Lossy compression of plant architectures

Anne-Laure Gaillard, Pascal Ferraro, Frédéric Boudon, Christophe Godin

Lossy compression of plant architectures

Anne-Laure Gaillard[1], Pascal Ferraro[1,2], Frédéric Boudon[3], and Christophe Godin[3]
[1] Université de Bordeaux 1, LABRI, Talence, France
anne-laure.gaillard@labri.fr,pascal.ferraro@labri.fr,
[2] CNRS - Pacific Institute for Mathematical Sciences, Calgary, Canada
[3] INRIA, Virtual Plants Team, Montpellier, France
frederic.boudon@cirad.fr, christophe.godin@inria.fr

Keywords: plant architecture; self-nestedness; compression; inverse problem

Introduction

Plants usually show intricate structures whose representation and management are an important source of complexity of models. Yet plant structures are also repetitive: although not identical, the organs, axes, and branches at different positions are often highly similar. From a formal perspective, this repetitive character of plant structures was first exploited in fractal-based plant models (Barnsley, 2000; Ferraro et al., 2005; Prusinkiewicz and Hanan, 1989; Smith, 1984). In particular, L-systems have extensively been used in the last two decades to amplify parsimonious rule-based models into complex branching structures by specifying how fundamental units are repeatedly duplicated and modified in space and over time (Prusinkiewicz et al., 2001). However, the inverse problem of finding a compact representation of a branching structure has remained largely opened, and is now becoming a key issue in modeling applications as it needs to be solved to both get insight into the complex organization of plants and to decrease time and space complexity of simulation algorithms. The idea is that a compressed version of a plant structure might be much more efficient to manipulate than the original extensive branching structure. For instance, Soler et al. (2003) have shown that the complexity of radiation simulation can be drastically reduced if self-similar representations of plants are used. Unfortunately, strict self-similarity has a limited range of applications, because neither real plants nor more sophisticated plant models are exactly self-similar. Consequently, we propose in this paper an algorithm that exploit approximate self-similarity to compress plant structures to various degrees, representing a tradeoff between compression rate and accuracy. This new compression method aims at making possible to efficiently model, simulate and analyze plants using these compressed representations.

Representing plant architecture

In many applications of plant modeling, detailed geometric representations of plant architecture are used (Godin, 2000). This comprises a description of both topology and geometry of the plant structure. Topology is usually represented as a graph where each plant component is represented by a vertex in the graph. Edges between vertices denote the adjacency relationship between the corresponding components in the plant. The resulting graph is an unordered tree graph (*i.e.* a graph with no cycles and a specified root vertex and such that no ordering is considered on the set of siblings of any vertex).

To represent plant geometry, a first simple solution consists in using a geometric primitive for each plant component and defining its actual position and orientation in space. Primitives can be cylinders, cone frustums, polygons or more complex geometric models, possibly resized to represent each particular plant component. As an alternative to this *absolute definition* of the plant geometry, the geometry of each component can be recursively defined relatively to the reference frame of its parent component in the plant. This *relative definition* scheme is the basis of the turtle geometric interpretation in L-systems (Prusinkiewicz and Lindenmayer, 1990). Here, the definition of a topological structure of plant components is required in order to define the parent relationship. Let us denote $p(x)$ the parent of a component x, g_x its geometric primitive and $\mathbf{M}_{x/p(x)}$ the transformation defining the scaling, location and orientation of the primitive representing the geometry of x with respect to the reference frame of its parent $p(x)$. To position the primitive of x in space, we must apply the series of transformations to

each point P of g_x to get a corresponding point P' correctly located in the 3D space. If $h(x)$ denotes the height of x in the tree, we thus have:

$$P' = \mathbf{M}_{x/p(x)}\mathbf{M}_{p(x)/p(p(x))}...\mathbf{M}_{p^{(h(x)-1)}(x)/p^{h(x)}(x)}P$$

In this relative definition, the plant geometry is thus defined by a set of triples $\{p(x), g_x, \mathbf{M}_{x/p(x)}\}_{x\in\mathcal{P}}$ made for each component x of its parent component, its geometric primitive and its transformation relative to its parent reference frame. If the topology of the plant is specified by a tree graph, we can assume that each edge from $p(x)$ to x in the graph bears the transformation $\mathbf{M}_{x/p(x)}$ while each vertex bears the component primitive g_x.

