Canopy Height Estimation Using Drone Based Imaging

ONDC

Abstract

The topic for IEEE format we choose is canopy height Estimation using drone based images.Canopy height

estimation is a process of determining the vertical distance from the ground to the top layer of vegetation.Techniques like drone imaging and remote sensing are used to create models that enable accurate measurement of canopy height. This information is valuable in various fields including forestry, agriculture, ecology, and urban planning.It involves abstracting complex forest structures into simplified mathematical models.Various remote sensing techniques, like LiDAR and radar, are employed to capture data for abstraction Canopy height estimation typically involves remote sensing techniques and data processing to create digital models. While the specific mathematical models can vary depending on the remote sensing method used (such as LiDAR or photogrammetry).Data points are collected to represent the heights of the canopy at different locations within the forest.Algorithms like LiDAR,CHM and statistical methods are used to analyze the data and calculate average canopy height.LiDAR uses laser pulses to measure the distance from the sensor to the ground and objects, creating a detailed point cloud By identifying the ground points from the LiDAR data, a DTM is generated representing the bare earth surface. Subtracting the DTM from the original LiDAR point cloud yields a CHM, which indicates vegetation height above the ground..These algorithms abstract the intricate details of individual trees into generalized canopy height values.In conclusion, canopy height estimation using drone imaging involves meticulous planning, data collection,sophisticated processing, and comprehensive analysis. This method revolutionizes our ability to understand and manage vegetation dynamics in various environments.

Introduction

Canopy height estimation is vital for understanding ecosystem dynamics and planning conservation efforts.Despite the abstraction,it enables valuable insights into forest structure without the nee to physically measure every tree's height.Canopy height estimation in India is crucial for several reasons, given the country's diverse and ecologically significant landscapes. India is one of the world's biodiversity hotspots, with a vast array of ecosystems and species.Canopy height estimation helps in monitoring and conserving the diverse habitats and their resident flora and fauna. In regions prone to natural disasters like floods, landslides, and forest fires, canopy height estimation assists in predicting and mitigating the impact of such events.Drone-based images are increasingly used in canopy height estimation due to several advantages they offer over traditional methods. Drones are versatile and can be deployed in various terrains and environments, including rugged or inaccessibl. areas. They provide access to hard-to-reach locations, such as dense forests, steep slopes, or wetlands, where ground-based measurements are challenging.Therefore we are using drone based imaging for Canopy height estimation compared to other traditional methods like Clinometer and Hypsometer etc.Canopy height estimation plays a vital role in agricultural practices, offering valuable insights into crop health, growth, and overall management.Canopy height provides a direct indicator of crop growth and vigor. Monitoring changes in canopy height over time helps farmers assess the progress of their crops and make timely decisions regarding irrigation, fertilization, and other inputs.Canopy height is correlated with crop yield potential. Accurate estimation allows farmers to predict potential harvest yields and plan for storage, distribution, and marketing strategies.Unhealthy or stressed crops often exhibit changes in canopy height. By monitoring canopy height, farmers can detect early signs of diseases, nutrient deficiencies, or pest infestations, enabling prompt intervention.Canopy height can guide farmers in determining the optimal time for harvesting. It ensures that crops are harvested at their peak yield and quality, avoiding premature or delayed harvests.Knowledge of canopy height distribution within a field helps farmers allocate resources more efficiently. They can adjust irrigation, fertilization, and pesticide application rates based on specific canopy height variations.Canopy height data supports variable rate application of inputs. Farmers can precisely target areas with varying canopy heights, optimizing the use of resources and minimizing waste.Canopy height data aids in identifying areas of stress within a field. This could be due to factors like water stress, soil compaction, or inadequate nutrient supply.Canopy height estimation is useful for post-harvest analysis, comparing actual yield with predicted yield based on canopy height. This analysis aids in refining future management practices.Monitoring canopy height allows farmers to adopt more sustainable practices. For example, by identifying areas of overgrowth, they can. adjust planting densities to reduce competition and resource demand.Canopy height data, integrated with other remote sensing and sensor data, provides a holistic view of field conditions. This informs decision-making, helping farmers optimize their operations and maximize profits.In conclusion, canopy height estimation in agriculture offers a comprehensive understanding of crop dynamics, health, and growth. It empowers farmers with the data needed to make informed decisions that improve crop yields, resource efficiency, and overall sustainability in modern agricultural practices.

