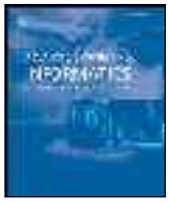
Advanced Engineering Informatics 57 (2023) 102076

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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



Full length article

Cross domain matching for semantic point cloud segmentation based on image segmentation and geometric reasoning

Jan Martens *, Timothy Blut, Jörg Blankenbach

Geodetic Institute and Chair for Computing in Civil Engineering & Geo Information Systems, RWTH Aachen University, Mies-van-der-Rohe Straße 1, Aachen, 52074, North Rhine-Westphalia, Germany

ARTICLE INFO

Keywords:
Infrastructure
Machine learning
Cross domain matching
Point clouds
Semantic segmentation
BIM

ABSTRACT

Many infrastructure assets in transportation such as roads and bridges represent challenges for inspection and maintenance due to advanced age, structural deficiencies and modifications. Concepts such as Building Information Modelling (BIM) aim to alleviate the problem of health monitoring and asset management by providing digital building models constructed from survey data to all stakeholders. Ageing and oftentimes poorly-documented infrastructure objects such as bridges in particular benefit from a continuous integration of changes to form a digital twin which reflects the asset's as-is state. However, the process of reconstructing geometric–semantic models from survey data is a manual and labour-intensive process and makes continuously updating the models a difficult task. To automate this process, a cross-domain approach using an artificial neural network is presented which performs semantic segmentation in the image domain and transfers the results over to the point cloud. For the following fine segmentation, geometric knowledge in the 3D domain is used for post-processing and filtering via geometric reasoning. Using this method, a 3D semantic segmentation is achieved which does not require any 3D point cloud training data and only a low amount of image training data.

1. Introduction

Digitalization has become a driving factor behind the shift towards Industry 4.0 and the adoption of Building Information Modelling (BIM) in the architecture, engineering and construction (AECO) industry. Benefits of BIM for digital design, construction and operation are apparent, as digital models including semantics and geometry allow for transparent data exchange and inspection [1,2]. The introduction of standardized BIM data models such as Industry Foundation Classes (IFC) further reinforce this point. In theory, this makes BIM an ideal fit for infrastructure assets, as a large number of ageing objects such as bridges exist, where continuous documentation of damages, repairs and modifications is of critical interest, as it greatly simplifies maintenance [3,4]. However, many of them have been built before the establishment of the BIM paradigm and consequently lack digital data and the adoption of BIM in this field has been a sluggish process, with appropriate IFC extensions like *IFC Bridge* only recently being introduced [5,6].

Ideally, a BIM model accurately reflects the current state of an asset at any given time akin to the digital twin concept to enable life cycle management. This motivates the need for 3D modelling in brownfield, i.e. the geometric–semantic as-is modelling of existing infrastructure assets. Continuously updating and enriching these models with further information such as data from inspection, maintenance or monitoring (including sensors) leads to the concept of the Digital Twin. However as a basis for Digital Twins, automated workflows for the modelling process of infrastructure objects in the brownfield and their components are highly-desirable as a means of saving time and the resources related to labour-intensive, manual modelling. The process of generating semantically rich digital as-is models of an asset is commonly referred to as Scan-to-BIM and can be subdivided into four distinct stages: data capturing, semantic segmentation, 3D modelling and BIM model generation (see Fig. 1).

Within Scan-to-BIM workflows, semantic segmentation precedes automated modelling and consequently helps inform other algorithms about relevant regions and the object class they belong to within point clouds. The process of modelling 3D asset components and assigning relevant semantic attributes is thus simplified as classification knowledge can be used to choose appropriate modelling workflows [7]. As seen in recent years, machine learning-based segmentation methods have matured notably and are an ideal fit for this problem [8,9]. Even

* Corresponding author.
E-mail addresses: jan.martens@gia.rwth-aachen.de (J. Martens), timothy.blut@gia.rwth-aachen.de (T. Blut), blankenbach@gia.rwth-aachen.de (J. Blankenbach).

https://doi.org/10.1016/j.aei.2023.102076
Received 16 December 2022; Received in revised form 11 April 2023; Accepted 23 June 2023
Available online 5 July 2023

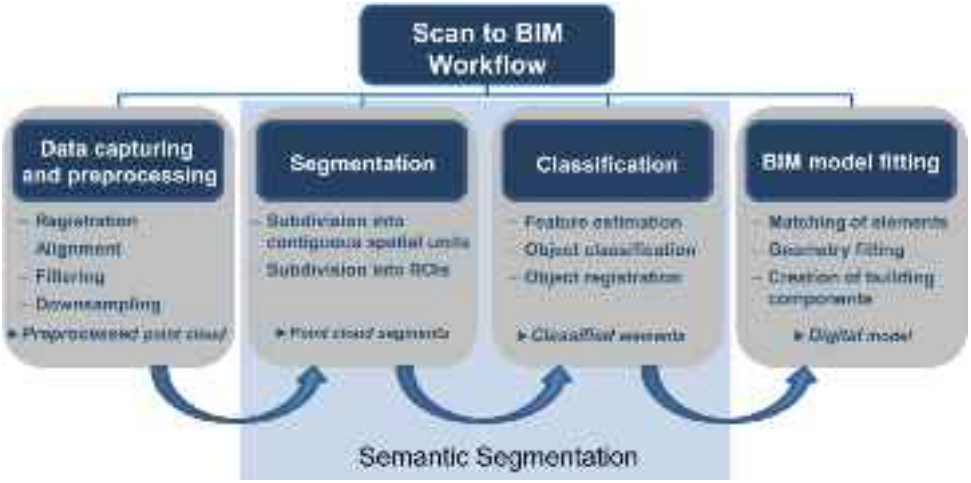


Fig. 1. Stages of the Scan to BIM process. The presented method falls into the categories of segmentation and classification (semantic segmentation). Consequently, it lays the foundation for the subsequent BIM model fitting.

though methods like artificial neural networks (ANN) for the direct segmentation of point clouds already exist, they require sufficient amounts of annotated training data. Training data requirements for complex ANN architectures which fall under the Deep Learning (DL) category are particularly high [10] and represent a limiting factor due to low availability of annotated 3D training data for specific domains [11].

Fortunately, 3D reality capturing technologies have made large strides in recent years, with high resolution terrestrial laser scanning (TLS), mobile laser scanning (MLS) and image-based data capturing by drones becoming more prevalent, as they allow for fast capturing of large objects and are therefore an ideal fit for the overall Scan-to-BIM workflow [12,13]. Nevertheless, annotated training data for point clouds is relatively scarce, with most existing datasets of outdoor data being focused on automated driving in urban settings [14–16]. 3D data with labels for infrastructure assets and bridge components in particular is thus not publicly available and in the light of costs of high-quality 3D capturing equipment, creating labelled custom datasets is extremely cumbersome. This issue affects image data to a lesser degree, where capturing and labelling images is an established process [17]. The ability of machine learning algorithms for image processing (convolutional neural network architectures in particular) to produce annotated segmentation masks means that tackling the segmentation problem from a 2D perspective represents a more practical solution.

The focus of this paper lies on a method for machine learning-based semantic point cloud segmentation for bridge assets which does not require any 3D training data. Given that image data is usually captured during surveys (either to colourize TLS/MLS point clouds or as drone footage), a convolutional neural network (CNN) for image segmentation is used, which has been trained using transfer learning to circumvent a limited pool of training images [18]. Using the combined information of images, point clouds and camera parameters typically captured during surveys, this network segments the images, with the resulting segmentation masks afterwards being projected into the point cloud. The pre-segmentation result is subsequently post-processed for fine segmentation using geometrical knowledge to propagate labels to regions occluded from camera view and to remove segmentation artefacts.

