ICTree: Automatic Perceptual Metrics for Tree Models

TOMAS POLASEK, Brno University of Technology, CPhoto@FIT
DAVID HRUSA, Purdue University
BEDRICH BENES, Purdue University; Czech Technical University in Prague, FEL
MARTIN ČADÍK, Brno University of Technology, CPhoto@FIT; Czech Technical University in Prague, FEL

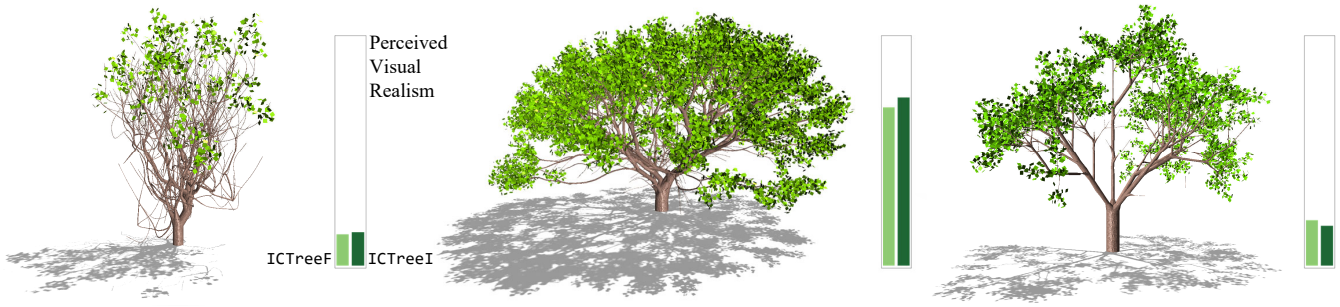

Fig. 1. A tree's perceived visual realism guides the user during a real-time interaction with a 3D tree modeling system. Two metrics are provided, a view-independent ICTreeF uses the 3D geometry solely (light green bar, left), and ICTreeI uses a 2D image of the tree (dark green bar, right). The initial tree (left) is estimated as not visually pleasing, as indicated by the low values of the ICTreeI and ICTreeF. After adding branches and randomizing their values, the perceived realism has improved significantly (middle). The user then decreases the branching angles' randomization and the branch lengths that impact the perceived realism (right).

Many algorithms for virtual tree generation exist, but the visual realism of the 3D models is unknown. This problem is usually addressed by performing limited user studies or by a side-by-side visual comparison. We introduce an automated system for realism assessment of the tree model based on their perception. We conducted a user study in which 4,000 participants compared over one million pairs of images to collect subjective perceptual scores of a large dataset of virtual trees. The scores were used to train two neural-network-based predictors. A view independent ICTreeF uses the tree model's geometric features that are easy to extract from any model. The second is ICTreeI that estimates the perceived visual realism of a tree from its image. Moreover, to provide an insight into the problem, we deduce intrinsic attributes and evaluate which features make trees look like real trees. In particular, we show that branching angles, length of branches, and widths are critical for perceived realism. We also provide three datasets: carefully curated 3D tree geometries and tree skeletons with their perceptual scores, multiple views of the tree geometries with their scores, and a large dataset of images with scores suitable for training deep neural networks.

CCS Concepts: • **Computing methodologies → Shape analysis**; **Perception**.

Authors' addresses: Tomas Polasek, ipolasek@fit.vutbr.cz, Brno University of Technology, CPhoto@FIT, Bozetechova 2, Brno, Czech Republic, 612 00; David Hrusa, hrusadav@gmail.com, Purdue University, 401 Grant St #150, West Lafayette, United States, 47906; Bedrich Benes, bbenes@purdue.edu, Purdue University; Czech Technical University in Prague, FEL, 401 Grant St #150, West Lafayette, United States, 47906; Martin Čadík, cadik@fit.vut.cz, Brno University of Technology, CPhoto@FIT; Czech Technical University in Prague, FEL, Bozetechova 2, Brno, Czech Republic, 612 00.

0730-0301/2021/12-ART1
https://doi.org/10.1145/3478513.3480519

Additional Key Words and Phrases: tree models, perceptual assessment, neural networks, geometry

ACM Reference Format:
Tomas Polasek, David Hrusa, Bedrich Benes, and Martin Čadík. 2021. ICTree: Automatic Perceptual Metrics for Tree Models. *ACM Trans. Graph.* 40, 6, Article 1 (December 2021), 15 pages. https://doi.org/10.1145/3478513.3480519

1 INTRODUCTION

There is a surprising disconnect between the perceived realism and geometric details in synthetic vegetation models in computer graphics (CG). Several decades of research have provided many methods for vegetation modeling ranging from fractal-based models [Oppenheimer 1986], methods based on biology [Prusinkiewicz and Lindenmayer 1990], interactive techniques [Lintermann and Deussen 1999], to physics-based simulations [Pirk et al. 2014, 2012].

While a large body of work in CG deals with the perceived realism of synthetic objects (*e.g.,* rendered images [Andersson et al. 2020; Čadík et al. 2013; Herzog et al. 2012; Wolski et al. 2018], urban models [Beneš et al. 2017], meshes [Torkhani et al. 2015]), only a handful of methods focus on the generated plant model validation (see Sect. 2). The validations are often subjective or provided as side-by-side images.

We introduce the ICTree. A novel no-reference metric that provides the perceived realism of an artificial tree. We gathered and curated a database of 100 tree geometries. Then we conducted an extensive user study in which 4,000 participants compared over one million image pairs with trees, and subjective scores were collected. The scores were used to train two deep predictors ICTreeI and ICTreeF. ICTreeI is applied to images where the tree is rendered without background, and it facilitates evaluations of existing and

forthcoming tree generation algorithms, making it possible even for cases where the tree geometry is difficult to process. ICTreeF works with a 3D model from which it uses a set of simple features. It evaluates tree models independent of tessellation and rendering style. Furthermore, it may help users identify and change important features while interactively generating a plausible tree model. The feature-based model makes it easy to use models in many possible representations. Moreover, training neural networks directly on 3D meshes, point clouds, or skeletons is challenging.

Our approach addresses a longstanding need in computer graphics for automatic metrics of the perceived visual realism of stochastic models. Our metrics can be coupled with interactive methods for tree modeling, where they answer the question of how the model would be perceived, as shown in Fig. 1. Each time the user changes the 3D model, our metrics instantaneously predict the perceived visual realism. Alternatively, it can be used in procedural methods that generate tree models automatically but without information about their visual realism. Such metrics can be helpful in deep-learning algorithms, such as GANs or reinforcement learning. An example of such an application is provided in Fig. 19 that takes a set of parameters of a procedural model and finds a model with similar parameters with a higher perceptual score. We have evaluated the intrinsic tree attributes. We observed that while some features significantly impact the perceived realism (*e.g.,* sibling angle, length of a branch, or branch volume), other attributes (*e.g.,* a thickness of a branch segment or deformation of the sequence of branches) are not so prominent. This observation was further confirmed by a follow-up user study.

We claim the following contributions: (1) ICTree, a novel automatic no-reference realism predictor that provides the perceived level of realism of synthetic model of a tree either from an image (ICTreeI) or from its set of easy to calculate features (ICTreeF), (2) three large datasets i) 100 carefully curated 3D tree models and their skeletons with perceived user realism scores from an extensive user study ii) 500 tree images with user-assigned perceived realism, and iii) 8,500 images of tree views used for training deep neural networks, and (3) an insight into what tree features have a significant effect for perceived realism. All datasets, trained neural networks, and the source code for feature generation are available at **cphoto.fit.vutbr.cz/ictree**.

2 RELATED WORK

Perception in Computer Graphics: Human perception has been successfully used in many fields of CG, and close to our work are image quality metrics (IQMs) that predict perceptual quality (or realism) of images. Full-reference IQMs compute perceptual differences between the reference and distorted images [Mantiuk et al. 2011; Wang et al. 2004; Wolski et al. 2018], while no-reference metrics [Bosse et al. 2018; Herzog et al. 2012; Jung et al. 2002; Moorthy and Bovik 2010; Suresh et al. 2009; Tang et al. 2011; Ye et al. 2014] predict the image quality in a challenging reference-less scenario. Traditionally, IQMs focus on the evaluation of compression [Jung et al. 2002; Wang et al. 2004], transmission [Kundu et al. 2018], or rendering artifacts [Aydın et al. 2010; Čadík et al. 2013; Ramanarayanan et al. 2007] and cannot be directly applied to the evaluation of naturalness of synthetic objects. Recently, Zhang et al. [2018] found that deep features outperform previous metrics by large margins, which holds across different deep architectures.

Perceptual realism of 3D shapes is significantly less explored than the realism of images. Specifically, Rogowitz and Rushmeier [2001] note that 2D IQMs do not adequately capture perceptual realism of the 3D object simplification and find that lighting and shadows impact the perceived realism. On the other hand, Lavoué et al. [2016] show in extensive experiments that IQMs may be efficient in evaluating the realism of different versions of the same object under a single type of distortion (*e.g.,* compression). However, IQMs are less accurate in comparing different distortions and distortions applied to different 3D models. Pan et al. [2005] show that human viewers are more sensitive to objects' texture rather than geometry. Furthermore, several full-reference 3D metrics [Corsini et al. 2013] have been proposed to assess the perceptual quality of meshes degraded by lossy compression, watermarking, or simplification [Lavoué 2011; Wang et al. 2012]. Unfortunately, such metrics cannot be used to assessing the naturalness of 3D objects because they are tailored to a particular distortion and require a reference 3D model. Rajasekaran et al. [2019] utilize extracted features for quality assessment of terrain models, which are then transferred using CycleGAN to less realistic models to enhance their perceived realism. Visual saliency predictors for 3D meshes have also been proposed [Wu et al. 2013].
Vegetation simulation is a significant open problem in computer graphics, and the methods for plant generation can be classified into interactive approaches, reconstruction, and biological modeling.

Biological modeling commenced with a simulation of cellular subdivision by means of Lindenmayer's systems [Lindenmayer 1968] (L-systems). Prusinkiewicz et al. [1993] extended the parallel string rewriting systems into the generation of 3D structures by introducing bracketed L-systems and the geometrical representation by a finite-state automaton [Prusinkiewicz 1986]. L-systems have been extended to capture continuous growth, growth limited by environment [Měch and Prusinkiewicz 1996; Prusinkiewicz et al. 1994], and the simulation of ecosystems [Deussen et al. 1998].

Environment plays an important role because it significantly affects the shape. Some early approaches considered space occupancy rules [Borchert and Honda 1984]. The competition for resources has been studied in the context of plants growing on obstacles [Benes and Millán 2002; Greene 1989] and even used as an interactive tool [Hädrich et al. 2017]. Space occupancy has been extended by Palubicki et al. [2009] as a means to simulate the generation of complex trees affected by the environment. Recent approaches focus on physics simulation, such as interaction with wind [Habel et al. 2009; Pirk et al. 2014], fire [Pirk et al. 2017], and some study plant material properties [Wang et al. 2017; Zhao and Barbič 2013].

Some of the *interactive methods* focus on usability and use ad-hoc approaches that mimic the repetitive structure of plants. An example is the X-Frog [Lintermann and Deussen 1999], which uses interactive recursive blocks to generate complex plants. Other methods use biological rules to control the plant's growth, such as the TreeSketch [Longay et al. 2012]. Sketch-based method use human input to define features of vegetation ranging from plants [Anastacio et al. 2006] to trees [Okabe et al. 2007; Tan et al. 2008].