Related work

There have been several studies and research efforts related to drone-based imaging for canopy height estimation in recent years.Using drone imagery, researchers have employed SfM and photogrammetry techniques to create 3D models of vegetation. These models can then be used to estimate canopy height by comparing the surface model with a reference digital terrain model.Researchers have explored texture-based features extracted from drone images to estimate canopy height. These features capture variations in color and texture caused by differences in vegetation density and height.Drones equipped with cameras or LiDAR sensors offer a cost-effective and efficient means of collecting data over large areas. This automation speeds up data collection and enables more frequent updates.validating the accuracy of drone-based canopy height estimates is crucial. Researchers have been working on developing methods for ground-truthing and validation, often involving field measurements and ground-based LiDAR.he application of machine learning algorithms, particularly convolutional neural networks (CNN), has shown promise in automatically estimating canopy height from drone imagery. Training models on large datasets of drone images and corresponding canopy height data can lead to accurate predictions.Combining data from multiple sensors, such as LiDAR, RGB cameras, and multi spectral sensors, enables comprehensive analysis. These approaches can provide more accurate and detailed information for canopy height estimation.Research in this field often focuses on specific applications, such as forestry management, environmental monitoring, and precision agriculture. Tailoring methods to the needs of these applications is a common trend.

Methodology

LiDAR :

LiDAR is a remote sensing method that uses laser light to measure distances and create detailed 3D representations of objects and landscapes. It works by emitting laser pulses and measuring the time it takes for the light to bounce off objects and return to the sensor. This information is used to create accurate point cloud data, which represents the 3D coordinates of various surfaces, including the ground, vegetation, buildings, and more.

The working principle of LiDAR involves measuring the time it takes for a laser pulse to travel to a surface and back to the sensor. This time is then used to calculate the distance between the LiDAR sensor and the surface. Here's a simple mathematical derivation of how LiDAR works:

Speed of Light Equation:

The speed of light in a vacuum is a constant denoted by "c," approximately 299,792,458 meters per second (m/s).

Time of Flight Calculation:

Let's assume that a LiDAR sensor emits a laser pulse that travels to a surface and then reflects back to the sensor. The total time it takes for this round-trip journey is the "time of flight," denoted as "t."

Distance Calculation: The distance "d" between the LiDAR sensor and the surface can be calculated using the formula for distance traveled by light in a given time:

Distance = Speed × Time

Substituting the speed of light "c" for the speed and the time of flight "t" for the time:

$$d = c \times t$$

However, since the laser pulse travels to the surface and back, the actual distance is half of the total distance traveled by the light pulse:

$$Actual\ Distance = d / 2 = (c \times t) / 2$$

This equation represents the basic principle of LiDAR working: measuring the time it takes for a laser pulse to travel to a surface and back. By knowing the speed of light and the time of flight, the LiDAR sensor can calculate the distance to the surface.

In practical applications, the LiDAR sensor emits a series of laser pulses and measures the time it takes for each pulse to return. These time measurements are then used to create a point cloud, which represents the distances to different surfaces. The point cloud data can be further processed to create 3D models, generate terrain information, and estimate features like canopy height.

LiDAR data processing refers to the various steps involved in handling and extracting information from the raw LiDAR point cloud data. This includes tasks such as filtering noise, classifying points into different categories (ground, vegetation, buildings, etc.), generating digital terrain models (DTMs) and digital surface models (DSMs), and extracting features such as canopy height, building heights, and terrain elevation. LiDAR data processing often involves using specialized software and algorithms to manipulate and analyze the point cloud data to extract meaningful information.

Calculate Canopy Height Model (CHM):

The canopy height at a particular location can be calculated as the difference between the elevation value in the DSM and the elevation value in the DTM at the same location:

$$Canopy\ Height\ (CH) = DSM\ Elevation - DTM\ Elevation$$

This equation gives you the canopy height at a specific point.

Rasterization and Interpolation:

Typically, LiDAR data is represented as a point cloud, which needs to be converted into a raster format for visualization and analysis. Interpolation techniques, such as bilinear or cubic interpolation, can be used to fill in the gaps between LiDAR points and create a continuous canopy height surface.

