The Visual Computer (2022) 38:3327–3339
https://doi.org/10.1007/s00371-022-02526-x

ORIGINAL ARTICLE

Urban tree generator: spatio-temporal and generative deep learning for urban tree localization and modeling

Adnan Firoze[1] · **Bedrich Benes**[1] · **Daniel Aliaga**[1]

Accepted: 10 May 2022 / Published online: 21 June 2022

Abstract

We present a vision-based algorithm that uses spatio-temporal satellite imagery, pattern recognition, procedural modeling, and deep learning to perform tree localization in urban settings. Our method resolves two primary challenges. First, automated city-scale tree localization at high accuracy typically requires significant acquisition/user intervention. Second, vegetation-index segmentation methods from satellites require manual thresholding, which varies across geographic areas, and are not robust across cities with varying terrain, geometry, altitude, and canopy. In our work, we compensate for the lack of visual detail by using satellite snapshots across twelve months and segment cities into various vegetation clusters. Then, we use multiple GAN-based networks to plant trees by recognizing placement patterns inside segmented regions procedurally. We present comprehensive experiments over four cities (Chicago, Austin, Indianapolis, Lagos), achieving tree count accuracies of 87–97%. Finally, we show that the knowledge accumulated from each model (trained on a particular city) can be transferred to a different city.

Keywords Tree location · Procedural generation · Shape and surface modeling · Shape analysis and image retrieval · Urban tree

1 Introduction

At present, urban greening has emerged to be one of the most critical objectives as a means to human sustainability. It has been reported that while efforts are being taken, there is a dire need of accurate data for proper management of such endeavors—that have shown to have saved over trillions of dollars in air pollution and carbon removal [58]. However, the spending has also been an average of over $10 billion in the USA (per city) [38]. In this work, we aim to bolster such efforts through localizing urban tree locations, even ones that are not government-owned through deep learning and computer vision approaches.

Recently, 3D urban modeling has received significant attention. One included task is determining the location of trees in urban environments. Tree modeling and localization have been pursued in various ways. Tree and vegetation modeling (e.g., [4,13,30]) renders/creates 3D models. Segmentation algorithms have been developed to isolate broad tree/canopy areas in captured overhead imagery (e.g., LiDAR, satellite, or aerial) [31]. USDA's i-Tree software toolkit [53] is a crowd-sourced method to report on trees. While precise, this approach does not scale, cannot be readily updated, and depends on the reliable participation of human workers. The recent proliferation of deep learning has introduced promising new methods (e.g., [3,46]). But due to occlusions and limited resolution, these methods cannot distinguish individual trees, do not estimate tree counts, and have accuracies only in the 60–80% range in the aforementioned literature.

Our tree modeling and localization work exploits two key observations. First, satellite imagery's frequent capture rate (e.g., weekly or daily) enables capturing the spatio-temporal vegetation footprint during a season or year, thus providing richer information than a single image. Second, vegetation in cities succumbs to urban management rules that regulate their development. Since individual trees cannot be readily discerned from a satellite due to occlusion and resolution limitations, we instead exploit our observations to enable a self-supervised generative (or procedural) approach to tree inventory modeling and cover estimation. To verify

✉ Adnan Firoze
afiroze@purdue.edu

1 Department of Computer Science, Purdue University, 305 N. University St, West Lafayette, IN 47907, USA

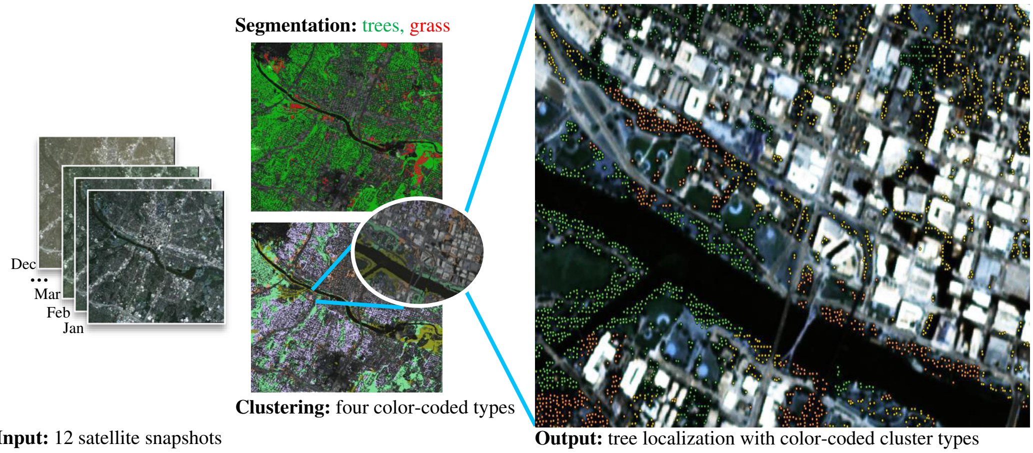


Fig. 1 Urban tree localization automatically infers tree counts and positions from spatio-temporal satellite images and procedural urban vegetation rule-sets using a generative algorithm

the correctness and robustness of our approach, we have used multiple ground truth datasets including human and government surveyed/vetted data [8,40], INRIA [32] datasets, and Google Earth [20] data.

Our approach exploits the multiple image-based and procedural-based rules for planting. It consists of preprocessing and runtime steps. The preprocessing trains an initial deep segmentation network on 12-month images. Then, using a three-tier set of urban vegetation management rules and our procedural modeling system, it trains generative networks for four different urban space configurations (residential, industrial, roadside, and park). Given 12 monthly satellite snapshots (i.e., PlanetScope Daily Imagery at three meters per pixel, or 3 *mpp* [41]), the runtime first performs an initial segmentation and clustering into the four mentioned types. Then, the generative modeling engine produces a tree distribution map for each cluster. Finally, from the map tree coverage, locations, and counts are obtained (Fig. 1).

We evaluated our method on four diverse cities: Chicago, Austin, Indianapolis—USA, and Lagos—Nigeria (spanning 84–225 km^2 and containing 17,652–144,788 trees). Our tree coverage and count calculations occur in seconds. We compare to ground truth (GT) tree counts and obtain an accuracy of 87–97%. We also compare our coverage estimation to other more costly methodologies, including ground-based crowd-sourced individual tree data and deep learning-based approaches, obtaining similar or better results but in only a fraction of the time and cost. We claim the following contributions: (1) segmentation of urban spatio-temporal satellite imagery into tree coverage, grass, and other areas; (2) clustering vegetation canopy into urban configurations (e.g., residential, industrial, roadside, parks/forestry); (3) estimation of tree locations to simulate proper tree count and placement; (4) creation of ground-truth datasets of approximately 10–20% of each city that identifies tree cover, counts, and placements that are released to others for further studies (see Sect. 3.1).

2 Related work

Procedural Urban Tree Generation: Urban procedural modeling has had much success in modeling and reconstruction [33,35]. Procedural modeling of vegetation has a long history [44]. While realistic modeling of vegetation is important in weather simulations and urban ecology modeling (e.g., [2,4]), most simulated city models use vegetation for aesthetic purposes and interactive simulations [24]. Recent works attempt to procedurally reconstruct trees by using deep learning [27,30], but do not focus on tree localization. Several works focus on using point-cloud data (e.g., [17,49]), and they focus primarily on ground-level data and small regions. In contrast, in our work, we use procedural modeling in determining tree localization (e.g., coverage and count) in real-world settings that scales to large areas or entire cities.