2. Related works

Point cloud segmentation has been a long-standing problem in geodesy, civil engineering and architecture as well as computer vision, with many early works focussing on highly-specialized algorithms for the extraction of geometric primitives such as planes and cylinders for computer-aided design (CAD) [19,20]. As new, affordable and user-friendly 3D capturing techniques became available in the following years, the field expanded gradually and gave rise to various segmentation methods with applications ranging from robotics, reverse engineering and 3D modelling coming into the fold [21]. Earlier works suggested combining techniques related to 3D laser scanning and 2D imaging with one another to overcome their individual shortcomings and create novel Scan-to-BIM workflows which could play a critical role in the automated as-is modelling of semantically-rich models [22]. These ideas even date back to the time before the rise of deep learning, with one workflow suggesting classical machine learning methods and computer vision for object detection in 2D images to provide semantic information for the 3D modelling process [23]. Many influences in segmentation can be traced back to classical and machine-learning-based image processing and are therefore worth discussing in detail.

2.1. ANN-based semantic segmentation

With segmentation being a crucial step in the creation of not only digital models but also fields such as robotics, many works in the past decade have moved towards machine learning to solve semantic segmentation problems in a flexible manner. Early approaches oftentimes relied on using the then-popular Support Vector Machines (SVM) alongside hand-crafted features to analyse 3D scenes on different scales for segmentation [24].

At around the same time, image recognition advanced significantly due to the rising popularity of ANNs and deep learning. Driven by the availability of large training data sets and the required computational power, a variety of architectures quickly emerged which outperformed traditional object detection techniques, with AlexNet [8] and YOLO [25] being notable examples. Models for semantic segmentation have experienced equally fast developments, with the R-CNN and its successor Fast R-CNN classifying and refining pre-segmented regions [26,27]. The noteworthy derivative Mask R-CNN performs instance segmentation directly on the input images, making it an optimal match for the given task [28].

For 3D point cloud segmentation, early neural network architectures project point clouds down to a 2.5D plane from a birds-eye view perspective to use previously established 2D object detection networks [29–32]. Voxel-based classifiers on the other hand use concepts from 2D CNNs and extend them to 3D voxels by aggregating points into discrete spatial units [33,34]. Methods such as PointNet and PointNet++ [9,35] on the other hand can be seen as of the most prominent examples which work directly on point clouds. Later methods combine information from the image and point cloud domains [36,37], adapt

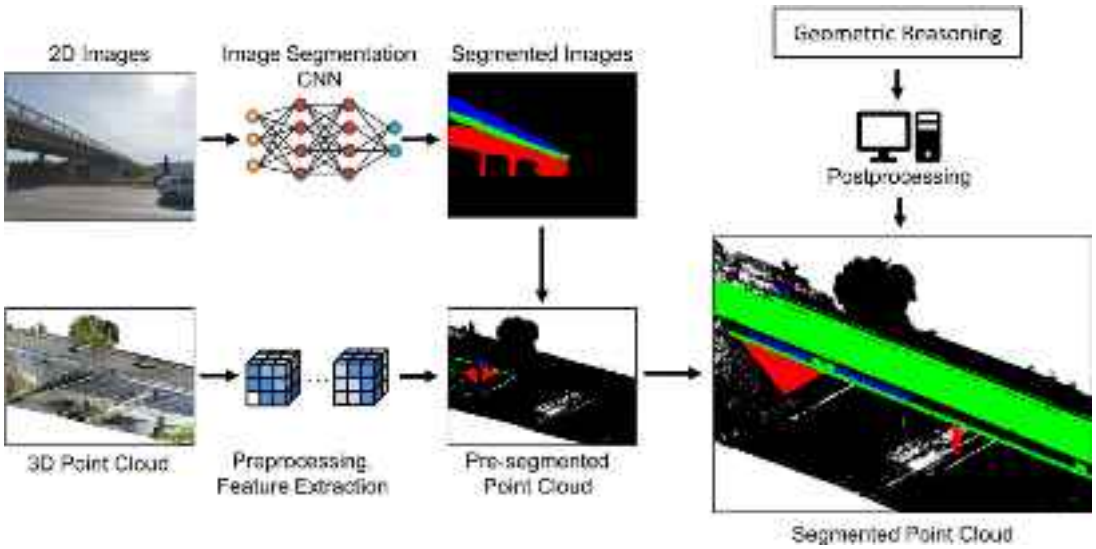


Fig. 2. Presented workflow for bridge point cloud semantic segmentation. Points clouds are preprocessed using common strategies. Images are classified and their labels projected into the point cloud to obtain a pre-segmented point cloud. For fine segmentation in the post-processing step, prior geometric knowledge is employed to refine the detected regions.

Fig. 3. Result of naive image label projection into the point cloud from one fixed camera perspective. Points occluded from view are incorrectly labelled, leading to many False-Positive detections.

operators seen in other domains for use with point clouds [38,39] or eliminate typical performance shortcomings of 3D neural networks with efficient sampling strategies [40].

As a related research branch, image-based techniques for 3D point clouds have been developed as well, where pre-segmented point clouds would be rendered from different viewports to enable classification [41, 42]. With this approach being successful in indoor scenarios, later works have proven that this strategy works equally well in outdoor environments in the context of automated driving with performance being in fact superior to most native 3D-based approaches [43]. One work of particular interest demonstrates how transfer learning can be used to adapt a pre-trained RGB-D image segmentation CNN on 3D indoor environments [44]. While similar in concept to the presented work, RGB-D images are much easier to segment due to the presence of geometrical information and existence of annotated data sets. Furthermore, issues discussed in the following sections like image-to-point cloud registration and labelling of occluded elements are not present in such scenarios.

2.2. Region growing segmentation

Generalized segmentation algorithms have been subject to research for years, with many of them drawing inspiration from classical image segmentation. Region growing in particular has been a well-established, fast and very flexible method for image segmentation which has found use in various fields, ranging from medical image analysis [45] to the segmentation of panorama images captured with laser scanners [46]. Classical region growing on images usually relies on one or more seed regions, which are expanded based on user-defined criteria [47]. While the growth criteria can be seen as an incorporation of domain knowledge into the system, seed selection plays an equally integral role. In image analysis, one or more seed regions are defined automatically, in interactive environments they are typically chosen by experts such that their knowledge contributes to the result [45,48]. Given the ill-defined neighbourhoods in the 3D domain, many 3D adaptations of this method sidestep this problem by relying on 2D neighbourhoods. This means that depth images or cases where a transformation from 2D to

Fig. 4. Depth map created by projecting the captured point cloud to a virtual camera plane. Note that for each pixel only points with the lowest distance to the camera are kept to solve the visibility problem. Extrinsic parameters describing the camera pose and camera intrinsics are required for projection.

3D space is well-defined, benefit from these techniques [46,49]. Direct conversions of the algorithm to 3D point clouds usually use neighbourhoods based on voxels [50] or graph-like neighbourhoods based on either k-nearest neighbours or fixed-radius distances [51]. In non-interactive 3D scenarios, seed points are usually chosen automatically based on the desired features [13,52]. With the segmentation of planar, man-made structures being a common goal, smoothness constraints are oftentimes used to guide the growth process, resulting in segments that resemble planar surfaces [13,53]. If required, the resulting surface patches might be filtered and merged with one another based on geometrical constraints or semantic properties [54].

