ISPRS Journal of Photogrammetry and Remote Sensing 224 (2025) 235–250

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Evaluating saliency scores in point clouds of natural environments by learning surface anomalies

Reuma Arav [a,b,*], Dennis Wittich [c], Franz Rottensteiner [c]

[a] Dept. of Geodesy and Geoinformation, TU Wien, Wiedener Hauptstrasse 8, Vienna, 1040, Austria
[b] Institute of Geomatics, University of Natural Resources and Life Sciences, Vienna (BOKU), Peter-Jordan-Strasse 82, Vienna, 1190, Austria
[c] Institute of Photogrammetry and GeoInformation, Leibniz University Hannover, Hannover, Germany

ARTICLE INFO

Keywords:
Salient object detection (SOD)
Anomaly detection
Geomorphological entities
Deep neural network

ABSTRACT

In recent years, three-dimensional point clouds are used increasingly to document natural environments. Each dataset contains a diverse set of objects, at varying shapes and sizes, distributed throughout the data and intricately intertwined with the topography. Therefore, regions of interest are difficult to find and consequent analyses become a challenge. Inspired from visual perception principles, we propose to differentiate objects of interest from the cluttered environment by evaluating how much they stand out from their surroundings, i.e., their geometric salience. Previous saliency detection approaches suggested mostly handcrafted attributes for the task. However, such methods fail when the data are too noisy or have high levels of texture. Here we propose a learning-based mechanism that accommodates noise and textured surfaces. We assume that within the natural environment any change from the prevalent surface would suggest a salient object. Thus, we first learn the underlying surface and then search for anomalies within it. Initially, a deep neural network is trained to reconstruct the surface. Regions where the reconstructed part deviates significantly from the original point cloud yield a substantial reconstruction error, signifying an anomaly, i.e., saliency. We demonstrate the effectiveness of the proposed approach by searching for salient features in various natural scenarios, which were acquired by different acquisition platforms. We show the strong correlation between the reconstruction error and salient objects. To promote benchmarking and reproducibility, the code used in this work can be found on https://github.com/rarav/salient_anomaly/releases/tag/v1.0.0 while the datasets are published on doi:10.48436/mps0m-c9n43 and 10.48436/fh0am-at738.

1. Introduction

Three-dimensional point clouds have become an essential tool for geoscientific studies. Everything within the natural environment is being documented and monitored: from millimetre-wide cracks, to centimetre-long blocks and metre-wide rivers (Telling et al., 2017; Tarolli and Mudd, 2020; Kyriou et al., 2021). The acquired point clouds provide a high resolution description of the landscape, enabling analyses that would otherwise be impossible. These datasets are characterized by a massive amount of unorganized points, which span over wide areas at different point spacing. The collected data comprise a diverse array of objects of interest with varying shapes and sizes, distributed throughout the dataset and embedded within the topography. Due to acquisition conditions, the data hold a significant amount of noise and uninteresting regions make up a larger portion of the point cloud (Arav et al., 2022a).

Studies have shown that focusing on important regions within the point cloud improves scene understanding (Alexiou et al., 2019; Liang et al., 2023). This can be accomplished through *visual saliency*, which is defined as the subjective quality that makes certain objects or regions stand out in their environment, capturing the observer's attention (Akman and Jonker, 2010). In 3D, saliency is defined as objects (or regions) that stand out from their surroundings, also geometrically. Common saliency approaches in 3D point clouds focus on small object models (e.g., Guo et al., 2018; Alexiou et al., 2019; Ding et al., 2019; Leal et al., 2019), where the point cloud is confined, resolution is approximately constant, and noise levels are often low. Studies that wish to extend the detection to larger scenes usually focus on urban environments. There, salient objects hold distinct features, so that first-order features, such as normal, height, or orientation are sufficient for saliency detection (Hao et al., 2019; Yun and Sim, 2016; Fan et al., 2022). However,

* Corresponding author at: Institute of Geomatics, University of Natural Resources and Life Sciences, Vienna (BOKU), Peter-Jordan-Strasse 82, Vienna, 1190, Austria.
E-mail addresses: reuma.arav@boku.ac.at (R. Arav), wittich@ipi.uni-hannover.de (D. Wittich), rottensteiner@ipi.uni-hannover.de (F. Rottensteiner).

https://doi.org/10.1016/j.isprsjprs.2025.03.022
Received 6 September 2024; Received in revised form 27 December 2024; Accepted 26 March 2025
Available online 12 April 2025

these approaches fail in natural environments, where entities transform smoothly into the background.

In this paper we introduce a new approach to estimate saliency in 3D point clouds of natural environments. To do so, we estimate anomaly probability within a surface. Based on the fact that landscapes are generally continuous and smooth, salient features will present an unexpected change in the surface. We propose to use a deep neural network to predict small parts of the landscape providing only a reduced amount of information. Then, we interpret the deviation between the actual and the predicted surface as a measure of saliency for that area. Specifically, we train a network by inputting the outer cells (a shell) of a voxelized region (voxel grid) and generating a predicted voxel grid as output. It is assumed that the shell contains all the required information to predict the surface described by the voxel grid, as long as the inner part is regular. However, whenever the inner part is irregular, the reconstruction error will be large, and thus will signify high saliency. We demonstrate the proposed approach in three real-world settings, that substantially differ one from the other. We show, both visually and quantitatively, the strong link between the reconstruction error and salient objects. Doing so, we propose a new approach for evaluating saliency in 3D point clouds, which, unlike current deep learning approaches, does not consider saliency detection as a classification problem. Therefore, it does not require pre-trained classifiers. By predicting the surface from the obtained point cloud, the proposed approach can detect saliency in open terrain datasets and is not limited to small objects. Furthermore, is can handle substantial data volumes, high noise levels, and irregular point distribution, all of which are inherent characteristic to 3D point clouds acquired by laser scanning platforms. To promote further study of saliency estimation algorithms, we release our source code (Arav and Wittich, 2023).

2. Related work

Saliency detection in 3D point clouds has been gaining popularity for several years as a preliminary process for various complex processing tasks. For example, Qin et al. (2023) use saliency to register multiple point clouds of an indoor scene, Laazoufi and Hassouni (2022) use salient points to evaluate point cloud quality, Liang et al. (2023) enhance point cloud models by reducing excessive non-salient points that obscure the overall shape of the model, and Hong et al. (2023) employ salient regions in data augmentation learning models for segmentation.

Saliency detection approaches in full 3D point cloud are quite rare. Nonetheless, there are many works that deal with saliency detection in RGB-D (colour and depth) images. There, saliency is found mostly based on RGB information, while the depth map is used to improve results. In recent years, most methods adopt deep learning models for the task. Chen et al. (2021b) and Zhou et al. (2021) differentiate between early, late, and middle aggregation approaches. In early aggregation models, both RGB and depth images are fused in the input level, and then a CNN-based network is used to extract the features for saliency detection (e.g., Zhang et al., 2020, 2021a). In late aggregation approaches, saliency cues are learned separately from the depth and colour channels before being fused to obtain the saliency map (e.g., Li et al., 2022; Sun et al., 2022; Chen et al., 2021a). For example, Chen et al. (2021a) learn the relevant cues for saliency detection from each channel, and then select cues that exist only in one channel. The saliency inference is carried out by fusing low- and high-level cues from both channels. Middle aggregation models try to combine both early and late aggregation approaches, so that learning is carried out in two phases. In the first phase, saliency features are obtained for each modality. In the second phase, they are fused to generate the final saliency map (Chen et al., 2021b; Zhou et al., 2021; Han et al., 2018; Zhang et al., 2021b). For instance, Zhou et al. (2021) first feed the depth and colour images into two learning networks to obtain corresponding multi-level feature representations. These representations are fused using an integration module, where a shared learning network enhances the features for saliency detection.

However, the works above capitalize on existing saliency approaches in colour images, while assuming a corresponding depth map. Yet, colour information in 3D point clouds is not always available, necessitating a greater emphasis on geometric features. Moreover, the data is unstructured, with varying point spacing, and in three dimensions, making such raster-based approaches inapplicable.

Doraiswamy et al. (2013) proposed to combine topological information with spatial one in order to define saliency. The topological persistence of a set of initial features was weighted in reference to their neighbourhood. In this framework, the saliency of a feature is a combination of the feature's topological persistence with that of its neighbourhood. Peng et al. (2021) also suggested to use feature topology as an indication for saliency. The data were first segmented and the topological relations between the segments (i.e., nested or parallel structures) were the indicator to its saliency. This was based on the observation that topologically complex regions are more salient than others. Both of these methods require an initial segmentation or feature selection to analyse the topology. It might be that due to that fact, none of these methods was tested on 3D point clouds. Works that define saliency particularly in point clouds are rare. Still, we divide them here to handcrafted and deep learning based approaches.

2.1. Handcrafted saliency approaches in 3D point clouds

Shtrom et al. (2013) were first authors to introduce saliency in point clouds which completely relies on geometric characteristics. The authors computed a fast point feature histogram descriptor (FPFH, Rusu et al. (2009)) and then evaluated its distinction from the local neighbourhood. A global rarity was then estimated by measuring the dissimilarity between every two points in the cloud. This approach was applied successfully to both small object models and urban environments in other works (e.g., Kobyshev et al., 2016; Hao et al., 2019). Tasse et al. (2015), Yun and Sim (2016) and also Ding et al. (2019) improved its computational efficiency by using cluster-wise comparison rather than a point-wise one. Other approaches for saliency detection proposed to use different metrics of local distinctness. Nonetheless, these were also based on normal computation and the distinction of the point's normal from its immediate surrounding. Wang et al. (2015) measured the difference of a point's normal from the dominant normal in the scene. Applied to roads scanned by mobile scanners, this approach is aimed specifically to highlight off-road objects. Guo et al. (2018) defined a point descriptor based on principal component analysis (PCA). The descriptor was composed of sigma-sets extracted from the covariance matrix of each point's normal and curvature. Arvanitis et al. (2022) defined salient points as those belong to non-flat surfaces. The flatness is determined by the covariance matrix eigenvalues of a local neighbourhood. Non-flat areas produce low eigenvalues that correspond to high saliency values. In such normal-based approaches, the assumption is that a salient feature is defined by an abrupt change in orientation. However, in natural environments this might not be the case. There, entities such as gullies, landslides, rockfalls, sinkholes, or cracks, are parts of the underlying surfaces. Such objects have intermediate borders, which gradually and continuously change from background to entity (Molenaar and Cheng, 2000; Liu et al., 2019). Therefore, though they differ from their surroundings, their borders are mostly vague and are hard to define (Molenaar and Cheng, 2000). To overcome this problem, Arav and Filin (2020) proposed a method that is attuned to detect vague objects as salient features. Instead of looking for an immediate change in the local surrounding, the authors suggest to look at a farther neighbourhood. Furthermore, to allow for more subtle objects, the authors do not only take the normal change into account, but also the change rate, i.e., the curvature. The advantage of this approach was shown in later works (Arav and Filin, 2022; Arav et al., 2022b) detecting salient objects in different types of natural

scenes, including a complete 3D scenario (i.e., a cave). Nonetheless, this approach would fail in cases of rough surfaces (e.g., riverbeds, alluvial fans). There, the difference between a point and its wider surrounding is high, leading to an increased sensitivity in detection. Moreover, outliers (i.e., measurement noise) will also be highlighted, as their normals and curvatures completely differ from their surroundings.

The review has shown that handcrafted approaches for saliency estimation evaluate how much a point differs from its surrounding (aka. centre-surround principle Itti et al., 1998), mostly focusing on the difference in normal direction. In such schemes, a larger context of salient features is missing, leading to high sensitivity to local variations.

