UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2020

Title: Ground Plane Removal of Drone Images from Orchards

Author: Chiadika Emeruem (EMRCHI001)

Project Abbreviation: PLANER

Supervisor(s): Patrick Marais

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	5
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	15
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	15
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	

Ground Plane Removal from Drone Images of Orchard

Chiadika Emeruem
Department of Computer Science
University of Cape Town
Cape Town, South Africa

ABSTRACT

Remote sensing techniques and in particular, unmanned aerial vehicles (UAVs) are increasingly popular methods of data collection in the maintenance of farmland. UAVs provide absolute height values for each recorded point, therefore digital elevation models (DEMs) created using this data record only the top surfaces of these surveyed areas. One of the applications of the collection of this survey data on farms, is tree height quantification (determining the heights of trees). Theoretically, given height information of an orchard, one can extract tree heights by subtracting the heights of the terrain from those of tree canopies. However, given UAV-sourced DEMs, tree canopies occlude the ground and thus terrain values under canopies are missing. This project and paper focus on using the available terrain height values to determine the missing values, so that the normalised DEMs can be calculated. This paper explores the use of a common interpolation algorithm, the local modified Shephard method and a more novel in-painting technique, Contextual Void Patching, to solve this problem. The in-painting technique was found to under-perform in comparison to Shephard's algorithm on most terrain types. However CVP was better suited to estimating ground values where the slopes are unstable (i.e. the hügelland terrain). Shephard's algorithm produced interpolated regions with small (<1) RMSE values and variance for most terrain types.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**; *Model verification and validation.*

KEYWORDS

Ground plane removal, Tree height quantification, Spatial Data Interpolation, Digital Elevation Models, Inverse Distance Weighting, Contextual Void Patching, Local Modified Shephard

1 INTRODUCTION

The practice of agriculture requires farmers to monitor progress on farms in order to gauge information relating to crop production, soil quality and plantation status[18, 24], and to ensure that they meet greening regulations[2, 20]. One of the metrics used in the monitoring of farmland is tree height, which can be checked periodically. Tree heights can be used to detect, for example, trees with stunted growth which would inform a farmer of erosion/poor nutrient quality/pests/etc. For years, tree heights have been monitored be measuring and logging tree heights manually which, while prone to fewer errors, is time consuming and labour intensive, particularly as trees grow taller. Over the past score years, remote sensing: the use of unmanned aerial vehicles (drones) has become an increasingly popular method for capturing tree heights. The height data is usually captured using Photogrammetry or Light Detection and Ranging (LiDAR)[7]. Photogrammetry applied to this problem, uses aerial photographs from two or more separate vantage points to obtain depth and perspective information. LiDAR uses lasers to measure distances and create a 3-dimensional point cloud from which the height of trees and ground can be extracted. However, these methods are more expensive and data intensive[3]. This project aims to explore the use of simple Digital Elevation Models (DEMs) which are 2-dimensional grids with only one grid value for each grid block, corresponding to the absolute heights of trees and adjacent terrain.

The focus of this project is tree height quantification from simple DEMs of an orchard and this paper focuses on estimating the missing terrain values using interpolation. The approach is to subtract the digital terrain model (of bare-earth terrain) from the DEM, so that what is left is a DEM of the true heights of off-terrain objects[3, 19]. We make a simplifying assumption that all such objects are trees. The DEM resulting from subtracting the bare-earth elevation values is termed the *normalised* DEM. The first step is to classify grid cells as terrain or off-terrain points and extract the bare-earth terrain elevation across the entire DEM. Due to the structure of the DEMs—*i.e.* only one height value for each grid cell—there will be gaps in the resultant DEM, where tree canopies occlude the ground. Therefore, determining the heights of the ground below tree canopies will require filling in said gaps. So the next step is interpolation of the ground plane from the extracted terrain values. Interpolation relies on available values to provide estimations. Thus challenges arise when the tree canopies are clustered together with minimal ground visible between individual trees/rows of trees.

