



# Temporal Upsampling of Point Cloud Sequences by Optimal Transport for Plant Growth Visualization

Tim Golla,<sup>1</sup> Tom Kneiphof,<sup>1</sup> Heiner Kuhlmann,<sup>2</sup> Michael Weinmann<sup>1</sup> and Reinhard Klein<sup>1</sup>

<sup>1</sup>Institute of Computer Science II - Visual Computing, University of Bonn  
{golla, kneiphof, mw, rk}@cs.uni-bonn.de

<sup>2</sup>Institute of Geodesy and Geoinformation, University of Bonn  
heiner.kuhlmann@uni-bonn.de

## Abstract

*Plant growth visualization from a series of 3D scanner measurements is a challenging task. Time intervals between successive measurements are typically too large to allow a smooth animation of the growth process. Therefore, obtaining a smooth animation of the plant growth process requires a temporal upsampling of the point cloud sequence in order to obtain approximations of the intermediate states between successive measurements. Additionally, there are suddenly arising structural changes due to the occurrence of new plant parts such as new branches or leaves. We present a novel method that addresses these challenges via semantic segmentation and the generation of a segment hierarchy per scan, the matching of the hierarchical representations of successive scans and the segment-wise computation of optimal transport. The transport problems' solutions yield the information required for a realistic temporal upsampling, which is generated in real time. Thereby, our method does not require shape templates, good correspondences or huge databases of examples. Newly grown and decayed parts of the plant are detected as unmatched segments and are handled by identifying corresponding bifurcation points and introducing virtual segments in the previous, respectively successive time step. Our method allows the generation of realistic upsampled growth animations with moderate computational effort.*

**Keywords:** point-based animation, animation, point-based methods, methods and applications, scientific visualization, visualization, optimal transport, temporal upsampling, plant growth, point clouds

**ACM CCS:** • Computing methodologies → Shape modelling; Animation; Point-based models; • Human-centred computing → Scientific visualization

## 1. Introduction

Modelling, visualizing and analysing plant growth is of great relevance in various domains including biology, agriculture and computer graphics with applications in movie and game productions. Therefore, the coupling of respective plant models to measurements of growing plants allows a better understanding regarding the underlying growth processes for individual plant species. While a conventional video recording has shown great potential for plant monitoring, its limitations regarding the flexibility of inspecting plants from arbitrary views and performing three-dimensional (3D) measurements can only be overcome by directly considering a complete 3D scan of the plant of interest. However, capturing the plant growth with all its details relies on a (ideally) temporally continuous scanning of the plant state which is hindered by the fact that capturing the plant state at a certain time step usually involves a time-consuming manual acquisition. This induces larger time intervals between successive measurements. As a plant often has grown

significantly between successive scans, a smooth animation of the growth process from coarsely taken discrete measurements is not easily possible. Additionally, suddenly arising structural changes induced by bifurcation or the evolution of leaves have to be taken into account when aiming for a smooth completion of the intermediate states between successive measurements. Such topological changes of the underlying ground truth model and the fact that the measurement process inherently does not guarantee perfectly corresponding points or point numbers for successive point cloud scans cannot be handled by trajectory-based deformation and morphing approaches. These require exact point correspondences that can only be reliably obtained for small deformations and motions and, hence, are by design not suitable for analysing growth processes with changing topology [XSWL15]. Further limitations of previous work on deformations based on exact point correspondences regarding their applicability on growth processes include the assumption of piecewise rigid motion as used for object tracking [BIZ18], the requirement of a 3D object template that is deformed and fitted



**Figure 1:** Temporal upsampling of two synthetically generated tree point clouds obtained with our method. The original point clouds for the time steps  $t_i$  and  $t_{i+1}$  are highlighted with a grey background and the in-between states have been computed based on our method.

to the point clouds in adjacent time steps using respective priors (without enforcing temporal coherence) [ZFG\*17], the involvement of a visual hull prior that biases the optimization in the context of mesh-based approaches [LLV\*12] or the need for large databases required by learning-based methods [WLX\*18] that are hard to acquire due to the time-consuming nature of the scanning process and the growth of the plants themselves.

In this paper, we address the aforementioned limitations by a novel method that allows temporal upsampling of coarsely acquired point cloud sequences of growth processes including topological changes without requiring exact point correspondences, shape priors or huge databases. For this purpose, our approach involves an initial point cloud segmentation and the generation of a hierarchy on the segments for each of the individual scans, the matching of the segment hierarchies of successive scans and the final computation of a segment-wise regularized optimal transport to morph the point sets associated with matched segments. Newly grown and decayed parts of the plant, that are detected as unmatched segments, are handled by identifying corresponding bifurcation points and introducing virtual segments in the previous, resp. successive time step. Furthermore, we present a heuristic segmentation approach that is particularly suitable for thin structures as occurring for maize plants or tree structures. By using a memory-efficient formulation of the optimal transport problem, our approach is also capable of handling large point clouds. As demonstrated in the evaluation, our approach yields smooth temporal upsamplings, allowing for realistic growth animations with moderate computational effort.

## 2. Related Work

Temporal upsampling for modelling and visualizing growth and deformation processes can be approached based on several strategies such as procedural modelling, template-based approaches, correspondence-based approaches and data-driven approaches. In the following, we will discuss respective developments in the context of plant growth simulation. In addition, we provide a review of applications of optimal transport between data distributions in the context of graphics applications as the use of optimal transport allows our method to overcome the need for perfect one-to-one correspondences or templates.

**Procedural modelling for plant growth.** Modelling structures and phenomena from sets of rules has received a lot of attention in the contexts of facade modelling and plant modelling. However, the underlying rules are often not directly known and

have to be derived, for example, using inverse procedural modelling. In the context of modelling plant growth, L-systems have been extended by growth rules such as relational growth grammars [KKBS05] or tree-species-specific growth rules [LKM05], component-wise time functions of logistic growth [CAJBS05], guiding vectors to provide local control over branch orientation during tree growth [XM15], generic rules to provide sequences of symbols and branching [SBM\*10], the localization and tracking of topological events like budding and bifurcation [LPY\*12], growth models [SPK\*14] or plant modules for capturing species-specific branching pattern characteristics that can be combined to model whole individual plants [MHS\*19]. While individual instances can be derived via the use of meaningful parameters and rules, a smooth interpolation between individual objects or process instances is challenging as smooth deformations of plants cannot simply be achieved by interpolating within the parameter space. In addition, inverse procedural growth models may fail to capture the nature of input examples that do not capture all of the relevant growth phenomena and may be limited to certain species.

**Template-based and mesh-based interpolation.** Early work on template-based growth analysis and visualization include the modelling of plant growth in terms of flexible spatio-temporal templates to represent non-rigid motions [LXZ\*97]. Regarding mesh morphing, the morphing based on local transformations between source and target meshes such as As-Rigid-As-Possible Shape Interpolation [ACOL00] and affine transformations [SZGP05] is not capable of handling large deformations. The latter also applies to morphing based on intrinsic shape information such as differential coordinates [Ale03], linear rotation-invariant (LRI) coordinates [LSLCO05], Laplacian coordinates with invariance to rotation and isotropic scaling [SCOL\*04], mean curvature flow Laplacian coordinates [HLW07], gradient fields [XZWB06] or the combination of near-rigid priors with a local gradient field interpolation [CL09]. Further work used spatio-temporal skeleton-based models [ZLLZ12]. Bansal *et al.* [BT18] introduced a mesh morphing and interpolation framework based on the Lie Bodies representation. For large deformations, the authors perform mesh segmentation and apply the Lie Bodies framework on the respective components. Furthermore, Chen *et al.* [CFB16] presented a method for the simultaneous morphing between two mesh sequences for blending motions and interpolating shapes. Zhao and Barbic [ZB13] segment plants from the geometric information to establish a structural hierarchy that is connected with an Finite Element Methods (FEM) simulation model to allow deformations. However, template-based

approaches rely on the availability of 3D object templates that are deformed and fitted to the observations in adjacent time steps as well as adequate priors and larger deformations cannot be handled easily.