Let us consider a plant with N components encoded using machine addresses (pointers) to represent the linking relation that exists between nodes. If we assume that the size of a pointer to a parent component is a constant [4] p, that the size of the primitive representing a segment is a constant s, and that the transformation has a size t, without any compression the amount of memory required to store the plant topology and geometry so far is $N \times (p + s + t)$ bytes.

Exact compression

The problem of constructing a compression of a tree structure has been raised in the early 1970's. For ordered trees (trees in which an order between children of any nodes has been defined), previous algorithms have been proposed to allow the reduction of a tree with complexities ranging in $O(n^2)$ to $O(n)$ (Busatto et al., 2008; Chen and Reif, 1996).Mondet et al. (2009) show also the interest of model compression for data transmission over the web. In there case, a purely geometric compression scheme is proposed where branches are encoded only as differences to a template, simply defined as average geometry of a group of branches with similar complexities. In the case of unordered trees and their applications to plant architecture, only a few attempts to quantify *self-redundancy* of plants have been made in the last ten years (Boudon et al., 2006; Ferraro et al., 2005; Prusinkiewicz, 2004). In a more recent work (Godin and Ferraro, 2009), we introduced the notion of *plant self-nestedness* that makes it possible to formally trade accuracy *vs.* compression rate in the representation of any branching system. In this first approach, the similarity between branching systems was considered to be purely structural. In order to propose a more complete compression method for plants, we extend this notion to take into account geometry and quasi-isomorphisms between trees.

Self-nested trees are such that all their subtrees of a given height are isomorphic. This notion has been derived from the possibility to compress unlabeled unordered trees without loss of information as more compact Directed Acyclic Graphs (DAGs). In any given tree, isomorphisms between subtrees (*i.e.* subtrees with exactly the same topological structure) can be exploited to compress the tree by suppressing completely the related redundancies of tree structure (Godin and Ferraro, 2009). If roots of isomorphic subtrees are *merged*, the resulting equivalent structure is a DAG, more compact but strictly equivalent to the original tree. The topological compression is thus said to be exact.

To extend this exact compression scheme in order to integrate geometry, let us consider the relative definition of geometry introduced above. We shall now say that two branching structures are isomorphic if they have both the same structure and the same geometric attributes on all their vertices and edges. In the relative definition of plant geometry, we can observe that this may be readily exploited. Although the position and orientation of two branching systems A and B can be different, their geometric models can be identical, and then, can be represented only once in the plant architecture representation. It suffices to store the respective transformations in the corresponding edges in the DAG, but keep only one copy of the entire branching system topology and geometry.

The evaluation of the amount of memory required to store this new data structure is trickier and it basically depends on the number of nodes and edges of the DAG. As an illustration, let us consider to simplify a perfect sympodial tree with B branches at each branching point and where all the subtrees

4. In usual implementation, p is generally setup to 4 bytes, however in order to optimized space, p should be proportional to $\log_2 N$

of height h have the same geometry (except for the position and orientation in space). Let us denote H the height the topological tree. Then the number of vertices in the tree is $N = B^H - 1$. The size of the memory required for the classical representation of the tree is thus $(B^H) \times (p + s + t)$ bytes. Now, if we compress the tree as described above, we obtain a *linear DAG* (Godin and Ferraro, 2009). The size of the representation is now $H \times (p + s + B \times t)$. For a dichotomic tree of height 10 (containing $N = 2^{10} = 1024$ vertices), and typical values of $s = 8$ bytes, $p = 4$ bytes, and $t = 16$ bytes, we reduce the size of the tree representation in memory from $1024 \times 13 = 28672$ bytes to $10 \times (4 + 8 + 16 \times 2) = 440$ bytes. The quality of a compression scheme is usually expressed using the compression ration: $cr = \frac{\text{size of the output stream}}{\text{size of the input stream}}$.