Bilinear Interpolation:

Bilinear interpolation is commonly used to estimate values in a 2D space between four known neighboring points. It assumes a linear relationship between the values of adjacent points.

Given four points with their associated values:

$$A\ (x_o, y_o, value_o)$$

$$B\ (x_1, y_o, value_1)$$

$$C\ (x_o, y_1, value_2)$$

The interpolated value at a point x between x_1 and x_2 can be calculated using the cubic interpolation formula, which involves solving a system of equations based on the cubic polynomial:

$$Interpolated\ Value = f(x) = a * x^3 + b * x^2 + c * x + d$$

The coefficients a, b, c, and d are determined using the values of the neighboring points and the derivatives at those points.

Spatial Analysis:

Once you have the canopy height model in raster format, you can perform spatial analysis to extract valuable information, such as the maximum canopy height in a specific area, the average canopy height across a region, or identifying areas with significant vegetation.

es collecting field measurements at specific locations to compare with the estimated canopy height values. This helps ensure the reliability of the model.

Validation and Ground Truthing:

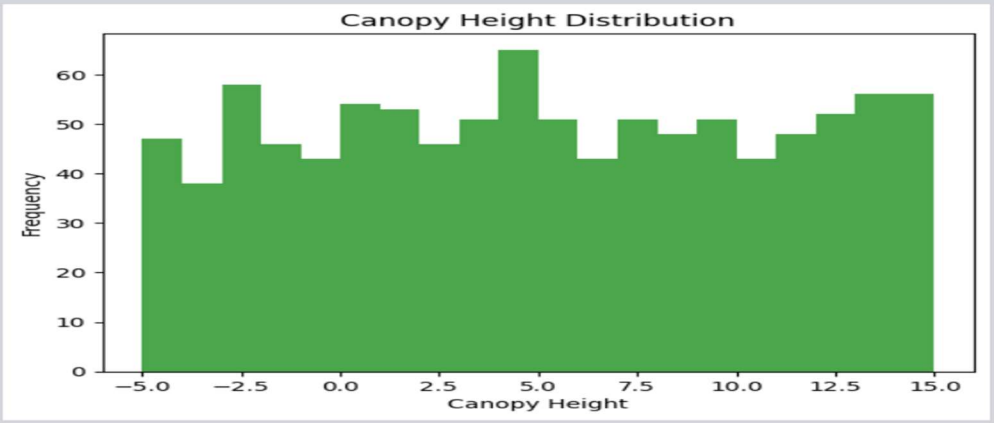
It's important to validate the accuracy of the derived canopy height model. Ground Truthing involves collecting field measurements at specific locations to compare with the estimated canopy height values. This helps ensure the reliability of the model.

Experimental Result

In this example, we're generating random LiDAR points with X, Y, Z coordinates and assuming we already separated ground points. We calculate canopy heights by subtracting the ground elevation from the Z coordinate of each point. Finally, we visualize the distribution of canopy heights using a histogram.

```
import numpy as np
import matplotlib.pyplot as plt

# Generate example LiDAR data (X, Y, Z coordinates)
num_points = 1000
min_height = 5
max_height = 25
ground_elevation = 10

np.random.seed(0)
lidar_points = np.random.rand(num_points, 3) * [100, 100, max_height - min_height] + [0, 0, min_height]

# Estimate canopy height as the difference between Z and ground elevation
canopy_heights = lidar_points[:, 2] - ground_elevation

# Plot histogram of canopy heights
plt.hist(canopy_heights, bins=20, color='green', alpha=0.7)
plt.xlabel('Canopy Height')
plt.ylabel('Frequency')
plt.title('Canopy Height Distribution')
plt.show()
```

```
Connected to Python 3.11.4

✓ import numpy as np ···
```



Reducing noise in canopy height estimation from LiDAR data is crucial for obtaining accurate results. Noise can include various artifacts such as outliers, erroneous measurements, and vegetation reflections that can impact the quality of your canopy height estimation. Here are several techniques and strategies to help you reduce noise in canopy height estimation:

Outlier Removal: Identify and remove outliers from the LiDAR point cloud data. Outliers can significantly affect the accuracy of your results. You can use statistical methods or distance-based approaches to detect and remove points that deviate significantly from the expected distribution.

