POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

GreenCanvas

3D Model Plant Generator System Based on Drawings

Ana Martins

School of Management and Technology
Department of Computer Engineering
Master in Computer Engineering - Mobile Computing

Leiria, September 2025

This page intentionally left blank.

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

GreenCanvas

3D Model Plant Generator System Based on Drawings

Ana Martins

Supervisor: Anabela Marto
Full Professor, Polytechnic of Leiria

Co-supervisor: Roberto Ribeiro
Full Professor, Polytechnic of Leiria
Alexandrino Gonçalves
Full Professor, Polytechnic of Leiria
Rita Ascenso
Full Professor, Polytechnic of Leiria

School of Management and Technology
Department of Computer Engineering
Master in Computer Engineering - Mobile Computing

Dissertation

Leiria, September 2025

GreenCanvas

Acknowledgements

I would like to express my gratitude to all of my supervisors and co-supervisors: Professors Anabela Marto, Roberto Ribeiro, Alexandrino Gonçalves, and Rita Ascenso. Their guidance and insightful feedback were invaluable throughout this dissertation, particularly regarding the systematic review and the preparation of related articles. I am also grateful for the warm and supportive environment of our weekly meetings, as well as their empathy and encouragement during challenging moments.

I would also like to thank Professor Nuno Rodrigues, who, although not officially part of the project due to the number of supervisors, provided useful insights during the initial stages of the systematic review.

My sincere thanks go to my family and friends, whose unwavering support has been a constant source of strength throughout my undergraduate and graduate studies in computer science.

I am grateful to all participants who generously contributed their time and feedback during the usability studies. Their insights were crucial for evaluating and refining the add-on.

This research was supported by the Portuguese Foundation for Science and Technology (FCT) under the project UIDB/04524/2020. I also acknowledge the use of the Blender platform and associated open-source resources, which made the implementation of the framework possible.

This page intentionally left blank.

Abstract

The generation of three-dimensional plant models is increasingly important for applications in agriculture, scientific visualization, landscape design, and immersive virtual environments. Existing approaches often rely on specialized hardware or sensor data, and publicly available models are limited, repetitive, or constrained to specific species, limiting creative flexibility. This dissertation addresses these challenges by focusing on sketch-based plant modelling as an accessible and flexible approach.

A systematic review of state-of-the-art methods was conducted following the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) guidelines, identifying two main directions: sketch-based modelling and tree generation from photographic data. Insights from this review informed the design of a novel framework that integrates single- and multi-sketch strategies to generate 3D plant models from user drawings. The framework was implemented as a Blender add-on, supporting features such as variable levels of detail, leaf propagation, and generation of multiple model variations, while providing a streamlined and intuitive interface.

The add-on was evaluated through quantitative and qualitative analyses. Task completion times demonstrated efficient modelling, and a mean System Usability Scale (SUS) score of 56.4 indicated moderate usability. Correlations with demographic factors revealed influences of user experience, background, and familiarity with input devices on usability. Qualitative feedback highlighted areas for improvement in feature intuitiveness and onboarding. Despite a limited and unevenly distributed participant sample, the results indicate that the framework is a promising tool for procedural plant modelling and can enhance creative workflows for users of varying expertise.

This work contributes both a systematic understanding of visual-based plant modelling approaches and a practical tool that bridges sketching and 3D model creation, providing a foundation for future enhancements such as AI-assisted sketch interpretation, expanded plant element support, and improved interface features.

Keywords: Plant, 3D Plant Modelling, Blender add-on, 2D to 3D, Sketch to 3D, 3D Generation

Contents

Appendices

List of Figures

List of Tables

This page intentionally left blank.

Acronyms

AI	Artificial Intelligence. (*p. 54*)
API	Application Programming Interface. (*p. 32*)
gpencil	Grease Pencil. (*p. 33*)
ICGI	International Conference on Graphics and Interaction. (*p. 3, 25*)
Instant-NGP	instant neural graphics primitives. (*p. 15–17, 19, 73*)
jDE	self-adaptive Differential Evolution. (*p. 19, 20*)
L-system	Lindenmayer system. (*p. 5*)
LiDAR	Light Detection and Ranging. (*p. 5, 21, 68*)
LoD	Level of Detail. (*p. ix, 2, 3, 29, 35, 36, 42, 53*)
PRISMA	Preferred Reporting Items for Systematic reviews and Meta-Analyses. (*p. iii, ix, 1, 6, 8, 9, 53*)
PSNR	Peak Signal-to-Noise Ratio. (*p. 19, 20*)
SFM	Structure from Motion. (*p. 13, 14*)
SUS	System Usability Scale. (*p. iii, ix, xi, 3, 41–46, 49, 50, 53*)
UI	User Interface. (*p. 2, 12, 15, 17, 20, 26, 27, 31, 32, 36, 46*)
WSL	Windows Subsystem for Linux. (*p. 31*)

This page intentionally left blank.

1

Introduction

The generation of three-dimensional plant models is an expanding area of research with applications across various fields, including agriculture (Mitsanis et al., 2024) and computer graphics. It supports a range of tasks such as plant monitoring (ZHANG et al., 2018; Paturkar et al., 2022), scientific visualization (Lin et al., 2022), landscape design (P. Liu et al., 2016; Muhar, 2001; Xu et al., 2024), and the creation of immersive virtual environments (e.g., games and virtual reality) (A. Hu et al., 2024b). Common methods include rule-based systems such as L-systems (Prusinkiewicz et al., 1990; Lu et al., 2015), visual-based generation from photographs or sketches, and sensor-enhanced techniques for increased geometric fidelity, such as Light Detection and Ranging (LiDAR) (Nguyen et al., 2015; Bailey et al., 2018; Xie et al., 2018) or the Kinect sensor (Li et al., 2015; Y. Liu et al., 2023). A recent cross-cutting review (Okura, 2022) offers a comprehensive overview of plant modelling and reconstruction methodologies, encompassing these and additional techniques. Furthermore, specialized software such as SpeedTree, xFrog, TheGrove3D, and TreeIt incorporate botanical knowledge and offer tailored interfaces that support the efficient and accurate creation of 3D plant models.

Despite growing interest in the field, recent advancements in 3D plant generation have largely relied on specialized hardware and sensor data to produce high-fidelity models. The limited availability of publicly accessible 3D plant models often results in repetitive and generic content, which poses challenges for virtual environment creators seeking to populate scenes with specific plant species, as well as for game developers designing fictional worlds with entirely imagined flora. Generating 3D plant models from visual inputs, such as photographs or sketches, whether based on real specimens or imagined designs, offers a more accessible and creative alternative.

To understand the current state of the field, a systematic review was conducted following the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) guidelines. To our knowledge, this is the first systematic review of plant modelling approaches that rely exclusively on visual inputs, excluding specialized hardware or external sensors. The review identified two main directions: 1) sketch-based modelling, and 2) tree generation from photographic data. Since the focus of this dissertation is on sketch-based modelling, further analysis was conducted on this approach. The available studies were limited and somewhat outdated (2005–2008), covering multi-sketch flower modelling (Ijiri et al., 2005a; Ding et al., 2008a) and single-sketch tree generation (Okabe et al., 2007a).

Building on the reviewed literature, a framework was proposed that integrates elements from both single- and multi-sketch strategies. The framework provides a streamlined interface that allows users to design plants by sketching branches and leaves. These sketches are processed through a structured pipeline involving convex hull construction, swept surface modelling, and B-spline-based leaf generation. The proposed design also incorporates additional features from the analysed methods, including the application of normal maps to the 3D models, Level of Detail (LoD) variation, and the ability to generate multiple 3D variations from the resulting plant model. Together, these features offer enhanced control and creative flexibility for the user.

The implementation of the framework resulted in a Blender add-on, allowing users to integrate more quickly and easily into the modelling software they are already familiar with, while leveraging Blender's native tools to simplify post-processing tasks, such as scale and rotation operations, texture application and mesh deformation.

The dissertation is organised as follows: Chapter 2 presents a systematic review of state-of-the-art methods for modelling plants from photographs and/or sketches, excluding approaches that rely on external hardware or sensors. Chapter 3 introduces the proposed framework, outlining the implementation plan for the add-on along with initial User Interface (UI) prototypes, based on insights from the review. Chapter 4 details the implementation of the add-on, including its user interface and key features. Chapter 5 reports on the usability and effectiveness tests and their results. Chapter 6 discusses these findings in depth. Finally, Chapter 7 summarises the dissertation and outlines directions for future work.

1.1 Motivation

Existing publicly available plant models are limited, often repetitive, and may not cover all desired plant species. Moreover, due to the inherent complexity of plant structures, their modelling process is typically challenging and frequently requires specialized hardware or relies on sensor data to enhance accuracy. The ability to generate a 3D plant model directly from a photograph or a sketch, whether based on a real specimen or an imagined design, would introduce a level of creativity and uniqueness currently lacking, significantly enriching virtual environments.

1.2 Objectives

The main objectives of this dissertation are the following:

- Conduct a systematic review of state-of-the-art methods for generating 3D plant models from visual inputs (photographs and/or sketches), excluding approaches that rely on specialized hardware or external sensors.
- Propose a framework capable of converting drawings into 3D plant models, generating varying Level of Detail (LoD) for each model, and producing multiple variations to reduce repetition.
- Implement the proposed framework, providing an intuitive interface and a seamless workflow for 3D plant creation.
- Evaluate the system in a use-case scenario, assessing both its usability and effectiveness.

1.3 Contributions

The main contributions of this dissertation are as follows:

- A systematic review of state-of-the-art methods for plant modelling from photographs and/or sketches, without relying on specialised hardware or external sensors. Two main approaches were identified and, to our knowledge, this is the first systematic review on the topic.
- Design of a plant modelling framework based on sketches that synthesises insights from the review. An article describing this proposal was submitted and accepted for the 2025 edition of the International Conference on Graphics and Interaction (ICGI)[1], as part of the Conference track.
- Development of a Blender add-on capable of converting drawings into 3D plant models, supporting additional features such as LoD variations, leaf propagation, and the generation of multiple model variations, while including custom interface elements tailored to the system.
- A quantitative and qualitative assessment of the add-on's usability, using System Usability Scale (SUS) scores, and its effectiveness through task performance, followed by an analysis correlating the metrics with demographic data, providing insights into the system's strengths, limitations, and potential areas for improvement.

---

[1] https://gpcg.pt/icgi2025/

This page intentionally left blank.

2

Systematic Review

The generation of three-dimensional plant models is an expanding area of research with applications across various fields, including agriculture (Mitsanis et al., 2024) and computer graphics. It supports a range of tasks such as plant monitoring (ZHANG et al., 2018; Paturkar et al., 2022), scientific visualization (Lin et al., 2022), landscape design (P. Liu et al., 2016; Muhar, 2001; Xu et al., 2024), and the creation of immersive virtual environments (e.g., games and virtual reality) (A. Hu et al., 2024b). Common methods include rule-based systems such as Lindenmayer system (L-system) (Prusinkiewicz et al., 1990; Lu et al., 2015), visual-based generation from photographs or sketches, and sensor-enhanced techniques for increased geometric fidelity, such as Light Detection and Ranging (LiDAR) (Nguyen et al., 2015; Bailey et al., 2018; Xie et al., 2018) or the Kinect sensor (Li et al., 2015; Y. Liu et al., 2023). A recent cross-cutting review (Okura, 2022) offers a comprehensive overview of plant modelling and reconstruction methodologies, encompassing these and additional techniques. Furthermore, specialized software such as SpeedTree, xFrog, TheGrove3D, and TreeIt incorporate botanical knowledge and offer tailored interfaces that support the efficient and accurate creation of 3D plant models.

Despite growing interest and a diverse body of literature, research on 3D plant generation remains fragmented, with heterogeneous objectives, diverse input modalities, and a wide range of technical approaches. Many recent advancements rely heavily on specialized hardware and sensor data to achieve high-fidelity models. Additionally, the limited availability of publicly accessible 3D plant models often leads to repetitive or generic content, creating challenges for virtual environment creators who require specific plant species, as well as for game developers designing fictional worlds with entirely imagined flora. Generating 3D plant models directly from visual inputs—such as photographs or sketches, whether based on real specimens or original designs—offers a more accessible and creative alternative. To our knowledge, no systematic review has yet focused exclusively on methods that rely solely on visual data, independent of specialized hardware or external sensors.

A systematic review was conducted focusing exclusively on approaches that generate 3D plant models from photographs and/or sketches. By limiting the scope to methods that require minimal effort and no additional equipment, we highlighted techniques that are both accessible and widely applicable. Our findings reveal two dominant approaches in the field: 1) sketch-based modelling, and 2) photographic generation of trees. For each approach, we synthesize key processes, underlying techniques, and representative results.

2.1 Preliminary Steps

Before conducting the systematic review, we undertook several preliminary steps. An initial exploratory search using broad terms was carried out to familiarize ourselves with the terminology commonly used in the field. Potential search terms were then identified by examining the titles, abstracts, and keywords of the retrieved records. Based on these findings, we developed a draft search strategy, which was iteratively refined by testing combinations of recurrent terms and filtering for relevance.

Initial Search The initial search was conducted using straightforward search keywords, such as 1) 3D plant reconstruction, 2) 3D plant generator, 3) 3D plant modelling / modeling, 4) 3D tree reconstruction, and 5) 3D tree generator. Each of the aforementioned keywords was entered into both Google Scholar and IEEE Xplorer databases. For each search, the initial one hundred articles—ordered by relevance—were reviewed and categorised as excluded or not, based on a list of exclusion terms that was iteratively refined during the process based on the result of each keyword search. The following terms were excluded: nuclear, power, radioactive, CAD, artery, angiography, and lasers. The search yielded 647 articles, of which 556 were excluded. None of the included articles were used in the systematic review, since the statement of Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) was further used as a guideline. However, common words among the classified articles were helpful in defining the search key used in the review and an initial set of exclusion reasons.

Search Key A search key for the systematic review was defined to ensure its applicability across all identified databases. The search key was first divided into three "`OR`" conditions, including inputs, methods and outputs. The initial search key used is presented in Listing 1. The search key was then refined based on the resulting articles in the Web of Science, Scopus, IEEE Xplorer and Springer databases. The most relevant words in the titles that appeared to be relevant to the aim of the review were added, as well as obvious exclusion terms, e.g. nuclear. The process of creating the final search key is illustrated in Table A.1, while Table 2.1 presents the resulting articles for each of the above steps. The final key is presented in Listing 2.

Listing 1: *Initial Search Key.*

```
illustration / images / 2D scan / photograph / sketch / drawing / concept art
OR
procedural model / procedural reconstruction / generative / generation
OR
3D mesh / 3D tree / 3D plant
```

Table 2.1: *Search key resulting articles from the various literature.*

Search	Search Name	Web of Science	Scopus	IEEE Xplorer	Springer
1	Initial topics	112	918	529	3006
2	Modeling & Modelling	113	923	529	3012
3	Removed 3D mesh	21	51	8	271
4	3D flower	24	58	9	551
5	Mesh terms	25	63	9	606
6	Model terms	47	159	41	4083
7	Asset terms	47	160	41	4110
8	Input variants	71	167	64	5245
9	Generate variants	192	446	123	10323
10	Reconstruction	285	651	189	10523
11	Output Variants	395	651	231	14053
12	Exclusions plant related	381	552	204	6392
13	Exclusions tree model related	300	345	144	2255
14	Exclusions tree related	266	269	122	1713
15	Exclusions sensor related	153	111	65	716
16	Exclusions method related	74	27	26	316
17	Exclusions flower related	73	27	26	304
18	Springer excluded words	67	13	19	36
19	Scopus excluded words	67	11	19	33
20	IEEE Xplorer excluded words	66	11	14	26

Listing 2: *Final Search Key.*

```
illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D
↪  scanning / photograph / photographs / sketch / sketches / sketching / drawing /
↪  drawings / concept art
OR
procedural model / procedural modeling / procedural modelling / procedural
↪  reconstruction / reconstruction / generative / generation / generate / generates /
↪  generated / generating
OR
3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees
↪  mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes /
↪  plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree
↪  model / trees model / tree models / trees models / plant model / plants model /
↪  plant models / plants models / flower model / flowers model / flower models /
↪  flowers models / tree asset / trees asset / tree assets / trees assets / plant
↪  asset / plants asset / plant assets / plants assets / flower asset / flowers asset
↪  / flower assets / flowers assets
AND NOT
```

```
nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 /
↪  decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular
↪  / angiography / cancer / hematopoietic / human pose / laser / lasers / scan / scans
↪  / sensor / sensors / LiDAR / UAV / stereo / stereoscopy / intrusive / depth / x-ray
↪  / radar / radars / sensing / classification / recognition / validation / predict /
↪  prediction / predicting / dataset / datasets / L-system / L-systems / nanoflower /
↪  nanoflowers / battery / batteries / semiconductor / semiconductors / CAD /fuzzy /
↪  stress / database / logic / time-varying / pattern / cell growth / residual /
↪  financial / economy / butterfly / SPQR-tree / driving / phylogenetic / supervisory
↪  / recursive / voltage / task-tree / animal / moduli / management / motor / chaos /
↪  PID / elastic / fowler / dendritic / ecotoxicity / IP / galaxy / vision tracking /
↪  moving person / PI / accelerator / accelerators.
```

2.2 Methods

The statement of Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) (Page et al., 2021) was used as a guideline for the systematic review.

2.2.1 Data Sources and Search Strategy

For the conducted systematic review, the Web of Science, Scopus, IEEE Xplorer, and Springer databases were used up to November 2024. The search was performed using the defined search key (mentioned in Section 2.1). In addition to the resulting articles, their reference lists were also analysed, resulting in the inclusion of some additional articles. An attempt was made to contact one of the authors of an unavailable article, but no response was received.

2.2.2 Study Selection

Both the literature search and the review of the titles and abstracts to select eligible studies were performed independently by one reviewer. The full-text articles were also reviewed independently, applying the following exclusion criteria: 1) does not use images and/or sketches as input; 2) uses 3D sketches as input; 3) uses specialised hardware or sensor data in the process; 4) requires prior knowledge of the specific plant species. The inclusion criteria for the studies consist of the non-application of the aforementioned exclusion criteria. The PRISMA flow diagram (see Figure 2.1) illustrates the study selection process.