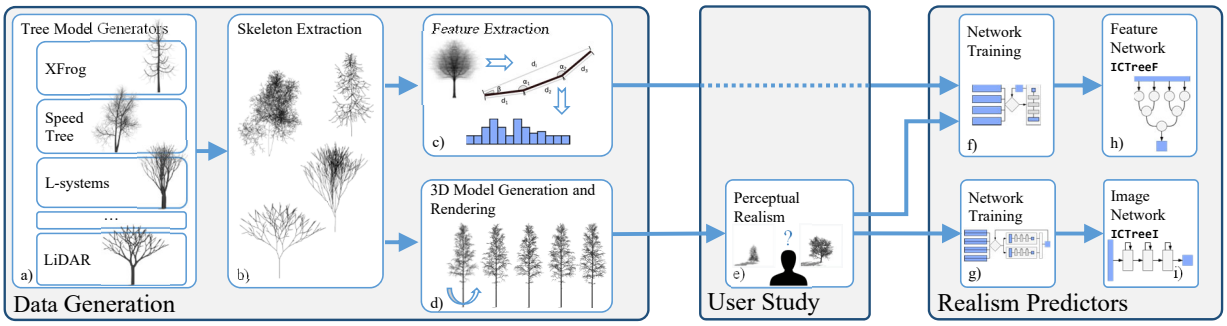

Fig. 2. **Overview:** Data from different sources are processed and unified by extracting the tree skeleton. Each plant is then described by its unique list of features. Simultaneously, 3D plant geometric models are generated and rendered from fixed views and lighting conditions. The images are then ranked in a user study that assigns perceived visual realism to each 3D model. The determined visual realism is then used to train two neural network classifiers. The `ICTreeF` works with the tree features, and the `ICTreeI` works directly with images. The two networks are then used to estimate the tree model's realism from its 3D information via the extracted features or images.

Plant reconstruction brings real trees to CG. Point clouds are commonly captured by Lidar technology, and many approaches attempt to reconstruct 3D tree geometry [Liu et al. 2021; Livny et al. 2011; Xu et al. 2007]. Small plant growth was reconstructed by using the approach of Li et al. [2013], and others attempt to find a plant skeleton [Du et al. 2019; Huang et al. 2013; Qin et al. 2020]. While point clouds are becoming common, most of the existing algorithms deal with reconstruction from photographs. Reche-Martinez et al. [2004] use images from various views to generate 3D tree reconstruction, and Neubert et al. [2007] use a similar approach combined with particle flow. Tan et al. [2007] and recently Li et al. [2021] use a single image to estimate the 3D structure of a tree. While the tree branches are most important for the overall tree appearance, foliage is a more complex problem. The work of Bradley et al. [2013] solves this by matching and fitting predefined leaves into point clouds.

To our best knowledge, no metric that evaluates the perceptual quality of CG tree models exists. Moreover, the perception of trees has not been studied systematically, and we are unaware of any curated dataset that could be used to understanding the perception of trees and the training of machine learning models.

3 METHOD OVERVIEW

Our method consists of the three main steps (Fig. 2): data generation and unification, user study, and realism predictor training.

Data generation: We seek to provide a perceived realism descriptor that will work with various methods and models. However, there are many highly different algorithms for vegetation generation (see Sect. 2), as well as vegetation acquisition methods and technologies (see Tab. 1). A common way to represent trees is by using their skeletons [Du et al. 2019] and assuming a unique algorithm that generates 3D geometry from the skeleton. This representation is acceptable for trees that have not been affected by the environment or otherwise modified, and it has been commonly used in many CG approaches (*e.g.,* [Pirk et al. 2012; Stava et al. 2014; Wang et al. 2018; Zhao and Barbič 2013]). We follow this approach, and in the first stage of our pipeline, we convert input data from various sources (polygons, point clouds, procedural models) into skeletons.

A logical choice of data classification is to use biological tree species. However, CG applications and growth models do not confine to tree species and often generate trees that cannot be assigned to a particular genus. Tree geometry in CG, especially in interactive modeling, is often driven by the artist's intuition and controlled by higher-level features, such as branching angle, width ratio, or branch length. We have retrieved a large dataset of 3D tree models, but many of them were generated by interactive applications and growth models that are species oblivious. Instead of guessing the tree species, we focus on the perceived effect of individual features shared across the species. A purely procedural tree can also be visually plausible. The key questions of our approach are "Does the tree look realistic?" and "What makes it look realistic?".

We extract a large feature vector from the tree geometries that provide rich information about the tree height, branching, angles, width, approximate envelope, topology, etc. (see Tab. 2). The feature vector captures a vast amount of tree properties yet is simple to generate from any existing method and can be easily incorporated into existing algorithms. The feature vector is used as a descriptor of the tree's perceived visual realism.

We also generate 3D geometry for each tree and render it in a controlled environment with a fixed light and camera position. We rotate the tree around its central axis (indicated by the arrow in Fig. 2d) to capture it from different angles as detailed in Sect. 4.4.

User Study: To estimate the perceived visual realism, we conducted a large perceptual two alternative force choice (2AFC) on 100 3D tree models, each rendered in five different views (500 images). Over 4,000 human subjects were shown pairs of images and were asked to select the more realistic one. The results provide a ranking of the tree models and views based on the perceived realism.

Realism Predictors: The tree ranking from the user study and the features of the 3D models were used to train a feature-based predictor `ICTreeF` that estimates a tree's perceived realism from its features. Also, the image-based predictor `ICTreeI` is trained on the generated and augmented images (Fig. 2d) and the ranking from the user study. Because the tree geometry is available, we augmented

the image views to 8,500 images used for training. Both models were thoroughly evaluated using a cross-validation protocol.

Finally, we identify features essential for the tree model's perceived realism. The realism is quantified using an auxiliary regression forest model trained on feature-realism pairs. Next, we generate a large dataset of 3D tree models spanning the entire parameter space of the biologically-based tree generator [Stava et al. 2014]. Finally, we select the trees which vary in identified important features and run a second user study to validate and justify our findings.

4 DATA ACQUISITION AND GENERATION

We gathered and made publicly available three datasets, each accompanied by the perceived realism scores. 1) Tree geometries, skeletons, and features of 100 manually curated 3D tree models. 2) Tree views: each tree was rendered from five views, providing 500 images. 3) Training data: views were augmented by random, giving a total of 8, 500 training images.

4.1 Tree Models

Tree geometry changes considerably through tree lifetime. On the one hand, young trees do not have clearly established features, they vary significantly, and they may be difficult to discern even for the trained arborist. On the other hand, old trees have often undergone heavy environment-induced variations. For example, they lose large branches that leave empty spaces re-filled by a young canopy.

We have performed an initial user study (Sect. 4.4) to gather further insight into the ranges of suitable parameters. We have determined that young and old trees and environmental effects skew the users' perceptional evaluation. For these reasons, we focus on mid-age trees that are old enough to be discernible yet not too old to be heavily affected by the environment. From the original dataset of 110 usable trees, we have removed ten trees (four young and six old) with seemingly extreme age or environmental influences, such as heavy leaning due to phototropism, wind, or gravity (Fig. 3). We also prioritized getting a fair split between various species and broader categories. We have manually selected 100 trees from the available sources (see Tab. 1), including 65 deciduous trees, 29 coniferous trees, four palms, and two shrubs.

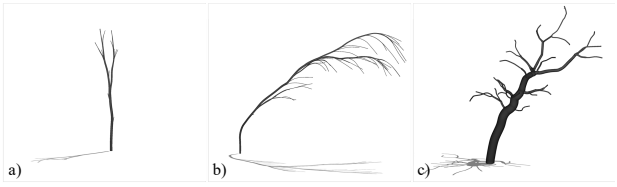

Fig. 3. **Excluded Tree Models:** Young trees do not contain enough branches (a), trees with environmental influence or reconstruction errors (b), and old trees (c). Trees in this figure have been scaled for visibility.

There is no database of a wide variety of tree 3D geometries, and our paper provides one (Tab. 1). Some models were scans of real trees represented as point clouds, but scanning trees does not capture small branches. Therefore, we included only point clouds with minimal noise caused by the small branches, resulting in accurate reconstructions without the minor branches. Others were from simulations, interactive models, and skeleton databases. They were either stored as meshes or tree skeletons. The models have considerable differences: some were highly detailed with many segments per branch or detailed triangulation, while others were simplified. They only recorded the general shape of the plant. The skeletal representations also suffered from varying sampling rates. We also used several software packages to generate some trees that were not represented well, balancing their utilization to prevent dataset bias. Each tree was manually inspected for errors.

Data Source	#	Data Type	Reference
ModelNet	4	R (Δ)	Wu et al. [2015]
LaserScan	6	R (PC)	Sloup et al. [2013]
TreeParts	8	R (Δ, S), A (Δ)	Xie et al. [2016]
ReconTree	2	R (PC), A(Δ)	Livny et al. [2010]
AdTree	2	R (PC), A(Δ)	Du et al. [2019]
SpeedTree	12	A (S)	SpeedTree [2021]
XFrog	25	A (S)	Lintermann and Deussen [1999]
Shape Space	14	A (Δ, S)	Wang et al. [2018]
InvTree	22	R (Δ, S), A (Δ, S)	Stava et al. [2014]
ShapeNet	1	A (Δ)	Chang et al. [2015]
Multiple Views	4	R (Δ)	Guo et al. [2018]

Table 1. Sources of Real (R) or Synthetic (A) tree geometry data. The data were either Triangular Meshes (Δ), Point Clouds (PC), or Skeletons (S).

4.2 Skeleton Extraction

Skeletal representation allows for tree comparison and feature extraction. The skeleton is an oriented tree graph from the trunk to the end branches, each node carrying information about the position, branch width, and connectivity to children and parents. Skeletons are further transformed into a detailed 3D geometry by the reconstruction algorithm (Sect. 4.4).

The skeletonization algorithm automatically places branching points and estimated internode end-points, extracting tree skeleton from triangular meshes and point clouds, and it is based on methods [Du et al. 2019; Lin et al. 2020; Tagliasacchi et al. 2012]. We adopted the AdTree skeleton extraction algorithm [Du et al. 2019] by adding an automated pre-sampling of a polygonal model and utilizing its connectivity information.

If the branch width is unknown (*e.g.*, if the input data was already a skeleton), it is calculated by the Da Vinci formula [Prusinkiewicz and Lindenmayer 1990, page 57]

$$w^D = \sum_{i=0}^{n-1} w_i^D, \tag{1}$$

where w is the cross-sectional area of the parent branch, and w_i are its siblings. The coefficient $D = 2$ was used in the original formula but can vary for some trees $1.8 \leq D \leq 2.3$ as studied by Minamino and Tateno [2014]. The widths are calculated recursively from the leaves of the mathematical tree graph, and it is assumed that the smallest branches have a fixed cross-sectional area.

4.3 Feature Generation

Understanding tree shape is difficult, especially for procedural models that may have no basis in biological growth. Previous work has identified and used common features that stem from biology and horticulture [De Reffye et al. 1988], and they have been used for natural tree comparison [Stava et al. 2014]. Although these features are used to identify and classify trees, their effect on the perceived visual realism is unknown. We classify the tree features into local and global, depending on the part of the tree they cover (Tab. 2).

Local Features (Fig. 4) are based on statistics representing topological and geometrical attributes of the tree geometry. Most of the local features are represented by their mean μ, variance σ^2, and the ratio of min/max x_m, x_M, along with a histogram.

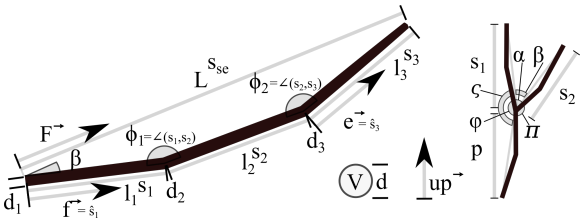

Fig. 4. **Local features** are extracted from the tree skeleton. A single chain (left) is a sequence of segments between two branching points or root/leaf nodes. Relations surrounding a branching point (right) contain a parent chain (**p**) and two or more children/siblings ($\mathbf{s_1}$, $\mathbf{s_2}$).