Following interpolation, we subtract the ground plane elevation values from the original DEM to extract the normalised DEM. To enable testing of algorithms used in this process, synthetic DEMs are created and manipulated (*i.e.* trees are added to the DEM and patches of missing terrain values are created). As such, the true heights and elevation values are known and can be compared with our results. This subproject aims to determine the accuracy with which the height of the ground plane can be calculated given a height map occluded by canopy cover. Interpolation accuracy should be consistent across terrain types and results should strongly agree with the actual landscapes.

2 BACKGROUND

2.1 Digital Elevation Models

A digital elevation model (DEM) is a 2-dimensional image where pixel coordinates correspond to coordinates in some model, and the value of each pixel represents the elevation of the model at the corresponding coordinates. They are used to represent an equally spaced grid of height values and are often referred to as *height maps*[13]. The grid structure of a DEM makes them easier to work with and

reduces the need for computationally intensive mesh/triangulation pre-processes[8]. DEMs can contain multiple bands of information. A band is a 2-dimensional grid with the same dimensions as the DEM but with a pixel depth of 1. It can be thought of as a layer or channel of the DEM. Multiple bands are stacked in a single DEM. For example, a 4-band DEM can store elevation values as well as *RGB*-colour values per pixel. In this project, we will be working with single-band DEMs containing only height information.

In this paper, the term digital terrain model (DTM) will be used to refer to a DEM of bare-earth terrain. DEM will refer to models containing off-terrain elevation values (*e.g.* trees) as well as any synthesised DEMs. Although the ground/bare-earth terrain may not always be planer, we use the term Ground Plane Removal to refer to the extraction of the normalised DEM, by subtracting the ground elevation values. The resulting DEM will contain the true heights of off-terrain objects on a plane.

2.2 Interpolation

Interpolation is the term for the process of estimating missing data. Interpolation algorithms use surrounding data in a local or global neighbourhood to determine estimates for the missing data points. There are roughly three categories of interpolation algorithms: regression-based, inverse distance weighting (IDW) and kriging[6]. Kriging is a geostatistical method of estimation that has been proven to be quite effective as an interpolation method (interpolation with kriging gives an RMSE value of 0)[21]. While Kriging is very effective, it is a very slow and computationally intensive algorithms[11]. IDW methods are a class of interpolation algorithms that use a weighted sum of neighbouring points with weights corresponding to the inverse distances from the missing point[1]. Regression-based interpolation methods use regression models to estimate missing data[1]. In this project, interpolation will be used to estimate the bare-earth terrain elevations occluded by tree canopies. Local neighbourhoods will be used in estimations to reduce computation.

2.3 Related Work

Most of the research available on ground plane removal pertains to the urban areas, particularly determining buildings heights[23]. However, unlike orchards, urban settings have distinct boundaries between buildings/ off-terrain objects and the ground. Ground occlusion is also not as prominent or variable a problem as it is in orchards. So these methods are generally not directly applicable to our problem of calculating tree heights in orchards. A lot of the literature uses LiDAR data and constructs surfaces from 3D point clouds[4, 14, 16, 17]. As a result, a lot of explored methods have to do with triangulated irregular networks (TIN) which have a different structure to DEMs[14, 16]. DEM interpolation methods that have been explored with similar applications include interpolation using linear/quadratic trend surfaces or cubic spline interpolation, kriging, linear regression, and inverse distance weighting/citeM21,M22. IDW is usually used on irregularly spaced data, but it produces some of the best results after kriging, on most terrain types[**?**].

3 EXPERIMENT DESIGN

The goal of the Ground Plane Removal subsection of this project is to reconstruct the complete terrain (ground plane) from the scarce captured ground height values. Using interpolation, we fill in missing ground points that were obstructed by tree canopies during the retrieval of orchard DEMs. This paper evaluates and compares the inverse distance weighting method and contextual void patching[22] (CVP), for interpolation.

