Automatic Malleefowl Mound Detection using Robust LiDAR-based Features and Classification

Author

Hossain, Nazia, Murshed, Manzur, Lu, Guojun, Awrangjeb, Mohammad, Florentine, Singarayer Kumardas, Irvin, Marc, Teng, Shyh Wei

Published

2022

Conference Title

2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)

Version

Accepted Manuscript (AM)

DOI

10.1109/dicta56598.2022.10034606

Automatic Malleefowl Mound Detection using Robust LiDAR-based Features and Classification

Nazia Hossain*, Manzur Murshed†, Guojun Lu*, Mohammad Awrangjeb§,
Singarayer Kumardas Florentine‡, Marc Irvin¶ and Shyh Wei Teng*
*Centre for Smart Analytics, Federation University Australia, VIC, Australia
†School of Information Technology, Deakin University, VIC, Australia
‡Future Regions Research Centre, Federation University Australia, VIC, Australia
§School of Information and Communication Technology, Griffith University, QLD, Australia
¶Department of Planning and Environment, NSW, Australia
{n.hossain, guojun.lu, s.florentine, shyh.wei.teng}@federation.edu.au,
m.murshed@deakin.edu.au
m.awrangjeb@griffith.edu.au,
marc.irvin@environment.nsw.gov.au

***Abstract*—Malleefowl is listed as one of the vulnerable birds in Australia. To track the pattern of presence and abundance of Malleefowl, surveying the egg-incubator (a.k.a. nest or mound) is an extensively used technique. However, on large conservation areas, Malleefowl mound detection by manually inspection on land or from air is challenging for various environmental and technical reasons. Usually, mounds are built on the ground, and they are widely scattered over the large areas. Hence, in recent years, airborne Light Detection and Ranging (LiDAR) techniques have been used for data acquisition and analysis. However, such existing methods are still limited in terms of detection accuracy and system automation. In this paper, we propose a novel method to address these limitations. We have designed robust features which effectively represent the key visual characteristics of candidate mounds captured in LiDAR point cloud data. These features include: (1) differences of elevation between original ground points and the corresponding feet of these ground points fitted plane, and (2) convex-hull measurements. Using these features, we then use machine learning methods, i.e., clustering to differentiate the true mounds among the candidate mounds, and bagged-tree classifier to learn a model for classifying whether a patch contains a mound or not. Our training and testing datasets contain LiDAR point cloud data captured from the Tarawi Nature Reserve, and are provided by the New South Wales Government Department of Planning and Environment, Australia. They comprise a total of 1,060 patches (each 20 m × 20 m) - half of which contain mounds, and the remaining half contain no mound. Our experimental results show that our proposed method has more than 84% accuracy in detecting patches with mounds.**

***Index Terms*—LiDAR point cloud, Malleefowl mound, Clustering, Classification, Boosted Tree Classifier, Spatial-Density-based Clustering, DBSCAN.**

I. Introduction

Malleefowl (*Leipoa ocellata*) is a domestic chicken sized ground-dwelling bird from arid and semi-arid Australia. The Malleefowl is listed as vulnerable by International Union for Conservation of Nature (IUCN) Red List of Threatened Species [1]. It is also listed as Endangered Nationally and Vulnerable in NSW (where the research site is located). Malleefowl constructs mound and uses this mound as a chamber or incubator to lay and breed eggs. During the breeding season, Malleefowl adds and removes sand and leaf-litters each day to maintain the temperature inside the mound between 32-34°C which is required for egg incubation. The fundamental tasks of detecting and analysing the locations and surrounding of these mounds are significant in the conservation process of Malleefowl in different ways. For example, to protect eggs from the predators, like red-foxes, as well as to track the presence of Malleefowl itself, and to preserve its habitat. There are several ways to survey Malleefowl mounds from the large conservation area. Firstly, walking transects across suitable habitat on foot, and observing from a helicopter at low altitude along transects are the two manual surveys [3]. Both of these techniques are the most accurate ones; however, the most labour extensive as well. Secondly, semi-automated techniques are taken into consideration. Often high-definition aerial images are taken for further investigation of the mounds [6]. Sometimes, software tools are used for further processing of those images, e.g., orthorectification to ease the detection. However, aerial imaging-based survey suffers incompetency in providing the height details. To overcome the limitation, Light Detection and Ranging (LiDAR) has gained popularity among the remote-sensing technologies for various mounds, including Malleefowl mound detection [4]. In general, the light (laser) rays have the ability to pierce the canopies or other obstacles, and they reflect or refract the light rays partially or fully based on the object types. The significant data, e.g., position (easting, northing, elevation), intensity, light ray return number or sequence after each reflection are stored for further processing as LiDAR point cloud data.