The tree *segment* is an oriented edge between two consecutive nodes within the skeleton, and it roughly corresponds to an internode. We collect segment's width d and cone frustum volume V.

A *chain* c_i is a sequence of segments that start from either a branching node or the root and end with the consecutive branching node or a leaf node. Each chain has a single unique parent chain, except for the root chain and a set of child chains.

We store each chain's number of segments S, the number of chains from the root node (depth) h, length $l = \sum_{j=1}^{S} \|\mathbf{s_j}\|_2$, total length L as $\|s_{se}\|_2$, deformation measured as the sum of the angles between successive segments $\sigma = \sum_{j=2}^{S} \angle(s_{j-1}, s_j)$, straightness that measures the geodesic length divided by the spatial distance $\tau = L/l$, chains' slope expressing its upward tendency $\upsilon = \angle(f^{\rightarrow}, up^{\rightarrow})$, minimum and maximum width d_m, d_M, and tapering corresponding to how much the chain shrinks $\Delta d = d_m/d_M$. Further, we also include the number of internodes $I = l/j_\mu$, where j_μ is the estimated internode length (see the global features below). Finally, the chain tropism values are calculated as $\zeta_b = \angle(f^{\rightarrow}, e^{\rightarrow})$, $\zeta_h = \angle(f^{\rightarrow}, up^{\rightarrow})$ and $\zeta_p = \angle(e^{\rightarrow}, F^{\rightarrow})$. The tropism values approximate the response of the tree to various environmental factors.

Siblings are chains sharing a specific parent chain c_p, and they correspond to branches starting in the same node. For each pair of sibling chains c_{i_1} and c_{i_2}, we collect their angle of incidence α as the angle between their first segments $f^{\rightarrow}$, and the total angle β as the angle between their last segments $e^{\rightarrow}$, which corresponds to the concurrency of their growth. Further, we aggregate the individual angles α and β over all siblings into A and B, respectively.

Parent features are calculated at each branching point. Let us denote the parent chain c_p and its child chains $c_{c_p j}$. For each parent-child pair (c_p, c_c), we collect the length ratio $\Delta l = l_{c_c}/l_{c_p}$, the difference of angle sums $\Delta\phi = \sum_i^{c_p} \phi_i - \sum_i^{c_c} \phi_i$, continuation coefficient representing how well the child chain holds its parent's direction $\pi = \angle(e^{\rightarrow}_{c_p}, f^{\rightarrow}_{c_c})$, and preservation coefficient reflecting the ongoing direction of growth $\varsigma = \angle(F^{\rightarrow}_{c_p}, F^{\rightarrow}_{c_c})$. We also aggregate the continuation and preservation coefficients on a per-parent basis, resulting in their minimum and maximum values $\pi_{m|M}$ and $\varsigma_{m|M}$.

The *branching* features capture the topological properties of a branching point. Asymmetry κ represents the differences in pairs of sub-trees (t_1, t_2) stemming from a given branching point

$$\begin{aligned} \kappa_e &= \chi_e \left(|\text{leaves}_{t_1}| - |\text{leaves}_{t_2}| \right), \\ \kappa_l &= \chi_l \left(\sum_c^{t_1} l_c - \sum_c^{t_2} l_c \right), \\ \kappa_v &= \chi_v \left(\sum_s^{t_1} V_s - \sum_s^{t_2} V_s \right), \end{aligned}$$

where $\chi_{e|l|v}$ are normalization constants calculated at each branching point. Branching feature λ represents how well a given branching point matches a branching model: *Monopodial*, *Sympodial Monochasial*, and *Sympodial Dichasial* (Fig. 5). We take the base model as defined by De Reffye et al. [1988] and calculate a normalized score. First, the primary and the secondary child branches are estimated by weighing their size as the number of leaf nodes present in each sub-tree. Next, the corresponding model requirements are taken into account – symmetric bifurcation, sub-tree axis, and sub-chain continuation. The final normalized score for each model is determined by a fitness coefficient in each requirement category. Similarly, the fitness for each ramification model μ from De Reffye et al. [1988] (Fig. 5) – *Continuous*, *Rhythmic*, and *Diffuse* – is calculated. We use the estimated internode length in order to determine how regular the ramification process is for a given tree.

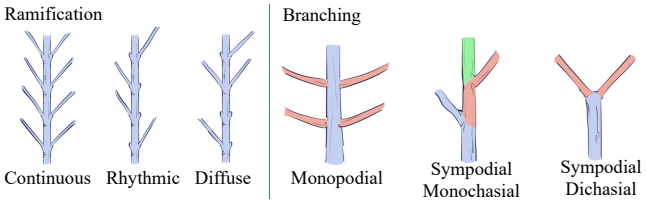

Fig. 5. **Tree Typology:** Ramification defines the branching pattern. *Continuous* ramification develops lateral branches at all axillary meristem points, *rhythmic* shows regular patterns, and *diffuse* is irregular. Branching type describes how higher-order branches form. *Monopodial* branching keeps its apical axis; *sympodial* trees terminate the apex and branch out.

Global features are calculated for the complete tree model. While the local features are stored as histograms, each global feature is represented by a single value. First, we estimate the internode length i_μ by calculating the mean chain length, resulting in a variance value i_{σ^2}. The tree trunk is the longest chain stemming from the root node with length l_t. We collect the number of (topological) leaf nodes c_t. The tree age is estimated for the trunk age_t and the

Local			Local			Local		
Type	Name	Symbol	Type	Name	Symbol	Type	Name	Symbol
Segment	Width	d	Chain	Tapering	Δd	Parent	Preservation	$\varsigma_{\|m\|M}$
	Volume	V		Internodes	I	Branching	Asymmetry	$\kappa_{e\|l\|v}$
Chain	Segments	S		Tropism	$\zeta_{b\|h\|p}$		Branching	$\lambda_{m\|s\|d}$
	Depth	h	Sibling	Angle	α		Ramification	$\mu_{c\|r\|d}$
	Length	l		Total Angle	β	Global		
	Total Length	L		All Angle	$A_{m\|M}$	Internode	Length	$i_{\mu\|\sigma^2}$
	Deformation	σ		Tot. All Angle	$B_{m\|M}$	Trunk	Length	l_t
	Straightness	τ	Parent	Length Ratio	Δl	Leaf	Count	c_l
	Slope	υ		Angle Sum	$\Delta\phi$	Tree	Age	age
	Width	$d_{m\|M}$		Continuation	$\pi_{\|m\|M}$		Volume Imprint	$volume$

Table 2. **Tree Features:** List of features extracted from each tree. The symbols $\alpha_{|m|M}$ represent the feature (α), its minimum (α_m), and maximum (α_M).

crown age_c separately, resulting in the final $age = age_t + age_c$. The trunk age is $age_t = l_t / i_\mu$, while the crown age is deduced from the maximal chain depth as $age_c = \max_i \{h_i\} - 1$. An important global feature is the tree area, represented by its volume imprint (Fig. 6) that is generated by casting rays through the reconstructed tree model (Sect. 4.4) and accumulating volume contained within the model for each ray. To get a rotation-invariant representation of the complete model, we aggregate the accumulated values over 360^o of the rotation of the base model. We have 40 local features with 960 values and eight global features with 270 values.

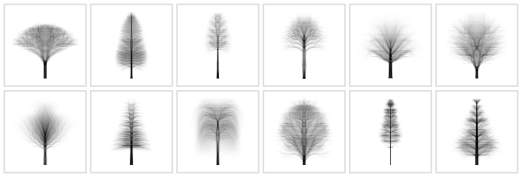

Fig. 6. **Volume imprint** is a global visual feature generated by casting rays through rotating reconstruction of the tree model and recording volumetric information along the rays. The grid contains a visualization of randomly selected imprints from our dataset.

4.4 3D Model Generation and Rendering

The final step in our data generation pipeline is the 3D model generation and rendering (Fig. 7). The input is the refined skeleton (Sect. 4.2), and the output is the 3D mesh of the tree, including detailed branch geometry without leaves. The mesh is generated on the GPU directly using tessellation and geometry shaders, extending the approach of Stava et al. [2014]. The generation algorithm utilizes the skeleton data, including topological information, branch radii, and rotation minimizing orthonormal bases [Wang et al. 2008]. Hermite interpolation is used to calculate smooth positions from the skeleton nodes. Branches are then generated as generalized cylinders with sweeping circles with the appropriate diameters.

The generated mesh is then rendered. Previous studies determined that lighting and geometric artifacts profoundly impact perceived realism [Pan et al. 2005; Rogowitz and Rushmeier 2001]. To eliminate bias caused by varying colors, we use grayscale shading and a

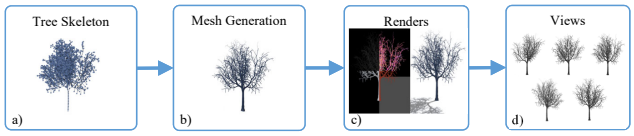

Fig. 7. **Mesh Generation and Rendering:** Finalized tree skeleton (a) is used to generate triangular mesh on the GPU (b). The tree is rendered in various modes (c), and the views are stored (d).

shadow from a parallel light source (Fig. 8). The lighting conditions are the same for all models. We experimented with different shading, and the initial pilot study we performed directed the choice of colors and shading. The pilot study was conceptually the same as the study in Sect. 5. We interviewed 13 test subjects who had no previous experience with tree modeling algorithms or modeling in general. We showed them various scenarios, including multiple visual and tree styles such as black/white shading, fully textured models with backgrounds, and real tree photographs with a natural background. Tree styles included various species both with and without leaves and different scaling options. We were in direct verbal contact with the subjects to better understand their reasoning.

Fig. 8. **Shaded Tree Views:** Cropped views generated by rotating the tree.

Overall, the users considered trees with extreme age differences incomparable, and strong environmental effects were perceived as unrealistic. Further, we were able to consistently mask unrealistic branching structures by a suitable parameterization of the procedural foliage, leading to biased results, supporting the choice to omit them. Finally, realistic shading made the users rate the visualization instead of the tree geometry. Thus, we arrived at our experiments' final form that focuses on clarity to allow the users to focus on the important branching structures in our trees. Moreover, the final

shading can be easily achieved, making ICTreeI usable for new algorithms or applications.

Camera properties were chosen to correspond to a human observer of the average height of 1.8m. Tree models were automatically scaled based on the available estimate of internode length and then manually verified.

We generate n views of the tree by rotating it around the y-axis $n\times$. When the same algorithm is used for deep learning data generation, we randomly jitter the camera yaw, pitch, and roll angles and add a small random delta to its position.

5 USER STUDY

We performed a comparative user study to acquire the ground truth perceptual data using the Amazon Mechanical Turk. We showed a pair of images, asked the question, "Which tree looks more realistic?" and the participants were required to choose one of them (Fig. 9).

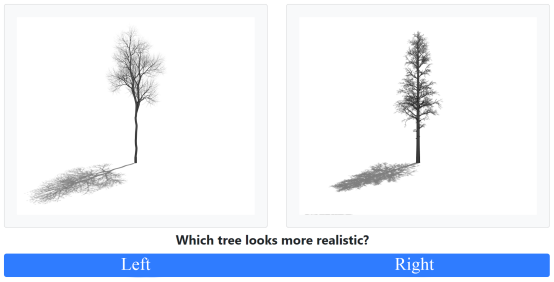

Fig. 9. **User Study:** Example of the pairwise experiment presented to the user utilizing the 2AFC method.

5.1 Tree Image Data

Our dataset contains different trees covering a wide range of attributes, and it provides solid ground base data.