3.1 Design Framework

Tree height quantification from a single-band DEM requires segmentation for identifying tree pixels and terrain pixels, and interpolation to determine elevation values of missing terrain pixels. Thus our system, has two height extraction sub-modules: Tree Segmentation, and Ground Plane Removal. Due to our provided inputs lacking true height values for comparison, our system also required additional testing data. The DEM Generation module handles the fabrication of test data for the other two modules. The system architecture diagram is presented in figure 1.

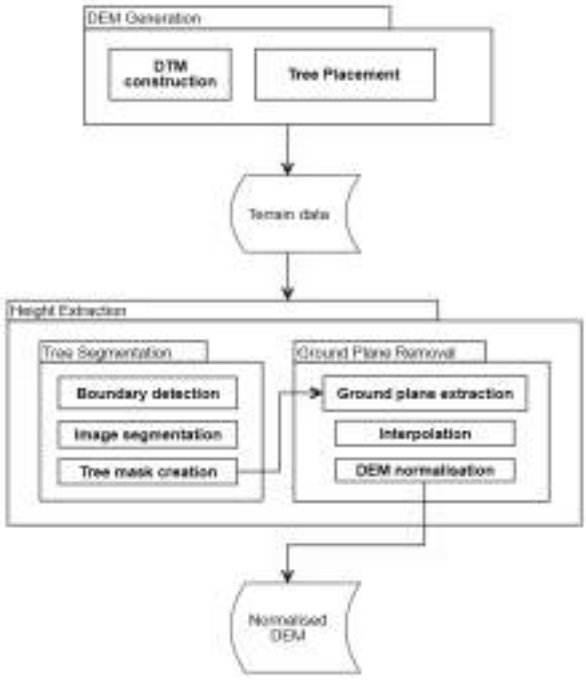

Figure 1: System architecture diagram
The system has a height extraction module with two sub-modules: Tree Segmentation and ***Ground Plane Removal**, and a DEM generation module which provides sample DEMs with which the sub-modules are evaluated.

Each of these modules (and sub-modules) is able to function independently of the rest, provided correctly formatted input data is used.

3.1.1 DEM Generation. The DEM generation module is responsible for DTM creation and tree placement. Tree placement involves adding canopy heights for grids of trees to the DTM, to create

a DEM that mimics height maps to be expected of an orchard. These sample orchard DEMs are saved as greyscale (single-band) tiff images. These, along with the DTM tiffs and binary mask images of the tree canopies, serve as the input into the next module(s).

3.1.2 Tree Segmentation. This submodule is the first stage in the height extraction pipeline. It receives a greyscale tiff image as input and produces segmented images as output. Of these, a binary tree-mask of the same dimensions as the input tiff image is used in the next stage. The non-zero values of the mask image data represent grid positions that contain tree elevation values.

3.1.3 Ground Plane Removal. The Ground Plane Removal module receives as input, a DEM (of an orchard) and a binary mask, indicating which portions of the DEM are to be interpolated over. These are both expected to be in the tiff image format. The mask is applied to the DEM and both are fed to an interpolation algorithm (inverse distance weighting—the modified Shephard[15] algorithm for a radius/neighbourhood around the missing point). The result of this is a DEM estimating the original terrain DTM and is saved as a greyscale tiff image. The interpolated DEM is subtracted from the original to create the normalised DEM, which is also saved as a greyscale tiff image. The normalised DEM should contain only the true heights of the trees. This is the final output of the system.

3.1.4 Development Tools. The program was written in the C++ programming language and compiled using CMake version 2.8 with C++ compiler GNU 7.5.0. The main libraries used were OpenCV version 4.2.0 and LibTiff version 4 for reading of the tiff images and manipulation of the contained data. Statistical analysis was done in R version 4.0.2.