2.2. Deep learning-based saliency approaches in 3D point clouds

To the best of our knowledge, only a few approaches were proposed for saliency evaluation in point clouds using deep learning. These tend to use pre-trained models and are mostly in the context of shape recognition and classification. Zheng et al. (2019) assert that salient points explicitly explain which points are key for model recognition. The authors assume that points that lie on the object's borders contribute more to shape recognition than those that lie on its inner surface. Therefore, they suggest that elimination of unimportant points or their movement towards the object's inner surface are equivalent. Under this assumption, salient points are marked by the change in prediction loss using a pre-trained classifier for shape recognition. The change in prediction loss is approximated by the gradient of the loss when shifting points to the centre of the object's point cloud. These gradients were interpreted as saliency scores. Another semi-supervised approach was proposed by Jiang et al. (2023). The authors use objects that were previously classified in order to learn the saliency. This is carried out in two main branches: a classification branch, which uses category labels for feature extraction, and a saliency branch that uses a multi-scale point cluster matrix to provide coherent saliency regions. Both approaches target point clouds of objects whose category labels are known. Within the natural environment, where objects may be restricted only to one region or may appear only a few times, training data for classification may be difficult to acquire. Moreover, manual labelling which marks *salient* and *non salient* features in scenes as large as point clouds of natural environment are, is not only time-consuming, but also prone to perception bias and degrades the detection accuracy (Hillier et al., 2014; Scheiber et al., 2015; Vinci et al., 2016). Therefore, the aforementioned methods cannot be applied to point clouds of natural environment. To overcome this problem, we propose a new approach to highlight salient regions that is independent of previous classification. The detection is driven by the notion that in natural environments salient object are in a way an anomaly in the general surface.

3. Methodology

We seek to highlight salient features in point clouds, focusing on datasets that document natural environments (i.e., non-urban scenes). Following the notion that salient features are a sudden change in the surface, we assume that they will present an irregularity at that location. Therefore, we consider the task of highlighting saliency as marking anomalies in the scene. To do so, we first train a deep neural network to reconstruct the surface from a reduced subset of the data. Then, we reconstruct the surface and evaluate the reconstruction error. This error is interpreted as the saliency score, since the reconstruction error will be larger in irregular regions. In this way, we highlight salient features in 3D point clouds where external information is used only to find the best hyper-parameters of the method (i.e., hyper-parameters tuning).

We begin with the details of our proposed method to mark salient regions (Section 3.1). This section also includes a formal definition of the problem and the notations used in this work. Then, we outline the network architecture (Section 3.2), followed by details concerning the loss function and training procedures (Section 3.3). Lastly, we describe how saliency scores are estimated (Section 3.4).

3.1. Saliency estimation in 3D point clouds by anomaly detection

Let P be a point cloud, defined as a set of N 3D points. These compile the main input to the method. Additionally, as an input, we introduce two subsets of P: H, which is composed of points that are expected to have high saliency scores; and L, composed of points that are expected to have lower saliency scores. These subsets are required to tune the hyper-parameters of the method. Note that $H \cup L \neq P$. They are only samples of each group H and L. The output of the method is a saliency map, where each point $p_i \in P$ has a saliency score ξ_i.

The saliency score is in fact an interpretation of the reconstruction error. It is obtained by a reconstruction network R for a point's surrounding region. This network is pretrained on random regions extracted from P to enable a reconstruction of the surface recorded by the point cloud. This is based on the assumption that salient regions are rare, and therefore, the network will not learn them, but it will rather learn regular surfaces. This approach has a major advantage, as we do not require any manually generated reference to train R. Instead, we use arbitrary sub-regions of P for the task.

To evaluate the reconstruction error as a saliency score, one has to formulate the reconstruction task in a way that R could reconstruct the surroundings of a point, as long as these surroundings are regular. Yet, if the surroundings are irregular, the reconstruction should be incorrect, yielding a high reconstruction error, i.e., a high saliency score. To do so, we use a voxel-based representation of the point cloud. Then, we formulate the reconstruction task to predict the inner part of a voxel grid based on its outer voxels (the grid's shell).

Let V_i be the representation of a region in P in terms of a voxel grid of size $n \times n \times n$ that contains the surrounding region of p_i, such that V_i is centred at p_i. The side-length of each voxel cell in V_i is parameterized by w, resulting in a volume of $w \times w \times w$ for each voxel cell and a total volume of $(w \cdot n) \times (w \cdot n) \times (w \cdot n)$ for V_i. The value of a voxel $V_{i,(\hat{x},\hat{y},\hat{z})}$ at voxel coordinates $(\hat{x}, \hat{y}, \hat{z})$ in V_i corresponds to the number of 3D points in that cell.

Next, we introduce S_i. This is a modified version of V_i, where the values in the inner cells are set to zero. Consequently, S_i contains only the information from the shell of V_i. In particular, the value of the voxel cell $S_{i,(\hat{x},\hat{y},\hat{z})}$ is $V_{i,(\hat{x},\hat{y},\hat{z})}$ if $\min(\hat{x}, \hat{y}, \hat{z}) \leq m$ or $\max(\hat{x}, \hat{y}, \hat{z}) \geq n - m - 1$, and zero otherwise. Here, m denotes the thickness of the shell, i.e., how many voxels compile the shell. Using this notation, the reconstruction task is carried out by R. The network predicts the values of a voxel grid $\hat{V}_i$ based on the shell S_i, such that $\hat{V}_i$ is similar to V_i. To measure the similarity between $\hat{V}_i$ and V_i, we introduce a function $\mathcal{R}(\hat{V}_i, V_i)$ that measures the reconstruction error. The selection of the reconstruction error function $\mathcal{R}$ is be discussed in Section 3.3.

The overall training scheme of our proposed method is shown in Fig. 1. To train the reconstruction network R, we randomly select points from the cloud P, then voxelize their surrounding, resulting in voxel grids V_i. Based on these grids we generate corresponding shells S_i. Then, the parameters of R are obtained by minimizing the reconstruction error $\mathcal{R}(\hat{V}_i, V_i)$. Eventually, the saliency score ξ_i for a point p_i is estimated by the reconstruction error $\mathcal{R}(\hat{V}_i, V_i)$ for each point in P using the trained network. It should be mentioned that as the network is trained to reconstruct 'regular' surfaces, and in each scene this 'regularity' is defined differently, training has to be conducted for each new scene.

3.2. Architecture

To perform the reconstruction task, we use a 3D convolutional neural network (CNN) as the reconstruction network R. The architecture is shown in Fig. 2. R takes a shell S_i as an input and outputs the values of a reconstructed voxel grid $\hat{V}_i$. All tensors S_i, V_i and $\hat{V}_i$ have the same shape, which is $n \times n \times n$, where n is the side-length of the voxel grid.

The architecture of R follows the encoder–decoder scheme with skip connections, similar to the U-Net architecture (Ronneberger et al.,

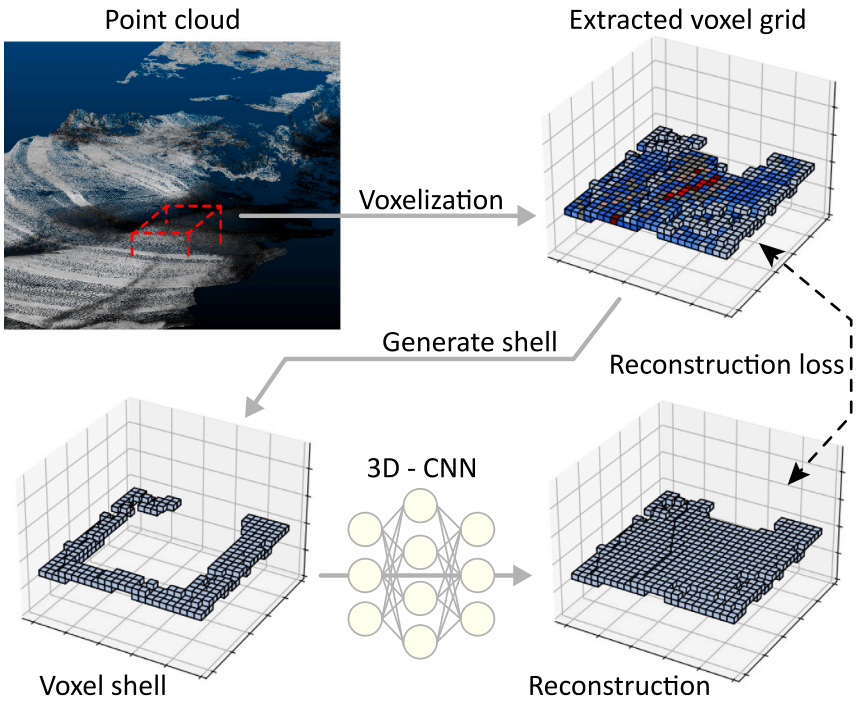

Fig. 1. Overview of the training scheme. We generate random voxel grids from a set of given point-clouds. Next, the shells are generated by setting the inner voxels to zero. The task of the CNN is then to reconstruct the original input. Note that the extracted voxel grid is coloured by the number of points in each cell. This information is used to calculate a weighted reconstruction loss. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2015). In this layout the spatial size of the feature maps is halved in each stage on the decoder and doubled in the decoder (cf. Fig. 2). An illustration of the architecture is shown in Fig. 2 with the corresponding layers described in Table 1.

The capacity of the network is parameterized by the parameter f, which describes the number of feature maps in base resolution. In each new stage, the number of feature maps is doubled. For example, with $f = 24$ the network has about 354K learnable parameters. Independent from f, the network has a theoretical perceptive field of $49 \times 49 \times 49$ voxels, which was found to be suitable in preliminary experiments.

All convolutional layers use $3 \times 3 \times 3$ kernels and a leaky rectified linear unit (Xu et al., 2020) as the activation function, except for the very last layer, where the sigmoid function is applied instead. Applying the sigmoid function to the output of the last convolution yields values in a range of (0, 1), which are interpreted as the probability of the respective voxel to be occupied by the surface. The downsampling and upsampling operations are performed by nearest neighbour interpolation along all three spatial dimensions with scaling factors of 0.5 and 2.0, respectively.

Table 1
Layers of the architecture of $\mathcal{R}$. 3D-Conv: 3D Convolutional layer. LRL: Leaky ReLU. BN: 2D batch normalization; Cat(L_X): Depth-wise concatenation of the output of layer L_X and the current layer. side-length: Output dimensions. w is the side-length of the input shell S_i.

	Layer name	Type	Side-length	Num. chn.
Encoder		Input layer	n	1
	Conv-(1,2)	3D-Conv, LRL	n	f
	Dw-1	Downsample	$n/2$	f
	Conv-(3,4)	3D-Conv, LRL	$n/2$	$2f$
	Dw-2	Downsample	$n/4$	$2f$
	Conv-(5,6)	3D-Conv, LRL	$n/4$	$4f$
	Conv-7	3D-Conv, LRL	$n/4$	$2f$
Decoder	Up-1	Upsample, Cat(6)	$n/2$	$(2+2)f$
	Conv-8	3D-Conv, LRL	$n/2$	$2f$
	Conv-9	3D-Conv, LRL	$n/2$	f
	Up-2	Upsample, Cat(3)	n	$(1+1)f$
	Conv-10	3D-Conv, LRL	n	f
	Conv-11	3D-Conv, Sigmoid	n	1

3.3. Loss function and training

The loss function used in this work is a variant of the dice-loss, which is applied in classification problems (Sudre et al., 2017). This loss is well suited for our problem because it insusceptible to data imbalance. In our case, such imbalance occurs since there is much more empty space than occupied voxels in the reconstruction task.

Let V_i^β be a binary version of a voxel grid V_i, where

$$V^\beta_{i,(\hat{x},\hat{y},\hat{z})} = \begin{cases} 1 & \text{if } V_{i,(\hat{x},\hat{y},\hat{z})} \leq t_b, \\ 0 & otherwise. \end{cases}$$

Here, t_b is a threshold value for the minimum number of 3D points in each voxel cell so it will be considered occupied. In our experiments, we set $t_b = 2$. We assume that voxels which contain only a single 3D point are more likely to represent noise. Therefore, this is a measure aimed to deal with noise, so that only voxels with more than two points are regarded.