To force the user's focus on the geometry features of the given tree [Pan et al. 2005], we use the soft shading approach described in Sect. 4.4. A tree model is a 3D structure, which is notoriously difficult to evaluate in perceptual studies [Rogowitz and Rushmeier 2001]. Providing users with an interactive 3D model to score, while certainly within the bounds of possibility, is problematic and could lead to users focusing on details. To alleviate this problem, we render each tree from five uniformly distributed views by rotating the tree around its trunk while keeping the scene and lights fixed to keep the shadow and lighting effects the same [Rogowitz and Rushmeier 2001]. We scale the tree model proportionally to its estimated real size to fit the whole range of trees into one camera view to keep them easily comparable in a side-by-side manner. The camera's height is placed at ground level, around 1.8m above ground, to simulate how people usually observe trees in their daily lives. The distance was chosen so that all trees fit into view.

5.2 Experiment Design

We generated the set of all view pair combinations $(c_t\ c_v)((c_t - 1)\ c_v)/2$, where $c_t = 100$ is the number of trees and $c_v = 5$ views per tree, resulting in 123,750 comparison pairs. Since we could make no assumptions about the pairwise comparison's transitivity given two tree views, it was necessary to include all pairs. We randomized the sides so that the distribution of trees being on the left or right for each view is uniform. Further, we divide these pairs into 495 equally-sized batches of 250, ensuring each tree is fairly represented in each batch. To get unambiguous results, multiple users should review each pair, so we repeated the batch generation 10×, leading to the total number of 4,950 unique batches.

The perceptual study was completed using the Amazon Mechanical Turk, utilizing "Turk Masters" – verified and reliable participants. We adopt the two-alternative forced-choice (2AFC) methodology [David 1988], commonly used for subjective judgment and preference experiments (Fig. 9). To further enhance the results' stability, each user could respond only to a single batch of experiments.

5.3 Experiment Evaluation and Outcomes

After completion of the experiment, we removed dubious results, such as users answering in patterns or taking too short/long time to complete the experiment. After cleaning, we received a total of 1,041,000 pairwise choices from 4,164 unique experiment participants – counted by their unique MTurk identifiers – from which 50.86% were female, and 49.14% were male, aged between 18 and 60+, 73.85% between 20 and 40 years old. This corresponds to 8.41× coverage of each comparison in the dataset.

We processed the pairwise comparison data using the approach presented in [Perez-Ortiz and Mantiuk 2017], resulting in per-tree perceptual realism scores (Fig. 10). The perceptual score uses the just-objectionable-difference (JOD) unit, where the difference of one JOD indicates that 75% of observers selected one tree as more realistic than the other. Please see the supplementary material for a full list of all trees in the dataset, ordered by their JOD score.

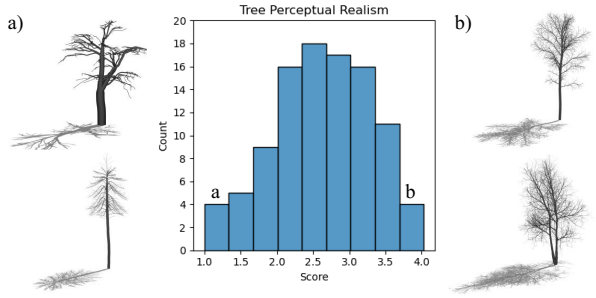

Fig. 10. **Tree Perceptual Realism:** Histogram of tree scores from the pairwise comparison data using the approach based on [Perez-Ortiz and Mantiuk 2017]. Two of the worst (a) and best (b) perceived trees are displayed.

The trees with bare geometries and unnatural angles rated among the lowest (Fig. 10a)), while well-developed tree models ranked high (Fig. 10b)). Most of the users preferred trees with large amounts of branches. However, as shown in Fig. 14d), thick growth leads to less realistic trees, and we further study the effect of thickness in Sect. 7.5 and Fig. 20. Palms were some of the least favorite trees in our dataset, probably because palms were rendered without leaves. Coniferous and deciduous trees are otherwise evenly mixed, with shrubs ranking high.

The results of this study include 100 3D geometric models (skeletons and meshes) with the perceived realism scores, 500 views, each with the perceived scores, and they are available on the project website **cphoto.fit.vutbr.cz/ictree**.

6 PERCEPTUAL REALISM PREDICTORS

To gain in-depth insight into what makes trees look real and allow their automated realism assessment, we introduce two no-reference metrics and an additional regressor for importance analysis. The `ICTreeF` predictor uses features, and the `ICTreeI` predictor estimates perceived realism solely based on an image of the target tree. We utilize the Decision Tree ensemble [Breiman 2001] to analyze which tree properties affect the perceived realism.

6.1 Feature-Based Predictor – `ICTreeF`

`ICTreeF` predictor is used for regression of the tree perceptual realism using features extracted from the input tree skeleton (Sect. 4.3). The prediction model (Fig. 11, left) consists of a sequence of tapering, fully connected layers. Each layer is followed by a ReLU activation except the last one, which uses Leaky ReLU [Maas et al. 2013] with a slope of $\delta = 0.1$.

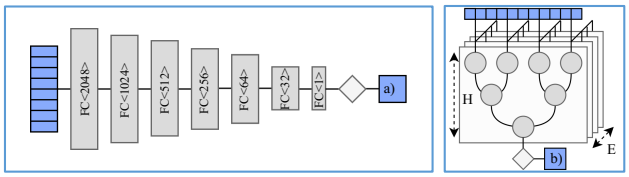

Fig. 11. **`ICTreeF` Model (left):** The feature-based realism predictor consists of a shrinking sequence of fully connected layers, each followed by a ReLU activation. Finally, the last layer utilizes a Leaky ReLU activation and outputs the predicted score value (a). **Feature Importance Model (right):** analysis is performed by fitting a Random Forest Regressor ensemble, using MSE criterion to train against the ground-truth tree scores (b).

Model fitting uses the ground-truth perceptual user scores and the full vector of features collected from each tree. We perform tree model augmentation to expand the data by creating 16 tree model variants for each of the 100 trees in the dataset. They are generated by randomly jittering the tree node location by $\leq 1\%$ of the branch length around their original position. We use the original realism scores combined with recalculated feature vectors. This augmentation allows us to expand our dataset by a factor of 17× (from 100 trees to 1,700 trees), hardening the models to small changes in the feature space. Further, we use both score types, randomly pairing each feature vector with either the tree score or any of the view scores. The dataset is then split into 80 : 20% for training and testing, randomizing the batches for each training run, ensuring all tree variants are restricted to a single split. We optimized the model using the AMSGrad variant of the ADAM method [Kingma and Ba 2017; Reddi et al. 2019], running it for 2,000 epochs with a mini-batch size of 64. We use the Mean Squared Error (MSE) function $\sum(\hat{y_i} - y_i)^2$ as the loss function. Learning rate is initially set to the recommended value of $\alpha = 0.002$ [Kingma and Ba 2017], reducing it by a factor of ten each time the loss function does not improve by at least 0.01% in the last ten epochs, *i.e.,* using the reduce-on-plateau technique.

We also experimented with other predictors, but they were less effective at predicting realism. We have also applied the multi-staged pre-training regime used for the image-based predictor (Sect. 6.2), which lead to considerable improvements in the generalization performance of the final model. See the ablation experiments in Sect. 7.4.

6.2 Image-Based Predictor – `ICTreeI`

`ICTreeI` predicts the perceptual score from a rendered tree image (Sect. 4.4). The prediction model (Fig. 12) is based on the *Res2Net* [Gao et al. 2019] backbone architecture. We prioritized model size and inference speed and chose the *Res2Net50* variant, utilizing four stages with 3, 4, 6, and 3 multi-scale residual blocks [Gao et al. 2019]. We modified the *Res2Net* architecture by removing layers from the last average pooling layer onward, cutting the network after the last residual addition and ReLU activation. The predictor processes the input image, gradually down-sampling it through the network by halving its resolution after every stage. The residual blocks perform analysis on multiple scales simultaneously within each stage, aggregating the resulting activations into the final residual signal.

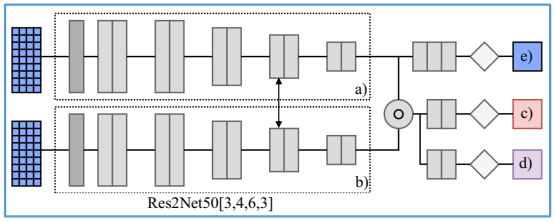

Fig. 12. **`ICTreeI` model:** The image-based realism predictor is based on the *Res2Net* model, utilizing a Siamese architecture, where upper (a) and lower (b) branches share weights. The model is first pre-trained on binary (c) and differential (d) data. Outputs of a single branch are used as features to the score prediction network (e). Only a single branch is used at inference time, along with the score predictor (e).

`ICTreeI` uses the ground-truth perceptual scores from the user study combined with shaded view images of the tree. The user study provided 500 base images with scores that were augmented by the following strategy. First, we generated 16 additional views for each base image by randomly jittering the camera position, resulting in 85 images for each tree. The scores for the new views were obtained by interpolation from the known data. We transformed the views into spherical coordinates, normalizing their radius. Next, we placed interpolation knots with the ground-truth user scores into a grid. Finally, we interpolated the missing values using a radial basis function combined with the great-circle distance to achieve a smooth and border-continuous mapping. We utilize both score types, placing the knots representing the tree scores into the top and bottom positions. The resulting training set contains 8,500 images, each with their ground-truth or interpolated scores.

Next, the rendered images (Fig. 8) were scaled, cropped, and normalized to cover a 256×256 image raster. Then a random crop of 224×224 pixels was performed. The dataset was split into 80 : 20%

training and test sets, randomizing the selection for each training run. The split was made along the tree exemplars, always preserving split boundaries concerning trees and their view variants. `ICTreeI` works only with a single rendered image leading to a challenging optimization problem. Therefore, we perform the optimization in three stages: *binary*, *differential*, and *score*. The three stages go from easier to more difficult tasks, starting with the binary selection of the more realistic tree, estimating the score difference between the two trees, to finally estimating the score itself from a single image.

We started with a *Res2Net50* network pre-trained on the ImageNet [Krizhevsky et al. 2012] dataset. We transformed the model into a siamese architecture, inspired by multi-view shape recognition [Su et al. 2015], by using the same network in a two-branch configuration. Each branch uses the *Res2Net50* architecture, sharing their weights. We also experimented with different backbone architectures (*e.g., ResNet18*, *Res2Net18*) and randomly initialized weights, all of which lead to inferior prediction performance. For details, see the ablation experiments in Sect. 7.4.

Binary optimization utilizes the binary choices from the user study. We divide them according to our initial 80 : 20% split, making sure trees are always limited to precisely one of the splits. The siamese network was then modified by concatenating both branches' outputs, adding average pooling, and a fully connected layer combined with a sigmoid activation function. We optimized the augmented model using the AMSGrad variant of the ADAM method [Kingma and Ba 2017; Reddi et al. 2019] for 50 epochs, with a mini-batch size of 32. The learning rate was set to the initial value of $\alpha = 0.0001$, using the reduce-on-plateau technique. We used the Binary Cross-Entropy loss. View variants are chosen randomly with a uniform distribution.

Differential optimization also utilizes pairwise data. However, the ground-truth is given by $y_1 - y_2$, where y_1, y_2 are the JOD scores in a given comparison pair. The view variants were chosen randomly, each associated with its respective score. Again, the siamese architecture is used as a base, continuing with the weights from the *binary* stage. We added an average pooling and a fully connected layer after the concatenation, combined with a ReLU activation. We optimized this model using the same optimization procedure as in the *Binary* case, except for using the MSE loss function instead of the Binary Cross-Entropy.

Finally, the *score* optimization stage uses only a single branch, *i.e.,* a single instance of the *Res2Net* model, along with the perceptual tree scores. We initialized the models' weights from the previous optimization stage, essentially gaining a pre-trained feature extraction network. Next, we added the average pooling layer followed by two fully connected layers with $1,000$ and 1 units, respectively, using the ReLU nonlinearity after the second fully connected layer. Optimization was performed using the AMSGrad variant of ADAM, running for 500 epochs with a mini-batch size of 32. We use MSE as the loss function, along with the reduce-on-plateau technique.