2.2.3 Data extraction

Data extraction was carried out independently, and no study authors were contacted for additional information. The review did not specify predefined quantitative outcome domains. Instead, the goal was to extract all commonly reported information across studies to identify patterns and methodological trends in the creation of 3D

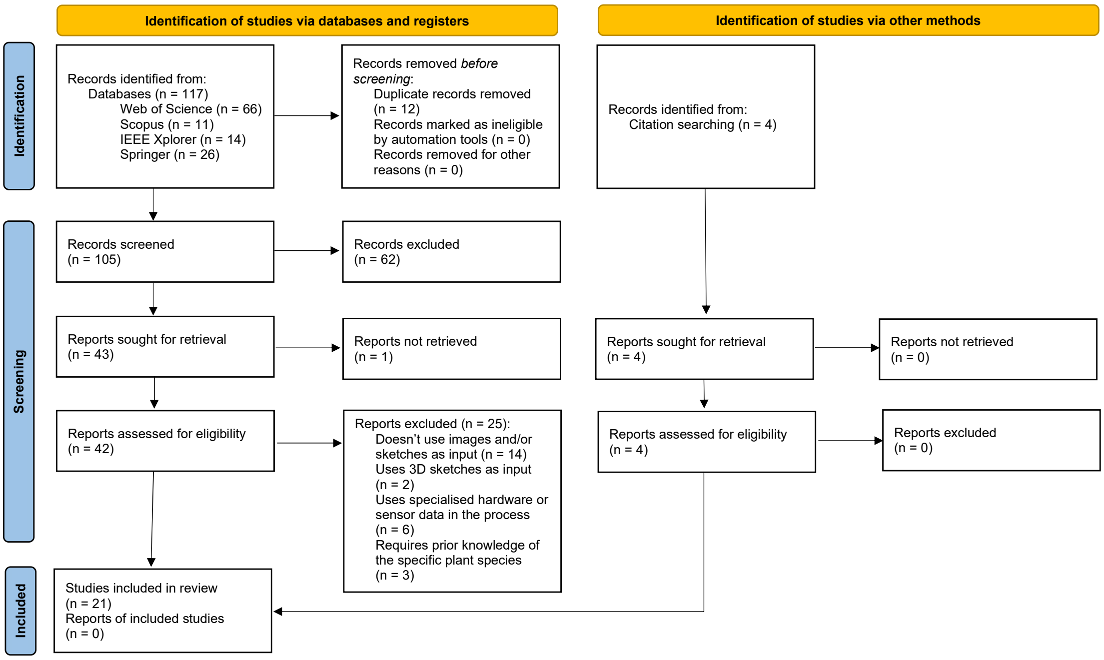


Figure 2.1: *PRISMA flow chart of study selection*

plant models from photographs or sketches. In total, 40 distinct features were identified and organized into three groups:

- *Overall Process Features:* The overall process features refer to the methods and technologies employed from input acquisition to the final output representation. These include: 3D generation type, plant type, pre-input operations, input type, foreground extraction method, data extraction technique from the input, small branch integration method, pre-modelling operations, 3D creation method, foliage and petal generation type, output format, presence of model textures, developed plant animations, and the publication date (year and month).
- *Evaluation and Results Features:* Characteristics that may be relevant to the evaluation methods and results were also noted, consisting of: number of samples used and photos/sketches used per sample, operating system, programming language, technical details (i.e. CPU, GPU and RAM), loss functions and the best, number of independent runs and maximum iterations, optimiser and its type (time, accuracy or both), evaluation type (accuracy, realism, effectiveness or usability), evaluation functions and the best, time of the process (and whether the time of the process depended on the user), and final performance value.
- *Binary Features:* A set of additional binary features (with "yes" or "no" values) was recorded, representing characteristics that may hold relevance for subsequent analyses. These included the use of external tools, inspiration from prior studies, the requirement of user actions beyond data input, the presence of a user interface, the availability of optional editing operations following model generation, the inclusion of a cost function, and any constraints on input image resolution.

When a study did not report a particular characteristic, it was considered either not reported or not applicable to the study methodology, but was still used for the analysis.

Due to the qualitative and methodological diversity of the included studies—varying widely in design and reporting standards—and because this review focused on analysing methodological patterns rather than evaluating intervention outcomes, no formal risk of bias assessment was conducted. The objective was not to synthesize quantitative results but to identify trends and shared practices in 3D plant generation approaches using photos and/or sketches.

2.2.4 Statistical analysis

As this review focused on identifying and comparing methodological features across studies rather than focusing on predefined outcomes or synthesising effect measures (e.g. risk ratios or mean differences), no summary statistics, group comparisons, or effect estimates were collected or computed.

Due to the complexity and volume of the data collected, six key features were selected as the primary basis for comparison in the cross-analysis: 3D Generation Type, Input Type, Plant Type, 3D Creation Method, Processing Time, and Evaluation Type. These features were chosen for their central relevance in capturing methodological and practical differences across studies, and were compared against all other identified features to reveal patterns and trends in the field. The corresponding values for these features are summarized in Table B.1.

In order to prepare the data for synthesis, the processing times reported in the studies were grouped into six distinct intervals to enable comparison across articles: 1) < 30 seconds, 2) < 1 minute, 3) $5-10$ minutes, 4) $20-25$ minutes, 5) $20-96$ minutes, and 6) $30-40$ minutes. Additionally, the evaluation methods described in the studies were classified into four types: accuracy, realism, effectiveness and usability. These categorisations enabled heterogeneous data to be organised into comparable groups for cross-analysis.

Cross-analysis tables were created for each of the six main features to support the comparison and interpretation of results across studies. The columns in these tables represent the distinct values of each main feature, and the rows list all the other identified features. This structure enables patterns and common approaches associated with each value of the main features to be identified visually.

Due to the absence of quantitative outcomes and the focus on a qualitative cross-analysis of descriptive features, this review did not conduct a meta-analysis, statistical heterogeneity assessment, or formal sensitivity analyses. Instead, a narrative synthesis was adopted, with patterns and correlations identified through visual inspection of cross-analysis tables based on six key features. Notable and consistent patterns were summarized in the text, emphasizing significant relationships across studies.

A formal statistical or graphical assessment of reporting bias (e.g., funnel plots) was not conducted, as the review did not involve effect estimates. Nevertheless, the potential for bias resulting from missing results was incorporated as part of the overall analysis. Rather than evaluating potential reporting biases on an individual study basis, missing information was included in the comparative analysis. The presence or absence of features was analysed to identify patterns that could be indicative of methodological trends or selective reporting.

A structured confidence assessment was not conducted. However, given the descriptive and exploratory nature of the review, patterns identified in a limited number of studies are presented with the understanding that their reliability may be affected by sparse or inconsistent reporting.

2.3 Results

The search resulted in 117 articles. Following the removal of 12 duplicates, 62 records were excluded based solely on their titles and abstracts. This process resulted in 43 articles selected for full-text review. Of these, 26 were further excluded according to the predefined exclusion criteria. An additional 4 relevant articles were identified through the reference lists of the included studies. Consequently, a total of 21 articles were retained for inclusion in the systematic review (Wang et al., 2024; Zamuda et al., 2011; A. Hu et al., 2024a; Lopez et al., 2010; Zamuda et al., 2014; Neubert et al., 2007; Zheng et al., 2011; Scharr et al., 2017; W. Hu et al., 2003; Yang et al., 2009; Teng et al., 2009; Kuwahara et al., 1995; Yan et al., 2014; Kim et al., 2014; Z. Liu et al., 2021; Ding et al., 2008b; Ijiri et al., 2005b; Okabe et al., 2007b; Tan et al., 2008; J. Liu et al., 2010; J. Liu et al., 2012). The screening process is detailed in Figure 2.1.

We excluded two studies that used 3D sketches as input (Yuan et al., 2021; Z. Liu et al., 2019), as our review focused on two-dimensional inputs only. Additionally, we excluded three studies that relied on prior knowledge of specific plant species and were tailored to those species (Guénard et al., 2013; Dror et al., 2009; Cai et al., 2012), as our aim was to include more generalisable generation approaches.

The cross-analysis was based on the same set of 21 studies and used six key features as the primary basis for comparison: 3D generation type, input type, plant type, 3D creation method, processing time and evaluation type.

Most of the studies focused on reconstruction rather than modelling. Multi-view photographs were the most common input type, with only three studies using sketches exclusively and another three using both sketches and photographs. The majority of studies focused on trees (13), while a similar number addressed general plant categories ('any plant') and flowers. Only one study targeted roots. Of the six features, the 3D creation method showed the greatest variety, with 3D skeleton-based approaches being the most frequently used. The evaluation types were primarily centred on accuracy and realism, with fewer studies addressing effectiveness and usability. Notably, a significant proportion of studies (9) did not report processing time; of those that did, the results were evenly distributed across the predefined time intervals.

The qualitative synthesis is presented in six subsections corresponding to the key features analysed: 1) 3D Generation Type, 2) Input Type, 3) Plant Type, 4) 3D Creation Method, 5) Processing Time, and 6) Evaluation Type. For each feature, a summary table and descriptive narrative outline the observed patterns and trends. Missing data was indicated by a dash ("—") in the corresponding tables.

2.3.1 3D Generation Type

3D Generation Type was considered a critical classification feature, as it separates the literature into two primary categories: reconstruction and modelling. This distinction influences a wide range of methodological and process choices. The most relevant cross-features associated with each generation type are summarised in Table 2.2.

Table 2.2: *3D Generation Type cross-relationships features.*

Feature	**Reconstruction** ($n = 16$)	**Modelling** ($n = 5$)
Input Type	Photos, 1P & MS*	Sketches, 2P & 2S*
Includes UI	✗ (11/16)	✓
Cost Function	✓ (11/16)	–
Pre-Modelling	✓ (12/16)	✓
Has Textures	✓ (9/16)	✓ (4/5)
Plant Animation	✓ (4/16)	–
Technical Parameters	✓ (3 – 9/16)	–
Optimizer	✓ (15/16)	–
Evaluation Type	Accuracy (10/16), Realism (5/16), Effectiveness (1/16)	Usability (3/5), Realism (2/5)
Processing Time	$< 30\ sec$ to $30 - 40\ min$ (8/16)	$< 30\ sec$ to $30 - 40\ min$ (4/5)

*1P & MS: 1 Photo & Multiple Sketches; *2P & 2S: 2 Photos & 2 Sketches.

While both reconstruction and modelling studies used a combination of photos and sketches as inputs, only reconstruction studies used photos alone, and only modelling studies used sketches alone. User Interface (UI) design appeared in all of the modelling studies, whereas most reconstruction papers didn't include UI elements. Cost functions were a recurring feature in reconstruction, but entirely absent in modelling.

Pre-modelling operations varied between types, though 2D skeletonization was the only shared method (4/16 reconstruction, 2/5 modelling). All modelling studies included some pre-modelling operation, while some of the reconstruction studies did not report one.

In terms of textures, around half of the reconstruction papers and nearly all modelling studies included them. Plant animation was found exclusively in reconstruction papers, which covered three types of animation (seasons, wind, and tree growth), with no modelling studies addressing it.

Certain technical implementation details, such as operating system, programming language, GPU, RAM, best loss function, number of independent runs, and maximum iterations, were reported only in reconstruction studies. Additionally, the optimizer was consistently reported in these studies.

Regarding evaluation, accuracy was the most common metric in reconstruction, while usability dominated in modelling. Realism was the only shared evaluation criterion but was not the leading metric in either category. Both generation types covered a wide span of processing times, from under 30 seconds to over 30 minutes. Most studies reported processing duration, showing that both fast and slow times occurred across both types.

2.3.2 Input Type

Input type is a defining factor in 3D plant generation, influencing both the processing pipeline and the applied techniques. Table 2.3 highlights the most relevant cross-relationships identified for each input category.

Table 2.3: *Input Type cross-relationships features.*

Feature	**1) MP** ($n = 10$)	**2) 2P** ($n = 1$)	**3) MPP** ($n = 1$)	**4) SP** ($n = 3$)	**5) MS** ($n = 2$)	**6) SS** ($n = 1$)	**7) 1P&MS** ($n = 1$)	**8) 2P&2S** ($n = 2$)
Plant Type	Any, Trees, Roots	Trees	Any	Flowers, Trees	Flowers	Trees	Trees	Trees
Uses External Tools	✗ (3/10)	✓	✓	✓	✓	✓	✓	✓
Pre-Input Operation	Turntable, Frame Extr.	–	–	Close Up	–	–	–	–
Extract Data Input	✓ (SFM, Volume, Self-calib.)	–	–	–	–	–	–	–
3D Creation Methods	3D Coord, 3D Skelet, Proc, Others	Proc	3D Coord	3D Coord, Proc, Other	Other	Other	3D Skelet	3D Skelet
Output Format	3D Mesh, Others	3D Mesh	3D Mesh	3D Mesh, Others	3D Mesh	3D Mesh	3D Mesh	3D Mesh
Has Textures	✗ (6/10)	✓	✓	✗ (1/3)	✗ (1/2)	✓	✓	✓
Plant Animation	✓ (2/10)	✓	–	✓ (1/3)	–	–	–	–
Used Samples	$1 - 5$	1	1	> 5 (1 had $\leqslant 5$)	> 5	> 5 (1 had $\leqslant 5$)	1	> 5 (1 had $\leqslant 5$)
Processing Time	$< 30\ sec$, $< 1\ min$	$< 1\ min$	$< 30\ sec$	$< 30\ sec$, $< 1\ min$	$30 - 40\ min$	$5 - 10min$	$30 - 40\ min$	$< 30\ sec$, $< 1\ min$
User-Dependent Time	✓ (2/10)	✗	✗	✓ (2/3)	✓	✓	✗	✗

MP: Multiple Photos; 2P: 2 Photos; MPP: Multiple Pairs of Photos; SP: Single Photo; MS: Multiple Sketches; SS: Single Sketch; 1P & MS: 1 Photo & Multiple Sketches; 2P & 2S: 2 Photos & 2 Sketches.

The reviewed studies employed a variety of input types, including multi-photo setups, photo-sketch combinations, and purely sketch-based inputs. Multi-photo input ($n = 10$) was the most prevalent and covered the widest range of plant types, including any plant, trees, and roots. In contrast, multi-sketch inputs were typically restricted to more specific plant categories such as flowers. Notably, only studies using multi-photo inputs had instances where external tools were not used, indicating potential for lower system complexity.

Pre-input operations varied significantly across input types. Multi-photo workflows included video frame extraction and turntable capture, while single-photo studies occasionally used close-up photography. Among all types, only the multi-photo group extracted data from the input, with Structure from Motion (SFM) and volume density estimation being the most common methods.

Regarding 3D plant generation techniques, three methods emerged as most commonly shared: 3D coordinates (used in multi-photo, multi-pair, and single-photo studies), 3D skeletons (multi-photo and photo-sketch inputs), and procedural modelling (multi-photo, two-photo, and single-photo). All input types produced 3D models or meshes, with only the multi-photo and single-photo types yielding additional output formats.

Textures were included across all input types, although studies with multi-photo, single-photo, and multiple-sketch inputs occasionally had studies without texture integration. Animation was reported exclusively in multi-photo, two-photo, and single-photo studies. Sample size usage also varied: inputs involving multiple photos often used fewer evaluation samples (typically one or five), whereas single photo and sketch-only types used larger sets, though a few exceptions used as few as five.

Processing times showed two shared fast intervals—under 30 seconds and under 1 minute—across multi-photo, single-photo, and mixed-modality inputs. Meanwhile, multi-sketch studies consistently reported longer processing times in the range of 30–40 minutes, whereas the single-sketch reported a short time of 5–10 minutes. All sketch-based input types had studies with user-dependent times, as did some photo-based inputs, suggesting variation in interaction complexity.

2.3.3 Plant Type

The plant type has been identified as a significant feature, as it provides an overview of the most studied plant species and types in the current literature. The main cross-features associated with each plant type are outlined in Table 2.4.

Studies on 3D plant generation display some variation across plant types, with the majority focusing on trees, followed by "any plant" representations, flowers, and roots. In terms of temporal distribution, studies addressing "any plant" and trees span the longest periods—21 and 26 years, respectively—and include the most recent publications (2024 and 2021). Conversely, research on roots and flowers remains scarce, with the latest studies dating back to 2011 and 2014, respectively, indicating potential gaps in these areas.

Reporting on user interaction and interface design was inconsistent across plant categories. All flower-related studies required user actions beyond initial data entry, whereas the single root study reported no such interactions. Interface components were included in some studies involving "any plant", trees, and flowers, but were entirely absent in the root-focused study. Pre-processing procedures also varied by plant type: tree studies did not employ any pre-input operations, while studies categorized under "any plant" commonly utilized video frame extraction. Turntable capture

Table 2.4: *Plant Type cross-relationships features.*

Feature	**Any Plant** ($n = 4$)	**Trees** ($n = 13$)	**Flowers** ($n = 3$)	**Roots** ($n = 1$)
Requires User Actions	✓ (2/4)	✓ (8/13)	✓	✗
Includes UI	✓ (2/4)	✓ (8/13)	✓ (2/3)	✗
Pre-Input Operation	Frame Extr., Turntable	–	Close Up (1/3)	Turntable
Pre-modelling Operation	Others	2D Skeletonization (6/13), Others	Pattern Choice (2/3), Other	–
3D Creation Methods	Instant-NGP (2/4), 3D Coord, Others	3D Skelet (6/13), 3D Coord, Proc, Others	Drag and Drop Assemble (2/3), 3D Coord	Visual Hull
Foliage/Petal Generation	Together (4/4)	Separate (9/13)	Together (3/3)	–
Output Format	3D Mesh (2/4), Volumetric, Other	3D Mesh (11/13), Triangular, Other	3D Mesh (2/3), Triangular	Volumetric
Has Textures	✓ (2/4)	✓ (8/13)	✓ (2/3)	✗
Published Date	2003 – 02/2024	05/1995 – 09/2021	07/2005 – 05/2014	2011
Independent Runs	–	User-specified (2/13), 30 (1/13)	–	–
Evaluation Type	Accuracy (4/4)	Realism (7/13), Accuracy, Usability	Usability (2/3), Effectiveness	Accuracy

emerged as the only shared pre-processing method, observed in both "any plant" and root studies.