LiDAR is an emerging technology that offers versatile solutions for diverse real-life problems. Object classification/detection and objection segmentation are two analytics operations commonly used with LiDAR data to extract information-of-interest for decision-making in various appli-

This research is supported by the Research Excellence Scholarship and DPE NSW Top-Up Scholarship.

cations. We can generally classify LiDAR-based object detection applications into three categories: (i) urban, (ii) forest or conservation, and (iii) water-based areas. Among urban-based applications there are human, buildings and/or building roofs, urban-trees, ground or street, standing or motion vehicle detection [13] [14]. Usually Digital Elevation Model (DEM) is widely used technique for structure detection in both on-ground and off-ground level. Additionally, multi-spectral imagery enhances the performance. For forest area, human intervention is quite rare and difficult. Therefore, remote-sensing technology like LiDAR is gaining popularity. Automatic power line and pylon detection [15], tree detection in forest area [16] are a few examples of them. Lastly, under-water object detection, e.g., fish detection [17] is also getting popularity from LiDAR. Several end-to-end learning methods and handcrafted feature-based machine learning methods are well-established in LiDAR data-based classification and segmentation tasks. For example, PointNet for classification [18] and PointNet++ [19] and PointSeg [20] for segmentation. However, most of these researches have been done with terrestrial LiDAR data, and a pre-acknowledged and controlled environment. On the other hand, our aim is to find Malleefowl mounds in a diverse and large conservation area automatically.

For mound detection, usually, airborne LiDAR technique is used due to the diverseness of the surrounding environment in terms of size, canopy density and versatility of the conservation area landscape. Some researchers have used LiDAR for mound detection whether this is a geographical pile (anthropogenic) or Malleefowl Mound. D. Davis *et.al.* [5] have used an object template-based image analysis tool to examine earthen mounds, shell heaps or rings from LiDAR data. However, this semi-automated technique missed some ploughed or tempered mounds or rings; as well as, their method partially depends on templates of previously known mounds. Likewise, for Malleefowl mound detection, LiDAR with aerial images have started to be used [7]. Umwelt Pty. Ltd. [7] has developed a technique to identify and rank the candidate mound locations based on the likelihood to be an actual mound, and named the software as *Anditi*. Along with the software, this company had ecological expertise. They have also used aerial images for habitat and landscape information for determining Malleefowl mounds with their corresponding locations. However, they did not differentiate the active and inactive mounds [7], and obtained a large number of false positives. Considering all the facts, J. Read *et. al.* [8] surveyed three basic mound detection techniques: (i) helicopter spotter, (ii) manual walking survey, and (iii) airborne LiDAR data. Among them, the LiDAR technique is comparatively versatile and accurate. This is more effective in terms of time and energy than an in-person survey, and more accurate than a helicopter survey because of its canopy piercing ability. However, this LiDAR assessment is combined with the orthorectified image for the detection process which is also not entirely automated. The first technique where automated mound detection is done by the system after providing necessary inputs has been proposed by N. Parvin *et. al.* [9]. They developed a morphological image-filtering-based mound detection method [9]. However, hard assumptions of thresholds caused unnecessary non-mound objects as the mounds in real environmental data.

In summary, for not considering the dynamic change in landscape, and analysing certain elevated ground portions, the performances of the existing mound detection techniques are still limited in certain aspects. Those methods also ignored Malleefowl mound shape characteristics in both 2D and 3D manners. Therefore, our proposed method focuses on finding and extracting features from the mounds, like elevated ground point (candidate mound points) within a small area (patch) in various orientations (Note that, mound points are a cluster of LiDAR ground returns that collectively contribute to a malleefowl mound shape). The proposed method comprises two ideas: (i) differentiating the most probable mound point cluster from multiple candidate mound point clusters within a patch, and (ii) classifying mound and non-mound patches.

The rest of the paper is organised as follows. Section II describes the experimental site and dataset, Section III describes the proposed method. Our experimental results are presented in Section IV and Section V concludes our findings. For the rest of the paper, we refer to "Malleefowl mound" as "mound".

II. Experimental Site and Dataset Preparation

Our project site is Tarawi Nature Reserve (NR), that covers an area of 33,573 hectares or 335.73 square kilometres of relatively undisturbed Scotia Mallee and She-oak communities. This area also supports Belah communities with relatively large understoreys [2]. The main aim of this project is to find the true mounds from a large conservation area. In this project, there are basically two types of data: i) airborne LiDAR point cloud and ii) aerial imagery.

In this research, we have used LiDAR point cloud data for the Tarawi NR site provided by the Department of Planning and Environment (DPE) of New-South Wales (NSW) Government. DPE contracted with *Anditi* to analyse the data for mound detection. Hence, *Anditi* classified the point cloud data using their software. They used an aggressive classification algorithm for generating the bare earth DEM. Their classification algorithm is proprietary. They have four classes of data points: (i) ground, (ii) low vegetation, (iii) medium vegetation, and (iv) high vegetation. The 3D LAS point cloud data also include the point location (x, y, z, n), where x is easting, y is northing, z is elevation of height and n is return number. The characteristic data (radius, height) of actual mounds as ground-truth for Tarawi NR are also provided by DPE. We are using data from 53 ground-truth mounds for this experiment. *Anditi* also shared 1,397 .las files as classified point cloud data for the entire site. Hence, the input .las files are consisting of the information shown in Table I for each point.