Our work most closely relates to [51] and [36]. Niese et al. [36] used high-resolution aerial and satellite imagery to generate tree coverage maps and used procedural rules to plant trees in urban configurations, using NYC Open Data [10] at 0.3 *mpp*. This work focused on photorealism from various viewing angles in NYC; tree count and land cover correctness were not addressed. Yao et al. [51] used high-resolution satellite imagery and several deep networks (AlexNet [25], U-Net [45], and VGG-Net [48]) to output tree counts using density regression, but they do not output tree locations. Moreover, [51] uses 0.8 *mpp* data on several provinces in China. Our method outperforms their average count accuracies of 62–83%. Moreover, we also pursue outputting tree locations, which is not performed by prior work.

Vegetation Segmentation: Segmenting land cover into classes has received significant traction [31]. Deep learning

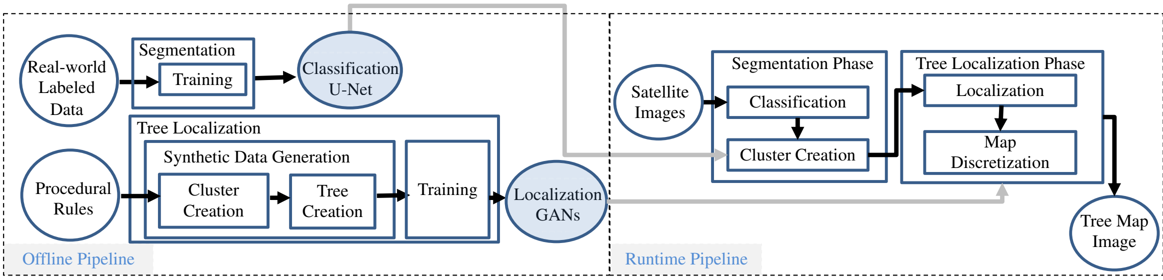


Fig. 2 *Workflow.* Rectangles are processes, clear ovals are data, and shaded ovals are deep learning networks

has introduced many new approaches using a variety of networks. For example, Arief et al. [3] compared different deep learning networks to classify land into eight classes with a validation accuracy of 66.67% using high-resolution LiDAR or aerial data. Lee et al. defined SegNet [26] as a method for segmentation of land using an encoder–decoder method, achieving accuracies of 85% from unmanned aerial vehicle (UAV)-captured images (i.e., 0.5 *mpp*). Field-obtained data acquisition as discussed in [15] is done manually in dense forests, which is both costly and time-consuming. However, the data collection (e.g., Field, LiDAR, or UAV) is difficult to scale to an entire city or region, and obtaining repeated acquisitions is costly. Moreover, the methods have not focused on the urban tree localization task.

Global-scale acquisition efforts such as ICESat-2 [11], GEDI dataset [42], or the JAXA dataset [21] do not obtain data at sufficient resolution. For example, ICESat-2 captures height along sparse, thin bands, and GEDI's and JAXA's resolution is about 30 *mpp*. These resolutions are too coarse for us. We focus on urban extents that are not well captured by these acquisition efforts.

Geographic information systems (GIS) have also used vegetation indices (e.g., NDVI [18,56]). These indices give a vegetation probability value. However, one major drawback of NDVI is finding a parameter set that works universally. Thus, traditional NDVI lacks robustness and needs experimentally determined inter- and intra-city customization. Jiang et al. [22] analyzed this technique and its drawbacks in detail.

3 Spatio-temporal segmentation

The first phase of our pipeline (Fig. 2) includes a spatio-temporal vegetation cover classification of satellite images which partitions a city into tree, grass, and background classes, followed by a cluster creation process.

3.1 Spatio-temporal data

One of the novel features of our work is using spatio-temporal satellite data for segmentation and localization. As

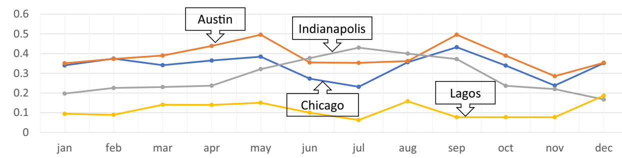


Fig. 3 *NDVI Fluctuations.* Mean NDVI over 12 months for our four test

also shown in Shen et al. [47], NDVI maps of cities change in shape, color, and surface reflectance over time (Fig. 3). Thus, instead of having only one snapshot, we use a monthly snapshot of a city over 12 months to capture the spatial and temporally varying features. In particular, our approach uses PlanetScope's 3 *mpp* and four-channel data (red, green, blue, near-infrared) with cloud coverage filter set to under 5% [41]. The per-city satellite images are vertically stacked to create 48-dimensional tensors (4 channels $\times$12 snapshots). Moreover, we join relevant tiles to capture the extent of four test cities (Chicago: 10$\times$10 km or 72.4% of total extent, Austin: 15$\times$15 km or 91.2%, Indianapolis: 12$\times$7 km or 94.05%, and Lagos: 7.7$\times$5.9 km or 82.41%). For experimental comparisons, we used 12-month data from 2020 aligning with the canopy data from Google Earth [20] of the same period (for Lagos and Indianapolis), alongside ground-based manually collected and well-vetted government released tree locations from Austin, TX [8]. We assume based on [7,8,40] that the number and location of trees remain approximately the same in the span of 12 months of a given year. Thus, using this GT data can accurately gauge the performance of our approach. Our annotated dataset and code are available at https://github.com/adnan0819/Urban-Tree-Generator/.

3.2 Classification

Our vegetation classifier is based on a U-Net [45], and it classifies any city into tree, grass, and background. Our output provides the same dimensions in width and height but with n channels where n is the number of classes in the segmentation (in our case, $n = 3$). The size of the tiles is 256^2 pixels. The input dimension of our data (per tile) is $256^2 \times 48$ and the output is $256^2 \times 3$. The tiles are stitched to curate the

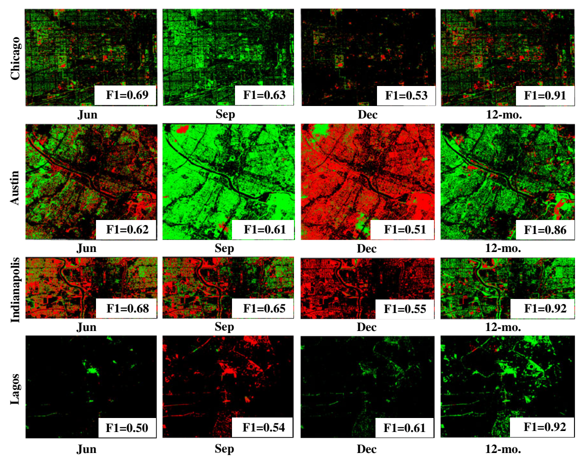


Fig. 4 *Single- vs 12-month segmentation.* Segmentation of four cities into trees (green), grass (red), and background (black) using single month vs. 12-month data. F1-scores are shown, indicating a clear superior accuracy of our 12-month solution

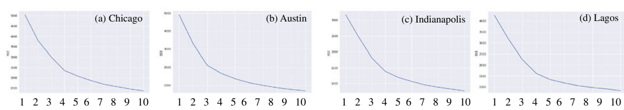


Fig. 5 *Selection of k in k-means clustering.* Calibration of the optimal number of k in k-means clustering

full maps. We developed a novel data generator for U-Net and performed data augmentation specific to our 48 channel data. Figure 4 shows that using 12-month data outperforms a single-month data for all our test cities.

3.3 Cluster creation

After the spatio-temporal classification, the input data are clustered into various urban configurations that are representative of different tree placement strategies. We created a clustering engine using k-means feature clustering, varied the values of k from 2 to 8, and computed the sum of squared errors (SSE). Using the elbow method, we found the optimal value for k, across our multiple test cities, to be $k = 4$. Heuristics and subjective observation were used to label the clusters as residential, roadside, industrial, and park areas. The output of the four types can be seen in color-coded Fig. 8. The optimal cluster number, $k = 4$, was chosen using the elbow method upon plotting number of clusters versus SSE in Fig. 5.