Distinct approaches were evident in pre-modelling operations. Tree-focused studies predominantly implemented 2D skeletonization, while flower studies prioritized pattern choice. Root studies, in contrast, reported no pre-modelling operations. Methods for generating 3D plant geometry also varied considerably by plant type. "Any plant" studies frequently employed instant neural graphics primitives (Instant-NGP), tree studies combined 3D skeletons and procedural modelling, flower studies favoured drag and drop assemble, and the root study utilized a visual hull approach.

Foliage generation was typically integrated within the modelling process in "any plant" and flower studies. However, most tree studies handled foliage generation as a distinct stage. Across all categories, 3D model/mesh output formats were most commonly used, particularly in "any plant", tree, and flower studies. Volumetric representations were noted in both "any plant" and root studies, while triangular meshes were reported in tree and flower categories. Notably, only the root study lacked any form of texture integration in the resulting models.

Among the reviewed literature, only tree studies documented the number of independent runs, often allowing users to specify this parameter. Evaluation methodologies differed widely. Accuracy emerged as the most frequently reported metric across "any plant", tree, and root studies. Assessments of realism were exclusive to tree studies, whereas usability was most commonly evaluated in flower studies. The latter also included the only instance of an effectiveness evaluation.

2.3.4 3D Creation Method

The 3D creation of the plant is also a key part of the process, as it's one of the main steps, and is therefore included as one of the main columns. Table 2.5 presents the relevant features outlined.

Table 2.5: *3D Creation Method cross-relationships features.*

Feature	**1) NGP** ($n = 2$)	**2) PM** ($n = 3$)	**3) 3D CS** ($n = 3$)	**4) 3D S** ($n = 6$)	**5) VH** ($n = 1$)	**6) Oct** ($n = 1$)	**7) FG** ($n = 1$)	**8) GD** ($n = 1$)	**9) MO** ($n = 2$)	**10) DDA** ($n = 1$)
Plant Type	Any	Trees	Any, Trees, Flowers	Trees	Roots	Any	Trees	Trees	Flowers	Trees
Pre-Input Op.	Frame Extr.	–	Close Up (1/3)	–	Turntable	Turntable	–	–	–	–
Extr. Foreground	–	Ext. Crop Tool (1/3), Other	Ext. Crop Tool (1/3), Matting (1/3)	Ext. Crop Tool (2/6), Matting (1/6)	Other	Other	Other	Matting	–	–
Pre-Modelling Op.	–	Param. Opt. (2/3), Other	2D Skelet. (1/3), Others	2D Skelet. (4/6), Others	–	Voxel Grid	–	2D Skelet.	Pattern Choice	2D/3D Convex H.
Foliage/ Petal Gen.	Together	Separate (2/3)	Together (2/3)	Separate	NA	Together	Together	Together	Together	Separate
Has Textures	✓	✓ (2/3)	✓ (2/3)	✓ (5/6)	✗	✗	✗	✗	✓ (1/2)	✓
Published Date	2024	12/2011–09/2021	2003 – 2014	2011	09/2017	05/1995	06/2014	06/2014	07/2005–04/2008	08/2007
Programming Lang.	–	C++ (1/3), Python (1/3)	–	–	–	C++	–	C++	–	–
Best Loss Function	–	Fitness (2/3), Other	Min. Match (1/3), Fitness (1/3)	Min Dist. (2/6), Other (1/6)	Min. Path	–	Min. Match	–	–	–
Processing Time	20 – 25min	< 30sec	< 1min to 5 – 10min	< 30sec to 30 – 40min	–	< 1min	–	–	30 – 40min	5 – 10min
User-Depend. Time	✗	✗ (2/3)	✗ (2/3)	✗ (5/6)	✗	✗	✗	✓	✓	✓

Due to the variety of methods employed, they were numbered as follows: 1) Instant-NGP, 2) Procedural Modelling, 3) 3D Coordinate System, 4) 3D Skeleton, 5) Visual Hull, 6) Octree, 7) Fractal Geometry, 8) Gaussian Distribution, 9) Drag and Drop Assemble, and 10) Depth Modulation. Among these, the 3D skeleton method was the most frequently used, followed by procedural modelling and 3D coordinate systems. Method 3) stood out for its broader applicability, as it was the only one to support multiple plant types—including trees, flowers, and general plant forms.

Pre- and post-processing steps varied notably between methods. Video frame extraction was exclusively associated with Instant-NGP, while visual hull and octree methods consistently used a turntable setup. Foreground extraction was commonly performed using external cropping tools or matting algorithms, particularly among methods 2), 3), 4), and 8). The most widespread pre-modelling operation was 2D skeletonization, frequently used in conjunction with 3D skeletons. Some methods, such as Instant-NGP, visual hull, and fractal geometry, skipped pre-modelling steps entirely. Notably, depth modulation (method 10) uniquely employed convex hulls as a pre-modelling step.

Leaf and texture generation practices revealed further methodological distinctions. Most methods integrated leaf generation within the model, while methods 6) and 10) treated foliage separately. Method 3) again displayed flexibility, supporting both approaches. Texture usage was common in studies applying Instant-NGP, depth modulation, procedural modelling, and 3D skeletons, while older or more geometric-focused methods like visual hull, octree, fractal geometry, and Gaussian distribution did not incorporate textures. Loss functions, when present, were typically tied to optimization steps during the generation process and were absent from methods with higher user dependence or minimal automation.

In terms of implementation and performance, C++ was predominantly used, with Python appearing only in one procedural modelling study. Methods such as procedural modelling, 3D coordinate systems, octrees, and depth modulation achieved faster processing times (under 5-10 minutes), whereas Instant-NGP and drag and drop assemble had notably longer durations (over 20-25 minutes). Studies involving Gaussian distribution, drag and drop assemble, and depth modulation were also more user-dependent, typically including UI components, being inspired by previous work, and not including cost functions, whereas the methods 1), 5), 6) and 7) operated autonomously and without reference to prior studies. The most recent publications applied Instant-NGP and procedural modelling, whereas methods like fractal geometry, drag and drop assemble, and depth modulation have not been used since 2008.

2.3.5 Processing Time

Processing time was considered relevant for further analysis as it could define an important measure of performance and divide the literature into important patterns. Table 2.6 presents the identified features that were considered relevant.

Table 2.6: *Processing Time cross-relationships features.*

Feature	**1)** $< 30s$ $(n = 3)$	**2)** $< 1m$ $(n = 3)$	**3)** $5 - 10m$ $(n = 2)$	**4)** $20 - 25m$ $(n = 1)$	**5)** $20 - 96m$ $(n = 1)$	**6)** $30 - 40m$ $(n = 2)$
Input Type	MP, SP, 2P & 2S	MP, SP, 2P & 2S	MP, SS	MP	MP	MS, 1P & MS
Plant Type	Trees	Any, Trees, Flowers	Trees	Any	Trees	Trees, Flowers
Uses Ext. Tools	✓	✓	✓	✓	✓	✓
Has Textures	✓	✓ (2/3)	✓ (1/2)	✓	✓	✓
Plant Animation	–	–	–	–	–	–
Mentioned CPU	✓	✓ (2/3)	✓ (1/2)	✗	✓	✓ (1/2)
Optimizer Type	Time, Accuracy	Both	Accuracy (1/2)	Both	Accuracy, Both	–
Evaluation Type	Accuracy, Realism	Accuracy, Realism, Effectiveness	Accuracy, Usability	Accuracy	Realism	Usability, Realism
User-Dependent Time	✗ (2/3)	✗	✓	✗	✓	✓

As previously mentioned, the processing times across the reviewed studies were categorized into seven discrete intervals: 1) $<$ 30 seconds, 2) $<$ 1 minute, 3) $5 - 10$ minutes, 4) $20 - 25$ minutes, 5) $20 - 96$ minutes, and 6) $30 - 40$ minutes. A total of nine studies did not report processing time. Among those that did, the most frequently reported durations were in categories 1) and 2), each cited by three studies. Two studies reported processing times in the categories 3) and 6), while single studies were associated with categories 4), and 5), suggesting a skew toward shorter durations. All studies that reported processing times used external tools, indicating a strong association between tool integration and the ability to quantify processing durations.

Regarding input and plant types, studies using multiple photos as input spanned a range of processing times—categories 1) through 5)—demonstrating both flexibility and scalability. By contrast, studies utilizing either a single photo or a combination of two photos and two sketches consistently achieved faster processing times, limited to categories 1) and 2). In terms of plant types, trees emerged as the most frequently cited category, appearing across processing time groups 1), 2), 3), 5), and 6), while the broader category "any plant" was represented in groups 2) and 4).

All studies achieving the fastest processing time (category 1) explicitly documented the CPU specifications used, reflecting a tendency to provide detailed system context in highly optimized workflows. The only study that reported the optimizer type "time" and a processing time reached the shortest duration 1), while those using optimizers designed for "both" and accuracy exhibited times from categories 1) to 5). Studies in the longer time range (6) did not mention the use of optimizers. Texture integration was common across most studies; only categories 2) and 3) included studies without textures, indicating that textures are not a major contributor to prolonged processing times. Additionally, none of the studies that reported processing time involved plant animations, potentially due to added complexity.

Finally, differences in evaluation approaches and user involvement showed clear patterns. Studies employing realism as the evaluation criterion demonstrated a wide span of processing durations (1–6), suggesting variability in implementation complexity. The same was noted in studies focused on usability (3—6). Importantly, all studies associated with extended processing durations (5—6) indicated that the time was user-dependent. Interestingly, user dependency was also noted in some faster processing studies—specifically, those within categories 1) and 3)—highlighting that user interaction can impact performance regardless of overall system efficiency.

2.3.6 Evaluation Type

The evaluation type was identified as a key factor, given the categorisation of extant literature into four distinct evaluation purposes. This approach provides a more detailed understanding, especially given the absence of a universally applicable evaluation metric across the studies. The primary features delineated are exhibited in Table 2.7.

Distinct input types were associated with each category. Usability studies exclusively employed sketches, both single and multiple, while accuracy studies were limited to photographic inputs (single, multiple, pairs, and two-photo sets). Realism was the only evaluation type encompassing both photo-based inputs and combinations

Table 2.7: *Evaluation Type cross-relationships features.*

Feature	**Accuracy** ($n = 10$)	**Realism** ($n = 7$)	**Effectiveness** ($n = 1$)	**Usability** ($n = 3$)
Input Type	MP, 2P, MPP, SP	MP, SP, 1P & MS, 2P & 2S	SP	MS, SS
Plant Type	Any, Trees, Roots	Trees	Flowers	Flowers, Trees
Requires User Actions	✓ (6/10)	✗ (4/7)	✓	✓
Cost Function	✓ (7/10)	✓ (3/7)	–	–
Int. Small Branches	Threshold (2/10)	–	–	–
Pre-Modelling Operation	2D Skelet. (1/10), Voxel, Others	2D Skelet. (5/7), Voxel, Other	Other	Others
3D Creation Method	Proc (3/10), 3D Coord (2/10), Others	3D Skeleton (6/7), Other	3D Coord	Others
Foliage/Petals Generation	Together (6/10)	Separate (6/7)	Together	Together (2/3)
Plant Animation	Seasons, Tree Growth & Wind	Tree Growth, Wind	–	–
Used Samples	$1 - 20$	$1 - 8$	11	$1 - 12$
Technical Parameters	CPU (4), GPU (3), RAM (4)	CPU (All), GPU (1), RAM (3)	–	–
Optimizer	Adam (3/10), jDE (2/10), Others (6/10)	Others (5/7)	Other	–
Best Evaluation Function	PSNR (2/10), Fitness (2/10), Others (6/10)	–	–	User-Studies (2/3)
Processing Time	$< 30\ sec$ to $20 - 25\ min$	$< 30\ sec$ to $30 - 40\ min$	$< 1\ min$	$5 - 10\ min$ to $30 - 40\ min$
User-Dependent Time	✗ (8/10)	✗ (5/7)	✗	✓

with sketches (i.e. one photo with multiple sketches, or two photos with two sketches). The sole effectiveness study used a single photo. With respect to required user interaction beyond input provision, realism studies predominantly did not require additional user actions, whereas all usability and effectiveness studies necessitated further user involvement.

In terms of system implementation, cost function usage was primarily associated with accuracy studies, with no consistent pattern across other evaluation types. Small branch integration was solely addressed in two accuracy studies, both employing threshold techniques. Pre-modelling operations varied: 2D skeletonization was prevalent in realism studies and appeared in one accuracy study, while voxel grid pre-processing was applied in one study each from the accuracy and realism categories. Regarding 3D plant creation, realism studies favoured a 3D skeleton-based approach, which was used in the majority of the studies and exclusively within this type. In contrast, 3D coordinate systems were employed in both accuracy and the single effectiveness study. Instant-NGP and procedural modelling techniques were exclusive to accuracy-focused studies, with procedural modelling emerging as the most frequently used method in that category.

Foliage generation methods further differentiated the evaluation types. Accuracy studies often integrated foliage generation with the primary modelling process, though a subset excluded this step entirely. Conversely, realism studies typically treated foliage generation as a separate stage (6 out of 7). Regarding texture integration, accuracy studies were evenly split, while most studies from other evaluation types incorporated textures. Plant animation was generally under-represented; only two studies—

one each from accuracy and realism—addressed this feature, both implementing tree growth animations. The number of samples used varied widely within each category (e.g. 1–12 in realism, 4–20 in accuracy).

Computational specifications and optimization strategies were inconsistently reported. Only realism studies mentioned in all of the studies CPU information, while usability and effectiveness studies omitted hardware details altogether. Accuracy studies universally reported optimizers, with Adam and self-adaptive Differential Evolution (jDE) being the most common. Usability studies did not report optimizer use. Evaluation functions were limited in scope: accuracy studies employed Peak Signal-to-Noise Ratio (PSNR) and pixel-tree fitness measures, while realism and effectiveness studies did not include formal evaluation functions. Usability studies relied solely on user-based evaluations. In terms of processing time, all evaluation types reported durations up to 5–10 minutes, although one effectiveness study reached under 1 minute. Usability and realism studies showed a balanced distribution between fast and extended durations. Lastly, processing time was generally not dependent on the user across accuracy, realism, and effectiveness studies, but was user-dependent in all usability studies.

2.3.7 Common Methodological Approaches Identified

The analysis revealed distinct methodological patterns and recurring characteristics among the reviewed studies, allowing us to identify two predominant approaches within the field. These approaches, namely modelling from sketches and generation of trees from photographic data, represent foundational paradigms that can guide future research and implementation strategies.

Sketch modelling Studies utilizing sketches ($n = 3$), including Ding et al., 2008b; Ijiri et al., 2005b; Okabe et al., 2007b from 2005 to 2008, uniformly adopted a modelling-based pipeline and have shown a notable similarity in methodology. All implementations incorporated a graphical UI and a pre-modelling step, typically requiring user actions beyond the initial sketch. External tools were consistently employed, and outputs were universally in the form of 3D meshes or models. Each study used more than five input samples, and the mentioned processing time was user-dependent. Notably, none of the sketch-based studies implemented cost functions, animations, or optimizers. Furthermore, there was an absence of technical system specifications (e.g. CPU, GPU, RAM), data extraction from input sketches, or detailed modelling of fine structures such as small branches. These omissions highlight both current limitations and opportunities for extending the fidelity and automation of sketch-driven plant modelling.

Tree generation from photographic data The second prominent direction involved the generation of trees from photographic data accounted for the majority of the selected studies ($n = 12$), three of which also incorporated sketches as part of the input, spanning from 1995 to 2021 (Zamuda et al., 2011; Lopez et al., 2010; Zamuda et al., 2014; Neubert et al., 2007; Yang et al., 2009; Teng et al., 2009; Kuwahara et al., 1995; Kim et al., 2014; Z. Liu et al., 2021; Tan et al., 2008; J. Liu et al., 2010; J. Liu et al., 2012). Of these, the vast majority (10/12) focused on reconstruction, i.e. creating plant shapes or structures that resemble real plants, rather than modelling, which involves simulating the shapes and structures of non-existent plants. Tree-specific methods frequently employed 2D skeletonization as a pre-modelling operation, progressing to 3D skeleton-based reconstruction—techniques that were exclusive to this plant category. A recurrent feature was the separation of foliage generation from structural modelling, suggesting a modular approach to botanical realism. Moreover, this was the only group to explicitly report independent processing runs and to include realism evaluations, which were present in most studies (7/12). Such evaluations were exclusively associated with tree models and often accompanied CPU specifications, reflecting an emphasis on reproducibility and computational performance. These methods typically did not model fine-scale botanical features such as small branches, nor did they implement formal evaluation functions. Nevertheless, the consistency in methodology and emphasis on realism within this subgroup marks it as a particularly mature area within the broader landscape of 3D plant generation.

2.3.8 Review Outcomes

This review appears to be the first to focus exclusively on 3D plant generation from purely visual inputs, such as photographs or sketches, without relying on specialised hardware like LiDAR. It identified common methodologies and recurring patterns specific to visual-input approaches, offering a focused perspective on non-invasive and widely accessible techniques for 3D plant modelling.

Two main methodological paradigms emerged, providing a useful framework for categorising current approaches. Sketch-based systems emphasise user control and design flexibility, making them well-suited for creative and exploratory applications. In contrast, photograph-driven pipelines excel at producing realistic, high-fidelity representations, particularly of arboreal forms.

The body of included studies was relatively limited and somewhat outdated, with only three published after 2021. More recent approaches, particularly those leveraging advances in generative modelling or incorporating specialised techniques for 3D plant generation, were excluded because they relied on inputs beyond photographs or sketches and thus fell outside the scope of this project. In addition, the diversity of methodological objectives, combined with inconsistent or incomplete reporting of key features, restricted the extent of cross-study comparisons. As a result, it was not possible to formally evaluate performance metrics beyond descriptive comparisons, and the decision not to contact study authors for clarification of missing or unclear data may have contributed to information gaps.