**Correspondence-based interpolation.** Instead of exploring templates and shape priors given by the mesh representation, several approaches focused on unstructured representations such as point clouds, where the missing topological information complicates the morphing. This has been addressed based on improving the robustness of correspondences based on local mappings between the projections of the input point clouds onto a common parameter space which can be performed in a segment-wise manner for complex-shaped objects [XZPF04]. Alternatives include the guidance of the morphing by registration and warp functions [THCF06], exploiting differential geometry and continuum mechanics to get physically meaningful transitions [BGQ05] or the guidance of the interpolation based on gradient fields [TZZ09]. Wang *et al.* [WZH12] presented a method for morphing point-sampled geometry based on parameterizing the point clouds onto spheres and then comparing features. Solenthaler *et al.* [SZP07] refine irregularly sampled, dynamic points in the context of particle-based fluid simulation while preserving edges and avoiding point collisions. Further work on temporal upsampling by interpolating between temporally consecutive point cloud measurements includes the use of 4D spatio-temporal models as proposed in the context of large dynamic urban scenes [JBB13]. Lie *et al.* [LLV\*12] incorporate the visual hull as a shape topology prior and surface bending energy for shape completion. Subsequently, inter-frame correspondences are used to obtain temporally coherent dynamics. The approach can handle larger deformations and certain topological changes, however, the topology of the output is tied to the possibly incorrect topology of the visual hull. Li *et al.* [LFM\*13] exploit a forward-backward analysis for spatial and temporal analysis of 4D point clouds for tracking plant parts over time. The co-segmentation framework proposed by Yuan *et al.* [YLX\*16] decomposes articulated point cloud sequences into near-rigid moving parts. Segments derived using trajectory analysis are progressively propagated to neighbouring frames and merged through all frames using space-time segment grouping. This results in robustness regarding noise, occlusions as well as variations of pose and view. Similarly, Bertholet *et al.* [BIZ18] address motion segmentation from RGB-D videos by representing motion as a sequence of rigid transformations through all input frames in an energy optimization framework. Vlachos *et al.* [VLSM18] focus on the improvement of animation reconstruction based on rank minimization while imposing spatial coherence between successive frame clusters. Furthermore, Xu *et al.* [XSWL15] approach trajectory-based shape interpolation by solving Poisson equations defined on a domain mesh. Interpolated shapes are reconstructed from interpolated gradient fields that exploit both point coordinates and surface orientations instead of directly using point coordinates. The technique shows stable area and volume changes and overcomes the shrinkage problem of linear shape interpolation. Zheng *et al.* [ZFG\*17] used a template-based dynamic tracking algorithm for the temporal reconstruction of geometry and deformation in point cloud sequences, tailored to the reconstruction of flower petals. The use of specific priors to assist the tracking allows handling occlusions and collision-based interactions at the cost of requiring a suitable template. However,

trajectory-based deformation/morphing approaches require exact point correspondences that can usually only be reliably obtained for small deformations/motions and, hence, are by design not suitable for analysing growth processes with changing topology due to newly added branches or leaves [XSWL15]. Furthermore, the points in adjacent point clouds may not exactly represent corresponding surface parts as the point distribution is determined by the scanning process. Our approach overcomes these limitations by considering the point distributions instead of the point cloud itself. As a result, we can circumvent the establishment of exact point correspondences that are hard to establish for growing objects with possibly changing topology.

**Data-driven and learning-based interpolation.** Data-driven approaches exploit the availability of large collections of related exemplars to describe the characteristic deviations between objects and, hence, to guide the interpolation between object states, which allows handling large deformations. This has been used in the context of mesh interpolation based on patch-based LRI coordinates [BVGPO9], using an iterative Gauss–Newton method for shape interpolation [FB11], solving a shortest path problem in the local linear subspaces derived from a given model database [GLHH13], using low-dimensional shape spaces that exploit the specific structure of the interpolation problem [VTSSH15] and using rotation/translation-invariant representations that define plausible deformations in a global continuous space [GCLX17]. Furthermore, the approach by Wang *et al.* [WLX\*18] explores a shape space technique that relies on the correspondence computation between trees of different structure and using geodesics to obtain the continuous blending between trees. However, such collections of examples that represent the complete range of possible variations are often not available due to the impractically time-consuming capture processes.

**Optimal transport in graphics and vision.** Optimal transport has been used successfully in various computer graphics and computer vision applications. Early applications include its use for the measurement of the similarity of images by defining the earth mover's distance between colour distributions and between texture feature distributions as used by Rubner *et al.* [RTG98, RTG00]. Furthermore, optimal transport has been successfully used for colour transfer in 2D images based on the regularization with the preservation of the gradient of the original picture [PKD07], an approximate variational formulation using an approximate Wasserstein constraint on colour statistics and a generic geometric-based regularization [RP11], regularized discrete optimal transport within a unified convex variational framework [FPPA14], relaxed optimal transport using a regularization that considers the spatial distribution of colours [RFP14], by using separate illuminant matching and optimal transfer of dominant colours regularized by thin plate splines [FSDH14] and extending standard entropy regularization based optimal transport to unbalanced problems [CPSV16].

In addition, optimal transport has been applied for interpolation. Respective examples include the interpolation between textures [RPDB11], reflectance models such as bidirectional reflectance distribution functions (BRDFs) [BVDPPH11, SDGP\*15] or bidirectional texture functions (BTFs) [GK18] based on statistical representations [GK17] as well as geometric models in terms of

volumetric representations [SDGP\*15]. Digne *et al.* [DCSA\*14] used optimal transport for surface reconstruction and simplification from point clouds and Merigot *et al.* [MMT18] investigated optimal transport between a simplex group and a point cloud.

**Plant segmentation.** The numerous approaches for plant segmentation include the combination of colour information and 3D models [ADT11, MBW\*18], adapted surface feature-based techniques [PDMK13], model-based approaches based on cylinder representations of stems and separate segmentation tailored to leaves [GDHB17], skeleton-based stem and leaf point recognition approaches 2019, facet region growing approaches after an initial oversegmentation [LCT\*18] as well as automated, data-driven approaches for plant structure segmentation based on geometric features and Random Forests classifiers (e.g. [DNC\*18]) or geometric features and clustering 2015.

### 3. Temporal Upsampling of Point Cloud Sequences Using Optimal Transport for Plant Growth Visualization

Given a time series of point cloud scans obtained by a 3D scanner, our goal is to produce a temporal upsampling of the plant growth that happened between the individual measurements that results in a realistic, smooth plant growth visualization. As illustrated in Figure 2, our method consists of the following steps: Initially, the scans are coarsely aligned to each other (Section 3.1). While it is possible to set up and solve a single optimal transport problem for the whole dataset, the result is not satisfactory, as artefacts like unnatural-looking breaking up of plant parts occur (see Figure 3). In order to alleviate this, we perform the following steps: First, a hierarchical segmentation is derived for each scan individually (Section 3.2). Afterward, the corresponding segments of successive scans are matched (Section 3.3), which is facilitated by the initial coarse alignment. Finally, for each pair of corresponding segments, a regularized optimal transport problem is set up (Section 3.4). Its solution delivers the information required for a temporal upsampling, which is obtained by interpolation.

#### 3.1. Coarse alignment

Continuously monitoring plant growth with a fixed arrangement of scanners is complicated due to self-occlusions induced by growing plant parts and results in incomplete 3D scans. As a consequence, detailed plant scans require a manual scanning process, ideally with a movable scanner platform that can be guided by a user to obtain complete scans. As this may require the relocation of plants to a certain scanner platform, successive scans have to be aligned in a common coordinate system in order to register the individual plant segments and to allow the consideration of changes induced by the growth process. For this purpose, we align each scan in the time series coarsely via the iterative closest point algorithm [BM92] to its predecessor. We assume that scans in a time series have a consistent scaling.

#### 3.2. Hierarchical segmentation

If used without any further constraints, the later described OT solution will lead to an unrealistic breaking up of the dataset in the

upsampling as shown in Figure 3. To avoid this problem, we segment the respective point cloud  $P_i$  of each time step  $t_i$  into segments  $s \in S_{P_i}$ , onto which we restrict the optimal transport problem setup. These segments are not necessarily matched. Additionally, new segments might appear due to growth or might disappear due to cropping. In order to be able to better match the segments and handle growth and decay events, we introduce a hierarchy  $H_{P_i}$  on the segments  $S_{P_i}$  of each point cloud  $P_i$ .

##### 3.2.1. Creating a hierarchy on existing segmentations

In principle, different segmentation techniques may be used to derive the segmentation  $S_{P_i}$ . However, they should be suitable for the expected plant structures. For a given segmentation of a plant into its semantic entities like stem, branches, leaves or blossom petals, we first identify the root segment of the hierarchy  $H_{P_i}$  as the segment containing the point with the lowest vertical coordinate. Alternatively, the respective segment may also be directly chosen by the user.