Another way to accommodate for noise in the data is by introducing a modified version of the dice-loss. This version uses weights at voxel level. As voxels that hold only one point may distract the regression model, they should be ignored. To this end, a weight tensor W_i is computed for each voxel grid V_i, where $W_{i,(x,y,z)} = 0$ if $V_{i,(x,y,z)} = 1$ and $W_{i,(x,y,z)} = 1$, otherwise.

The basic reconstruction error is

$$\mathcal{R}(\hat{V}_i, V_i^\beta) = 1 - \frac{\mathcal{I}(\hat{V}_i, V_i^\beta)}{\mathcal{U}(\hat{V}_i, V_i^\beta)}, \tag{1}$$

with the intersection term, $\mathcal{I}$,

$$\mathcal{I}(\hat{V}_i, V_i^\beta) = \sum_{\hat{x}=0}^{n-1}\sum_{\hat{y}=0}^{n-1}\sum_{\hat{z}=0}^{n-1} \hat{V}_{i,(\hat{x},\hat{y},\hat{z})} \cdot V^\beta_{i,(\hat{x},\hat{y},\hat{z})}, \tag{2}$$

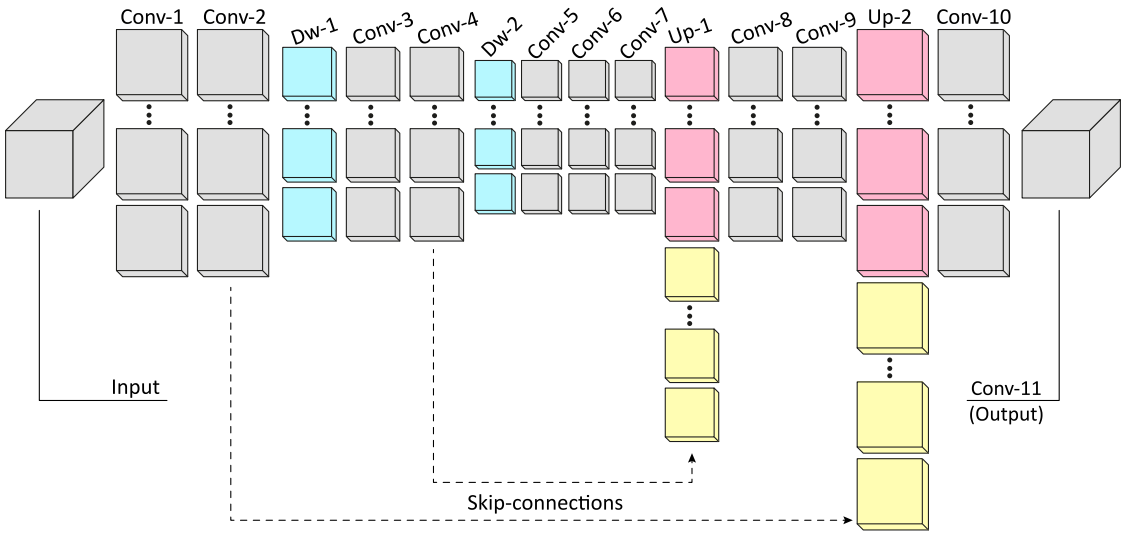

Fig. 2. Illustration of the variational auto-encoder used to reconstruct the surface based on the information in the shell of the voxel grid representation. Blue: Output of downsampling. Red: Output of upsampling. Yellow: Concatenated feature tensors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and the weighted union term, $\mathcal{U}$,

$$\mathcal{U}(\hat{V}_i, V_i^{\beta}) = \sum_{\hat{x}=0}^{n-1}\sum_{\hat{y}=0}^{n-1}\sum_{\hat{z}=0}^{n-1} \max(\hat{V}_{i,(\hat{x},\hat{y},\hat{z})}, V^{\beta}_{i,(\hat{x},\hat{y},\hat{z})}) \cdot W_{i,(\hat{x},\hat{y},\hat{z})}. \tag{3}$$

Using this formulation of the reconstruction error, the overall training loss $\mathcal{L}$ is

$$\mathcal{L} = \frac{1}{B}\sum_{b=0}^{B} \mathcal{R}(\hat{V}_b, V_b^{\beta}), \tag{4}$$

where B is the batch size and b is the index of a sample in the batch. Note that in preliminary experiments, we found this loss to outperform other loss definitions. Particularly, we compared to minimizing the mean squared error and a variant of the dice-loss without weighting.

To train the network, its parameters are randomly initialized and then iteratively updated using ADAM optimizer with a learning rate of $\lambda = 0.0001$ and hyper-parameters $\beta_1 = 0.0$ and $\beta_2 = 0.999$. During training, we sample a batch of B voxel grids (V_i) from the point cloud. Data augmentation is performed by randomly rotating the point cloud along the height axis before extracting each voxel grid. This is done after selecting a random point to be the centre of the voxel grid. The grid position is then frozen and the point cloud is randomly translated along the height axis before the voxelization step. In preliminary experiments we found that this step improves the trained models substantially, with respect to the reconstruction capability. Models that were trained without this augmentation step tended to be biased towards predicting occupied voxels in the centre of the voxel grid. Following the data augmentation step, the voxel grids in the batch are binarized.

Next, the shells (S_i) are created as described in Section 3.1. These are presented to the network resulting in a predicted voxel grid ($\hat{V}_i$) for each shell in the batch. Using Eq. (1) the reconstruction error for each sample is calculated. The average reconstruction error over all samples in a batch corresponds to the reconstruction loss of the batch. The parameters of the network are then iteratively updated using ADAM optimizer to minimize the reconstruction loss. Training is stopped when the performance on a validation subset does not increase for n_{ST} iterations. The parameter set resulting in the highest validation performance is used for the inference. The performance measures are described in Section 4.4.

3.4. Inference

After training, R is used to predict the voxel grid $\hat{V}_i$ for the extracted shell S_i of each point $p_i \in P$. The reconstruction error $\mathcal{R}(\hat{V}_i, V_i)$ is then interpreted as a measure of saliency for p_i. Eventually, a saliency map is received, where each point has a saliency score of

$$\xi_i = \mathcal{R}(\hat{V}_i, V_i). \tag{5}$$

4. Test setup

4.1. Experiment setup

Experiments were carried out using an AMD Ryzen Theadripper 1900X 8-core processor machine with a CPU memory of 32 GB and an NVIDIA GeForce RTX 2080 Ti GPU.

Network parameters were optimized according to Section 3.3. In all datasets, the batch size B was set to 16 voxel grids per batch. The classifier was evaluated on the validation sets every 1000 training iterations. The hyper-parameter n_{ST}, which is the training stop parameter, was set to 10,000 iterations. The shell size m was set to 3 for all datasets.

4.2. Datasets

To demonstrate the proposed method we used three datasets that differ by scene, acquisition platform, extent, number of points, point spacing, etc. In the following, we characterize each dataset. Table 2 provides a summary of the key characteristics.

Dataset #I. An airborne laser scan of an alluvial fan along the Dead Sea coast, Israel (open to the public Geological Survey of Israel and Arav, 2013). It holds above 1.5 million points, at 0.5 m point spacing. The scanned surface is relatively flat, punctured by sinkholes and dissected by gullies (Fig. 3a). Being an airborne laser scan, some overlapping scanlines exist, which leads to a change in point density in some regions.

Dataset #II. An airborne topo-bathymetric laser scan of a 750 m long section of a meandering river (Pielach River, Austria; Fig. 3b). This scanner is characterized by its elliptic scanning pattern, which affects the average point density throughout the scan (Arav et al., 2024). This dataset holds over 50 million points. Focusing on the river, vegetation was removed using the hierarchic robust interpolation method (Pfeifer and Mandlburger, 2018) as implemented in OPALS (Pfeifer et al., 2014).

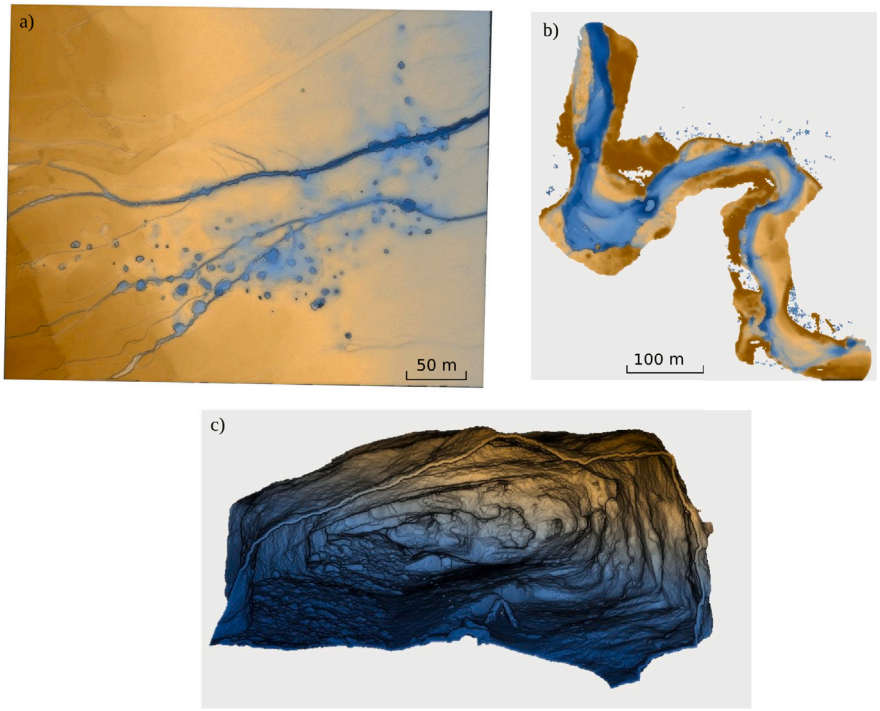

Fig. 3. Datasets analysed in the study. (a) Airborne dataset (Dead Sea Coast); (b) UAV-borne dataset (Pielach River); (c) Terrestrial dataset (Traisenbacher cave) The entrance to the cave is 4.7 m long and 2.6 m high. Colours refer to elevation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Acquisition characteristics of the analysed datasets.

Dataset #	Scanning platform	Scanner type	PRR[a] [kHz]	Mean point spacing [m]	No. of points
I	Airborne	Optech ALTM 2050	100	0.5	1,632,928
II	Airborne	Riegl VQ880-GH	200	0.075	50,813,569
III	Terrestrial	Riegl VZ2000	550	0.01	786,267

[a] Pulse repetition rate.

Dataset #III. A terrestrial laser scan of a small cave, the Untere Traisenbacher Höhle, Austria (Fig. 3c; open to the public Wimmer and Oberender, 2022). Representing a cave, this dataset is fully three-dimensional, which makes it a challenging scene to analyse (Arav et al., 2022b). A single scanning position was used here. Therefore, on the one hand there are no overlapping scanlines. On the other hand, the scan features occlusions, as there were no additional positions to mitigate them. These occlusions are characteristic to terrestrial laser scans in general, and in cave measurements in particular. Hence, this scene is a good example for 3D terrestrial scan.

4.3. Validation and test subsets

In each dataset, we specify two types of subsets: a validation subset (D) – for stopping the training process and for tuning the hyper-parameters; a test subset (T) – for testing and comparison purposes. Each subset is divided into 'salient' (H) and 'non-salient' (L) regions. These correspond to the expected regions that should have higher and lower saliency scores, respectively.

From each dataset a different number of subsets was extracted, depending on the scene. Non-salient areas were selected after visual inspection, to minimize the existence of salient regions within them. However, as delineation was done manually, the subsets still included some small parts of the other class (i.e., 'salient' in 'non-salient' regions, and vice versa). Nonetheless, the analysis only compares mean values of the same regions, so that inaccuracies in sampling are insignificant.