These factors limited both the comprehensiveness and reliability of the findings and underscored the importance of more standardised reporting in future primary studies. Furthermore, the diversity and relative immaturity of existing methods, coupled with inconsistent evaluation practices, currently hinder their immediate adoption in standardised workflows. A key future direction identified through the review is the development of a unified evaluation framework that translates diverse performance indicators, such as realism, processing time, user interaction, and accuracy, into a standardised scale. Such a framework would enable more impartial benchmarking and accelerate methodological progress in this evolving field.

For the project, the review offered both a foundation and a filter for methodological choices. While photograph-based methods demonstrated clear strengths in producing realistic representations, their practical relevance was secondary. The focus instead shifted to sketch-based systems, which aligned more closely with the project's creative and design-oriented objectives. At the same time, recognising the shared limitations across both paradigms highlighted the importance of adopting robust evaluation strategies and workflows that prioritise accessibility and standardisation. In this way, the review ensured that the project was grounded in the most relevant and effective practices.

2.4 Sketch-based Modelling Approaches

Aligned with the scope of this work, we conducted a closer analysis of the sketch-based modelling approaches, which are represented by only three studies published between 2005 and 2008. This small and outdated body of work highlights a notable gap in recent research, suggesting that freehand sketch-based techniques—without reliance on external tools such as sensors—remain largely unexplored. These studies fall into two main categories: multi-sketch methods for flower modelling Ijiri et al., 2005a; Ding et al., 2008a, and a single-sketch method for tree modelling Okabe et al., 2007a, which the authors suggest could be extended to other plant types.

2.4.1 Process Analysis

To compare both approaches, we developed a visual scheme (Figure 2.2) outlining their workflows. The multi-sketch method follows an iterative process where users draw separate sketches for each plant component (e.g., petals, leaves), select structural diagrams, and assemble the model using drag-and-drop operations. In contrast, the single-sketch method begins with a single tree sketch, from which the system generates a 3D model using convex hull construction and depth modulation. While the multi-sketch method achieves flexibility through multiple user-drawn inputs, the single-sketch approach compensates for its singular input by incorporating optional editing operations throughout the process, offering a comparable degree of control and adaptability.

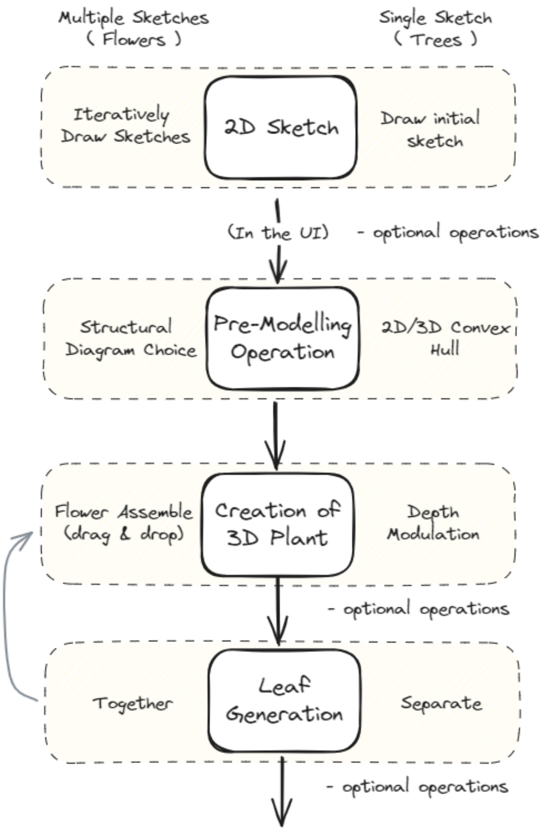


Figure 2.2: *Comparison of the process steps identified in each approach.*

2.4.2 Implementation Analysis

To further investigate the differences between approaches, we conducted a detailed analysis of their implementation strategies and technologies, summarized in Table 2.8. Although we contacted the original authors requesting source code for additional insights, no responses were received. From the information analysed, we observed that the modelling process generally involves shaping individual components (such as branches, leaves, and petals) followed by assembling the complete model. For the flower components within the multi-sketch approach, we focused specifically on sepals, petals, and stems due to their structural similarity to branches and leaves, aligning with the goal of developing generalized plant models. Furthermore, we noted that the earliest multi-sketch method (Ijiri et al., 2005a) served as a foundation for subsequent studies: the single-sketch approach (Okabe et al., 2007a) adopted its stem modulation technique to shape branch curves, while the second multi-sketch method (Ding et al., 2008a) extended this framework by introducing a merging algorithm to create solid waterproof models.

2.4.3 Strengths and Limitations

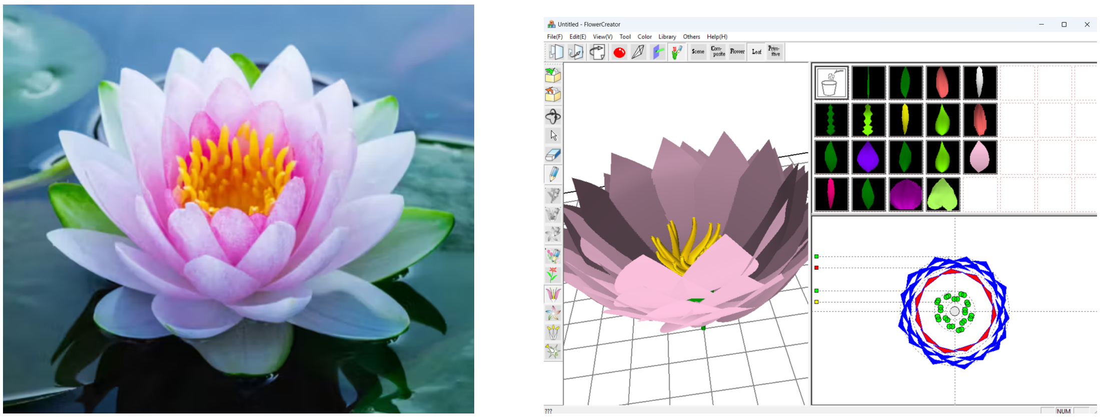
The existing approaches offer a solid foundation, featuring well-defined modelling processes and reasonable user flexibility. The single-sketch method provides a more au-

Table 2.8: *Implementation strategies and technologies employed across approaches.*

Process Step	**Single Sketch (08/2007)** (Okabe et al., 2007a)	**Multiple Sketch (07/2005)** (Ijiri et al., 2005a)	**Multiple Sketch (04/2008)** (Ding et al., 2008a)
Drawing Method	Tree structure (whole)	Flower components (parts)	Flower components (parts)
Drawing Operation(s)*	• Cut & erase stroke/branches • Define detailed shape of branches	—	—
Pre-Components Modelling Operation	3D convex hull construction	—	—
Components Modelling Operation(s)	• Child branch: length, angle and 2D distance field • Branch curves: depth modulation	• Petal & sepal: B-spline surface • Stem: energy-minimization curve reconstruction	• Petal & sepal: B-spline surface • Stem: energy-minimization curve reconstruction
Pre-Model Modelling Operation	—	Structural diagram choice	Structural diagram choice
Model Modelling Operation	Merge the two 3D trees, rotating one by 90°	Flower assemble (drag & drop) and fine-tune	Flower assemble (drag & drop), fine-tune and merging algorithm
Post-Modelling Operation(s)*	• Branch multiplication • Leaf arrangement • Branch propagation • Reproduce 3D model from overall sketching	—	—

Asterisk (*) denotes optional operations.

tomated workflow with fewer user interactions, optional operations, and an interface that adapts easily to different plant types. It also incorporates botanical terminology to enhance realism, for example, by maximizing the spacing between branches. However, the implementation details regarding leaf generation are limited, with the process described only as a ”loop stroke” rather than an open stroke, with no further elaboration. In contrast, the multi-sketch method offers a strong basis for creating accurate flower models with relative ease. Although the source code was unavailable, we accessed an executable version of the method in Ijiri et al., 2005a. The interface requires some prior familiarity, with non-English documentation and icon-based navigation, but is well-structured, uses botanical terminology, and provides effective controls. After testing, we successfully generated an accurate 3D flower model, as shown in Figure 2.3.



(**a**) *Target flower* (**b**) *Modelled Flower*

Figure 2.3: *Results of modelling the flower using the executable from Ijiri et al., 2005a.*

3

Framework Proposal

Building on the systematic review presented in Chapter 2, we organized and designed a framework proposal that combines the most suitable features and technologies identified. An article describing this proposal was submitted and accepted for the 2025 edition of the International Conference on Graphics and Interaction (ICGI)[1] as part of the Conference track.

3.1 Framework Design

The proposed framework is designed to generate 3D plant models directly from freehand sketches provided by the user, without the need for supplementary input from devices such as sensors. Building on the comparative analysis of existing approaches, it selectively integrates the most effective features from prior methods to offer a flexible, intuitive, and generalizable modelling solution. To illustrate its functionality, we present two distinct usage scenarios and detail the corresponding design workflows.

3.1.1 Workflow

Building on the approaches discussed in Section 2.4, as well as the comparative overview in Figure 2.2 and Table 2.8, we combined elements from both methodologies to propose a generalized and efficient workflow from the initial sketch to the generation of a 3D plant model. Users begin by creating a complete 2D sketch of the plant, following the principles of the single-sketch method Okabe et al., 2007a, while incorporating leaf generation alongside structural modelling, as in Ijiri et al., 2005a; Ding et al., 2008a. Once the sketch is complete, the model can be generated with a single button click, as in Okabe et al., 2007a. To enhance control and realism, leaf curvature can also be specified, following the approach proposed in Ijiri et al., 2005a.

---

[1] https://gpcg.pt/icgi2025/

3.1.2 Use-Case Scenarios

All of the analysed approaches rely on stand-alone applications. However, we aim to prioritize usability and integration with modern 3D modelling tools, given their rapid development and the outdated nature of the existing approaches. Blender was selected as a potential use-case scenario because it is widely used, open-source, and benefits from continuous advancements in modelling capabilities. Both workflow scenarios share the same initial steps, starting with a 2D sketch and culminating in the creation of a 3D plant model.

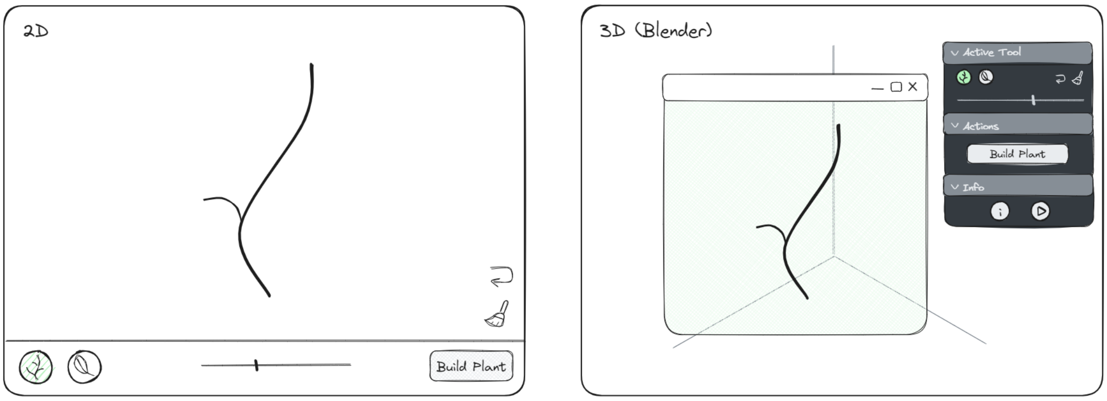
The key difference between the two workflows occurs in the post-sketch-to-model phase. In the Blender add-on, the process ends once the 3D model is created, with any further editing handled using Blender's native tools. By contrast, the stand-alone application provides additional optional adjustments, as illustrated in Figure C.1 in the appendix. These adjustments include modifying branch size, rotating branches along the X, Y, and Z axes, and applying textures using predefined or custom colours, or uploaded image textures. After making adjustments, users can preview the model, download it via a downward arrow icon, access a library of previously generated models through the plant icon, or return to the start page to begin a new modelling session.

3.1.3 Prototypes

Based on the proposed workflow and usage scenarios, we designed an initial prototype of the User Interface (UI) for both the stand-alone application and the Blender add-on.

Inspired by the dedicated tab used in previous approaches Ijiri et al., 2005a; Ding et al., 2008a, our UI includes two interaction modes: "Branch" and "Leaf".

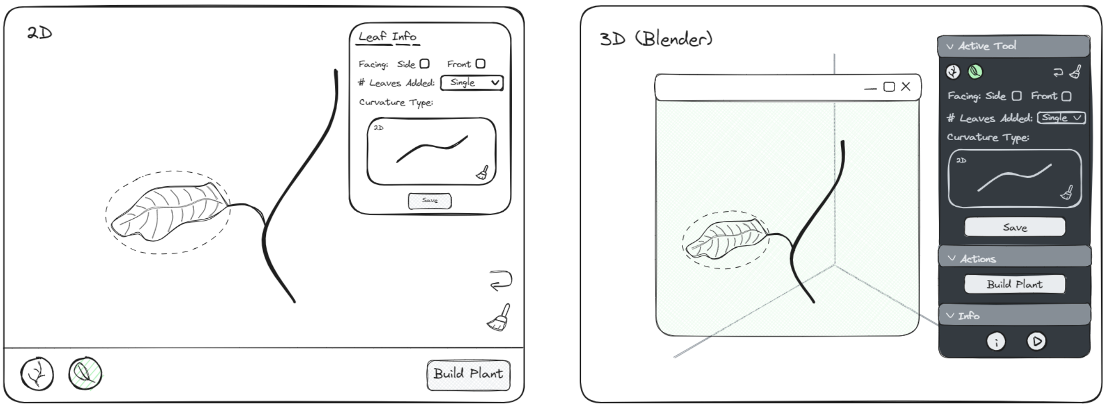
In "Branch" mode, shown in Figure 3.1, users sketch the overall branch structure, with brush size controlling branch thickness. Each new stroke is automatically connected to the existing structure and saved in real time, following the single-sketch approach.



(a) *Stand-alone application* (b) *Blender add-on*

Figure 3.1: *Branch panel design.*

In "Leaf" mode, illustrated in Figure 3.2, users draw individual leaf strokes and specify the type of curvature. To increase flexibility and capture more modelling data, we added a checkbox to indicate whether the leaf is viewed from the side or front, and a selection box labelled "# Leaves Added" to specify whether a single leaf or multiple leaves are being added. Once defined, the leaf data is stored by clicking the "Save" button.



(a) *Stand-alone application* (b) *Blender add-on*

Figure 3.2: *Leaf panel design.*

After completing the 2D sketch, users click the "Build Plant" button to automatically generate the corresponding 3D plant model, consistent with the single-sketch workflow.

To address usability limitations identified in the multi-sketch executable of Ijiri et al., 2005a (discussed in Section 2.4.3), we added two additional UI elements:

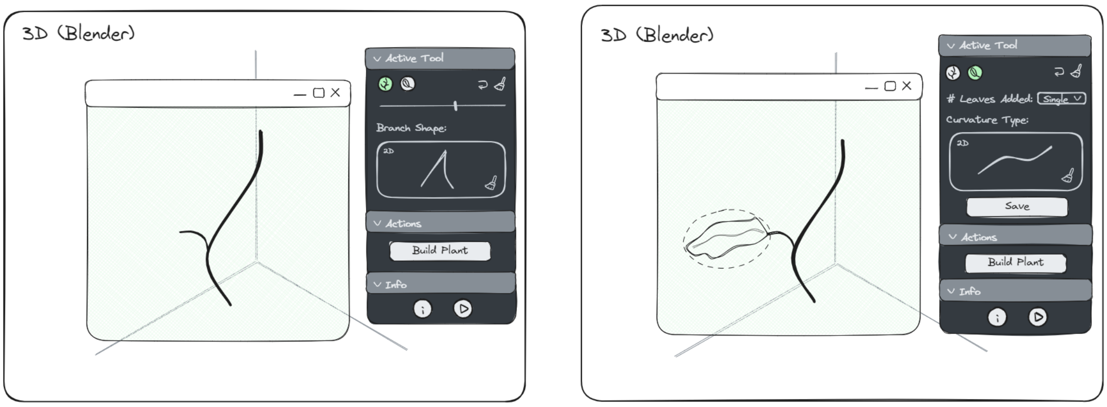
- An information button ("i"), providing a step-by-step guide to the modelling process.
- A video button (arrow icon), which plays a tutorial demonstrating how to use the application.

The framework also supports optional operations, including undoing a stroke via $Ctrl + Z$ or the left-looping arrow button, and erasing all strokes using a button resembling a broom.

3.2 Framework Implementation

Based on the scenarios described above and the proposed workflow, the Blender add-on scenario was chosen due to its broader usability and the available development time. As Blender already provides robust tools for 3D model adjustments, the additional features of a stand-alone application were deemed unnecessary. Since Blender add-ons are exclusively programmed in Python, the project will be implemented in this language.

We selected the most suitable technologies from those discussed in Section 2.4 and adapted the UI design and workflow steps accordingly. The updated interface, shown in Figure 3.3, introduces an optional feature in the “Branch” panel for drawing the branch shape, inspired by Okabe et al., 2007a. In the “Leaf” panel, the frontal/side checkbox was removed, as it is incompatible with the chosen leaf generation method, and each leaf is now represented by three strokes instead of an arbitrary number.



(**a**) *Branch panel design* (**b**) *Leaf panel design*

Figure 3.3: *Blender add-on revised design for implementation.*

3.2.1 Implementation Workflow

The following steps outline the implementation workflow, highlighting the technology choices and the approach each step is based on.