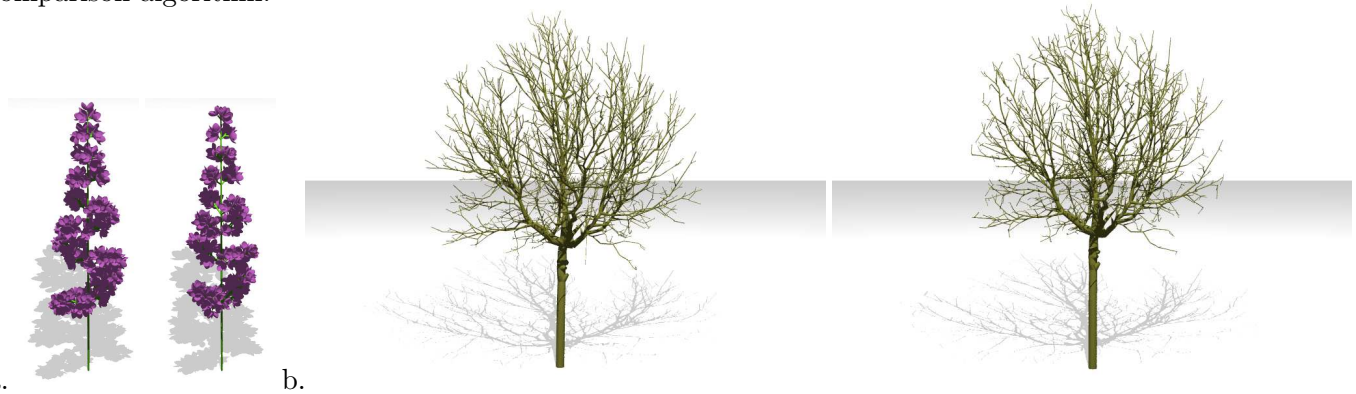
Therefore, for our theoretical perfect dichotomic tree, the compression factor is equal to $cr = 0.2\%$ meaning the DAG uses only 0.2% of the storage space of the original encoding.

Approximate compression

In real plants, branching structures are not perfectly isomorphics and may be isomorphic only to some degree. Interestingly, tree edit-distances can be used to carry the identification of quasi-isomophisms in (ordered or unordered) trees (Ferraro and Godin, 2000; Pinter et al., 2008; Zhang, 1996). The complexity of these algorithms is kept polynomial by the use of bottom-up recursion and dynamic programming. A null distance between two trees (or subtrees) denotes the existence of an isomorphism between the two structures. In our context, the key idea is to use these algorithms to compute the distance from a tree T to itself. Obviously the resulting distance is null, but as a by-product, all the distances between any two subtrees of T are computed recursively in close to quadratic time. Since a null distance between two subtrees denotes isomorphic structures, these algorithms can be exploited to build the compression graph of T (Godin and Ferraro, 2009). More generally, a non-null, but small distance between any two subtrees denotes a quasi-isomorphism between these subtrees. We thus extended the perfect compression scheme by gathering quasi-isomorphic subtrees into p classes of equivalence according to an agglomerative clustering method (Ward, 1963).

For each class, an average subtree was determined. Its topology was chosen among the most represented topology into the class. Its geometry was infered by averaging the geometry of the subtrees of the class (mean diameter, mean length and mean insertion angle) with regards to their parents. This made it possible to define a lossy, approximate, DAG compression of T.

The time complexity of the whole process in $O(|T|^2 deg(T) \log deg(T))$ is finally due to the tree comparison algorithm.



a. b.

Figure 1. a. A model Lilac and its compression such that $cr = 0.58$. b. The original digitalized Walnut and its compression with a compression ratio of $cr = 0.66$.

Applications to digitized and simulated plants

The above approach was tested on different plant architectures. We present here results corresponding to the analysis of a model of inflorescence of common lilac Syringa vulgaris (Fig. 1.a), already described in (Ferraro et al., 2005; Prusinkiewicz et al., 2001) and a digitized walnut tree (Sinoquet

et al., 1997) (Fig. 1.b). On both figures, the left image represent the original plant in which length, diameter, and orientations of each internodes has been stored as well the topology. Right images show respectively a compression ratio of 58% and 66% of the storage space but have little noticeable loss of details or visible artifacts. The lilac inflorescence has been modeled using the notion of branch mapping (Prusinkiewicz et al., 2001): given two branches of the same order, the shorter branch is identical (up to the effects of tropisms) to the top portion of the longer branch. This assumption allows building perfectly self-nested plants and then reaching better compression factors. However, once a certain threshold of compression is passed, compressed plants show increasingly visible defects.