Then, successive nearest neighbour searches are performed to populate the hierarchy  $H_{P_i}$ . All segments containing points that are closer than a user-specified threshold to points of the currently considered segment are added as children of this segment in the hierarchy. This is iterated until all segments have been processed. An exemplary decomposition into a segment hierarchy derived for a synthetic tree model is illustrated in Figure 4. Here, the ground truth segmentation of the tree into the stem and individual branches was directly obtained from the synthesis and we chose the trunk as the root segment, which is also the segment with the lowest vertical coordinate. In the case of a given segmentation with no meaningful root segments, such as in the Komatsuna dataset [USM\*17], we introduce a virtual, empty root segment and add the other segments as direct successors in the hierarchy  $H_{P_i}$ . An example is shown in Figure 5.

##### 3.2.2. Simultaneous segmentation and hierarchy generation for thin structures

For plants with thin structures, we can derive the segment hierarchy  $H_{P_i}$  together with the corresponding segmentation  $S_{P_i}$  without the requirement of an initial segmentation of a plant into its parts.

First, we build a  $k$ -nearest neighbour graph on the point cloud, where  $k$  is a user-defined parameter. We found  $k = 15$  to deliver good results in our experiments. On this graph, we compute a minimum spanning tree (MST). We assume the point with the smallest value along the vertical axis to be the root point of the MST, thus inducing a direction on the graph. When operating on single plants, this is a reasonable assumption as this point corresponds to the location where the stem exits the ground.

For each node  $v$  in the MST, we compute the longest possible path length  $m_v$  to a leaf node in the MST subtree rooted at  $v$ . Starting with the root, we traverse the MST. If we encounter a node  $v$  in the tree where the number of successors  $u \in \text{succ}(v)$  with longest path length  $m_u > c$  is more than one, for a user-defined threshold  $c$ , we found a bifurcation. All of  $v$ 's successor nodes  $u$  with  $m_u > c$  are then considered to be roots of trees corresponding to new biological

branches. The edges from  $v$  to these successors are inserted into a set  $E$  of edges to be removed from the MST, which also define the connectivity between created segments. After visiting all nodes, the edges  $e \in E$  are removed from the MST and the connected components of the resulting graph, respectively the corresponding 3D points, define the segmentation  $S_{P_i}$  on the point cloud  $P_i$ . At the same time, a separate graph representing the hierarchy  $H_{P_i}$  of the segments in  $S_{P_i}$  is created. The root segment is chosen such that it contains the MST root node. Two segments  $s_1, s_2 \in S_{P_i}$  are connected in  $H_{P_i}$  if an edge  $(u, v) \in E$  exists with  $u \in s_1$  and  $v \in s_2$ . An example of the resulting segmentation and the corresponding hierarchy graph is depicted in Figure 6.

### 3.3. Segment matching

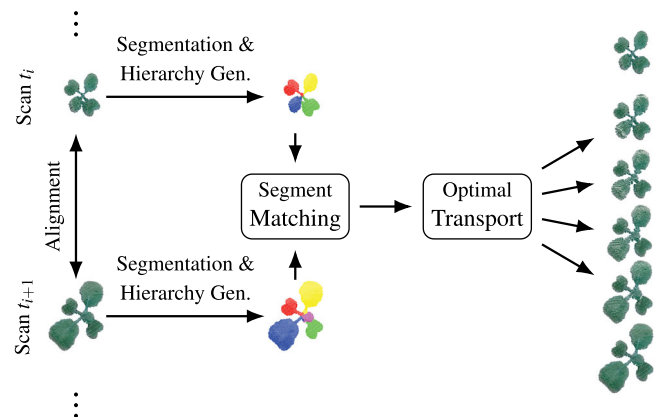
Given segmentations of two point clouds  $P$  and  $Q$  for successive time steps, we match the segments of  $P$  and  $Q$  using the hierarchies  $H_P$  and  $H_Q$  built in the previous step. We differentiate the following cases:

1. The labels are consistent and the number of segments is identical. This is usually the case with manually segmented data, when no new segments have grown. An example is shown in Figure 5(a).
2. The labels are consistent and the number of segments varies. This is usually the case with manually segmented data, when new structures represented by new segments have grown or parts have decayed. An example is shown in Figure 5(b). For growth scenarios, small new virtual segments are added to the first point cloud  $P$  and matched to the otherwise unmatched segments of the second point cloud  $Q$ . These virtual segments are used to generate a growth animation. For decay scenarios, we add a point set that represents the decayed leaf. As a heuristic, we generate this point set by moving the segment's points towards the smallest vertical coordinate, which is perceived as the floor. Alternatively, this segment may be simply faded out. We emphasize that the respective choice depends on the preferences of the user.
3. The labels are inconsistent. This can occur if the labelling has been generated automatically with no additional information. In this case, we perform a matching algorithm, which is described in Section 3.3.1. Growth or decay may also occur and is handled analogously to the scenario with a consistent labelling.

#### 3.3.1. Segment matching with inconsistent labelling

If different IDs are assigned to corresponding segments  $s_P, s_Q$  in consecutive scans, we use the Wasserstein distance  $d_w(s_P, s_Q)$  (see Equation (1)) as a similarity measure between the two segments  $s_P, s_Q$ , that is, the cost of warping the point distributions onto each other. Initially, only the root segments of the hierarchy are considered to be matched. We then alternate between merging and matching segments in the hierarchy, which is explained in detail in the following paragraphs.

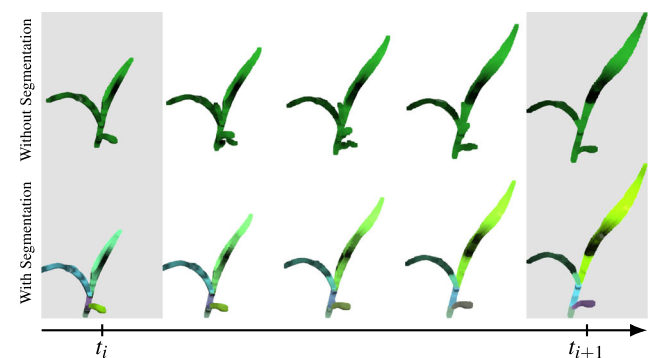
**Merging.** For each newly created match of the segments ( $s_P \in H_P, s_Q \in H_Q$ ), we try to merge the segment in the second point cloud  $s_Q$  with one of its children  $c_Q \in \text{succ}(s_Q)$ , as long as this reduces



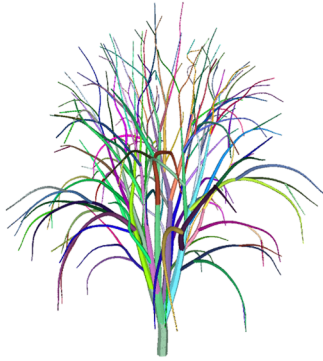
**Figure 2:** Overview of our method: First, a hierarchical segmentation is generated for each scan, which is followed by a matching of the segments of point clouds obtained for successive measurements at the time steps  $t_i$  and  $t_{i+1}$  and the computation of segment-wise optimal transport solutions used for temporal upsampling.

the Wasserstein distance  $d_w(s_P, s_Q)$  between the associated point clouds. The rationale behind this step is to adapt to different segment hierarchies induced by growth or decay processes. For example, a new biological branch might have appeared in the later states of the plant growth, which leads to the splitting of the segment of the previous scan at the location where the new part starts growing. This process is repeated until no further improvement is possible.

**Matching.** When no further merging steps are possible, we create a matching of the remaining child segments. Since the scans are segmented individually, the association of corresponding parts between the segmentations has still to be computed in order to be able to set up an optimal transport problem between segment pairs. An



**Figure 3:** Comparison between animations generated from the same data without segmentation (top row) and with segmentation (bottom row). Images with a grey background correspond to the input scans. Images with a white background represent upsampled data. In the case without segmentation (top row), the plant breaks up in an unrealistic way. Using segmentation (depicted in the bottom row), this problem is solved, because transport is restricted to take place only within a segment.



**Figure 4:** Exemplary segment hierarchy obtained from an initially provided plant segmentation: The artificially generated tree has a natural segmentation. We impose a hierarchy by heuristically or manually selecting a root (the tree trunk) and then performing successive nearest neighbour searches in order to generate a hierarchy on the segments.

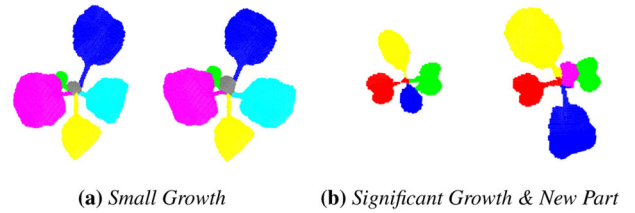
example of the segmentations and the associated graphs of two consecutive scans is shown in Figure 6.