1. **2D Sketch Input** (Single & Multiple Sketch-inspired)
   The process begins with the user drawing a 2D sketch representing the overall structure of the plant, including both branches and leaves.
2. **Build Plant** (Single Sketch-inspired)
   Clicking the *“Build Plant”* button triggers the sketch-to-model process.
3. **3D Convex Hull Construction** (Single Sketch-inspired)
   Inspired by the pre-modelling step in Okabe et al., 2007a, a 2D convex hull is first generated from the user’s sketch, followed by the creation of a 3D convex hull to define the spatial boundary of the plant structure.
4. **Trunk and Main Branches Modelling** (Single Sketch-inspired)
   The initial strokes are projected onto a base plane at a fixed depth ($y = 0$). A 2D distance field is then computed to maximise the spacing of branches within the 3D convex hull. After determining the new positions, the branches are projected from their start points to the updated end points, resulting in an evenly distributed, planar branch layout within the 3D space.
5. **Child Branch Modelling** (Single Sketch-inspired)
   For each child branch, the process involves:

   (a) Calculating branch length
   (b) Determining branching angle

   (c) Using a 2D distance field to infer the y-coordinate
   (d) Projecting the branch into 3D space

6. **Branch Curves Modelling** (Multiple Sketch-inspired)
   Although two of the reviewed studies used depth modulation with energy minimisation for modelling branch curves, the reference article describing this method was unavailable. Given the available development time, it was decided that swept surfaces, as proposed in Ding et al., 2008a, represented a more practical alternative. Each branch is defined by a 2D stroke and brush size, as well as an optional user-drawn shape, which enables variation in thickness and curvature across branches.
7. **Leaf Modelling** (Multiple Sketch-inspired)
   As mentioned in Section 2.4.3, the multi-sketch method provides a more detailed modelling process for leaf components (sepals and petals). Consequently, its approach is considered more suitable for guiding the modelling process. Leaves are therefore generated using B-spline surfaces, guided by three user-drawn 2D strokes (two outer contours and one inner line), as well as the drawn curvature type. Although the multi-sketch approach allows for either two or three strokes (e.g. using only two for petals), as we are not modelling petals, our method always assumes three strokes. In cases where leaf positions overlap, we intend to determine their depth order in the final model based on the drawing sequence, with leaves drawn most recently rendered on top.
8. **Normal Maps** (New Feature)
   To enhance realism and reduce manual effort, a set of predefined normal map presets (such as bark and leaf textures) are available to users. These maps may be generated using well-known tools like CrazyBump[2], xNormal[3], or ShaderMap[4].
9. **Level of Detail (LoD) Variations** (New Feature)
   Implementing multiple LoD is essential for balancing visual fidelity with computational performance, especially when dealing with large quantities of plant models. By offering adjustable LoDs, users can customize the complexity of each model to suit the specific needs of their target scenario. One notable tool for vegetation optimization is Simplygon[5], which offers features such as billboard cloud generation and has been integrated into Blender as an add-on. However, its use is limited by licensing, as it has not been freely available since January 2017. Given the choice of Blender add-on scenario, this framework can leverage Blender's built-in modifiers, such as Remesh and Decimate, to generate multiple LoD versions of the generated model.
10. **3D Model Variations** (New Feature)
   Visually distinct variations of the final 3D plant model are created by making

---

[2] http://www.crazybump.com/ (accessed July 18, 2025)
[3] https://xnormal.net/ (accessed July 18, 2025)
[4] https://shadermap.com/home/ (accessed July 18, 2025)
[5] https://www.simplygon.com/ (accessed July 18, 2025)

small adjustments to the saved generation parameters, offering users greater creative freedom and design flexibility.

Common exportable formats for 3D models, such as `.obj` and `.mtl`, which are also used in the approaches reviewed, ensure broad compatibility. However, since our framework is implemented as a Blender add-on, users do not need to handle exporting, as the generated models are immediately accessible within Blender.

4

Implementation

In this chapter, we detail the implementation process of the proposed framework. We describe the development environment, discuss the challenges encountered while developing the user interface in Blender, present the resulting User Interface (UI) components, and provide an overview of the plant modelling process. All code for the add-on, along with a demonstration video, is available at the GreenCanvas Github repository[1].

4.1 Development Environment

For the development of the Blender add-on, Visual Studio Code was used together with the *Blender Development* extension by Jacques Lucke. Because the code was developed on Windows Subsystem for Linux (WSL), Blender 3.0 was chosen for implementation, as WSL could not display the user interface of newer Blender versions. Nevertheless, the add-on was regularly tested on newer versions by installing the zipped add-on and running local feature tests. This iterative approach ensured compatibility with both older and newer Blender versions, despite differences in variables and functions between releases.

An agile development methodology was adopted, with the work divided into sprints. Tasks were planned and implemented incrementally, and features were delivered at the end of each cycle. Weekly meetings were held to review progress, rearrange tasks, and plan the subsequent sprint.

As Blender add-ons are exclusively supported in Python, the add-on was implemented in Python. To support certain features, external dependencies such as SciPy and NumPy were included in the project. Blender allows this by downloading the corresponding wheel files locally and specifying their paths in the Blender manifest file. To ensure cross-platform compatibility, wheel files were prepared for Windows, macOS, and Linux, and a separate `ZIP` package of the add-on was built for each platform using the following command:

---

[1] https://github.com/anamsmartins/GreenCanvas

```
blender --command extension build --split-platforms
```

4.2 UI Implementation Challenges

Blender's Python Application Programming Interface (API) is highly restrictive in terms of available components and their customization; for example, it does not allow changing the colour of a button. Consequently, additional challenges arose that would not have been encountered when developing a stand-alone application. Several approaches were explored in an attempt to include a drawing canvas within the add-on:

- **Blender built-in floating components:** Several of Blender's built-in floating components were evaluated, but all presented limitations. For example, *QDialog* and *QPopup* were not draggable and disappeared when clicked outside. *QEventLoop* and *Modal Operators* were also did not produce draggable panels and did not function as intended.
- **External UI frameworks:** Attempts were made to create draggable windows using external frameworks, namely *PyQt5* and *Tkinter*. While draggable windows could be created with both frameworks, integrating them into Blender caused the application to freeze, as each framework required its own process. When executed as a subprocess to allow Blender to run concurrently, the external windows would lose focus or close when interacting with Blender's panels. Attempts to maintain the windows always on top were unsuccessful.
- **Duplicate Blender Window:** An attempt was made to duplicate a Blender window and configure it as a canvas, similar to the *2D Animation* mode used when starting a new project. However, it was not possible to set up the window programmatically, and the same issues encountered with external frameworks persisted: the duplicated window would close when interacting with the add-on's panels.

Fortunately, we found a GitHub repository[2] containing custom UI widgets for Blender implemented using Blender's GPU shaders. Although the repository was somewhat outdated and targeted Blender 2.8, we adapted the code for our purposes, enabling the creation of a draggable window and the development of new UI widgets, such as a Canvas widget, to suit our requirements.

4.3 UI Components

The developed Blender add-on consists of five main UI components: the Main Drawing Canvas, the Actions Panel, the Active Tool Panel, the Plant Properties Panel, and the Information Panel. In this section, we present and describe the features included in each component.

---

[2] https://github.com/jayanam/bl_ui_widgets

4.3.1 Main Drawing Canvas

As previously mentioned, the *Main Drawing Canvas* is a fully customized widget created using Blender's GPU shaders. It features a greyish header bar at the top, which allows the canvas to be moved when clicked and dragged. The canvas is shown in Figure 4.1.

We initially attempted to use Blender's Grease Pencil (gpencil) tool for drawing on the canvas due to its convenient drawing capabilities. However, the strokes did not appear on top of the canvas widget. This issue occurred because the canvas is rendered using Blender's GPU shaders in 2D space, while the Grease Pencil operates in the 3D view space.

Consequently, to achieve smooth, continuous strokes on the canvas, we implemented a centripetal Catmull-Rom spline. This method generates a curve that passes through a series of control points, producing natural transitions between them. The curve is constructed by interpolating intermediate points, resulting in a dense set of points that can be rendered as a continuous stroke.

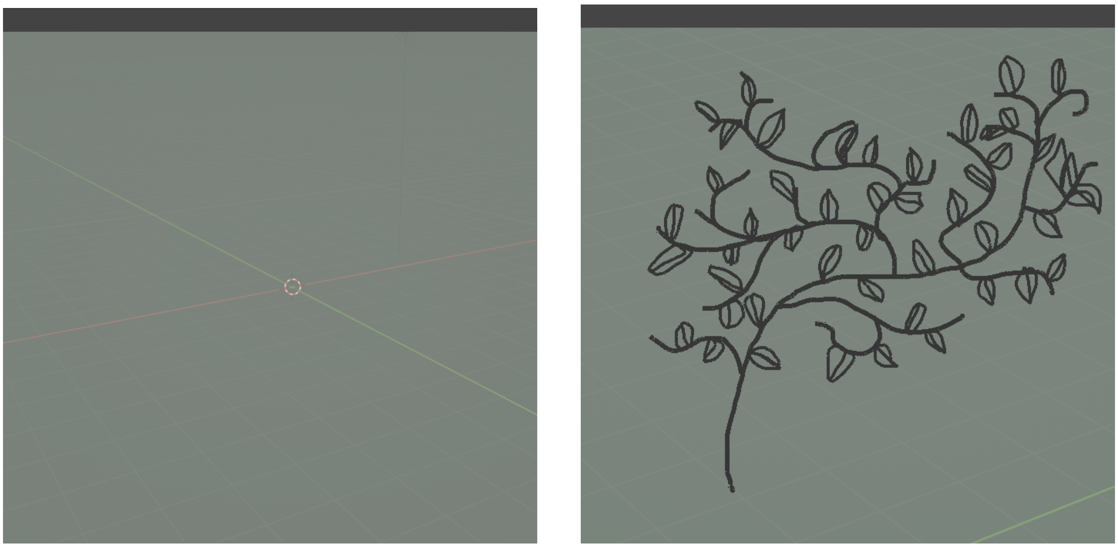


Figure 4.1: *Main drawing canvas of the add-on.*

4.3.2 Actions Panel

The *Actions Panel* updates dynamically according to the current drawing or modelling phase, presenting only the actions relevant to the next stage.

When the user initiates the add-on (see Figure 4.2), the panel displays a single *Start Drawing* button, which launches the main drawing canvas and activates the *Active Tool Panel*.

During the drawing stage (see Figure 4.3), the panel provides:

- A *Hide/Show Canvas* button to control the visibility of the main drawing canvas.
- A *Vary brush size naturally* checkbox, which overrides the user-specified brush size and varies branch thickness naturally along the parent branch to produce a

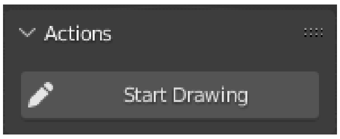


Figure 4.2: *Actions panel — Start drawing.*

more realistic appearance in the modelling process.This option is implemented as a checkbox to give users greater freedom when modelling, since the scope also includes imaginary plants.

- A select box with the options *Accurate* and *Realistic*, which determines the modelling mode used when generating the 3D plant model. In *Accurate* mode, the drawn branches preserve their original lengths and angles. In *Realistic* mode, branch lengths and angles are adjusted according to computed constraints for a more natural appearance, following the approach of Okabe et al., 2007a, which uses the length formula from Weber et al., 1995.
- A *Build Plant* button, which becomes active after at least one stroke has been drawn on the main canvas and initiates the modelling process to generate the corresponding 3D model. This design provides users with additional control over the modelling outcome.

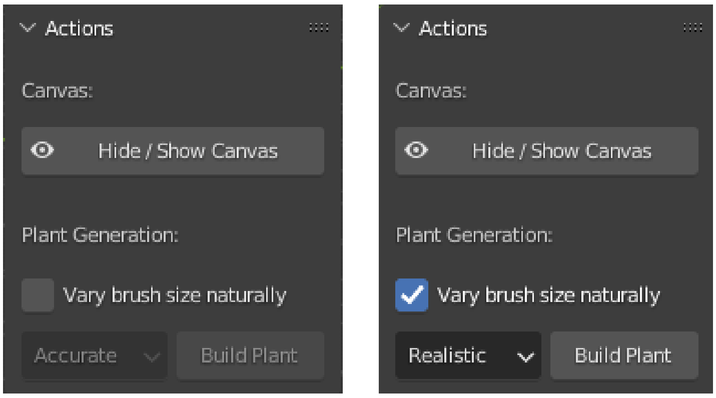


Figure 4.3: *Actions — Build plant.*

Finally, the panel allows the user to start a new drawing (see Figure 4.4) by providing a *New Plant / Drawing* button to begin a new sketch.

Figure 4.4: *Actions panel — New drawing.*

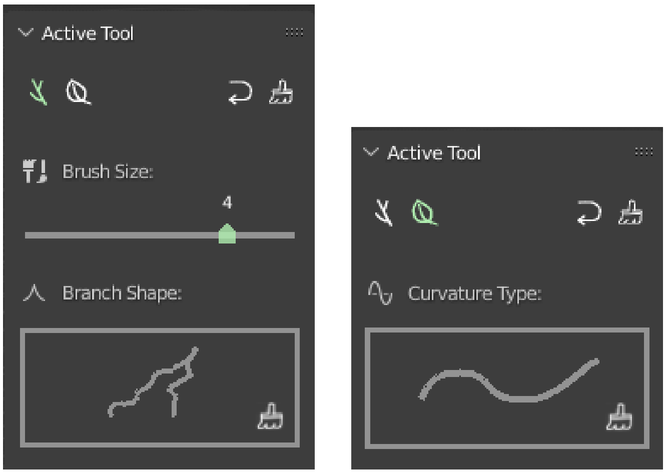
4.3.3 Active Tool Panel

The *Active Tool* panel is displayed only while the user is drawing a plant, that is, after clicking the *Start Drawing* or *New Plant/Drawing* buttons. The panel features a header containing four buttons: two on the left to switch the active tool to either *Branch* or *Leaf*, and two on the right to undo (also available via `Ctrl+U`) or clear the current strokes of the selected tool from the main drawing canvas.

When the **Branch Tool** is active (see Figure 4.5a), the panel provides two controls:

- A slider for adjusting the *brush size*, which determines the thickness of the modelled branches.
- An optional *branch shape* canvas, where the user can draw the desired shape with two strokes. This section also includes a clear button to erase the drawn shape.

When the **Leaf Tool** is active (see Figure 4.5b), the panel provides a single canvas for defining the *curvature type* of the leaf, which determines how the leaf will curve or fold and must be specified before a leaf can be drawn.



(**a**) *Branch view* (**b**) *Leaf view*

Figure 4.5: *Active tool panel.*

4.3.4 Plant Properties Panel

The *Plant Properties* panel is displayed only after the plant has been generated using the *Build Plant* button. It provides three optional operations, as shown in Figure 4.6:

- A slider to adjust the *Level of Detail* (*LoD*) of the currently selected part of the plant (branch or leaf), implemented using Blender's Decimate modifier. Figure 4.7 shows an example of its effect.
- A *Propagate Leaves* section with a select box offering the options *Realistic* and *Random*. In *Realistic* mode, the number of leaves is calculated according to the formula in Weber et al., 1995, leaves are evenly distributed along the branch, and their size decreases toward the branch tip and increases toward its base. In *Random* mode, a number between 8 and 16 is selected randomly, and leaves are distributed randomly along the branch. Providing two modes allows users greater design flexibility. Figure 4.8 provides an example of this behaviour.

- A *Generate Plant Variation* button, which generates a variation of the most recently created plant by modifying the size and positions of branches and leaves. An example result is shown in Figure 4.9.

Although we mentioned in Section 3.2 that normal maps would be included, we were unable to implement them in the add-on due to time constraints.

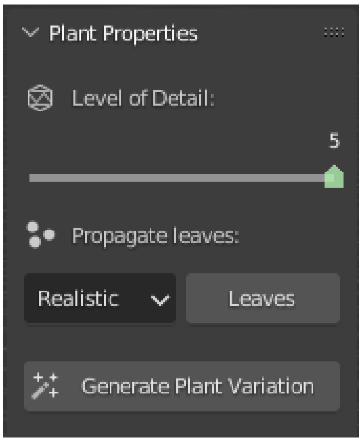


Figure 4.6: *Plant properties panel.*

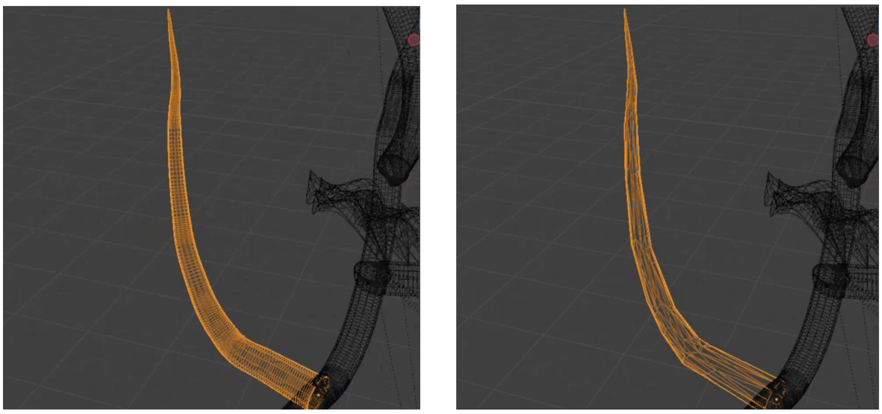


Figure 4.7: *Different LoD example.*

4.3.5 Information Panel

The *Information* Panel, shown in Figure 4.10, contains two buttons and is primarily designed to guide the user through the workflow. The *Help* button opens a floating panel displaying the general process steps, as shown in Figure 4.11a. The *Video* button (see Figure 4.11b) opens a panel containing a link to the GitHub page of the add-on, which also hosts a demonstration video. We chose not to include a direct hyperlink within the panel to avoid requiring additional internet permissions, and embedding the video directly was not possible due to Blender UI limitations.

