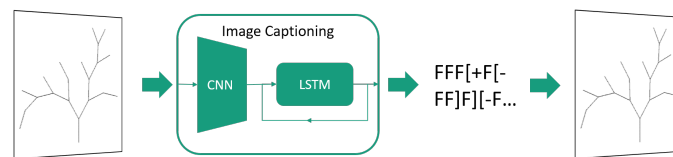


# Towards L-System Captioning for Tree Reconstruction

J. S. Magnusson<sup>1</sup>†, A. Hilsmann<sup>1</sup> and P. Eisert<sup>1,2</sup>

<sup>1</sup>Fraunhofer Heinrich Hertz Institute HHI, Germany

<sup>2</sup>Visual Computing, Institut für Informatik, Humboldt-University, Germany



**Figure 1:** Our goal is to train a model that can infer L-System words as a representation of trees from images.

## Abstract

This work proposes a novel concept for tree and plant reconstruction by directly inferring a Lindenmayer-System (L-System) word representation from image data in an image captioning approach. We train a model end-to-end which is able to translate given images into L-System words as a description of the displayed tree. To prove this concept, we demonstrate the applicability on 2D tree topologies. Transferred to real image data, this novel idea could lead to more efficient, accurate and semantically meaningful tree and plant reconstruction without using error-prone point cloud extraction, and other processes usually utilized in tree reconstruction. Furthermore, this approach bypasses the need for a predefined L-System grammar and enables species-specific L-System inference without biological knowledge.

## CCS Concepts

• *Computing methodologies* → *Shape representations; Reconstruction; Shape analysis; Neural networks;*

## 1. Introduction

Tree modeling and reconstruction are challenging problems in Computer Graphics and Vision and have gained increased interest with the digitalization in agriculture. Current methods for tree and plant reconstruction mostly start with point clouds measured by 3D scanners or extracted from multi-view images. They derive surface meshes or skeletons as a final representation [Oku22]. Reconstruction based on photogrammetric approaches is challenging due to occlusions, complex structures, or measuring errors. In modeling, fractal structures like trees and plants are generated by procedural modeling based on grammar- or graph-based representations [CRW\*20]. For example, Lindenmayer-Systems (L-Systems) were developed to model plant topologies. L-Systems are formal grammars which produce a sequence of commands. These commands describe geometric shapes based on turtle geometry [PL12].

Further extensions allow for 3D modeling, growth simulations, or even nutrient propagation inside plants which makes L-Systems a powerful tool for biological computations [PL12]. For these computations, L-System words are more suited to represent individual plants and trees than point clouds or meshes, which are usually derived in reconstruction. Therefore, we bridge the gap between the usual reconstruction and modeling processes and directly infer an L-System representation of a tree from image data.

The covered problem shows parallels to that in image captioning, where an image is translated into natural language describing the image content. It also relates to inverse procedural modeling which models a procedural representation of existing geometries. Inspired by advances in these areas, we propose to treat the problem as an image captioning problem to translate an image of a tree into a grammar-based representation used in inverse procedural modeling: L-Systems words.

In this paper, we present a proof of concept on simple 2D images as an important step towards solving the problem of inverse procedural modeling of trees and plants from images. The contributions of this work are the following:

† This research was grant-aided by the German Federal Ministry of Economic Affairs and Climate Action as part of the NaLamKI project under Grant 01MK21003D and the state Berlin (ProFIT) under grant number 10174498 (BerDiBa)

1. We present a new concept that combines advances in natural language processing (NLP) and image captioning with procedural modeling approaches for plants to directly infer an L-System representation from image data.
2. We prove the presented concept on simple 2D tree topology images which are automatically translated into L-System words. This approach is planned to be adapted to more complex data, i.e. RGB images.
3. We define requirements for an L-System word to ensure a bidirectional relation between image and derivation.

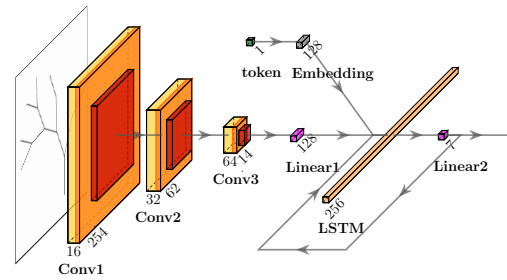
This paper is organized as follows. Sec. 2 presents related work before Sec. 3 describes our proposed approach. Sec. 4 presents the dataset and experimental setup. Then Sec. 5 evaluates our method for L-System captioning from images using standard NLP metrics.