In order to resolve this ambiguity, we perform the following matching step: We compute the pairwise Wasserstein distances  $d_w(c_P, c_Q)$  between the point clouds associated with the child segments  $c_P \in \text{succ}(s_P)$  and  $c_Q \in \text{succ}(s_Q)$ . The pairwise costs are stored in a cost matrix. The optimal association of the child segments is computed with optimal transport.

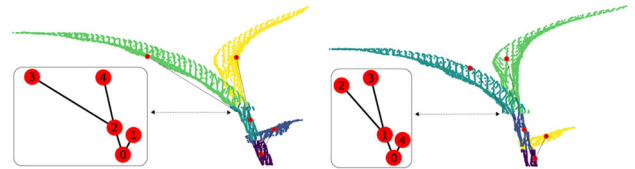
**Segment growth and cropping/decay.** A plant’s growth may yield new parts like new branches or leaves. Analogously, there may also be a loss of parts such as the falling of leaves or the breaking of branches. This is reflected in the segment hierarchy. If the current target segment  $s_Q$  has more children than  $s_P$ , a new part has grown. If  $s_Q$  has fewer segments, plant parts have been cropped or decayed and fallen off the plant. In order to handle these cases, we create virtual matching segments. The child segment with the largest Wasserstein distance to any child segment of the other point cloud is considered unmatched and a virtual segment is generated in the other scan.

For growth scenarios, the point cloud associated with the virtual segment is generated by searching  $l$  points in the point cloud of  $s_P$ , that are closest to the points of the unmatched segment of point cloud  $Q$ .  $l$  has to be chosen by the user, depending on the density of the point cloud. In our experiments, we found values between 10 and 100 to deliver good results. These points are duplicated, labelled as a new segment and added to the segment hierarchy. The newly generated virtual segment is matched with the previously unmatched segment. This procedure is repeated for all unmatched segments.

For cropping or decaying scenarios, our goal is to generate an animation of the cropped part falling to the ground. Note that other solutions like the fading out of the respective points or their immediate removal are also viable and can be generated analogously. We duplicate the points of the cropped segment, but set their vertical axis coordinate to the point cloud’s minimum. These points are



**Figure 5:** Labelled point cloud scans from the Komatsuna dataset [USM\*17]. (a) shows two successive point cloud scans which are consistently labelled and only a small amount of growth happened. In this case, no further matching of the segmentations is necessary. (b) shows two point clouds where there has been significant growth in the meantime. While consistent labels are assigned to the individual parts, for the newly grown part a small new virtual segment is added to the first point cloud and matched to the unmatched segment of the second point cloud.



**Figure 6:** Segmented point clouds of two successive scans of a maize plant. The hierarchy graphs are plotted on top of the plants and shown additionally in the boxes. Although the numberings of the nodes, indicated in terms of the segment colours, differ, our method allows a correct matching of corresponding segments.

added as a new segment to  $Q$ , which is then matched to the previously unmatched segment of  $P$ .

### 3.4. Optimal Transport plan for two point clouds

So far, we have outlined how point clouds are segmented and how the segments are associated between different time steps. Based on this information, we perform temporal upsampling by warping, respectively, matching the point distributions of previously matched segments. Let  $P$  and  $Q$  be their respective point clouds, which we want to match onto each other using optimal transport. Solving the optimal transport problem for given point clouds  $P$  and  $Q$  involves the computation of the Wasserstein distance  $d_w(P, Q)$ , which is essentially the cost of the optimal transport solution, that is, the cost of warping one distribution onto the other, and the temporal upsampling is performed by using the solution for the interpolation of the point clouds.

In the (discretized) optimal transport problem, each point  $p_a \in P$ ,  $1 \leq a \leq |P|$  and  $q_b \in Q$ ,  $1 \leq b \leq |Q|$  is assigned a mass  $m_a$  and  $m_b$ , respectively, such that  $\sum_{1 \leq a \leq |P|} m_a = 1 = \sum_{1 \leq b \leq |Q|} m_b$ . The goal is to find an optimal transport plan, which transports all mass from  $P$  to  $Q$ , minimizing the cost, which is given by a cost matrix  $C \in \mathbb{R}^{|P| \times |Q|}$ , with entries  $C_{ab}$  defining the cost of transporting a piece of mass from  $p_a$  to  $q_b$ , where one common choice is given by the euclidean distance. The optimal solution is a transport matrix  $T \in \mathbb{R}^{|P| \times |Q|}$  with

entries  $T_{ab}$  denoting the amount of mass transported from  $p_a \in P$  to  $q_b \in Q$ , chosen such that the total transport cost is minimized. The optimal transport problem can be written as a linear program:

$$\begin{aligned} d_w^2(P, Q) = \min_T \sum_{1 \leq a \leq |P|} \sum_{1 \leq b \leq |Q|} T_{ab} \cdot C_{ab} \quad (1) \\ \text{s.t. } \sum_{1 \leq a \leq |P|} m_a \cdot T_{ab} = m'_b \quad \forall 1 \leq b \leq |Q| \wedge b \in \mathbb{N} \\ \sum_{1 \leq b \leq |Q|} m'_b \cdot T_{ab} = m_a \quad \forall 1 \leq a \leq |P| \wedge a \in \mathbb{N}, \end{aligned}$$

which minimizes the (discretized) squared 2-Wasserstein distance  $d_w(P, Q)$  between the point clouds  $P$  and  $Q$ . For small point clouds, this problem can be solved by standard algorithms.

**Transport problem setup.** In order to compute the optimal transport from point cloud  $P$  to  $Q$ , we have to define the point masses  $m_a$  and  $m'_b$ , and the entries  $C_{ab}$  of the cost matrix  $C$ . Since we do not have any particular preference for certain points, we define their masses uniformly as  $m_a = 1/|P|$  and  $m'_b = 1/|Q|$  for each point  $p_a \in P$  and  $q_b \in Q$ , respectively.

For each point  $p_a \in P$ , we collect the  $k$  nearest neighbours  $q_b \in Q$ . This graph is represented by a sparse matrix  $C_P$ , where entry  $C_{P,ab} = d(p_a, q_b)$  is the distance between  $p_a$  and  $q_b$  if  $q_b$  is one of the nearest neighbours. If  $q_b$  is not among the  $k$  nearest neighbours of  $p_a$ , we define  $C_{P,ab} = \infty$ . The matrix  $C_P$  is the cost matrix of transport from the source cloud's points to the target cloud's points. This is a reasonable approximation to the true dense cost matrix, since points in  $Q$  that are far away from a point  $p_a$  are unlikely to be used as targets for transporting mass from  $p_a$  in the solution of the full optimal transport problem. The cost for transporting from  $p_a$  to non-neighbour points is thus infinite.

It is possible that some points of the target point cloud are not among the nearest neighbours. Since we want to compute a solution to the optimal transport problem that transports from each point and to each point, this is not desirable. In order to alleviate this, we also compute the  $k$  nearest neighbours  $p_a \in P$  for points  $q_b \in Q$ . This is also represented by a sparse matrix  $C_Q$ , where entry  $C_{Q,ab} = d(p_a, q_b)$  represents the distance between  $p_a$  and  $q_b$  if  $p_a$  is among the nearest neighbours of  $q_b$ . The resulting cost matrix  $C$  is then defined as the coefficient-wise minimum of  $C_P$  and  $C_Q$ .  $k$  determines the sparsity of the matrix and is user-defined. We found  $k = 15$  to deliver good results in our experiments. Thus, in the resulting graph, there are at least  $k$  connections from each point in the source cloud to points in the target cloud and also at least  $k$  connections from each point in the target cloud to points in the source cloud.