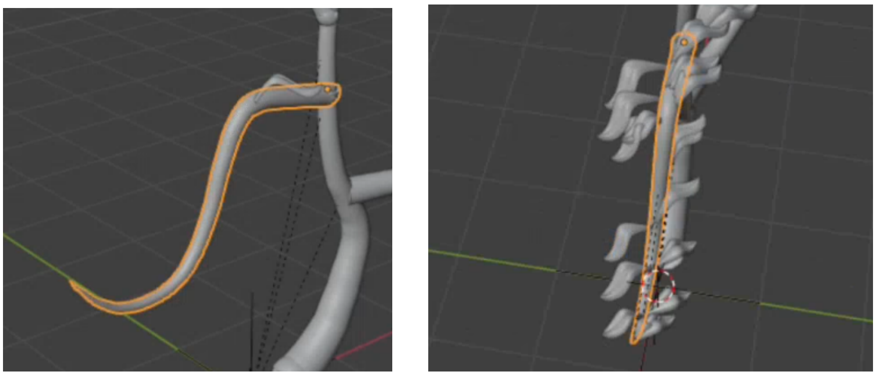


Figure 4.8: *Leaves propagation example.*

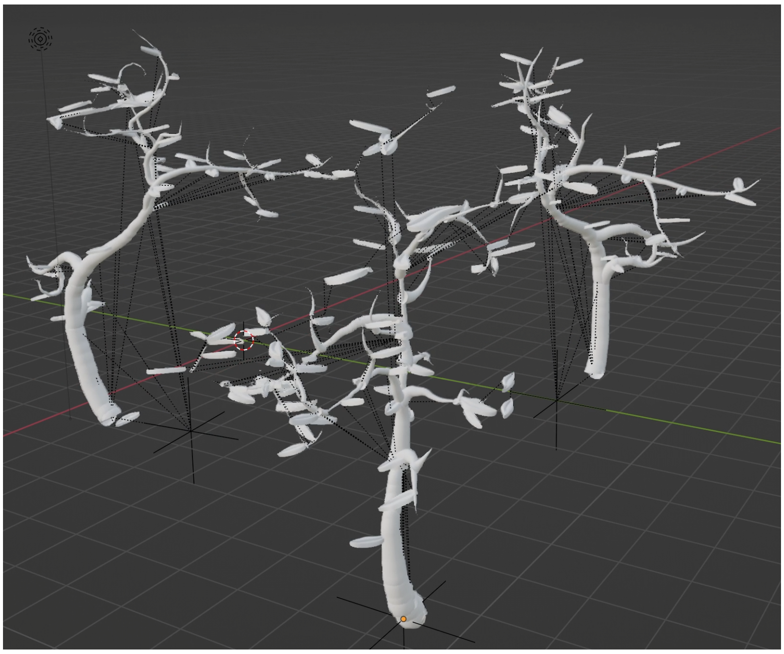
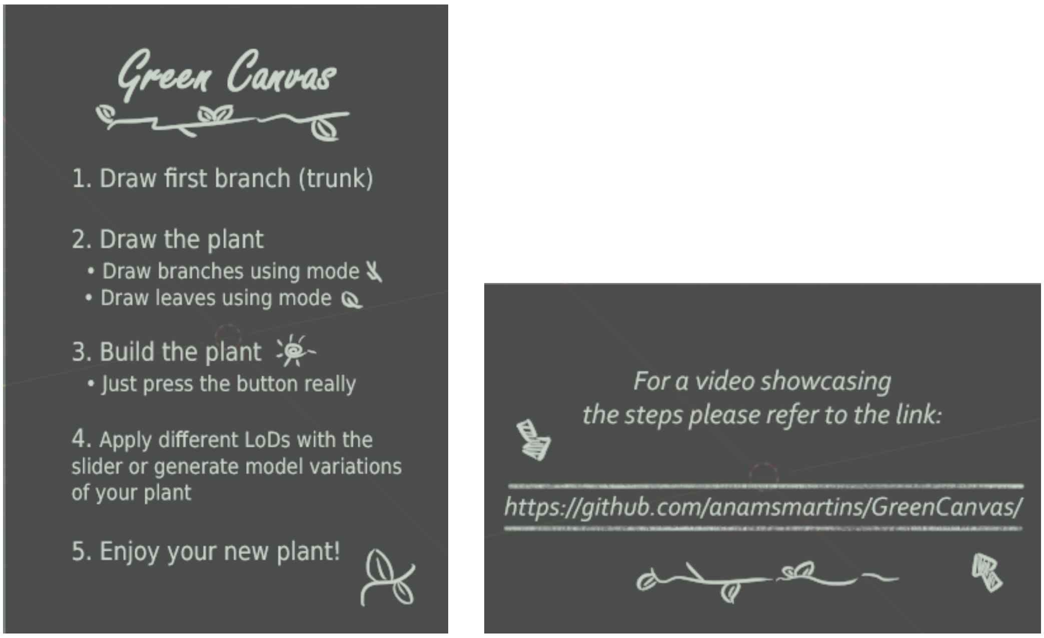


Figure 4.9: *Plant model variations.*

Figure 4.10: *Information panel.*



Green Canvas

1. Draw first branch (trunk)
2. Draw the plant
   - Draw branches using mode
   - Draw leaves using mode
3. Build the plant
   - Just press the button really
4. Apply different LoDs with the slider or generate model variations of your plant
5. Enjoy your new plant!

For a video showcasing the steps please refer to the link:

https://github.com/anamsmartins/GreenCanvas/

(**a**) *Help panel* (**b**) *Video panel*

Figure 4.11: *Information button panels.*

4.4 Plant Modelling Process

The plant modelling process begins when the user clicks the *Build Plant* button. An informative panel is then displayed to provide feedback to the user, as the modelling process can take some time (see Figure 4.12).

Figure 4.12: *Building plant panel.*

First, the branch and leaf coordinates are normalized from the canvas size to the corresponding range of Blender's 3D view space. The y-coordinates are swapped with the z-coordinates, since in 3D space the y-axis represents depth.

If the selected mode is *Realistic*, the length and angle constraints for each branch are computed using the formulas in Weber et al., 1995. The branch coordinates are updated to match these new values, and all child branches and the corresponding leaves are adjusted accordingly. Performing this step prior to constructing the convex hull, rather than restricting the search within the hull during the 2D distance field computation, resulted in faster and more efficient model generation.

Next, an initial 2D convex hull is constructed from the x–z coordinates of the drawn plant branches, following the approach described in Okabe et al., 2007a. The Shapely library was used for this purpose. A 3D convex hull is then generated by sweeping circles along the z-axis, with a radius equal to `(x_max - x_min)/2` of the 2D convex hull at each height. The radius is subsequently scaled by a constant factor (we used $\sqrt{2}$ as in Okabe et al., 2007a), and sample points are distributed along it.

We then initialize a 2D distance field as a 128×128 matrix (using NumPy), following Okabe et al., 2007a, and fill it with zeros. A trunk position is chosen near the center of the defined 3D view range (introducing a small degree of randomness) and this position is converted and added to the 2D distance field. All trunk child branches and leaves are updated accordingly.

Once the 2D distance field is set up, the positions of all branches are computed, constrained only to ensure that new branch tips remain inside the 3D convex hull. For each new position, branch coordinates are updated to match the new tip, and all child branches and leaves are updated as well.

Finally, the plant mesh is constructed. Two parent objects, *Plant* and *Branches*, are created, and the process is divided into two stages: branch modelling and leaf modelling:

- **Branch modelling:** We iterate over each branch, identify its parent (if it is not the trunk or a main branch), and assign a name to the new branch object. If the option to vary branch thickness naturally is enabled, the new thickness of the current branch is computed based on its offset along the parent. The branch mesh is then created. This always begins by generating a Bézier curve from the drawn stroke. If no branch shape was drawn, a curve object with a taper profile (polyline curve) is created instead; its bevel depth is set to `brush_size * 0.05`, the taper object is assigned, and any holes at the tips are capped. If a branch shape was drawn, a mesh is procedurally built from the two strokes, creating circular cross-section rings between them, which are then deformed and scaled along the Bézier curve to match the input stroke path using a curve modifier. After generating the branch mesh, the parent branch object is assigned as the parent, the pivot is set to the base of the branch, and a decimate modifier is added to the mesh.
- **Leaf modelling:** We iterate over each leaf and locate its corresponding branch. The leaf's two outline strokes, inner stroke, and drawn curvature profile are resampled to equal length. The depth of each point is computed from the curvature profile, and a grid of vertices is interpolated between the outline strokes toward the inner stroke with the respective depth at each point. We initially planned to use B-spline curves for this effect, but due to integration issues in Blender and time constraints, we opted for this simpler approach. Finally, the parent object is assigned as the branch, the pivot is set to the base of the leaf, and the leaf is rotated to lean more horizontally along the branch (rather than facing the camera vertically), resulting in a more natural appearance.

Some examples of resulting plant models generated using different modes, including *Accurate*, *Realistic*, and *Vary branch thickness naturally*, are shown in Figure 4.13.

Figure 4.13: *Examples of resulting plant models.*

5

Tests and Results

This chapter presents the procedures used to evaluate the developed Blender add-on and the results that were obtained. Usability was assessed using the System Usability Scale (SUS) questionnaire, and task performance was measured by the time participants required to model plants. Only descriptive statistics and relationships between variables are reported here. Interpretation of these results is provided separately in Chapter 6.

5.1 Test Setup

This section details the organization of the study, including participant recruitment, the collection of demographic information, and the design of tasks used to evaluate usability and efficiency.

5.1.1 Participants and Recruitment

The original recruitment plan aimed for approximately forty participants distributed across several groups:

- 10 non-artistic users with a mouse
- 10 non-artistic users with a pen tablet
- 10 artistic users with a pen tablet
- 10 game-related users with a pen tablet

Because of time and location constraints, we could only arrange thirty-two participants to complete the test. One respondent who entered the same score for all ten SUS questions was excluded from the analysis, leaving thirty-one valid responses. The final participant distribution resulted in:

- 2 non-artistic (technological and/or scientific) users with a mouse
- 12 non-artistic (technological and/or scientific) users with a pen tablet
- 8 artistic users with a pen tablet (of which 3 were game-related)

- 15 game-related users with a pen tablet (considering the ones with both technological and artistic backgrounds)

5.1.2 Demographic Data Collected

Each participant also provided demographic information such as age, gender, academic qualifications, main background (artistic, technological, scientific, humanities, education, health, or other), level of experience with game development (professional, amateur or hobbyist, student, or no experience), familiarity with 3D modelling software (rated from very unfamiliar to very familiar), primary drawing input device (mouse or pen tablet), frequency of use of the input device on a five-point scale, and optional comments.

5.1.3 Testing Procedure and Tasks

The study was conducted either locally or remotely using a Google Forms questionnaire, which provided installation instructions for the add-on, including a downloadable `ZIP` file for all supported platforms, and presented all test tasks and questions. Participants received no additional guidance. The evaluation consisted of four tasks designed to measure task completion times and assess usability through direct interaction with the add-on.

All tests were run in Blender version 4.5. Participants were instructed to self-record start and end times for each task and to complete the SUS questionnaire immediately after finishing all tasks. The tasks were as follows:

- **Task 1**: Using the GreenCanvas add-on, draw a plant with branches and leaves by changing the active tool in the side panel as needed. Each leaf requires three strokes (one inner and two outer). Once you have finished drawing, select Accurate mode and click 'Build Plant'.
- **Task 2**: Start a new drawing process by clicking 'New Plant / Drawing'. This time, select Realistic mode and check the 'Vary brush size naturally' box. Try to represent a natural-looking plant with leaves. When finished, click Build Plant.
- **Task 3**: Try out the following add-on features and spend a few minutes experimenting with each feature:
  - Model branches with a defined branch shape (created from two strokes);
  - Adjust the Level of Detail (LoD) slider and observe the changes using Blender's Viewport Shading mode;
  - Use Propagate Leaves on a selected branch (check out both Realistic and Random modes);
  - Generate different versions of your plant with Generate Plant Variations;
- **Task 4**: Create any kind of plant you can imagine, using any modes or properties you like. Feel free to experiment and be creative!

5.2 Results

This section reports the results of the evaluation, starting with task performance metrics, followed by usability (SUS) scores, and concluding with further analyses relating task performance and demographic data to SUS scores.

5.2.1 Task Completion Times

Table 5.1 presents summary statistics for the task completion times. Times are reported in minutes. Task 1 exhibited the greatest variability and included an extreme outlier with a reported time of 76 minutes. Tasks 1 and 3 generally took longer, whereas Task 2 had the shortest average time.

Table 5.1: *Measured statistical values for task completion times.*

Task	Mean	Median	Standard Deviation	Minimum	Maximum
1	9.48	5	14.19	1	76
2	4.10	3	2.49	0	10
3	7.29	7	4.03	2	19
4	4.97	4	3.90	1	19

5.2.2 SUS Scores

SUS scores were computed for each participant, and summary statistics are presented in Table 5.2. The mean SUS score recorded was 56.4, which is below the commonly accepted benchmark of 68 (Lewis et al., 2018). The observed scores varied substantially, as reflected by the high standard deviation (22.8). Scores ranged from a minimum of 10 to a maximum of 100, indicating considerable diversity in participants' experiences.

Table 5.2: *Measured statistical SUS values.*

Statistical Measure	Value
Mean	56.4
Median	60.0
Standard Deviation	22.8
Minimum	10
Maximum	100

5.2.3 Correlations with Numerical Variables

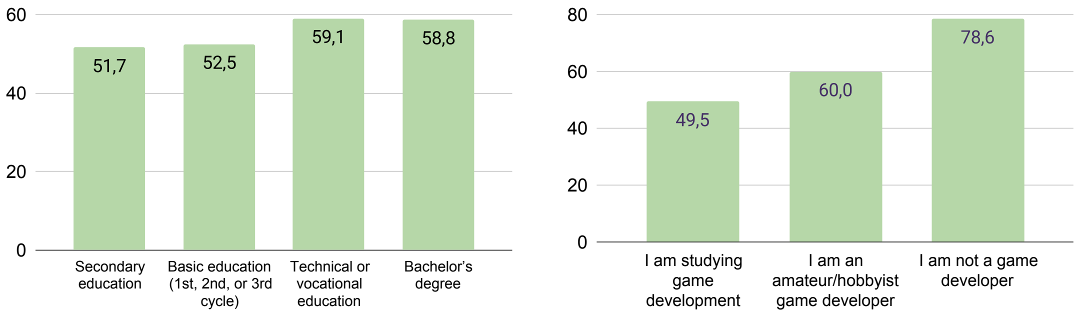
Correlations between SUS scores and task completion times, as well as between SUS scores and numerical demographic variables, were computed using the Google Sheets `=CORREL` function. The results are presented in Table Table 5.3. Negative correlation values were observed for task times, with the strongest association occurring for Task 2 ($r = -0.60$), whereas demographic variables showed positive correlations with SUS scores.

Table 5.3: *Measured correlation values.*

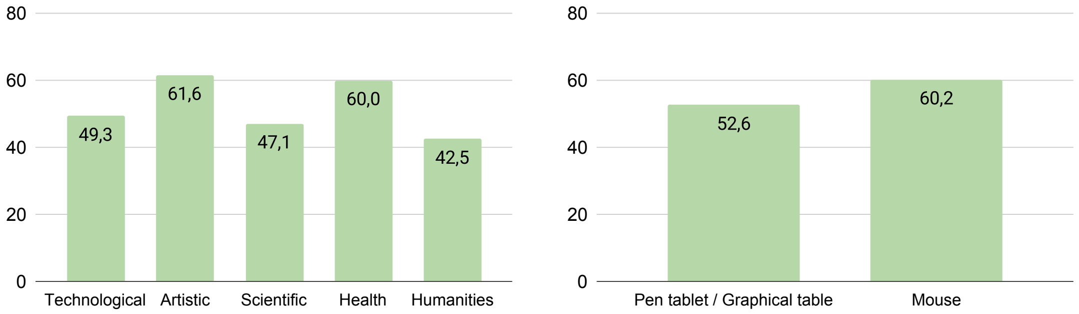
Metric	Correlation
Task 1 time	−0.32
Task 2 time	−0.60
Task 3 time	−0.15
Task 4 time	−0.11
Age	0.14
Familiarity level	0.20
Frequency input device	0.33

5.2.4 SUS Scores by Demographic Groups

In addition to the numerical correlation analysis, charts were generated to examine potential patterns in SUS scores across categorical variables. Figure 5.1 presents the distribution of SUS scores according to education level, game development experience, participant background, and input device.



(a) *SUS—Academic qualification.* **(b)** *SUS—Experience with game development.*



(c) *SUS—Background.* **(d)** *SUS—Input device.*

Figure 5.1: *Distribution of SUS scores by participant demographics.*

Exploring SUS scores by academic qualification, as shown in Figure 5.1a, indicates that technically oriented or higher-educated users rated the system's usability slightly higher. The maximum difference observed between groups was approximately 7.4 points.

Regarding experience with game development (Figure 5.1b), participants with no involvement in the field reported higher SUS scores ($mean = 78.57$), showing a substantial difference of 29.11 compared to participants who were currently studying game development, who reported lower scores ($mean = 49.46$). Users with some prior experience in game development rated the add-on between these two extremes. These results suggest that participants who are either not involved in game development or who are more experienced in the area tend to provide higher usability ratings.

The SUS scores by background, shown in Figure 5.1c, indicate that participants with artistic and health-related backgrounds reported the highest average usability, whereas those from humanities, technological, and scientific backgrounds were less satisfied. This suggests that users whose expertise aligns more closely with visual or creative work may find the add-on's interface and features better suited to their expectations and workflow.

We also observed a slight difference based on the input device, as shown in Figure 5.1d, with mouse users tending to rate the add-on more favourably ($mean = 60.23$), reflecting a modest difference of 7.6 points. This suggests that the add-on performs well with both input devices.

5.2.5 SUS Scores by Combined Analyses

For a further analysis, we grouped the previously explored three demographic features that showed greater differences on the SUS score to provide more detailed insights.

The analysis shown in Figure 5.2 presents a detailed distribution of SUS scores across different backgrounds for each input device. For both input types, participants with an artistic-related background reported higher SUS scores. Participants with non-artistic backgrounds tended to rate the add-on slightly higher when using a mouse; however, the differences between input types were generally small, with scores ranging from 42.5 to 51.9, excluding participants with a health-related background.