3.4 Training

We trained our spatio-temporal segmentation approach with two variants: *pre-tuned* and *fine-tuned*. The pre-tuned variant uses the data accumulation from various cities to create one system so that deployment to a new city requires only the 12-snapshots of satellite imagery at 3 mpp. The fine-tuned variant requires additional local data. Our analysis finds that the pre-tuned system has slightly lower performance but fewer data requirements.

Our system needs two additional city-specific datasets to perform fine-tuning for a city. First, about 10–20% of the city should be labeled into three classes (trees, grass, and background) to train a local segmentation engine. It took one person approximately 8–16 hours to perform this labeling for each test city (that we have made available for everyone for further research). Second, the fine-tuned clustering engine needs building footprints and road networks sourced, for example, from OpenStreetMap [39], and is used to improve the accuracy of clustering into various urban configurations. Since the GT and resolution of the building, and street locations were known, we could accurately extract the distances between reference locations and annotated trees. Section 5 discusses the additional, though not very large, accuracy gains from fine-tuning.

4 Tree localization

We perform tree localization using deep networks trained with parameterized urban procedural rules in the second runtime phase. We train one conditional GAN-based network for each of residential, roadside, industrial, and park cluster types. For training, we generate a large number of synthetic 80 m × 80 m tiles mimicking the typical spatial patterns of each of the four types. The output from the segmentation phase is used as input to the aforementioned localization GANs. The outputs of the GANs are then discretized, yielding individual tree locations.

4.1 Procedural rules

To train the cGANs, we generate tiles of a synthetic city that exhibit procedurally defined parameterized tree planting rules. We define a set of four parameterized rules U_i:

- U_1 No overlap: tree center points should not overlap with buildings, roads, and other trees.
- U_2 Minimum tree-to-tree distance: is a minimum distance between tree center points. It is heuristically determined to be half of the field of the neighborhood (FON) [43] of trees.
- U_3 Minimum tree-to-building distance: is a minimum distance between a tree center point and a building.
- U_4 Minimum tree-to-road distance: is a minimum distance between a tree center point and a road surface.

Subsequently, by varying the parameter values and their spatial coverage, multiple instances of the rules are defined and

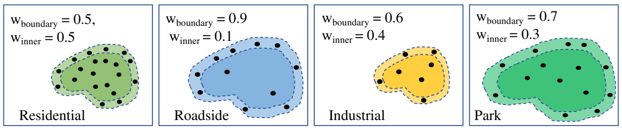


Fig. 6 *Example tree distribution for Chicago rule-sets*. Illustrating different distributions in rule-sets S_{rs_C}, S_{rd_C}, S_{pr_C}, and S_{ind_C} inside a 80 m × 80 m tile (T_j) represented as rectangles. The colored blobs are $B(T_j)$ inside the tile and black circles are tree locations

Table 1 Cluster rules V

Parameter	V_{rs}	V_{rd}	V_{ind}	V_{pr}
d_{bldg}	2 m	2 m	3 m	4 m
d_{street}	1 m	1 m	1 m	1 m
$w_{boundary}$	0.5	0.9	0.6	0.7
w_{inner}	0.5	0.1	0.4	0.3

For each cluster type, we show the rule parameter values

placed into three groups: *universal*, *cluster-specific*, and *city-specific*. When we lack the ground truth data for estimating cluster or city-specific parameter values, we use the average parameter values of clusters in other cities, as shown for Lagos.

We introduce some notations for clarity and brevity throughout the remainder of the paper. We abbreviate Chicago, Austin, Indianapolis, Lagos, and pre-tuned variant as C, A, I, L, P, respectively. Then we use rs, rd, pr, and ind to represent residential, roadside, park, and industrial, respectively. The minimum distances of a tree from the nearest building and street are denoted by d_{bldg} and d_{street}, respectively. A fixed-sized tile on the map (80 m × 80 m) is denoted by T_j where j is an index. $B(T_j)$ is the percentage of tree area (blob) in T_j, and n_j is the number of trees in the same tile. Among n_j trees, $w_{boundary}$ is the percentage of trees along one FON distance around $B(T_j)$, and w_{inner} refers to the remainder percentage of the trees inside the same tile (Fig. 6). Finally, we use U, V_c, and W_{c_x} to represent universal, cluster-type (in cluster c), and city-specific rules (in cluster c and city x), respectively.

Universal and Cluster-type Rules: The universal rules U are described at the beginning of this section. Cluster-type V_c rules are derived from city planning/municipal documents such as [7,9,52] for Chicago, Austin, and Indianapolis, respectively. All such codes stem from ANSI A300 Standards for tree management [50] for all municipal codes in the USA. Thus, the values for d_{bldg} and d_{street} were extracted from those standards. To verify their validity, we used a hand-labeled subset of tree locations for the three cities. The mean error of the values from the labeled data was $\leq 1\%$ from the city-planning standards. Since we have no such documentation for Lagos, we verified that the labeled Lagos data were within 3% from the values used for the other cities. Therefore, we adopted d_{bldg} and d_{street} from municipal standards as cluster-specific.

We note that $w_{boundary}$ and w_{inner} were chosen heuristically by overlaying precisely labeled tree locations on top of the output tree segments from our spatio-temporal segmentation phase. Upon deriving the statistics over all labeled data, the values of $w_{boundary}$ and w_{inner} were set for each cluster type. Further, we observed the average FON to be 4 ± 0.37 m in all test cities from annotation. Thus, we chose FON = 4 m that is in line with urban forestry literature [43].

City-specific Rules: For city-specific distribution rules W_{c_x}, a similar approach to deriving $w_{boundary}$ and w_{inner} was used with labeled ground truth data overlaid on tree coverage segments. We statistically derived the values based on the density and counts of the tree locations inside the segments.

Finally, for the complete system, the goal is to generate tree locations following the conjunction of all the procedural rules for a given city $x \in \{C, A, I, L, P\}$. Thus, we optimize and calibrate for rule sets for all values of $c \in \{rs, rd, pr, ind\}$:

$$S_{c\_x} = U \cap V_c \cap W_{c\_x}. \tag{1}$$

4.2 Synthetic data generation

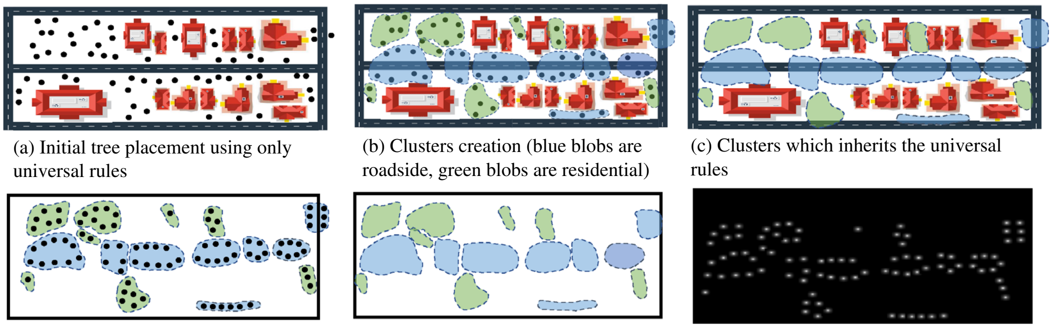
We use synthetic data to train one cGAN for each cluster type (residential, roadside, industrial, and park). Based on preliminary experiments, we found using at least 100,000 training images per GAN resulted in good learning results (Fig. 7).