To create our experimental datasets, we decided to partition the entire area into smaller patches for the ease of analysis. Using the 53 ground-truth information, first, we have cropped patches from large .las files each of size 60 m $\times$ 60 m with

TABLE I: Input: Patchwise LiDAR point cloud data.

properties	value : meaning
position	easting, northing, elevation
class	2 : ground 3: low vegetation 4: mid vegetation 5: high vegetation
return number	1 : first return 2 : second return 3: third return 4: forth/last return
number of return	1 to 4 (total number of returns for a certain laser ray)

every mound at the centre. Then, from each of these 53 (60 m $\times$ 60 m) patches, 20 more patches are generated, and each patch size is 20 m $\times$ 20 m. These 20 patches are classified into two types: (i) Type-1: 10 patches with mounds, and (ii) Type-2: 10 patches that do not contain any mound. The Type-1 patches are generated in such a way that any one of the ground-truth mounds remains anywhere in the patch. That means, using the ground-truth mounds centre location, provided by DPE, each ground-truth mound is being shifted 10 times at random locations for every patch generation. As a result, the entire or partial mounds can be seen in the Type-1 patch. On the other hand, for Type-2 patches, we have generated more 20 m $\times$ 20 m patches randomly with no mound points. Therefore, by considering the ground-truth mound centre location and radius information, it has been ensured that there will be no mound point at all inside the Type-2 patches. To keep the data balanced, we have generated an equal number of Type-1 and Type-2 patches to prepare the experimental dataset.

Therefore, we have generated total 1,060 patches (53 $\times$ 20 = 1,060) - each of 20 m $\times$ 20 m in size. Among them, the total number of Type-1 patches is 530, and the total number of Type-2 patches is 530. We named this dataset as *Dataset-1* and this dataset will be used for training the classifier model. Another dataset named *Dataset-2* is also generated following the same procedure and with the same size with randomly generated different patch-centre locations. *Dataset-2* will be used for testing the classifier only. In our datasets, each 20 m $\times$ 20 m patch contains only one mound. This is because, based on our dataset, we observed, this is unlikely that the distance between two mounds will be less than 20 m. Hence, our two experimental datasets are representing real conservation areas of Malleefowl habitat where the ground-truths are provided by a government organisation.

III. Proposed Method

From the previous discussion, we have acknowledged that Malleefowl builds its mound on the ground. Such a mound is like a small pile of sand and/or soil elevated in the landscape. Therefore, in this research, our main idea is to find and analyse the properties of elevated portions of the landscape. Figures 1a and 1b show two RGB images of two types of mounds (red-circled portion). Figure 1a shows the mound with a crater and the other one in Figure 1b shows a mound in full-form. Figure 1c presents the corresponding LiDAR point cloud view of the mound shown in Figure 1a in 3D. The black and green colours represent ground and vegetation points respectively. The points marked in red are for identifying the mound inside the .las file. However, in this figure, the elevations of the ground points are not clear. Hence, Figure 1d shows the ground points in a different view which better visualises that, the sudden elevation of ground points in red colour are the mound points.

In this regard, our primary aim is to find sudden elevated portion(s) of ground points inside a certain patch. We shall call these Patchwise Mound Cluster (PMC). We are considering a 20 m $\times$ 20 m patch for this experiment as mentioned in Section II. Mainly, we will identify whether this elevated portion inside a patch is a mound or not based on relevant features. The procedure is illustrated as follows and the workflow diagram is shown in Figure 2.

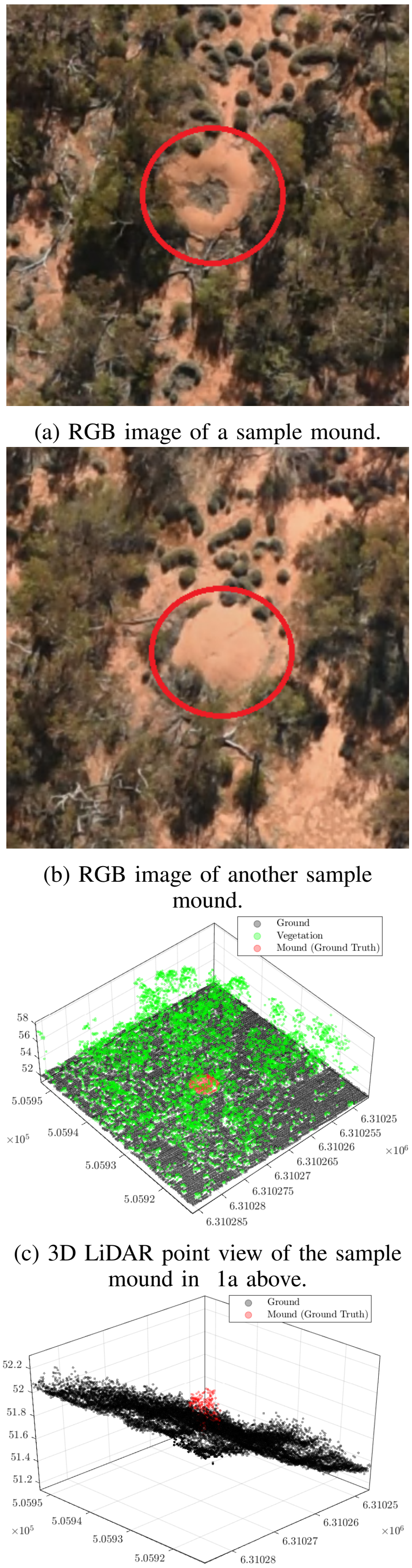
A. *Plane Fitting to the Ground Points*

Based on our datasets, we have observed that within a certain patch, the ground can be used as a reference to maintain a certain planarity property. Thus, by fitting a plane to the 3D ground points of a patch, we can then divide the ground points into three parts: (i) above, (ii) on the plane, and (iii) under the plane. Since our focus is to find the elevated portion of the land, we can then filter out the points which are below the fitted plane and keep the most elevated ground points for further analysis. To do this, the ground point positions: x, y, z are used to fit a plane at least square of the normal distance. To show an example, Figure 3a shows only the ground points in black colour. As illustrated here, without an appropriate reference, it is difficult to decide which elevated points are the Candidate Mound Points (CMPs). This is why, plane fitting is important. After fitting the yellow plane as shown in Figure 3b, the points which are on or above the plane remain black and the points below the plane are not visible from the top view, as shown in Figure 3c. So, only the remaining black points in Figure 3c are now considered for further analysis.