3. Methods

The presented approach requires both, image and point cloud survey data to be available alongside the intrinsic and extrinsic parameters of the camera used for capturing. Using this data, the workflow illustrated in Fig. 2 is performed which can be broken down into the three following steps:

(1) At the initial stage, a CNN is trained and used for image-based semantic segmentation in the 2D domain. **(2)** The results of this stage are projected into the point cloud. **(3)** In a step independent from the label projection, the point cloud data in the 3D domain is further processed to refine the semantic segmentation, remove outliers and pre-calculate geometric features relevant for later steps. Using these geometric cues, the point cloud is then post-processed to assign labels to unlabelled regions. This way, region labels are added to building components which were either occluded from view or suffer from poor visibility. A final filter uses geometric reasoning to check constraints specific to each object type and removes points displaying uncharacteristic features from their regions.

3.1. Image-based semantic segmentation

Semantic segmentation of the captured images is done using an ANN. For this specific work Mask R-CNN [28] was chosen as it allows for instance segmentation, thus outputting a separate segmentation mask for each object of interest, although other networks capable of semantic segmentation should be suitable as well. Furthermore, it has proven to be quite robust and benefits from a low number of false positive detections. Mask R-CNN relies on a feature detection backbone such as ResNet [55] or ResNeXt [56] for feature extraction at different scales which it combines with RoIAlign for region proposal. It generates pixel-accurate segmentation masks, bounding boxes and labels making it a fitting candidate for projection of 2D labels into 3D point clouds.

Due to only moderate amounts of training data being available for bridges, a Mask R-CNN network pre-trained on the COCO dataset [57] was used as a foundation. Following the transfer learning method [18], the pre-trained network was modified by removing the classification head and replacing it with one specific to the bridge components *deck*, *abutment* and *railing*. Despite this process preserving the ANN's weights related to feature extraction of generic objects, a re-training is still required to adapt these layers and to make sure that the new classification head learns how features relate to the new object classes. For retraining, a training dataset was created from image data captured during surveys, with a total number of around 600 images being labelled appropriately. To further extend the training set size and make the final model more robust, typical data augmentation operations such as random cropping and mirroring were applied.

3.2. Image-to-point cloud label projection

With classification masks for each image being available, these masks still need to be projected into the point cloud. Naive projection however poses two issues: the first problem are conflicts resulting from masks of different perspectives overlapping with one another. This issue can be circumvented by using majority voting for points with conflicting labels. The bigger issue stems from the lack of 3D information in the images which means that points occluded from each camera perspective are erroneously labelled (see Fig. 3). Dealing with this problem requires the intrinsic and extrinsic parameters and the captured point cloud. This means that a camera calibration is required to acquire the camera's intrinsic parameters, which define how points are projected to the camera image plane. Extrinsic parameters representing the camera pose must be captured during the survey and are reconstructed with relation to one another during registration of TLS or MLS scans. Together, intrinsic and extrinsic parameters allow for a projection of the point cloud onto a virtual image plane, where each pixel represents each point's distance to the camera. Fig. 4 illustrates the idea behind this method which has been borrowed from depth maps used in computer graphics and enables per-pixel occlusion testing by keeping only the points with minimal distance to the camera for each pixel [58]. Sparsely-sampled regions close to the camera may still display "holes" where some background points are visible. Increasing the radius of the rendered points deals with this problem though and helps simulate a continuous, closed surface.

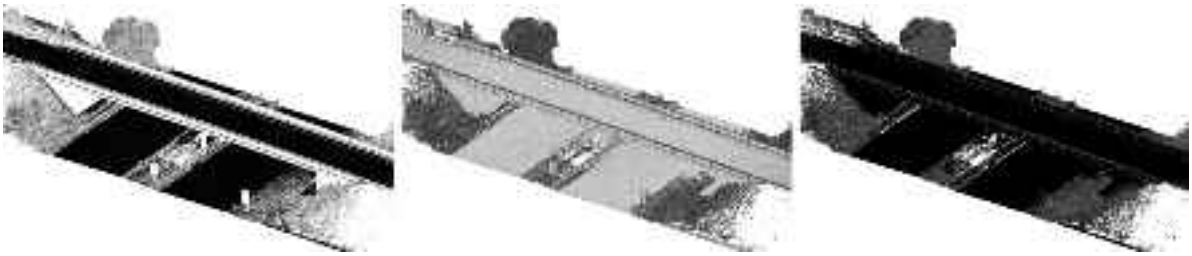


Fig. 5. Point cloud features used for geometric–semantic post-processing in the fine segmentation step. Verticality (left) describes the surface orientation, planarity (centre) and curvature (right) are indicators for the flatness of an area. Curvatures are generally low in flat regions and help identify edges where different surfaces meet in a robust way.

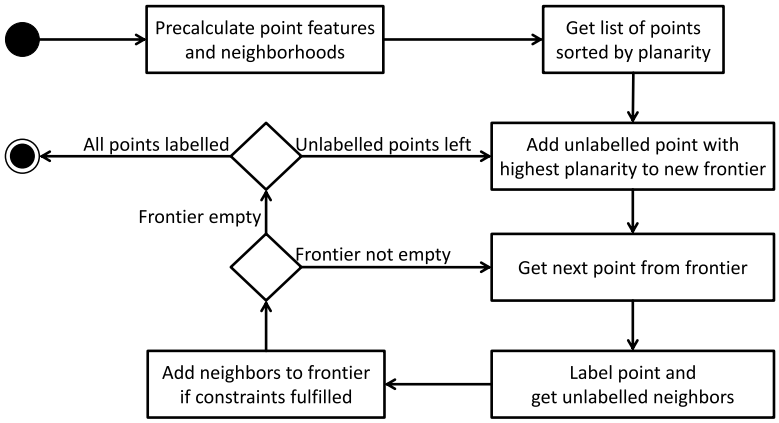


Fig. 6. Workflow of the region growing algorithm.

3.3. Fine-segmentation through geometric–semantic point cloud post-processing

Projection of the image masks into the point cloud results in a pre-segmented point cloud. The quality of this pre-segmentation is then improved by two post-processing steps for the fine segmentation.

At first, a region-growing algorithm is used to assign labels alongside the 3D geometry to label previously partially occluded or otherwise unlabelled points. Afterwards, a filtering process cleans up the resulting segmentation and removes any outliers and false positives.

As a prerequisite for post-processing, local features of each point are estimated. Based on a local neighbourhood, the feature estimation for each point p_i is performed using principal component analysis (PCA) to extract the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$. While the normal n_i of each point is given as the vector related to the smallest eigenvalue λ_3 alongside a global vertical axis $\vec{up}$ (here, $\vec{up}$ is defined as $(0, 0, 1)$), the features *verticality* V, *planarity* P and *curvature* C are estimated as follows:

$$V(n_i) = arccos(n_i \cdot \vec{up}) \tag{1}$$

$$P(p_i) = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{2}$$

$$C(p_i) = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \tag{3}$$

For reference, visual representations of these features are shown in Fig. 5. Due to the underlying point cloud offering a very high density and relatively uniform spacing, a K-nearest neighbourhood was chosen, as it has proven to be more predictable in terms of performance and memory requirements than radius-based neighbourhoods. As a way of improving performance, all neighbourhoods were pre-calculated, effectively forming a neighbourhood graph. Furthermore, this allows for the removal of outliers by checking if the minimal distance of a point to its neighbours exceeds a pre-defined threshold.

3.3.1. Region growing

As a method originally pioneered in image segmentation, region growing has later been adapted for use with point clouds. Unlike most methods which use voxels or octrees due to their simplicity and speed, the presented approach acts directly on the point clouds to yield a fine-grained segmentation. Using the verticality, planarity and curvature features, the steps for segmentation are as follows: all points are sorted by their planarity in descending order. Region growing is afterwards applied to each unlabelled point in the sorted list, starting with the highest-planarity point. Starting at a selected seed point, it is added as a seed to the frontier. The growth process will add and label the points in the frontier and check their unlabelled neighbours. These neighbours will then be added to the frontier if they fulfil specific constraints. In this case, the difference between the frontier's and neighbour's curvatures and verticalities are required to be below pre-defined thresholds. This way, smoothness and orientation of the surface play a decisive role in segmentation, with bending surfaces being segmented as long as their shape remains consistent. While the required thresholds may also be defined automatically by evaluating the available point cloud features statistically, user-defined thresholds were found to deliver more stable results. For clarity, the general workflow of the region growing method is depicted in Fig. 6.