3.2 Datasets

A third party DEM generator was used to create test Tiff images. The ground plane removal project was tested on different terrain types that orchards can be cultivated on: flat or gentle (<20°) slopes, mountain slopes and hills[6, 9, 12]. Specifically, these landscape DTMs (see figure 2) were used: flat (plane), gentle slope (plane), steep slope (plane), hill slope, interlocking spur, and a Hügelland landscape.

A landscape was generated for each terrain type and for each DTM, three masks corresponding to different levels of canopy cover was applied. The canopy levels are: wide canopies, small overlapping canopies and small, spaced canopies. The mask trees were arranged in rows (placed along contours—on slopes, and in grids—on planes and knolls) as they would be in an orchard.

3.3 Device Specifications

- OS: Ubuntu 18.04.5 LTS
- Architecture: x86_64
- Processor: Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz
- RAM: 8GB (7.87 GB usable)
- GPU: Intel(R) HD Graphics 630

3.4 Implementation

The ground plane removal subsystem, uses the c++ libraries opencv and libtiff to read and store the image data in the orchard DEM and tree mask tiff images—both tiff images are expected to have

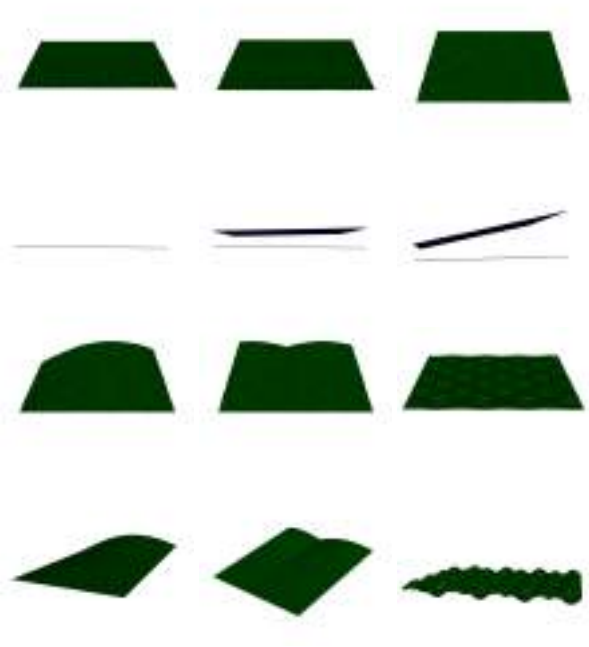

Figure 2: Test landscapes

(*from top left to bottom right*) Oblique aerial perspective of (a) flat terrain, (b) gentle sloping and (c) steep sloping DTM landscapes. Side view of (a), (b) and (c). Oblique perspective of (d) hill slope, (e) spur between two hills and (f) Hügelland DTM landscapes. Alternative oblique view of (d), (e) and (f).

the same dimensions. The mask is converted to a matrix of 8-bit unsigned values, and it is assumed that non-zero values represent the grid positions of tree canopies. To ensure that tree boundary pixels are removed from the orchard DEM, the mask is dilated. The driver applies the mask to the orchard DEM then stretches the masked DEM vertically by a scale of 1000 to pronounce the terrain features.

3.4.1 Inverse Distance Weighting. The inverse distance weighting method used for interpolation in this module is the local modified Shephard method[10, 15]. The algorithm goes through every grid point in the DEM corresponding to a non-zero value in the binary mask. At each missing point u, known points in a neighbourhood N_R of radius R are used to calculate the elevation value at u.