## 2. Related Work

While reconstruction refers to the process of extracting a precise 3D representation of a specific tree, modeling refers to approaches that are able to synthesize a representation of a virtual tree [Oku22]. Most approaches that use image data for tree reconstruction first extract 3D point clouds using photogrammetric procedures [GXY\*20, GZP\*21] or start with measured point clouds by e.g. Terrestrial Laser Scanners [LGB\*21, ZLT\*14] from which specific representations of shapes and structures are extracted [Oku22]. The main challenges of using point clouds is their noisiness and overlapping structures where only the surface of the object was captured. As the final representation, most works on image-based tree and plant reconstruction use surface meshes [ZLT\*14, GXY\*20].

Regarding tree modeling, procedural modeling approaches are usually exploited when modeling trees [CRW\*20]. Guo et al. [GXY\*20] estimate parameters like angles, diameter transmission coefficients, and the number of segment lengths from a point cloud. These parameters guide a procedural model to grow a tree filling the input point cloud. Similarly, Stava et al. [SPK\*14] estimate 24 parameters via a Monte Carlo Markov Chain and a similarity measure incorporating shape, geometry, and structure similarities. As an input they use a tree graph representation. Guo et al. [GJB\*20] detect atomic structures of tree topologies with an RCNN to infer a compact L-System grammar. This method is also capable of reconstructing user drawn sketches of 2D topologies. However, an adaptation to more complex data and 3D structure is difficult as it depends on detectable structures in an image. In contrast, we are interested in developing a modular method adaptable to real image data, that learns to reconstruct topologies even if they are invisible.

Our goal is to infer an L-Systems representation directly from an image. The task is similar to image captioning problems where images are translated into text using deep learning approaches. Typically, information in images is extracted and decoded into natural language. Mostly, a Convolutional Neural Network (CNN) is used as an encoder and a recurrent network or Transformer decodes the information extracted by the encoder [SCB\*21]. The image can be encoded globally [WSW\*18, DHG\*15] or separately over a grid [LXPS17] or visual regions. The encoding is then aggregated by an attention mechanism. The decoding process was



**Figure 2:** Model to reconstruct L-System words from images. Each convolution is followed by ReLUs. Red displays max pooling.

mostly done by Long Short-Term Memory Networks (LSTMs) [WSW\*18, DHG\*15, LXPS17, AHB\*18]. Lately, more methods focus on Transformers due to their advances in NLP [LYL\*20].

For this work, we focus on the most simplistic approach to prove the applicability of image captioning to the problem of reconstructing trees. Additionally, instead of inferring a procedural model and its parameters for a given input, an L-System word is directly derived with image captioning. The resulting word is the final representation instead of surface meshes or point clouds usually used in reconstruction pipelines. From multiple derivations, a generalized L-System can be easily inferred. This enables biological computations such as growth simulations or nutrient propagation which can be realized by L-Systems.

## 3. Inferring L-System Words

The proposed method receives an image of a tree topology. This image is transcribed into an L-System word via an established image captioning pipeline with a CNN+LSTM backbone (Fig. 1). In this work, we use an LSTM-based architecture instead of e.g. more state-of-the-art Transformers to demonstrate that simple methods are capable of solving this problem. As a proof of concept, we focus on 2D tree topologies generated by a probabilistic L-System. Before describing the method in detail, we formally define L-Systems and heuristic rules to ensure a bidirectional mapping between words and images.

### 3.1. Lindenmayer Systems

Let  $V$  be an alphabet with its set of words  $V^*$  and nonempty words  $V^+$ . We define L-Systems as a grammar  $G = (V, \omega, P, \pi, \delta, f)$  with the axiom  $\omega \in V^+$ , a set of productions  $P \subset V \times V^*$ , an branching angle  $\delta \in \mathbb{R}$  and a F-length  $f \in \mathbb{R}^+$ . A production is interpreted as the substitution of a symbol  $a \in V^+$  with a word  $b \in V^*$  [PL12]. For probabilistic L-Systems,  $\pi : P \rightarrow (0, 1]$  defines a mapping of productions to an occurrence probability.  $\delta$  and  $f$  are constants that are required to display an L-System word.