6.3 Feature Analysis

We analyzed features (Sect. 4.3) to gain deeper knowledge about the critical properties of trees concerning their perceived realism. We take inspiration from Čadík et al. [2013] and use ensembles of Decision Trees, *i.e.,* Random Forest Regressor model (Fig. 11 right). We set the number of trees $E = 1,000$ and reach depths in the range of $H = [9, 12]$, without setting a hard limit.

We divide the dataset into train/test sets in an 80 : 20% split. We train each ensemble on the full selection of our features and corresponding tree scores, using the MSE optimization criterion. We repeat the training 30 times, each with a different, randomly selected data split to get stable and reliable results. The importance analysis is performed by first calculating the normalized *Gini importance* [Hapfelmeier and Ulm 2013]

$$i_f = 1/n_s^{root} \sum \left(n_s^p n_i^p - n_s^l n_i^l - n_s^r n_i^r \right)$$

for each feature f, where n_s^x is the number of samples reaching node x, n_i^x is the Gini impurity for node x, $root$ is the root node, and p, l, r are the parent, left child and right child nodes. The final value i_f is calculated in each ensemble, followed by calculating a mean I_f and variance σ_f^2 over all ensembles.

Using the Gini method to quantify feature importance for high dimensional data may lead to a bias [Altmann et al. 2010; Boulesteix et al. 2012]. We performed an additional pass over all of the trained ensembles using the *Permutation importance* introduced by Altmann et al. [2010]. The relative ordering provided by both the Gini importance and the Permutation importance procedures were consistent for all the presented results.

7 IMPLEMENTATION AND RESULTS

Our neural-network-based predictors were implemented in *Python* with the *PyTorch* framework, accelerated using the *CUDA* backend. The training and inference measurements were performed on a desktop computer with AMD Ryzen 5 3600 processor, 16GB RAM, and NVIDIA GeForce RTX 3080 10GB GPU. Tab. 3 shows the timing of training, inference, and feature calculation for `ICTreeF`.

Network\Task	Training	Inference	Featurization
`ICTreeI`	36 [h]	200-300 [ms]	NA
`ICTreeF`	10-12 [min]	5-10 [ms]	60 [ms]

Table 3. Timing of the training and inference

We performed a correlation analysis to show the effectivity and fidelity of presented `ICTreeF` and `ICTreeI` metrics. Next, we analyze feature importance and verify its impact on perceptual realism in a second user experiment. We present the results of ablation studies to quantify the performance of our realism prediction models. Further, to show that our solution appropriately generalizes to other types of trees, we analyze its results on a large tree model dataset sub-spaces. Finally, we present possible applications of our metrics.

7.1 Dataset Analysis

We perform several statistical tests on the `ICTree` dataset and provide quantitative evidence of its significance to confirm the validity of our study.

Fig. 13b) shows the optimized scores and the corresponding view scores. We normalized the scores so that the minimum value is zero,

and we sort them from the minimal to the maximal value. The view scores copy the scores of complete trees closely. The Confidence intervals of the pairwise comparison method [Perez-Ortiz and Mantiuk 2017] are within ± 4.47% of the final optimized scores, ± 1.59% for tree scores, and ± 5.04% for view scores. For further analysis, see the supplementary material. The outliers correspond to trees that have favorable or unfavorable viewing angles. We calculated the voter agreement using Percentage Agreement $ag \approx 0.797$ and Fleiss' Kappa $\kappa \approx 0.740$ with respect to the ordering given by the optimized scores. These results show substantial voter agreement [Kundel and Polansky 2003], hinting at the subjective nature of perceived realism.

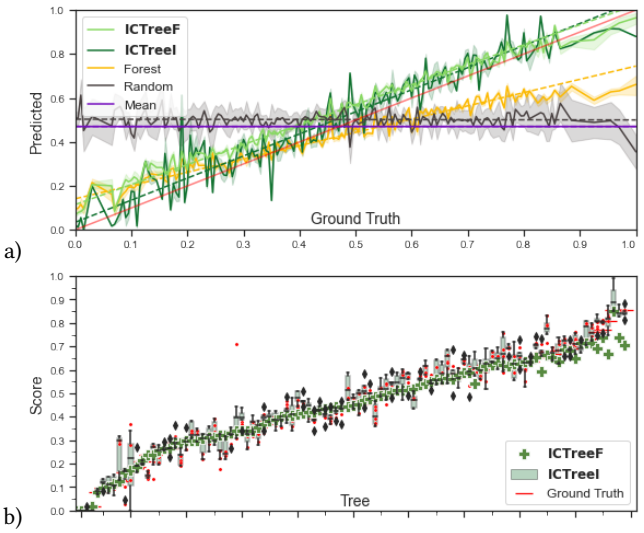

Fig. 13. **Cross-Validation:** Evaluation of realism predictors. Correlation graph (a) compares results of `ICTreeF`, `ICTreeI`, Random Forest, Random, and Constant predictors, corroborated by Pearson correlation scores of 0.842, 0.852, 0.769, 0.12, and 0.08 respectively. Predicted realism divergence plot (b) displays the ground truth regarding the predictions made by `ICTreeF` for each tree and `ICTreeI` for each tree view. The bottom graph is sorted by the increasing score.

7.2 Perceived Realism Predictors

Our predictors `ICTreeF` and `ICTreeI` estimate perceptual tree realism as close as possible to the ground-truth measured in the user study (Sect. 5). We performed eight cross-validation runs to quantify the prediction accuracy – each model was trained on 80 trees and tested on 20 trees – covering the complete dataset. The results (Fig. 13a) show a high correlation between the ground-truth and predicted realism for both predictors. We quantify this using MSE, Pearson correlation coefficient cor_p, and Spearman correlation coefficient cor_s. Both `ICTreeF` and `ICTreeI` achieve high scores: $cor_p \approx 0.842$, $cor_s \approx 0.836$ and $cor_p \approx 0.852$, $cor_s \approx 0.838$ respectively. The results are much higher than a chance modeled by random predictor ($cor_p \approx 0.119$, $cor_s \approx 0.086$) and constant (mean) predictor ($cor_p \approx 0.081$, $cor_s \approx 0.005$).

Fig. 13b) shows the variance in results of image-based predictor `ICTreeI` given different views of the same tree. In general, the scores predicted by `ICTreeI` fluctuate around the results of `ICTreeF`. We believe this indicates that similar information could be learned from tree features and images, possibly sharing some deeper structure.

As qualitative results, we show samples of the worst and the best-perceived trees, accompanied with their ground-truth and predicted realism scores in Fig. 14. Furthermore, we validated both predictors in the second user study in the next section.

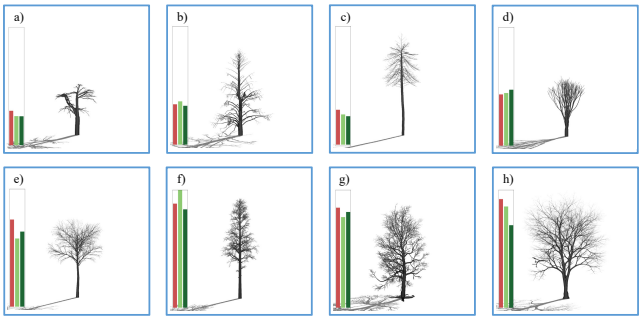

Fig. 14. **Scored Trees:** Examples of trees from the dataset sorted by their ground-truth perceptual realism (left to right). Each image shows ground-truth realism (in red) and predictions by `ICTreeF` (light green), `ICTreeI` (dark green) for the given image.

7.3 Analysis of Features

An analysis of the results of the user study data enables gaining deeper theoretical insight. Specifically, we were interested in the impact of tree features on the perceived tree realism. We trained an auxiliary random forest model on feature-realism pairs. The acquired feature importance distribution (Fig. 15) shows that the *sibling angle* and *segment volume* have the most noticeable impact on the perceived tree realism. On the other hand, *chain deformation* and *chain segments* have negligible influence. This is an expected intuitive result because the branching angles and length of an internode have been observed as the key allometric tree properties related to the internal flow of nutrients [West et al. 1999]. These features have also been used in CG literature for inverse procedural model parameter estimation [Stava et al. 2014]. It is unclear why chain deformation and chain segments have a small impact on perceived realism.

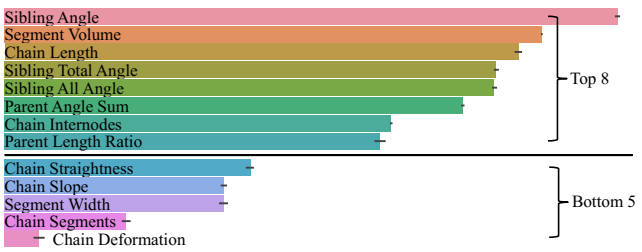

Fig. 15. **Feature Importance:** eight most important features (top) and five least important (bottom) are displayed. The $x-$axis is logarithmic and contains importance scores retrieved in feature analysis (Sect. 6.3).

To **validate** our findings experimentally, we constructed the Generated dataset consisting of 54,944 3D tree models spanning the entire parameter space of the biologically-based tree generator [Stava et al. 2014]. We select the trees that vary only in one specific feature shown in Fig. 16. According to our expectations, the most important features (*sibling angle* and *segment volume*, shown in the first two rows) significantly affect the perceived tree realism (high ground truth value, ICTreeI, and ICTreeF). On the other hand, the least essential features (*chain deformation* and *chain segments*, the last two rows) have minimal impact on perceived realism.

Presented novel ground-truth perceptual realism scores were collected in a further Importance study. This study had a setup similar to our first experiment (2AFC, Mechanical Turk, Sect. 5) but on a much smaller scale. The input stimuli were 25 images shown in Fig. 16. In total, 31 participants completed 6,000 pairwise responses, which were converted [Perez-Ortiz and Mantiuk 2017] to average perceptual scores (shown red in Fig. 16). As expected, the change in the first two features results in high variance of perceptual tree realism ($\sigma^2 = 0.308$ and $\sigma^2 = 0.203$). Conversely, changing the last two features has little impact on the tree realism ($\sigma^2 = 0.036$ and $\sigma^2 = 0.014$).

Finally, we use perceptual realism scores from the Importance study to create the Importance dataset containing the 25 tested trees. While the ICTree dataset contains trees from various sources (Tab. 1), the Importance dataset is fully procedural, and its distribution may be accordingly different. We use the Importance dataset to assess the generalization of our predictors. We train both ICTreeF and ICTreeI models using only the ICTree dataset. The feature-based predictor ICTreeF shows a high correlation with subjective scores: $cor_p \approx 0.803$ and $cor_s \approx 0.759$; ICTreeI reaches $cor_p \approx 0.782$ and $cor_s \approx 0.649$. These results indicate that ICTreeF is a robust predictor, compared to the ICTreeI model's lower generalization performance. This result is expected because the ICTreeF uses more information-rich 3D models, while the ICTreeI uses only images.

7.4 Ablation Experiments

We perform a series of ablation studies to quantify the performance of our realism prediction models. We run the same cross-validation experiments as in Sect. 7.2. The quantitative results are in Tab. 4, which includes MSE, Pearson correlation coefficient, and Spearman correlation coefficients on two datasets. The ICTree dataset represents the ICTree 100 tree dataset (Sect. 4), always split into 80 : 20 training and testing sets. Results on the ICTree dataset show how well the model performs when the testing data has a similar distribution to the training data. The second dataset Importance contains the 25 trees we performed our feature importance experimental verification on (Sect. 6.3). Conversely, results on this dataset indicate the degree of generalization a given model achieves. In total, we perform three types of ablation experiments: Basic Models, ICTreeF augmentations, and ICTreeI augmentations.

