3D detection of roof sections from a single satellite image and application to LOD2-building reconstruction

Johann Lussange, Mulin Yu, Yuliya Tarabalka, Florent Lafarge

3D detection of roof sections from a single satellite image and application to LOD2-building reconstruction

Johann Lussange[1] JOHANN.LUSSANGE@INRIA.FR
Mulin Yu[1] MULIN.YU@INRIA.FR
Yuliya Tarabalka[2] YTARABALKA@LUXCARTA.COM
Florent Lafarge[1] FLORENT.LAFARGE@INRIA.FR

[1] *INRIA Sophia Antipolis Méditerrannée, 2004 route des Lucioles, 06902, Valbonne, France.*
[2] *LuxCarta Technology, 460 avenue de la Quiéra, voie K, bat. 119 B, 06370, Mouans-Sartoux, France.*

Abstract

Reconstructing urban areas in 3D out of satellite raster images has been a long-standing and challenging goal of both academical and industrial research. The rare methods today achieving this objective at a Level Of Details 2 rely on procedural approaches based on geometry, and need stereo images and/or LIDAR data as input. We here propose a method for urban 3D reconstruction named KIBS(*Keypoints Inference By Segmentation*), which comprises two novel features: i- a full deep learning approach for the 3D detection of the roof sections, and ii- only one single (non-orthogonal) satellite raster image as model input. This is achieved in two steps: i- by a Mask R-CNN model performing a 2D segmentation of the buildings' roof sections, and after blending these latter segmented pixels within the RGB satellite raster image, ii- by another identical Mask R-CNN model inferring the heights-to-ground of the roof sections' corners via panoptic segmentation, unto full 3D reconstruction of the buildings and city. We demonstrate the potential of the KIBS method by reconstructing different urban areas in a few minutes, with a Jaccard index for the 2D segmentation of individual roof sections of 88.55% and 75.21% on our two data sets resp., and a height's mean error of such correctly segmented pixels for the 3D reconstruction of 1.60 m and 2.06 m on our two data sets resp., hence within the LOD2 precision range.

1. Introduction

In the rapidly evolving era of smart cities and intelligent urbanization, digital city models have become crucial tools for urban planning, environmental analysis, and infrastructure management. Relying on satellite, aerial, and *LIght Detection and Ranging* (LIDAR) imagery, these models offer detailed three-dimensional representations of urban environments, and facilitate better-informed decision-making processes. At the same time, computer vision research in satellite and aerial imagery has made great strides in recent years. However, the unique challenges posed by satellite, aerial, and LIDAR imagery, such as variation in perspective, scale, lighting, atmospheric conditions, and data density, necessitate constant technical advancements. New algorithms and models have allowed for much more accurate and efficient image analysis, notably with the rise of deep learning methods. In a larger scope, these have shown much promise in automatically detecting and classifying objects in images [64, 25, 10]. This object detection and classification is especially relevant to the fields of semantic segmentation [15] and 3D reconstruction [24]. With the ever-increasing availability of data (notably LIDAR data [3, 64]), new applications are constantly arising and allow for the automation of detection and correction of various types of distortion in images, such as those caused by atmospheric conditions [26] and the curvature of the Earth's surface [23], or building [9, 42, 11] and vegetation occlusion [53], etc. Also, new methods are being developed for automatically extracting information from raster images [16] such as land cover [21], or topographical features [44]. In this Paper, we will first give a brief overview of the related work and other pertaining methods in Section 2. We will then proceed to explain our own proposed KIBS (*Keypoints Inference By Segmentation*) approach in

Section 3, where we will describe the model's two-steps architecture and its data post-processing. In Section 4, we will then describe our experiment, with a presentation and discussion of the results of the method on our data set, together with details on the model generalisation and limitations. We also describe the training, validation and testing procedure for Mourmelon-le-Grand and Sissonne data sets, for both the 2D segmentation and the 3D reconstruction of the KIBS method in the Section 6 of the Supplementary Material.

2. Related work

The interest of using satellite data as input for reconstruction relies on the abundance and low costs of such data, compared to other sources such as LIDAR or aerial data, which face legal or technical constraints, flight authorisation issues over certain areas, etc. The Level Of Details (or LOD) is a usual metric that allows one to specify the desired precision of such reconstruction. As shown on Fig. 1, LOD1 denotes a building reconstruction precision looking like a rectangular shoe box, LOD2 denotes a reconstruction precision displaying the shape of the building's roof, while LOD3 denotes a reconstruction precision of objects' sizes below this range, such as windows, balconies, etc. A few studies have claimed to perform urban 3D reconstruction at a LOD1 [52], or similar outcomes of ground surface reconstructions, but the portability of such methods have often remained limited due to their highly procedural architectures [29]. A method for 3D reconstruction at LOD2 is being patented [38], but proposes a very different approach than the one-shot procedure presented here, by using pre-existing primes of rooftops. Most of such methods also rely on extra data sets that are not purely of satellite origin [63, 8], such as LIDAR data [5, 32], aerial photography [59], etc. Others rely on data sets of pre-existing primes of rooftops [47, 38]. We here review the latest advancements in 3D plane detection and reconstruction, which is an active area of research within computer vision, with substantial contributions made through the use of both single-image and multi-view images or point cloud data.

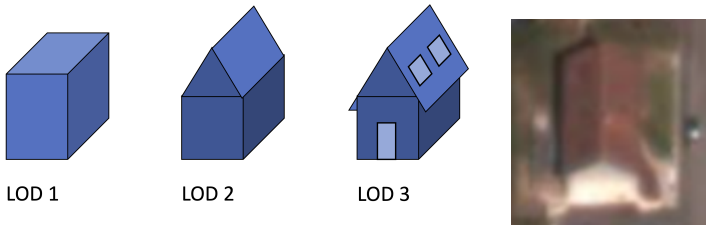

Figure 1: Left: examples of Level of Details (LOD) of 3D reconstructions. Right: random residential building in our satellite image input data set of Mourmelon-le-Grand, which has a precision of 0.38 meter per pixel, to compare its precision with that of the 3D reconstruction goal at a LOD2 (cf. car parked on the right-hand side of the house).