Cluster Creation: We first define an initial temporary set of potential tree locations and then group the trees into clusters of different types. First, trees are placed by using a Poisson distribution which has been shown to be a good distribution model for trees in prior work (Keren [23]). Second, we use DBSCAN clustering [14] to generate a set of clusters spanning the temporary trees (Fig. 7b). The members of a cluster M are trees x and y:

$$M(x, y) : d(x, y) \leq \epsilon_c, \tag{2}$$

where recall $c \in \{rs, rd, pr, ind\}$ and $d(x, y)$ is the straight-line distance between x and y, and ϵ_c is the distance threshold for each cluster type.

Tree Placement: To produce a set of trees in each cluster that follow the rules and desired density, we perform the following four steps that over-seed a cluster and iteratively calibrate the cluster to behave as desired i.e., follow the characteristics determined by the procedural rules.



(a) Initial tree placement using only universal rules

(b) Clusters creation (blue blobs are roadside, green blobs are residential)

(c) Clusters which inherits the universal rules

(d) Synthetically placed trees using all procedural rules (discriminative)

(e) Training sample of isolated tree segments as input to cGANs

(f) Training sample of Gaussian tree heatmaps as target of cGANs (generative)

Fig. 7 *Synthetic tree generation workflow.* Synthetic data generation to train planting and localization networks. The rationale for choosing a generative model over a discriminative approach is given in Sect. 4.4

Table 2 *City-specific rules W*

Parameter	W_{rs_C}	W_{rd_C}	W_{pr_C}	W_{ind_C}
Mean $B(T_i)$	14.91%	20.08%	34.18%	7.52%
Mean n_i	9	12	14	7
Rule id	W_{rs_I}	W_{rd_I}	W_{pr_I}	W_{ind_I}
Mean $B(T_i)$	26.46%	32.86%	55.57%	5.73%
Mean n_i	18	11	18	4
Rule id	W_{rs_A}	W_{rd_A}	W_{pr_A}	W_{ind_A}
Mean B(T_i)	39.02%	34.53%	61.17%	12.44%
Mean n_i	15	14	25	4
Rule id	W_{rs_L}	W_{rd_L}	W_{pr_L}	W_{ind_L}
Mean B(T_i)	16.82%	22.37%	44.17%	6.48%
Mean n_i	10	15	17	5
Rule id	W_{rs_P}	W_{rd_P}	W_{pr_P}	W_{ind_P}
Mean B(T_i)	24.30%	27.46%	48.77%	8.04%
Mean n_i	13	13	19	5

For each city (C, A, I, L) and for the pre-tuned variant (P), we show the parameter values for the distribution rules: mean percentage of tree coverage in a tile, and mean number of trees inside the same tile

(1) *Randomized placement:* First, we place trees inside the clusters in a random fashion enforcing only the universal rules. Contrary to the Poisson disc sampling, we do not enforce any distance such that we naturally get an overestimation of trees inside clusters of every configuration.

(2) *Rule enforcement:* For each iteration, until we find density and count close to GT, we remove trees that violate our procedural rules. We incorporate our procedural rules, i.e., the cluster-type rules and city-specific rules (numeric parameters of both are reported in Tables 1 and 2), to place trees only inside the clusters as derived in the rules by Eq. 1.

(3) *Density calibration:* We check the density $B(T_j)$ and n_j for each cluster. If it is suboptimal (i.e., it has a significant difference from ground truth), we adjust ϵ_c which affects the size of clusters in a fixed size tile—$B(T_j)$ and the number of trees in that cluster n_j, go back to step "(1) Randomized placement," and repeat. We continue until we cannot improve upon our tree segment percentage per tile $B(T_j)$ and the corresponding tree count n_j relative to ground truth. Once we reach peak accuracy for every cluster, we proceed to the next step. We observed that there is not a one-to-one relationship in the input and output densities of the translation networks. Therefore, we calibrated the tree segment (blob) percentages in fixed tiles $B(T_j)$ and their associated tree counts n_j. We tested numerous generative cGAN models with different values in realistic ranges of $B(T_j)$ and n_j to find the densities and coverage percentages that resulted in the highest tree location and count accuracy. Figure 9 shows the calibration plots that visualize the decision of tree densities in synthetic data for Chicago.

(4) *Heatmap creation:* When the rules and densities have been calibrated for locally optimal output, we rasterize our tree points to 2D Gaussian discs forming a heatmap which facilitates evaluation of similarity. At this point, it is feasible to generate our training and target data for our cGAN networks. As such, we use the class-encoded tree coverage segments (Fig. 7e) as our training images and generate the aforementioned heatmaps from the tree locations (Fig. 7f). We repeat this process 10× for each city, resulting in approximately 100,000 tiles per urban configuration cluster type per city. In the heatmaps, the center of a Gaussian represents the highest probability of the presence of a tree which decays exponentially away from the center of tree location: $f_i(x) = e^{-\lambda \cdot x}$, where x is the distance from a point on the map where a tree i was seeded using the synthetic data generator, and λ is the decay rate.

Treatment of pre-tuned vs. fine-tuned engines: Although the fundamental approaches for the generation of the synthetic data remain the same for both our pre-tuned P and fine-tuned engines $\{C, A, I, L\}$, we note that for the pre-tuned engine, we only have ground truth count information n_j for

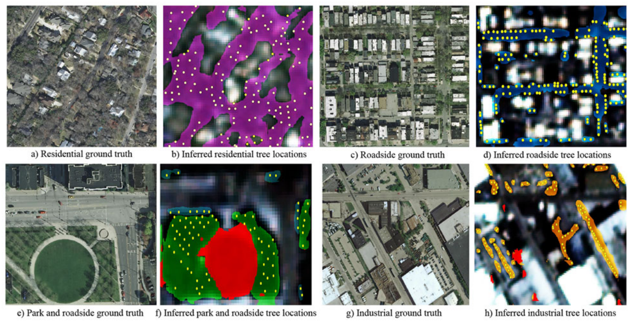


Fig. 8 *Qualitative results.* Real-world ground truth (from 0.3 *mpp* INRIA dataset for GT visualization) in (a, c, e, g) vs. trees located by our system (b, d, f, h). Here, purple, blue, green, brown, and red blobs refer to residential, roadside, park, industrial, and grass coverage. Yellow filled circles are inferred tree locations

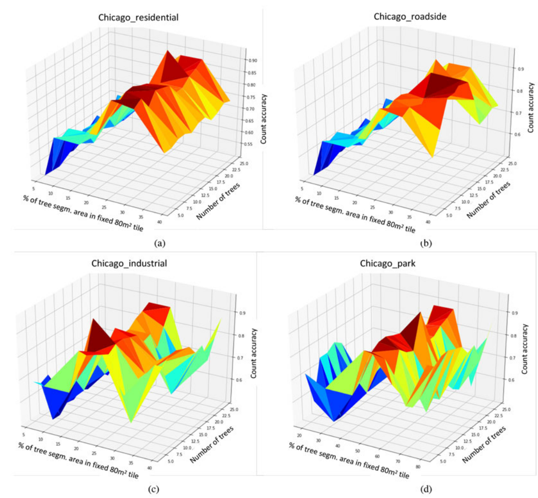


Fig. 9 *Synthetic data calibration.* Calibration of the optimal number of trees inside coverage percentage in fixed size of 80 m × 80 m tile to achieve highest count accuracy with respect to ground truth. Surface plots are shown for rule-sets in Chicago

the four cities that we have tested: Chicago, Austin, Indianapolis, and Lagos. Therefore, in the calibration phase, we accommodate the calibration of those cities to achieve the highest accuracy in terms of densities. However, we use the mean optimal $B(T_j)$ and mean optimal n_j of the known cities for an unknown city with no labeled data. Owing to the standardization of the city planning rules described previously, we showed that this generalization affects the performance marginally compared to the fine-tuned engines in Sect. 5. When performing fine-tuning in k-means clustering, additional features are included to account for building area and road network area, both sourced from OpenStreetMap [39].