Basic models: We tested 11 techniques and chose the best performing as a benchmark for our methods. Complete results can be found in the supplementary material. Summarized results in Tab. 4 show comparatively good performance for models based on Deep Neural Networks, Random Forests, and Automatic Relevance Determination Regression. We also experimented with Least-Angle Regression methods, which resulted in negative correlation scores. However, relatively good results on the ICTree dataset are paired with bad results on the Importance dataset (Sect. 7.3). This gap indicates that the models work well when testing data is of a similar distribution to the training data (Tab. 1), thus trading generalization for better accuracy on data with a similar distribution.

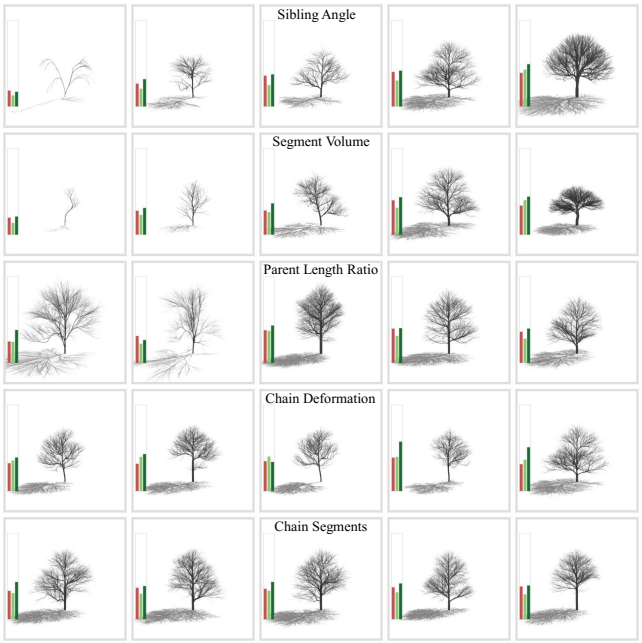

Fig. 16. **Experimental verification of feature importance:** trees varying in single feature value (horizontal axis) for five different features (vertical axis). Changes in the two most important features (first two rows) result in large changes in perceptual realism (the color bars vary from left to right). Changes in the two least important features (last two rows) have a negligible effect on realism (the color bars do not change significantly). The middle row shows a moderately important feature.

The ICTreeF model starts with the base architecture without any enhancements or data augmentation (ICTreeF_base Fig. 11, left). Prediction performance on the ICTree dataset improved by roughly 31% compared to the basic models. However, generalization to Importance data is limited ($cor_p = 0.375$, $cor_s = 0.195$). Utilizing pre-training on differential (ICTreeF_d) and binary (ICTreeF_db) data improves the generalization of the resulting model, enhancing the correlation coefficients by around 73%. This improvement is further amplified by using both tree and view scores in the training process (ICTreeF_dbs). An improvement of 13% in the MSE is also seen by using skeleton data augmentation (Sect. 6.1) and improves further with an increased number of augmented variants. We start by using only four variants (ICTreeF_dbsv4), including the base model and three additional augmented models. We see a small but significant improvement on both datasets (±12.6%). Increasing the number of variants to eight (ICTreeF_dbsv8) and the full 17 (ICTreeF_dbsv17) improves generalization even further by 13.4% in MSE and 18.4%, 32.5% in correlation coefficients.

Fig. 17. **Abstraction study:** A complex tree sketched by an artist is simplified by removing thin branches. The value of the ICTreeI shows that extreme values are not perceived as realistic.

The ICTreeI is based on the Res2Net [Gao et al. 2019] backbone architecture, commonly used in various computer vision tasks. We utilize the Res2Net50 version with 16 residual blocks in total, which resulted in the best results among the tested variants (Tab. 4). We tested two variants with widely different sizes: the smaller Res2Net18 and the larger Res2Net50. Res2Net50 is around 14.5% better, while its pre-trained variant is around 16.7% better than Res2Net18. However, the results of these unmodified models are still very subpar ($MSE \approx 0.924$), even compared to the relatively simple basic models based on features ($MSE \approx 0.277$). Utilizing the pre-training regime (Sect. 6.2) results in a 36% improvement to the correlation coefficients and 82% improvement to mean squared error (ICTreeI_PTdb). Further improvement is seen when we utilize both tree and view scores (ICTreeI_PTdbs), effectively enlarging the training dataset. We also see a significant improvement using jittered view variants (Sect. 6.2), which is consistent with enlarging the dataset by the factor of 17×. Both ImageNet pre-trained model (ICTreeI_PTdbsv) and randomly initialized model (ICTreeI_NPdbsv) were tested. While both of their results are relatively on par (±4%), the pre-trained model has a definite edge on the Importance dataset.

7.5 Applications and Experiments

The ICTree is the first metric to automatically provide the user-perceived level of realism to virtual tree models, and it can be coupled with any method that generates 3D tree models.

7.5.1 Realism-assisted tree authoring. Fig. 1 and the accompanying video show an application of ICTree where the perceived tree realism guides a designer in creating a tree model. The artist creates a 3D model of a tree. Instead of relying only on subjective evaluation, ICTree provides real-time feedback on perceived realism. The user can focus on technical aspects, such as the number of polygons, tree size, or complexity. As the user edits the model, ICTree provides real-time feedback about the perceived realism. It is important to note that the metrics do not measure the beauty of a tree, which is highly subjective.

7.5.2 Abstract realism. An interesting question commonly asked in CG is what level of abstraction of a structure is still perceived as realistic. One way to creating more abstract geometry is to remove irrelevant details [Mehra et al. 2009]. We asked an artist to draw a tree with decreasing details (increasing abstraction level), and we executed ICTreeI on the provided images. Fig. 17 shows the results and the values of ICTreeI were: a) 0.157, b) 0.332, c) 0.729, d) 0.447, e) 0.353, f) 0.337, and g) 0.372. The tree's overall structure is preserved because the main branches, the prevalent features needed for perceived realism, are present. However, adding too many branches decreases perceived realism (0.157), while removing too many leads to a similar decrease (0.372). The trees close to the middle have their ICTreeI value higher (0.729 and 0.447).

		ICTree			Importance		
	Model	*MSE*	cor_p	cor_s	MSE	cor_p	cor_s
Basic Models	Linear	1.559	0.458	0.492	1.087	0.005	0.194
	Lasso	0.334	0.479	0.371	1.040	0.157	0.402
	Ridge	0.909	0.512	0.431	**0.878**	0.501	**0.443**
	ElasticNet	0.940	0.524	0.388	1.061	0.150	0.367
	Bayes Ridge	0.917	0.547	0.240	1.169	0.135	0.275
	DNN	**0.277**	0.753	0.747	1.988	0.365	0.355
	Random Forest	0.326	**0.769**	**0.738**	1.038	**0.516**	0.418
ICTreeF	ICTreeF_base	0.191	0.756	0.725	0.913	0.375	0.195
	ICTreeF_d	0.181	0.759	0.740	0.870	0.650	0.463
	ICTreeF_db	0.147	0.825	0.794	0.845	0.666	0.540
	ICTreeF_dbs	0.141	0.845	0.809	0.715	0.678	0.573
	ICTreeF_dbsv4	0.136	0.844	0.802	0.625	0.764	0.618
	ICTreeF_dbsv8	0.133	**0.842**	0.816	**0.625**	0.770	0.692
	ICTreeF_dbsv17	**0.122**	**0.842**	**0.836**	**0.619**	**0.803**	**0.759**
ICTreeI	ICTreeI_RN18NP	1.279	0.384	0.311	0.985	0.411	0.287
	ICTreeI_RN18PT	1.207	0.430	0.450	0.913	0.596	0.565
	ICTreeI_R2N18NP	1.109	0.477	0.455	0.823	0.580	0.592
	ICTreeI_R2N50NP	0.948	0.582	0.608	0.813	0.610	0.622
	ICTreeI_R2N50PT	0.924	0.572	0.588	0.814	0.631	0.603
	ICTreeI_PTdb	0.168	0.778	0.774	0.698	0.696	0.519
	ICTreeI_PTdbs	0.168	0.816	0.820	0.641	0.709	0.569
	ICTreeI_NPdbsv	0.161	**0.846**	0.821	**0.626**	0.711	0.608
	ICTreeI_PTdbsv	**0.155**	**0.852**	**0.838**	**0.619**	**0.782**	**0.649**

Table 4. **Ablation Experiments:** Results of the ablation experiments performed on both the primary ICTree dataset and the results of our Importance user study. Mean Squared Error (*MSE*), Pearson (cor_p), and Spearman (cor_s) correlation coefficients are provided. Results on the ICTree dataset show how well a given model approximates similar data, while the results on the Importance measure its generalization ability.

7.5.3 Realistic growth. Fig. 18 shows another application of the ICTree. There are many algorithms and methods that generate 3D models of trees based on their growth. While the output is based on biology, the perceived realism is unknown. In this example, the user generates a 3D geometry by setting the parameters of a growth model [Stava et al. 2014]. The growth model provides the 3D geometry, and ICTreeF then measures the perceived visual realism. Next, we randomly vary the growth model's parameters to find a similar geometry with a higher realism score. The random variations move the model's parameters by small values (around $\pm(5-10)\%$),

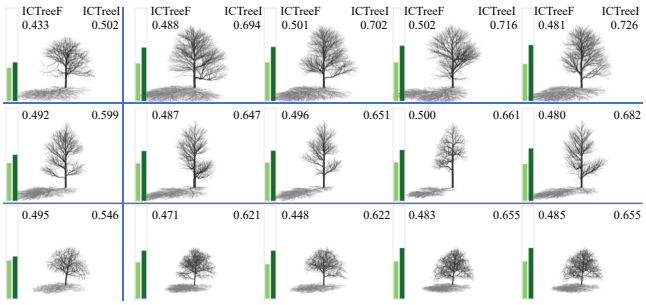

Fig. 18. **Increasing model realism:** the user provides parameters of a growth model, and 3D models are generated (left column). An automatic feedback loop then finds a more realistic model by slightly varying the parameters (right columns).

making sure the generated model is visually similar. The algorithm selects the tree that is perceived as the most realistic. We run 40-120 random variations, and it takes 97.5 *s* to explore around 80 trees. Most of the time is taken by the tree generator (71.9 *s*, 73.8%), while feature calculation takes around 4.8 *s* (4.9%), `ICTreeF` inference 0.6 *s* (0.6%), and `ICTreeI` inference 20.2 *s* (20.7%).

7.5.4 Exploration. of realistically grown trees If the user is not interested in similarity to the input, only the initial values of the growth model can be provided. The algorithm then explores a large space of values and generates trees with a higher value of perceived realism, as shown in an example in Fig. 19. We start with an existing tree on the left and optimize its perceptual score (right) by simulated annealing. We explore the parameter space by choosing a small random delta, modifying a single parameter at a time.

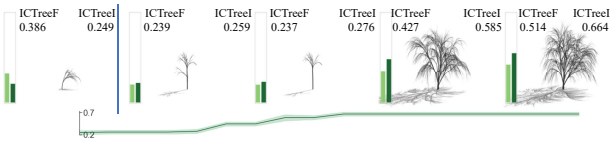

Fig. 19. **Automatic realistic model generation:** the user provides only the parameters of a growth model (left), and the model generates realistic trees by exploring a large range of values of the growth model.

7.5.5 Branch thickness. We show various examples of varying parameters and their importance in Fig. 16. Another example in Fig. 20 shows an effect of the branch thickness on the perceived level of realism. A fixed tree branching structure is used, and we change the tree branch thickness by varying the exponent D from Eqn. 1 as follows (Fig. 20): a) $D = 4.4$, b) $D = 3.7$, c) $D = 3.0$, d) $D = 2.3$, and e) $D = 1.6$. Very thick and thin branches are not perceived as realistic, and this observation is consistent with the findings from [Prusinkiewicz 1998]. The graph underneath each tree sequence was generated by a much higher sampling of the generated trees. We have generated 21 trees to have a smooth graph, and we show five trees to depict the visual difference.

