UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2020

Title: Clustering-Based Tree Segmentation in Aerial Images of Orchards

Author: Lynolan Moodley

Project Abbreviation: PLANER

Supervisor(s): Prof. Patrick Marais

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	5
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusions	*10*	*20*	20
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*	
Total marks		**80**	

Clustering-Based Tree Segmentation in Aerial Images of Orchards

Lynolan Moodley
Department of Computer Science
University of Cape Town
Cape Town, South Africa
mdllyn007@myuct.ac.za

ABSTRACT

The identification of trees from aerial images is a popular topic. They are useful for quickly gathering data about trees over large distances. Digital Elevation Models (DEMs) contain data about the height of a terrain, therefore, DEMs of orchards are useful for determining the growth rate of trees. This allows farmers to monitor their orchards which would allow them to customise their farming techniques according to their orchards' needs. To identify trees in these DEMs, an image segmentation technique must be used. There are many methods available and all differ in execution time, accuracy and the type of data they are suitable for. This paper investigates the use of two such models: Simple Linear Iterative Clustering (SLIC) and watershed segmentation when used on DEMs of orchards as well as on synthetic DEMs with the aim of understanding their viability when used together. With the use of 34 synthetic DEMs, a mean accuracy (when measured using the union over intersection calculation) of 88.16% per tree was obtained in one of the easy case DEM categories. Performance of the system ultimately depended on factors of the DEMs, such as noise, proximity of the trees to each other and the gradient of the ground.

CCS CONCEPTS

• Computing methodologies → Computer graphics → Image manipulation → Image processing

KEYWORDS

Tree segmentation, image segmentation, SLIC, watershed, DEM, orchard, OpenCV

1 Introduction

Humanity is dependent on the agricultural industry for survival [36]. It is vital that food security is improved upon. One of the ways to achieve this would be to increase the yield of existing farms. This can be accomplished by allowing farmers to customise their farming techniques according to tree health information, such as tree height (which can be used to gather information about a tree's mass and its fruit yield [41]), gathered from their orchards [23]. Many of these methods require trees to be identified from aerial imagery. In this project, the imagery used was Digital Elevation Models (DEMs), which are height maps of a terrain. This height data is obtained from Light Detection and Ranging (LiDAR) systems [15]. LiDAR sensors are mounted onto drones. As they fly over terrain, they emit a beam of light into the ground and measure the time it takes for the beam to bounce back to the drone. By doing this, they can indirectly measure the height of terrain.

LiDAR sensors are expensive [9]. Cheap, pseudo-LiDAR techniques, which aim to extract terrain height data from low-resolution visible spectrum images, are not very accurate [40]. Photographic images from cheaper systems are often not usable due to their low-resolution. Therefore, to allow farmers to measure tree height (and other metrics) cheaply, methods must be developed to allow for tree data extraction from DEMs that do not have a very high depth precision. This paper aims to investigate image segmentation techniques – Simple Linear Iterative Clustering (SLIC) and watershed segmentation – that can be used to identify trees in DEMs of orchards as well as in synthetic DEMs, without the use of photographic (RGB) data. This does pose a challenge as it reduces the amount of data we can work with. A secondary aim is to compare the performance of SLIC with that of the watershed algorithm when used in this context. Metrics such as execution time and Intersection Over Union (IOU) values were obtained. The results produced by this project, the location information of trees in DEM, can used in other projects to further extract information from DEMs, such as tree height data.

The segmentation worked well on flat terrains, obtaining a mean IOU per tree of 88.16%. It also performs better in cases where trees have no noise, therefore, noise and ground gradient are significant factors which affects segmentation accuracy. The other important factor is the proximity of the trees to each other. Due to these factors, the segmentation system does not work as well in DEMs that have uneven terrain and very noisy, compacted trees,

This paper discusses background information, and related work, on the concepts involved in this project in Section 2. In Section 3, the design and implementation of the solution developed will be discussed. Section 4 contains methodology about the experiment design and execution. Section 5 presents the results obtained from the experiments in Section 4 and also discusses findings. The limitations of the project are discussed in Section 6, while the ethical, professional and legal issues are covered in Section 7. Finally, Section 8 concludes the paper and offers insight about future work.

2 Background and Related Work

This section presents background information regarding concepts used in this project.

2.1 DEMs

Digital Elevation Models (DEMs) are raster images that contain the height data of some terrain [43]. They are single band images where the value of each pixel represents a height of a piece of land. This height data can be collected by LiDAR sensors [15], which work by emitting light at a target and measuring the time it takes to return. This data is then used to indirectly measure distance between the target and sensor. Synthetic data can also be produced for DEMs [28]. In this project, both DEMs of real terrain and synthetic DEMs are considered.

2.2 Filters

Filters are operations that can be used to manipulate images. They are useful for extracting certain areas from images or even for emphasising certain parts of images.

Blurring filters are useful for removing noise from images. As noise can negatively impact segmentation [26], this is a useful operation in image segmentation. A simple blurring technique would be to assign pixels the value of the average of its neighbourhood. Gaussian blur, a more sophisticated method, is also useful for smoothing images [17]. It uses a matrix to scale pixels in a neighbour. Bilateral blurring is useful for cases where one would want to smooth an image, but leave the edges sharp [29], which would be useful for edge detection.

Threshold functions are useful for generating a mask image, a binary image, based on some limit [5]. If a pixel is less the limit, it is set to 0 and if it is equal to or greater than the limit, it is set to 1. Adaptive thresholding techniques exist, which can locally threshold an image by automatically calculating the local limit, as investigated by Xu et al. [42], however, such a technique has difficulties when two image features with variances that are very different, lie next to each other. As the height of the trees used in this project are variable, this approach was not used.

2.3 Simple Linear Iterative Clustering (SLIC)

Superpixels are a collection of similar, neighbouring pixels [2]. *Clusters,* in this project, refers to a collection of pixels. Simple Linear Iterative Clustering (SLIC) is a way of grouping pixels into superpixels [1], by using high-dimensional space. To do this, it requires the number of clusters (superpixels), *k*, required in an image. The centres, *centroids*, of the clusters are spread out at regular intervals throughout the image. The distance between each pixel and centroid is then calculated. Distance, in this case, refers to the Euclidean distance when considering all the dimensions of a pixel, such as the RGBA bands and x, y coordinates. As these values may have different scales, normalisation is required before SLIC processing is performed. To improve performance, the Euclidean distance calculation is normally only performed between pixels and centroids that lie in the same neighbourhood, often defined as a window of n pixels wide. Once all pixels have been assigned to their nearest centroid, the clusters have been formed. The centre of each cluster is then calculated, by averaging all the pixels in a cluster – again, considering all dimensions. The centroid is moved to the location of the newly calculated cluster centre. Pixels are then assigned to centroids again and the process repeats until the centroids no longer move when the cluster centre is recalculated. Variables can be introduced to control the size of the superpixels [31]. This processes of calculating new clusters improves the segmentation, but this can take a long time to occur, so the algorithm is often limited in the number of iterations it can run, as investigated by Ren et al [31]. This greatly reduces execution time and is used in this project. A decision must be made to achieve the optimal balance between accuracy and execution time.

In OpenCV, the library that was used to execute the SLIC function, there is an option that allows for hexagonal initial cluster shapes [13]. This is an improvement over square shaped clusters as it means that clusters would be able to fit the round shape of the trees, when the algorithm is executing its first few iterations, better than square-shaped clusters. OpenCV also automatically calculates the number of clusters to use, and instead, requests a cluster size as input.

In an investigation by Han et al. [20], they found that SLIC, due to its finite number of superpixels, often under-segments images. Their solution was to use another K-means iteration in each segment to improve segmentation accuracy. This idea was used here, by using watershed segmentation to further segment the clusters produced by SLIC.

2.4 Watershed Segmentation

The watershed algorithm is a technique, in image segmentation, that uses the concept of *catchment basins* [12], which are low-lying areas that are noted for their ability to collect precipitation, to segment images [11]. *Watersheds* are high-lying regions which separate catchment basins. In an image, it is up to the user to defining what constitutes as low- or high-lying regions [30]. To process the image, a "flood" is performed at predetermined points (there is usually one point in every catchment basin). As the basins fill, the pixels that are covered by the flood are classified as being part of the catchment basin. If two basins meet, the pixels along their point of contact are considered. If they are very similar to the pixels found in both basins, then the basins will merge, forming a single, larger basin. If the pixels at the meeting point are not similar to the pixels in the basin, then a boundary (watershed) is formed at the meeting point, separating the two basins [6]. After the flooding has been completed, all pixels that have not been flooded are classified as watershed lines. It is these watershed lines that mark the boundaries of the different segments the algorithm has identified in the image.

Watershed segmentation is well suited for problems that deal with ground height, as the user can very clearly define what constitutes as a low- and high-lying region in a heightmap image. However, noise in these images can lead to inaccurate segmentations. A common issue is over-segmentation, where multiple, small basins are identified in an area where a single, large basin should be. Markers are a way of overcoming this problem, as they allow the algorithm to identify the significant basins in the image [11]. Fig. 1(a) shows a function with many local minima, while Fig. 1(b) shows the same function with its significant local minima (markers) identified.