Plane detection from single image Single-image plane detection and reconstruction have seen remarkable progress thanks to advancements in deep learning. Researchers have developed several methods to detect and reconstruct planes using just a single image. For example, in the PlaneFormers paper [2], they utilize deep learning to develop an algorithm that can reconstruct 3D planes from sparse view planes. Another method, PlaneRCNN, was proposed by [36] that detects and reconstructs 3D planes from a single image using a Region Convolutional Neural Network. Similarly, [61] proposed a method for single-image piece-wise planar 3D reconstruction via associative embedding, and [35] introduced PlaneNet for piece-wise planar reconstruction from a single RGB image. Further, [58] employed convolutional neural networks for recovering 3D planes from a single image, highlighting

the potential of deep learning in plane detection and reconstruction from single images. However these methods are designed to extract a few large planes in certain types of images, typically indoor scenes, and fails to detect the numerous small planes, e.g. hundred of thousand, contained in a satellite image representing a city.

Plane detection from point clouds and multiview images Moving beyond single images, point clouds and multiview images offer additional information that can be leveraged for plane detection and reconstruction. Classical methods such as region growing [46, 54] and RANSAC [48, 51] have been widely used for this purpose. On the other hand, energy minimization methods [41, 60] provide a more rigorous approach, leveraging the mathematical foundation of energy functions for plane detection. Scale-space exploration, as demonstrated by [13, 27], is another valuable technique that adapts to various scales for improved detection. Recently, deep learning-based methods have shown great promise, offering new opportunities for plane detection from point clouds and multiview images [30, 49, 39, 22, 57]. Unfortunately, such techniques cannot be used in our context where point clouds generated by MVS from satellite imagery have a very low precision on the spatial coordinates of points.

Building reconstruction Roof reconstruction has been a challenging task in 3D building modeling, requiring special attention. Different data sources provide different opportunities and challenges for roof reconstruction. In this context, roof skeletonization techniques [47] and deep learning-based aerial image analysis methods [59, 63] have shown promising results. In the case of LiDAR data, methods like [3] have proven effective. Generative models like Roof-GAN [45] have demonstrated the ability to learn and generate roof geometry and relations for residential houses. In another approach, [62] proposed neural procedural reconstruction for residential buildings, merging the power of deep learning with the procedural generation approach. Such approaches operate from aerial data and are not robust anymore from satellite data.

Reconstructing urban environments in 3D out of satellite raster images has been a long-standing ambitious objective of both industrial and academic research [17]. One exciting application of plane detection and reconstruction is in building reconstruction from satellite images. Researchers have employed a variety of methods for this task. For instance, automated building extraction from satellite imagery is explored in [18], and building detection from remotely sensed data is studied in [43]. In the context of LOD2 models, roof type classification plays a crucial role, as seen in [29], which utilized PointNet for this purpose. For LOD1 models, beyond [52], methods like Voronoi-based algorithms [9] and polygonalization of footprints [42, 65, 31] have shown effectiveness. Furthermore, advancements in dense mesh and Digital Surface Model (DSM) generation techniques, such as IMPLICITY, which uses deep implicit occupancy fields for city modeling from satellite images [50], have further pushed the boundaries of what is possible in this domain.

To the best of our knowledge, we are the first to try reconstruct LOD2-building by detecting and assembling planes directly from one single satellite image.

3. Proposed approach

3.1 Overview

The KIBS procedure performs the 3D reconstruction of urban areas at a LOD2 with, compared to previous methods, two new features: i- a full deep learning solution for the 3D detection of the buildings' roof sections, and ii- an input consisting in only one single satellite raster image. In order to do this, the KIBS model follows a two-step procedure: first a 2D segmentation task identifies the roof sections, and then a second 3D reconstruction task infers those roof sections' corners with their

height-to-ground (as a unique class). Such monocular or single-view 3D reconstruction approaches have been recently used in the general field of computer vision [7, 6, 14], however, these methods were applied to simpler images (like those of individual objects or indoor scenes), and applying them to complex raster data like satellite imagery of urban areas is a more challenging problem.

Input The KIBS method is here trained on a data set of satellite raster images with a precision of 0.38 meter per pixel (see Fig. 1 for a sample). The input of the first data set used for the training, validation, and testing of this method derives from one RGB satellite image of Mourmelon-le-Grand, France, of size 30564×26320 pixels, corresponding to a surface area of $\sim 73\text{km}^2$. This raster image comes from Maxar's Worldview-3 satellite, and was acquired on the 13th of August 2020, with a satellite azimuth angle of 181.10°, elevation angle of 59.30°. This raster image is accompanied with a data set serving as ground truth for this outcome of urban 3D reconstruction, that consists in a hand-annotated shapefile of all individual roof sections' contours, together with their corners' heights above mean sea level. It is also accompanied with a Digital Terrain Model (DTM, i.e. an elevation map of the ground surface, without its urban or natural objects), courtesy of LuxCarta. Once the KIBS method has been developed for Mourmelon-le-Grand, the model has been trained, validated and tested on a second, similar, data set in order to further confirm its validity, this time on the city of Sissonne, France, whose raster data is of size 19120×17420, corresponding to a surface area of $\sim 25\text{km}^2$. This raster image also comes from Maxar's Worldview-3 satellite, and was acquired on the 4th of November 2020, with a satellite azimuth angle of 172.9°, elevation angle of 66.6°. It also comes with a DTM specifying the altitudes to sea-level of the terrain, and a hand-annotated shapefile of all individual roof sections' contours, together with their corners' altitudes to sea-level.

Output Once trained, the first part of the KIBS model outputs a 2D segmentation of the roof sections, which is fed into a second part of the model employing panoptic segmentation in order to derive those roof section corners' height-to-ground, so as to compute their associated 3D planes coefficients, unto full building and urban area reconstruction. We finally use the Kinetic Shape Reconstruction (KSR) method developed in [60] in order to visualize it. A sketch of the whole KIBS procedure is shown in Fig. 2.

Hypotheses The general working hypothesis of this research study is that it is possible to perform the 3D reconstruction of buildings at a LOD2, for a model taking as input only one single, non-orthogonal, satellite raster image with a resolution of 0.38 meter per pixel (see Fig. 1 for a comparison). More specifically, within the scope of the KIBS method, our working hypothesis is that a deep learning approach can segment in 2D and reconstruct in 3D the roof sections of the buildings of an urban area with a LOD2, at this image resolution, and based on a single-shot satellite raster image. The fundamental intuition behind this hypothesis is that the non-orthogonality of the satellite raster image provides the deep learning algorithms with non-trivial information (e.g. buildings' walls' inclination, buildings' shadows, roof peak or ridge perspective, etc.) allowing them to infer the height-to-ground of the roof sections' corners with a precision within the bounds of the LOD2 requirement.

3.2 2D detection of roof sections

The process for training data preprocessing for the Mask-RCNN model for 2D segmentation of roof lines involves several steps. Firstly, the initial 8687×9890 satellite image is segmented into individual 230×230 tiles, overlapping by a margin of 10 pixels. Subsequently, ground truth shapefile polygons delimiting roof sections are extracted from these tiles. Each tile then gets a set of corresponding black and white images with white pixels representing a unique roof section per image.