**The Sinkhorn algorithm for regularized optimal transport.** The optimal transport problem is approximated by the regularized optimal transport problem, which we solve using the Sinkhorn algorithm [Cut13]. In the regularized transport problem, the objective function in Equation (1) is modified to

$$\min_T \sum_{1 \leq a \leq |P|} \sum_{1 \leq b \leq |Q|} T_{ab} \cdot C_{ab} - \gamma H(T), \quad (2)$$

where  $H(T)$  is the entropy of the matrix  $T$ , defined as

$$H(T) = - \sum_{1 \leq a \leq |P|} \sum_{1 \leq b \leq |Q|} T_{ab} \log T_{ab}. \quad (3)$$

Inserting Equation (3) into Equation (2) yields

$$\min_T \sum_{1 \leq a \leq |P|} \sum_{1 \leq b \leq |Q|} T_{ab} \cdot C_{ab} + \gamma \sum_{1 \leq a \leq |P|} \sum_{1 \leq b \leq |Q|} T_{ab} \log T_{ab}. \quad (4)$$

It can be shown [Cut13] that minimizing Equation (4) leads to the following condition for the elements of  $T$ :

$$T_{ab} = u_a K_{ab} v_b, \quad (5)$$

where  $K_{ab} = \exp(-C_{ab}/\gamma)$  are the entries of a matrix  $K$  and  $u = (u_a)_a$ ,  $v = (v_b)_b$  are unknown vectors.  $K$  is also a sparse matrix and the infinite distances in  $C$  become 0 in  $K$ . This property is important in order to be able to handle large point clouds. A full cost matrix for large point clouds would not fit into the computer's memory. We initialize the vectors  $u = (u_a)_a = \mathbf{1}$ ,  $v = (v_b)_b = \mathbf{1} \forall a \in A, b \in B$ . The Sinkhorn algorithm then finds an approximate solution by iterating the following steps:  $m = (m_a)_a$ ,  $m' = (m'_b)_b$

1.  $u \leftarrow m \oslash Ku$
2.  $v \leftarrow m' \oslash Kv$ ,

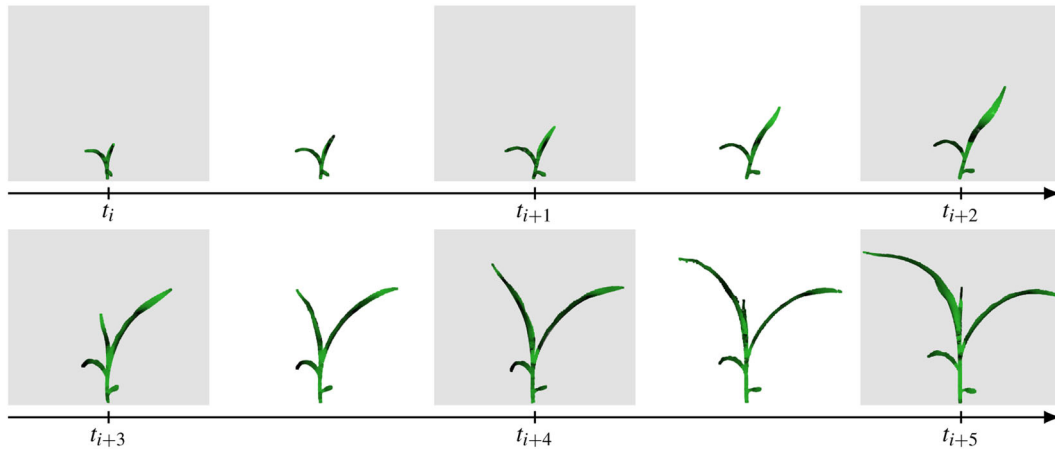
where  $\oslash$  denotes an element-wise division. The elements of the approximate solution matrix  $T$  are then  $T_{ab} := u_a K_{ab} v_b$ .

#### Using the solution of the transport problem for interpolation.

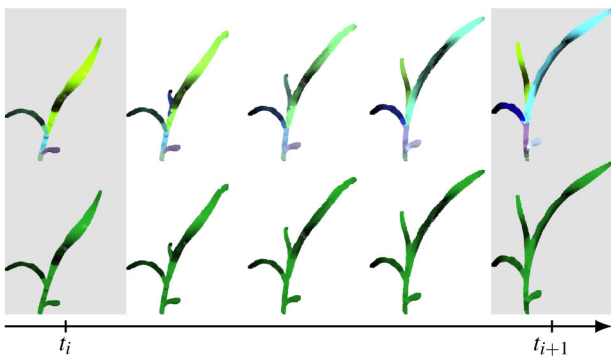
The solution of the transport problem is a transport matrix  $T$  of the same size as  $C$ , where each entry  $T_{ab}$  denotes the amount of mass that has to be transported from  $p_a$  to  $q_b$ . Since we are interested in an interpolation animation that is visually appealing, we set up the following requirements:

- No point can disappear, that is, each point of  $P$  has to move to at least one point of  $Q$ .
- No point can appear out of nowhere, that is, for each point of  $Q$  there must be at least one point of  $P$  moving to it.
- For efficiency, as few as possible point pairs should be used for the transport.

A solution to the first two requirements is to use each entry  $T_{ab} > 0$  of the transport matrix and interpolate  $p_a$  to  $q_b$ . If there are several  $T_{ab} > 0$  for fixed  $a$ , we create copies of  $p_a$ , which follow different animation paths to their respective  $q_b$ . Since the cost matrix  $C$  allows up to  $k$  targets for each  $p_a$ , the number of transported points is at most  $k \cdot (|P| + |Q|)$  instead of  $|P| \cdot |Q|$ . Yet, this may be an unnecessarily huge amount of transported points, which is inefficient for real-time animations as the number of point copies may still become very big. In order to alleviate this, we perform the following optimization: We create a matrix  $T'$ , which contains a 1 only at the maximum entry of each row of  $T$  at their respective position. All other entries are set to 0. We create another matrix  $T''$ , which is 1 only at the column maxima of  $T$  and 0 otherwise. We then generate a matrix  $\hat{T} = \frac{1}{2}(T' + T'')$  which is used for the transport instead of  $T$ .



**Figure 7:** Several stages of a growing maize plant, visualized with our method. Real measurements are highlighted with a grey background, while temporally upsampled data generated with our method are shown with a white background.



**Figure 8:** Several stages of a growing maize plant. Images with a grey background are input data. Images with a white background were generated with our method. Between the two successive scans, the plant has grown regarding its size as well as a newly evolved leaf. The budding process was generated automatically. In the upper row, the points were colorized according to their label. Note that the labels and their respective colours in the input data (grey background) do not match, because the input scans are segmented individually. Our matching method automatically produces correct assignments. The label colour of matched segments is interpolated over time (white background) in order to visualize this.

Having this transport plan, we can visualize the result as an animation, where each point  $p_i$  and its copies are transported linearly to the  $q_j$ 's positions in real time.

#### 4. Results

In this section, we show results obtained with our method. All experiments were performed on a PC with an Intel Core i7-8700K CPU and 64 GB RAM. We implemented our approach in Python. An overview of the computation times, separated into segmentation (where applicable) and segment matching and solving of the optimal transport problem is given in Table 1. The upsampling using

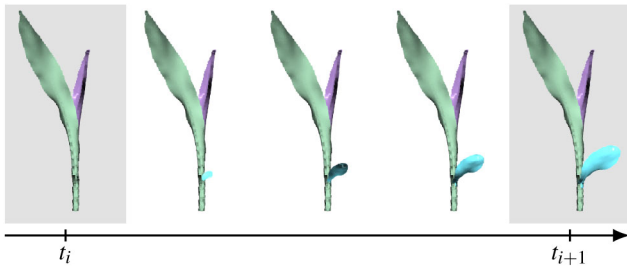
**Table 1:** Overview of computation times for the segmentation step and the combined matching and optimal transport step for different datasets. For datasets that are already segmented, we do not report timings for segmentation.

Dataset	Segmentation	Matching + optimal transport
Maize plant	12.1 s	50 s
Cropped maize	3 s	6 s
Synthetic Tree 1	4.2 s	42 s
Synthetic Tree 2	–	88 s
Komatsuna Plant 0	–	105 s
Komatsuna Plant 0/10	–	8 s

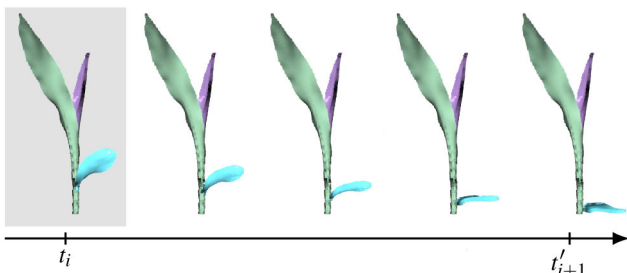
the optimal transport solution and the visualization run in real time, thus no computation times are given for this step.

**Maize plant time series.** We generated a smooth animation from eight scans of a maize plant. These were acquired with a hand-held laser scanner, in a fashion comparable to the one described by Paulus *et al.* [PSKL14]. Exemplary input scans and upsampled data are shown in Figure 7 and a full animation is shown in the accompanying video. Figure 8 shows two input scans from this dataset and their upsampling in more detail. The segmentation of these scans is shown by the colouring of the plant. The labels and their respective colours do not match in the input scans, because the point clouds were segmented individually. Our matching method automatically assigns the correct segments to each other. This is visualized by interpolating the segment colour in the upsampled data. The scans in this dataset have between 1700 and 5000 points, which took 12.1 s. The point cloud matching and computation of optimal transport took 50.0 s. The total computation time was 62.2 s. The visualization of the interpolation runs in real time.