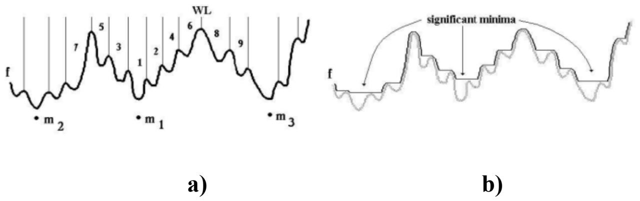


Figure 1: A function showing watershed lines in an unprocessed function (a) and significant minima (markers) identified in the function (b) [10].

Gies et al. described abnormal segmentations, with no proposed solution, that can occur when noise is present at the edge of an image segment [19] and similar phenomena were observed during this project. Finding a way to highlight the areas of interest is important when using watershed, as demonstrated by Frucci et al. [18] as this allows the watershed algorithm to be more contained as it will be able to identify edges more easily, prevent it from spreading into areas it should not consider.

2.5 Region Growing

Region growing is an image segmentation method based on seed points [4] and the flood-fill technique, where an action is performed on a pixel and recursively on all of the pixel's neighbours until some terminating condition is met [27]. Region growing attempts to identify connected pixels of similar value in an area. This was considered for this project. As the algorithm expanded from its seed point, it would calculate the difference in height (based on the values of pixels) from the seed pixel to the current pixel. If the height is greater than some pre-determined limit, then that pixel represents the edge of the tree. The issue with this approach is that the height of the trees can vary. A height limit may suit one tree very well, but not its shorter (or taller), neighbouring tree. This was experienced with some of the DEMs and, as a result, we decided not to investigate this method further.

3 Framework Design and Development

This section present information about the system developed and the techniques used to process DEMs.

3.1 Design

An agile approach was adopted when designing and developing the system. This was beneficial, as it allowed us to quickly change course when we felt the system need significant changes, a useful aspect of agile, given the short time frame we had to complete the project. Watershed was meant to be the sole segmentation means, however, it was very sensitive to noise. SLIC performed much better in this regard, but SLIC has a longer execution time. Therefore, both were investigated to determine the improvement the combination of SLIC and watershed offers. An investigation was also done to determine the difference in execution time and performance between SLIC and watershed.

There are 3 parts of the system: pre-processing, segmentation and evaluation, seen in Fig. 2. Pre-processing is where DEMs are formatted to the form required by the segmentation part of the system. Markers are also extracted (for watershed) and filters are applied. Segmentation is where watershed segmentation takes place. Evaluation tests the results of the system, by comparing the tree masks outputted by the system against the ground truths. The UI is minimal, taking place through terminal and input and output are done through TIFF images.

3.2 Development Tools

C++ was chosen for this project, due to its robust and fast nature [21]. The OpenCV library was used to implement many of the image processing techniques. Visual Studio Code was used as the IDE and CMake was used to build the project. For debugging, the GNU Project Debugger and Valgrind were used. Git was used for version control. Doxygen was used for documentation.

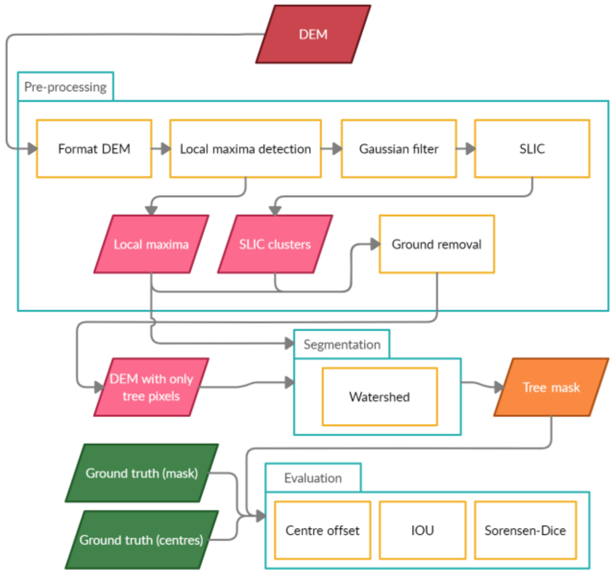


Figure 2: System diagram including the pre-processing, segmentation and evaluation frameworks.

3.3 Implementation

The main parts of the system, pre-processing and segmentation, can be reduced to the pseudocode in Algorithm 1.

Algorithm 1: Segmentation

```
Input: DEM
Output: A tree mask
1.  Set null values in the DEM to 0
2.  LocalMaxima := FindLocalMaxima ( DEM )
3.  Apply Gaussian filter to DEM
```

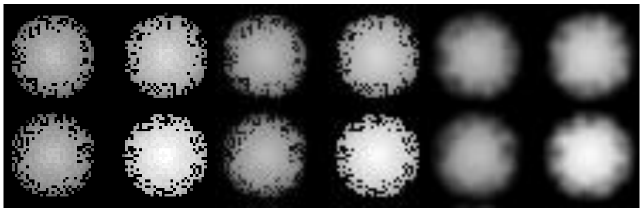
```
Perform SLIC operation on DEM to get clusters
FOR every cluster:
    IF cluster contains a local maximum:
        Set cluster to tree
    END IF
END FOR
Calculate the average variance of all tree clusters in the DEM
FOR every cluster in the DEM that is NOT a tree:
    IF cluster has a neighbour that is a tree AND cluster has a variance of at least 90% of the average variance:
        Set cluster to tree
            Find local maximum of cluster and add it to LocalMaxima
    END IF
END FOR
SLICMask: = DEM
FOR every cluster in the DEM:
IF cluster is a tree:
    Set all cluster pixels in SLICMask to white
ELSE:
    Set all cluster pixels in SLICMask to black
END ELSE
DEM := bitwise-and ( DEM , SLICMask )
Assign every point in LocalMaxima a unique value
WatershedMask := watershed ( DEM , LocalMaxima )
RETURN WatershedMask
```

SLIC was used to segment the trees from the ground, which follows an approach by Frucci et al. [18] to highlight the areas of interest in an image. Once SLIC has determined which clusters are trees (which are cluster with a significant local maximum, discussed below, present), the system will classify all other clusters as being part of the ground (and will set those pixel values to 0). This allows the watershed algorithm to focus only on the tree clusters, providing a sharper edge for the tree clusters, thus improving the chance that the watershed algorithm will detect the edge of a tree.

Distance transform [16], an algorithm which assigns every pixel a value according to its distance to the nearest black pixel (usually the background of the image), is a popular way of determining markers for the watershed algorithm [3]. However, this proved difficult to implement as the DEMs were not guaranteed to have any black pixels (the ground part of the DEM could consist entire of non-black pixels). Instead, we assumed that the centre of a tree would also be its highest point. We then used a significant local maxima detection algorithm to find local maxima in the image, to pose as markers for the watershed algorithm. This works by expanding the high areas in the image, to emphasise the high points, and then using a window to traverse through the image. The highest point in the window is chosen to be the local maximum [39].

To identify trees that have been missed by the local maxima detection algorithm, a variance check was done on all non-tree SLIC clusters [32]. The average variance of all tree clusters was calculated, where the variance of a cluster is the variance of all pixel values in that cluster [35]. Each cluster was then checked. If the variance of that cluster was at least 10% (this value was derived after several tests and is small to cater for the small, shallow trees of the orchard DEMs) of the average variance, then that cluster would be reclassified as a tree. A neighbour check was also done to ensure that only clusters neighbouring trees were considered as we made the assumption that trees in orchards form a grid and would, therefore, have tree neighbours. This prevented incorrectly classifying noise (or other obstacles) as trees. This neighbour checking algorithm was done in place of a row finding algorithm [38], which was too time consuming to implement accurately with the time constrains we were under.

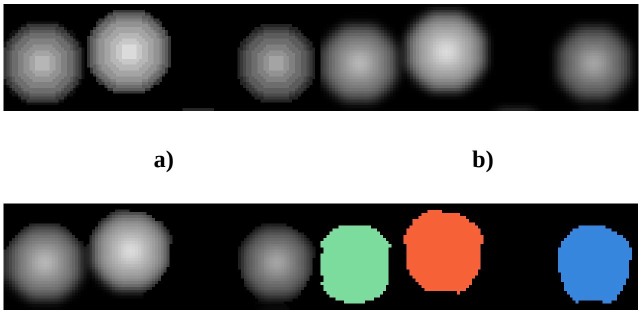
One of the other decisions that had to be made was determining the appropriate filter to use to alleviate noise. Even though a bilateral filter is useful in cases of edge detection, we decided to use a gaussian filter instead. The noise at the edges of trees, which are essentially gaps through which the ground can be seen, meant that the bilateral filter would leave these intact, which would cause the watershed algorithm to produce abnormal segments, as discussed by Gies et al. [19]. The Gaussian filter was better at smoothing these artefacts, which can be seen in Fig. 3.



a) b) c)

Figure 3: A series of images displaying a sample DEM (a), the DEM after a bilateral filter has been applied (b) and the DEM after a Gaussian filter has been applied (c).

The results from each of the significant operations can be seen in Fig. 4 below.



a) b)

c) d)

Figure 4: An illustration of the segmentation process, showing the original DEM (a), the DEM after a Gaussian filter has been applied (b), the DEM after a SLIC mask has been applied (c) and the trees identified from the DEM (d).