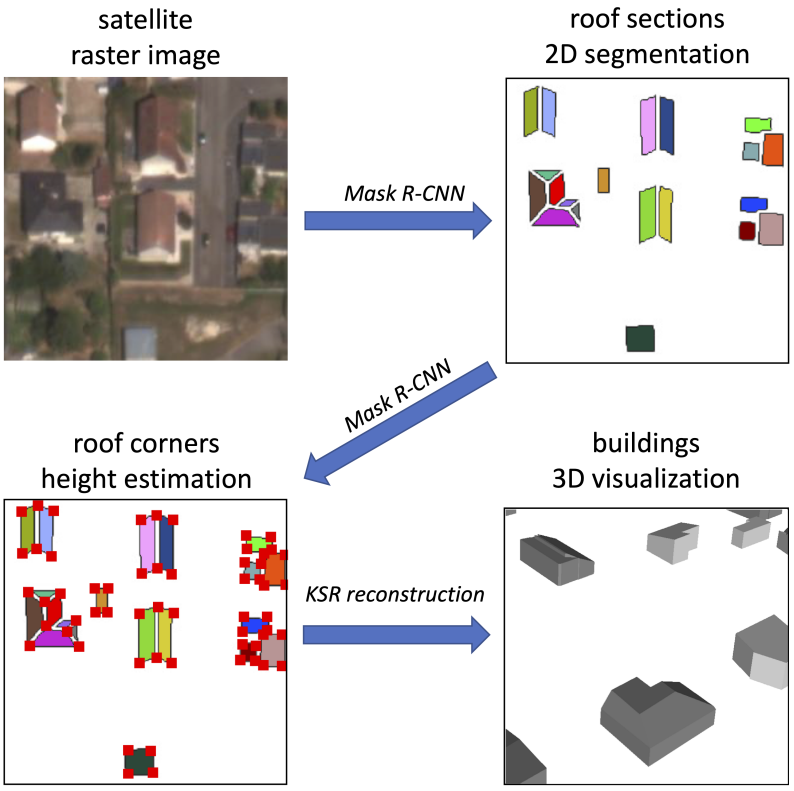

Figure 2: General procedure of the KIBS method. A first Mask R-CNN model takes the satellite raster image as input and performs an individual 2D segmentation of the buildings' roof section. Then each segmented pixel of this output is blended back into that same RGB raster image, and serves as input to a second, distinct, Mask R-CNN model in order to both identify the roof corners, and estimate their height-to-ground (as a class in meters, according to the LOD2 precision standards, i.e. 1 m, 2 m, 3 m, etc.). The inference of at least three roof corners allows one then to derive the associated roof section plane 3D coefficients, and hence recover the whole building 3D reconstruction, that one can then visualise with the KSR geometrical procedures.

Using the `pycococreator` algorithm [55], annotation files in PYCOCO format are created for these ground truth masks. After randomly shuffling the set of tiles and associated ground truth images, it is divided into three disjoint sets: training (60%), validation (20%), and testing (20%).

These sets and their associated annotation files are then fed into a Mask-RCNN neural network named `mask_rcnn_R_50_FPN_3x`, a combination of a ResNet-50 model stacked with a Feature Pyramid Network (FPN). This model was chosen due to its robustness and ability to handle complex segmentation tasks. The training, which ran for six days on a Dell T630 GPU node with four GeForce GTX 1080 Ti GPUs, was monitored via TensorBoard to manage regularization issues. The trained network weights are available on the KIBS GitHub repository. More implementation details on the training procedure, as well as the training metrics are given in Section 6.1 and 6.3 of the Supplementary Material, respectively. The weights of the trained model, which represent the learned features, are available on the KIBS GitHub repository for further exploration and reproducibility of our results [37].

3.3 3D roof corners extraction

The 3D reconstruction training leverages a Mask-RCNN model, similar to the 2D segmentation process but geared towards panoptic segmentation. This involves marking roof section corners on the image output of the 2D segmentation and assigning unique class labels to these corners, representing their heights.

After training, the 2D segmentation output is integrated with the original RGB raster image, improving the 3D reconstruction's efficiency in identifying roof corners. Class labels corresponding to specific heights are used in the Detectron2 framework, extendable to handle taller structures.

Generating the training, validation, and testing sets follows a similar procedure to the 2D segmentation. Each blended raster image is linked with ground truth images representing roof corners, and this data is processed via pycococreator to create annotation files compatible with the Detectron2 framework. A Mask-RCNN model is trained to recognize roof corners and their heights. The training process, monitored online to manage regularization issues, leverages the same hardware as the 2D segmentation, with model weights available on the KIBS GitHub repository. More implementation details on the training procedure, as well as the training metrics are given in Section 6.2 and 6.4 of the Supplementary Material, respectively.

3.4 Plane estimation and meshing

As said, once at least three roof section's corners are inferred, and their height-to-ground estimated, one can easily geometrically derive the 3D plane coefficients of the associated roof section, and hence the height-to-ground of each pixel belonging to this roof section, unto full building and then city-wide 3D reconstruction. Now for a number $N \geqslant 4$ of segmented roof section corners, the algorithm proceeds to select three corners among these forming the largest triangle area via a basic Delaunay triangulation, so as to increase 3D reconstruction accuracy, as shown in Fig. 2.

3.5 Implementation details

We can cover the testing procedure of the KIBS method in five general steps.

Firstly, the whole satellite raster image is split in a grid of 230×230 tile images, with a margin overlap of 10 pixels on each four sides of the image.

Secondly, the aforementioned Mask-RCNN model trained for 2D segmentation is applied to each of these tile images so as to infer the roof sections 2D segmentation.

Thirdly, these segmented pixels are blended within their associated raster tile image as blue pixels, with a value $\{0, 0, 200\}$ if belonging to the training data set, $\{0, 0, 210\}$ if belonging to the validation data set, and $\{0, 0, 220\}$ if belonging to the testing data set.

Fourthly, the aforementioned Mask-RCNN model trained for 3D reconstruction is then applied to each of these blended tile images so as to infer the roof section corners as keypoints, with their own height-to-ground (as a class in meters, according to the LOD2 precision standards, i.e. 1 m, 2 m, 3 m, etc.). After some postprocessing, the output represents these roof section squares as red squares of 15×15 pixels where the red RGB channel is given the value $200 + z$, where $z \in \mathbb{N}^*$ is the height-to-ground of the corner, as shown on Fig. 6 for Mourmelon-le-Grand and 7 for Sissonne.

