 10.14627/537740018.
[8] Morano, P., Tajani, F., Guarnaccia, C., Anelli, D., An Optimization Decision Support Model for Sustainable Urban Regeneration Investments", *WSEAS Transactions on Environment and Development*, 17, pp.1245-1251, 2021, DOI: 10.37394/232015.2021.17.114.
[9] Yazdi, H., Shu, O., Ferdinand, L., A Target-driven Tree Planting and Maintenance Approach for Next Generation Urban Green

Infrastructure (UGI), *Journal of Digital Landscape Architecture*, 8, pp.178-185, 2023, DOI: 10.14627/537740019.

[10] Fox, N., Lidquist, M., Berkel, V.D., Vergel, R.S., A Collaborative Augmented Reality Decision Support System for Crowdsourcing Urban Designs, *Journal of Digital Landscape Architecture*, 8, pp.195-202, 2023. DOI: 10.14627/537740021.

[11] Hasanagas, N.D., Styliadis, A.D., Papadopoulou, E.I, Environmental policy and science management: Using a scientometric-specific GIS for E-learning purposes, *Int. J. Comput. Commun. Control*, 5(2), pp.171-178, 2010, DOI: 10.15837/ijccc.2010.2.2472.

[12] Styliadis, A.D., E-learning documentation of historical living systems with 3-D modeling functionality, *Informatica*, 18(3), pp.419-446, 2007, DOI: 10.15388/Informatica.2007.186.

[13] Mavlutova, I., Lesinskis, K., Liogys, M., Hermanis, J., Innovative Teaching Techniques for Entrepreneurship Education in the Era of Digitalisation, *WSEAS Transactions on Environment and Development*, 16, pp.725-733, 2020, DOI: 10.37394/232015.2020.16.75

[14] Sharma, R.P., Vacek, Z., VacekA, S. et al., Modelling individual tree height–diameter relationships for multi-layered and multi-species forests in central Europe, *Trees*, 33(1), pp.103-119, 2019, DOI: 10.1007/s00468-018-1762-4.

[15] Basdekidou, V., Papapanagos, H., Empirical Model for Estimating Sustainable Entrepreneurship's Growth Potential and Positive Outlook, *Baltic J. Modern Computing*, 11(1), pp.181-201, 2023, DOI: 10.22364/bjmc.2023.11.1.11.

[16] Li, D., Jia, W., Guo, H., Wang, F., Ma, Y., Peng, W., Zhang, S., Use of terrestrial laser scanning to obtain the stem diameters of Larix olgensis and construct compatible taper-volume equations, *Trees*, 37(1), pp.749-760, 2023, DOI: 10.1007/s00468-022-02381-2.

[17] Hasanagas, N.D., Styliadis, A.D., Papadopoulou, E.I., Sechidis, L.A. E-Learning & environmental policy: The case of a politico-administrative GIS, *Int. J. Comput. Commun. Control*, 5(4), pp.517–524, 2010, DOI: 10.15837/ijccc.2010.4.2509.

[18] Xofis, P., Spiliotis, J.A., Chatzigiovanakis, S., Chrysomalidou, A.S., Long-Term Monitoring of Vegetation Dynamics in the Rhodopi Mountain Range National Park - Greece, *Forests*, 13(3), 377, 2022, DOI: 10.3390/f13030377.

[19] Davies, P.-M., Weir, P., Byrne, S., Building Educational Simulations Using KSA Open Data, *WSEAS Transactions on Environment and Development*, 16, pp.802-810, 2020, DOI: 10.37394/232015.2020.16.83.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.