4.3 Calibration and parameters of training data

We calibrated blob percentages in fixed tiles $B(T_j)$ and their associated tree counts n_j. We tested numerous generative models with different values in realistic ranges of $B(T_j)$ and n_j to find the densities and coverage percentages that resulted in the highest accuracy of tree location and count and selected the ones producing peak performance. The calibration plots that visualize the decision of this step in our synthetic data for Chicago are shown in Fig. 9.

The following section presents the derived values of all the parameters of cluster-specific and city-specific rules, as discussed. The sources and derivations are noted in Sect. 4.1. Table 1 reports the parameter values pertaining to the cluster-specific rules, whereas Table 2 reports the city-specific rules' parameter values.

We calibrated blob percentages in fixed tiles $B(T_j)$ and their associated tree counts n_j. We tested numerous generative models with different values in realistic ranges of $B(T_j)$ and n_j to find the densities and coverage percentages that resulted in the highest accuracy of tree location and count and selected the ones producing peak performance. The calibration plots that visualize the decision of this step in our synthetic data for Chicago are shown in Fig. 9.

A proper loss function selection was imperative for the success of the networks. As noted in the *4) Heatmap Generation* step, we used $\lambda = 0.25$ as the Gaussian decay rate, and the rationale and experiment are detailed in Sect. 4.4. The selection of Multi-scale SSIM-based loss function to be used as our generator's loss function is also discussed and quantified with experiments presented in Sect. 4.4.

4.4 Training, loss function, and evaluation metric

The objective of the training phase for tree location estimation is to use synthetically generated coverage segments as inputs to the networks and output realistic (spatially and count-wise) trees as Gaussian heatmaps that are later discretized to points. The tree location estimation is achieved by using multiple cGAN models [19] for translating tree coverage segments generated by our segmentation phase and further classified into classes, to Gaussian heatmaps of tree locations (Fig. 7f). Several illustrations of our final tree location extractions (beside corresponding ground truth) are depicted in Fig. 8 (visualization of planting in Chicago and Austin—since 0.3 *mpp* data were available for those two cities only to qualitatively compare clearly).

We implement the cGANs (illustrative inputs and outputs are Figs. 7e and 7f, respectively) to perform the tree localization tasks instead of a discriminative approach because, like real-world, we simulated the existence of a tree in a generative manner. To be more precise, the input segments/blobs to the networks are of non-uniform shapes (see Fig. 8), and our generative approach is robust to such variations. Secondly, this makes every point of a map to be a likely candidate of

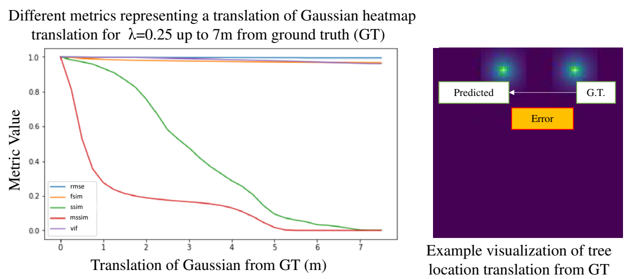


Fig. 10 *Loss functions and error metrics.* Different metrics showing effect of translating Gaussian disc with $\lambda = 0.25$ gradually away from the ground truth position

being a tree/non-tree entity and the cGANs attribute probabilities (shown as heatmaps in Figs. 7 and 10).

Now we discuss three intertwined concepts used in our approach. First, we discuss the process of determining the optimal decay rate λ of the Gaussian discs in the heatmaps, i.e., the spread of the Gaussian distribution of each tree in our approach. Then, we show why we selected Multi-scale SSIM as the loss function for the generator in our cGAN tree location approximator. Lastly, we show an experiment using multiple metrics in order to determine the best one to evaluate tree locations.

Recall that in the heatmaps, the center of a Gaussian represents the highest probability of the presence of a tree which decays exponentially away from the center of tree location: $f_i(x) = e^{-\lambda \cdot x}$, where x is the distance from a point on the map where a tree i was seeded using the synthetic data generator, and λ is the decay rate.

First, we observed heatmaps by varying the value of λ in the range $[0.01, 0.3]$ and plotted the impact on different similarity metrics. For each λ in the range $[0.01, 0.3]$ with increments of 0.01, we plotted five similarity metrics: Multi-scale SSIM [55], Visual Information Fidelity (VIF) [5], Feature Similarity Index (FSIM) [59], root-mean-squared error (RMSE), and standard SSIM [54] (as shown for $\lambda = 0.25$ in Fig. 10). In this experiment we seek a locally optimal value of λ and the locally optimal similarity metric for our tree generating cGANs. For the experiment, we placed one tree's Gaussian heatmap in a chosen position in a fixed tile as ground truth. Then we placed another tree initially at the same position ($d = 0$) and gradually moved it away from ground truth in 0.25 m increments until a distance of 7 m. We computed all the similarity metrics at each position and plotted them as shown in Fig. 10. A well-calibrated Multi-scale SSIM (MSSIM) came out to be the best choice (see Fig. 10) where as the experimental tree moved away from the ground truth position, we observed a rapid decay (but not exceedingly fast) as it was erroneously positioned until it was approximately less than two FONs which is approximately $(2 \times \text{FON}) - 1 = 7\,\text{m}$ apart. At $\lambda = 0.25$, the loss function's penalty showed the desired sensitively. Thus, we chose $\lambda = 0.25$ and incorporated MSSIM into the loss function of our generator in our planting GANs.

A similar approach as above was employed to find the appropriate evaluation metric to employ in evaluating the performance of tree location approximation. While keeping $\lambda = 0.25$ fixed and plotting different similarity metrics as shown in Fig. 10, we choose SSIM because it exhibits a more linear behavior and it was also used in related prior works (see Sect. 5).

5 Results and evaluation

Our framework was implemented in Python using TensorFlow on a machine equipped with four NVIDIA RTX-3090 GPUs. The training time for the segmentation model took less than 2 hours per city, and for the tree generation, the GANs took approximately 5 hours to train per cluster (each with over 100,000 synthetic tiles) with batch size of 16. We use F1-score/Dice coefficient, which is equivalent to IoU in our context, as the metric for segmentation performance and comparative published literature and governmental databases along with human-surveyed data (where available) to evaluate the accuracy of our tree counts and positions. We experimented with several metrics to numerically evaluate tree localization. We tested pixel-based L2-norm, Structured Similarity Index Measure (SSIM) [54], Visual Fidelity (ViF) [5], and Feature-based similarity index [59]. We found SSIM to yield a good correspondence between quantitative and qualitative outputs. The experiment and resultant plots for this choice are given in Sect. 4.4. We further reinforce this selection by noting that SSIM was used in literature (e.g., [1] and [57]) with heatmaps and object counting.

5.1 Parameter values for procedural rules

We derive parameter values for cluster-type rules and city-specific rules using the sources listed in Sect. 4.2. The exact values are reported in Sect. 4.4, and calibration is shown in Fig. 9.