$$\hat{u} = \frac{\sum_{Pj \in N_R} W_{P_j}(u) P_j}{\sum_{P_i \in N_R} W_{P_i}(u)}$$

$$W_{P_i}(u) = \frac{max\,(0, R - d_2(u, P_i))}{R d_2(u, P_i)}$$

In the program, the neighbourhood is retrieved by taking the submatrix of size $2R$ by $2R$ centred at u. To deal with sparse data, one can increase the radius R, to ensure that enough known points are used in the estimation of u. However, that increases computation time and intensity. Points further away will also still have very small weights. I attempted to alleviate this constraint by introducing a threshold variable $\beta \in [0, 1)$ and making the process iterative. Let

the number of known values in N_R be n_{N_R}, $\hat{u}$ will be calculated on this iteration if $\frac{n_{N_R}}{4R^2} > \beta$. Each iteration uses interpolated estimates from previous iterations. This has the effect of filling in the images from the boundaries of the voids to their centres with successive iterations. This method achieved 3–4x speed-up on sparse data, while allowing more flexibility with the radius.

3.4.2 Contextual Void Patching. The three steps of CVP that were used in the programme were (i) identify and extract voids in the DEM along the 4 direction curves, (ii) estimate the missing values along each curve, and (iii) combine the solutions of each curve to get the final interpolated DEM. The curve direction vectors are u=(0,1), v=(1,0), w=(1,1) and t=(-1,1). Below is pseudocode for this algorithm:

```
initialise vector of coordinate pairs for each curve direction
(V_u, V_v, V_w, V_t)
for each curve direction i :
|  for each curve j in direction i on Image :
|  |  k = 0
|  |  while not at the end of the curve :
|  |  |  travel length of the curve until you find a void
|  |  |  k = k + 1;
|  |  |  store the pixel coordinates beg_i jk
|  |  |  continue down the curve until the end of the void
|  |  |  store the pixel coordinates end_i jk
|  |  |  add (beg_i jk, end_i jk) to V_i
|  |  |  end loop
|  |  end loop
|  for each coordinate pair in V_i :
|  |  use Hermite curves to create a smooth patch between
      the pair in direction i
|  |  end loop
|  end loop

for each missing point in Image :
|  combine solution from V_u, V_v, V_w and V_t to get estimate
|  end loop
```

To combine the four direction curves, the inverse distance from the nearest known point in each direction is used to weight the individual curve solutions, which are then summed. Let P_u, P_v, P_w and P_t be the minimum distances from point P to a known point in the subscripted directions. Let S_u, S_v, S_w and S_t be the curve solutions at P's pixel coordinates. The final CVP estimate is obtained as follows:

$$totinv = \frac{1}{P_u} + \frac{1}{P_v} + \frac{1}{P_w} + \frac{1}{P_t}$$

$$P = \frac{S_u}{P_u \times totinv} + \frac{S_v}{P_v \times totinv} + \frac{S_w}{P_w \times totinv} + \frac{S_t}{P_t \times totinv}$$

The interpolated grids in both cases are saved as a greyscale tiff file. The IDW interpolation results is subtracted from the original DEM and the result—the normalised DEM—is saved as a tiff file.

3.5 Evaluation

To evaluate the performance of these algorithms, the system as run using the dataset described in **section 3.2** as input. The interpolated images are evaluated qualitatively, by comparing them against the expected landscape visually. An error map is also generated to show the distribution of the error for each landscape. This can be used to identify patterns in the error as well as areas of concentration[9].

Quantitatively, the following error test statistics are presented and compared for each combination of terrain and canopy type:

- Root mean squared error (RMSE)

$$\sqrt{\frac{1}{N}\sum_{k=1}^{N}(I_k - O_k)^2}$$

  where N is the number of interpolated values, I_k is the kth interpolated value and O_k is value of the corresponding actual landscape pixel. The RMSE gives us an idea of the magnitude of the error from the interpolation estimates. A good algorithm will produce RMSE values close to zero.
- Variance (σ^2) of the error tells us about the distribution/spread of error from the mean and helps determine the confidence of estimates—a smaller variance is preferable. The standard deviation is the square root of the variance.