Fifthly, as already said, for $N \geqslant 3$, one can easily geometrically derive the 3D plane coefficients of the roof section, and hence the height-to-ground of each pixel belonging to this roof section, unto full building and then city-wide 3D reconstruction. The latter can then be visualised in 3D via the Kinetic Shape Reconstruction (KSR) method developed in [60] (see below).

The details of the data postprocessing of the KIBS model are given in the section 6.5 of the Supplementary material. We here simply sum up this procedure via Fig. 3 as a general description.

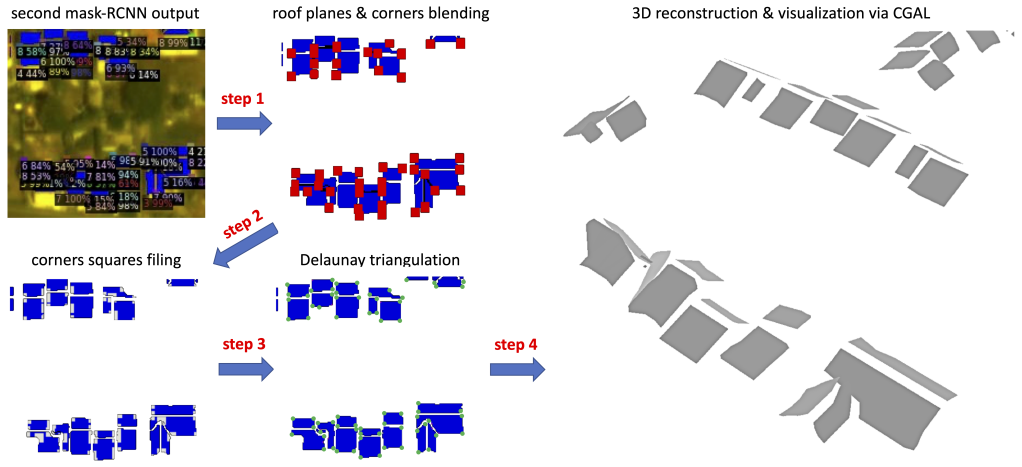

Figure 3: Step by step data postprocessing pipeline of the 3D reconstruction output. Firstly, the code translates the raw output of the 3D reconstruction Mask-RCNN algorithm into a blending of the roof sections from the first algorithm (blue pixels) with their corners (red pixels). Then in a second step, the red 15×15 roof section squares are filed so as to remain on their associated roof sections only, and not on neighbouring ones, nor outside of any plane structure at all. In a third step, the code determines by Delaunay triangulation which are the three roof corner pixels forming the largest possible triangle area for each roof section (when possible), so as to derive its 3D plane coefficients in a more precise manner. Eventually, the code can use the Computational Geometry Algorithms Library [4] (CGAL) in order to visualize the basic roof sections' reconstruction in 3D.

4. Experiments

4.1 Qualitative results

The results of the 2D segmentation of the roof sections for all data sets (training, validation, testing) are shown in Fig. 4 for Mourmelon-le-Grand and Fig. 5 for Sissonne. These figures provide a detailed visual comparison between the original satellite images and the output of the 2D segmentation part of the KIBS model, allowing to qualitatively assess the accuracy and precision of our model in identifying and segmenting the roof sections from the satellite images. One can thus see the model's capability to accurately perform 2D segmentation of urban satellite images, which is a crucial step towards achieving our ultimate goal of 3D urban reconstruction.

Figure 4: 2D segmentation output visualization of Mourmelon-le-Grand (on the testing set only, which consists in shuffled 230×230 tiles from the whole raster data), with the ground truth being displayed in light green and the model output in blue. The model predicts segmented pixels correctly when both overlap, in dark green.

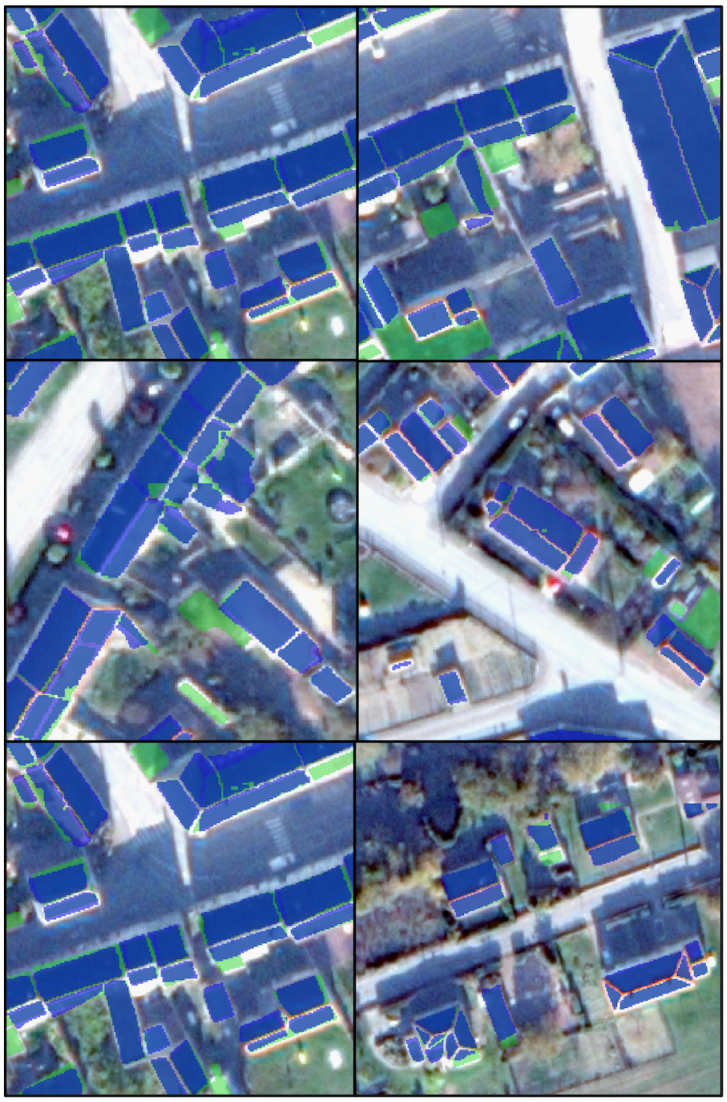

Figure 5: 2D segmentation output visualization of Sissonne (on the testing set only, which consists in shuffled 230×230 tiles from the whole raster data), with the ground truth being displayed in light green and the model output in blue. The model predicts segmented pixels correctly when both overlap, in dark green.