4 Experiment Design and Execution

This section presents information about the experiment regarding computer hardware used, data used and the evaluation methods used to test the tree masks produced by the system.

4.1 Hardware and Environment

The experiments were carried out on a virtual machine running Ubuntu 20.04. The virtual machine was assigned 4 Intel Core i7-6700HQ CPU cores and 8GB DDR4 RAM. None of the techniques used made use of GPU processing, therefore, the dedicated GPU was not required.

4.2 Data

Two DEM datasets were considering in this experiment. The first of which was provided by Aerobotics. These were single-layer, greyscale TIFF images of orchards which consisted of 32-bit float pixel values. Using the GeoTIFF standard, they contained location metadata, which allowed us to establish boundaries in order to crop out areas in the DEM that were not of interest. An example can be seen in Fig. 5.

As this is real data, measured using LiDAR, it can be prone to error [7]. Abnormal artefacts can be observed in the DEM, that would not exist on an ordinary photograph of the same terrain. The height resolution can be poor as well, which is one of the side-effects of using cheaper LiDAR equipment, or pseudo-LiDAR techniques [40]. This can be observed in Fig. A1 and Fig. A2, Appendix A, where the height profile of samples of the orchard DEM are displayed. With an orchard tree, we expect the curve to form a dome shape at the canopy of the tree, with rapid drops to either side. The ground between trees should also be flat. However, from the images in Fig. A2(b.ii & c.ii), we see that the canopy part of the trees is very steep, almost forming a sinusoidal shape. The profile of Fig. A1(a.i) has a very irregular shape, which the maximum height of trees differing greatly from each other. These factors negatively affected the results of the segmentation system.

Figure 5: A DEM of an orchard, where light areas represent high lying regions and dark areas represent low lying regions.

The second set of DEMs was provided by a related project. This synthetic data, which was also 32-bit float, single-layer greyscale images, was used test our algorithm against a wide variety of terrains and it allowed for the tuning of parameters in the system. From the height profiles of the images in Fig. A3, Appendix A, it can be seen that these DEMs follow more closely the expected shape of orchard trees. The images could be broken up into five categories: flat terrain, gentle gradient terrain, steep gradient terrain, large-hilled terrain and small-hilled terrain. Within each category, there were three subcategories: smooth trees; noisy, dispersed trees; noisy, compacted trees.

There were several properties of the DEMs that had to be considered. One of which was the space between trees. *Spread* DEMs had a clear space in between trees. The gradient of the ground was important as well. The *flat, gentle,* and *steep* DEMs had flat, gentle and steep gradients respectively. The *largeHill* DEMs represented a large rounded hill. The *smallHill* DEMs represented terrain with many little hills spread throughout the land. Noise was another factor. *Smooth* DEMs had trees with no noise (their canopies were a smooth curve), while images without the smooth indicator had noise in their trees.

Some additional challenges are presented with the Aerobotics data, as there are some unknown obstacles in the image, such as a non-orchard tree.

4.3 Evaluation

Evaluating the images produced by this system required a *ground truth*, which is a set of measures that is more accurate than the measurements produced by the system being tested. In this project, two such ground truths, created manually, were considered: tree masks (an image with all tree pixels set to white and the rest set to black) and tree centres (an image where only the pixel representing a tree centre is set to white, while other pixels are set to black).

The first evaluation done was to determine the accuracy of the tree centre detection. This was vital as the tree centres were used as markers for the watershed algorithm. A ground truth image contains the actual location of tree centres. For every local maximum in the ground truth, the Euclidean distance [14] to every predicted local maximum was calculated, and the closest local maximum was regarded as being that tree's centre. If, however, the distance is greater than the radius of a tree, then that indicated that the algorithm had failed to detect the local maximum of that tree.

The second test performed, which aimed to determine the ratio between the number of correctly identified trees and the number of incorrectly identified trees, obtained the Sorensen-Dice coefficient (a number from 0 to 1) for every DEM [8]. This involved comparing the tree mask calculated and information about the tree clusters produced by the system with the ground truth centres. If a tree centre matched with a white pixel on the mask, then that tree had been correctly identified (a true positive). If the centre did not match with the mask, then that indicated that the tree was not identified (a false negative). If a tree cluster did not contain a centre, then that indicated that the system had identified a tree that did not exist (a false positive).

The third test performed was an Intersection Over Union (IOU) calculation [33], which determined how closely the calculated tree mask fitted the ground truth mask, both globally and per tree. In this project, to determine the IOU per tree, we used bounding boxes [24] to estimate the extent of the actual tree and the extent of the calculated tree. The minimum and maximum coordinates of each tree was determined, and this was used to generate a box to represent each tree (both for ground truth and calculated). The area of the overlapping region of the two boxes, along with the total area of each box, was then used to calculate IOU. A separate calculation, to obtain the overall IOU, had to be done as the IOU per tree figure would not account for false positive tree identifications. This involved comparing the ground truth mask with the calculated mask. The number of matching pixels were then divided by the number of non-matching pixels to produce the IOU.

Qualitative evaluation was also required to assess the shapes of the tree masks and distribution of errors. This was also required to inspect the non-orchard tree region of the orchard DEM. Additionally, as no ground truth was provided for the orchard images (and it was difficult to manually draw a ground truth mask due to the low resolution of the DEMs), the orchard DEMs relied primarily on visual inspection to determine the performance of the system when used on those images.

Lastly, execution time of SLIC, watershed and the system as a whole were also measured to determine their performance.

5 Results and Discussion

This section presents the results of the tests described in Section 4.3 and discusses the findings from these results.

5.1 System Tuning

Some of the algorithms used in the system required adjustments to their parameters. This allowed them to work more accurately. Optimal parameter values were chosen based on logical decisions, such as setting the initial cluster size of the SLIC function to be the diameter of a tree, and then using trial and error to determine if better parameter values could be obtained. The same was done with the local maxima detection algorithm. This can be seen in Fig. 6, where the local IOU per tree is plotted according to differing SLIC cluster and local maxima window sizes. SLIC performs better when its cluster size is higher than 27 pixels, while the local maxima algorithm performs better when its neighbourhood size is less than 25 pixels. This makes sense considering what the aim of each algorithm is. SLIC is not responsible for separating trees from each other, therefore, it can have large sizes, which may contain more than one tree. The local maxima algorithm is responsible is locating the centre of trees. If its window size is too large, it may miss out trees entirely.

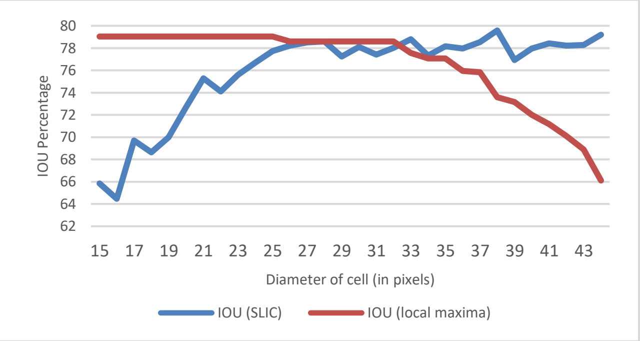


Figure 6: A graph showing the average IOU percentages obtained for different cluster sizes in the SLIC and different window sizes in the local maxima detection algorithm for the flat terrain.

The next parameter that required tuning was the number of iterations that the SLIC operation ran for. The higher the number of iterations SLIC can run for, the higher the accuracy of its segmentation. We could have let SLIC run to completion, i.e. run it until the centroids no longer move between iterations, but that could be very time consuming. Instead, we ran SLIC many times while changing the number of times it iterated for, to determine the optimal iteration number. The results of this can be seen in Fig. 7, where the IOU percentage and execution time are plotted for different iteration amounts of SLIC. Execution time has a linear relationship, with it increasing steadily as the number of iterations increases. IOU, however, experiences a sharp increase at the beginning, before plateauing. This indicates that the SLIC algorithm does produce an accurate result early on. There is only a 1.4% increase in IOU when increasing the number of iterations from 3 to 20, while the execution time increases by 360.48% during this same period. Therefore, 3 iterations seem like the optimal value, however, to cater for edge cases, we choose 6, allowing for increased accuracy at a reasonable execution time.

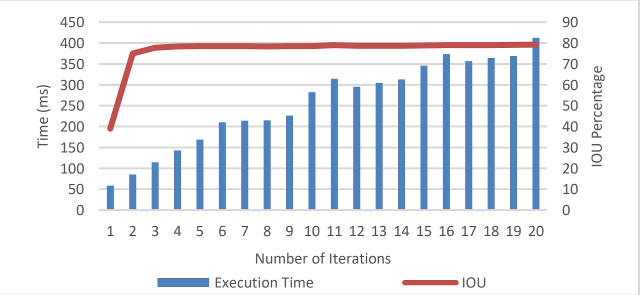


Figure 7: A graph showing the IOU percentages obtained and time taken to complete segmentation for different SLIC iteration amounts for the flat terrain.