Despite labels obtained from the image segmentation masks being obvious seed point candidates, using them would result in planar segments being partially claimed by separate regions with possibly conflicting labels. Hence, seed points with maximal planarity have proven superior to seed points based on labels which may be imprecise at object borders. Given these seed properties and defined region growth constraints, the resulting regions are generally planar, with vegetation and other highly-irregular surfaces only consisting of single points or small patches which can be discarded based on their limited size (see Fig. 7). Regions will then be assigned a label based on the labels of the points inside them. Given that labels are assumed to suffer from very few false positives and mostly false negatives due to occlusions, all points within a region are labelled as one of the bridge components

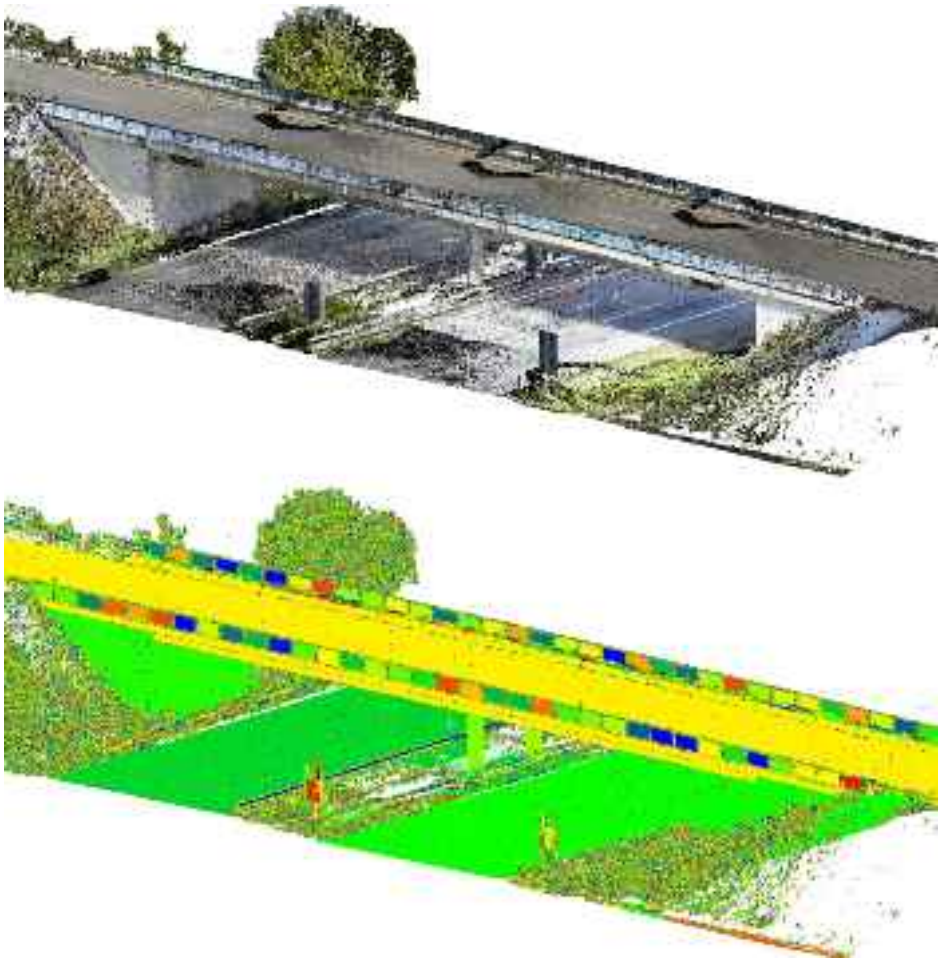


Fig. 7. Results of the region growing algorithm after fine segmentation step. Top: Input point cloud. Bottom: Resulting regions. The algorithm favours smooth surfaces and treats sharp geometrical edges as segment boundaries. The tolerance towards small variations in local curvature allows the round columns to be treated as individual segments. Vegetation provokes oversegmentation in the form of small segments.

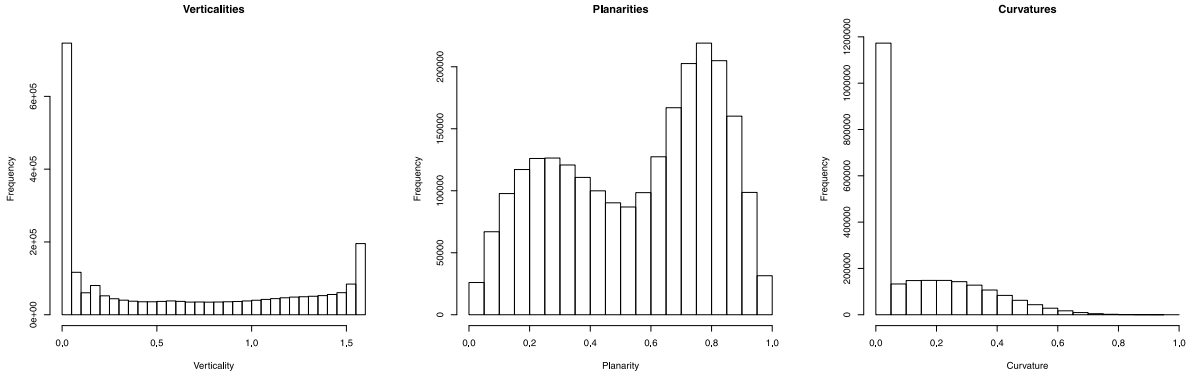


Fig. 8. Histograms for point cloud features. Left: Verticalities display two distinct peaks, which identify horizontal and vertical surfaces. Values between the peaks belong to vegetation and clutter objects. Centre: Planarites appear to consist of two overlapping distributions with high standard deviation, indicating that they do not allow for a sharp distinction between man-made and natural objects. Right: Curvatures have a distinct peak related to man-made surfaces such as roads and bridge components.

(here: *abutment*, *deck*, *railing*), if at least a minimum number of points within them has a bridge component label. Possible conflicts between competing object labels are resolved using majority voting.

3.3.2. Feature-based filtering

While the previous steps for region growing has proven instrumental in the labelling of points originally missed by the image-based semantic segmentation and label projection steps, the removal of false positives and incorrectly labelled regions is not their main objective. These artefacts are usually produced by the 2D classifier, as it has no knowledge of 3D information. In a worst case scenario with extremely low-quality input, they may impact the results negatively as well. Point filtering based on geometric constraints can however rectify these issues and remove points where these constraints are inconsistent with their label. The constraints for this task are based on the curvature and verticality, which typically lie within specific ranges (see Fig. 8). As evident by the distribution of these features, a simple automated way for determining

suitable parameters is the use of quantiles. Verticalities have two dominant peaks related to artificial structures, whereas curvatures have one single peak related to them. Using these geometric cues, it is easy to identify bridge components (*abutment*, *deck*, *railing*) and background objects. Points labelled *abutment* and *railing* with verticalities lower than the global 75% quantile typically represent false positives and can be labelled as background. Likewise, all labelled points with curvatures higher than the 75% quantile are considered false positives too and are relabelled as background objects.

Planarities however cannot be used for clean separation of the represented classes due to their global distribution resembling two overlapping Gaussian distributions. This means that a clean separation between the classes is not possible and may result in the removal of true positive points. Consequently, planarities are ignored for filtering.