Since saliency estimation is a subjective measure (Akman and Jonker, 2010), we describe below which objects/areas we expect to have higher saliency scores in each dataset. Accordingly, we define the minimal object size. The voxel size is then set to be half of the minimal object size. Table 3 summarizes these features.

Dataset #I. Salient areas are defined either as sinkholes or as parts of gullies (e.g., Fig. 4a–b). The sinkholes typically have 4–20 m diameter, while gullies are 2–9 m wide. Therefore, the minimal size is 2 m and the consequent voxel size is set to 1 m (Table 3). As for the non-salient, these reflect the fan surface (Fig. 4c). A total of nine regions were extracted as salient areas and nine as non-salient ones.

Dataset #II. Salient features are defined as the riverbanks as well as objects on the riverbed that are larger than 0.3 m (e.g., driftwood, stone blocks). Accordingly, the voxel size is set to 0.15 m (Table 3). Three areas with stone blocks, which were extracted in previous works (Mandlburger et al., 2015), were used as 'salient' subsets (Fig. 5). Of these, two were chosen for testing and one for validation. The low number of extracted regions is a result of the complexity of the terrain. Non-salient regions were chosen along the river and reflect the riverbed which has varying surface roughness (Fig. 6).

Dataset #III. Salient features refer to niches and pockets in the cave's walls and ceiling, as well as to some ledges and objects on the floor, with a minimal size of 0.1 m. Consequently, the voxel

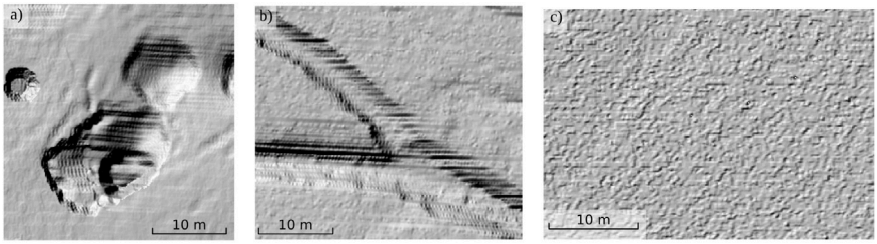

Fig. 4. Examples of subsets chosen for validation in the dataset #I. (a) and (b) — salient regions; (c) non-salient. To improve visualization, the datasets presented here are hillshade reliefs of the point clouds. Note the different scales.

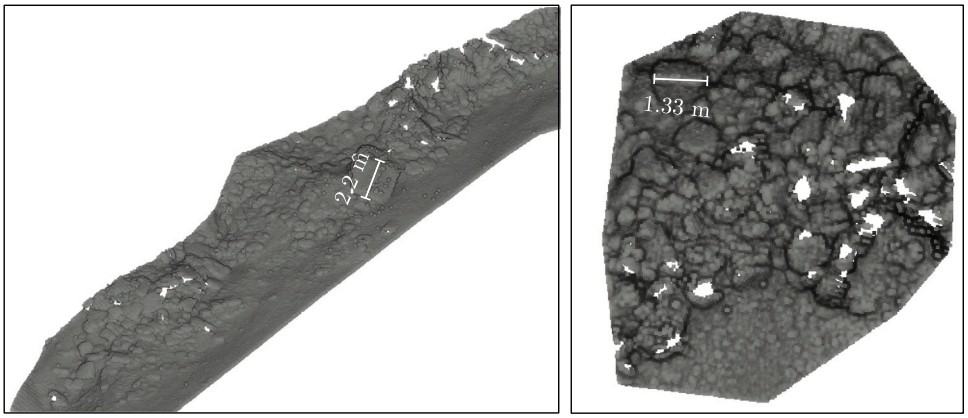

Fig. 5. Dataset #II. Examples of high salient score subsets (H).

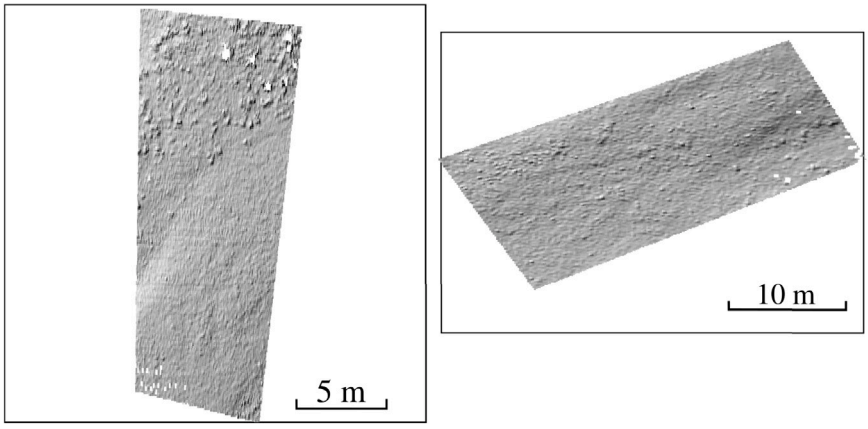

Fig. 6. Dataset #II. Examples of non salient subsets (ν). Note that the surface is not smooth but has some roughness (small stones). To improve visualization, the datasets presented here are hillshade reliefs of the point clouds.

size was set to 0.05 m (Table 3). To provide well-distributed subsets, both salient and non-salient subsets were chosen from the walls, the ceiling, and the floor. While salient subsets were chosen to include niches and blocks (Fig. 7a), non-salient subsets were focusing on the walls and ceiling that did not include any apparent niches (Fig. 7b).

4.4. Evaluation metrics

Measuring the performance of saliency scores is difficult. This is because the success rate cannot be easily quantified and may depend on user's understanding of the data (Hillier et al., 2014; Scheiber et al., 2015; Vinci et al., 2016). Moreover, since we use saliency as a relative measure within the dataset, it is impossible to compare values of one method to another. In most reviewed literature, saliency was used as a preliminary step for other analyses (e.g., Laazoufi and Hassouni, 2022; Liang et al., 2023). Then, the quantitative quality was measured according to the success rate of the procedures that follow. For example, Tinchev et al. (2021) assessed the registration quality, which was carried out based on keypoint detection using estimated saliency. Other works were comparing the results to existing benchmarks (e.g., Fan et al., 2022). Such a benchmark does not exist in our case. Therefore,

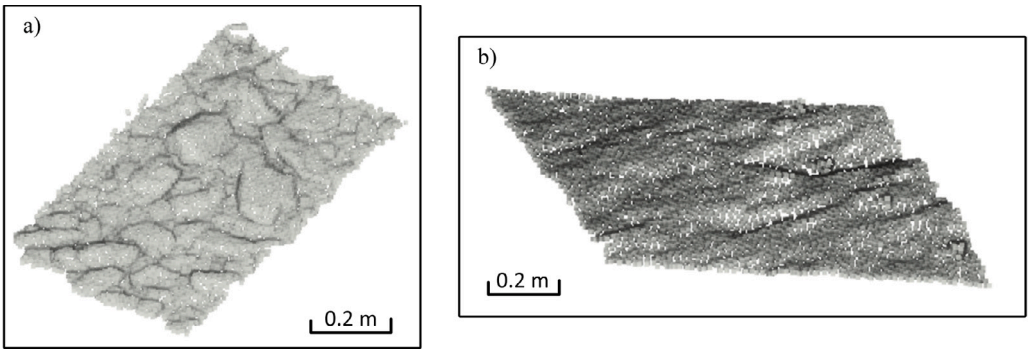

Fig. 7. Dataset #III. Examples of (a) salient (H) and (b) non-salient (ν) subsets. Note the roughness of the cave's surface (b).

Table 3
Defined salient features in each dataset, minimal object and voxel sizes, as well as number of test and validation subsets.

Dataset #	Predefined salient features	Min. object size [m]	Voxel size [m]	No. of test sets		No. of validation sets	
				H	L	H	L
I	Gullies, sinkholes	2	1	6	6	3	3
II	Riverbanks, stone blocks	0.3	0.2	2	3	1	3
III	Boulders, niches, pockets	0.1	0.05	4	4	3	3

we propose a saliency ratio for quantitative evaluation in addition to the visual inspection of the results.

Using the subsets D and T defined in Section 4.3, we define the saliency ratio $\hat{r}$. In particular, we define the ratio

$$\hat{r}_D = \frac{\bar{\xi}_{D,H}}{\bar{\xi}_{D,L}}, \tag{6}$$

with $\bar{\xi}_{D,H}$ the mean saliency score for points with a high expected saliency scores and $\bar{\xi}_{D,L}$ the mean saliency score for those with a low expected saliency, both in the validation subsets. Similarly, for the final testing of the method, we define

$$\hat{r}_T = \frac{\bar{\xi}_{T,H}}{\bar{\xi}_{T,L}}, \tag{7}$$

using the test subsets (T) instead of the validation (D).

Ratios that are larger than 1 suggest that the mean estimated saliency scores in H is higher than those in L, which is the expected result. As these ratios approach 1, the difference in estimated scores between salient and non-salient regions decreases. That is to say, the distinction between the two regions decreases. When the ratio is smaller than 1, the saliency was not estimated correctly, as regions that are expected to be with lower values yielded higher ones, and vice versa.

The metric $\hat{r}_D$ is used to tune the hyper-parameters of the method. The metric $\hat{r}_T$, which assesses the performance on the test subsets, is used to compare the method to existing approaches.

4.5. Baseline approaches

To compare our method to state-of-the-art, we used the following two baseline methods:

4.5.1. Plane-based approach

Given that our methodology hinges on reconstructing topography surfaces that may exhibit local planarity, a planar reconstruction is worth examining. Therefore, to highlight the merits of our learning approach, we advocate for its comparison against a plane-based anomaly search. With this in mind, we propose an alternative strategy to reconstruct the core of a voxel grid V_i by leveraging its shell S_i. At the heart of this method is the concept of fitting a plane to the voxels within the shell and subsequently projecting this plane onto the voxel grid, resulting in the reconstructed grid $\hat{V}_i$.

To find the best-fit plane for the shell voxels, we commence by computing the coordinates covariance matrix of each of the voxels that lie on the shell. The eigenvector corresponding to the smallest eigenvalue of this matrix provides us with the normal vector $\mathbf{n}$ of the optimal plane. Combined with the distance d from the origin, this establishes the plane equation in 3D space.

For any voxel $(\hat{x}, \hat{y}, \hat{z})$ within this space, its perpendicular distance $d_{p_{\hat{x},\hat{y},\hat{z}}}$ from the plane is derived from the plane's equation as: $d_{p_{\hat{x},\hat{y},\hat{z}}} = |\mathbf{n} \cdot (\hat{x}, \hat{y}, \hat{z}) - d|$. Here, $\mathbf{n} \cdot (\hat{x}, \hat{y}, \hat{z})$ represents the dot product between the normal vector and the voxel.

To represent the plane in the voxel grid, any voxels where $|d_{p_{\hat{x},\hat{y},\hat{z}}}| < t_d$, with $t_d = 0.5 \cdot w$ (with w the voxel side length) are assigned a value of one. By processing each voxel in this manner, the resultant grid is the reconstructed voxel grid $\hat{V}_i$. Then, the loss is computed by Eq. (1) and the saliency is estimated by Eq. (5). This way, the plane-based reconstruction is in fact a simplified comparative to our primary approach.

The voxel grid sizes to which the plane was fitted were chosen according to the those used in the proposed method, i.e., $n = 16,\ 24$ and 32.