The results of the 3D inference on all data sets (training, validation, testing) are shown in Fig. 6 for Mourmelon-le-Grand and Fig. 7 for Sissonne. The 3D inference results are represented via color-coded roof section corners, each color code being derived from a unique class corresponding to the discrete corner's height-to-ground in meters. This visual representation and panoptic segmentation allows us to qualitatively evaluate the model's ability to infer the 3D structure of the urban landscape from the 2D segmentation output. It is noteworthy that the model exhibits a high level of detail in the 3D inference, successfully capturing the complex architectural features and the varying heights of the buildings in both cities.

The visualization of this 3D inference, scaled to DSM values, is displayed after the KSR reconstruction [60] in Fig. 8 for Mourmelon-le-Grand and 9 for Sissonne. This provides a more tangible and

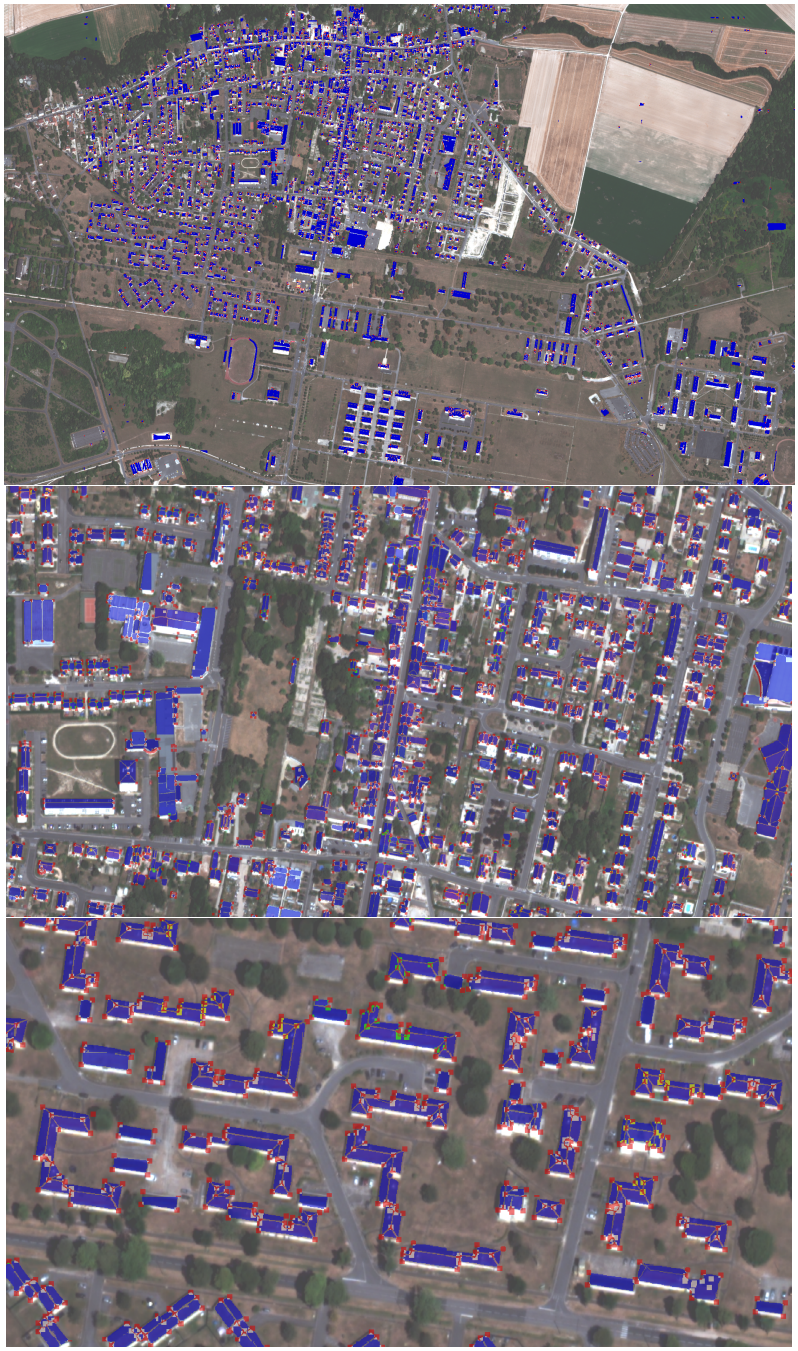

Figure 6: Outputs of the roof section keypoints inference before data postprocessing, overlaying the associated raster image of Mourmelon-le-Grand. Each red square contains the height-to-ground z of its associated roof corner within the red channel of the RGB picture, with a value $200 + z$. For visualization purposes, when these overlap the blue segmented pixels of their associated roof section(s), they take a white, green, or yellow color if they belong to the training, validation, or testing set, respectively.