Note that, the patch size cannot be too large or small. This is because, if the patch size is too large, then the expected planarity property will fail. If the patch size is too small, then we may not get the proper mound shape, if there is any. For this experiment, we have fixed our patch size as 20 m $\times$ 20 m.

B. *Candidate Mound Points (CMPs) Extraction*

Among the black points (points above the plane) shown in Figure 3c, we need to filter out those points that would be responsible for building Candidate Mound Clusters (CMCs). Therefore, we take the difference of elevation from the ground points above the plane to the projected feet on the plane. The difference between two of these elevations is expressed as ΔeZ. We then sort ΔeZ in descending order. Then, we take the top 245 elevated ground points. The number 245 is calculated at 99% confidence of mound points according to the ground-truth data provided by DPE and *Anditi* [7]. With the



(a) RGB image of a sample mound.

(b) RGB image of another sample mound.

(c) 3D LiDAR point view of the sample mound in 1a above.

(d) 3D view of the ground points only in 1a above.

Fig. 1: A sample ground-truth mound in different view.

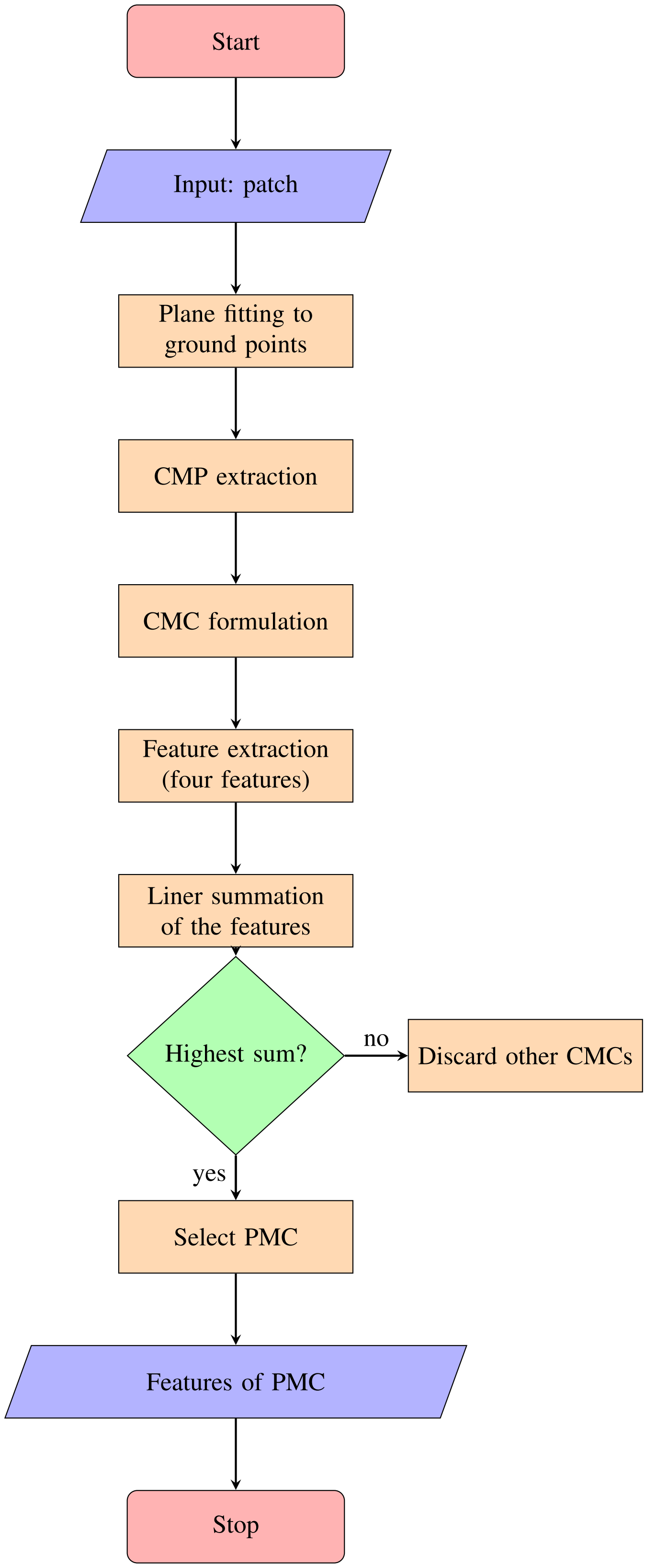
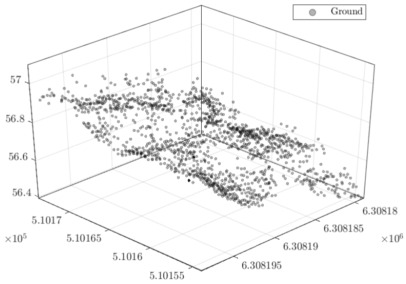
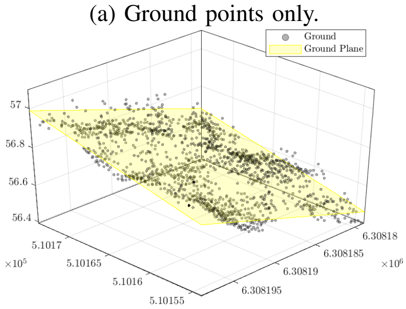