We also computed an animation without computing a segmentation, using the described interpolation method on the whole point clouds, as shown in Figure 3. Here, parts of one leaf get transported to a different leaf, because it exhibits a stronger growth than before.



**Figure 9:** Temporal upsampling generated from the two input point clouds highlighted with a grey background. The inputs are identical, except that the light blue leaf was removed in  $t_i$ , simulating the growth of a new plant part. The new leaf grows smoothly.



**Figure 10:** Animation generated from the two input point clouds from Figure 9 in reverse order. The inputs are highlighted with a grey background, depicting the full plant as the source and the cropped plant as the target (automatically modified version shown here). The cropped leaf falls slowly to the ground.

Employing the segmentation prohibits this artefact because the transport is restricted to take place only between matched segments.

**Cropped maize.** In order to test the handling of cropped plant segments, we manually removed a leaf of the maize plant. As a first test, we used the cropped plant as the source and the complete plant as the target. The result is an animation of a growing leaf, shown in Figure 9. For the second test, we used the full plant as the source and the cropped plant as the target. Our method is able to generate an animation of the cropped leaf falling to the ground, shown in Figure 10 and the accompanying video. The full plant had 5080 points, the cropped one 4100. The total time required for segmentation was 6 s. The time for computing the upsampling was 3 s in both cases.

**Synthetic Tree 1 time series.** We generated a dataset of four virtual scans of a simulated growing tree with 778 to 2253 points. The segmentation took 4.2 s. Matching and optimal transport took 42.0 s. Our visualization runs in real-time. The dataset and its upsampling are shown in Figure 11 and the accompanying video.

**Synthetic Tree 2 dataset.** We generated a synthetic tree dataset consisting of two point clouds, representing tree states for different times with a considerable growth in-between including the growth of many new branches, where the first has 77,448 points and the sec-

ond one 326,032 points. A huge amount of growth with many new branches occurred. The segmentation was given due to the synthesis, as shown in Figure 4. We created a hierarchy on the segments as described in Section 3.2.1. The computation of the matching and optimal transport plan took 88 s. Our algorithm was able to generate a smooth and realistic animation, which is depicted in Figure 1 and shown in the accompanying video.

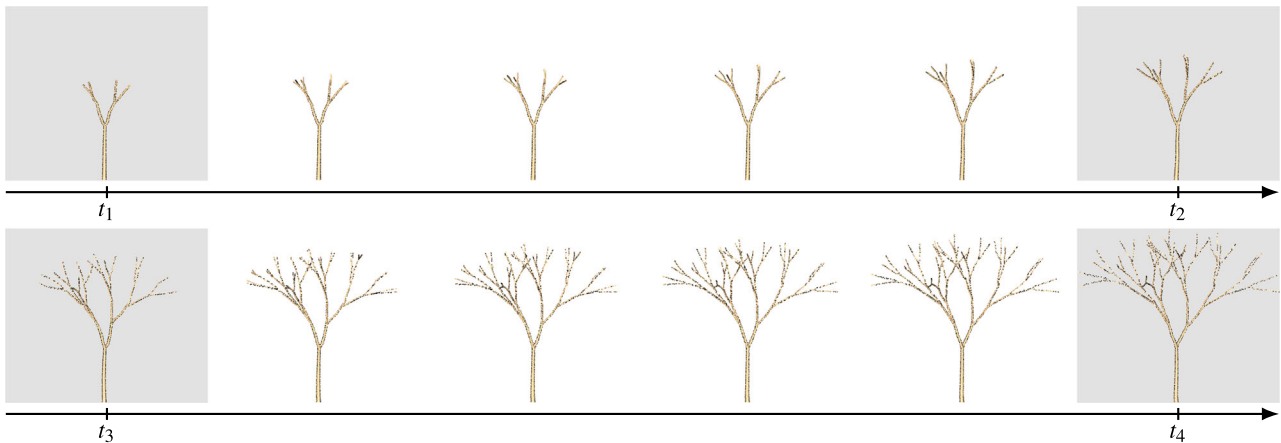
**Komatsuna dataset for instance segmentation, tracking and reconstruction.** The Komatsuna RGB-D dataset described by Uchiyama *et al.* [USM\*17] consists of a time series of RGB and depth images of five plants acquired with an RGB-D camera, as well as a manually created labelling of the plant parts. All plants were recorded six times a day over a period of 10 days. For our experiments, we used plant 0 and constructed coloured 3D point clouds from the RGB-D images for each plant part. Each label in the dataset defines a segment in our context and is interpolated separately. In order to obtain a hierarchical representation, we create an empty virtual root node and added child nodes representing the plant's leaves.

Due to noise in the original data, the depth-discontinuities at leaf boundaries are not precisely aligned with the RGB images and label data, and the depth of thin branches is not well-captured. Therefore, we apply a semi-automatic pre-processing step before creating the 3D point clouds: First, we remove all depth values which we consider to be incorrect in a manual process and mark them as 'unknown', which usually occurs at the boundaries of leaves. In a second step, we fill in the 'unknown' depth values based on the known valid depth values, similar to the approach of Desbrun *et al.* [DMSB99]. This is carried out for each label of each plant separately.

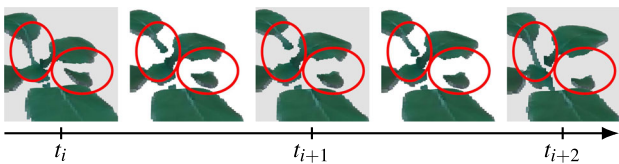
This dataset shows partial occlusions and in some cases slightly misplaced manual labels. In Figure 12 and in the accompanying video, it can be seen, that, while the general animation is smooth, these two aspects lead to some visual artefacts. The temporal occlusion leads to leaves seemingly shrinking away from the centre of the plant and growing back to the centre shortly later. The misplaced labelling leads to some points moving between leaves. Both are limitations of our method and may be addressed either in pre-processing or in future work. The computation time was 105 s.

As an additional experiment, we used only each 10-th scan of the Komatsuna plant 0 dataset. The resulting upsampling looks convincing and shows fewer visual artefacts since by skipping input data, we essentially removed high-frequency temporal noise as also demonstrated in the accompanying video. The computation time was 10 s.

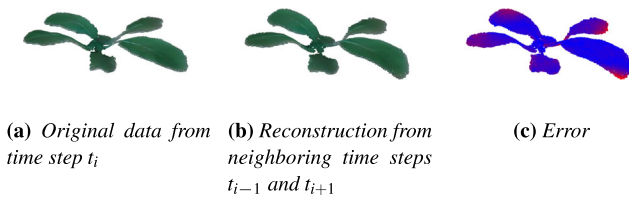
**Verification: Reconstruction of the Komatsuna dataset by skipping original data.** To analyse the quality of the upsampling, we omit every second frame of the original dataset and compare it with a reconstruction generated with our method. As an example, we show scan 3 of day 9 of plant 0 of the dataset. The whole dataset was re-scaled to fit into the  $[0, 1]^3$  unit volume. Comparing the reconstruction to the original data resulted in a mean distance of 0.001728 unit lengths, a standard deviation of 0.001130 unit lengths and a maximum distance of 0.00787517 unit lengths. A visual comparison is shown in Figure 13.



**Figure 11:** The Synthetic Tree 1 dataset consists of four time steps  $t_1$  to  $t_4$ . Input scans are highlighted with a grey background. The dataset was temporally upsampled with our method in the intervals  $[t_1, t_2]$  and  $[t_3, t_4]$ , respectively, shown with a white background.



**Figure 12:** Artefacts in the input data, like the temporarily severed stalk in the back and the partially occluded leaf on the right in this example are temporally upsampled as shrinking and growing animations. Input scans are highlighted with a grey background. We suggest a pre- or post-processing in order to mitigate these artefacts for future work.



**Figure 13:** Reconstruction of a point cloud of the Komatsuna dataset. Left: Original data of scan 3 of day 9 of plant 0. Centre: Reconstruction of this data, using scans 2 and 4 of day 9 of plant 0. Right: Colour-coded reconstruction error. Lower values are coloured in blue, higher values in red. The maximum error is 0.00787517 unit lengths, with input data scaled to the unit volume.