5.2 Marker Detection

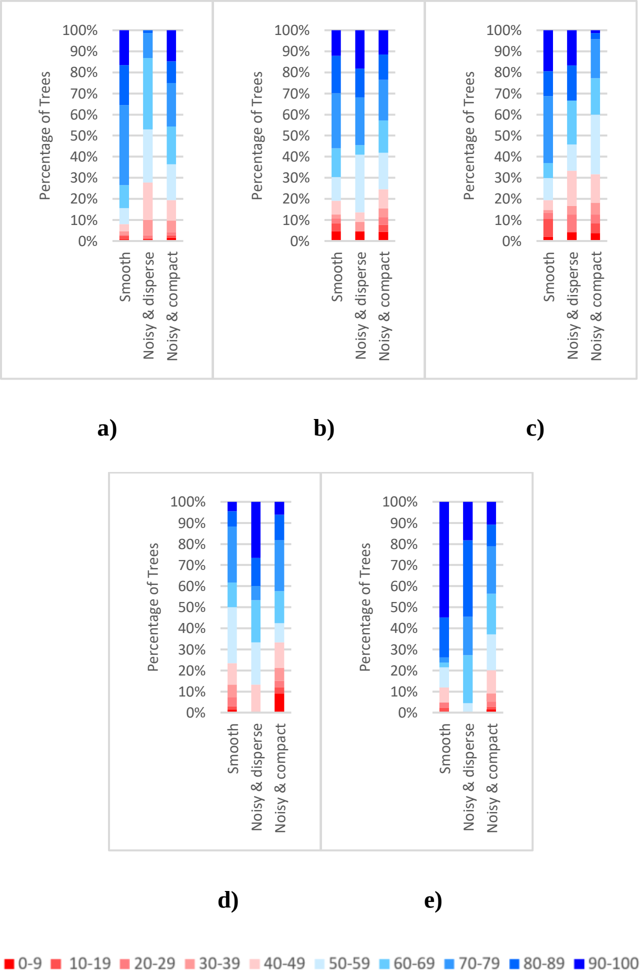
As watershed relies on the local maxima detection algorithm to identify tree centres, it is important to test the local maxima detection as well. Table 1 contains the results of these tests. It can be seen that the algorithm performs best on flat terrain, with 98.312% of tree centres identified and a calculated mean distance of only 0.308 pixels from the ground truth centre. More detailed results are displayed in Table B1, Appendix B. It can be seen that the spread terrains are more likely to have all trees identified. This would be due to the size of the local maxima window. As the trees are spread out, there is only 1 tree present in the window at a time as it traverses the image. In non-spread images, where trees can occlude each other, hence their centres will be closer to each other, there is a chance of 2 trees ending up in the local maxima window at the same time. In this case, the algorithm will only identify the taller tree and will ignore the shorter one. This leads to trees being unidentified. It should also be noted, however, that the spread terrains do not always have low mean distances, such as steep_spread, which has a mean distance of 0.726 pixels and a standard deviation of 0.483. As the trees are spread out, they can often be cut out by the local maxima window, i.e. part of the tree lies outside the window. When this occurs, the centre of the tree (and the tree's highest point) may be excluded from the local maximum calculation, and the calculated local maximum would be that of another high point of the tree. However, considering that, in the DEMs, the centre of a tree may not necessarily be the highest point, the local maxima detection algorithm does detect the centres adequately as, in most cases, it is off by an average of less than a pixel.

Table 1: A table displaying the mean Euclidean distance of the calculated centres from the actual centres for the different terrains.

Terrain	Total Trees	% of Trees Identified	Mean Distance from Centre (in pixels)	Standard Deviation
Flat	711	98.312	0.308	0.325
Gentle	530	95.849	0.350	0.440
Steep	540	95.926	0.376	0.430
Large hills	466	82.403	1.257	2.366
Small hills	568	94.014	0.403	0.479

5.3 Segmentation

To determine the segmentation accuracy per tree, the IOU value per tree was obtained. In Fig. 8, the stacked graphs are used to display the percentage of trees that obtained a certain IOU percentage. We see that for the flat, gentle and steep terrains, the smooth subcategory has more IOUs with a value of 70% or more, than the other subcategories. This is because the trees in these terrains are smooth. As the watershed algorithm is sensitive to noise, the smooth terrains pose less of a challenge. It is also clear that the gradient of the ground affects the accuracy of segmentation as the DEMs with shallower gradients have a better accuracy than DEMs with steeper gradients. Except for the flat terrain, the noisy and disperse subcategory have more trees with an IOU of at least 60%, than the noisy and compact subcategory. The spread-out nature of the trees in the noisy and disperse subcategory means that there is a greater chance that only one tree is present in a SLIC tree cluster, thus ensuring that tree masks fit tighter. The noisy and compact subcategory has the worse segmentation, as expected. The lack of space between trees increases the chance of under-segmentation. Due to the close nature of the trees, the gradient in-between trees can be shallow (as seen in Fig. A3(c.ii), Appendix A). This means that the algorithm will find it difficult to determine the boundary between two trees. The noise factor leads to abnormal segments, which reduces IOU which can be seen, as non-smooth DEMs have a lower accuracy than smooth DEMs. An unusual phenomenon can be seen in Fig. B1(c), Appendix B, where some of the tree segments have a square shape. This is a side effect of the Gaussian filter, which often leaves images grid-like. The watershed segmentation follows these patterns, which then produces a square tree segment.



a) b) c)

d) e)

Figure 8: A series of graphs showing the percentages of trees that obtained a certain IOU percentage (shown in the legend), when using different subcategories of terrains in the flat terrain (a), gentle terrain (b), steep terrain (c), large-hilled terrain (d) and small-hilled terrain (e).

Although Fig. 8 displays data per tree, it is also important to gather a high-level view of the segmentation. Table 2 displays the mean Sorensen-Dice coefficient and mean overall IOU of DEMs. The small-hilled category provides a good example of why an overview is important. Its noisy and disperse subcategory has a good mean IOU per tree of 80.2%, but a poor overall IOU of 24.3%. This is due to false positives in the image, which would not affect per tree calculations, but would affect the overall IOU. Fig. B1(e), Appendix B is an example of an image with many false positive identifications. Cases with high mean Sorensen-Dice values, but low mean overall IOUs or low mean per tree IOUs indicate that the system had successfully identified the location of most trees, but had trouble determining their boundaries. This occurs frequently in the smooth subcategory where the gentle terrain, for example, has a mean Sorensen-Dice of 0.987 but a mean overall IOU of only 75.94%. This is counterintuitive, as we expected noise to have a greater negative impact on segmentation than proximity. The images here, as they can be dispersed or compacted, have watershed tree clusters with more than one tree. The clusters often include ground space, which decreases overall IOU, as seen in Fig. B1, Appendix B, where the tree masks have abnormal shapes.

The accuracy of the segmentation of the flat, smooth DEMs is comparable to that of Martins et al. [25] (as they have achieved accuracies of 87.50% to 89.01%) although, they made use of CNNs with their SLIC implementation, without using watershed.

Table 2: A table displaying the mean Sorensen-Dice coefficients, mean overall IOU, mean IOU per tree (and standard deviation) for different terrains.

Terrain	Mean Sorensen-Dice	Mean Overall IOU (%)	Mean IOU Per Tree (%)	Standard Deviation (%)
a) Flat				
Smooth	1	87.52	88.16	18.28
Noisy, disperse	0.6577	65.43	69.8	17.09
Noisy, compact	0.7197	67.41	63.45	17.41
b) Gentle				
Smooth	0.987	75.94	72.07	25.44
Noisy, disperse	0.597	60.63	68.42	23.81
Noisy, compact	0.6973	58.6	61.59	20.97
c) Steep				
Smooth	0.9701	68.55	69.95	28.17
Noisy, disperse	0.5	56.61	62.38	24.71
Noisy, compact	0.718	58.03	57.21	21.76
d) Large hill				
Smooth	0.9262	74.22	55.22	26.61
Noisy, disperse	0.6617	55.11	69.76	18.53
Noisy, compact	0.7277	62.69	54.3	22.38
e) Small hill				
Smooth	0.831	57.08	75.81	22.14
Noisy, disperse	0.4941	24.3	80.2	11.77
Noisy, compact	0.6239	42.6	61.76	19.1

A visual inspection of the orchard DEM was performed. In Fig. 9, a sample of the orchard DEM with a non-orchard tree can be seen (the light area in Fig. 9(a)). The system had classified most of this region as a tree. The use of the local maxima detection meant that the high points in the non-orchard tree were found. Improvements to remove false positives should be made.

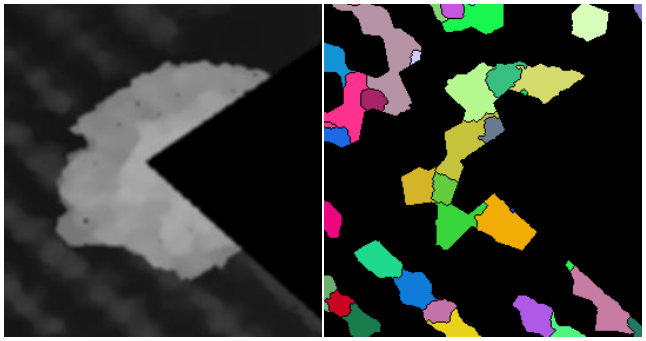


Figure 9: A series of images showing the original DEM (a) and tree mask produced by the system, where coloured regions represent a tree.