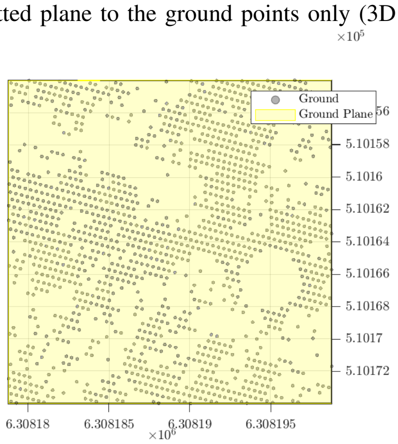
Fig. 2: Flowchart of PMC feature extraction.



(a) Ground points only.

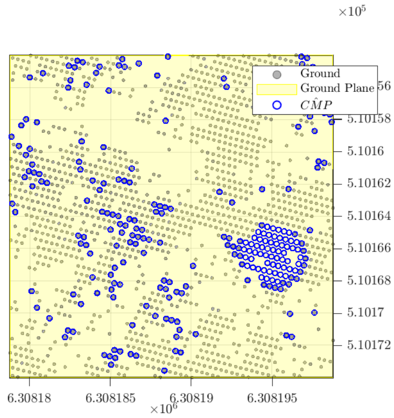


(b) Fitted plane to the ground points only (3D view).

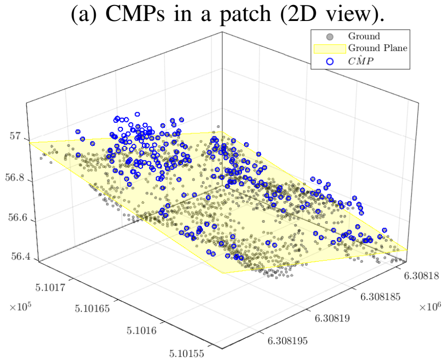


(c) Fitted plane to the ground points only (2D view).

Fig. 3: Plane fitting.



(a) CMPs in a patch (2D view).



(b) CMPs in a patch (3D view).

Fig. 4: CMP filtering.

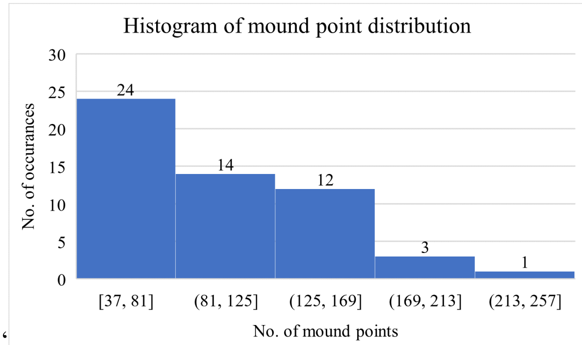


Fig. 5: Histogram of mound point distribution.

ground-truth mound locations (centre of the mound) and radius information provided by our data providers, we have calculated the total number of mound points for each patch. From the number of mound points distribution, the total number of CMP within a single patch is fixed to 245. A histogram of the actual mound point is shown in Figure 5 with 99% confidence. The blue dots in Figure 4a and 4b each shows examples of 245 CMPs.

C. Candidate Mound Clusters (CMCs) Detection

In Figure 4a, the blue points are the points which likely to represent mounds. However, some of them have created a big group of points and some of them have created smaller groups of points. Some of these groups are also spatially separated. To get the mound point groups for further analysis, various clustering techniques (e.g. [10], [11], [12]) could be used. We have applied DBSCAN (Spatial-Density-based Clustering of Applications with Noise [10] to group these groups of points into bigger candidate clusters which may represent the true mound. The primary reason to use DBSCAN is, we needed a spatial-density-based partitioning method to segregate densely grouped CMPs. Secondly, the number of probable clusters within a patch is unknown in our case. Thirdly, the DBSCAN algorithm needs only two predefined parameters to partition the clusters. Fourthly, if we can properly set the parameters, the ungrouped or small group of points will be separated as a noise cluster and will not affect in building CMCs. For DBSCAN clustering, we set two parameters: minimum number of points, $minPts$ = 6 and radius/epsilon of the circle, eps = 1.2. For this clustering process, we have used 2D data (x, y) for each CMPs only, because, z will be further needed for the cluster shape analysis but not for spatial-density-based clustering. We wanted to segregate closely residing grouped ground points in the $x-y$ plane. Hence, the DBSCAN algorithm [10] randomly selects a point from the input dataset and creates a circle with a radius equal to the predefined eps value. This algorithm also checks whether this circle consists of a $minPts$ number of data points. If the total number of data points within a circle is equal to or greater than $minPts$, then the centre of the circle is called the "core" point. Moreover, if a circle has more than 1 point, but less than $minPts$ number of points, then the centre of the circle is called the "border" point. On the other hand, if there is no data point at all around the centre of the circle within the eps radius, this point is called the "noise" point. This is how, circles are generated around each data point to determine whether the relationship among the data points is density-reachable and/or density-connected. The density-connected data points are in the same cluster. Figure 6 shows three clusters in three colours. The unclustered scattered blue points are not responsible in building mounds and DBSCAN successfully separated them as noise points.