4.5.2. Handcrafted saliency estimation

We use the handcrafted saliency proposed in Arav and Filin (2022) as another baseline method. This is because, to the best of our knowledge, it is the only point cloud based saliency estimation method that is attuned for natural environments. It is based upon the assumption that when dealing with topography distinctness would not be apparent in the immediate surroundings of a point. Therefore, it uses a weighting function that gives lower weights to nearby points and higher weights to more distant ones. To do so, the size of the surroundings and the minimal object size are set. Here, we set these according to Table 3, where the voxel size corresponds to the size of the surroundings. The saliency is then evaluated according to the deviation in surface normals and curvature within the defined surrounding. It is calculated as

$$\xi_i = 2 - \left[\exp\left(-d\mathbf{n}(p_i)\right) + \exp\left(-d\kappa(p_i)\right)\right], \tag{8}$$

with $d\mathbf{n}$ and $d\kappa$ are the sum of deviations in normal and curvature within the defined surroundings.

Table 4
Dataset #I. Saliency ratio (average and standard deviation over 5 runs) for the validation subsets ($\hat{r}_D$). It can be seen that for all combinations, saliency scores are higher at salient regions than non-salient ones. This implies that the proposed method highlighted salient regions correctly. f is the number of features in base layer and n is the voxel grid size length, represented by the number of voxels.

n	f		
	8	16	32
16	**2.51** ± 0.08	2.31 ± 0.04	2.33 ± 0.03
24	2.35 ± 0.04	2.29 ± 0.03	2.30 ± 0.01
32	2.20 ± 0.03	2.21 ± 0.01	2.24 ± 0.03

Table 5
Dataset #II. Saliency ratio (average and standard deviation over 5 runs) for the validation subsets ($\hat{r}_D$). It can be seen that for all combinations, saliency scores are higher at salient regions than non-salient ones. This implies that the proposed method highlighted salient regions correctly. f is the number of features in base layer and n is the voxel grid size length, represented by the number of voxels.

n	f		
	8	16	32
16	2.42 ± 0.03	2.40 ± 0.01	2.41 ± 0.03
24	2.48 ± 0.02	2.49 ± 0.02	**2.52 ± 0.08**
32	2.36 ± 0.01	2.39 ± 0.02	2.47 ± 0.05

Table 6
Dataset #III. Saliency ratio (average and standard deviation over 5 runs) for the validation subsets ($\hat{r}_D$). For all combinations the ratio values are close to 1, implying that the difference between estimated salient and non-salient values is small. Despite that, the estimated ratios are still larger than 1, meaning that the method evaluated salient regions correctly. f is the number of features in base layer and n is the voxel grid size length, represented by the number of voxels.

n	f		
	8	16	32
16	1.12 ± 0.04	1.18 ± 0.02	**1.19 ± 0.01**
24	1.15 ± 0.02	1.16 ± 0.03	1.18 ± 0.02
32	1.14 ± 0.04	1.16 ± 0.01	**1.19 ± 0.02**

Table 7
Training time (in minutes) for dataset #II. f is the number of features in base layer and n is the voxel grid size length, represented by the number of voxels.

n	f		
	8	16	32
16	8	5	8
24	22	17	40
32	20	26	70

4.6. Experiments description

For each dataset, we first performed a tuning for two hyper-parameters: the number of features in base resolution, f, and the voxel grid side length, n. We focused on these two parameters as they are considered the most important parameters of the method. We test their effect and discuss the saliency evaluation results achieved when using different combinations of the two. In all experiments, we used magnitudes of 8, 16, and 32 for f, and 16, 24, and 32 for n. The network was trained five times in each combination. After training, saliency scores were evaluated for the validation subsets. The saliency ratio $\hat{r}_V$ (Section 4.4) was then computed. Eventually, the mean saliency ratio and its standard deviation over the five runs were evaluated. Then, based on the best achieved results, we evaluated saliency scores for the entire dataset.

For each dataset, saliency scores were also evaluated using the baseline methods (Section 4.5). The comparison is carried out by evaluating the saliency ratio for the test subsets T for all applied methods. These, together with the visual impression of the saliency maps of the entire scene, enabled an evaluation of the saliency results.

5. Results and discussion

5.1. Hyper-parameters tuning

Tables 4–6 present the average saliency ratio results over five tests at each combination and for datasets #I, II, and III, respectively. It can be seen that in each dataset, the saliency ratios using the different hyper-parameters are similar. These range between 2.2–2.5 in dataset #I; 2.36–2.52 in dataset #II; and 1.12–1.19 in dataset #III (Tables 4, 5, and 6, respectively). Additionally, it can be seen that in all three datasets and for all combinations of f and n the saliency ratio is larger than 1. This indicates that regions which are defined as salient have higher saliency scores than the non-salient ones, as expected. However, a homoscedastic t-test did not show statistical distinction between 'salient' and 'non-salient' at 85% probability for the validation subsets.

To better understand the effect of each hyper-parameter on the saliency map, we visually examine the results achieved when one parameter is fixed and the other changes. We begin by testing the effect of the number of feature maps in base resolution (f). To do so, we fixed the size of the voxel grid n at the size which yielded the highest $\hat{r}_D$. Fig. 8 shows the results for $n = 16$ using the validation subsets for dataset #I. The effect of f is mostly seen in the non-salient regions. There, the least regions are being marked with high saliency scores when $f = 16$. This is because the number of features dictates the capacity of the network to reconstruct the surface. Too few features in base resolution will lead to a larger discrepancy from the original point cloud, and thus to higher saliency scores in non-salient regions (e.g., $f = 8$). On the other hand, too many features will lead to overfitting. Then, the reconstructed surface will deviate from the original cloud and result in incorrectly estimated high saliency scores ($f = 32$ in both Fig. 8). The number of features, however, may differ from one dataset to another, depending on the scene's surface. Therefore, it has to be tested for each dataset individually.

Similarly, we examined the effect of the voxel grid size, n, by fixing f with the number that achieved the highest ratio. Fig. 9 presents an example of parts from the validation subset in dataset #III, where $f = 32$ and $n = 16, 24$, and 32. It shows that it mainly affects the extent of the regions that receive higher saliency scores. The larger n is, the larger the inferred area, and thus the discrepancy from the original point cloud is larger, leading to less localized marking. Therefore, as the grid size increases the highlighted area increases as well.

It should be noted, however, that as the saliency ratio suggests, there are hardly any visual differences between the saliency maps generated by different hyper-parameters.

Between datasets, it can be seen that while dataset #I yielded the largest $\hat{r}_D$, dataset #III yielded the lowest. This fact can be attributed to the complexity of the analysed surfaces. In dataset #I the terrain is quite smooth and almost planar; dataset #II features a rougher but still mostly planar terrain; and dataset #III is composed of non-planar and mostly uneven and rough surfaces. This means that as the surface becomes less smooth (i.e., with higher surface variability), the network's ability to reconstruct the surface decreases, and thus the difference between $\bar{\xi}_H$ and $\bar{\xi}_L$ decreases.

Table 7 details the training time (in minutes) for dataset #II, which held the largest number of points. It can be seen that as the number of feature maps and/or grid size increases, so does the training time. This is an expected result, as one needs more computational power for larger grid sizes and more feature maps. However, note that in most cases the training time using $f = 16$ was the lowest. This is probably because the stopping criteria was fulfilled faster in this case.

5.2. Saliency estimation

The hyper-parameters used for the saliency evaluation for the entire datasets were those that produced the highest saliency ratio in

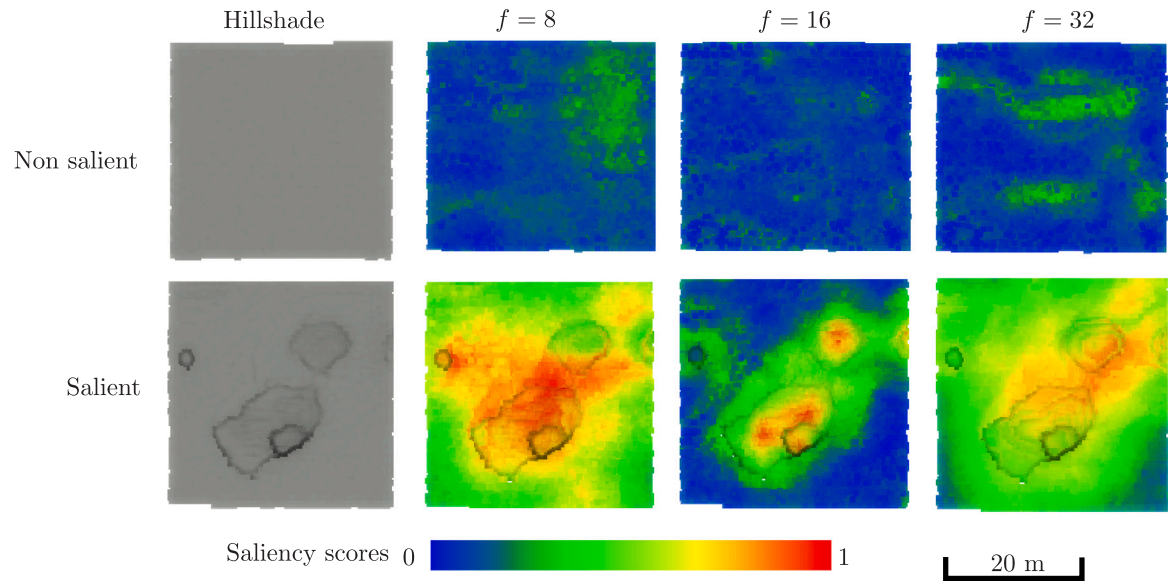

Fig. 8. Dataset #I. Saliency scores estimated for the validation subsets with $n = 16$ and different numbers of feature maps in base resolution.

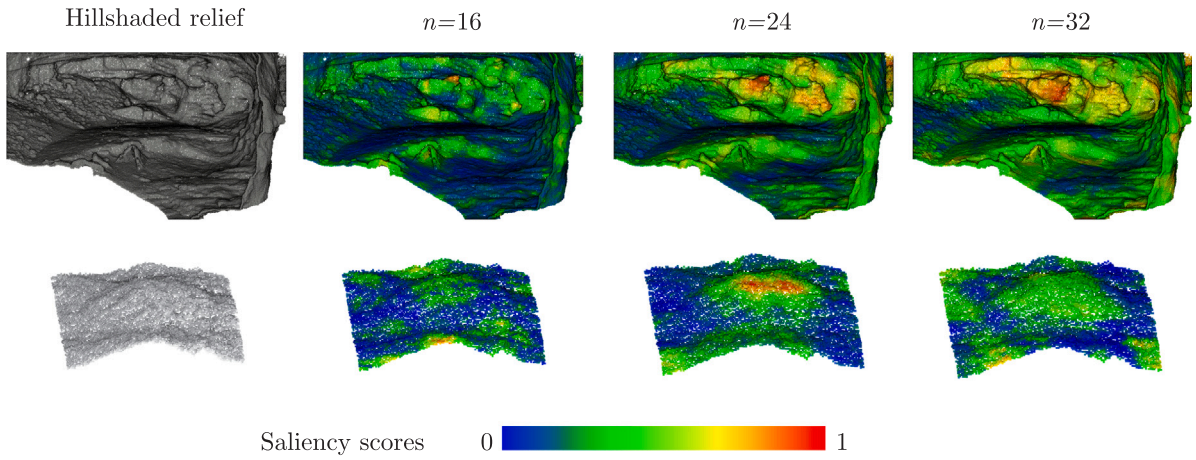

Fig. 9. Dataset #III. Saliency scores for $f = 32$ at different sizes of the voxel grid. It can be seen that as the grid size grows, more regions are marked as salient.

Table 8
Mean saliency ratio values on test subsets using the hyper-parameters that yielded the largest saliency ratio values in the tuning phase.

	f	n	Ours	Plane-based	Handcrafted
Dataset #I	8	16	2.44	1.43	2.85
Dataset #II	32	24	2.49	1.9	1.07
Dataset #III	32	16	1.23	1.06	10.9

the tuning phase (Section 5.1). This is based on the assumption that these hyper-parameters will provide the most pronounced distinction between 'salient' and 'non-salient' regions. After the inference phase, saliency ratios were evaluated for the test subsets (i.e., $\hat{r}_T$). These produced similar magnitudes as those estimated for the validation subsets (Table 8). In the following, we present the saliency map of each dataset and discuss the results separately, as we compare them to the results of the baseline methods.