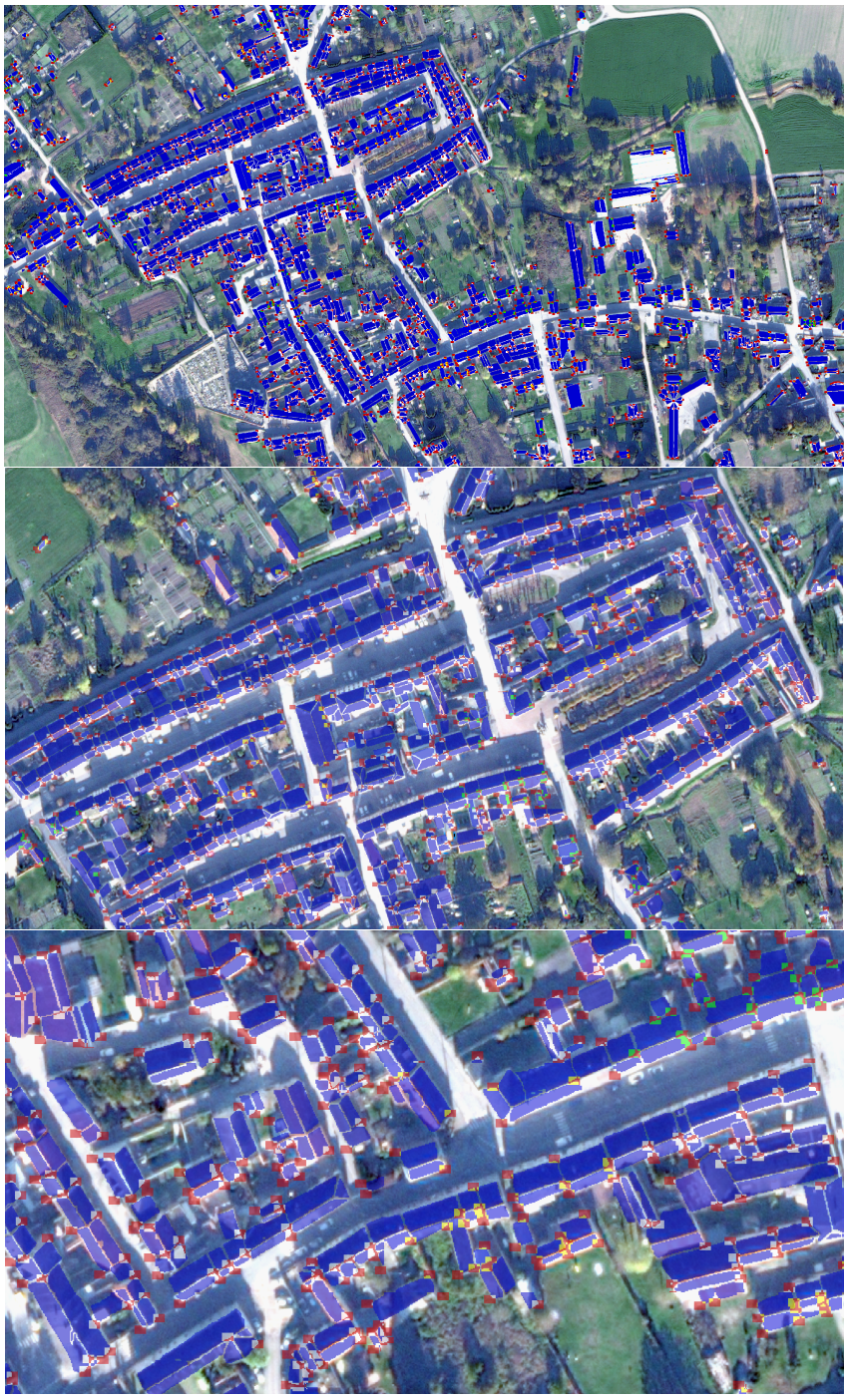

Figure 7: Outputs of the roof section keypoints inference before data postprocessing, overlaying the associated raster image of Sissonne. Each red square contains the height-to-ground z of its associated roof corner within the red channel of the RGB picture, with a value $200 + z$. For visualization purposes, when these overlap the blue segmented pixels of their associated roof section(s), they take a white, green, or yellow color if they belong to the training, validation, or testing set, respectively.

intuitive understanding of the model's output, effectively transforming the aforementioned panoptic segmentation into a 3D model of the urban landscape, not only for the roof structures but for the whole buildings underneath.

4.2 Quantitative results

The results of the KIBS model are shown in Tab. 1 for Mourmelon-le-Grand and Tab. 2 for Sissonne. The results of the 2D segmentation can be summed up through the Jaccard index, also called Intersection over Union (IoU), which is the percentage of the M accurately segmented pixels on the 2D map, with respect to the ground truth pixels. We obtain an IoU of 88.55% for the testing set.

The accuracy of the 3D inference can be summed up *for these pixels that were correctly 2D segmented wrt. ground truth*, through their heights mean accuracy, and mean square error. The heights mean accuracy is the average of the absolute differences between the heights of each correctly segmented pixels $\hat{z}$, and the height of its associated ground truth pixels z, expressed as a percentage of the latter: $\Sigma_{i=0}^{M} 100|\hat{z}_i - z|/zM$. We find a heights' mean accuracy for the testing set of 74.85% for Mourmelon-le-Grand, and 72.57% for Sissonne. And we find a heights' mean value for the testing set of 1.60 m for Mourmelon-le-Grand, and 2.06 m for Sissonne.

And one can study the 3D reconstruction efficiency via the mean square error, knowing our data set has an average roof height of 6.36 m for Mourmelon-le-Grand, and 7.53 m for Sissonne. The heights' mean square error is given by the average squared difference between the heights of each correctly segmented pixels $\hat{z}$, and the height of its associated ground truth pixels z: $\Sigma_{i=0}^{M}(\hat{z}_i - z)^2/M$. We thus find a heights' mean square error for the testing set of 2.35 m^2 for Mourmelon, and 7.41 m^2 for Sissonne.

We can see from these two latter statistics, that the aim of urban 3D reconstruction at LOD2 is reached.

Statistic (Mourmelon)	**Training**	**Validation**	**Testing**
Jaccard index (IoU)	88.56 %	86.96 %	88.55 %
Heights' mean accuracy	76.87 %	74.03 %	74.85 %
Heights' mean difference	1.47 m	1.65 m	1.60 m
Heights' mean square error	1.99 m^2	2.51 m^2	2.35 m^2

Table 1: Results of the KIBS model for Mourmelon-le-Grand. The results of the 2D segmentation shown by the Jaccard index (i.e. Intersection over Union, or IoU), which is the percentage of the M correctly segmented pixels compared to ground truth. For $\hat{z}$ and z the heights-to-ground of these correctly segmented pixels and of ground truth pixels respectively, the results of the 3D reconstruction is shown by the heights' mean accuracy $\Sigma_{i=0}^{M} 100|\hat{z}_i - z|/zM$ and mean square error $\Sigma_{i=0}^{M}(\hat{z}_i - z)^2/M$.

4.3 Performance

Let $s \in \mathbb{N}^*$ be the number of pixels giving the (squared) raster tile images' size (e.g. for $s = 230$, the raster tile images are of size $s \times s = 230 \times 230$), $p \in \mathbb{N}^*$ the number of pixels of the raster tile images' margin overlap, and $q \in \mathbb{N}^*$ the number of pixels of the size of the segmented roof section corners (e.g. for $q = 15$, the roof section corners were segmented as red squares of sizes $q \times q = 15 \times 15$). The KIBS model performance has been explored through several combinations of the model hyperparameters on the validation set, by visualizing the output results for combinations of these hyperparameters s, p, q.