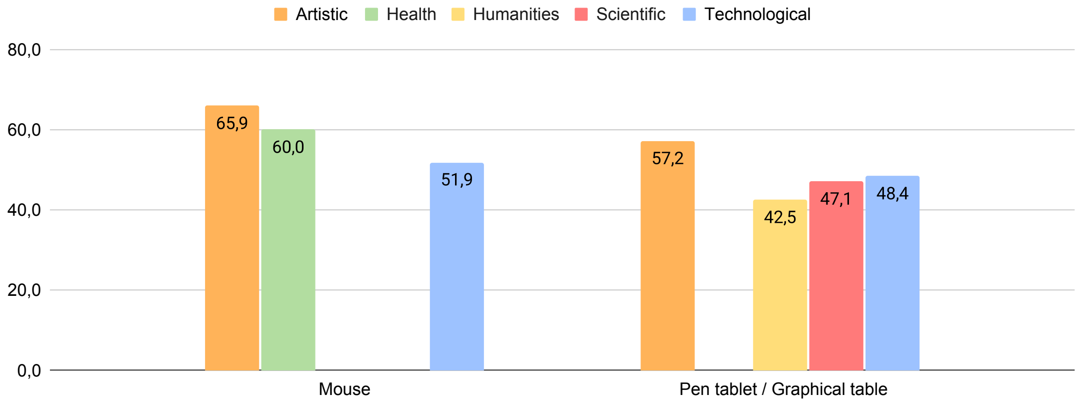


Figure 5.2: *SUS—Input device—Background relation chart.*

We then analysed the distribution of SUS scores across levels of experience in game development for each input device, as shown in Figure 5.3. The results revealed particularly high scores from participants with no involvement in game development who used a mouse, with an average score of 91.3.

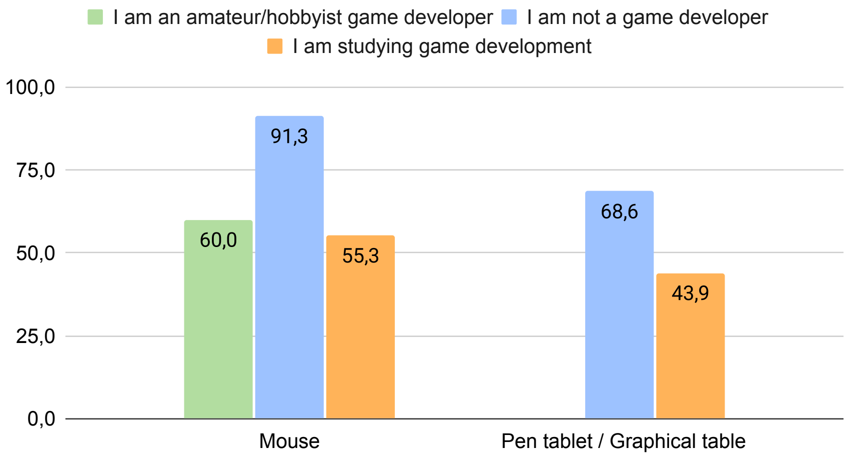


Figure 5.3: *SUS—Input device—Experience in game development relation chart.*

Lastly, we analysed the distribution of SUS scores across participant backgrounds for each level of experience in game development (Figure 5.4). The results indicate that users with artistic or scientific backgrounds who were not involved in game development reported very high SUS scores. This suggests that the earlier observation that participants with scientific backgrounds typically rate usability lower is not entirely accurate. Instead, satisfaction appears to depend on game development experience, with users who have experience in game development and a scientific background tending to be less satisfied with the add-on's usability.

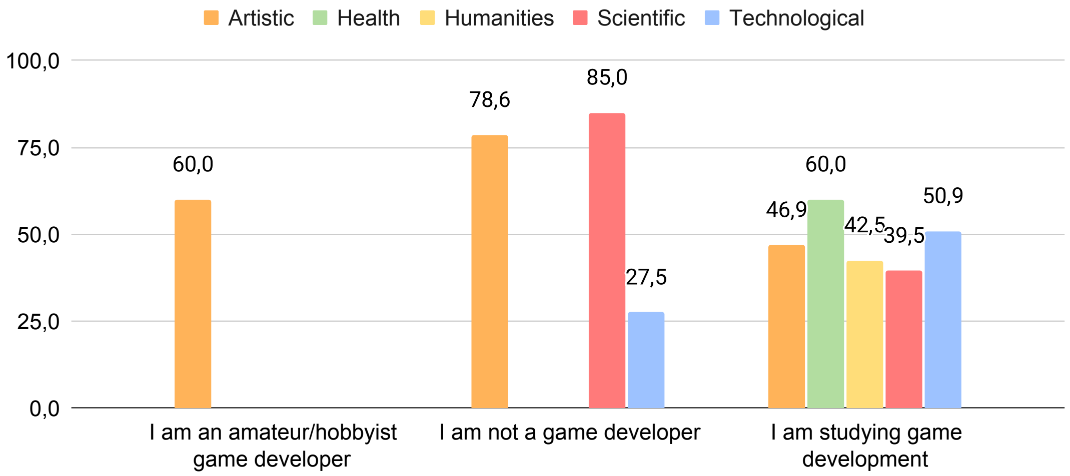


Figure 5.4: *SUS—Experience in game development—Background relation chart.*

5.2.6 Qualitative Analysis

Approximately half of the participants provided optional qualitative feedback. Several reported issues with the `Ctrl+Z` shortcut, noting that it reset or closed User Interface (UI) elements or the drawing phase rather than undoing strokes on the main canvas. Participants also indicated that the branch shape and curvature type features were not well explained or immediately intuitive. Additionally, some noted that the instructional video was insufficiently descriptive and that the panel containing the video link within the add-on was ineffective, as the link was not clickable. Despite these issues, positive feedback was received, with participants describing the concept as promising or enjoyable once learned, and some finding the add-on easy and intuitive to use.

6

Discussion

This chapter interprets the results presented in Chapter 5, evaluating the usability and effectiveness of the Blender add-on for procedural plant modelling. Usability was assessed using the System Usability Scale (SUS), while efficiency was measured via participants' task completion times. The discussion highlights trends in these metrics and explores how participant demographics, experience, and input devices influenced outcomes.

6.1 Efficiency (Task Completion Times)

Task completion times varied across the four tasks. The extreme outlier in Task 1 may be attributed to some participants not fully reading the instructions. Several participants assumed the Google Form was to be completed after downloading the provided `ZIP` file rather than simultaneously.

Task 2 exhibited shorter completion times, likely reflecting increased familiarity with the add-on and the participants' realization of how to follow the tasks correctly. Task 4 also showed a low mean time, which may indicate either greater user familiarity or a reduced willingness to experiment extensively.

Task 3 required participants to explore multiple optional features, which likely contributed to its longer completion times relative to Tasks 2 and 4.

6.2 Usability (SUS Scores)

The overall mean SUS score was below the benchmark of 68. This may be partly due to the add-on's novel interface and features, which could have caused initial unfamiliarity. Some participants reported that branch shape and curvature type features were not immediately clear, which may have further influenced usability ratings.

6.3 Influence of Demographics

To understand how participant characteristics affect usability perceptions, we analysed correlations between SUS scores, task completion times, and demographic factors.

Correlation Patterns Participants who took longer to complete tasks generally reported lower SUS scores, highlighting the importance of an intuitive interface and efficient workflows. Mild positive correlations with age, 3D modelling familiarity, and frequency of input device use suggest that more experienced users or those comfortable with their input device perceive slightly higher usability. While these correlations are not strong enough to be predictive, they indicate which user groups might benefit from targeted guidance.

SUS Scores by Demographic Groups Analysis of categorical demographics revealed several interesting patterns. Users with artistic or health-related backgrounds rated usability higher than those from humanities, technological, or scientific backgrounds, suggesting that familiarity with visual or creative work enhances user satisfaction. Technically oriented or higher-educated participants also reported slightly higher usability scores, likely reflecting greater familiarity with advanced tools. Participants currently studying game development reported lower scores compared to those with no involvement, indicating potential differences in expectations or evaluation criteria. Finally, mouse users rated the add-on marginally higher than tablet users; however, these differences were minor, suggesting that the system performs comparably across input devices.

Combined Analyses Further analysis, combining participants' backgrounds with their input device or game development experience, refined these observations. Participants with artistic backgrounds reported higher SUS scores regardless of the input device used. In contrast, scientific-background users who were not involved in game development rated the add-on highly, whereas those with game development experience expressed lower satisfaction. These findings highlight how demographic and experiential factors interact to shape usability perceptions.

6.4 Qualitative Feedback

The confusion around the undo shortcut suggests that the participants tried to use commonly known commands in the area. Although the undo function had already been remapped and a dedicated button added, we could've emphasized it more in the questionnaire.

The difficulties reported by participants regarding the branch shape and curvature type features suggest that the representative icons included were insufficient to ensure clarity. Providing additional, easily accessible information, such as tooltips, could enhance user understanding and reduce initial confusion. We therefore propose implementing such informational buttons, as illustrated in Figures 6.1 and 6.2.

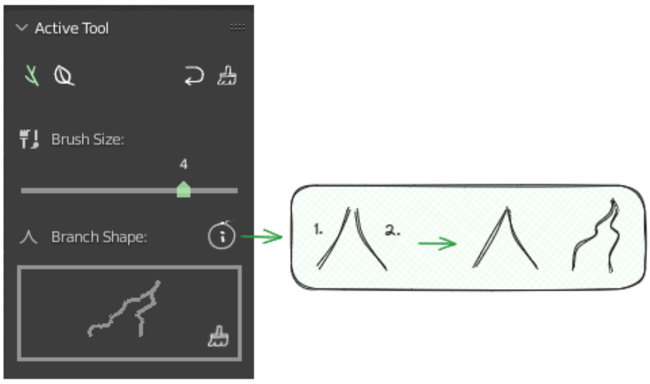


Figure 6.1: *Proposal of an information button for the branch shape feature.*

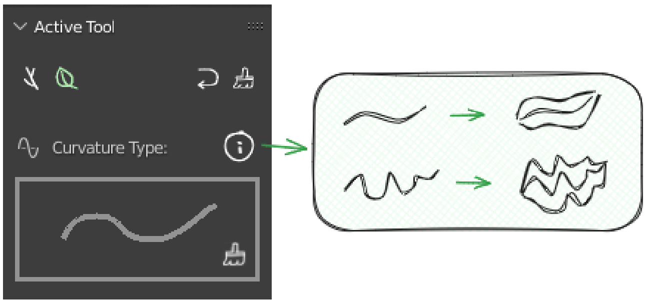


Figure 6.2: *Proposal of an information button for the curvature type feature.*

Comments noting that the video was insufficiently descriptive and that the link panel was ineffective highlight the need for more accessible, step-by-step guidance. Since Blender limits embedded video and clickable links require additional permissions, as discussed in Section 4.3.5, alternative approaches, such as including a QR code or providing a walk-through video, may better support users.

7

Conclusion and Future Work

Recent advancements in 3D plant generation have largely relied on specialized hardware and sensor data to produce high-fidelity models. At the same time, the limited availability of publicly accessible 3D plant models often results in repetitive and generic content, restricting creativity in virtual environments and posing challenges for both game developers and virtual environment creators.

This dissertation addressed these challenges by focusing on sketch-based plant modelling as an accessible and flexible approach. A systematic review following the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) guidelines provided a comprehensive overview of existing methods that rely exclusively on visual inputs, identifying two main directions: sketch-based modelling and tree generation from photographic data. Insights from this review informed the design of a novel framework that integrates elements from both single- and multi-sketch strategies.

The proposed framework was implemented as a Blender add-on, allowing users to convert hand-drawn sketches into 3D plant models. The system supports additional operations such as generating multiple variations of models, applying variable Level of Detail (LoD), and incorporating leaf propagation, while providing a streamlined interface that integrates seamlessly with Blender's existing workflow.

Evaluation of the add-on combined quantitative and qualitative analyses. Task completion times demonstrated that users could model plants efficiently, while a mean System Usability Scale (SUS) score of 56.4 indicated moderate usability. Correlation analyses and demographic comparisons revealed that factors such as experience with game development, familiarity with input devices, and user background influenced usability ratings. Qualitative feedback highlighted areas for improvement, particularly in feature intuitiveness, onboarding, and guidance. Despite limitations such as a relatively small and unevenly distributed participant sample, these findings suggest that the add-on is a promising tool for procedural plant modelling based on sketches and has potential for further enhancement.

Overall, this work contributes a practical and flexible approach to 3D plant modelling, bridging the gap between intuitive sketching and fully realized virtual models.

It provides both a systematic understanding of the field and a functional tool that enhances creative possibilities for users with varying levels of expertise.

7.1 Future Work

Several directions could extend and improve the proposed framework. Expanding the system to support additional plant elements, such as flowers or fruits, or allow to draw holes in the leaves like certain plant species (e.g. monsteras), would broaden creative possibilities. Incorporating branch propagation, multiplication, or alternative modelling strategies could further optimize the workflow for more complex plant types.

Advances in Artificial Intelligence (AI) present promising opportunities for improving sketch recognition and model generation. For instance, AI-based techniques could classify user-drawn strokes, suggest plausible completions of incomplete sketches, or provide adaptive modelling suggestions based on user behaviour, reducing manual effort while preserving creative control.

Interface improvements could also enhance usability, such as adding zooming and canvas scaling, enabling users to return to the drawing stage after model creation, and providing clearer onboarding and guidance for advanced features. Collectively, these enhancements would further increase the accessibility, flexibility, and efficiency of sketch-based plant modelling workflows.

This page intentionally left blank.

This page intentionally left blank.

Bibliography

Bailey, Brian N. and Miguel H. Ochoa (2018). “Semi-direct tree reconstruction using terrestrial LiDAR point cloud data”. In: *Remote Sensing of Environment* 208, pp. 133–144.

Cai, J. and S. Miklavcic (2012). “Automated extraction of three-dimensional cereal plant structures from two-dimensional orthographic images”. In: *IET Image Processing* 6, pp. 687–696.

Ding, Zhan et al. (2008a). “Flower solid modeling based on sketches”. In: *Journal of Zhejiang University-Science A* 9, pp. 481–488.

Ding, Zhan et al. (2008b). “Flower solid modeling based on sketches”. In: *Journal of Zhejiang University-Science A* 9, pp. 481–488.

Dror, Ran and Ilan Shimshoni (2009). *Using Phyllotaxis for Date Palm Tree 3D Reconstruction from a Single Image.* Vol. 2, p. 296.

Guénard, Jérôme et al. (2013). “Reconstructing Plants in 3D from a Single Image Using Analysis-by-Synthesis”. In: *Advances in Visual Computing*. Springer, pp. 322–332.

Hu, Anqi, Nobuyoshi Yabuki, and Tomohiro Fukuda (2024a). “GENERATING 4D PLANT MODELS FOR VIRTUAL REALITY ENVIRONMENTS USING THE INSTANT NEURAL GRAPHICS PRIMITIVES AND STABLE DIFFUSION MODEL”. In: *Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia*. Vol. 3, pp. 421–430.

Hu, Anqi, Nobuyoshi Yabuki, and Tomohiro Fukuda (2024b). *Generating 4D Plant Models for Virtual Reality Environments Using the Instant Neural Graphics Primitives and Stable Diffusion Model.*

Hu, WX et al. (2003). “An efficient image-based 3D reconstruction algorithm for plants”. In: *CAD/ Graphics Technology and Its Applications, Proceedings*, pp. 418–419.

Ijiri, Takashi et al. (2005a). “Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints”. In: *ACM Trans. Graph.* 24, pp. 720–726.

Ijiri, Takashi et al. (2005b). “Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints”. In: *ACM Trans. Graph.* 24, pp. 720–726.

Kim, Jaehwan and Il-Kwon Jeong (2014). “Single Image–Based 3D Tree and Growth Models Reconstruction”. In: *ETRI Journal* 36, pp. 450–459.

Kuwahara, Noriaki, Shinichi Shiwa, and Fumio Kishino (1995). “Three-dimensional tree shape reconstruction from multiviewpoint images based on fractal geometry”. In: *Proceedings of SPIE - The International Society for Optical Engineering* 2501, pp. 1267–1278.

Lewis, James R. and Jeff Sauro (2018). “Item benchmarks for the system usability scale”. In: *J. Usability Studies* 13.3, pp. 158–167.

Li, Dawei et al. (2015). “Digitization and Visualization of Greenhouse Tomato Plants in Indoor Environments”. In: *Sensors* 15, pp. 4019–4051.

Lin, Yi and Juha Hyyppä (2022). “Towards 3D basic theories of plant forms”. In: *Communications Biology* 5, pp. 1–6.

Liu, Jia et al. (2010). “Creation of Tree Models from Freehand Sketches by Building 3D Skeleton Point Cloud”. In: *Entertainment for Education. Digital Techniques and Systems*, pp. 621–632.

Liu, Jia et al. (2012). “Easy modeling of realistic trees from freehand sketches”. In: *Frontiers of Computer Science* 6, pp. 756–768.

Liu, Pu, Yuan Liu, and Xiaosheng Xu (2016). “Three-Dimensional Modeling Technology of Virtual Plant in Digital Landscape”. In: Atlantis Press, pp. 146–149.

Liu, Yadong et al. (2023). “Fast reconstruction method of three-dimension model based on dual RGB-D cameras for peanut plant”. In: *Plant Methods* 19, p. 17.

Liu, Zhihao et al. (2019). “Interactive Modeling of Trees Using VR Devices”. In: pp. 69–75.

Liu, Zhihao et al. (2021). “Single Image Tree Reconstruction via Adversarial Network”. In: *Graphical Models* 117.

Lopez, Luis D., Yuanyuan Ding, and Jingyi Yu (2010). “Modeling Complex Unfoliaged Trees from a Sparse Set of Images”. In: *Computer Graphics Forum* 29, pp. 2075–2082.

Lu, Chenyao, Li Deng, and Minrui Fei (2015). “An improved visualization modelling method of greenhouse tomato plants based on L-system”. In: pp. 480–485.

Mitsanis, Christos, William Hurst, and Bedir Tekinerdogan (2024). “A 3D functional plant modelling framework for agricultural digital twins”. In: *Computers and Electronics in Agriculture* 218, p. 108733.

Muhar, Andreas (2001). “Three-dimensional modelling and visualisation of vegetation for landscape simulation”. In: *Landscape and Urban Planning* 54, pp. 5–17.

Neubert, Boris, Thomas Franken, and Oliver Deussen (2007). “Approximate image-based tree-modeling using particle flows”. In: *ACM Trans. Graph.* 26, 88–es.

Nguyen, Thuy Tuong et al. (2015). “Structured Light-Based 3D Reconstruction System for Plants”. In: *Sensors* 15, pp. 18587–18612.

Okabe, Makoto, Shigeru Owada, and Takeo Igarashi (2007a). “Interactive design of botanical trees using freehand sketches and example-based editing”. In: *ACM SIGGRAPH 2007 Courses*, 26–es.