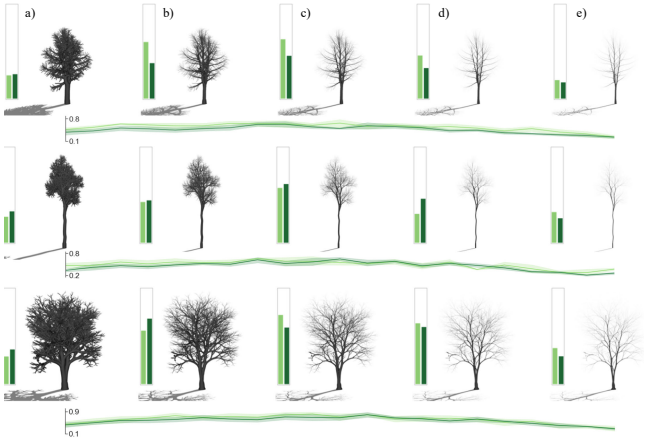

Fig. 20. **Branch thickness:** Trees with a fixed branching structure are changing their branch width. Both metrics show that extreme values are perceived as less realistic.

8 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

We introduced two deep neural perceptual predictors for synthetic models of trees. `ICTreeI` predicts the expected perceived realism from images, and `ICTreeF` uses easily computable geometric features. We used the results of a large user study to train the predictors, and we validated them by several correlations and a follow-up user study. Our analysis provides an in-depth understanding of what geometric features affect the perceived realism, giving tree modelers insight into what needs to be considered when designing their systems.

Limitations and future work: Trees are highly complex objects, and there are many possible avenues for future work. This work focuses on branching structure and does not consider leaves, color, texture, and other visual attributes. Studies exist that concur that leaves are important for species identification (*e.g.,* [Cerutti et al. 2013; Kumar et al. 2012]). We speculate that leaves can hide some properties that can affect the perceived realism, and future studies should provide more insight into this issue. However, gathering real-world data of trees with leaves is an open and unsolved problem. CG algorithms often approximate leaves by procedural methods [Livny et al. 2011]. `ICTreeI` assumes that the tree is rendered in grayscale. Tree color and texture should be considered in future studies. Our method uses a branch width, but it does not study its effect on perceived realism. Varying branch width has an impact on perceived realism. In particular, if only the skeleton would be shown, the trees would not even be recognized [Prusinkiewicz 1998]. We showed an example in Fig. 20, but a detailed study could bring more insight.

An important aspect is the tree species. As highlighted in Sect. 3, computer models often do not consider species at all. The models are generic, and they do not simulate a specific biological tree. While we study how different features affect tree's perceived realism, it would be interesting to consider species and how individual features affect perceived realism. However, such a study would require a large dataset of real trees that is not currently available.

The state-of-the-art algorithms for 3D tree reconstruction do not capture small branches and foliage. These errors lead to a lower average score for scanned trees within the `ICTree` dataset. We expect that increasing the precision of algorithms for 3D vegetation reconstruction will also lead to increased perceived realism of reconstructed models.

Our metrics show a perceived crowdsourced realism. However, realism does not mean beauty, as studied in [DeBruine et al. 2007]. One example is the tree in Fig. 3b) that received low perceptual realism scores but was considered nice by some (re)viewers. While the beauty of the trees seems to affect the user choices, the score differences are much more significant for biologically implausible trees. However, as realism and beauty are related, but not inclusive, we believe that further work in this area should address this question. Moreover, the beauty of correctly reconstructed trees should also be high.

Last but not least, we explicitly excluded outliers, such as too young or too old trees. However, old trees have their unique beauty, and their understanding is out of the scope of this study.

ACKNOWLEDGMENTS

This work was supported by *LTAIZ19004 Deep-Learning Approach to Topographical Image Analysis*; by the Ministry of Education, Youth and Sports of the Czech Republic within the activity INTER-EXCELENCE (LT), subactivity INTER-ACTION (LTA), ID: SMSM2019LTAIZ and by *Research Center for Informatics* No. CZ.02.1.01/0.0/0.0/16_019/0000765. This research was funded in part by National Science Foundation grant #10001387, *Functional Proceduralization of 3D Geometric Models* and by the Foundation for Food and Agriculture Research Grant ID: 602757. The content of this publication is solely the responsibility of the authors and does not necessarily represent the official views of the FFAR. We thank Bosheng Li for help with the 3D models.

REFERENCES

André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. 2010. Permutation importance: a corrected feature importance measure. *Bioinformatics* 26, 10 (2010), 1340–1347.

Fabricio Anastacio, Mario Costa Sousa, Faramarz Samavati, and Joaquim A. Jorge. 2006. Modeling Plant Structures Using Concept Sketches. In *Proc. of NPAR (NPAR '06)*. Association for Computing Machinery, 105–113. https://doi.org/10.1145/1124728.1124746

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D Fairchild. 2020. FLIP: A Difference Evaluator for Alternating Images. *ACM Trans. Graph.* 3, 2 (2020), 23. https://doi.org/10.1145/3406183

Tunç Ozan Aydın, Martin Čadík, Karol Myszkowski, and Hans-Peter Seidel. 2010. Video Quality Assessment for Computer Graphics Applications. In *ACM Trans. on Graph. (Proc. of SIGGRAPH Asia)*. ACM, 1–10.

Bedrich Benes and Erik Uriel Millán. 2002. Virtual Climbing Plants Competing for Space. In *Proceedings of the Computer Animation (CA '02)*. IEEE Computer Society, USA, 33.

Jan Beneš, Tom Kelly, F Děchtěrenko, Jaroslav Křivánek, and Pascal Müller. 2017. On realism of architectural procedural models. In *Comp. Graph. Forum*, Vol. 36. Wiley Online Library, 225–234.

Rolf Borchert and Hisao Honda. 1984. Control of development in the bifurcating branch system of Tabebuia rosea: a computer simulation. *Botanical Gazette* 145, 2 (1984), 184–195.

Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2018. Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment. *IEEE Transactions on Image Processing* 27, 1 (2018), 206–219. https://doi.org/10.1109/TIP.2017.2760518

Anne-Laure Boulesteix, Andreas Bender, Justo Lorenzo Bermejo, and Carolin Strobl. 2012. Random forest Gini importance favours SNPs with large minor allele frequency: impact, sources and recommendations. *Briefings in Bioinformatics* 13, 3 (2012), 292–304.

Derek Bradley, Derek Nowrouzezahrai, and Paul Beardsley. 2013. Image-based Reconstruction Synthesis of Dense Foliage. *ACM Trans. Graph.* 32, 4, Article 74 (2013), 74:1–74:10 pages.

Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. http://dx.doi.org/10.1023/A%3A1010933404324

Martin Čadík, Robert Herzog, Rafal Mantiuk, Radoslaw Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. 2013. Learning to Predict Localized Distortions in Rendered Images. *Comp. Graph. Forum* 32, 7 (2013), 401–410. https://doi.org/10.1111/cgf.12248

Guillaume Cerutti, Laure Tougne, Julien Mille, Antoine Vacavant, and Didier Coquin. 2013. Understanding leaves in natural images–a model-based approach for tree species identification. *Computer Vision and Image Understanding* 117, 10 (2013), 1482–1501.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

Massimiliano Corsini, Mohamed-Chaker Larabi, Guillaume Lavoué, Oldřich Petřík, Libor Váša, and Kai Wang. 2013. Perceptual Metrics for Static and Dynamic Triangle Meshes. *Comp. Graph. Forum* (2013). https://doi.org/10.1111/cgf.12001

Herbert Aron David. 1988. *The Method of Paired Comparisons*. C. Griffin. https://books.google.cz/books?id=bB21VsB_GyYC

Phillippe De Reffye, Claude Edelin, Jean Françon, Marc Jaeger, and Claude Puech. 1988. Plant models faithful to botanical structure and development. *ACM Siggraph Computer Graphics* 22, 4 (1988), 151–158.

Lisa M DeBruine, Benedict C Jones, Layla Unger, Anthony C Little, and David R Feinberg. 2007. Dissociating averageness and attractiveness: attractive faces are not always average. *Journal of Exp. Psychology: Human Perc. and Performance* 33, 6 (2007), 1420.

Oliver Deussen, Pat Hanrahan, Brend Lintermann, Radomír Měch, Max Pharr, and Przemyslaw Prusinkiewicz. 1998. Realistic Modeling and Rendering of Plant Ecosystems. *ACM Trans. Graph.* (1998), 275–286.

Shenglan Du, Roderik Lindenbergh, Hugo Ledoux, Jantien Stoter, and Liangliang Nan. 2019. AdTree: Accurate, Detailed, and Automatic Modelling of Laser-Scanned Trees. *Remote Sensing* 11, 18 (9 2019), 2074. https://doi.org/10.3390/rs11182074

Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. 2019. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence* (2019).

Ned Greene. 1989. Voxel Space Automata: Modeling with Stochastic Growth Processes in Voxel Space. *SIGGRAPH Comp. Graph.* 23, 3 (1989), 175–184.

Jianwei Guo, Shibiao Xu, Dong-Ming Yan, Zhanglin Cheng, Marc Jaeger, and Xiaopeng Zhang. 2018. Realistic procedural plant modeling from multiple view images. *IEEE Trans. on Vis. and Comp. Graphics* 26, 2 (2018), 1372 – 1384.

Ralf Habel, Alexander Kusternig, and Michael Wimmer. 2009. Physically Guided Animation of Trees. *Comp. Graph. Forum* 28, 2 (2009), 523–532.

Torsten Hädrich, Bedrich Benes, Oliver Deussen, and Sören. Pirk. 2017. Interactive Modeling and Authoring of Climbing Plants. *Comp. Graph. Forum* 36, 2 (2017), 49–61.

Alexander Hapfelmeier and Kurt Ulm. 2013. A new variable selection approach using random forests. *Computational Statistics & Data Analysis* 60 (2013), 50–69.

Robert Herzog, Martin Čadík, Tunç O. Aydın, Kwawng In Kim, Karol Myszkowski, and Hans-Peter Seidel. 2012. NoRM: No-Reference Image Quality Metric for Realistic Image Synthesis. *Comp. Graph. Forum* 31, 2 (2012), 545–554.

Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. 2013. L1-medial skeleton of point cloud. *ACM Trans. Graph.* 32, 4 (7 2013), 1–8. https://doi.org/10.1145/2461912.2461913

Mathieu Jung, D. Léger, and M. Gazalet. 2002. Univariant assessment of the quality of images. *J. Electronic Imaging* 11 (2002), 354–364.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:cs.LG/1412.6980

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 1097–1105.

Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. 2012. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*. Springer, 502–516.

Harold L Kundel and Marcia Polansky. 2003. Measurement of observer agreement. *Radiology* 228, 2 (2003), 303–308.

Debarati Kundu, Lark Kwon Choi, Alan C. Bovik, and Brian L. Evans. 2018. Perceptual quality evaluation of synthetic pictures distorted by compression and transmission. *Signal Processing: Image Communication* 61 (2018), 54 – 72.

Guillaume Lavoué, Mohamed-Chaker Larabi, and Libor Vása. 2016. On the Efficiency of Image Metrics for Evaluating the Visual Quality of 3D Models. *IEEE TVCG* 22, 8 (2016), 1987–1999.

Guillaume Lavoué. 2011. A Multiscale Metric for 3D Mesh Visual Quality Assessment. *Comp. Graph. Forum* 30, 5 (2011), 1427–1437.

Bosheng Li, Jacek Kałużny, Jonathan Klein, Dominik Michels, Wojtek Pałubicki, Bedrich Benes, and Sören Pirk. 2021. Learning to Reconstruct Botanical Trees from Single Images. *ACM Trans. Graph.* 40, 6 (2021).