The compression method presented in this paper is based on suppressing the redundancy embedded in branching systems. This technique is complementary to more geometric oriented compression techniques such as developed in (Mondet et al., 2009), where the geometric models of branches themselves are compressed. Both techniques could in principle be combined and lead to extremely efficient compression schemes of plant architectures. For instance, an optimized pointer size of DAG encodings would reduce the compression ratio of both models showed in Fig. 1 to respectively 10% and 28%.

Acknowledgments

The authors would like to thank P. Prusinkiewicz for many and fruitful discussions about compression in plants. This work has been partially sponsored by the French ANR Brasero (ANR-06-BLAN-0045) project and the Multiscale Modeling of Plants associated team (INRIA).

References

Barnsley, M. (2000). *Fractals everywhere*. Morgan Kaufmann Publishers Inc., second edition.

Boudon, F., Godin, C., Puech, O., Pradal, C., and Sinoquet, H. (2006). Estimating the fractal dimension of plants using the two-surface method: An analysis based on 3D-digitized tree foliage. *Fractals*, 14(3):149–163.

Busatto, G., Lohrey, M., and Maneth, S. (2008). Efficient memory representation of XML document trees. *Inf. Syst.*, 33(4-5):456–474.

Chen, S. and Reif, J. H. (1996). Efficient lossless compression of trees and graphs. In *IEEE Data Compression Conference (DCC)*, page 428.

Ferraro, P. and Godin, C. (2000). A distance measure between plant architectures. *Annals of Forest Science*, 57(5/6):445–461.

Ferraro, P., Godin, C., and Prusinkiewicz, P. (2005). Toward a quantification of self-similarity in plants. *Fractals*, 13(2):1–25.

Godin, C. (2000). Representing and encoding plant architecture: A review. *Annals of Forest Science*, 57(5-6):413–438.

Godin, C. and Ferraro, P. (2009). Quantifying the degree of self-nestedness of trees. Application to the structural analysis of plants. *IEEE/ACM TCBB*, in press.

Mondet, S., Cheng, W., Morin, G., Grigoras, R., Boudon, F., and Tsang Ooi, W. (2009). Compact and progressive plant models for streaming in networked virtual environments. *ACM TOMCCAP*, 5(3):1–22.

Pinter, R. Y., Rokhlenkoa, O., Tsurb, D., , and Ziv-Ukelson, M. (2008). Approximate labelled subtree homeomorphism. *Journal of Discrete Algorithms*, 6(3):480–496.

Prusinkiewicz, P. (2004). *Thinking in Patterns: Fractals and Related Phenomena in Nature*, chapter Self-similarity in plants: Integrating mathematical and biological perspectives, pages 103–118. Novak.

Prusinkiewicz, P. and Hanan, J. (1989). *Lindenmayer systems, fractals and plants*. Springer.

Prusinkiewicz, P. and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. Springer, 2nd edition.

Prusinkiewicz, P., Mündermann, L., Karwowski, R., and Lane, B. (2001). The use of positional information in the modeling of plants. In *Proceedings of SIGGRAPH 2001*, pages 289–300.

Sinoquet, H., Rivet, P., and Godin, C. (1997). Assessment of the three-dimensional architecture of walnut trees using digitising. *Silva Fennica*, 31(3):265–273.

Smith, A. (1984). Plants, fractals and formal languages. *Computer Graphics*, 18(3):1–10.

Soler, C., Sillion, F., and de Reffye, P. (2003). An efficient instantiation algorithm for simulating radiant energy transfer in plant models. *ACM Trans. Graph.*, 22(2):204–233.

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Zhang, K. (1996). A constrained edit distance between unordered labeled trees. *Algorithmica*, 15(3):205–222.