4. Evaluation

For evaluation purposes, a bridge located at the A9 motorway near Munich, Germany was chosen as a reference object. A high-quality TLS point cloud captured from 20 positions with a Riegl VZ-400 and evenly downsampled to 4 mm resolution was generated alongside 140 images captured with a Nikon D800 camera which was mounted on the scan device. Hand-segmented data sets of the point cloud and images were generated for use as ground truth data for the evaluation.

In accordance with the workflow, results are presented for each individual stage. Image classification and point cloud classification are evaluated independently from one another using commonly-used statistical scores based on *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP) and *False Negatives* (FN):

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$overall\ Precision = \frac{1}{N}\sum_{i=0}^{N} Precision_i \tag{5}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$overall\ Accuracy = \frac{1}{N}\sum_{i=0}^{N} Accuracy_i \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$mean\ IoU = \frac{1}{N}\sum_{i=0}^{N} IoU_i \tag{9}$$

All scores relate to the classes *railing*, *abutment* and *deck*. For image classification averaged scores were calculated. Additionally, mean average scores are provided as aggregates for the scores of all N classes. Results for the evaluation point cloud are split into scores for pre-segmentation and post-processing.

4.1. Image semantic segmentation

Even with a limited pool of training images, training and validation loss of the Mask R-CNN model quickly converge and indicate a good fit for the evaluation subset used during training (see cross entropy loss shown in Fig. 9). The scores are quite inconsistent (see Fig. 10), but *Accuracy* is relatively high for *railing* and *abutment* with values of 78.64% and 66.02% respectively. The low *Accuracy* for *deck* components of 36.89% decreases the *mean Accuracy* to 60.52% though. Part of the reason is that evaluation images were taken in 360° rotation around the scan position, with many of them not containing any bridge components. The results in Fig. 11 prove that the classification masks for the building components generally cover and label most bridge components correctly. Due to background objects being correctly and consistently ignored, this means that the number of True Negatives benefits the scores. Common segmentation issues stem from incomplete

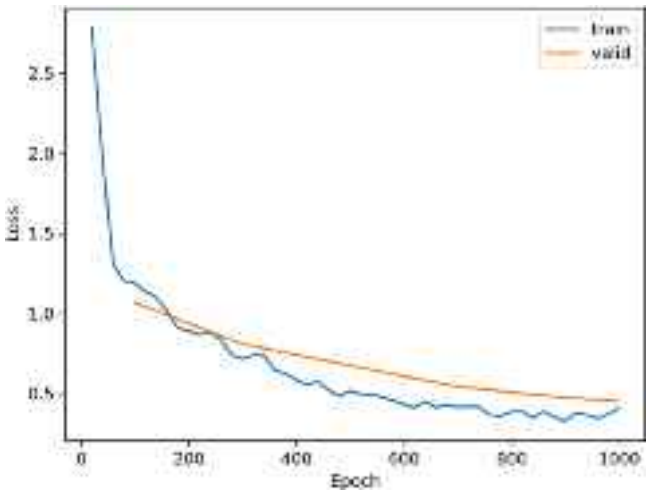


Fig. 9. Training and evaluation cross entropy loss of Mask R-CNN. The model converges quite quickly and indicates the benefit of transfer learning.

masks and imprecise mask borders, which lessen the *IoU*, bumping the mean IoU down to 4.32%. This problem is more prevalent for structures extremely close or far away from the camera where even human inspectors have a hard time telling components apart. Especially up close, distinguishing between the deck and abutment components seems to pose a challenge, as they share a similar texture. The affected masks for either of these structures are thus mislabelled with poor distinction between classes, as not enough contextual cues are present for the model to work with. Due to these regions covering much of the relative area in the affected images, every mislabelled region strongly contributes to the *mean IoU* and adds a large bias. Raw scores such as *Recall* for *deck* (9.68%) and *abutment* (0.1%) suffer as a result and leave a poor impression contrasting with the high *Recall* of the *railing* (75.0%). Therefore it stands to reason that either the amount of information given in the images is insufficient or that the model is still undertrained and does not contain a sufficient number of camera perspectives, bridge types and/or component variations. As the following section shows, combining masks from multiple overlapping perspectives does however redeem the discussed shortcomings.

4.2. Point cloud semantic segmentation before post-processing

As a baseline for semantic segmentation, hand-segmented ground truth segmentation masks of the presented images are projected into the point cloud to determine an idealized result for the best possible outcome of the presented technique.

As seen in Fig. 12, even label projection of the ground truth masks has the tendency to miss regions with poor visibility like the upper side of the deck or the backside of the columns and has a negative impact on averaged *Recall* (69.21%) and *IoU* (64.29%) in such cases. *Railing* and *Deck* display low *Recall* (58.38% and 61.98%) in comparison, which contrast with the high *Recall* for the *abutment* (91.79%). Cause of these problems is the fact that regions not visible in any image are not labelled due to lack of data, but the strategy for resolving conflicting labels from different images also partially contributes to this problem. Boundary regions where different bridge components meet are occasionally incorrectly labelled. The aforementioned idealized results achievable under these conditions are depicted in Fig. 13 (left) and are calculated by comparing the segmented point cloud resulting from projection of the ground truth masks to the fully-labelled ground truth 3D point cloud data. The averaged scores for *Precision* (89.65%) and *Accuracy* (91.61%) show that high-quality results are theoretically possible under idealized circumstances, with the *Accuracies* for *Railing* (97.63%) and *Abutment* (98.85%) being notable standouts.

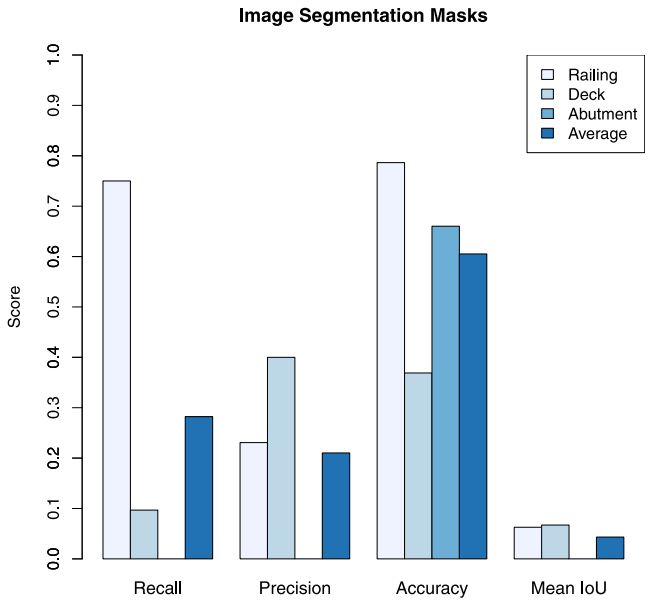


Fig. 10. Performance scores (Recall, Precision, Accuracy and Mean IoU) for semantic image segmentation masks generated by Mask R-CNN.

When projecting the classification masks generated by Mask R-CNN, results strongly depend on the quality of the segmentation masks. Fig. 13 (centre) shows how *Recall* (11.71%) and *IoU* (11.68%) are drastically worse when compared to the projected ground truth masks. This effect is caused by the low number of *True Positive* points and low overall coverage of the bridge components. *Accuracy* (82.71%) remains quite high though due non-bridge points being ignored and attests that the detection of components is highly reliable. *Precision* (98.21%) seems paradoxically high as well, but the relatively high number of background points compared to a low number of bridge component points means that the *True Negatives* lead to a dominance of the enumerator, thus skewing the term towards 100%.