5.2.1. Dataset #I

Fig. 10a shows the saliency map generated by the proposed approach for the dataset #I using $f = 8$ and $n = 16$. It can be seen that the expected gullies and sinkholes were highlighted. Higher saliency scores were given for the gullies' thalweg, the bottom of the sinkholes, and to smaller channels.

Fig. 10b and (c) show the saliency maps generated by the baseline methods. We use $n = 16$ for the plane-based method and a minimal object size of 2 m for the handcrafted one. It can be seen that the plane-based method (b) yielded poor saliency map. Though the gullies did receive higher scores, these are lower than other regions that locally deviate from planarity. Furthermore, points that belong to sinkholes were not marked relative to their surroundings. Instead, they were grouped together with other highlighted regions. The handcrafted method provided a better picture (Fig. 10c). There, most gullies and sinkholes were highlighted as well as small micro-channels. Still, the map seems noisy and regions with overlapping scanlines are marked as more salient (light green bounded by light blue). When comparing to the proposed method, the impression of the saliency map is of more consistent salient regions and less noise.

Table 8 shows the saliency ratio evaluation of the test subsets for each method. The plane-based approach shows the smallest difference, with a ratio of 1.43. The handcrafted approach yielded the highest ratio of $\hat{r}_T = 2.85$. This is in the same scale of the proposed method ($\hat{r}_T = 2.44$). It is important to mention that no statistical significance was found between H and L testing subsets using a homoscedastic t-test for all saliency methods (at 85% probability).

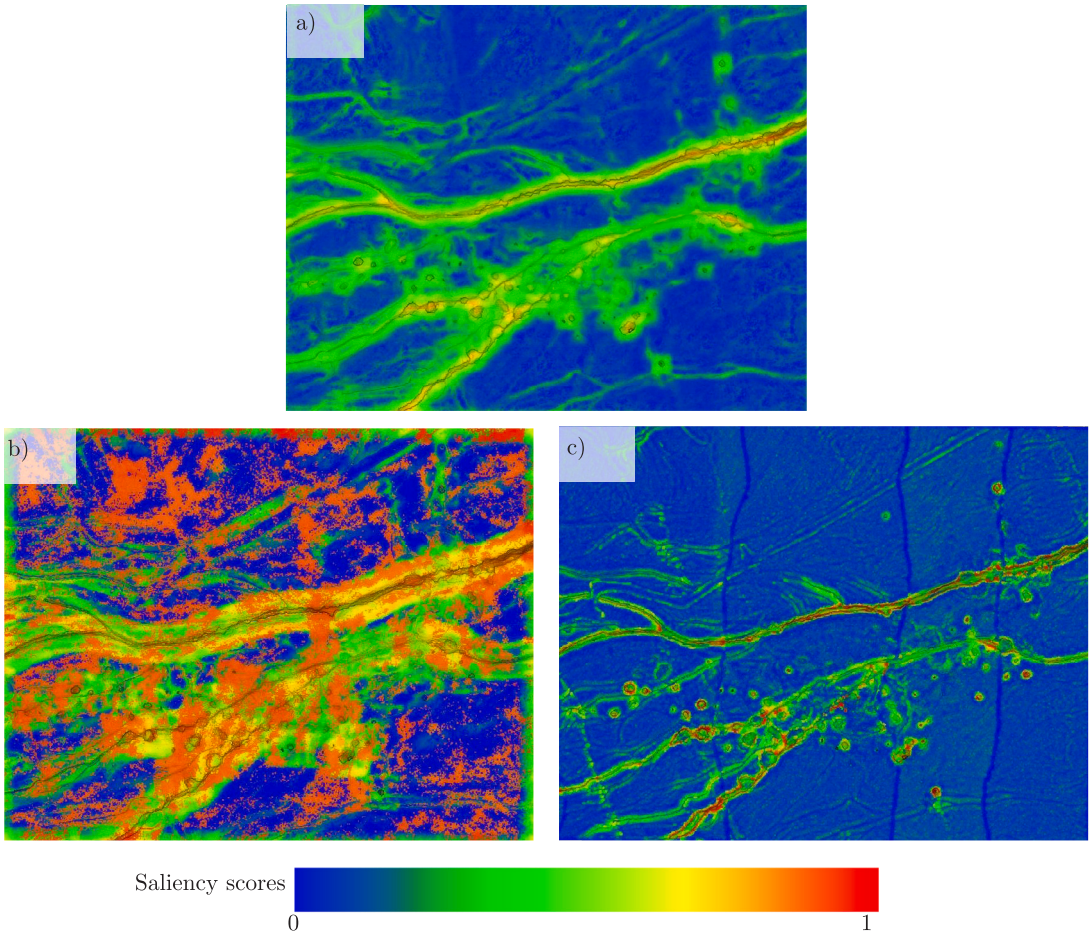

Fig. 10. Dataset #I. Saliency results using (a) proposed method with $f = 8$ and $n = 16$; (b) Plane-based highlighting with $n = 16$; (c) Handcrafted approach (Arav and Filin, 2022) using $\rho = 2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

These results are substantiated by the visual map of the detected saliency (Fig. 10b).

5.2.2. Dataset #II

Fig. 11a shows the saliency scores using $f = 32$ and $n = 24$. It can be seen that higher saliency scores refer to the river banks. However, a closer inspection discovers other expected features on the riverbed, such as boulders and hanging vegetation (Fig. 11b). Notably, other entities were found, e.g., submerged driftwood and a small incised gully (Fig. 11c–d). This result emphasizes the advantages of the proposed method: the searched features are not defined in advance, only the minimal size of interesting features needs to be specified.

We use $f = 32$ for the plane-based baseline method, and a minimal object size of 0.25 m for the handcrafted baseline method. Fig. 12 presents the saliency results of the three methods in salient and non-salient test regions. Of the three methods, the best visual results were achieved for the proposed approach (a). There, boulders are highlighted in the salient subset, whereas in the non-salient region, only the frame of the subset was marked as salient. This is an expected result, as it is more difficult to predict the surface at the edges, due to the lack of information and training data in these regions. The plane-based method highlighted most of the surface, irrespective to the data (b); the handcrafted method successfully highlighted some of the boulders (c, left), but arbitrary patterns are marked in the non-salient subset (c, right). This maybe as a result of either the scanning pattern, which yields overlapping scanlines, or due to the high surface roughness in this dataset. The visual results are generally corroborated by the saliency ratios (Table 8).

5.2.3. Dataset #III

Fig. 13a shows the saliency map using the proposed method both inside the cave (left) and on the ceiling (right). It can be seen that the points which received higher saliency scores mostly belong to niches and pockets in the walls. Additionally, points that lie on some larger rocks also have higher saliency scores, as well as a tripod that stands close to the entrance. Points belonging to blocks on the floor near the entrance were estimated with lower saliency scores. This is probably due to the fact that they cover a large part of the cave floor. Therefore, they are considered as roughness that can be predicted by the proposed model.

We used $n = 16$ for the plane-based method and a minimal object size of 0.1 m for the handcrafted approach. It can be seen that for the plane-based method, regions that deviate from planarity, which compose the majority of the dataset, were given higher scores (Fig. 13, b). The handcrafted method (Fig. 13c) provided less noisy saliency map. Most of the rocks on the ground have lower saliency scores, similar to the proposed method. However, much less points that belong to niches in the ceiling were estimated with high saliency scores compared to the proposed method. This leads to much more focused areas of interest.

Looking at the saliency ratio in the test data (Table 8), it can be seen that the handcrafted method achieved the highest values by far, whereas the plane-based method yielded the lowest. This is in accordance with the visual impression.

6. Conclusions

In this paper we proposed an unsupervised method that highlights saliency in non-urban, natural environments. Driven by the notion

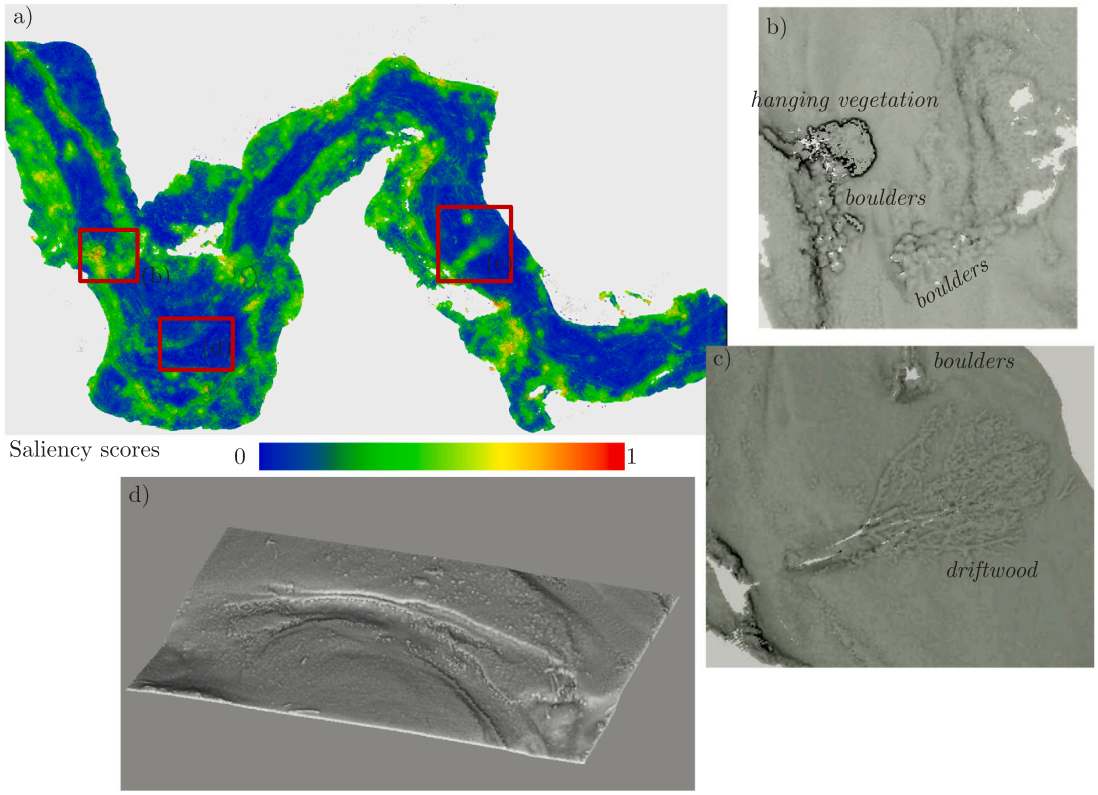

Fig. 11. Top: (a) saliency in dataset #II using 0.2 m voxel size, $n = 24$ and $f = 8$. (b–d) Hillshaded representation of regions on the riverbed that were detected as salient — (b) submerged boulders and vegetation; (c) submerged driftwood and boulders; and (c) banks of a small gully that was incised within the riverbed.

that salient regions stand out in their environment and knowing that topography is generally smooth, we search for anomalies within a scanned surface. The proposed approach is trained to reconstruct the surface based on voxel grids extracted from the data. Based on training, it reconstructs the local surface and evaluates the difference between the inferred surface and the original point cloud. Saliency scores are defined based on the difference from the expected surface. Therefore, the network should be trained for every dataset. However, the model requires some examples for salient and non-salient areas in order to tune the hyper-parameters. Nevertheless, these samples are not required for the learning process per se.

The proposed method was demonstrated on three datasets acquired by various scanning platforms in different types of scenes and presented three levels of surface complexity (from smooth, almost planar surface, to rough riverbed and to a complex 3D cave). We have shown that it was able to discern between 'salient' and 'non-salient' regions, yielding high saliency ratio.