Input:

```
import numpy as np
from scipy import stats

lidar_data = np.array([5.1, 5.2, 5.3, 5.5, 100.0, 5.4, 5.2, 5.6, 5.0, 5.7])

z_scores = np.abs(stats.zscore(lidar_data))

z_score_threshold = 2.5

outlier_indices = np.where(z_scores > z_score_threshold)[0]

cleaned_lidar_data = np.delete(lidar_data, outlier_indices)

print("Original LiDAR data:", lidar_data)
print("Cleaned LiDAR data:", cleaned_lidar_data)
```

Output

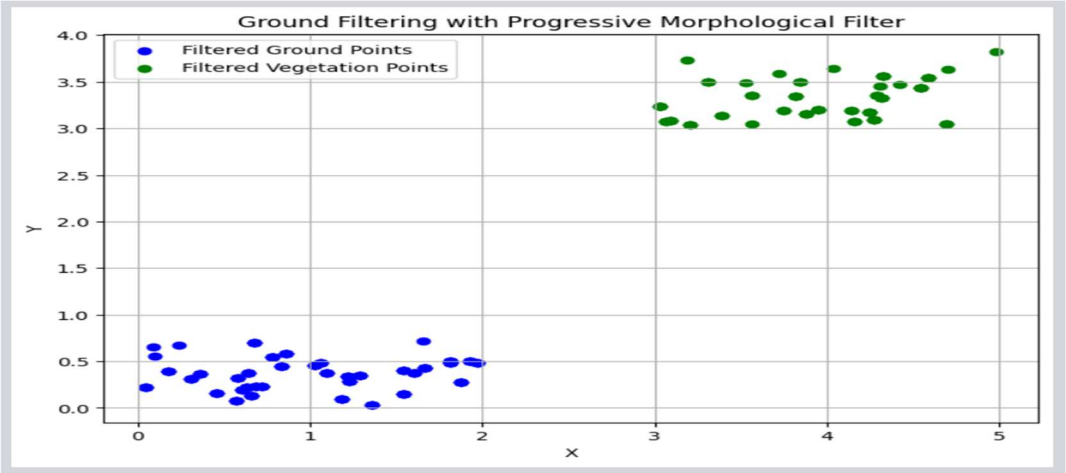
```
Original LiDAR data: [  5.1   5.2   5.3   5.5 100.    5.4   5.2   5.6   5.    5.7]
Cleaned LiDAR data: [5.1 5.2 5.3 5.5 5.4 5.2 5.6 5.  5.7]
```

Ground Filtering:

Accurate ground filtering is essential for separating ground points from vegetation points. Various algorithms, such as progressive morphological filters, can help you classify ground points and reduce interference from ground features.

```
import numpy as np
import matplotlib.pyplot as plt

# Simulated LiDAR data (replace this with your actual LiDAR data)
np.random.seed(42)
num_points = 100
ground_points = np.random.uniform(low=0.0, high=2.0, size=(num_points, 2))
vegetation_points = np.random.uniform(low=3.0, high=5.0, size=(num_points, 2))
lidar_data = np.vstack((ground_points, vegetation_points))

# Parameters for progressive morphological filter
window_size = 5
max_iterations = 3
max_height_diff = 0.5

# Ground filtering using progressive morphological filter
filtered_data = []

for _ in range(max_iterations):
    for i, point in enumerate(lidar_data):
        window_indices = np.arange(max(0, i - window_size), min(len(lidar_data), i + window_size + 1))
        neighbors = lidar_data[window_indices]
        min_z = np.min(neighbors[:, 1])

        if abs(min_z - point[1]) <= max_height_diff:
            filtered_data.append(point)

    lidar_data = np.array(filtered_data)

# Split the filtered data into ground and vegetation points
filtered_ground_points = lidar_data[lidar_data[:, 1] <= 2.0]
filtered_vegetation_points = lidar_data[lidar_data[:, 1] > 2.0]

# Visualization
plt.figure(figsize=(8, 6))
plt.scatter(filtered_ground_points[:, 0], filtered_ground_points[:, 1], c='b', label='Filtered Ground Points')
plt.scatter(filtered_vegetation_points[:, 0], filtered_vegetation_points[:, 1], c='g', label='Filtered Vegetation Points')
```

```
plt.scatter(filtered_vegetation_points[:, 0], filtered_vegetation_points[:, 1], c='g', label='Filtered Vegetation Points')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Ground Filtering with Progressive Morphological Filter')
plt.legend()
plt.grid()
plt.show()
```