The low *Recall* and *IoU*, are quite obviously caused by large parts of the *True Positives* being missed and some occasional labelling artefacts. Noteworthy in this context is that the point cloud density can lead to problems in the projection stage. Camera positions close to sparse regions may lead to gaps in the depth map which result in labels being projected at points which should not be visible. Another issue are classification artefacts caused by incorrect classification masks which occasionally manifest themselves in isolated regions. These *False Positives* oftentimes originate from vegetation being inside the projection frustum or imprecise classification mask borders. They are quite easy to detect though and therefore subject to post-processing.

4.3. Point cloud semantic segmentation through post-processing

A dominant issue seen in the previous section are partially labelled regions. Despite labelled segments being incomplete, their precision indicates that they serve as high-quality input for the region growing algorithm. Consequently, post-processing builds on this foundation and ensures that missing bridge components no longer remain unlabelled. Fig. 14 shows that the visual jump in quality is quite apparent. While the relabelling based on region growing can also affect and grow incorrectly labelled regions, the feature-based filtering remedies such issues. Since the resulting labels are quite close to the ground truth labels of the point cloud, scores for projected reference image segmentation masks combined with post-processing are henceforth not discussed.

As shown in Fig. 13 (right), segmentations for *abutment* and *deck* are now almost completely correct (with *Accuracies* of 99.55% and 92.41%), with errors only occurring at structures on the bottom side of the deck which are completely occluded from camera view. Large parts of the *railing* however remain missing and impact the averaged scores negatively (*Recall* with 8.08% in particular). Reason for this is the geometric structure of the railing which makes segmentation of its non-planar components difficult. This problem is furthermore compounded by poor visibility of the *railing* in images, which means that both, the initial image semantic segmentation and the post-processing do not get much information for labelling and propagation. Results consequently depend on how well asset component classes have been recognized in the image segmentation step.

Despite limited training data and inferior results in image processing, the overall results can not only compete with the idealized results obtained by projecting the ground truth segmentation masks, but even outperform them in most cases. Averaged *Recall* (62.73%; 6.48% decrease) and *IoU* (61.69%; 2.6% decrease) are now on almost equal levels while *Precision* (98.51%; 6.9% improvement) and *Accuracy* (94.74%; 5.09% improvement) are even slightly better now. The scores for each individual object class support these findings and prove that the post-processing considerably aids in improving segmentation quality.

5. Discussion

Training data for semantic segmentation on 3D point clouds using machine learning is quite hard to come by in the infrastructure field. Given the prevalence of this issue for point clouds, the presented image-based approach proves to be a viable alternative. Notable problems are caused by limited image resolutions and camera perspectives, which lead to occluded points or regions far away from the camera not being labelled correctly during the image segmentation and projection pass. Furthermore, the positions of TLS devices and consequently the camera mounted on them are restricted to areas simultaneously providing decent coverage of the 3D geometry and safety from incoming traffic. However this conflicts with the idealized camera positions, as positions very close to large bridge components allow large parts of the 3D geometry to be captured, but result in pictures of homogeneous surfaces with little context for image segmentation. This means that either more freedom in scan position choice or a separate pass for image capturing are preferable. While permits, accessibility and safety distances still apply, MLS and photogrammetric drone footage might yield better results than TLS and could be used to complement them. Additionally, projection of the hand-segmented reference masks indicates that a larger pool of training data could improve results further. However, the presented methods for post-processing have proven quite capable of dealing with problems like missing or conflicting labels and even assign labels to areas not visible in the images. The high precision and reliability of the image-based pre-segmentation complements and supports the region growing substantially by providing the necessary cues required for propagating labels into missing regions. Incorporating domain knowledge and geometric reasoning further adds to this by removing segmentation artefacts through filtering.

6. Outlook

The results show that even with a limited pool of training images and a high potential for gaining superior results by incorporating more training data, image-based semantic segmentation holds much potential for future projects, as it allows for quick labelling of arbitrary point clouds regardless of their size. No point cloud training data is required and common issues related to deep learning on point clouds are avoided without resorting to workarounds like reducing the point cloud resolution. The combination of this technique with point cloud segmentation algorithms and filtering based on geometric features furthermore represents a perspective rarely explored in most works. As shown in this article, the benefits are quite apparent and non-learning-based algorithms like selective search [59] have already been

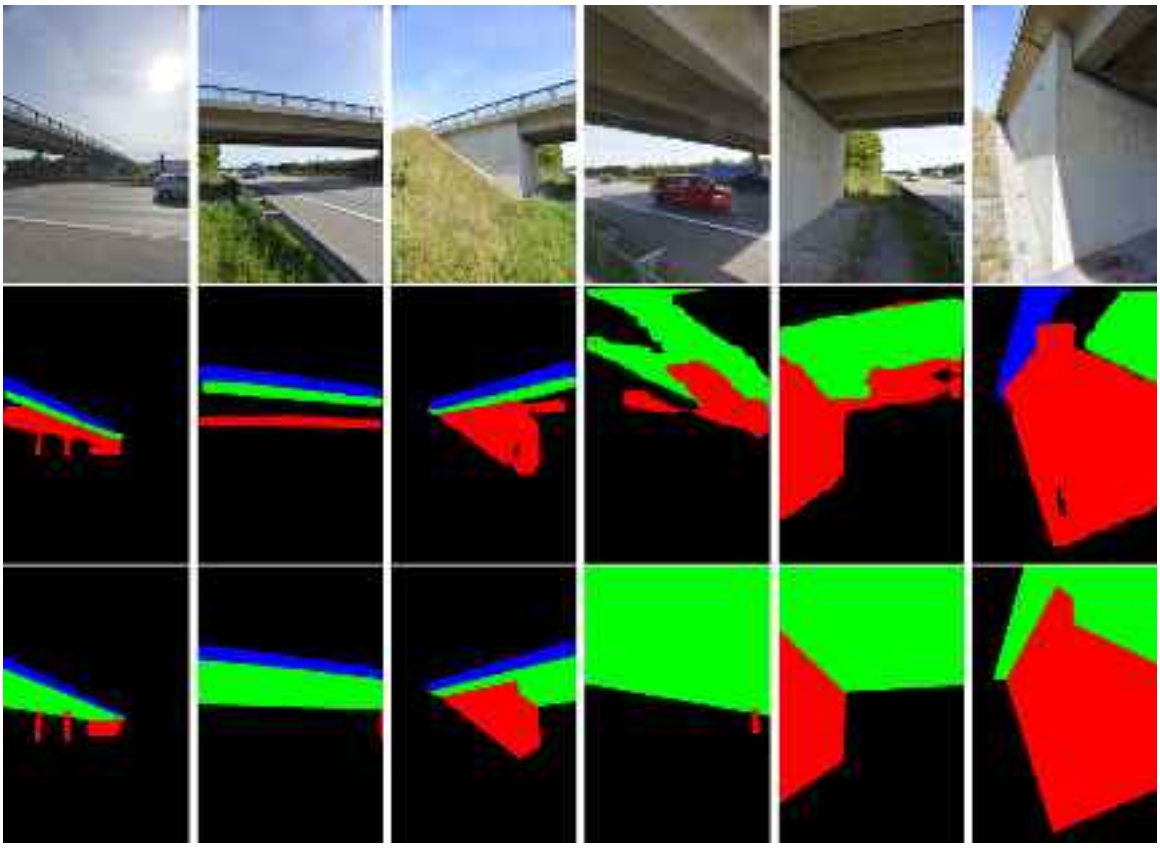


Fig. 11. Examples for generated image segmentation masks. Top row: input images used for semantic segmentation. Centre row: Generated classification masks. Bottom row: Ground truth masks. Red masks represent abutment, green masks the deck and blue masks the railing.