For evaluation, we proposed a saliency ratio metric, which measures the ratio between regions previously known to have higher and lower salient scores. In addition, we visually inspected the results, while comparing them to other baseline approaches of saliency detection. We have shown that in most cases, the propose metric corresponds to the visual results.

Further examination into the more important hyper-parameters, f and n, revealed that the size of the voxel grid dictates the size of the detected region. As n increases, a larger region is reconstructed, and evidently, larger parts will deviate from the original cloud. This will result in generally higher saliency scores. Though the number of feature maps in base resolution is important to reconstruct the surface, its effect is limited. Nonetheless, a sufficient number of feature maps in base resolution is required for the reconstruction. Too many, or too less maps, will lead to higher saliency scores, also in non-salient regions. That said, we have shown that the effect of these parameters on the final results (both visually and quantitatively) is limited, especially when the surface is more complex (the cave, as an example).

When compared to baseline methods, the handcrafted approach showed some advantage over the proposed method, as it delivered more focused results. However, we have shown that its results highly depend on the scanning pattern. When point density was changing drastically (dataset #II, for example), the handcrafted method estimated high saliency scores in non-salient regions. In contrast, the proposed method was unaffected, and showed similar results independently.

CRediT authorship contribution statement

Reuma Arav: Writing – review & editing, Writing – original draft, Visualization, Project administration, Methodology, Funding acquisition. **Dennis Wittich:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Franz Rottensteiner:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant 896409. The authors would also like to thank Prof. Gottfried Mandlburger for sharing dataset #II.

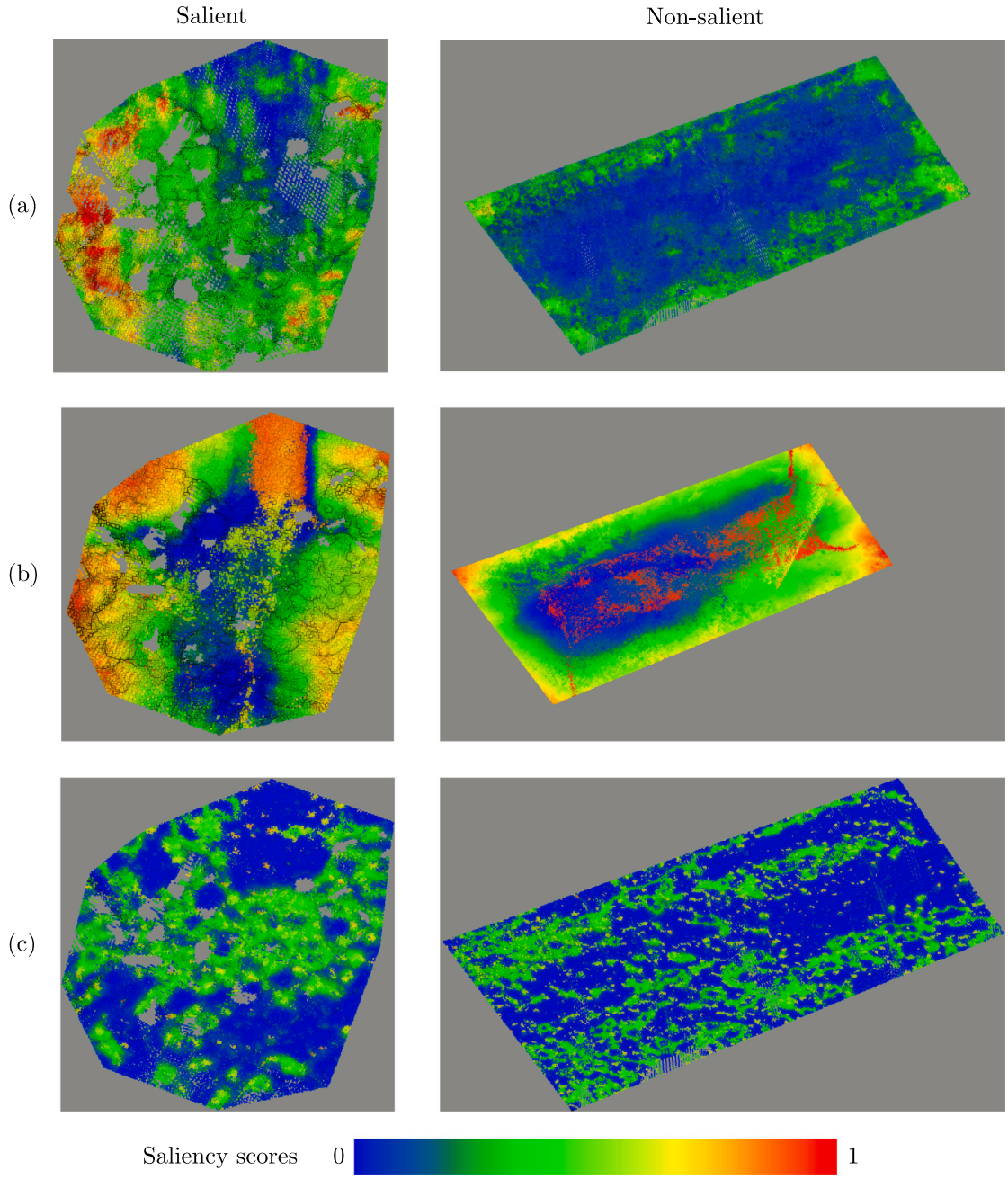

Fig. 12. Dataset #II. Saliency scores using the three methods on test regions: (a) proposed method using $n = 24$ and $f = 8$; (b) plane-based reconstruction using $f = 32$; (c) handcrafted method using $\rho = 0.25$ m.

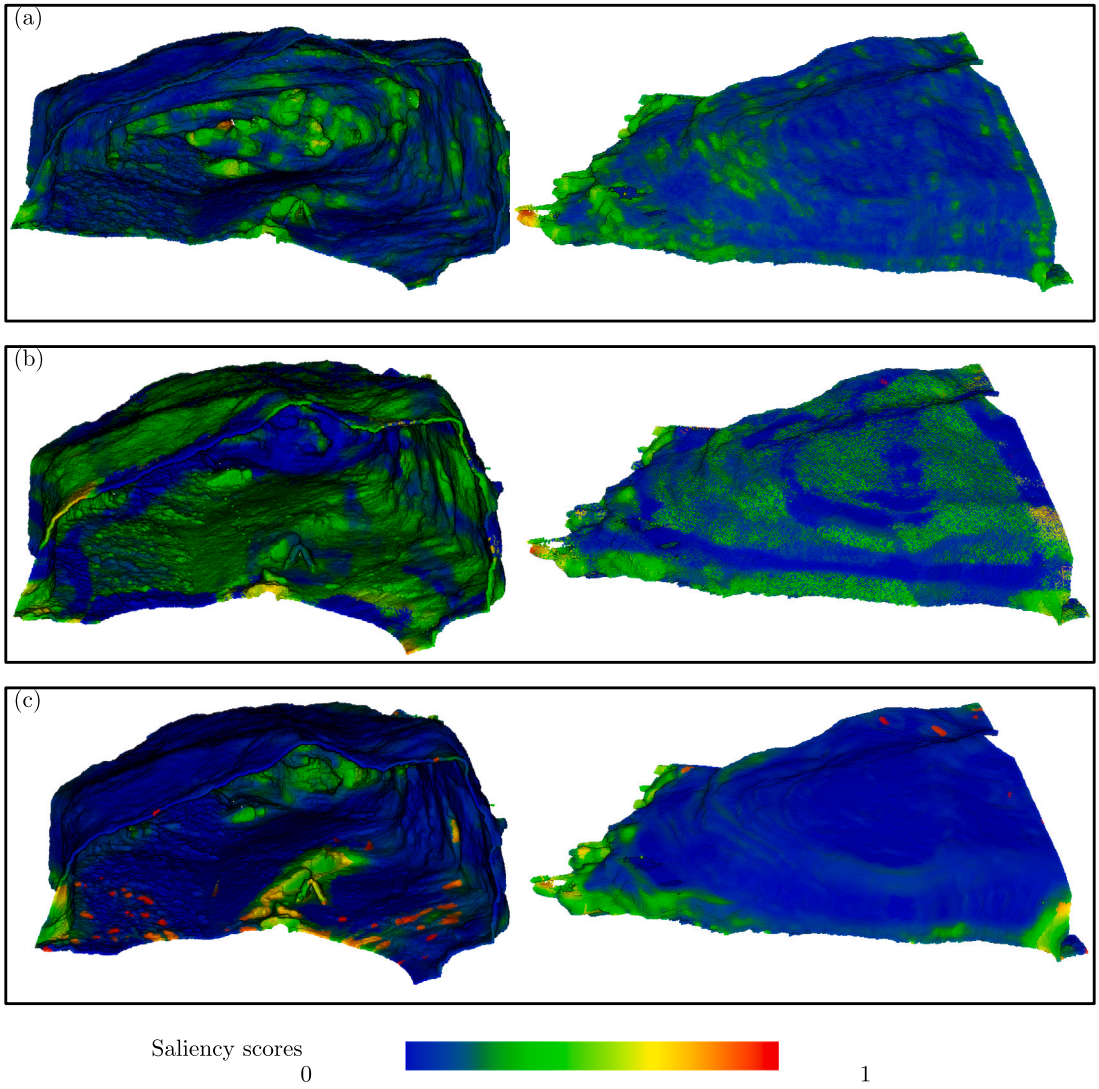

Fig. 13. Saliency scores using the three methods on the terrestrial dataset. Horizontal look into the cave (left) and at the walls and ceiling (right). Note that the ceiling point cloud was acquired from within the cave. (a) proposed method using $n = 16$ and $f = 64$; (b) plane-based reconstruction using $f = 24$ (c) handcrafted method using $\rho = 0.3$ m.

References

Akman, O., Jonker, P., 2010. Computing saliency map from spatial information in point cloud data. In: Advanced Concepts for Intelligent Vision Systems. Springer Berlin Heidelberg, pp. 290–299. http://dx.doi.org/10.1007/978-3-642-17688-3_28.

Alexiou, E., Xu, P., Ebrahimi, T., 2019. Towards modelling of visual saliency in point clouds for immersive applications. In: 2019 IEEE International Conference on Image Processing. ICIP, pp. 4325–4329. http://dx.doi.org/10.1109/icip.2019.8803479.

Arav, R., Filin, S., 2020. Saliency of Subtle Entities within 3-D Point Clouds. Copernicus GmbH, pp. 179–186. http://dx.doi.org/10.5194/isprs-annals-v-2-2020-179-2020.

Arav, R., Filin, S., 2022. A visual saliency-driven extraction framework of smoothly embedded entities in 3D point clouds of open terrain. ISPRS J. Photogramm. Remote. Sens. 188, 125–140. http://dx.doi.org/10.1016/j.isprsjprs.2022.04.003.

Arav, R., Filin, S., Pfeifer, N., 2022a. Content-aware point cloud simplification of open scenes. IEEE Trans. Geosci. Remote Sens. 60, 1–12. http://dx.doi.org/10.1109/TGRS.2022.3208348.

Arav, R., Pöppl, F., Pfeifer, N., 2022b. A Point-Based Level-Set Approach for the Extraction of 3D Entities from Point Clouds – Application in Geomorphological Context. Copernicus GmbH, pp. 95–102. http://dx.doi.org/10.5194/isprs-annals-v-2-2022-95-2022.

Arav, R., Ressl, C., Weiss, R., Artz, T., Mandlburger, G., 2024. Evaluation of active and passive UAV-based surveying systems for eulittoral zone mapping. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. XLVIII-2–2024, 9–16. http://dx.doi.org/10.5194/isprs-archives-xlviii-2-2024-9-2024.

Arav, R., Wittich, D., 2023. Salient_anomaly. URL: https://github.com/rarav/salient_anomaly/releases/tag/v1.0.0.