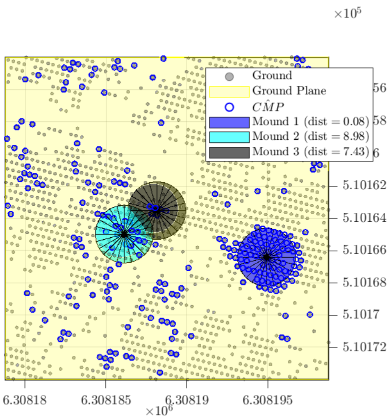


Fig. 6: CMCs detection (2D).

D. Feature Extraction and Patchwise Mound Cluster (PMC) Selection

We observe that, creating clusters is not enough to find the mound. This is because a single patch can have multiple clusters and not all the clusters represent the true mound. Therefore, to find the best mound-like cluster within a patch, our next step is to extract the features that would represent mound shape properties. We derive a total of four feature coefficients. Finally, a linear summation of these four feature coefficients is calculated for the individual cluster. The best valued cluster is selected as PMC.

The first extracted feature (f_1) is the total number of ground points within the CMCs. The blue points in Figure 7a show an example of the mound points. The second feature (f_2) is the mean elevation difference from the mound points to the feet on the fitted plane expressed as $\mu(\Delta eZ)$ within prefixed CMC-radius = 2.5 m (99% confidence of mound radius from the ground-truth dataset). Here, all the ground points inside the CMC are used for elevation calculation. Figure 7b shows

TABLE II: Feature-Weight List.

SL (i)	Feature (f)
1	Total no of ground points within the cluster
2	Mean distance of a ground point to the foot at the plane $\mu\Delta eZ$
3	Area of the cluster's convex-hull
4	Cluster points density among respective convex-hull

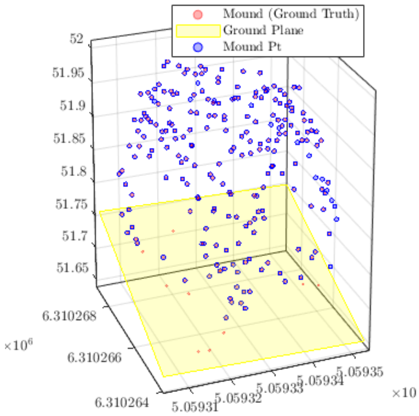
an example of ΔeZ calculation. Here, a = elevation value of a sample ground point and a' = elevation value of the foot of a on the plane. ΔeZ is calculated by simply subtracting a and a'. The third feature f_3 which is the convex-hull area, is then calculated for each cluster. The forth feature f_4, which is the density of the convex-hull, is calculated for each cluster. f_4 is calculated by dividing the total number of points belonging to the cluster divided by the respective area of the convex-hull. Figure 7c shows an example of the convex-hulls of the clusters. The blue solid line denotes the boundary of the convex-hull. Finally, we calculated the summation of all these four feature coefficients and choose CMC as a PMC which contains the highest summation value. These features indirect (f_1) or directly (f_2,f_3 and f_4) represent the mound shape. Figure 8 shows a 3D patch after PMC detection. The blue cluster is the selected PMC.

E. Classifier Modelling

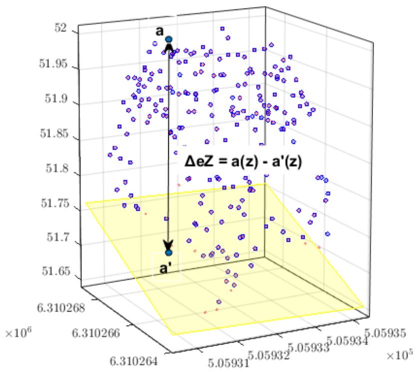
In our proposed method, each patch will have at least one PMC even if the patch has no real mound. Therefore, we need to train a classifier model with two types of patches: (i) Type-1: has a mound, (ii) Type-2: no mound. The binary classifier is trained to classify patches represented by PMC features similar to the features extracted to represent Type-1 patches as true mounds, and vice versa for PMC features extracted to represent Type-2 patches. This feature set is modelled to feed a binary classifier for further analysis to predict whether a given patch has a mound or not.

IV. Result Analysis

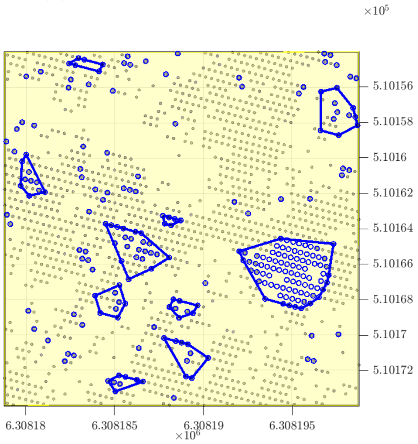
In this section, the experimental results and corresponding analysis are presented. We used MATLAB R2021b programming platform. The system configuration is 11^{th} Gen Intel(R) Core(TM) with 2.80GHz processing speed and 16.0GB RAM. According to the discussion in Section II, we have used Dataset-1 to train the classifiers and Dataset-2 for testing only. Figure 9 presents the accuracy comparison between training (Dataset-1), and testing (Dataset-2) datasets correspondingly. We have plotted results using five different classifiers. Among them, *bagged-Tree* classifier [21] performs the best in both cases which is more than 88% for our testing dataset (Dataset-2). Unlike *decision-tree*, the difference and significance of *bagged-tree* is, this uses multiple *decision-trees* rather than one which improves the predictive capability. Along with this, *boosted-tree* and *Fine Gaussian Support Vector Machine (SVM)* are showing good performance as well.