5.2 Spatio-temporal segmentation

Figure 4 shows qualitatively and quantitatively the segmentation performance of using single vs. 12-month snapshots. Further, Fig. 8 shows the visual performance of segmentation over several areas in two of our test cities. For comparison, we also show higher-resolution aerial imagery next to the automatic output produced by our system using 3 *mpp* satellite imagery. We observed that labeling approximately only 10%–20% of a city extent achieved good accuracy in segmentation F1-score and tree localization. Using less than 10% of

Table 3 Tree counts

	C $\times 10^3$	A $\times 10^3$	I $\times 10^3$	L $\times 10^3$	A (subset) $\times 10^3$
Hand-labeled	15.9	19.7	26.83	13.15	–
Austin Tree Inv. [8]	–	–	–	–	7.31
Indiana MFRA [40]	–	–	57.32	–	–
Ours	16.74	21.30	30.33/53.29	13.52	6.84
Our acc. [%]	95.53	91.88	86.94/92.98	97.19	93.82

The raw tree counts from different sources and our output along with accuracy. We note that Indianapolis and Austin had two sources—we report both

Table 4 Comparison of location accuracy

	Chicago		Austin		Indianapolis		Lagos		Combined		
	MSE	SSIM	MSE	SSIM	MSE	SSIM	MSE	SSIM	MAE 4-cities	MSE 4-cities	Median SSIM 4-cities
GT	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.00	1.00
Ours	**1.14**	**0.92**	**1.47**	**0.93**	**1.55**	**0.94**	**1.24**	**0.85**	**0.48**	**1.39**	**0.91**
CSRNet [28] + IADM [29]	5.94	0.71	3.87	0.76	4.00	0.74	4.01	0.74	2.04	4.44	0.74
PSPNet [60]	6.02	0.68	4.01	0.70	5.14	0.68	5.41	0.70	2.41	4.99	0.69
U-Net [45] based on [51]	4.90	0.61	5.17	0.66	5.87	0.70	5.96	0.61	2.47	5.39	0.65
DeepLabV3+ [6]	6.18	0.72	5.08	0.72	5.91	0.70	4.90	0.70	2.58	5.59	0.71
VGG-Net [48] based on [51]	5.97	0.62	5.36	0.69	7.19	0.64	6.89	0.63	2.97	6.21	0.65
Alex-Net [25] based on [51]	9.06	0.56	8.03	0.59	9.33	0.62	8.91	0.69	4.33	8.76	0.62
MobileNetV3 [16]	7.22	0.60	9.15	0.65	9.79	0.59	9.18	0.60	4.68	8.86	0.61

Comparison to state-of-the-art (MSE and SSIM)
Bold values indicate the results showing the highest performance in comparison to other works/papers shown in the table

Table 5 Comparison of counts

	Chicago		Austin		Indianapolis		Lagos	
	MAE	Raw count	MAE	Raw count	MAE	Raw count	MAE	Raw count
GT	0.00	15,912	0.00	19,702	0.00	23,727	0.00	12,790
Ours	**0.30**	**16,624**	**0.64**	**21,301**	**0.52**	**26,826**	**0.25**	**13,150**
CSRNet [28] + IADM [29]	2.03	20,984	1.68	23,906	2.49	28,924	1.94	15,539
PSPNet [60]	3.07	23,593	1.74	24,059	2.67	29,070	2.32	16,082
U-Net [45] based on [51]	2.89	23,075	2.36	25,591	2.19	30,829	2.71	16,621
DeepLabV3 + [6]	2.63	22,475	2.07	24,875	3.13	29,982	2.58	16,447
VGG-Net [48] based on [51]	3.74	25,246	2.18	25,152	3.26	30,255	2.94	16,959
AlexNet [25] based on [51]	4.88	28,104	3.74	29,047	4.68	34,908	3.95	18,390
MobileNetV3 [16]	4.76	27,816	3.87	29,371	5.63	34,998	4.43	19,066

Comparison to state-of-the-art works (raw counts and MAE)
Bold values indicate the results showing the highest performance in comparison to other works/papers shown in the table

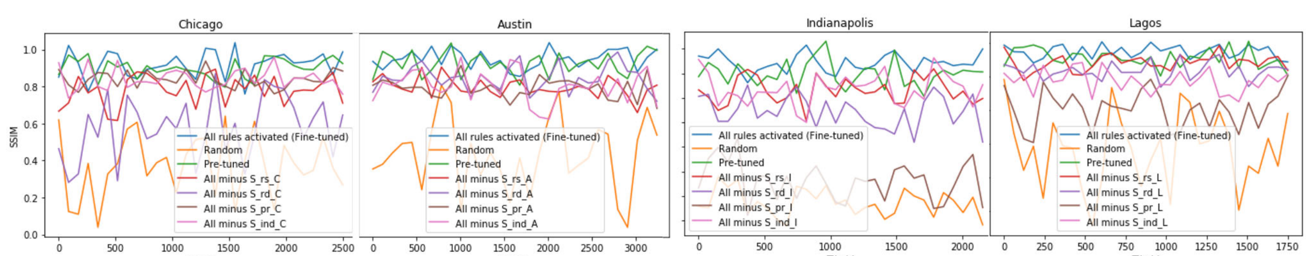


Fig. 11 *Ablation Plots.* Showing SSIM values with respect to ground truth for different rule-sets omissions

Table 6 Knowledge transfer and robustness

		Evaluated on (F1-score/tree count accuracy (%))			
		C	A	I	L
Trained on	Pre-tuned	0.90/93.44	0.84/89.02	0.89/83.79	0.91/95.81
	C	0.91/95.52	0.72/89.75	0.85/82.16	0.88/87.47
	A	0.81/79.06	0.86/91.88	0.79/75.89	0.82/80.62
	I	0.84/82.72	0.82/80.19	0.92/86.94	0.74/71.29
	L	0.86/84.71	0.78/73.55	0.74/72.96	0.92/97.19
	All but itself	0.88/86.83	0.83/80.87	0.87/86.03	0.90/88.91

F1-score and count accuracy (%) by transferring one city (or pre-tuned) model to predict another city

labeled data overfits models and further labeling ($> 20\%$) was not beneficial.

5.3 Tree localization

We present our tree localization performance using two metrics. First, we present a qualitative demonstration using figures to show the placement of trees in different urban configurations. Second, we quantitatively show through an ablation analysis that tree counts and placement accuracy show the best performance with all our rules activated by comparing the system to disabling each rule-set defined in Eq. 1. We also show that the pre-tuned model only marginally loses accuracy compared to the fine-tuned engine, thus exhibiting our approach to be robust. Tree location ground truth was derived by hand-labeling over 70,000 trees on 0.3 *mpp* INRIA dataset [32] and Google Earth [20] for evaluation. Further, we selected areas such that we keep the count of the trees as uniform as possible across all four configurations (residential, roadside, industrial, and park) to illustrate the most representative results. Figure 8 shows inferred tree locations, spatio-temporal segmentation, alongside ground truth (as a subjective illustration). It also shows the difference in image resolution through the map backdrop. We find it important to note as a demonstration of the impact of using temporal data to compensate for lower spatial resolution.

Table 3 and Fig. 11 report the tree counts and placement accuracy of our approach demonstrating the impact of each rule-set of our system. It also shows that we achieve high accuracy in tree count and placement across all test cities. Figure 11 reports the results of the ablation analysis. We illustrate the effect on tree localization as rules are progressively omitted. Figure 11 reinforces the fact that in different cities, certain rule-sets dominate more than others. For instance, it can be seen that in Lagos, the park configuration dominates (i.e., the omission of park rules has the biggest adverse impact). In contrast, for Chicago, roadside configurations make the largest impact.

5.4 Knowledge transfer and robustness

We experimented with training on data of one city and subsequently simulating tree coverage of every other city (including the training city itself, although only 10%–20% of that city was labeled)—see Table 6. We also evaluated and reported the performances on the test cities with cross-validation for the pre-tuned variant by leaving the tested city out of the training samples. The table shows that our approach is capable of being city agnostic with competitive accuracy.

5.5 Tree coverage and localization evaluation

We evaluate the *accuracy* by using governmental reports that encompass the same cities in terms of tree counts and cover. For segmentation/tree cover, we compare our findings to iTree (NLCD data) [53], NDVI-based literature that reported on same areas (as available), and governmental published data (as available) [37,40,53] in Table 7.