In Fig. 10, there is a sample of the orchard DEM and its calculated tree mask. There several gaps present in the mask and it is obvious, by looking at the DEM, that there should be a tree present there. The close proximity of the trees here presented an issue. The local maxima detection algorithm failed to detect trees as there were multiple trees present in its window at a time. The variance check has also failed to detect that these are trees. It is unlikely that their variance is very different from neighbouring trees. There are, in fact, mini clusters in between trees, that increased the neighbour count, which prevented the clusters from being reclassified as trees. The noise in the DEM (the areas around trees which extend into the ground space, which leads to trees from different rows touching each other) leads to the formation of these mini clusters. We also see cases where tree clusters, from two trees, extend past a middle tree to meet each other. Again, this is due to the noise in the DEM.

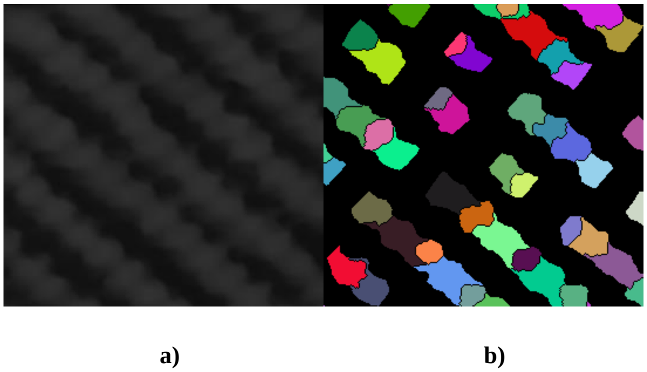


Figure 10: A series of images showing the original DEM (a) and tree mask produced by the system, where coloured regions represent a tree.

The following images, Fig. 11, show an area of the DEM with a very low-lying tree (with no neighbours), present where there is a dark region in Fig. 11(a). The system has successfully identified this tree with the local maxima detection (the variance check would not work here as the tree has no tree neighbours). We also see a case where a tree cluster is complete encircled by another cluster. The trees in that region have very shallow slopes, with no sharp drop-offs at their edges. This makes it difficult for the watershed algorithm to detect the edge of the tree. Thus, when being calculated, it expands until it meets and engulf another tree rather than stopping at the edge of the first tree.

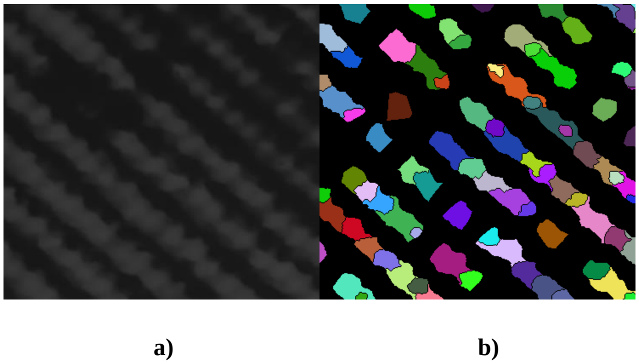


Figure 11: A series of images showing the original DEM (a) and tree mask produced by the system, where coloured regions represent a tree.

5.4 SLIC versus Watershed

As two segmentation techniques were used, it is important to determine the cost and impact on segmentation accuracy they each had. Fig. 12 shows a series of graphs that display the percentage of trees that had obtained a certain IOU percentage for different terrains. From the graphs, it can be seen that, overall, using a combination of SLIC and watershed does yield more matches with an IOU of 50% or more for the flat and gentle terrain. For the small-hilled terrain, it is about even, however, the watershed-only method has fewer trees in the 70-100% IOU ranges. Watershed is more sensitive to noise than SLIC. The noise at the edges of the trees presents a more difficult medium and results in abnormal segmentations where the watershed extends into the ground region. Using SLIC alone resulted in more matches with an IOU of 90% and higher. This is as a result of the lack of abnormal segmentations produced by the watershed algorithm. However, SLIC, tends to under-segment, as trees are either too close to each other or have too similar height profiles. This leads to SLIC clusters containing more than 1 tree, decreasing its accuracy.

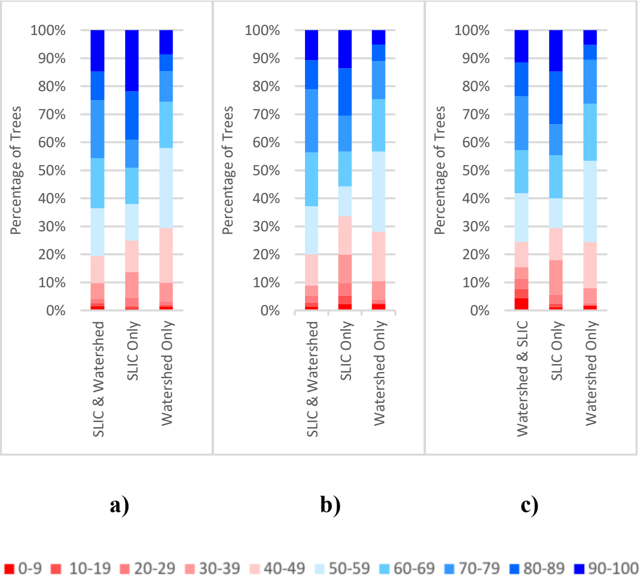


Figure 12: A series of graphs showing the percentages of trees that obtained a certain IOU percentage (shown in the legend), when using a certain segmentation technique, for the flat terrain (a), gentle terrain (b) and small hill terrain (c).

The execution time of the different segmentation techniques was also important. In Fig. 13, the average execution time of SLIC, watershed and other processes (pre-processing functions) are displayed for different terrains. From this, it can be seen that the pre-processing functions take up the most time in an execution, followed by SLIC. In the flat terrain, the pre-processing functions take 399.11ms, SLIC takes 211ms and watershed takes 9.89ms to execute. Therefore, a 1.62% increase in time, by adding watershed, would lead to a 5% increase in segmentations with an IOU of over 50% in flat terrain. If execution is very important to a user, and an IOU of greater than 70% is considered to be a good match, then using watershed, without SLIC, would be a good compromise as this would reduce execution time, but would still have an accuracy that is similar to a system that uses both SLIC and watershed.

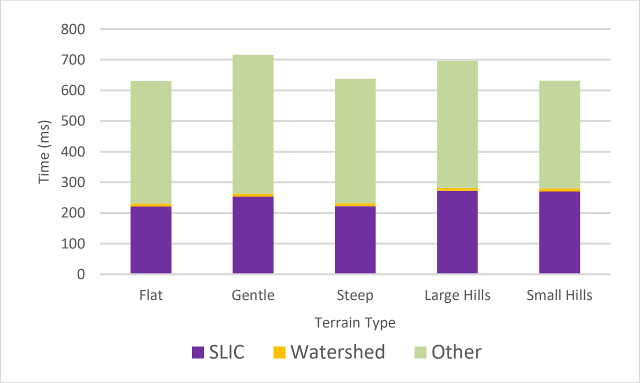


Figure 13: A graph showing the execution time taken by the SLIC, watershed and other processes for the different terrains.

6 Limitations

As a limited number of DEMs were available, the system could only be tested against a limited number of conditions. If we had more DEMs available, we could have gathered more data which would have allowed us to further optimise system parameters or even introduce new filters.

The IOU per tree calculation was done with bounding boxes due to time constraints. A more accurate, pixel-wise calculation would produce more accurate results.

Time was a significant factor. There were many variables to consider (both for system parameter optimisation and for pre-processing operations) and many different approaches that we could have taken. Had we had more time, we could have explored more avenues with regards to improving system accuracy. We also could have investigated, in greater detail, the effects the type of data (i.e. noisy, disperse or compact trees) had on the final result.

7 Ethical, Professional and Legal Issues

There were no ethical issues to consider during this project. DEMs of real terrain were used, therefore is it possible that a person located underneath the scanning equipment could have been subsequently recorded by the LiDAR system. However, it would not be possible to identify this person, because only distance data was collected. As the research did not involve human or animal subjects, ethical clearance from the university was not required. Communication with the DEM provider, Aerobotics, took place through the supervisor. As no novel engineering techniques were used, there was no dispute over the ownership of intellectual property creating during this project. As per the university's publishing policy, the work here shall be published under the creative commons licence.

8 Conclusions and Future Work

As the agricultural industry plays such a crucial role to keep humanity alive, it is important that we find ways to improving the industry in order to improve food security. By monitoring trees, we can gather data about them and use this to improve their growing conditions. Many of these methods require trees to be identified from aerial images. This project investigated ways of segmenting trees from DEMs. SLIC and watershed were considering as ways of achieving this. Many pre-processing functions, such as a Gaussian filter and a local maxima detection algorithm, were also used to prepare the DEMs for segmentation.

The results of the segmentation varied depending on the DEMs used. In a best case scenario, where trees are smooth topped, a mean IOU per tree of 88.16% can be achieved while in the worst case scenario, in a DEM with noisy, compacted trees on a large hill, a mean IOU per tree of 54.3% can be achieved. As a result, this system may be usable on orchards with spaced-out identical trees. However, more work is required to make it robust enough to operate in conditions with uneven terrain.