(a) Feature-1: numberOfGroundPoints (blue).



(b) Feature-2: meanElevationDelta.



(c) Feature-3 & 4: convex-hull.

Fig. 7: Elevation-based feature representation.

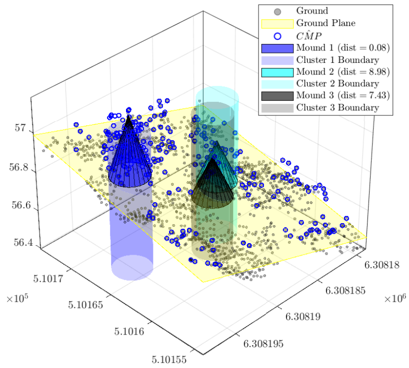


Fig. 8: PMC detection (3D).

Additionally, we have done different k-fold tests (5-fold and 10-fold) for Dataset-1 to get the best performing training model. Figure 10 shows that 10-fold classification model works better for overall classifiers. Moreover, we have done T-test for the features to find the level of importance. From Figure 11 this is obvious that, f_2 is the highest one. In both the figures: Figures 9 and 10, abbreviations of *WeiKNN* represents weighted K-Nearest Neighbours, and FineGauSVM represents fine Gaussian Support Vector Machine.

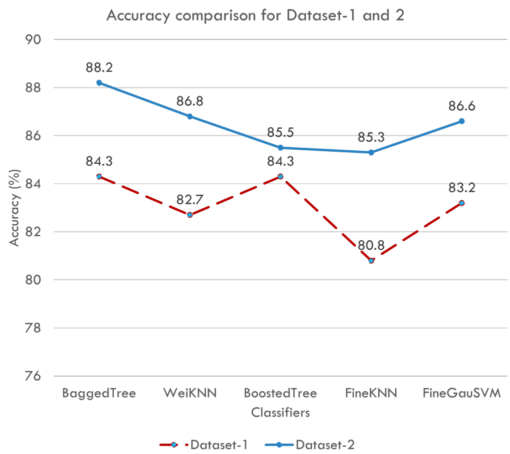


Fig. 9: Accuracy (%) comparison between two datasets.

V. Conclusion

Our research objective was to detect whether a sampled patch contains a Malleefowl mound, or not. Our proposed method uses dynamic elevation differences of ground points using the robust planarity property rather than the fixed elevation value derived from DEM. We applied a machine learning method to classify mound or non-mound patch, and predict a cluster as a probable mound automatically. Unlike the other methods, we have introduced the mound shape features like convex-hull which have improved the detection accuracy. Our experimental results show that our proposed method has more than 84% accuracy in detecting patches with mounds.

In our proposed method, the centre of PMC and the actual mound is not exact always. Usually, our PMC is ideally the

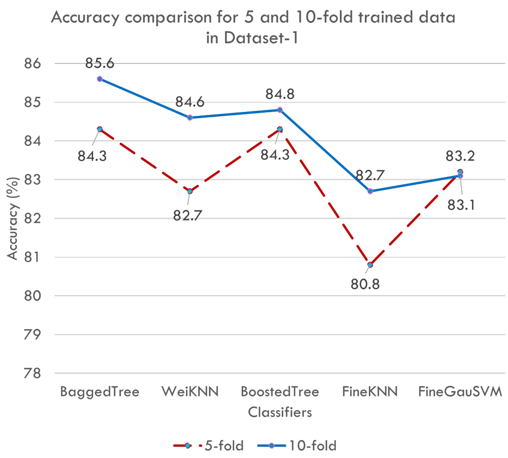


Fig. 10: Accuracy (%) comparison between k-folds training.

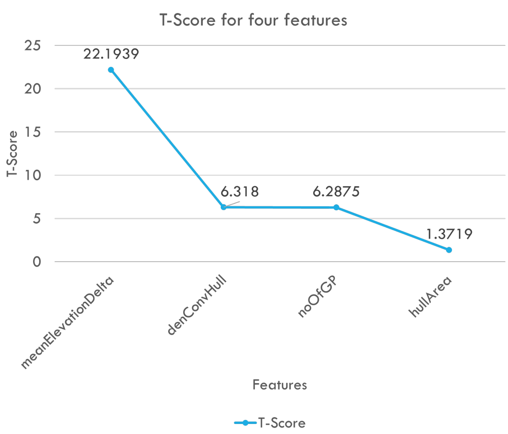


Fig. 11: T-Test result for four features.

nearest cluster from the mound if there is any. Thus in terms of future work, we will work on mound location accuracy. We also will focus on investigating the method with raw point cloud data with suitable and state-of-the-art classification methods, and testing the classification accuracy with a bigger dataset. Additionally, in our experiment, we have fixed the patch size as 20 m $\times$ 20 m. However, our future plan is to find the optimal patch size in different conservation areas. Note that, for Tarawi NR, the airborne LiDAR data collection is done in the month of June 2019. So, our mound information is dependent on that specific season. If we have datasets for at least two different seasons, then newly built mounds could easily be found just by subtracting two DEMs from two corresponding datasets. However, airborne LiDAR data collection is time, energy consuming and expensive. Therefore, automated Malleefowl mound detection is important.