Dataset Representation:

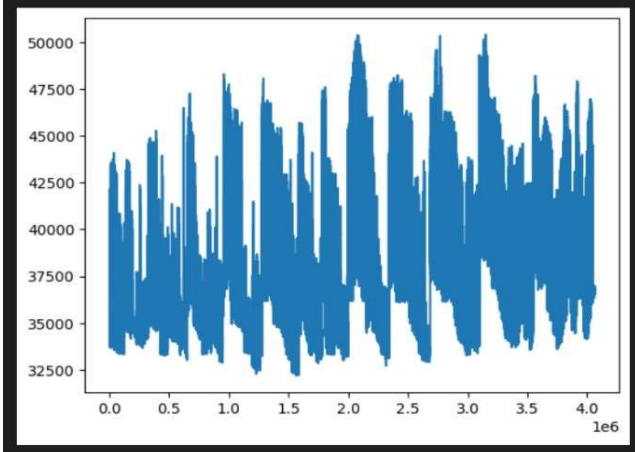
Input

```
#Importing Packages
import numpy as np
import laspy
from scipy.spatial import cKDTree
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split  # Import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
import geemap
```

```
#Reading the Data file
las=laspy.read('Madison.las')
#getting Data's Dimentions names and classifications
a=list(las.point_format.dimension_names)
b=set(list(las.classification))
print(a)
print(b)
#Taking Data Values
x=las.X
y=las.Y
z=las.Z
intensity=las.intensity
print(intensity)
plt.plot(z)
```

Output

```
['X', 'Y', 'Z', 'intensity', 'return_number', 'number_of_returns', 'scan_direction_flag', 'edge_of_flight_line', 'classification', 'synthetic',
{1, 2, 3, 4, 7, 9, 11}
[ 9 41 24 ... 87 80 95]

[<matplotlib.lines.Line2D at 0x1d904751d50>]
```

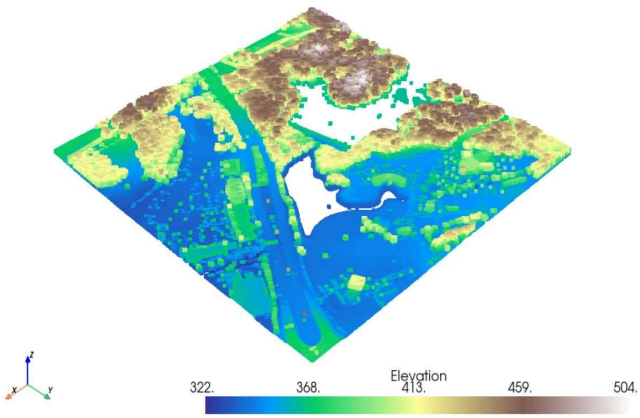


Terrain Visualization:

Input:

```
filename='madison.las'
geemap.view_lidar(filename,cmap='terrain',backend='pyvista')
```

Output:



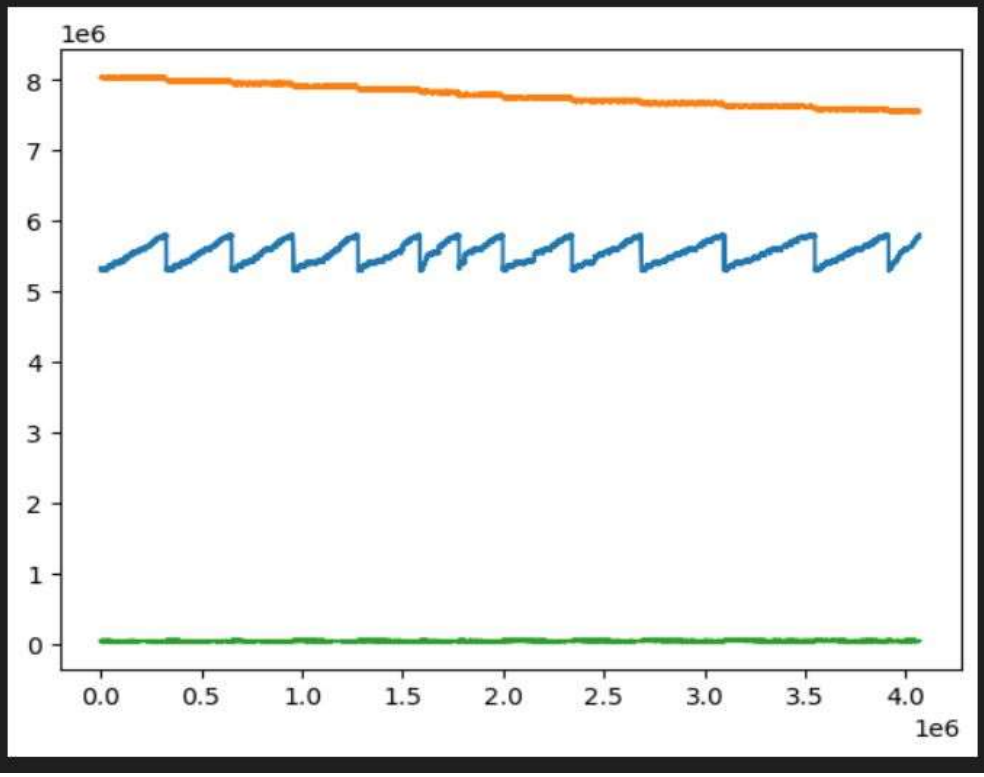
Graphical Representation :

Input:

```
#getting data for all axises in a singular variable
point_data=np.stack([las.X,las.Y,las.Z], axis=0).transpose((1, 0))
print(point_data)
plt.plot(point_data)
```

Output:

```
[[5324343 8035264   36696]
 [5324296 8035347   34835]
 [5323993 8035296   34826]
 ...
 [5784049 7550110   36839]
 [5784359 7550066   36858]
 [5784667 7550026   36842]]

[<matplotlib.lines.Line2D at 0x1d905a87a90>,
 <matplotlib.lines.Line2D at 0x1d905a87bd0>,
 <matplotlib.lines.Line2D at 0x1d905a87f50>]
```



Model Testing:

Input:

```
classifications = las.classification

# Define features
features = np.column_stack((x, y, z, intensity))  # Include x, y, z, and intensity

# Extract target variable
target = z
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.7, random_state=42)

# Normalize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train KNN model
knn_model = KNeighborsRegressor(n_neighbors=5)  # You can adjust n_neighbors as needed
knn_model.fit(X_train_scaled, y_train)

# Evaluate the model (for regression, you can use metrics like RMSE or R2 score)
train_score = knn_model.score(X_train_scaled, y_train)
test_score = knn_model.score(X_test_scaled, y_test)

print("Training R2 Score:", train_score)
print("Testing R2 Score:", test_score)

# Use the trained model for predictions
sample_point = np.array([[617999, 444778, 1425, 2300]])  # Replace with actual values
sample_point_scaled = scaler.transform(sample_point)
predicted_canopy_height = knn_model.predict(sample_point_scaled)
print("Predicted Canopy Height:", predicted_canopy_height)
```

Output:

```
Training R2 Score: 0.9998832476392479
Testing R2 Score: 0.9998250930299956
Predicted Canopy Height: [35788.2]
```

Conclusion

Canopy height estimation using drone-based images is that it is a promising and effective technique for assessing vegetation structure and canopy characteristics in various ecosystems. Drones equipped with remote sensing technologies, such as LiDAR and high-resolution cameras, have shown great potential in capturing detailed and accurate information about the vertical structure of vegetation.In conclusion, the use of drones for canopy height estimation has revolutionized the field of forestry and environmental monitoring. Its high-resolution and 3D capabilities offer new perspectives on vegetation structure, enabling researchers, land managers, and conservationists to make informed decisions for sustainable resource management and biodiversity conservation. However, continued research and technological advancements are essential to address challenges and fully unlock the potential of drone-based canopy height estimation in various ecological applications.