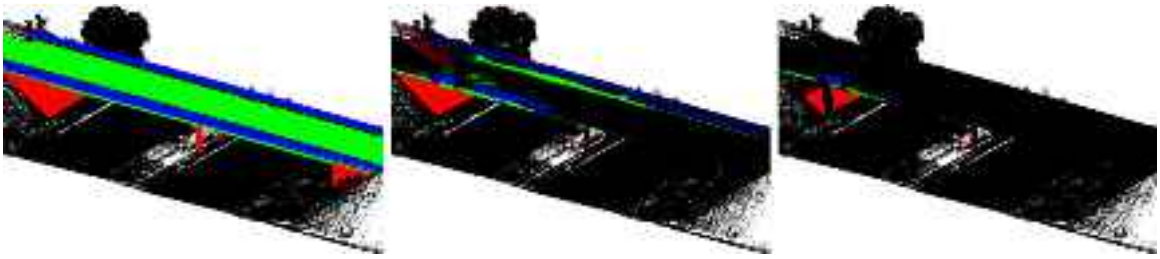


Fig. 12. Label projection results. Red represents the abutment, green the deck, blue the railing and black points are background objects. Left: Ground truth labels. Centre: Labels resulting from projection of ground truth image classification masks. Right: Labels resulting from projection of classification masks generated by Mask R-CNN model. Regions like the backside of the columns or large sections of the deck suffer from poor visibility and are left unlabelled after label projection. This issue even applies to the projected ground truth masks.

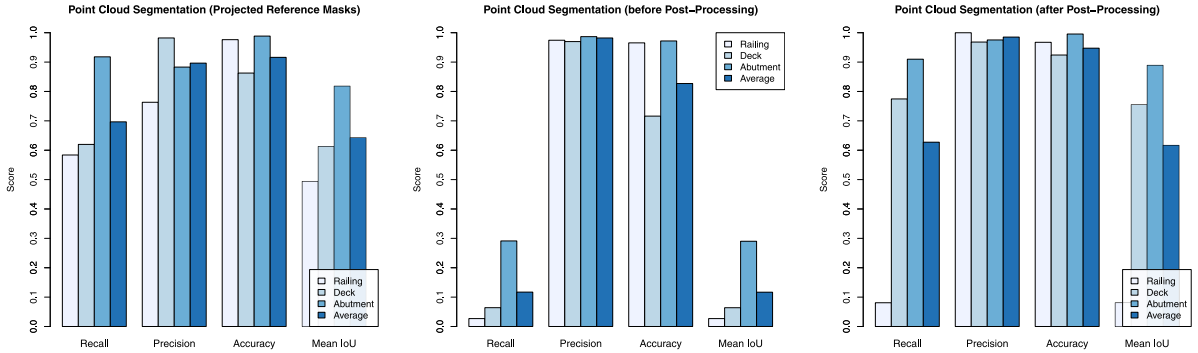


Fig. 13. Performance scores (Recall, Precision, Accuracy and IoU) for semantic segmentation of the bridge components in point clouds. Left: Idealized scores for the ground truth image masks projected to the point cloud. Centre: Scores for projected image masks generated by Mask R-CNN model before post-processing. Right: Scores for projected image masks generated by Mask R-CNN model after applying post-processing.

Fig. 14. Labels achieved through post-processing in the fine segmentation step. Red represents the abutment, green the deck, blue the railing and black points are background objects. Left: Projected labels of the Mask R-CNN model. Centre: After region growing, labels are assigned to partially unlabelled or occluded regions. Right: Filtering removes incorrect labels propagated along surfaces using geometric reasoning.

used as pre-processing steps for image-based segmentation. Pioneering works in image processing like the initial R-CNN [26] which rely on such combinations of pre-processing and ANNs further highlight this perspective. In consequence, combinations of machine learning and algorithms designed with domain-specific knowledge still hold much untapped potential, where advantages of either method are reflected.

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.

Data availability

The data that has been used is confidential.

Acknowledgements

This work has been supported and funded by the *German Federal Ministry for Digital and Transport* (BMDV) as part of the *TwinGen* research project. The authors thank their project partners from the *Chair of Computational Modelling and Simulation at the Technical University of Munich*, *Computing in Engineering at Ruhr University Bochum*, *Design Computation at RWTH Aachen University* and *ZPP Ingenieure AG*.

References

[1] A. Borrmann, M. König, C. Koch, J. Beetz (Eds.), Building surveying for as-built modeling, in: Building Information Modeling: Technology Foundations and Industry Practice, Springer International Publishing, Cham, 2018, pp. 393–411, http://dx.doi.org/10.1007/978-3-319-92862-3_24.

[2] R. Sacks, I. Brilakis, E. Pikas, S. Xie, M. Girolami, Construction with digital twin information systems, Data-Centric Eng. 1 (2020) 26, http://dx.doi.org/10.1017/dce.2020.16.

[3] C.-S. Shim, N.-S. Dang, S. Lon, C.-H. Jeon, Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model, Struct. Infrastruct. Eng. 15 (10) (2019) 1319–1332, http://dx.doi.org/10.1080/15732479.2019.1620789.

[4] I. Errandonea, S. Beltrán, S. Arrizabalaga, Digital twin for maintenance: A literature review, Comput. Ind. 123 (2020) 103316.

[5] A. Borrmann, M. König, C. Koch, J. Beetz (Eds.), Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models, in: Building Information Modeling: Technology Foundations and Industry Practice, Springer International Publishing, Cham, 2018, pp. 81–126, http://dx.doi.org/10.1007/978-3-319-92862-3_5.

[6] A. Borrmann, S. Muhic, J. Hyvärinen, T. Chipman, S. Jaud, C. Castaing, C. Dumoulin, T. Liebich, L. Mol, The IFC-bridge project – extending the IFC standard to enable high-quality exchange of bridge information models, 2019, pp. 377–386, http://dx.doi.org/10.35490/EC3.2019.193.

[7] M. Bassier, M. Yousefzadeh, M. Vergauwen, Comparison of 2D and 3D wall reconstruction algorithms from point cloud data for as-built BIM, J. Inf. Technol. Constr. 25 (2020) 173–192, http://dx.doi.org/10.36680/j.itcon.2020.011.

[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, http://dx.doi.org/10.1145/3065386.

[9] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, 2016, CoRR abs/1612.00593.

[10] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J.N. Chiang, Z. Wu, X. Ding, Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation, Med. Image Anal. 63 (2020) 101693.

[11] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3D point clouds: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2021) 4338–4364, http://dx.doi.org/10.1109/TPAMI.2020.3005434.

[12] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, M.A. Chapman, Mobile laser scanned point-clouds for road object detection and extraction: A review, Remote Sens. 10 (10) (2018) http://dx.doi.org/10.3390/rs10101531.

[13] E. Che, J. Jung, M.J. Olsen, Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review, Sensors 19 (4) (2019) http://dx.doi.org/10.3390/s19040810.

[14] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset, Int. J. Robot. Res. (2013).

[15] X. Roynard, J.-E. Deschaud, F. Goulette, Paris-lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification, Int. J. Robot. Res. 37 (6) (2018) 545–557, http://dx.doi.org/10.1177/0278364918767506.

[16] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, A. Kim, Complex urban dataset with multi-level sensors from highly diverse urban environments, Int. J. Robot. Res. 38 (6) (2019) 642–657.

[17] P. Bhagat, P. Choudhary, Image annotation: Then and now, Image Vis. Comput. 80 (2018) 1–23.

[18] W. Zhu, B. Braun, L.H. Chiang, J.A. Romagnoli, Investigation of transfer learning for image classification and impact on training sample size, Chemometr. Intell. Lab. Syst. 211 (2021) 104269.