For the 2D case, the alphabet is defined as  $V = F, +, -, [, ]$ . Each symbol can be interpreted as a command for a turtle which navigates on a plane and draws its path (see 1).

In order to build a grammar which can describe structures in an image, we ensure a bidirectional relationship between image and

Command	Description
$F$	move forward and draw a line
$+$	rotate right
$-$	rotate left
$[$	save current state
$]$	restore last saved state

**Table 1:** L-System commands to draw turtle geometry.

L-System word. We identified five ambiguities regarding the relation of image and L-System word. From these we derived five rules which must be fulfilled by all L-System words to ensure a bi-direction between both modalities. By these, we process and filter our generated data to stabilize the training procedure.

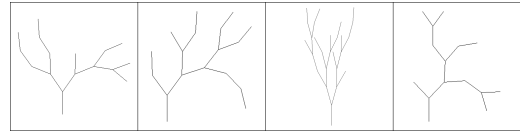
1. An L-System word might contain segments that not only cross each other but be directly on top of each other: We exclude these cases by calculating the positions of all segments and filter out doubled segments.
2. Rotations can cancel each other out:  $+/-$  cannot follow  $-/+$
3. Branches can be empty and therefore invisible: forbid  $[ ]$
4. Branches starting at the same point can be ordered differently: we sort branches from right to left, e.g.  $F[-F][+F]F$  is rewritten as  $F[+F][-F]F$
5. The last segment of a branch can be a branch or the extension of a previous branch: a branch cannot end with a subbranch, e.g.  $F[+F]$  must be rewritten as  $F+F$

### 3.2. Learning L-System Image Captions

Our model follows a standard image captioning pipeline with a CNN that encodes global features from the input image. The flattened encoding is resized by a linear layer and concatenated to each embedded token. This vector is inserted into a 1-layer LSTM to predict the next token until a maximal sequence length is reached or the  $\langle \text{eos} \rangle$  token is predicted (Fig. 2). Tokenization describes the process of splitting a string into its entities defined by the vocabulary e.g. using its characters or words. Regarding the tokenization for L-Systems, we utilize the vocabulary  $\{\langle \text{bos} \rangle, \langle \text{eos} \rangle\} + \{F, +F, -F, [, ]\}$  (compare with Tab. 1). Because each rotation is always followed by an  $F$ , we fuse these to the tokens  $+F$  and  $-F$  to prevent violations of rule 2 from Sec. 3.1. In the evaluation, we show that this strategy is advantageous over character based tokenization. Moreover, rotations can be parameterized so that multiple rotations are unnecessary to enable different rotations per branch.

### 4. Experimental Setup

For the training data, we generated 48267 unique samples from a simple probabilistic L-System with a random number of derivations  $n \in [1, 7]$  which follow the rules from Sec. 3.1. For data augmentation, the angle  $\delta$  is randomly set to  $\delta \in [15^\circ, 60^\circ]$  each epoch while the  $F$ -length  $f = 100$  stays constant. The angle interval is chosen arbitrarily based on example values in [PL12]. The tree is rendered so that the tree spans the whole image. No further augmentation is applied since the data is not noisy. We apply a 0.9, 0.05, 0.05 split for training, validation, and testing. Examples of the input images are given in Fig. 3.



**Figure 3:** Example images with a resolution of  $512 \times 512$  px.

The proposed model is trained for 495 epochs with a learning rate of 0.00025 and an ADAM optimizer. Without teacher forcing, the model optimizes the cross-entropy loss of all predicted tokens. Therefore, the loss is equally sensitive to all tokens and enforces the exact L-System word given by the ground truth.