Okabe, Makoto, Shigeru Owada, and Takeo Igarashi (2007b). “Interactive design of botanical trees using freehand sketches and example-based editing”. In: *ACM SIGGRAPH 2007 Courses*, 26–es.

Okura, Fumio (2022). “3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping”. In: *Breeding Science* 72, pp. 31–47.

Page, Matthew J et al. (2021). “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews”. In: *BMJ* 372.

Paturkar, Abhipray, Gourab Sen Gupta, and Donald Bailey (2022). “Plant trait measurement in 3D for growth monitoring”. In: *Plant Methods* 18, p. 59.

Prusinkiewicz, Przemyslaw and Aristid Lindenmayer (1990). *The Algorithmic Beauty of Plants*. Springer.

Scharr, Hanno et al. (2017). “Fast High Resolution Volume Carving for 3D Plant Shoot Reconstruction”. In: *Frontiers in Plant Science* 8.

Tan, Ping et al. (2008). “Single image tree modeling”. In: *ACM Trans. Graph.* 27, pp. 1–7.

Teng, Chin-Hung and Yung-Sheng Chen (2009). “Image-based tree modeling from a few images with very narrow viewing range”. In: *Visual Computer* 25, pp. 297–307.

Wang, Bohuai and Xizhong Shen (2024). “3D Reconstruction of Plant Based on NeRF”. In: *2024 IEEE 3rd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pp. 345–348.

Weber, Jason and Joseph Penn (1995). “Creation and rendering of realistic trees”. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery, pp. 119–128.

Xie, Donghui et al. (May 2018). “Reconstruction of Single Tree with Leaves Based on Terrestrial LiDAR Point Cloud Data”. In: *Remote Sensing* 10, p. 686.

Xu, Duo et al. (2024). “Three-Dimensional Reconstruction of Forest Scenes with Tree–Shrub–Grass Structure Using Airborne LiDAR Point Cloud”. In: *Forests* 15, p. 1627.

Yan, Feilong et al. (2014). “Flower reconstruction from a single photo”. In: *Computer Graphics Forum* 33, pp. 439–447.

Yang, Meng et al. (2009). “Multi-level tree branch modeling and animation”. In: *2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pp. 96–99.

Yuan, Qi and Yongjian Huai (2021). “Immersive sketch-based tree modeling in virtual reality”. In: *Computers & Graphics* 94, pp. 132–143.

Zamuda, Ales and Janez Brest (2014). “Vectorized procedural models for animated trees reconstruction using differential evolution”. In: *Information Sciences* 278, pp. 1–21.

Zamuda, Ales et al. (2011). “Differential evolution for parameterized procedural woody plant models reconstruction”. In: *Applied Soft Computing* 11, pp. 4904–4912.

ZHANG, Yu et al. (2018). “3D monitoring for plant growth parameters in field with a single camerby multi-view approach”. In: *Journal of Agricultural Meteorology* 74, pp. 129–139.

Zheng, Ying et al. (2011). “Detailed Reconstruction of 3D Plant Root Shape”. In: *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 2026–2033.

This page intentionally left blank.

Appendices

A

Search Key Progress

This appendix presents the steps of the process to obtain the final search key, which is fully described in the Section 2.1.

Table A.1: *Search key progress.*

Number	Name	Inputs	Methods	Outputs	Exclusions
1	Number Topics	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / procedural reconstruction / generative / generation	3D mesh / 3D tree / 3D plant	-
2	Modeling & Modelling	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / **procedural modeling** / **procedural modelling** / procedural reconstruction / generative / generation	3D mesh / 3D tree / 3D plant	-
3	Removed 3D mesh	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation	3D tree / 3D plant	-
4	3D flower	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation	3D tree / 3D plant / **3D flower**	-
5	Mesh terms	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation	3D tree / 3D plant / 3D flower / **tree mesh** / **plant mesh** / **flower mesh**	-
6	Model terms	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation	3D tree / 3D plant / 3D flower / tree mesh / plant mesh / flower mesh / **tree model** / **plant model** / **flower model**	-

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
7	Asset terms	illustration / images / 2D scan / photograph / sketch / drawing / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation	3D tree / 3D plant / 3D flower / tree mesh / plant mesh / flower mesh / tree model / plant model / flower model / **tree asset** / **plant asset** / **flower asset**	-
8	Input variants	illustration / **illustrations** / **illustrating** / **image** / images / 2D scan / **2D scans** / **2D scanning** / photograph / **photographs** / sketch / **sketches** / **sketching** / drawing / **drawings** / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation	3D tree / 3D plant / 3D flower / tree mesh / plant mesh / flower mesh / tree model / plant model / flower model / tree asset / plant asset / flower asset	-
9	Generate variants	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / generative / generation / **generate** / **generates** / **generated** / **generating**	3D tree / 3D plant / 3D flower / tree mesh / plant mesh / flower mesh / tree model / plant model / flower model / tree asset / plant asset / flower asset	-
10	Reconstruction	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / **reconstruction** / generative / generation / generate / generates / generated / generating	3D tree / 3D plant / 3D flower / tree mesh / plant mesh / flower mesh / tree model / plant model / flower model / tree asset / plant asset / flower asset	-
11	Output Variants	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / **3D trees** / 3D plant / **3D plants** / 3D flower / **3D flowers** / tree mesh / **trees mesh** / **tree meshes** / **trees meshes** / plant mesh / **plants mesh** / **plant meshes** / **plants meshes** / flower mesh / **flowers mesh** / **flower meshes** / **flowers meshes** / tree model / **trees model** / **tree models** / **trees models** / plant model / **plants model** / **plant models** / **plants models** / flower model / **flowers model** / **flower models** / **flowers models** / tree asset / **trees asset** / **tree assets** / **trees assets** / plant asset / **plants asset** / **plant assets** / **plants assets** / flower asset / **flowers asset** / **flower assets** / **flowers assets**	-

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
12	Exclusions plant related	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	**nuclear** / **radioactive** / **radioative** / **power** / **powers** / **powered**
13	Exclusions tree model related	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / **Markov** / **FID** / **FID3** / **decision** / **Bayesian** / **algebraic**

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
14	Exclusions tree related	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / **disease** / **diseases** / **artery** / **arteries** / **vascular** / **angiography** / **cancer** / **hematopoietic** / **human pose**
15	Exclusions sensor related	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular / angiography / cancer / hematopoietic / human pose / **laser** / **lasers** / **scan** / **scans** / **sensor** / **sensors** / **LiDAR** / **UAV** / **stereo** / **stereoscopy** / **intrusive** / **depth** / **x-ray** / **radar** / **radars**

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
16	Exclusions method related	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular / angiography / cancer / hematopoietic / human pose / laser / lasers / scan / scans / sensor / sensors / LiDAR / UAV / stereo / stereoscopy / intrusive / depth / x-ray / radar / radars / **sensing** / **classification** / **recognition** / **validation** / **predict** / **prediction** / **predicting** / **dataset** / **datasets** / **L-system** / **L-systems**
17	Exclusions flower related	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular / angiography / cancer / hematopoietic / human pose / laser / lasers / scan / scans / sensor / sensors / LiDAR / UAV / stereo / stereoscopy / intrusive / depth / x-ray / radar / radars / sensing / classification / recognition / validation / predict / prediction / predicting / dataset / datasets / L-system / L-systems / **nanoflower** / **nanoflowers** / **battery** / **batteries** / **semiconductor** / **semiconductors**

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
18	Springer excluded words	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular / angiography / cancer / hematopoietic / human pose / laser / lasers / scan / scans / sensor / sensors / LiDAR / UAV / stereo / stereoscopy / intrusive / depth / x-ray / radar / radars / sensing / classification / recognition / validation / predict / prediction / predicting / dataset / datasets / L-system / L-systems / nanoflower / nanoflowers / battery / batteries / semiconductor / semiconductors / **CAD** / **fuzzy** / **stress** / **database** / **logic** / **time-varying** / **pattern** / **cell growth** / **residual** / **financial** / **economy** / **butterfly** / **SPQR-tree** / **driving** / **phylogenetic** / **supervisory** / **recursive** / **voltage** / **task-tree** / **animal** / **moduli** / **management** / **motor** / **chaos** / **PID** / **elastic** / **fowler** / **dendritic**

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
19	Scopus excluded words	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular / angiography / cancer / hematopoietic / human pose / laser / lasers / scan / scans / sensor / sensors / LiDAR / UAV / stereo / stereoscopy / intrusive / depth / x-ray / radar / radars / sensing / classification / recognition / validation / predict / prediction / predicting / dataset / datasets / L-system / L-systems / nanoflower / nanoflowers / battery / batteries / semiconductor / semiconductors / CAD / fuzzy / stress / database / logic / time-varying / pattern / cell growth / residual / financial / economy / butterfly / SPQR-tree / driving / phylogenetic / supervisory / recursive / voltage / task-tree / animal / moduli / management / motor / chaos / PID / elastic / fowler / dendritic / dendritic / **ecotoxicity** / **IP** / **galaxy**

Continued on the next page.

Table A.1 continued from previous page.

Number	Name	Inputs	Methods	Outputs	Exclusions
20	IEEE Xplorer excluded words	illustration / illustrations / illustrating / image / images / 2D scan / 2D scans / 2D scanning / photograph / photographs / sketch / sketches / sketching / drawing / drawings / concept art	procedural model / procedural modeling / procedural modelling / procedural reconstruction / reconstruction / generative / generation / generate / generates / generated / generating	3D tree / 3D trees / 3D plant / 3D plants / 3D flower / 3D flowers / tree mesh / trees mesh / tree meshes / trees meshes / plant mesh / plants mesh / plant meshes / plants meshes / flower mesh / flowers mesh / flower meshes / flowers meshes / tree model / trees model / tree models / trees models / plant model / plants model / plant models / plants models / flower model / flowers model / flower models / flowers models / tree asset / trees asset / tree assets / trees assets / plant asset / plants asset / plant assets / plants assets / flower asset / flowers asset / flower assets / flowers assets	nuclear / radioactive / radioative / power / powers / powered / Markov / FID / FID3 / decision / Bayesian / algebraic / disease / diseases / artery / arteries / vascular / angiography / cancer / hematopoietic / human pose / laser / lasers / scan / scans / sensor / sensors / LiDAR / UAV / stereo / stereoscopy / intrusive / depth / x-ray / radar / radars / sensing / classification / recognition / validation / predict / prediction / predicting / dataset / datasets / L-system / L-systems / nanoflower / nanoflowers / battery / batteries / semiconductor / semiconductors / CAD / fuzzy / stress / database / logic / time-varying / pattern / cell growth / residual / financial / economy / butterfly / SPQR-tree / driving / phylogenetic / supervisory / recursive / voltage / task-tree / animal / moduli / management / motor / chaos / PID / elastic / fowler / dendritic / dendritic / ecotoxicity / IP / galaxy / **vision tracking** / **moving person** / **PI** / **accelerator** / **accelerators**

B

Relevant characteristics of systematic review articles

This appendix presents the defined most relevant characteristics of the systematic review articles, mentioned in Section 2.2.4.

Table B.1: *Relevant characteristics of systematic review articles.*

ID	Reference	3D Generation Type	Input Type	Plant Type	3D Plant Creation Method	Processing Time	Evaluation Type
1	Wang et al., 2024	Reconstruction	Multiple Photos	Any Plant	Instant-NGP	$21 - 23\ min$	Accuracy
2	Zamuda et al., 2011	Reconstruction	Multiple Photos	Trees	Procedural Modeling	–	Accuracy
3	A. Hu et al., 2024a	Reconstruction	Multiple Photos	Any Plant	Instant-NGP	–	Accuracy
4	Lopez et al., 2010	Reconstruction	Multiple Photos	Trees	3D Coordinate System	$\sim 7.5\ min$	Accuracy
5	Zamuda et al., 2014	Reconstruction	2 Photos	Trees	Procedural Modeling	–	Accuracy
6	Neubert et al., 2007	Reconstruction	Multiple Photos	Trees	3D Skeleton	$\sim 20s$	Realism
7	Zheng et al., 2011	Reconstruction	Multiple Photos	Roots	Visual Hull	–	Accuracy
8	Scharr et al., 2017	Reconstruction	Multiple Photos	Any Plant	Octree	$\sim 37.7s / \sim 85.7s$	Accuracy
9	W. Hu et al., 2003	Reconstruction	Multiple Pairs of Photos	Any Plant	3D Coordinate System	–	Accuracy
10	Yang et al., 2009	Reconstruction	Multiple Photos	Trees	3D Skeleton	–	Realism
11	Teng et al., 2009	Reconstruction	Multiple Photos	Trees	3D Skeleton	$20 - 96\ min$	Realism
12	Kuwahara et al., 1995	Reconstruction	Multiple Photos	Trees	Fractal Geometry	–	Accuracy
13	Yan et al., 2014	Reconstruction	Single Photo	Flowers	3D Coordinate System	$< 1\ min$	Effectiveness
14	Kim et al., 2014	Reconstruction	Single Photo	Trees	Gaussian Distribution	–	Realism
15	Z. Liu et al., 2021	Reconstruction	Single Photo	Trees	Procedural Modeling	$\sim 14.34s$	Accuracy
16	Ding et al., 2008b	Modelling	Multiple Sketches	Flowers	Merging Algorithm	–	Usability
17	Ijiri et al., 2005b	Modelling	Multiple Sketches	Flowers	Merging Algorithm	$< 40\ min$	Usability

Continued on the next page.

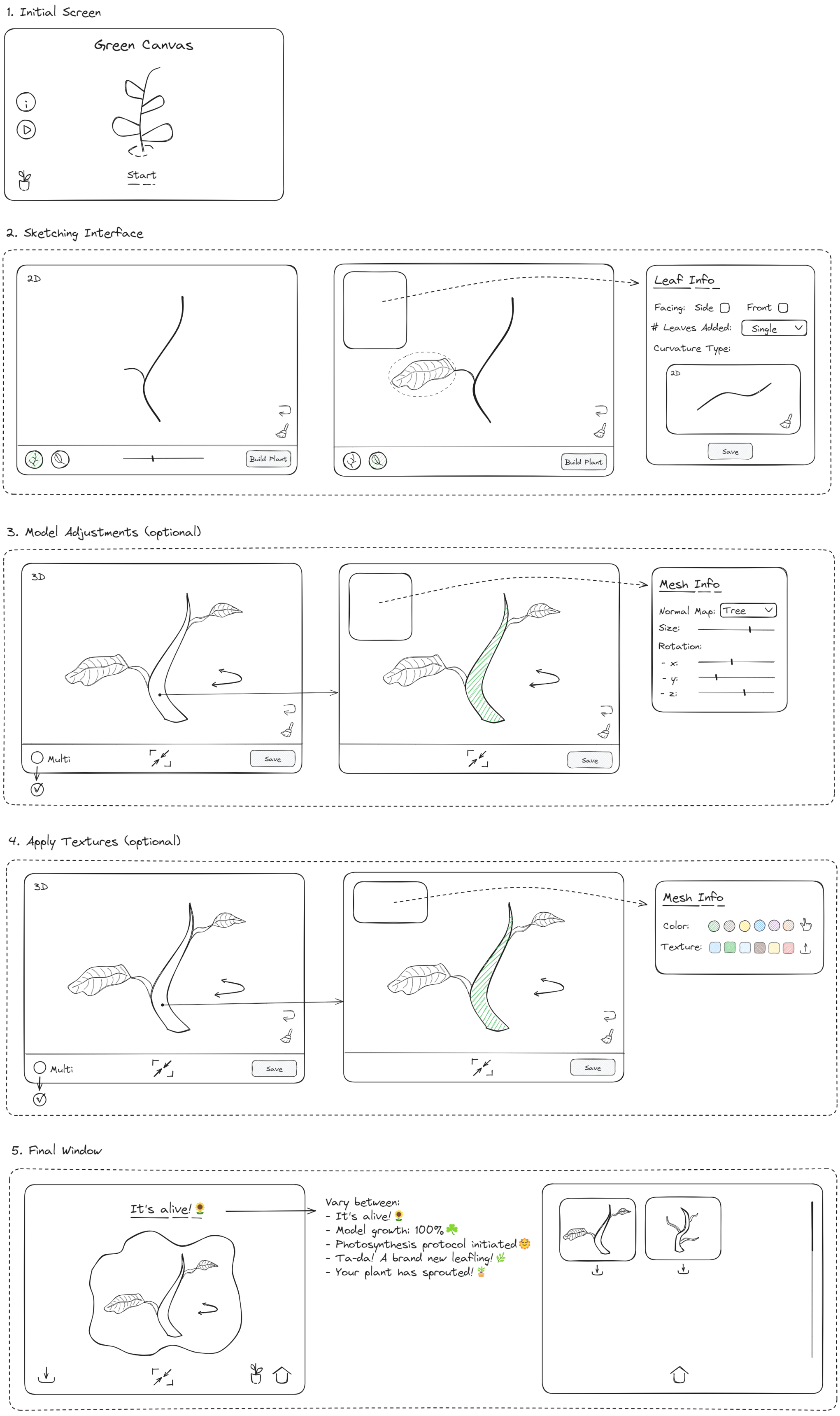
Table B.1 continued from previous page.

ID	Reference	3D Generation Type	Input Type	Plant Type	3D Plant Creation Method	Processing Time	Evaluation Type
18	Okabe et al., 2007b	Modelling	Single Sketch	Trees	Depth Modulation	$510\ min$	Usability
19	Tan et al., 2008	Reconstruction	1 Photo & Multiple Sketches	Trees	3D Skeleton	$\sim 30min$	Realism
20	J. Liu et al., 2010	Modelling	2 Photos & 2 Sketches	Trees	3D Skeleton	$< 30s$	Realism
21	J. Liu et al., 2012	Modelling	2 Photos & 2 Sketches	Trees	3D Skeleton	$< 1min$	Realism

C

Stand Alone Application Design

This appendix presents the full workflow design of the stand-alone application, as mentioned in Section 3.1.2.



Vary between:
- It's alive! 🌻
- Model growth: 100% ☘️
- Photosynthesis protocol initiated 😊
- Ta-da! A brand new leafling! 🌿
- Your plant has sprouted! 🪴

Figure C.1: *Stand-alone application design.*