The combination of SLIC and watershed has been shown to be more effective than when either are used alone. SLIC has a much longer execution time than watershed segmentation. In time critical operations, SLIC can be removed from the system to reduce execution time, with a small reduction in segmentation accuracy.

Overall, this project has shown that SLIC and watershed are viable options when segmenting trees in DEMs. Although DEMs were tested to optimise the system parameters, our estimations for them, using information about the DEMs, were very close to the optimal values. Therefore, test data and training is not required for this system, unlike with the neural network approaches [22].

There is much to be done and explored in the field of image segmentation. A different set of image processing libraries could be used to implement the significant operations done in this project. As implementations of these methods may differ slightly in different software libraries, a different result may be produced by using different image processing libraries. Convolutional Neural Networks (CNNs) [34, 37] are an alternative image segmentation approach that could be integrated into this project to improve tree segmentation, although, they would drastically increase execution time of the system [22].

ACKNOWLEDGMENTS

The author thanks Prof. Patrick Marais for his supervision and guidance throughout the project. Thanks are also extended to Aerobotics for their contribution of DEMs – particular, Michael Malahe, who provided insight about the ground height data.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua and Sabine Süsstrunk. *Slic superpixels*. 2010.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua and Sabine Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34, 11 (2012), 2274-2282.

[3] PP Acharjya, A Sinha, S Sarkar, S Dey and S Ghosh. 2013. A new approach of watershed algorithm using distance transform applied to image segmentation. *International Journal of Innovative Research in Computer and Communication Engineering*, 1, 2 (2013), 185-189.

[4] Rolf Adams and Leanne Bischof. 1994. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16, 6 (1994), 641-647.

[5] Salem Saleh Al-Amri and Namdeo V Kalyankar. 2010. Image segmentation by using threshold techniques. *arXiv preprint arXiv:1005.4020* (2010).

[6] Min Bai and Raquel Urtasun Deep watershed transform for instance segmentation. City, 2017.

[7] Christopher W Bater and Nicholas C Coops. 2009. Evaluating error associated with lidar-derived DEM interpolation. *Computers & Geosciences*, 35, 2 (2009), 289-300.

[8] Mariana Belgiu and Lucian Drăguţ. 2014. Comparing supervised and unsupervised multiresolution segmentation approaches for extracting buildings from very high resolution imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96 (2014/10/01/ 2014), 67-75.

[9] Michael Bennett *Doppler lidar for boundary-layer measurements: must it be expensive?* International Society for Optics and Photonics, City, 2003.

[10] Serge Beucher Watershed, hierarchical segmentation and waterfall algorithm. Springer, City, 1994.

[11] Serge Beucher and Fernand Meyer. 1993. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, 34 (1993), 433-481.

[12] Emmanuel John M Carranza. 2010. Catchment basin modelling of stream sediment anomalies revisited: incorporation of EDA and fractal analysis. *Geochemistry: Exploration, Environment, Analysis*, 10, 4 (2010), 365-381.

[13] S Crommelinck, R Bennett, M Gerke, MN Koeva, MY Yang and G Vosselman. 2017. SLIC superpixels for object delineation from UAV data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4 (2017), 9.

[14] Per-Erik Danielsson. 1980. Euclidean distance mapping. *Computer Graphics and image processing*, 14, 3 (1980), 227-248.

[15] Ralph O Dubayah and Jason B Drake. 2000. Lidar remote sensing for forestry. *Journal of Forestry*, 98, 6 (2000), 44-46.

[16] Ricardo Fabbri, Luciano Da F Costa, Julio C Torelli and Odemir M Bruno. 2008. 2D Euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, 40, 1 (2008), 1-44.

[17] Jan Flusser, Sajad Farokhi, Cyril Höschl, Tomáš Suk, Barbara Zitová and Matteo Pedone. 2015. Recognition of images degraded by Gaussian blur. *IEEE transactions on Image Processing*, 25, 2 (2015), 790-806.

[18] Maria Frucci. 2006. Oversegmentation reduction by flooding regions and digging watershed lines. *International Journal of Pattern Recognition and Artificial Intelligence*, 20, 01 (2006), 15-38.

[19] Valentin Gies and Thierry M Bernard *Statistical solution to watershed over-segmentation*. IEEE, City, 2004.

[20] Chun-Yan Han. 2017. Improved SLIC imagine segmentation algorithm based on K-means. *Cluster Computing*, 20, 2 (2017), 1017-1023.

[21] Scott W Haney. 1994. Is c++ fast enough for scientific computing? *Computers in Physics*, 8, 6 (1994), 690-696.

[22] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton *Imagenet classification with deep convolutional neural networks*. City, 2012.

[23] Pierre-Eric Lauri, Evelyne Costes, Jean-Luc Regnard, Laurent Brun, Sylvaine Simon, P Monney and Hervé Sinoquet Does knowledge on fruit tree architecture and its implications for orchard management improve horticultural sustainability? An overview of recent advances in the apple. City, 2008.

[24] Victor Lempitsky, Pushmeet Kohli, Carsten Rother and Toby Sharp *Image segmentation with a bounding box prior*. IEEE, City, 2009.

[25] José Martins, José Marcato Junior, Geazy Menezes, Hemerson Pistori, Diego Sant and Wesley Gonçalves *Image Segmentation and Classification with SLIC Superpixel and Convolutional Neural Network in Forest Context*. IEEE, City, 2019.

[26] Sijie Niu, Qiang Chen, Luis De Sisternes, Zexuan Ji, Zeming Zhou and Daniel L Rubin. 2017. Robust noise region-based active contour model via local similarity factor for image segmentation. *Pattern Recognition*, 61 (2017), 104-119.

[27] Eva-Marie Nosal Flood-fill algorithms used for passive acoustic detection and tracking. IEEE, City, 2008.

[28] Jacob Olsen. 2004. Realtime procedural terrain generation (2004).

[29] Sylvain Paris, Pierre Kornprobst, Jack Tumblin and Frédo Durand *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009.

[30] Bernhard Preim and Charl Botha. 2014. Chapter 4—Image Analysis for Medical Visualization. *Visual Computing for Medicine, 2nd ed.; Preim, B., Botha, C., Eds* (2014), 111-175.

[31] Carl Yuheng Ren and Ian Reid. 2011. gSLIC: a real-time implementation of SLIC superpixel segmentation. *University of Oxford, Department of Engineering, Technical Report* (2011), 1-6.

[32] Chantal Revol and Michel Jourlin. 1997. A new minimum variance region growing algorithm for image segmentation. *Pattern Recognition Letters*, 18, 3 (1997), 249-258.

[33] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid and Silvio Savarese *Generalized intersection over union: A metric and a loss for bounding box regression*. City, 2019.

[34] Olaf Ronneberger, Philipp Fischer and Thomas Brox *U-net: Convolutional networks for biomedical image segmentation*. Springer, City, 2015.

[35] Henry Scheffe *The analysis of variance*. John Wiley & Sons, 1999.

[36] Josef Schmidhuber and Francesco N Tubiello. 2007. Global food security under climate change. *Proceedings of the National Academy of Sciences*, 104, 50 (2007), 19703-19708.

[37] E. Shelhamer, J. Long and T. Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 4 (2017), 640-651.

[38] Henning Tangen Søgaard and Hans Jørgen Olsen. 2003. Determination of crop rows by image analysis without segmentation. *Computers and electronics in agriculture*, 38, 2 (2003), 141-158.

[39] Murala Subrahmanyam, RP Maheshwari and R Balasubramanian. 2012. Local maximum edge binary patterns: a new descriptor for image retrieval and object tracking. *Signal Processing*, 92, 6 (2012), 1467-1479.

[40] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell and Kilian Q Weinberger *Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving*. City, 2019.