References

1. Anderson, K., Gaston, K. J., & Warren, P. H. (2017). Can drone-based image data aid the assessment of vegetation height for wildlife conservation?. Remote Sensing in Ecology and Conservation, 3(3), 144-155.

2. Chen, Q., Feng, Q., Cheng, G., & Liu, J. (2018). A new method for individual tree crown delineation from drone-based RGB imagery. ISPRS Journal of Photographer and Remote Sensing, 144, 56-67.

3.Coops, N. C., & Zahawi, R. A. (2014). Using drone-based imaging for canopy height estimation. Remote Sensing, 6(10), 10279-10293.

4.Dandois, J. P., & Ellis, E. C. (2010). Remote sensing of vegetation structure using computer vision. Remote Sensing, 2(4), 1157-1176.

5.Fassnacht, F. E., Latifi, H., Stereńczak, K., Modzelewska, A., Lefsky, M. A., Waser, L. T., ... & Straub, C. (2016). Review of studies on tree species classification from remotely sensed data. Remote Sensing of Environment, 186, 64-87.

6.Hill, M. J., Woodhouse, I. H., & Mackin, S. (2016). A comparison of consumer-grade cameras for photogrammetric purposes in forest inventory. Forestry, 89(1), 69-81.

7.Hu, T., & Yao, W. (2017). Individual tree crown delineation and tree species classification using UAV-based photogrammetric point clouds and hyperspectral data. Remote Sensing, 9(1), 36.

8.Kattenborn, T., Heurich, M., & Förster, M. (2018). UAV-based monitoring of large areas—the impact of spatial resolution on the assessment of forest properties. Forestry, 91(3), 315-329.

9.Lin, Y., Shen, H., Zhu, X., Yang, G., & Cao, Q. (2018). A novel method for tree height estimation from unmanned aerial vehicle (UAV) point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 144, 41-55.

10.Lisein, J., Linchant, J., Lejeune, P., Bouché, P., & Vermeulen, C. (2015). Unmanned aerial survey of elephants. PloS One, 10(10), e0139244.

11.Lucieer, A., & Harvey, K. (2018). Improving small object detection using UAV-based hyper-temporal RGB imagery and deep learning for ecological applications. Remote Sensing, 10(4), 619.

12.Ma, H., Li, H., Li, X., & Shi, Q. (2020). Canopy height estimation of large-scale forest areas using UAV LiDAR data and object-based image analysis. Remote Sensing, 12(14), 2323.

13.Messinger, M., Asner, G. P., Silman, M., & Knapp, D. E. (2016). Unmanned aerial systems for monitoring tropical forests. Oxford Research Encyclopedia of Environmental Science.

14.Mou, W., Zhang, J., Tan, X., Liu, S., Zhou, G., & Liu, J. (2018). A review of remote sensing of forest change. Forest Ecology and Management, 422, 62-75.

14.Puletti, N., Calders, K., Burt, A., & Coomes, D. A. (2018). Beyond canopy height: incorporating individual tree structure into large-scale forest modelling with UAV-derived photogrammetric point clouds. Journal of Applied Ecology, 55(1), 281-293.

15.Rango, A., & Laliberte, A. (2010). Semi-automated object-based extraction of shrubs in a desert riparian ecosystem. Journal of Arid Environments, 74(8), 955-964.

16.Wallace, L., Lucieer, A., & Watson, C. (2012). Evaluating tree detection and segmentation routines on very high resolution UAV LiDAR data. IEEE Transactions on Geo science and Remote Sensing, 50(6), 2186-2198.

17.Wallace, L., Lucieer, A., Malenovský, Z., & Turner, D. (2016). Assessment of forest structure using two UAV techniques: A comparison of airborne laser scanning and structure from motion (SfM) point clouds. Forests, 7(3), 62.

18.Wang, L., Sousa, W. P., & Gong, P. (2004). Integration of object-based and pixel-based classification for mapping mangroves with IKONOS imagery. International Journal of Remote Sensing, 25(24), 5655-5668.

19.Zhang, C., Kovacs, J. M., & Flores-Anderson, A. (2018). Remote sensing canopy height estimation and change detection for a mixed temperate forest using UAV Lidar data. Remote Sensing, 10(5), 716.