Next, we *compare* the performance of our approach to state-of-the-art approaches. We took inspiration from [51] where they adapted recent segmentation networks (e.g., AlexNet [25], VGG-Net [48], and U-Net [45]) to produce tree counts. Contrary to [51], who used 0.8 *mpp* satellite imagery, we use coarser 3 *mpp*. We also compare to DeepLabV3+ [6], MobilenetV3 [16], and PSPNet [60]. Further, we compare to one of the most recent crowd counting networks, namely CSRNet backbone [28] using IADM [29] which is one of the current top benchmarks for crowd counting for the ShanghaiTech dataset. For all of these comparisons, we re-train the solution with our dataset and, where appropriate, adapt the output to density-based heatmaps where the tree count is the integral over the full heatmap (same methodology defined in [51]). For [29], we partition every month's 4D images and map them to one target (thereby utilizing 48D data) to adapt the problem statement in our paper to their paper's original architecture.

Table 4 compares all our test cities with results sorted in order of decreasing average performance over all cities (in the set). Our method performs best *in all cases*. We empha-

Table 7 Tree coverage

	C (%)	A (%)	I (%)	L (%)
J. McBride [34]	18.54	–	–	–
Nowak et al. [37]	–	30.8	–	–
Indiana MFRA [40]	–	–	20.5	–
[GT for US] iTree/NLCD [53])	11.61	34.42	18.98	–
[GT for Lagos] UNFAO [12]	–	–	–	9.71
Ours	**12.98**	**32.42**	**20.91**	**8.67**

Evaluation of our system with respect to other sources of land/tree cover percentage

size that our work produces tree locations, as well as tree counts, for which a deep learning-based approach at city scale has not been published to the best of our knowledge. Further, our method requires significantly less effort (i.e., crowd-sourcing-based manual tree count estimation is not needed).

6 Conclusions and future work

We have shown an approach that exploits spatio-temporal satellite images and urban procedural vegetation rules to create a system for high-quality tree localization. Our method processes entire cities automatically and quickly, obtaining tree count accuracy in the 87–97% range and overall performance superior to a wide range of recent deep segmentation and counting methods.

We foresee potential in identifying species by extending our method to consider their different year-long behavior. Further, we surmise our method shows promise in other domains besides vegetation where any entity is spatially semi-stationary yet temporally dynamic (e.g., crowds, celestial bodies, ant colonies, bee swarms, etc.). Therefore, our work is the basis for a future framework to model temporally varying data patterns with spatial features.

Acknowledgements This research was funded in part by National Science Foundation grant #10001387, Functional Proceduralization of 3D Geometric Models and by the Foundation for Food and Agriculture Research Grant ID: 602757. We thank the Integrated Digital Forestry Initiative (iDIF) at Purdue University for their partial support. We also thank NSF grant #1835739 “U-Cube: A Cyberinfrastructure for Unified and Ubiquitous Urban Canopy Parameterization” and NSF grant #2106717 “Deep Generative Modeling for Urban and Archaeological Recovery.”

Declarations

Conflict of interest The authors claim and announce no conflict of interest with any entity or party in relation to this work.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: NeurIPS, NIPS’18, pp. 9525–9536 (2018)
2. Aliaga, D.G., Vanegas, C., Lei, M., Niyogi, D.: Visualization-based decision tool for urban meteorological modeling. Environ. Plann. B. Plann. Des. **40**(2), 271–288 (2013)
3. Arief, H.A., Strand, G.H., Tveite, H., Indahl, U.G.: Land cover segmentation of airborne lidar data using stochastic atrous network. Remote Sens. **10**(6), (2018)
4. Benes, B., Massih, M.A., Jarvis, P., Aliaga, D.G., Vanegas, C.A.: Urban ecosystem design. In: Symposium on Interactive 3D Graphics and Games, I3D ’11, pp. 167–174. Association for Computing Machinery (2011)
5. Bovik, A.: A visual information fidelity approach to video quality assesment (2005)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
7. Chicago Department of Transportation: Street and site plan design standards (2007). https://www.chicago.gov/dam/city/depts/cdot/StreetandSitePlanDesignStandards407.pdf
8. City of Austin Texas: Downtown Tree Inventory (2013). https://data.austintexas.gov/Locations-and-Maps/Downtown-Tree-Inventory-2013
9. City of Indianapolis/Marion, Indiana: Code of ordinances—chapter 701—trees and flora (2021). https://library.municode.com/in/indianapolis-marion_county/codes/code_of_ordinances
10. City of New York: NYC open data map tiles (2021). https://maps.nyc.gov/tiles
11. Earth Observing System (EOS), NASA: The Ice, Cloud, and Land Elevation Satellite-2 (2021). https://icesat-2.gsfc.nasa.gov/icesat-2-data
12. Food and Agriculture Organization of the United Nations: Global forest resources assessment country report series—Nigeria. Tech. rep, United Nations (2010)
13. 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. Graph. **39**(5) (2020)
14. Hahsler, M., Piekenbrock, M., Doran, D.: dbscan: Fast density-based clustering with R. J. Stat. Softw. **91**(1), 1–30 (2019)
15. Hojas-Gascon, L., Eva, H.: Field guide for forest mapping with high resolution satellite data. monitoring deforestation and forest degradation in the context of the un-redd programme. the tanzania redd + initiative (2014). https://doi.org/10.2788/657954
16. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H.: Searching for mobilenetv3. In: ICCV (2019)
17. Hu, Q., Yang, B., Khalid, S., Xiao, W., Trigoni, N., Markham, A.: Towards semantic segmentation of urban-scale 3d point clouds: a dataset, benchmarks and challenges (2020). https://doi.org/10.48550/ARXIV.2009.03137
18. Huang, S., Tang, L., Hupy, J., Wang, Y., Shao, G.: A commentary review on the use of normalized difference vegetation index (ndvi) in the era of popular remote sensing. J. For. Res. **32** (2020)
19. Huang, Z., Arian, A., Yuan, Y., Chiu, Y.C.: Using conditional generative adversarial nets and heat maps with simulation-accelerated training to predict the spatiotemporal impacts of highway incidents. Transp. Res. Rec. **2674**(8), 836–849 (2020)
20. Inc., G.: Google Earth (2021). http://earth.google.com
21. Japan Aerospace Exploration Agency—JAXA: JAXA (2021). https://global.jaxa.jp