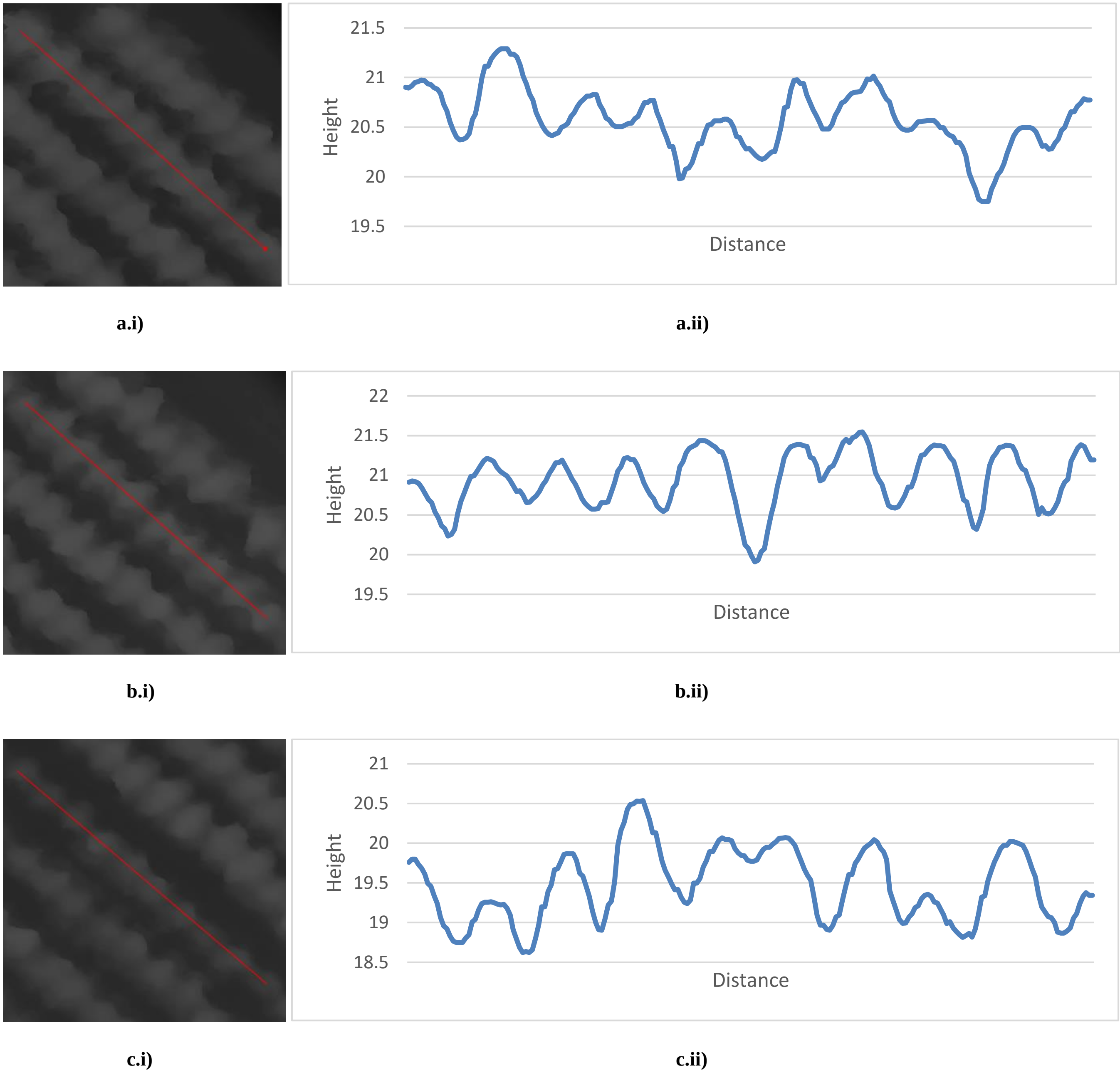
[41] TA Wheaton, JD Whitney, WS Castle, RP Muraro, HW Browning and DPH Tucker. 1995. Citrus scion and rootstock, topping height, and tree spacing affect tree size, yield, fruit quality, and economic return. *Journal of the American Society for Horticultural Science*, 120, 5 (1995), 861-870.

[42] Xiangyang Xu, Shengzhou Xu, Lianghai Jin and Enmin Song. 2011. Characteristic analysis of Otsu threshold and its applications. *Pattern recognition letters*, 32, 7 (2011), 956-961.

[43] Weihua Zhang and David R Montgomery. 1994. Digital elevation model grid size, landscape representation, and hydrologic simulations. *Water resources research*, 30, 4 (1994), 1019-1028.

APPENDIX A

Height Profiles of DEMs

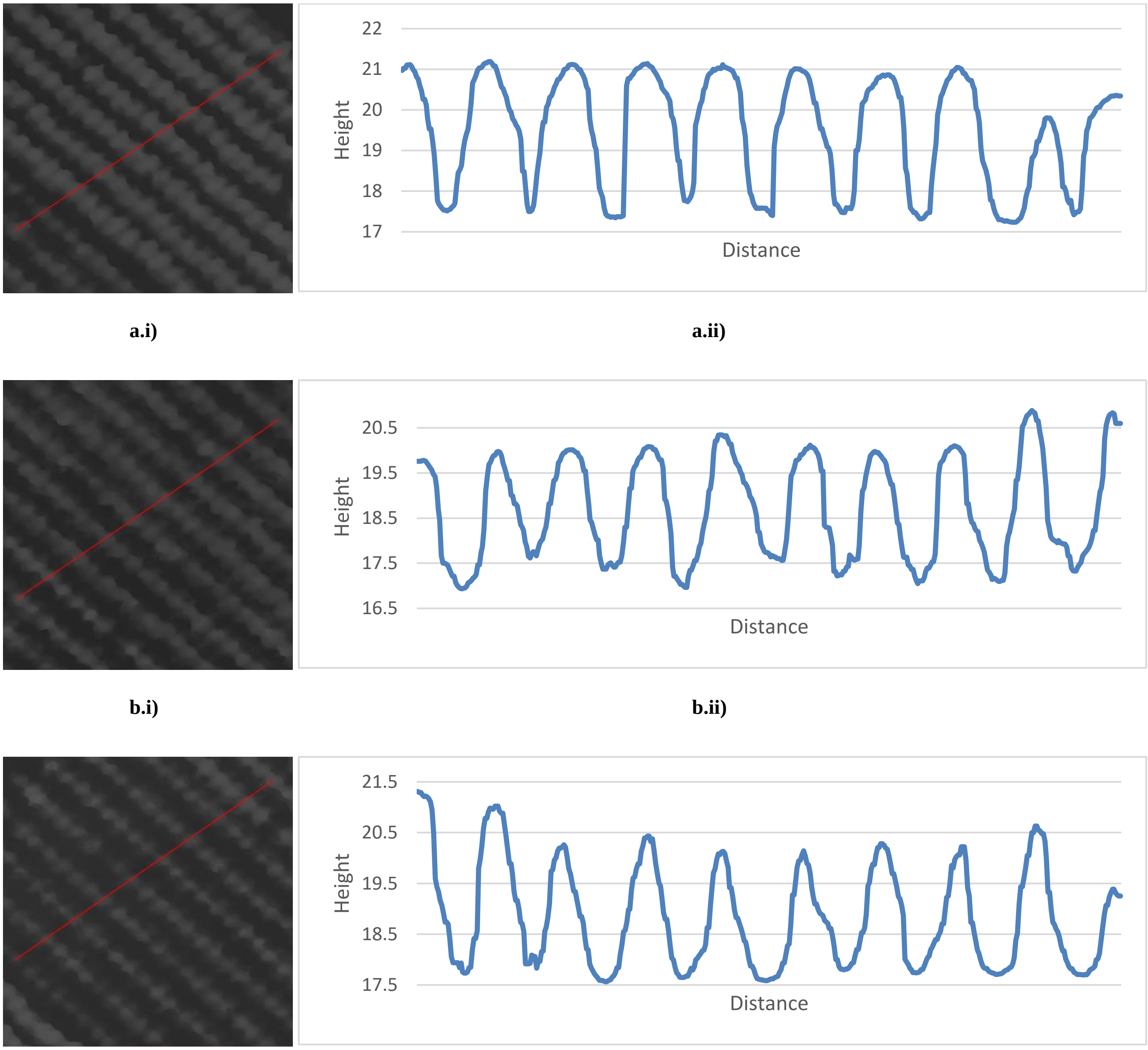


a.i) **a.ii)**

b.i) **b.ii)**

c.i) **c.ii)**

Figure A1: A series of samples of a DEM of an orchard (a.i, b.i and c.i) and their corresponding height profiles (a.ii, b.ii and c.ii), demonstrating the intrarow heights of the DEM.