## 5. Limitations and Future Work

The segmentation algorithm described in Section 3.2.2 is designed for thin structures as occurring in the plant data considered, whereas the matching and optimal transport components of our method do not have such limitations. In the future, we intend to investigate the combination of our method with other segmentation techniques, such as PointNet [QSMG17]. Furthermore, our method is tailored to data that can be represented as a hierarchy of segments, and the

results of our method depend on the quality of the input data. If the input data seemingly move back and forward, as in the Komatsuna dataset, this is also mirrored in our upsampling. The plant movement can be due to the natural movement of the plant or due to scanner noise. Varying sampling in the input data can also lead to counter-intuitive movement of the individual points in the upsampled dataset in some cases. This can either be resolved by resampling the data in a pre-processing step or by introducing additional constraints to the optimal transport problem, which would be an interesting direction for future work. Noisy points are not smoothed out by the method itself. In this case, we rely on pre-processing. Furthermore, our method does not fill in temporarily occluded regions of the point cloud data. This can be resolved by preprocessing the data with existing methods like 3D shape inpainting.

Our method matches the segments traversing the hierarchy in a greedy fashion. While this strongly reduces computational complexity, it can also lead to errors that could be prevented by comparing all possible successor combinations. Segments that were pruned completely can be recognized and handled in a user-defined fashion. Partially pruned segments are currently not recognized and thus result in a shrinking animation. In the future, we would like to explore possibilities for handling these cases, for example, by comparing the volume and only allowing volume increases for the animation. Volume decreases could be either faded out or filled.

## 6. Conclusion

We presented a novel, efficient approach for the temporal up-sampling of point cloud sequences that, by design, circumvents limitations of previous approaches that rely on exact point correspondences, shape priors or the availability of huge datasets and allows handling topological changes to be expected for several growth processes. For this purpose, our approach involves an initial automatic alignment of successive scans, the generation of a segment hierarchy for each individual scan, the matching of the segment hierarchies obtained for successive scans and the final computation of segment-wise regularized optimal

transport in a memory-efficient way that allows morphing the distributions of the point sets associated with matched segments onto each other. As demonstrated by our evaluation, our approach allows the generation of a smooth temporal upsampling of the input point cloud sequence and, hence, a realistic growth animation. We believe that our method will be useful for biologists and researchers in the field of agriculture, as well as in the entertainment industry.

### Acknowledgements

We would like to thank the reviewers and the editor for providing valuable feedback and suggestions for improvements. We would like to thank Alexander Schier for providing the Synthetic Tree 1 dataset and Lasse Klingbeil for his helpful comments.

### References

- [ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 157–164.
- [ADT11] ALENYA G., DELLEN B., TORRAS C.: 3d modelling of leaves from color and ToF data for robotized plant measuring. In *2011 IEEE International Conference on Robotics and Automation* (Shanghai, China, May 2011), IEEE, pp. 3408–3414.
- [Ale03] ALEXA M.: Differential coordinates for local mesh morphing and deformation. *The Visual Computer* 19, 2 (May 2003), 105–114.
- [BGQ05] BAO Y., GUO X., QIN H.: Physically based morphing of point-sampled surfaces. *Computer Animation and Virtual Worlds* 16, 3-4 (July 2005), 509–518.
- [BIZ18] BERTHOLET P., ICHIM A., ZWICKER M.: Temporally consistent motion segmentation from RGB-D video. *Computer Graphics Forum* 37, 6 (2018), 118–134.
- [BM92] BESL P. J., MCKAY N. D.: Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures* (Apr. 1992), vol. 1611, International Society for Optics and Photonics, pp. 586–606.
- [BT18] BANSAL S., TATU A.: Lie bodies based 3d shape morphing and interpolation. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production* (New York, NY, USA, 2018), CVMP '18, ACM, pp. 5:1–5:10.
- [BVDPPH11] BONNEEL N., VAN DE PANNE M., PARIS S., HEIDRICH W.: Displacement interpolation using lagrangian mass transport. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–12.
- [BVGPO9] BARAN I., VLASIC D., GRINSPUN E., POPOVIĆ J.: Semantic deformation transfer. *ACM Transactions on Graphics (TOG)* 28, 3 (July 2009), 36:1–36:6.
- [CAJBS05] CHUAI-AREE S., JÄGER W., BOCK H. G., SIRIPANT S.: Simulation and visualization of plant growth using Lindenmayer systems. In *Modeling, Simulation and Optimization of Complex Processes* (2005), Springer, Berlin Heidelberg, pp. 115–126.
- [CFB16] CHEN X., FENG J., BECHMANN D.: Mesh sequence morphing. *Computer Graphics Forum* 35, 1 (Feb. 2016), 179–190.
- [CL09] CHU H.-K., LEE T.-Y.: Multiresolution mean shift clustering algorithm for shape interpolation. *IEEE Transactions on Visualization and Computer Graphics* 15, 5 (Sept. 2009), 853–866.
- [CPSV16] CHIZAT L., PEYRÉ G., SCHMITZER B., VIALARD F.-X.: Scaling algorithms for unbalanced transport problems. *arXiv preprint arXiv:1607.05816* (2016).
- [Cut13] CUTURI M.: Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, Burges C. J. C., Bottou L., Welling M., Ghahramani Z., Weinberger K. Q., (Eds.), vol. 26. Red Hook, New York, USA, Curran Associates, Inc., 2013, pp. 2292–2300.
- [DCSA\*14] DIGNE J., COHEN-STEINER D., ALLIEZ P., DEGOES F., DESBRUN M.: Feature-preserving surface reconstruction and simplification from defect-laden point sets. *Journal of Mathematical Imaging and Vision* 48, 2 (Feb. 2014), 369–382.
- [DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 317–324.
- [DNC\*18] DIGUMARTI S. T., NIETO J., CADENA C., SIEGWART R., BEARDSLEY P.: Automatic segmentation of tree structure from point cloud data. *IEEE Robotics and Automation Letters* 3, 4 (Oct. 2018), 3043–3050.
- [FB11] FRÖHLICH S., BOTSCH M.: Example-driven deformations based on discrete shells. *Computer Graphics Forum* 30, 8 (2011), 2246–2257.
- [FPPA14] FERRADANS S., PAPADAKIS N., PEYRÉ G., AUJOL J.-F.: Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences* 7, 3 (2014), 1853–1882.
- [FSDH14] FRIGO O., SABATER N., DEMOULIN V., HELLIER P.: Optimal transportation for example-guided color transfer. In *Computer Vision – ACCV 2014* (Cham, Germany, 2014), Springer International Publishing, pp. 655–670.
- [GCLX17] GAO L., CHEN S.-Y., LAI Y.-K., XIA S.: Data-driven shape interpolation and morphing editing. *Computer Graphics Forum* 36, 8 (2017), 19–31.
- [GDHB17] GELARD W., DEVY M., HERBULOT A., BURGER P.: Model-based segmentation of 3d point clouds for phenotyping sunflower plants. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (2017), SCITEPRESS - Science and Technology Publications, pp. 459–467.