Finally, Cohen's[5] Kappa statistic is averaged over the canopy types to assess the overall performance of the algorithms on the different terrain types. This is a measure of agreement between the actual and interpolated terrain. It is a real value between -1 and 1, with the value -1 indicating a strong disagreement and 1 indicating strong agreement. A value of 0 suggests a chance agreement *i.e.* success due to randomness.

4 RESULTS

4.1 Qualitative

Tables 1 (*page 5*) and 2 (*page 5*) show interpolation results for the different test DEMs in the same orientations as in figure 2 (*page 3*). Next to those are colour-coded error maps of the absolute differences between the original landscape and the results.

4.2 Statistics

Table 3 (*page 6*) shows the root mean squared error (RMSE) and variance (σ^2) for each terrain and canopy combination of test DEMs. Table 4 (*page 7*) shows the average Kappa statistics for each landscape.

5 ANALYSIS AND DISCUSSION

5.1 Inverse Distance Weighting

Qualitatively the system produces DEMs that closely resemble smoothed versions of the original landscapes in all cases but that of the hügelland. The smoothing in the results contributes to the error. On the gentle slope with small overlapping canopies (shown in table

Table 1: Error maps for Inverse Distance Weighting interpolated terrain

Original Terrain	*Interpolated Landscape*	*Error Map*
Flat:		
Gentle:		
Steep:		
Hill slope:		
Interlocking spur:		
Hügelland:		

Error Key: 

Table 2: Error maps for Contextual Void Patching interpolated terrain

Original Terrain	*Interpolated Landscape*	*Error Map*
Flat:		
Gentle:		
Steep:		
Hill slope:		
Interlocking spur:		
Hügelland:		

Error Key: 

1), the gradual changes in the neighbourhood combined with the smoothing property of the algorithm leads to flatter interpolated patches under each tree.

Hügelland terrain interpolation's RMSE statistic is consistently high, and it is the terrain type with the lowest Kappa statistic (0.3136). The error map of the hügelland terrain shows the error to be concentrated on the peaks of the knolls and at the bottom of the valleys between them. This can be explained looking at the interpolation result. The algorithm flattened the peaks obstructed by trees and levelled out the valleys. There are thin spikes in the interpolated image that hint at the correct elevations but they are few and far apart, thus they do not influence the estimations greatly. The algorithm aggregates values within a neighbourhood to determine estimates. A consequence of this, and especially in the hügelland result, is therefore that in terrains where there are repeated height fluctuations in close proximity, the algorithm will always predict values close to the centre of these fluctuations. This property of the system, does not affect the other more stable slopes.

The interlocking spur is another terrain type with concentrations of higher magnitude error. This error is concentrated along the top of the fold between the slopes, where the change in direction is sharpest. The interpolated image shows that the fold is bridged over in the result. The system visually appears to perform best on the gradual non-planar slopes and worst on the gentle sloping plane and hügelland terrain. However, the statistics reveal that the strongest results are actually on the steep and flat planes, which have the highest Kappa statistics (>0.6) and lowest RMSEs.

5.2 Contextual Void Patching

The results from CVP are generally worse than from IDW—they have the higher RMSE and lower Kappa values. However CVP's statistics for the gentle slope are more consistent over the canopy types and its results on the hügelland terrain are better. CVP uses the gradients at known points in its attempts to estimate underlying curves of the terrain surface. This contributes to the improved accuracy on the hügelland terrain. The error map shows