REFERENCES

[1] BirdLife International (2016). *Leipoa ocellata*. [Online] IUCN Red List of Threatened Species. 2016, ”Available: https://www.iucnredlist.org/species/22678646/212766400,” (Accessed on 25 January 2022).

[2] Tarawi Nature Reserve. [Online] ”Available: https://www.nationalparks.nsw.gov.au/ visit-a-park/parks/tarawi-nature-reserve,” [Accessed: 05.05.2022]

[3] H.J. Frith, ‘Conservation of the mallee-fowl, Leipoa ocellata Gould (Megapodiidae),” Reprinted from CSIRO Wildlife Research v.7 no.1 pp.33-49, 1962.

[4] J. Sackmann, P. Jamieson, ‘Comparative Assessment of Malleefowl Mound Search Techniques,” Proceedings of the National Malleefowl Forum. 2018.

[5] D. S. Davis, M. C. Sanger, and C. P. Lipo, ‘Automated mound detection using LiDAR and object-based image analysis in beaufort county, south carolina,” Southeastern Archaeology, vol. 38, no. 1, pp. 23–37, 2019.

[6] S. Thompson, G. Thompson, J. Sackmann, J. Spark, and T. Brown, ‘Using high-definition aerial photography to search in 3D for malleefowl mounds is a cost-effective alternative to ground searches,” Pacific Conservation Biology, vol. 21, no. 3, pp. 208–213, 2015.

[7] Dr Vi Saffer, ‘The use of LiDAR to determine the presence of Malleefowl mounds,” Proceedings of 5th National Malleefowl Forum, Dubbo, New South Wales, pp. 144-150, September 2014.

[8] John Read, ‘Comparison of three survey techniques for locating Malleefowl mound,” Proceedings of 5th National Malleefowl Forum, Dubbo, New South Wales, pp. 174-182, September 2014.

[9] N. Parvin, M. Awrangjeb, M. Irvin, S. Florentine, M. Murshed and G. Lu, ‘Detection of Malleefowl Mounds from Point Cloud Data,” Proceedings of International Conference on Digital Image Computing: Techniques and Application (DICTA), Gold Coast, Australia, pp. 1-7, November 2021.

[10] M. Ester, H.P. Kriegel, J. Sander, Xiaowei X, ‘A density-based algorithm for discovering clusters in large spatial databases with noise,” Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, vol. 96, no. 34, pp. 226-231. August 1996.

[11] S.C. Tan, K.M. Ting, S.W. Teng, ‘Simplifying and improving ant-based clustering,” Procedia Computer Science, vol. 4, pp. 46-55, 2011, doi: 10.1016/j.procs.2011.04.006.

[12] S.C. Tan, K.M. Ting, S.W. Teng, ‘Reproducing the results of ant-based clustering without using ants,” Proceedings of IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, pp. 1760-1767, July 2006, doi: 10.1109/CEC.2006.1688520.

[13] W. Yao, J. Wu, ‘Airborne LiDAR for Detection and Characterization of Urban Objects and Traffic Dynamics,” Urban Informatics. The Urban Book Series. Springer, Singapor, pp. 367-400, 2021.

[14] M. Awrangjeb and C. S. Fraser, ‘Automatic segmentation of raw LiDAR data for extraction of building roofs,” Remote Sensing, vol. 6, no. 5, pp. 3716–3751, 2014.

[15] M. Awrangjeb, “Extraction of power line pylons and wires using airborne LiDAR data at different height levels” remote-sensing, vol. 11, no. 15, pp. 1798, 2019.

[16] S. Jeronimo, V. Kane, D. Churchill, ‘Applying LiDAR individual tree detection to management of structurally diverse forest landscapes,” Journal of Forestry, vol. 116, np. 4, pp. 336-346, 2018.

[17] V. Mitra, C. Wang, S. Banerjee, “LiDAR Signal Processing for Under-Water Object Detection,” Advances in Neural Networks, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 3497, pp. 556-561, 2005.

[18] R.Qi. Charles, S. Hao, M. Kaichun and J.G. Leonidas, ‘Pointnet: Deep learning on point sets for 3d classification and segmentation,” In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, pp. 652-660. 2017.

[19] R.Qi. Charles, S. Hao, and J.G. Leonidas, ‘PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” Advances in neural information processing systems, vol. 30, 2017.

[20] Z. Wang, Y. Xu, J. Yu, G. Xu, J. Fu and T. Gu, ‘Instance segmentation of point cloud captured by RGB-D sensor based on deep learning,” International Journal of Computer Integrated Manufacturing, vol. 34, no. 9, pp. 950-963, 2021.

[21] K.K. Al-Barazanchi, A.Q. Al-Neami and A.H. Al-Timemy, ‘Ensemble of bagged tree classifier for the diagnosis of neuromuscular disorders,” In Proceedings of the IEEE International Conference on Advances in Biomedical Engineering, Lebanon, pp. 1-4, 2017.