22. Jiang, Z., Huete, A., Chen, J., Chen, Y., Li, J., Yan, G., Zou, Y.: Analysis of ndvi and scaled difference vegetation index retrievals of vegetation fraction. Remote Sens. Environ. **101**, 366–378 (2006)
23. Keren, S.: Modeling tree species count data in the understory and canopy layer of two mixed old-growth forests in the dinaric region. Forests **11**(5) (2020)
24. Kim, J.S., Kavak, H., Crooks, A.: Procedural city generation beyond game development. SIGSPATIAL Special **10**(2), 34–41 (2018)
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS, vol. 25 (2012)
26. Lee, S., Park, S., Son, S., Han, J., Kim, S., Kim, J.: Land cover segmentation of aerial imagery using segnet. In: Earth Resources and Environmental Remote Sensing/GIS Applications X, vol. 11156, pp. 313–318. Intl. Society for Optics and Photonics, SPIE (2019)
27. Li, B., Kałużny, J., Klein, J., Michels, D.L., Pałubicki, W., Benes, B., Pirk, S.: Learning to reconstruct botanical trees from single images. ACM Trans. Graph. **40**(6) (2021)
28. Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density map regression guided detection network for rgb-d crowd counting and localization. In: CVPR, pp. 1821–1830 (2019). https://doi.org/10.1109/CVPR.2019.00192
29. Liu, L., Chen, J., Wu, H., Li, G., Li, C., Lin, L.: Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting. In: CVPR, pp. 4823–4833 (2021)
30. Liu, Y., Guo, J., Benes, B., Deussen, O., Zhang, X., Huang, H.: Treepartnet: Neural decomposition of point clouds for 3d tree reconstruction. ACM Trans. Graph. **40**(6) (2021)
31. Lu, D., Weng, Q.: A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens. **28**(5), 823–70 (2007)
32. Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P.: The inria aerial image labeling benchmark. In: 2017 IEEE Intl. Geoscience and Remote Sensing Symposium (IGARSS) (2017)
33. Martinovic, A., Van Gool, L.: Bayesian grammar learning for inverse procedural modeling. In: CVPR, pp. 201–208 (2013)
34. McBride, J.: Mapping chicago area urban tree canopy using color infrared imagery. Ph.D. thesis, Lund University, Lund, Sweden (2011)
35. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Van Gool, L., Purgathofer, W.: A survey of urban reconstruction. In: Computer Graphics Forum, vol. 32, pp. 146–177. Wiley Online Library (2013)
36. Niese, T., Pirk, S., Albrecht, M., Benes, B., Deussen, O.: Procedural urban forestry. ACM Trans. Graph. **41**(2) (2022)
37. Nowak, D., Bodine, A.R., Hoehn, R.E., Edgar, C., Hartel, D., Lister, T., Brandeis, T.: Austin's urban forest (2016)
38. Nowak, D., Heisler, N.: Air quality effects of urban trees and parks. Research Series Monograph. National Recreation and Parks Association Research Series Monograph, Ashburn, VA (2010). https://www.fs.usda.gov/treesearch/pubs/52881
39. Open Street Map Contributors: Planet Dump (2021). https://planet.osm.org. https://www.openstreetmap.org
40. Paula, P., Gregory, M., James, S., Kelaine, V., Qingfu, X.: City of Indianapolis, Indiana Municipal Forest Resource Analysis. Tech. rep. Center for Urban Forest Research USDA Forest Service (2008)
41. Planet: Planet Explorer (2021). http://planet.com/explorer
42. Potapov, P., Li, X., Hernandez-Serna, A., Tyukavina, A., Hansen, M., Kommareddy, A., Pickens, A., Turubanova, S., Tang, H., Silva, C.E., Armston, J., Dubayah, R., Blair, J.B., Hofton, M.: Mapping global forest canopy height through integration of gedi and landsat data. Remote Sens. Environ. **253**, 112–165 (2021)
43. Pretzsch, H., Biber, P., Uhl, E., Dahlhausen, J., Rötzer, T., Caldentey, J., Koike, T., van Con, T., Chavanne, A., Seifert, T., du Toit, B., Farnden, C., Pauleit, S.: Crown size and growing space requirement of common tree species in urban centres, parks, and forests. Urban For. Urban Greening **14**(3), 466–479 (2015)
44. Prusinkiewicz, P., Lindenmayer, A.: The Algorithmic Beauty of Plants. Springer-Verlag, New York (1990)
45. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI 2015, pp. 234–241. Cham (2015)
46. Rußwurm, M., Körner, M.: Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In: CVPR, pp. 1496–1504 (2017)
47. Shen, B., Fang, S., Li, G.: Vegetation coverage changes and their response to meteorological variables from 2000 to 2009 in Naqu, Tibet, China. Can. J. Remote Sens. **40**, 67–74 (2014)
48. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
49. Tinchev, G., Nobili, S., Fallon, M.: Seeing the Wood for the Trees: Reliable Localization in Urban and Natural Environments (2018). https://doi.org/10.48550/ARXIV.1809.02846
50. Tree Care Industry Association (TCIA): ANSI A300 Standards: Tree, Shrub and Other Woody Plant Management—Standard Practices. American National Standards Institute (2017). https://www.tcia.org/TCIA/Build_Your_Business/A300_Standards/A300_Standards.aspx
51. Tree counting with high spatial-resolution satellite imagery based on deep neural networks. Ecol. Ind. **125**, 107591 (2021)
52. Urban Design Division/Planning & Development Review Department: City of austin great streets (2012)
53. USDA Forest Service: iTree (2020). https://www.itreetools.org
54. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Img. Proc. **13**(4), 600–612 (2004)
55. Wang, Z., Simoncelli, E., Bovik, A.: Multiscale structural similarity for image quality assessment. Conf. Record Asilomar Conf. Signals Syst. Comp. **2** (2003)
56. Weier, J., Herring, D.: Measuring vegetation (ndvi and evi (2000). https://earthobservatory.nasa.gov/features/MeasuringVegetation
57. Yang, S.D., Su, H.T., Hsu, W.H., Chen, W.C.: Class-agnostic few-shot object counting. In: IEEE/CVF WACV, pp. 870–878 (2021)
58. Zabelskyte, G., Kabisch, N., Stasiskiene, Z.: Patterns of urban green space use applying social media data: a systematic literature review. Land **11**(2) (2022)
59. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: A feature similarity index for image quality assessment. IEEE Trans. Img. Proc. **20**(8) (2011)
60. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR, pp. 6230–6239 (2017)

Adnan Firoze is a Research Assistant and PhD student of Computer Science at Purdue University. His research focuses on pattern recognition, computer vision, machine learning, and graphics. He was formerly a Teaching Fellow at the Computer Science Department at Columbia University in the City of New York. He completed his Dual M.S. in computer science and journalism in 2016 from Columbia University with distinction. He had graduated summa cum laude in B.S. in Computer Science from North South University, Bangladesh. He worked at the Computer Vision and Cybernetics Group BD. His interdisciplinary works are based on computer vision, image processing, machine learning, fuzzy logic, and graphics. His works have appeared in numerous conferences and journals, e.g., IEEE International Conference of Machine Learning and Applications (ICMLA'18), IEEE's 2012 International Conference on Machine Learning and Cybernetics (ICMLC), ACM's 13th International Conference on Enterprise Information Systems (ICEIS), International Journal of Healthcare Information Systems and Informatics (IJHISI), among others. His present research revolves around pattern recognition and deep learning in the space of graphics and computer vision.

Bedrich Benes is a Professor of Computer Science at Purdue University. He received his Ph.D. from Czech Technical University in Prague. Bedrich is a senior member of ACM and IEEE and a Fellow of the European Association for Computer Graphics. He is a member of seven Journal Editorial Boards, including Associate Editor of in Silico Plants (Oxford Academic), Associate Editor of Computer Graphics Forum (Wiley), Associate Editor of Computers & Graphics (Elsevier), Associate Editor of Computer Animation and Virtual Worlds (Wiley), and Associate Editor of IEEE Computer Graphics Applications. Bedrich was Editor in Chief of Computer Graphics Forum (Willey) from 2018 to 2021. He works in generative methods for geometry synthesis, and his main focus is on procedural, inverse procedural modeling, simulation of natural phenomena, and additive manufacturing. He has published over 180 research papers in the field.

Daniel Aliaga is an Associate Professor of Computer Science at Purdue University. He has PhD degree in Computer Science from UNC Chapel Hill. Dr. Aliaga's first computer graphics publication was in 1991, and he now has over 140 refereed publications covering multiple disciplines, holds membership in 80+ program committees, and has given over 50 invited national and international talks. His research is in the areas of inverse procedural modeling, urban modeling and simulation, and imaging and 3D reconstruction. Dr. Aliaga's funding sources include NSF, IARPA, Internet2, MTC, Google, Microsoft, and Adobe. Prof. Aliaga is Associate Editor for IEEE TVCG and for Visual Computing Journal (previously for Computer Graphics Forum and Graphical Models) and PC member for SIGGRAPH, CVPR, ICCV, Eurographics, AAAI, NeurIPS, I3D, IEEE Vis. He has received a Fulbright Scholar Award, a Discovery Park Faculty Research Fellowship, and his PhD advisees have received a total of 11 Purdue fellowships/grants. He is a member of ACM SIGGRAPH and ACM SIGGRAPH Pioneers.