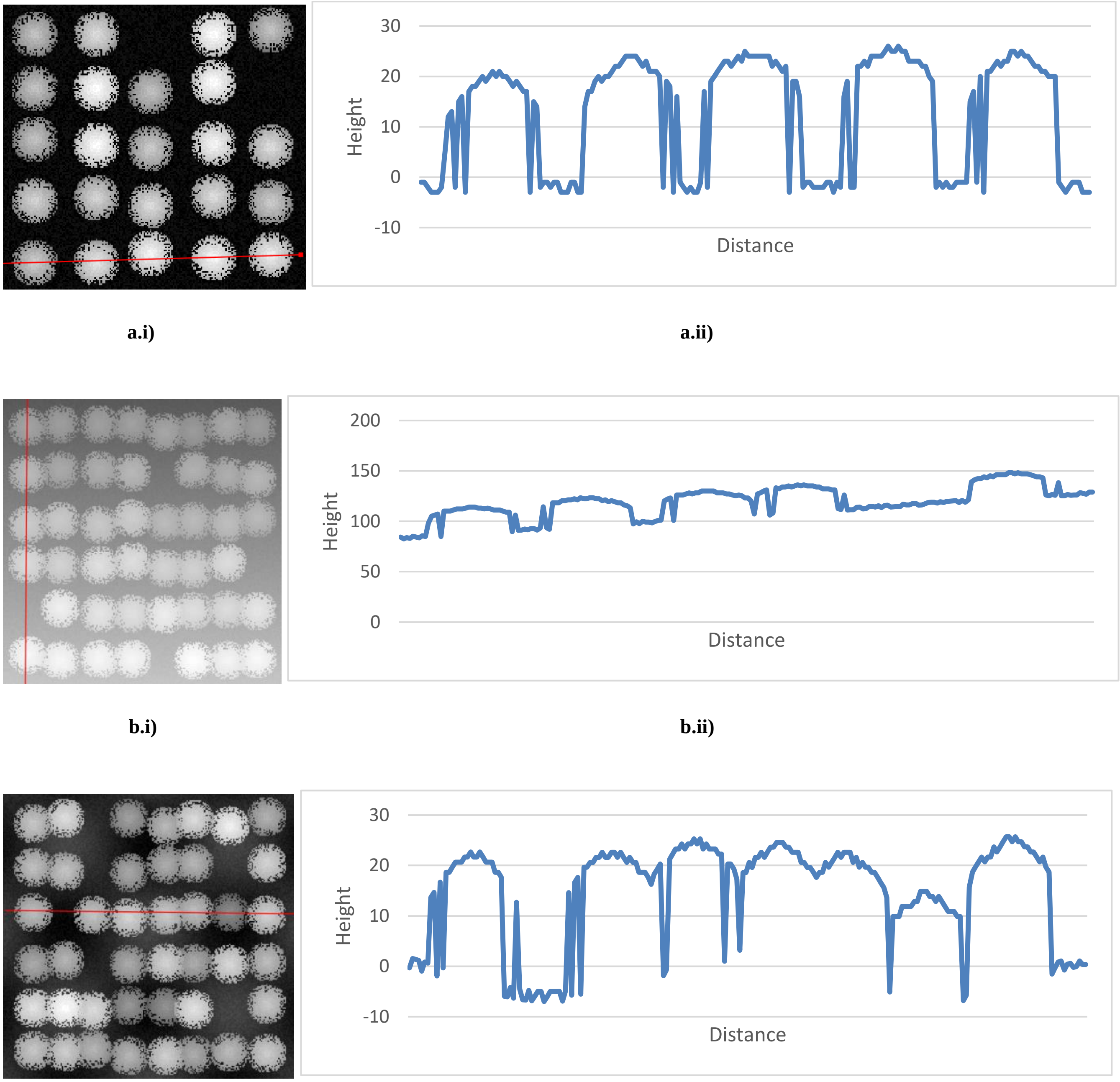


a.i) a.ii)

b.i) b.ii)

c.i) c.ii)

Figure A2: A series of samples of a DEM of an orchard (a.i, b.i and c.i) and their corresponding height profiles (a.ii, b.ii and c.ii), demonstrating the interrow heights of the DEM.



a.i) a.ii)

b.i) b.ii)

c.i) c.ii)

Figure A3: A series of synthetic DEMs, showing a flat terrain (a.i), a steep terrain (b.i) and a small-hilled terrain (c.i) and their corresponding height profiles (a.ii, b.ii and c.ii respectively).

APPENDIX B

Local Maxima Test Results

Table B1: A table displaying the mean Euclidean distance of the calculated centres from the actual centres of the different terrains. The word *easy* indicates that the image has smooth trees.

Terrain	Total Trees	% of Trees Identified	Mean Distance from Centre (in pixels)	Standard Deviation	Terrain	Total Trees	% of Trees Identified	Mean Distance from Centre (in pixels)	Standard Deviation
Flat_easy	42	100	0	0	LargeHill_easy	35	85.714	2.8858	5.63904
Flat_spread_easy	29	100	0	0	LargeHill_join_easy	33	81.818	2.70638	5.32497
Flat_small_easy	211	97.630	0.0194175	0.138323	LargeHill_join_small_easy	62	80.645	0.3	1.0351
Flat_spread	23	100	0.539748	0.535106	LargeHill_small_easy	68	79.412	0.54153	1.63202
Flat_small_spread	151	100	0.542381	0.526493	LargeHill_spread	16	100	0.650888	0.530504
Flat	44	100	0.554868	0.559346	LargeHill_small_spread	39	100	0.457139	0.534345
Flat_small	211	96.682	0.497477	0.518207	LargeHill_join_spread	15	100	0.521895	0.592476
Gentle_easy	43	97.674	0.119048	0.452763	LargeHill	35	85.714	1.75338	4.02479
Gentle_small_easy	214	93.925	0.0298507	0.1706	LargeHill_small	68	70.588	0.734222	1.06316
Gentle_spread	22	100	0.564282	0.534294	LargeHill_join_small	62	72.581	0.809205	1.06102
Gentle	43	95.349	0.48362	0.534101	LargeHill_join	33	90.909	2.4713	4.58551
Gentle_small	208	97.115	0.552757	0.510004	SmalHill_small_easy	207	93.720	0.0721649	0.259431
Steep_easy	46	95.652	0.0909091	0.362047	SmallHill_easy	42	92.857	0.102564	0.383534
Steep_small_easy	211	95.261	0.0497512	0.217974	SmallHill_spread	22	100	0.590909	0.503236
Steep_spread	24	100	0.725592	0.482943	SmallHill	42	88.095	0.443627	0.519567
Steep	44	100	0.505365	0.56574	SmallHill_small	210	95.714	0.61094	0.529042
Steep_small	215	95.348	0.509684	0.518035	SmallHill_smoothGround	45	91.111	0.597042	0.67796

Tree Mask Results

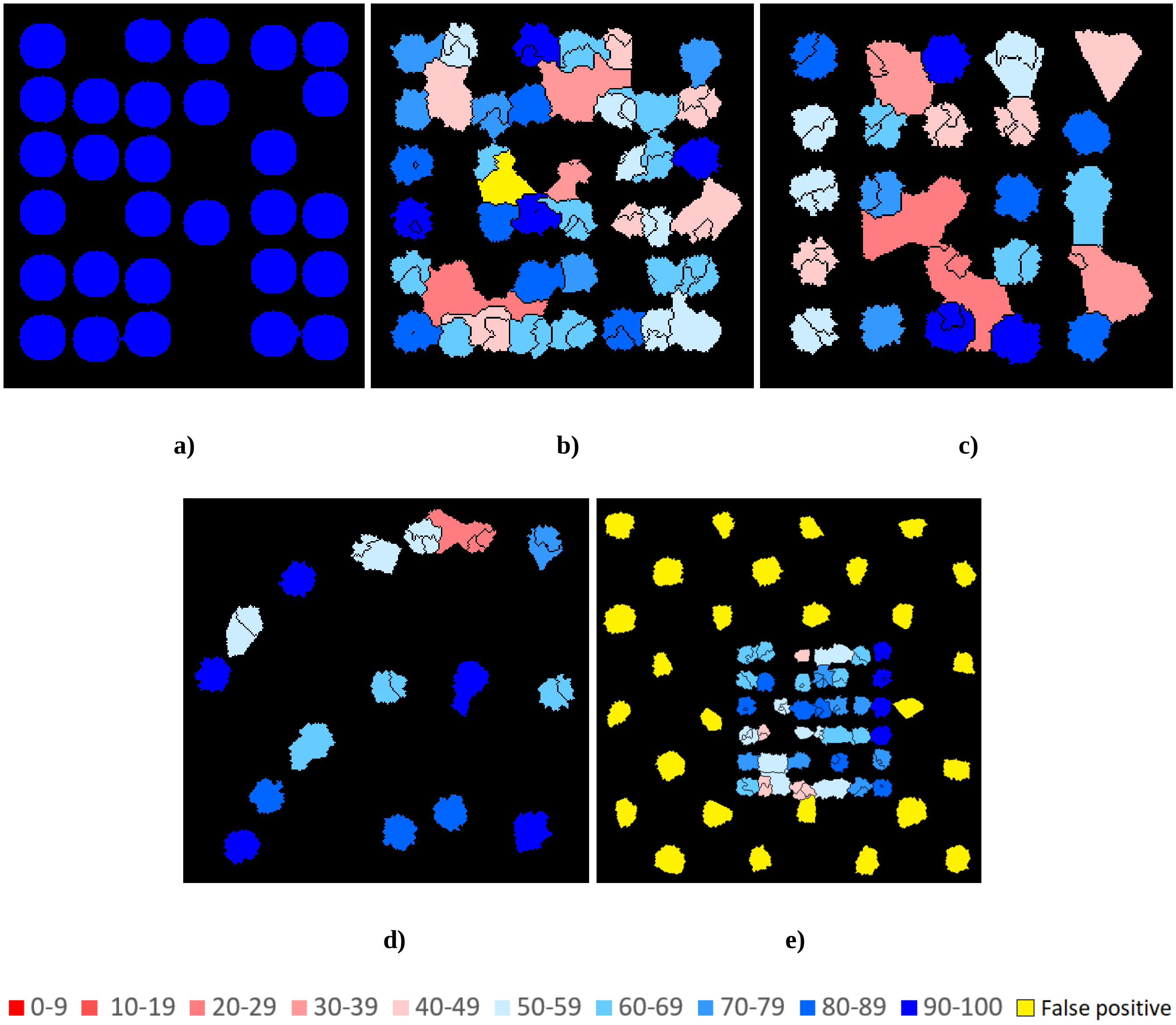


Figure B1: A series of images showing the tree masks produced, and their IOU percentage (as shown in the legend above), for the flat_spread_easy DEM (a), gentle DEM (b), steep_spread DEM (c), largeHill_spread DEM (d) and smallHills DEM (e).