[19] R. Becker, Differentialgeometrische Extraktion von 3D-Objektprimitiven aus terrestrischen Laserscannerdaten, 2005.

[20] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, Comput. Graph. Forum 26 (2) (2007) 214–226.

[21] Z. Ma, S. Liu, A review of 3D reconstruction techniques in civil engineering and their applications, Adv. Eng. Inform. 37 (2018) 163–174, http://dx.doi.org/10.1016/j.aei.2018.05.005.

[22] H. Son, F. Bosché, C. Kim, As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey, Adv. Eng. Inform. 29 (2) (2015) 172–183, http://dx.doi.org/10.1016/j.aei.2015.01.009, Infrastructure Computer Vision.

[23] I. Brilakis, M. Lourakis, R. Sacks, S. Savarese, S. Christodoulou, J. Teizer, A. Makhmalbaf, Toward automated generation of parametric BIMs based on hybrid video and laser scanning data, Adv. Eng. Inform. 24 (4) (2010) 456–465, http://dx.doi.org/10.1016/j.aei.2010.06.006, Construction Informatics.

[24] N. Brodu, D. Lague, 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology, ISPRS J. Photogramm. Remote Sens. 68 (2012) 121–134.

[25] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 779–788, http://dx.doi.org/10.1109/CVPR.2016.91.

[26] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, http://dx.doi.org/10.1109/CVPR.2014.81.

[27] R. Girshick, Fast R-CNN, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 1440–1448, http://dx.doi.org/10.1109/ICCV.2015.169.

[28] K. He, G. Gkioxari, P. Dollar, R.B. Girshick, Mask R-CNN, CoRR abs/1703.06870, 2017, arXiv:1703.06870.

[29] W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, A.E. Sallab, YOLO3D: End-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud, in: Computer Vision – ECCV 2018 Workshops, Springer International Publishing, Cham, 2019, pp. 716–728.

[30] M. Simon, S. Milz, K. Amende, H.-M. Gross, Complex-YOLO: An Euler-region-proposal for real-time 3D object detection on point clouds, in: Computer Vision – ECCV 2018 Workshops, Springer International Publishing, Cham, 2019, pp. 197–209.
[31] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, PointPillars: Fast encoders for object detection from point clouds, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 12689–12697, http://dx.doi.org/10.1109/CVPR.2019.01298.
[32] S. Mohapatra, S. Yogamani, H. Gotzig, S. Milz, P. Mader, BevDetNet: Bird's eye view LiDAR point cloud based real-time 3D object detection for autonomous driving, in: 2021 IEEE International Intelligent Transportation Systems Conference, ITSC, 2021, pp. 2809–2815, http://dx.doi.org/10.1109/ITSC48978.2021.9564490.
[33] D. Maturana, S. Scherer, VoxNet: A 3D convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2015, pp. 922–928, http://dx.doi.org/10.1109/IROS.2015.7353481.
[34] Y. Zhou, O. Tuzel, VoxelNet: End-to-end learning for point cloud based 3D object detection, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490–4499.
[35] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, 2017, arXiv preprint arXiv:1706.02413.
[36] V.A. Sindagi, Y. Zhou, O. Tuzel, MVX-net: Multimodal VoxelNet for 3D object detection, in: 2019 International Conference on Robotics and Automation, ICRA, 2019, pp. 7276–7282, http://dx.doi.org/10.1109/ICRA.2019.8794195.
[37] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, V. Vasudevan, End-to-end multi-view fusion for 3D object detection in LiDAR point clouds, 2019.
[38] A. Boulch, ConvPoint: Continuous convolutions for point cloud processing, Comput. Graph. 88 (2020) http://dx.doi.org/10.1016/j.cag.2020.02.005.
[39] N. Engel, V. Belagiannis, K. Dietmayer, Point transformer, IEEE Access 9 (2021) 134826–134840, http://dx.doi.org/10.1109/ACCESS.2021.3116304.
[40] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, RandLA-Net: Efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020.
[41] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3D shape recognition, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 945–953, http://dx.doi.org/10.1109/ICCV.2015.114.
[42] V. Stojanovic, M. Trapp, J. Döllner, R. Richter, Classification of indoor point clouds using multiviews, in: The 24th International Conference on 3D Web Technology, in: Web3D '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–9, http://dx.doi.org/10.1145/3329714.3338129.
[43] J. Wolf, R. Richter, S. Discher, J. Döllner, Applicability of neural networks for image classification on object detection in mobile mapping 3D point clouds, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-4/W15 (2019) 111–115, http://dx.doi.org/10.5194/isprs-archives-XLII-4-W15-111-2019.
[44] T. Czerniawski, F. Leite, Automated segmentation of RGB-D images into a comprehensive set of building components using deep learning, Adv. Eng. Inform. 45 (2020) 101131, http://dx.doi.org/10.1016/j.aei.2020.101131.
[45] I. Abundez, C. Estrada, S. Zagal, M. Perez, Segmentation of medical images by region growing, 2008, pp. 441–444, http://dx.doi.org/10.1109/IRI.2008.4583071.
[46] H. Mahmoudabadi, M. Olsen, S. Todorovic, Efficient terrestrial laser scan segmentation exploiting data structure, ISPRS J. Photogramm. Remote Sens. 119 (2016) 135–150, http://dx.doi.org/10.1016/j.isprsjprs.2016.05.015.
[47] R. Adams, L. Bischof, Seeded region growing, IEEE Trans. Pattern Anal. Mach. Intell. 16 (6) (1994) 641–647, http://dx.doi.org/10.1109/34.295913.
[48] A. Pratondo, S.H. Ong, C.K. Chui, Region growing for medical image segmentation using a modified multiple-seed approach on a multi-core CPU computer, in: The 15th International Conference on Biomedical Engineering, Springer International Publishing, Cham, 2014, pp. 112–115.
[49] Z. Jin, T. Tillo, W. Zou, X. Li, E. Lim, Depth image-based plane detection, Big Data Anal. 3 (2018) 1–18, http://dx.doi.org/10.1186/s41044-018-0035-y.
[50] J. Papon, A. Abramov, M. Schoeler, F. Wörgötter, Voxel cloud connectivity segmentation - supervoxels for point clouds, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2027–2034, http://dx.doi.org/10.1109/CVPR.2013.264.
[51] T. Rabbani, F. Heuvel, G. Vosselman, Segmentation of point clouds using smoothness constraint, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 36 (2006).
[52] H. Masuta, S. Makino, H.-o. Lim, 3D plane detection for robot perception applying particle swarm optimization, in: 2014 World Automation Congress, WAC, 2014, pp. 549–554, http://dx.doi.org/10.1109/WAC.2014.6936041.
[53] P. Besl, R. Jain, Segmentation through variable-order surface fitting, IEEE Trans. Pattern Anal. Mach. Intell. 10 (2) (1988) 167–192, http://dx.doi.org/10.1109/34.3881.
[54] L. Truong-Hong, R. Lindenbergh, Extracting structural components of concrete buildings from laser scanning point clouds from construction sites, Adv. Eng. Inform. 51 (2022) 101490, http://dx.doi.org/10.1016/j.aei.2021.101490.
[55] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.
[56] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 5987–5995, http://dx.doi.org/10.1109/CVPR.2017.634.
[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, L. Zitnick, Microsoft COCO: Common objects in context, in: ECCV, 2014.
[58] W. Straßer, Schnelle Kurven- und Flächendarstellung auf grafischen Sichtgeräten (Ph.D. thesis), Technical University of Berlin, 1974.
[59] J. Uijlings, K. Sande, T. Gevers, A. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2013) 154–171, http://dx.doi.org/10.1007/s11263-013-0620-5.