Table 3: Statistical performance on the test DEMs

Canopy:	Wide		Small, Overlap		Small, Spaced	
Inverse Distance Weighting (IDW)						
Terrain	*RMSE*	σ^2	*RMSE*	σ^2	*RMSE*	σ^2
Flat	0.826	0.668	0.837	0.673	0.818	0.668
Gentle	0.820	0.667	7.247	12.082	0.819	0.670
Steep	0.818	0.668	0.841	0.672	0.817	0.666
Hill slope	0.906	0.684	0.975	0.703	0.936	0.703
Spur slope	1.293	1.658	1.082	1.157	0.914	0.828
Hügelland	6.292	6.288	6.148	6.140	6.148	6.140
Contextual Void Patching (CVP)						
Terrain	*RMSE*	σ^2	*RMSE*	σ^2	*RMSE*	σ^2
Flat	0.930	0.864	0.925	0.855	0.970	0.940
Gentle	1.416	1.732	1.624	2.173	1.133	1.163
Steep	7.535	41.834	9.015	61.315	4.279	12.854
Hill slope	3.749	6.054	2.595	2.780	2.277	2.711
Spur slope	3.096	9.563	2.199	4.836	1.929	3.708
Hügelland	5.957	5.880	5.088	5.083	6.844	6.640

that the concentration of high magnitude errors reduces on the knoll slopes but remains high at the peaks and valleys. Those are areas where the gradient direction changes.

The steep slope results are the worst on CVP unlike IDW where the results were best. Both algorithms have consistently low variances for the flat and hill slope interpolated results.On the gentle and steep slopes, the error appears to increase diagonally (in a north-easterly direction). The algorithm does its estimations in that direction, so the further points use previously visited points and this compounds the error. The error on the hill slope seems to increase in a north-westerly direction instead. This is however, more likely due to the west-facing slope being steeper then the east-facing slope. The error still increases in the north direction for all three.

Both algorithms perform better on wide canopies than small overlapping canopies—spaces between wide canopies tend to be bigger so there is more ground in the neighbourhoods for use in estimation—and have best performance on small spaced canopies.

6 CONCLUSIONS AND FUTURE WORKS

The result accuracy across terrain types for IDW is for the most part consistent with regards to RMSE and even the Kappa statistic. Its performance is adequate for interpolating farm ground, as bumps and height variations on the surface are minimal especially when scaled. The Kappa statistic, while expressing moderate-to-strong agreement for most terrain types, was lower than expected. The performance of IDW on the hügelland terrain is something that needs improvement. The in-painting technique, CVP is not comparable to IDW for most terrain types even though it outperforms it on the hügelland terrain. Its Kappa statistics by terrain are very low magnitudes (close to zero) and the RMSE values are almost all greater than one, and with large variances. A positive note in their favour however, is that the difference between their performances on small spaced canopies and wider canopies was not large for either algorithm.

REFERENCES

[1] Arun, P. V. A comparative analysis of different dem interpolation methods. *The Egyptian Journal of Remote Sensing and Space Science 16*, 2 (2013), 133–139.

[2] Bendig, J., Bolten, A., Bennertz, S., Broscheit, J., Eichfuss, S., and Bareth, G. Estimating biomass of barley using crop surface models (csms) derived from uav-based rgb imaging. *Remote Sensing 6* (10 2014), 10395–10412.

[3] Birdal, A. C., Avdan, U., and Türk, T. Estimating tree heights with images from an unmanned aerial vehicle. *Geomatics, Natural Hazards and Risk 8*, 2 (2017), 1144–1156.

[4] Clark, M. L., Clark, D. B., and Roberts, D. A. Small-footprint lidar estimation of sub-canopy elevation and tree height in a tropical rain forest landscape. *Remote Sensing of Environment 91*, 1 (2004), 68–89.

[5] Cohen, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement 20*, 1 (1960), 37–46.

[6] Deng, M., Fan, Z., Liu, Q., and Gong, J. A hybrid method for interpolating missing data in heterogeneous spatio-temporal datasets. *ISPRS International Journal of Geo-Information 5*, 2 (2016), 13.

[7] Dubayah, R. O., and Drake, J. B. Lidar remote sensing for forestry. *Journal of Forestry 98*, 6 (2000), 44–46.

[8] Duchaineau, M., Wolinsky, M., Sigeti, D. E., Miller, M. C., Aldrich, C., and Mineev-Weinstein, M. B. Roaming terrain: real-time optimally adapting meshes. In *Proceedings. Visualization'97 (Cat. No. 97CB36155)* (1997), IEEE, pp. 81–88.