Yangyan Li, Xiaochen Fan, Niloy J. Mitra, Daniel Chamovitz, Daniel Cohen-Or, and Baoquan Chen. 2013. Analyzing Growing Plants from 4D Point Cloud Data. *ACM Trans. Graph.* 32, 6, Article 157 (Nov. 2013), 10 pages.

Yan Lin, Ji Liu, and Jianlin Zhou. 2020. A novel tree-structured point cloud dataset for skeletonization algorithm evaluation. *CoRR* abs/2001.02823 (2020). arXiv:2001.02823

Aristid Lindenmayer. 1968. Mathematical models for cellular interaction in development. *J. Theor. Biol.* Parts I and II, 18 (1968), 280–315.

Berndt Lintermann and Oliver Deussen. 1999. Interactive Modeling of Plants. *IEEE Comput. Graph. Appl.* 19, 1 (1999), 56–65.

Yanchao Liu, Jianwei Guo, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, and Hui Huang. 2021. TreePartNet: Neural Decomposition of Point Clouds for 3D Tree Reconstruction. *ACM Trans. Graph.* 40, 6 (2021).

Yotam Livny, Sören Pirk, Zhanglin Cheng, Feilong Yan, Oliver Deussen, Daniel Cohen-Or, and Baoquan Chen. 2011. Texture-lobes for Tree Modelling. *ACM Trans. Graph.* 30, 4, Article 53 (2011), 10 pages.

Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. 2010. Automatic reconstruction of tree skeletal structures from point clouds. In *ACM Trans. Graph.*, Vol. 29. 1–8. https://doi.org/10.1145/1866158.1866177

Steven Longay, Adam Runions, Francois Boudon, and Przemyslaw Prusinkiewicz. 2012. TreeSketch: interactive procedural modeling of trees on a tablet. In *Proc. of the Intl. Symp. on SBIM*. 107–120.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. 3.

Rafal Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. 2011. HDR-VDP-2: A Calibrated Visual Metric for Visibility and Quality Predictions in All Luminance Conditions. *ACM Trans. Graph.* 30, 4, Article 40 (July 2011), 14 pages.

Ravish Mehra, Qingnan Zhou, Jeremy Long, Alla Sheffer, Amy Gooch, and Niloy Mitra. 2009. Abstraction of Man-Made Shapes. *ACM Trans. Graph.* 28, 5 (2009), 1–10.

Ryoko Minamino and Masaki Tateno. 2014. Tree branching: Leonardo da Vinci's rule versus biomechanical models. *PloS one* 9, 4 (2014), e93535.

Anush Krishna Moorthy and Alan Conrad Bovik. 2010. A Two-Step Framework for Constructing Blind Image Quality Indices. *IEEE Signal Processing Letters* 17, 5 (2010), 513–516. https://doi.org/10.1109/LSP.2010.2043888

Radomír Měch and Przemyslaw Prusinkiewicz. 1996. Visual models of plants interacting with their environment. In *Proc. of SIGGRAPH*. ACM, 397–410.

Boris Neubert, Thomas Franken, and O. Deussen. 2007. Approximate Image-based Tree-modeling Using Particle Flows. *ACM Trans. Graph.* 26, 3, Article 88 (2007).

Makoto Okabe, Shigeru Owada, and Takeo Igarashi. 2007. Interactive Design of Botanical Trees Using Freehand Sketches and Example-based Editing. In *ACM SIGGRAPH Courses*. ACM, Article 26.

Peter E. Oppenheimer. 1986. Real Time Design and Animation of Fractal Plants and Trees. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. Association for Computing Machinery, New York, NY, USA, 55–64. https://doi.org/10.1145/15922.15892

Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz. 2009. Self-organizing Tree Models for Image Synthesis. *ACM Trans. Graph.* 28, 3, Article 58 (2009), 10 pages.

Yixin Pan, Irene Cheng, and Anup Basu. 2005. Quality metric for approximating subjective evaluation of 3-D objects. *IEEE Transactions on Multimedia* 7, 2 (2005), 269–279. https://doi.org/10.1109/TMM.2005.843364

Maria Perez-Ortiz and Rafal K Mantiuk. 2017. A practical guide and software for analysing pairwise comparison experiments. *arXiv preprint arXiv:1712.03686* (2017).

Sören Pirk, Michal Jarząbek, Torsten Hädrich, Dominik L. Michels, and Wojciech Palubicki. 2017. Interactive Wood Combustion for Botanical Tree Models. *ACM Trans. Graph.* 36, 6, Article 197 (Nov. 2017), 12 pages.

Sören Pirk, Till Niese, Torsten Hädrich, Bedrich Benes, and Oliver Deussen. 2014. Windy Trees: Computing Stress Response for Developmental Tree Models. *ACM Trans. Graph.* 33, 6, Article 204 (Nov. 2014), 11 pages.

Sören Pirk, Ondrej Stava, Julian Kratt, Michel Abdul Massih Said, Boris Neubert, Radomír Měch, Bedrich Benes, and Oliver Deussen. 2012. Plastic Trees: Interactive Self-Adapting Botanical Tree Models. *ACM Trans. Graph.* 31, 4 (2012).

Przemyslaw Prusinkiewicz. 1986. Graphical applications of L-systems. In *Proc. on Graph. Interf.* 247–253.

Przemyslaw Prusinkiewicz. 1998. *In search of the right abstraction: the synergy between art, science, and information technology in the modeling of natural phenomena.* Art @ Science. Springer, Wien. 60–68 pages.

Przemyslaw Prusinkiewicz, Mark S. Hammel, and Eric Mjolsness. 1993. Animation of plant development. In *Proceedings SIGGRAPH (SIGGRAPH '93)*. ACM, 351–360.

Przemyslaw Prusinkiewicz, Mark James, and Radomír Měch. 1994. Synthetic Topiary. *Proceedings SIGGRAPH* (1994), 351–358. http://doi.acm.org/10.1145/192161.192254

Przemyslaw Prusinkiewicz and Aristid Lindenmayer. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc.

Hongxing Qin, Jia Han, Ning Li, Hui Huang, and Baoquan Chen. 2020. Mass-Driven Topology-Aware Curve Skeleton Extraction from Incomplete Point Clouds. *IEEE Trans. on Vis. and Comp. Graphics* 26, 9 (2020), 2805–2817.

Suren Deepak Rajasekaran, Hao Kang, Bedrich Benes, Martin Čadík, Eric Galin, Eric Guérin, Adrien Peytavie, and Pavel Slavík. 2019. PTRM: Perceived Terrain Realism Metrics. *arXiv preprint arXiv:1909.04610* (2019).

Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala. 2007. Visual Equivalence: Towards a New Standard for Image Fidelity. *ACM Trans. Graph.* 26, 3 (2007), 12.

Alex Reche-Martinez, Ignacio Martin, and George Drettakis. 2004. Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Trans. Graph.* 23, 3 (2004), 720–727.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. 2019. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237* (2019).

Bernice E. Rogowitz and Holly E. Rushmeier. 2001. Are image quality metrics adequate to evaluate the quality of geometric objects?. In *Human Vision and Electronic Imaging VI*, Vol. 4299. Intl. Soc. for Optics and Photonics, SPIE, 340 – 348.

Petr Sloup, Tomas Rebok, and Jan Hanus. 2013. Automatic Tree Reconstruction from its Laser Scan. In *Global Change and Resilience*. Global Change Research Centre, Academy of Sciences of the Czech Republic, Brno.

SpeedTree. 2021. *SpeedTree*. http://www.speedtree.com

Ondrej Stava, Sören Pirk, Julian Kratt, Baoquan Chen, Radomír Měch, Oliver Deussen, and Bedrich Benes. 2014. Inverse procedural modelling of trees. In *Comp. Graph. Forum*, Vol. 33. Wiley Online Library, 118–131.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *ICCV*.

Sundaram Suresh, R. Venkatesh Babu, and Hyoung Joong Kim. 2009. No-reference image quality assessment using modified extreme learning machine classifier. *Applied Soft Computing* 9, 2 (2009), 541 – 552. https://doi.org/10.1016/j.asoc.2008.07.005

Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. 2012. Mean Curvature Skeletons. *Comp. Graph. Forum* 31, 5 (2012), 1735–1744.

Ping Tan, Tian Fang, Jianxiong Xiao, Peng Zhao, and Long Quan. 2008. Single Image Tree Modeling. *ACM Trans. Graph.* 27, 5, Article 108 (Dec. 2008), 7 pages.

Ping Tan, Gang Zeng, Jingdong Wang, Sing Bing Kang, and Long Quan. 2007. Image-based Tree Modeling. *ACM Trans. Graph.* 26, 3, Article 87 (2007).

Huixuan Tang, Neel Joshi, and Ashish Kapoor. 2011. Learning a blind measure of perceptual image quality. In *CVPR 2011*. IEEE, 305–312.

Fakhri Torkhani, Kai Wang, and Jean-Marc Chassery. 2015. Perceptual quality assessment of 3D dynamic meshes: Subjective and objective studies. *Signal Processing: Image Communication* 31 (2015), 185–204.

Bohan Wang, Yili Zhao, and Jernej Barbič. 2017. Botanical Materials Based on Biomechanics. *ACM Trans. Graph.* 36, 4, Article 135 (July 2017), 13 pages.

Guan Wang, Hamid Laga, Ning Xie, Jinyuan Jia, and Hedi Tabia. 2018. The shape space of 3D botanical tree models. *ACM Trans. Graph.* 37, 1 (2018).

Kai Wang, Fakhri Torkhani, and Annick Montanvert. 2012. A fast roughness-based approach to the assessment of 3D mesh visual quality. *Computers and Graphics* 36, 7 (Nov. 2012), 808–818. https://doi.org/10.1016/j.cag.2012.06.004

Wenping Wang, Bert Jüttler, Dayue Zheng, and Yang Liu. 2008. Computation of Rotation Minimizing Frames. *ACM Trans. Graph.* 27, 1, Article 2 (March 2008), 18 pages.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *Trans. Img. Proc.* 13, 4 (April 2004), 600–612.

Geoffrey B West, James H Brown, and Brian J Enquist. 1999. A general model for the structure and allometry of plant vascular systems. *Nature* 400, 6745 (1999), 664–667.

Krzysztof Wolski, Daniele Giunchi, Nanyang Ye, Piotr Didyk, Karol Myszkowski, Radoslaw Mantiuk, Hans-Peter Seidel, Anthony Steed, and Rafal K. Mantiuk. 2018. Dataset and Metrics for Predicting Local Visible Differences. *ACM Trans. Graph.* 37, 5 (2018), 172:1–172:14. https://doi.org/10.1145/3196493

Jinliang Wu, Xiaoyong Shen, Wei Zhu, and Ligang Liu. 2013. Mesh Saliency with Global Rarity. *Graph. Models* 75, 5 (Sept. 2013), 255–264.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*. 1912–1920.

Ke Xie, Feilong Yan, Andrei Sharf, Oliver Deussen, Hui Huang, and Baoquan Chen. 2016. Tree Modeling with Real Tree-Parts Examples. *IEEE Trans. on Vis. and Comp. Graphics* 22, 12 (2016), 2608–2618. https://doi.org/10.1109/TVCG.2015.2513409

Hui Xu, Nathan Gossett, and Baoquan Chen. 2007. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.* 26, 4 (2007), Article 19.

Peng Ye, Jayant Kumar, and David Doermann. 2014. Beyond Human Opinion Scores: Blind Image Quality Assessment Based on Synthetic Scores. In *IEEE CVPR*. 4241–4248. https://doi.org/10.1109/CVPR.2014.540

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*. 586–595. https://doi.org/10.1109/CVPR.2018.00068

Yili Zhao and Jernej Barbič. 2013. Interactive Authoring of Simulation-ready Plants. *ACM Trans. Graph.* 32, 4, Article 84 (2013), 12 pages.