### 5. Evaluation

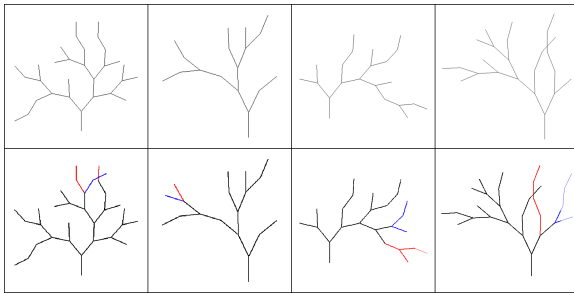
For numerical evaluation, we consider standard NLP metrics like bits-per-character (BPC) and perplexity (PPL) and analyze the correctness in different categories such as correct, false syntax, non-terminated, and residue. Correct and non-terminating sequences are self-explanatory. Any syntax error or violation of the rules from Sec. 3.1 is treated as false syntax although some failures are non-syntactical errors. The residue category contains all words that differ in one or more tokens from the ground truth while being syntactically correct. We achieve a PPL of 1.129 and a BPC of 0.403 which are both proportional to the cross-entropy of 0.1214 for the test set. A direct comparison to NLP methods is difficult due to the usage of a distinct vocabulary and language. The categorical evaluation in Tab. 2 states that roughly 80% of all words are reconstructed correctly.

Fig. 4, shows examples of the rendered outputs of the category residue. The most dominant errors in the residue class are wrong rotations or longer branches than given in the image. Due to the rules in Sec. 3.1, the zeroth hierarchy is the rightmost path from trunk to leaf. The next level describes all branches originating from the previous one. The proposed method is more confident in the first levels than the character based approach (for the second level 88.54% and 85.8%). In training, each token and therefore hierarchy level is treated equally. Accordingly, future methods will focus on assigning more importance to lower levels which have the highest visual impact.

To further validate our method, we test the effect of using a character tokenization. We argued that each rotation symbol is followed

Set	Correct	False Syntax	Non-Terminating	Residue
Train	86.23%	5.33%	0.38%	8.06%
Validation	80.32%	7.38%	0.54%	11.77%
Test	80.16%	7.75%	0.62%	11.48%
Test Char	77.13%	8.74%	0.25%	13.88%

**Table 2:** Categorical evaluation of the model’s reconstructions. A false syntax denotes any violation of the rules from Sec. 3.1 or syntax errors. The Char numbers report the performance for testing a model using a character tokenization.



**Figure 4:** Examples of rendered outputs for the category residue. The top row displays the ground truth. The bottom row shows the correct reconstructed part in black. Red and blue lines denote the part only contained in the ground truth or prediction, respectively.

Tokenization	+ - or - +	[]	empty branches
Char	0.17 %	1.32 %	0.05 %
Fused	-	1.51 %	0.19 %

**Table 3:** Errors in the test set for character (Char) and rotation-fused (Fused) tokenization. The first error describes a violation of rule 2 from Sec. 3.1. [] refers to invalid bracket pairs.

by an F and could be treated as a single token. Contrarily, one could use a character tokenization to have a completely disjoint vocabulary. Tab. 3 shows that with character tokenization violations of rule 2 are introduced. Additionally, Tab. 2 shows that the overall performance of correct L-System words is reduced by 3%. To calculate the relative errors of invalid rotation groups, all invalid groups of rotation symbols are counted and divided by the total number of rotation groups. The error of invalid brackets reflects the amount of uneven opened and closed brackets to all indicated bracket pairs. Lastly, any empty bracket pair or empty branch is related to all indicated bracket pairs.

## 6. Conclusion

This paper presents a novel approach to reconstruct L-System word representations for trees from images. It combines ideas from procedural modeling, image captioning, and NLP. Using a simple CNN+LSTM architecture, we translate images of tree topologies into corresponding L-System words. Using this representation, we bypass the necessity of a predefined L-System grammar, used in procedural modeling, or the explicit object detection used by Guo et al. [GJB\*20]. Furthermore, our approach allows to infer a species-specific L-System from a collection of words without biological and phenological knowledge using grammar inference techniques. These grammars enable biological computations such as growth modeling.

We proof this concept with simplistic data and achieve promising results that underline the applicability of image captioning to reconstruct trees using the representation of L-System words. Future work will concentrate on enabling different rotation angles and segment lengths inside a tree. Furthermore, we will incorporate multiview constraints and synthetic 3D data in order to advance the current approach towards more challenging and real data. Once

this is achieved, we will overcome challenges of point cloud reconstruction for trees.