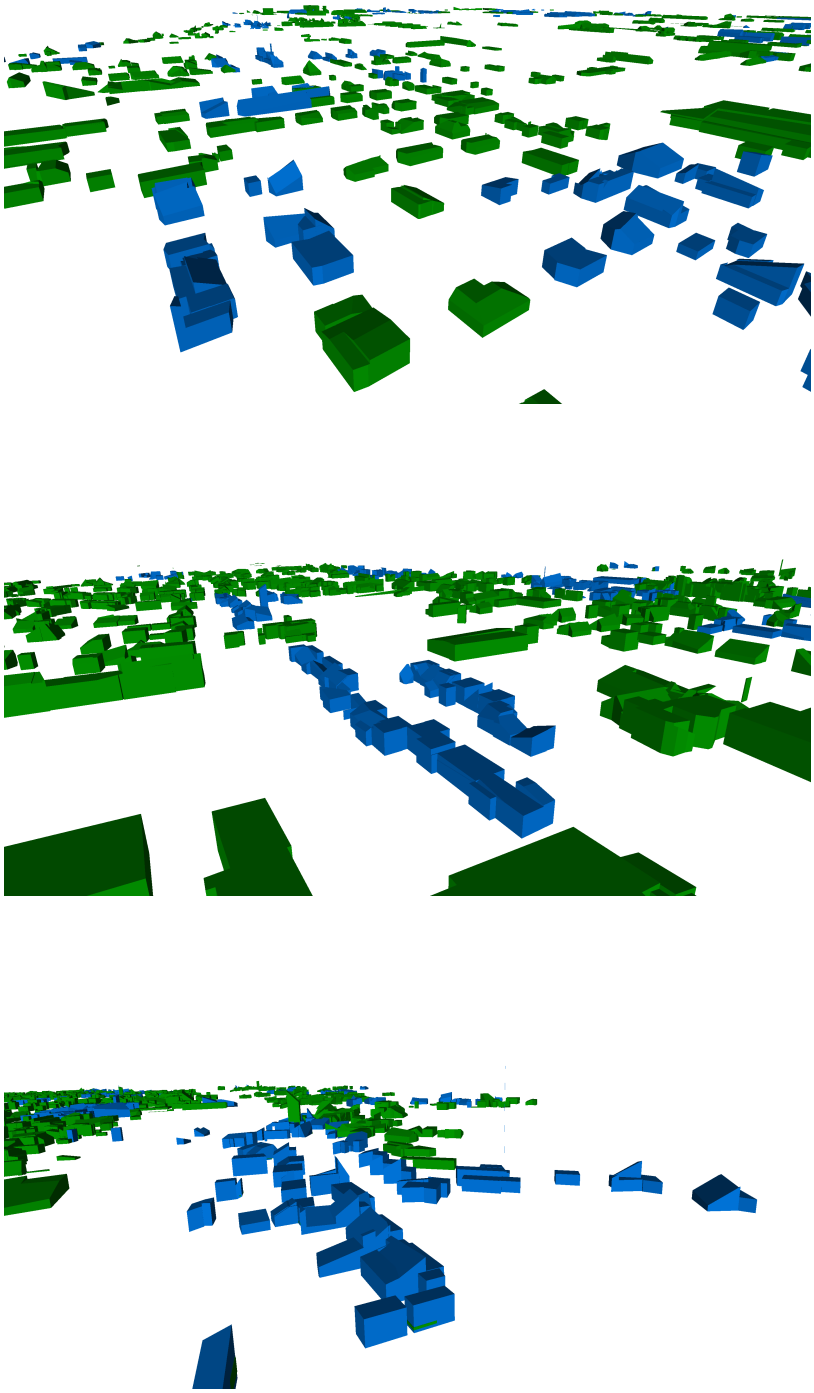

Figure 8: Visualization of the 3D reconstruction output of the testing set of Mourmelon-le-Grand with corrected DTM values. The training and validation data is displayed in green, and the KIBS model output in blue.

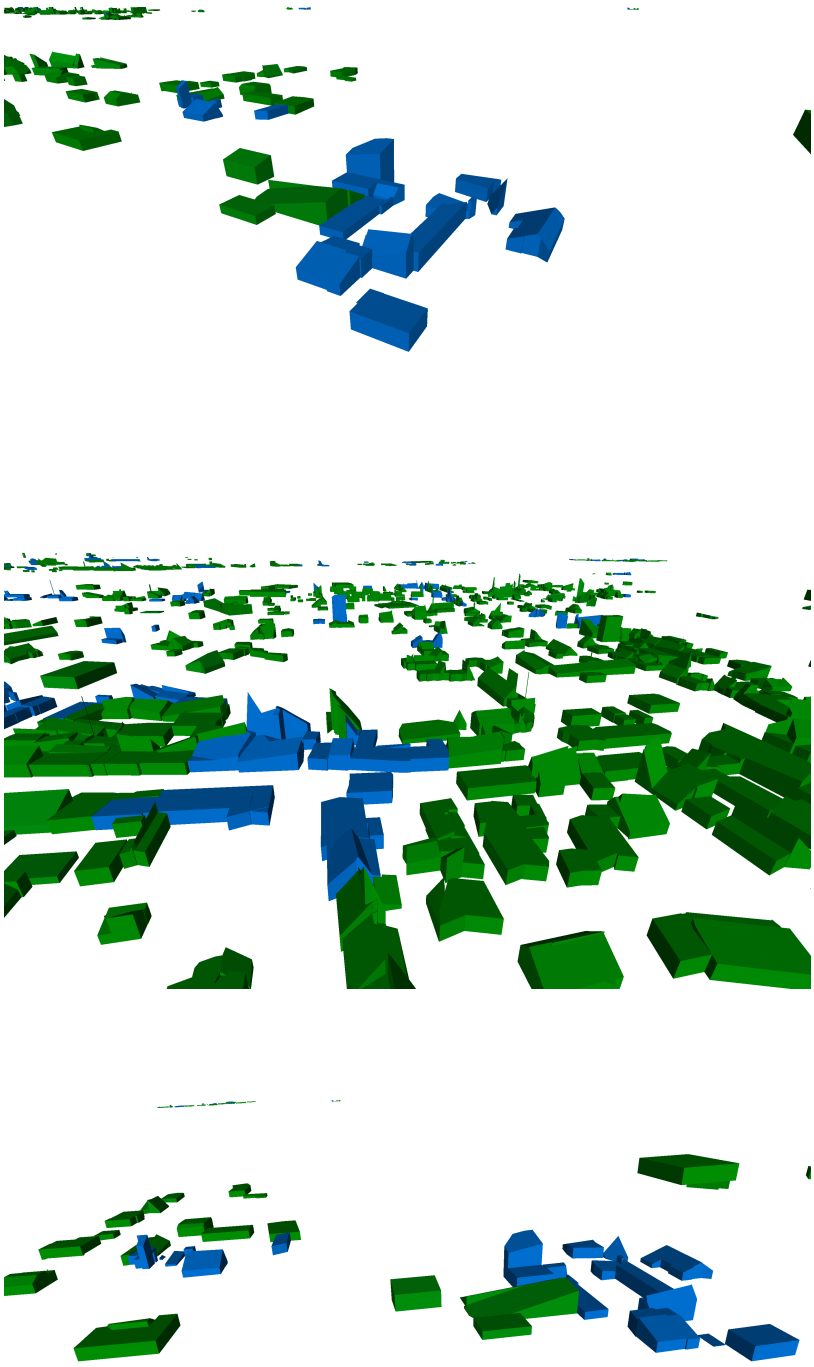

Figure 9: Visualization of the 3D reconstruction output of the testing set of Sissonne with corrected DTM values. The training and validation data is displayed in green, and the KIBS model output in blue.