[9] Erdoğan, S. Modelling the spatial distribution of dem error with geographically weighted regression: An experimental study. *Computers & Geosciences 36*, 1 (2010), 34–43.

[10] Franke, R., and Nielson, G. Smooth interpolation of large sets of scattered data. *International journal for numerical methods in engineering 15*, 11 (1980), 1691–1704.

[11] Gosciewski, D. Reduction of deformations of the digital terrain model by merging interpolation algorithms. *Computers & Geosciences 64* (2014), 61–71.

[12] Hemmati, A., Tabatabaeefar, A., and Rajabipour, A. Comparison of energy flow and economic performance between flat land and sloping land olive orchards. *Energy 61* (2013), 472–478.

[13] Hutchinson, M. F., Gallant, J. C., et al. Digital elevation models and representation of terrain shape. *Terrain analysis: principles and applications* (2000), 29–50.

[14] Maguya, A. S., Junttila, V., and Kauranne, T. Adaptive algorithm for large scale dtm interpolation from lidar data for forestry applications in steep forested terrain. *ISPRS journal of photogrammetry and remote sensing 85* (2013), 74–83.

[15] Shepard, D. S. Computer mapping: The symap interpolation algorithm. In *Spatial statistics and models*. Springer, 1984, pp. 133–145.

[16] St-Onge, B. Estimating individual tree heights of the boreal forest using airborne laser altimetry and digital videography. *Int. Arch. Photogramm. Remote Sens 32*, 3/W14 (1999), 179–184.

[17] Streutker, D. R., and Glenn, N. F. Lidar measurement of sagebrush steppe vegetation heights. *Remote Sensing of Environment 102*, 1-2 (2006), 135–145.

[18] Torres-Sánchez, J., López-Granados, F., Serrano, N., Arquero, O., and Peña, J. M. High-throughput 3-d monitoring of agricultural-tree plantations with unmanned aerial vehicle (uav) technology. *PloS one 10*, 6 (2015), e0130479.

[19] van Iersel, W., Straatsma, M., Addink, E., and Middelkoop, H. Monitoring height and greenness of non-woody floodplain vegetation with uav time series. *ISPRS Journal of Photogrammetry and Remote Sensing 141* (2018), 112–123.

[20] Vierling, K. T., Vierling, L. A., Gould, W. A., Martinuzzi, S., and Clawges, R. M. Lidar: shedding new light on habitat characterization and modeling. *Frontiers in Ecology and the Environment 6*, 2 (2008), 90–98.

[21] Wackernagel, H. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2013.

[22] Wecker, L., Samavati, F., and Gavrilova, M. Contextual void patching for digital elevation models. *The Visual Computer 23*, 9-11 (2007), 881–890.

[23] Weidner, U., and Förstner, W. Towards automatic building extraction from high-resolution digital elevation models. *ISPRS journal of Photogrammetry and Remote Sensing 50*, 4 (1995), 38–49.

[24] Zarco-Tejada, P. J., Diaz-Varela, R., Angileri, V., and Loudjani, P. Tree height quantification using very high resolution imagery acquired from an unmanned aerial vehicle (uav) and automatic 3d photo-reconstruction methods. *European journal of agronomy 55* (2014), 89–99.

Table 4: Average Kappa statistic for each landscape

Terrain type:	Flat	Gentle	Steep	Hill slope	Interlocking spur	Hügelland
IDW:	0.6669	0.4639	0.6713	0.6638	0.6213	0.3136
CVP:	0.6536	0.5312	0.1682	0.2873	0.2054	0.2877

Average terrain elevation value: 16.04808

Range: 15.55701 - 43.96076

Tree height range: -3.814697e-06 - 29.67413

previous range: 16.55701 - 45.23115