## References

- [AHB\*18] ANDERSON P., HE X., BUEHLER C., TENEY D., JOHNSON M., GOULD S., ZHANG L.: Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proc. IEEE Conf. on Comp. Vision and Pattern Rec. (CVPR)* (Salt Lake City, UT, June 2018), pp. 6077–6086. 2
- [CRW\*20] CHAUDHURI S., RITCHIE D., WU J., XU K., ZHANG H.: Learning Generative Models of 3D Structures. *Computer Graphics Forum* 39, 2 (May 2020), 643–666. 1, 2
- [DHG\*15] DONAHUE J., HENDRICKS L. A., GUADARRAMA S., ROHRBACH M., VENUGOPALAN S., SAENKO K., DARRELL T.: Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proc. IEEE Conf. on Comp. Vision and Pattern Rec. (CVPR)* (June 2015), pp. 2625–2634. 2
- [GJB\*20] GUO J., JIANG H., BENES B., DEUSSEN O., ZHANG X., LISCHINSKI D., HUANG H.: Inverse Procedural Modeling of Branching Structures by Inferring L-Systems. *ACM Trans. on Graphics* 39, 5 (Sept. 2020). 2, 4
- [GXY\*20] GUO J., XU S., YAN D.-M., CHENG Z., JAEGER M., ZHANG X.: Realistic Procedural Plant Modeling from Multiple View Images. *IEEE Trans. on Vis. and Comp. Graph.* 26, 2 (Feb. 2020), 1372–1384. 2
- [GZP\*21] GAO T., ZHU F., PAUL P., SANDHU J., DOKU H. A., SUN J., PAN Y., STASWICK P., WALIA H., YU H.: Novel 3D Imaging Systems for High-Throughput Phenotyping of Plants. *Remote Sensing* 13 (May 2021). 2
- [LGB\*21] LIU Y., GUO J., BENES B., DEUSSEN O., ZHANG X., HUANG H.: TreePartNet: neural decomposition of point clouds for 3D tree reconstruction. *ACM Trans. on Graphics* 40, 6 (Dec. 2021). 2
- [LXPS17] LU J., XIONG C., PARIKH D., SOCHER R.: Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. In *Proc. IEEE Conf. on Comp. Vision and Pattern Rec. (CVPR)* (Honolulu, HI, July 2017), pp. 3242–3250. 2
- [LYL\*20] LI X., YIN X., LI C., ZHANG P., HU X., ZHANG L., WANG L., HU H., DONG L., WEI F., CHOI Y., GAO J.: Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. *CoRR* (Apr. 2020). 2
- [Oku22] OKURA F.: 3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping. *Breeding Science* 72, 1 (2022), 31–47. 1, 2
- [PL12] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. The Virtual Laboratory. Springer Science & Business Media, 2012. 1, 2, 3
- [SCB\*21] STEFANINI M., CORNIA M., BARALDI L., CASCIANELLI S., FIAMENI G., CUCCHIARA R.: From Show to Tell: A Survey on Image Captioning. *CoRR* (July 2021). 2
- [SPK\*14] STAVA O., PIRK S., KRATT J., CHEN B., MĚCH R., DEUSSEN O., BENES B.: Inverse Procedural Modelling of Trees: Inverse Procedural Modeling of Trees. *Computer Graphics Forum* 33, 6 (Sept. 2014), 118–131. 2
- [WSW\*18] WU Q., SHEN C., WANG P., DICK A., HENGEL A. V. D.: Image Captioning and Visual Question Answering Based on Attributes and External Knowledge. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 40, 6 (2018), 1367–1381. 2
- [ZLT\*14] ZHEN WANG, LIQIANG ZHANG, TIAN FANG, MATHIOPOULOS P. T., HUAMIN QU, DONG CHEN, YUEBIN WANG: A Structure-Aware Global Optimization Method for Reconstructing 3-D Tree Models From Terrestrial Laser Scanning Data. *IEEE Trans. on Geoscience and Remote Sensing* 52, 9 (Sept. 2014), 5653–5669. 2