Arvanitis, G., Zacharaki, E.I., Vasa, L., Moustakas, K., 2022. Broad-to-narrow registration and identification of 3D objects in partially scanned and cluttered point clouds. IEEE Trans. Multimed. 24, 2230–2245. http://dx.doi.org/10.1109/tmm.2021.3089838.

Chen, Q., Fu, K., Liu, Z., Chen, G., Du, H., Qiu, B., Shao, L., 2021b. EF-net: A novel enhancement and fusion network for RGB-D saliency detection. Pattern Recognit. 112, 107740. http://dx.doi.org/10.1016/j.patcog.2020.107740.

Chen, H., Li, Y., Deng, Y., Lin, G., 2021a. CNN-based RGB-D salient object detection: Learn, select, and fuse. Int. J. Comput. Vis. 129 (7), 2076–2096. http://dx.doi.org/10.1007/s11263-021-01452-0.

Ding, X., Lin, W., Chen, Z., Zhang, X., 2019. Point cloud saliency detection by local and global feature fusion. IEEE Trans. Image Process. http://dx.doi.org/10.1109/tip.2019.2918735, 1–1.

Doraiswamy, H., Shivashankar, N., Natarajan, V., Wang, Y., 2013. Topological saliency. Comput. Graph. 37 (7), 787–799. http://dx.doi.org/10.1016/j.cag.2013.04.009.

Fan, S., Gao, W., Li, G., 2022. Salient object detection for point clouds. In: Lecture Notes in Computer Science. Springer Nature Switzerland, pp. 1–19. http://dx.doi.org/10.1007/978-3-031-19815-1_1.

Geological Survey of Israel, Arav, R., 2013. Zeelim fan, Israel (part). http://dx.doi.org/10.48436/mps0m-c9n43, [online], TU Data Repository.

Guo, Y., Wang, F., Xin, J., 2018. Point-wise saliency detection on 3D point clouds via covariance descriptors. Vis. Comput. 34 (10), 1325–1338. http://dx.doi.org/10.1007/s00371-017-1416-3.

Han, J., Chen, H., Liu, N., Yan, C., Li, X., 2018. CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion. IEEE Trans. Cybern. 48 (11), 3171–3183. http://dx.doi.org/10.1109/tcyb.2017.2761775.

Hao, W., Liang, W., Wang, Y., Zhao, M., Li, Y., 2019. Saliency-guided repetition detection from facade point clouds. IEEE Access 7, 150072–150081. http://dx.doi.org/10.1109/access.2019.2947537.

Hillier, J.K., Smith, M.J., Armugam, R., Barr, I., Boston, C.M., Clark, C.D., Ely, J., Frankl, A., Greenwood, S.L., Gosselin, L., Hättestrand, C., Hogan, K., Hughes, A.L.C., Livingstone, S.J., Lovell, H., McHenry, M., Munoz, Y., Pellicer, X.M., Pellitero, R., Robb, C., Roberson, S., Ruther, D., Spagnolo, M., Standell, M., Stokes, C.R., Storrar, R., Tate, N.J., Wooldridge, K., 2014. Manual mapping of drumlins in synthetic landscapes to assess operator effectiveness. J. Maps 11 (5), 719–729. http://dx.doi.org/10.1080/17445647.2014.957251.

Hong, T., Zhang, Z., Ma, J., 2023. PCSalmix: Gradient saliency-based mix augmentation for point cloud classification. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, http://dx.doi.org/10.1109/icassp49357.2023.10095576.

Itti, L., Koch, C., Niebur, E., 1998. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. 20 (11), 1254–1259. http://dx.doi.org/10.1109/34.730558.

Jiang, Z., Ding, L., Tam, G.K., Song, C., Li, F.W., Yang, B., 2023. C2SPoint: A classification-to-saliency network for point cloud saliency detection. Comput. Graph. 115, 274–284. http://dx.doi.org/10.1016/j.cag.2023.07.003.

Kobyshev, N., Riemenschneider, H., Bódis-Szomorú, A., Gool, L.V., 2016. 3D saliency for finding landmark buildings. In: 2016 Fourth International Conference on 3D Vision. 3DV, pp. 267–275. http://dx.doi.org/10.1109/3DV.2016.35.

Kyriou, A., Nikolakopoulos, K., Koukouvelas, I., 2021. How image acquisition geometry of UAV campaigns affects the derived products and their accuracy in areas with complex geomorphology. ISPRS Int. J. Geo- Inf. 10 (6), 408. http://dx.doi.org/10.3390/ijgi10060408.

Laazoufi, A., Hassouni, M.E., 2022. Saliency-based point cloud quality assessment method using aware features learning. In: 2022 9th International Conference on Wireless Networks and Mobile Communications. WINCOM, IEEE, http://dx.doi.org/10.1109/wincom55661.2022.9966464.

Leal, E.A., Sanchez-Torres, G., Branch-Bedoya, J.W., 2019. Point cloud saliency detection via local sparse coding. DYNA 86 (209), 238–247. http://dx.doi.org/10.15446/dyna.v86n209.75958.

Li, J., Ji, W., Zhang, M., Piao, Y., Lu, H., Cheng, L., 2022. Delving into calibrated depth for accurate RGB-D salient object detection. Int. J. Comput. Vis. 131 (4), 855–876. http://dx.doi.org/10.1007/s11263-022-01734-1.

Liang, A., Zhang, H., Hua, H., Chen, W., 2023. To drop or to select: Reduce the negative effects of disturbance features for point cloud classification from an interpretable perspective. IEEE Access 11, 36184–36202. http://dx.doi.org/10.1109/access.2023.3266340.

Liu, Y., Yuan, Y., Gao, S., 2019. Modeling the vagueness of areal geographic objects: A categorization system. ISPRS Int. J. Geo- Inf. 8 (7), 306. http://dx.doi.org/10.3390/ijgi8070306.

Mandlburger, G., Hauer, C., Wieser, M., Pfeifer, N., 2015. Topo-bathymetric LiDAR for monitoring river morphodynamics and instream habitats—a case study at the Pielach River. Remote. Sens. 7 (5), 6160–6195. http://dx.doi.org/10.3390/rs70506160.

Molenaar, M., Cheng, T., 2000. Fuzzy spatial objects and their dynamics. ISPRS J. Photogramm. Remote. Sens. 55 (3), 164–175. http://dx.doi.org/10.1016/s0924-2716(00)00017-4.

Peng, P., Yang, K.-F., Luo, F.-Y., Li, Y.-J., 2021. Saliency detection inspired by topological perception theory. Int. J. Comput. Vis. 129 (8), 2352–2374. http://dx.doi.org/10.1007/s11263-021-01478-4.

Pfeifer, N., Mandlburger, G., 2018. Topographic Laser Ranging and Scanning. CRC Press, p. 30. http://dx.doi.org/10.1201/9781315154381, chapter LiDAR Data Filtering and Digital Terrain Model Generation.

Pfeifer, N., Mandlburger, G., Otepka, J., Karel, W., 2014. OPALS – a framework for airborne laser scanning data analysis. Comput. Environ. Urban Syst. 45, 125–136. http://dx.doi.org/10.1016/j.compenvurbsys.2013.11.002.

Qin, Z., Wang, C., Peng, Y., Xu, K., 2023. CasViGE: Learning robust point cloud registration with cascaded visual-geometric encoding. Comput. Aided Geom. Design 104, 102217. http://dx.doi.org/10.1016/j.cagd.2023.102217.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp. 234–241.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE, pp. 3212–3217. http://dx.doi.org/10.1109/robot.2009.5152473.

Scheiber, T., Fredin, O., Viola, G., Jarna, A., Gasser, D., Łapińska-Viola, R., 2015. Manual extraction of bedrock lineaments from high-resolution LiDAR data: methodological bias and human perception. GFF 137 (4), 362–372. http://dx.doi.org/10.1080/11035897.2015.1085434.

Shtrom, E., Leifman, G., Tal, A., 2013. Saliency Detection in Large Point Sets. In: Computer Vision (ICCV), 2013 IEEE International Conference on. pp. 3591–3598. http://dx.doi.org/10.1109/ICCV.2013.446.

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M., 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Lecture Notes in Computer Science. Springer International Publishing, pp. 240–248. http://dx.doi.org/10.1007/978-3-319-67558-9_28.

Sun, P., Zhang, W., Li, S., Guo, Y., Song, C., Li, X., 2022. Learnable depth-sensitive attention for deep RGB-D saliency detection with multi-modal fusion architecture search. Int. J. Comput. Vis. 130 (11), 2822–2841. http://dx.doi.org/10.1007/s11263-022-01646-0.

Tarolli, P., Mudd, S.M., 2020. Introduction to remote sensing of geomorphology. In: Developments in Earth Surface Processes. Elsevier, pp. xiii–xv. http://dx.doi.org/10.1016/b978-0-444-64177-9.09992-6.

Tasse, P.F., Kosinka, J., Dodgson, N., 2015. Cluster-based point set saliency. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 163–171.

Telling, J., Lyda, A., Hartzell, P., Glennie, C., 2017. Review of earth science research using terrestrial laser scanning. Earth Sci. Rev. 169, 35–68. http://dx.doi.org/10.1016/j.earscirev.2017.04.007.

Tinchev, G., Penate-Sanchez, A., Fallon, M., 2021. SKD: Keypoint detection for point clouds using saliency estimation. IEEE Robot. Autom. Lett. 6 (2), 3785–3792. http://dx.doi.org/10.1109/lra.2021.3065224.

Vinci, A., Todisco, F., Mannocchi, F., 2016. Calibration of manual measurements of rills using Terrestrial Laser Scanning. CATENA 140, 164–168. http://dx.doi.org/10.1016/J.CATENA.2016.01.026.

Wang, H., Luo, H., Wen, C., Cheng, J., Li, P., Chen, Y., Wang, C., Li, J., 2015. Road boundaries detection based on local normal saliency from mobile laser scanning data. IEEE Geosci. Remote. Sens. Lett. 12 (10), 2085–2089. http://dx.doi.org/10.1109/LGRS.2015.2449074.

Wimmer, M., Oberender, P., 2022. Untere traisenbacher Höhle - UTB_104423. http://dx.doi.org/10.48436/fh0am-at738, [online], TU Data Repository.

Xu, J., Li, Z., Du, B., Zhang, M., Liu, J., 2020. Reluplex made more practical: Leaky ReLU. In: 2020 IEEE Symposium on Computers and Communications. ISCC, IEEE, pp. 1–7.

Yun, J.-S., Sim, J.-Y., 2016. Supervoxel-based saliency detection for large-scale colored 3D point cloud. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. APSIPA ASC, IEEE, pp. 193–1200.

Zhang, J., Fan, D.-P., Dai, Y., Anwar, S., Saleh, F., Aliakbarian, S., Barnes, N., 2021a. Uncertainty inspired RGB-D saliency detection. IEEE Trans. Pattern Anal. Mach. Intell. http://dx.doi.org/10.1109/tpami.2021.3073564, 1–1.

Zhang, J., Fan, D.-P., Dai, Y., Anwar, S., Saleh, F.S., Zhang, T., Barnes, N., 2020. UC-net: Uncertainty inspired RGB-D saliency detection via conditional variational autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Zhang, J., Fan, D.-P., Dai, Y., Yu, X., Zhong, Y., Barnes, N., Shao, L., 2021b. RGB-D saliency detection via cascaded mutual information minimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 4338–4347.

Zheng, T., Chen, C., Yuan, J., Li, B., Ren, K., 2019. PointCloud saliency maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 1598–1606.

Zhou, W., Lv, Y., Lei, J., Yu, L., 2021. Global and local-contrast guides content-aware fusion for RGB-D saliency prediction. IEEE Trans. Syst. Man Cybern.: Syst. 51 (6), 3641–3649. http://dx.doi.org/10.1109/tsmc.2019.2957386.