- [GK17] GOLLA T., KLEIN R.: An efficient statistical data representation for real-time rendering of metallic effect car paints. In *Virtual Reality and Augmented Reality: 14th EuroVR International Conference, EuroVR 2017* (2017), Springer, Cham, pp. 51–68.
- [GK18] GOLLA T., KLEIN R.: Interactive interpolation of metallic effect car paints. In *Vision, Modeling & Visualization, VMV 2018* (2018), EG VMV '18, The Eurographics Association, p. 11–20.
- [GLHH13] GAO L., LAI Y.-K., HUANG Q.-X., HU S.-M.: A data-driven approach to realistic shape morphing. *Computer Graphics Forum* 32, 2pt4 (2013), 449–457.
- [HLW07] HU J., LIU L., WANG G.: Dual Laplacian morphing for triangular meshes. *Computer Animation and Virtual Worlds* 18, 4-5 (Sept. 2007), 271–277.
- [JBB13] JÓZSA O., BÓRCS A., BENEDEK C.: Towards 4D virtual city reconstruction from lidar point cloud sequences. In *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2013), vol. II-3/W1, pp. 15–20.
- [KKBS05] KURTH W., KNIEMEYER O., BUCK-SORLIN G.: Relational growth grammars: A graph rewriting approach to dynamical systems with a dynamical structure. In *Unconventional Programming Paradigms* (2005), Springer, Berlin, Heidelberg, pp. 56–72.
- [LCT\*18] LI D., CAO Y., TANG X.-S., YAN S., CAI X.: Leaf segmentation on dense plant point clouds with facet region growing. *Sensors* 18, 11 (Nov. 2018), 3625.
- [LFM\*13] LI Y., FAN X., MITRA N. J., CHAMOVITZ D., COHEN-OR D., CHEN B.: Analyzing growing plants from 4d point cloud data. *ACM Transactions on Graphics (TOG)* 32, 6 (Nov. 2013), 1–10.
- [LKM05] LINSEN L., KARIS B. J., MCPHERSON E. G., HAMANN B.: Tree growth visualization. *Journal of WSCG* (2005), 8.
- [LLV\*12] LI H., LUO L., VLASIC D., PEERS P., POPOVIĆ J., PAULY M., RUSINKIEWICZ S.: Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics (TOG)* 31, 1 (Feb. 2012), 2:1–2:11.
- [LPY\*12] LIU Y., PAN J., YANG L., ZHU X., ZHANG N.: Visualization of virtual plants growth based on open L-system. In *Computer and Computing Technologies in Agriculture V*, Li D., Chen Y., (Eds.), vol. 368. Springer, Berlin, Heidelberg, 2012, pp. 90–96.
- [LSLCO05] LIPMAN Y., SORKINE O., LEVIN D., COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics (TOG)* (2005), 479–487.
- [LXZ\*97] LOOMIS J. J., XIUWEN Liu, ZHAOHUA Ding, FUJIMURA K., EVANS M. L., ISHIKAWA H.: Visualization of plant growth. In *Proceedings. Visualization '97 (Cat. No. 97CB36155)* (Oct. 1997), IEEE, pp. 475–478.
- [MBW\*18] MORTENSEN A. K., BENDER A., WHELAN B., BARBOUR M. M., SUKKARIEH S., KARSTOFT H., GISLUM R.: Segmentation of lettuce in coloured 3d point clouds for fresh weight estimation. *Computers and Electronics in Agriculture* 154 (Nov. 2018), 373–381.
- [MHS\*19] MAKOWSKI M., HÄDRICH T., SCHEFFCZYK J., MICHELS D. L., PIRK S., PAŁUBICKI W.: Synthetic silviculture: multi-scale modeling of plant ecosystems. *ACM Transactions on Graphics (TOG)* 38, 4 (July 2019), 14.
- [MMT18] MÉRIGOT Q., MEYRON J., THIBERT B.: An algorithm for optimal transport between a simplex soup and a point cloud. *SIAM Journal on Imaging Sciences* 11, 2 (Jan. 2018), 1363–1389.
- [PDMK13] PAULUS S., DUPUIS J., MAHLEIN A.-K., KUHLMANN H.: Surface feature based classification of plant organs from 3d laser-scanned point clouds for plant phenotyping. *BMC Bioinformatics* 14, 1 (July 2013), 238.
- [PKD07] PITIÉ F., KOKARAM A. C., DAHYOT R.: Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding* 107, 1 (2007), 123–137.
- [PSKL14] PAULUS S., SCHUMANN H., KUHLMANN H., LÉON J.: High-precision laser scanning system for capturing 3d plant architecture and analysing growth of cereal plants. *Biosystems Engineering* 121 (May 2014), 1–11.
- [QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), IEEE, pp. 652–660.
- [RFP14] RABIN J., FERRADANS S., PAPADAKIS N.: Adaptive color transfer with relaxed optimal transport. In *2014 IEEE International Conference on Image Processing (ICIP)* (2014), IEEE, pp. 4852–4856.
- [RP11] RABIN J., PEYRÉ G.: Wasserstein regularization of imaging problem. In *2011 18th IEEE International Conference on Image Processing* (2011), IEEE, pp. 1541–1544.
- [RPDB11] RABIN J., PEYRÉ G., DELON J., BERNOT M.: Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision* (2011), Springer, Berlin, Heidelberg, pp. 435–446.
- [RTG98] RUBNER Y., TOMASI C., GUIBAS L. J.: A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision* (1998), IEEE, pp. 59–66.
- [RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40, 2 (2000), 99–121.
- [SBM\*10] STAVA O., BENEŠ B., MĚCH R., ALIAGA D. G., KRIŠTOF P.: Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum* 29, 2 (May 2010), 665–674.

- [SCOL\*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (New York, NY, USA, 2004), SGP '04, ACM, pp. 175–184.
- [SDGP\*15] SOLOMON J., DE GOES F., PEYRÉ G., CUTURI M., BUTSCHER A., NGUYEN A., DU T., GUIBAS L.: Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 66.
- [SPK\*14] STAVA O., PIRK S., KRATT J., CHEN B., MĚCH R., DEUSSEN O., BENES B.: Inverse procedural modelling of trees. *Computer Graphics Forum* 33, 6 (Sept. 2014), 118–131.
- [SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 488–495.
- [SZP07] SOLENTHALER B., ZHANG Y., PAJAROLA R.: Efficient refinement of dynamic point data. In *Eurographics Symposium on Point-Based Graphics* (2007), Botsch M., Pajarola R., Chen B., Zwicker M., (Eds.), The Eurographics Association.
- [THCF06] TIAN H., HE Y., CAI H., FENG L.: Efficient metamorphosis of point-sampled geometry. In *16th International Conference on Artificial Reality and Telexistence—Workshops (ICAT'06)* (Hangzhou, Zhejiang, China, 2006), IEEE, pp. 260–263.
- [TZZ09] TAN G., ZHANG S., ZHANG Y.: Shape morphing for point set surface based on vertex deformation gradient. In *2009 WRI World Congress on Software Engineering* (May 2009), vol. 2, IEEE, pp. 466–471.
- [USM\*17] UCHIYAMA H., SAKURAI S., MISHIMA M., ARITA D., OKAYASU T., SHIMADA A., TANIGUCHI R.-I.: An easy-to-setup 3d phenotyping platform for KOMATSUNA dataset. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (Venice, Italy, Oct. 2017), IEEE, pp. 2038–2045.
- [VLSM18] VLACHOS E., LALOS A. S., SPATHIS-PAPADIOTIS A., MOUSTAKAS K.: Distributed consolidation of highly incomplete dynamic point clouds based on rank minimization. *IEEE Transactions on Multimedia* 20, 12 (Dec 2018), 3276–3288.
- [VTSSH15] VON-TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: Real-time nonlinear shape interpolation. *ACM Transactions on Graphics (TOG)* 34, 3 (May 2015), 34:1–34:10.
- [WLX\*18] WANG G., LAGA H., XIE N., JIA J., TABIA H.: The shape space of 3d botanical tree models. *ACM Transactions on Graphics (TOG)* 37, 1 (Jan. 2018), 1–18.
- [WPKM15] WAHABZADA M., PAULUS S., KERSTING K., MAHLEIN A.-K.: Automated interpretation of 3d laserscanned point clouds for plant organ segmentation. *BMC Bioinformatics* 16, 1 (Dec. 2015). 1–11.
- [WWX\*19] WU S., WEN W., XIAO B., GUO X., DU J., WANG C., WANG Y.: An accurate skeleton extraction approach from 3d point clouds of maize plants. *Frontiers in Plant Science* 10 (2019). 1–14.
- [WZH12] WANG R., ZHANG C., HU J.: Smooth morphing of point-sampled geometry. In *Computer Applications for Graphics, Grid Computing, and Industrial Environment* (2012), Kim T.-h., Cho H.-s., Gervasi O., Yau S. S., (Eds.), Communications in Computer and Information Science, Springer, Berlin, Heidelberg, pp. 16–23.
- [XM15] XU L., MOULD D.: Procedural tree modeling with guiding vectors. *Computer Graphics Forum* 34, 7 (Oct. 2015), 47–56.
- [XSWL15] XU W., SALZMANN M., WANG Y., LIU Y.: Deformable 3D fusion: From partial dynamic 3D observations to complete 4D models. In *2015 IEEE International Conference on Computer Vision (ICCV)* (Dec 2015), IEEE, pp. 2183–2191.
- [XZPF04] XIAO C., ZHENG W., PENG Q., FORREST A. R.: Robust morphing of point-sampled geometry. *Computer Animation and Virtual Worlds* 15, 3-4 (July 2004), 201–210.
- [XZWB06] XU D., ZHANG H., WANG Q., BAO H.: Poisson shape interpolation. *Graphical Models* 68, 3 (2006), 268–281.
- [YLX\*16] YUAN Q., LI G., XU K., CHEN X., HUANG H.: Space-time co-segmentation of articulated point cloud sequences. *Computer Graphics Forum* 35, 2 (2016), 419–429.
- [ZB13] ZHAO Y., BARBIČ J.: Interactive authoring of simulation-ready plants. *ACM Transactions on Graphics (TOG)* 32, 4 (July 2013), 84:1–84:12.
- [ZFG\*17] ZHENG Q., FAN X., GONG M., SHARF A., DEUSSEN O., HUANG H.: 4D reconstruction of blooming flowers. *Computer Graphics Forum* 36, 6 (2017), 405–417.
- [ZLLZ12] ZENG Q., LIN X., LIU H., ZHU T.: Plant's 3d modeling and growth simulation system based on skeleton extraction. *Journal of Information & Computational Science* 9 (May 2012), 1357–1363.

### Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

### Data video S1