Statistic (Sissonne)	**Training**	**Validation**	**Testing**
Jaccard index (IoU)	88.67 %	73.49 %	75.21 %
Heights' mean accuracy	74.28 %	71.08 %	72.57 %
Heights' mean difference	1.94 m	2.18 m	2.06 m
Heights' mean square error	7.21 m^2	8.29 m^2	7.41 m^2

Table 2: Results of the KIBS model for Sissonne. The results of the 2D segmentation shown by the Jaccard index (i.e. Intersection over Union, or IoU), which is the percentage of the M correctly segmented pixels compared to ground truth. For $\hat{z}$ and z the heights-to-ground of these correctly segmented pixels and of ground truth pixels respectively, the results of the 3D reconstruction is shown by the heights' mean accuracy $\Sigma_{i=0}^{M} 100|\hat{z}_i - z|/zM$ and mean square error $\Sigma_{i=0}^{M}(\hat{z}_i - z)^2/M$.

The change in performance for different raster tile images' sizes s was explored, with values $s = \{150, 230, 300, 768\}$. We found the use of larger resolution raster images as input to be a limiting factor to the number of roof section corners that could be detected by the second Mask-RCNN model (as shown in the Supplementary Material section 6.5 by comparison with Fig. 19, where $s = 768$, or on Fig. 20, where $s = 300$). We thus found a better performance for smaller resolutions, especially at $s = 230$ (calculated for values $p = 10$ and $q = 15$ only).

Secondly, another explored hyperparameter was the raster tile images' margin overlap p, with values $p = \{10, 150\}$ pixels (calculated for values $s = 230$ and $q = 15$ only). We found a large margin overlap value to cause intractable memory issues during the run time, and hence selected $p = 10$.

Thirdly, the size of the segmented roof section corners q was explored, with values $q = \{10, 15\}$ (for values $s = 230$ and $p = 10$ only). As said, this hyperparameter has a great impact on the overall KIBS model performance, since too large squares may assign the height of a given corner to several others as well, and too small squares may produce false negatives by not overlapping their associated segmented roof sections at postprocessing, We thus found better performance for $q = 15$ (calculated for values $s = 230$ and $p = 10$ only).

4.4 Limitations

As said, the core premise behind the KIBS model hypothesis is that the oblique perspective of the satellite raster image supplies the deep learning algorithms with valuable and complex information related to the roofs corners height-to-ground. This includes aspects such as the tilt of the buildings' walls, the shadows cast by the buildings, the perspective of the roof peak or ridge, and so on. These elements collectively enable the algorithms to deduce the height-to-ground of the corners of the roof sections with a level of accuracy that meets the standards of the LOD2 requirement. This is an important feature of the KIBS prior pertaining to its generalization, because each data set used to train the 3D reconstruction part of the model has its own specific buildings inclinations (related to the raster' satellite viewing angle α), and its own specific shading of the buildings (related to the raster' solar zenith angle θ), as aforementioned for a our satellite data sets. The KIBS method trained on a data set with such angles α and θ, should hence only generalize to new raster sets taken with angle parameters lying in the neighborhoods of those of the training set, so that the variations in the model inference of the buildings' height-to-ground are negligible within the requirements of a LOD2 precision range.

4.5 Baseline comparison

Due to the uniqueness of the results of this study, the KIBS method faces a challenge in finding relevant methods for a useful baseline comparison. Other interesting research works like [29] (which does 3D urban reconstruction at LOD1), and [47, 38] (which both use roof primes for the urban reconstruction) rely on third parties code which is not accessible. But a rigorous approach can be to use the 2D segmentation step of our KIBS approach, and then assign the segmented pixels' height-to-ground via another DSM, courtesy of LuxCarta, which is of LOD1. The resulting point cloud can then be approximated as roof sections unto 3D reconstruction, as shown on Fig. 10 below, with a rather poor precision.

4.6 Ablation study

We have used different ablations of the model and studied its change in performance.

Firstly, if one tries to infer the roof section corners' position and heights (x, y, z) directly through the raster satellite image (cf. images on the left of Fig. 20 of section 6.5 in the Supplementary Material), the result output shows a poor performance.

Secondly, when the blending of the 2D segmentation output is performed on either the red or green channels of the RGB raster image, the results and model accuracy do not change much with our current approach (cf. images in the center of Fig. 20 of section 6.5 in the Supplementary Material).

Thirdly, if one tries to improve the 2D location (x, y) of the roof section corners by an image input consisting in the direct raster satellite image, or in the binary masks of the roof sections (cf. images on the right of Fig. 20 of section 6.5 in the Supplementary Material), the output results are very unsatisfying.

Fourthly, if one tries and infers the roof section corners' heights out of a blending of three stereo satellite pictures of the same geographical area, each taken with different satellite viewing angle and solar zenith angle, the output results show the poor performance of this approach (cf. Fig. 21 of section 6.5 in the Supplementary Material).

Fifthly, other very different models and algorithms than the Mask-RCNN solution were tried and used in this research, both for the 2D segmentation part and the 3D reconstruction part with the work of [61, 28], but as can be shown on Fig. 22 of section 6.5 in the Supplementary Material, this approach failed completely.

Sixthly, likewise, a RegNet architecture [56] of the Detectron2 model zoo (`regnety_4gf_dds_FPN`) has been used instead of the Mask-RCNN blocks, but with poor results in our time-constrained hyperparameter space optimization procedure so far.

5. Conclusion

We have thus presented a new method named KIBS for the urban 3D reconstruction of satellite images at a LOD2, with two central features: an end-to-end deep learning approach, and a model input based on a one-shot satellite raster image. The backbone of this deep learning model is a two-step method relying firstly on a Mask-RCNN algorithm performing the 2D segmentation of the individual roof sections, and secondly on another Mask-RCNN algorithm of exact same architecture using the latter output blended into the raster image in order to infer the roof section corners and their heights. The performance of this KIBS approach is displayed by a Jaccard index for the 2D segmentation of the roof sections of 88.55% (Mourmelon-le-Grand) and 75.21% (Sissonne), and a

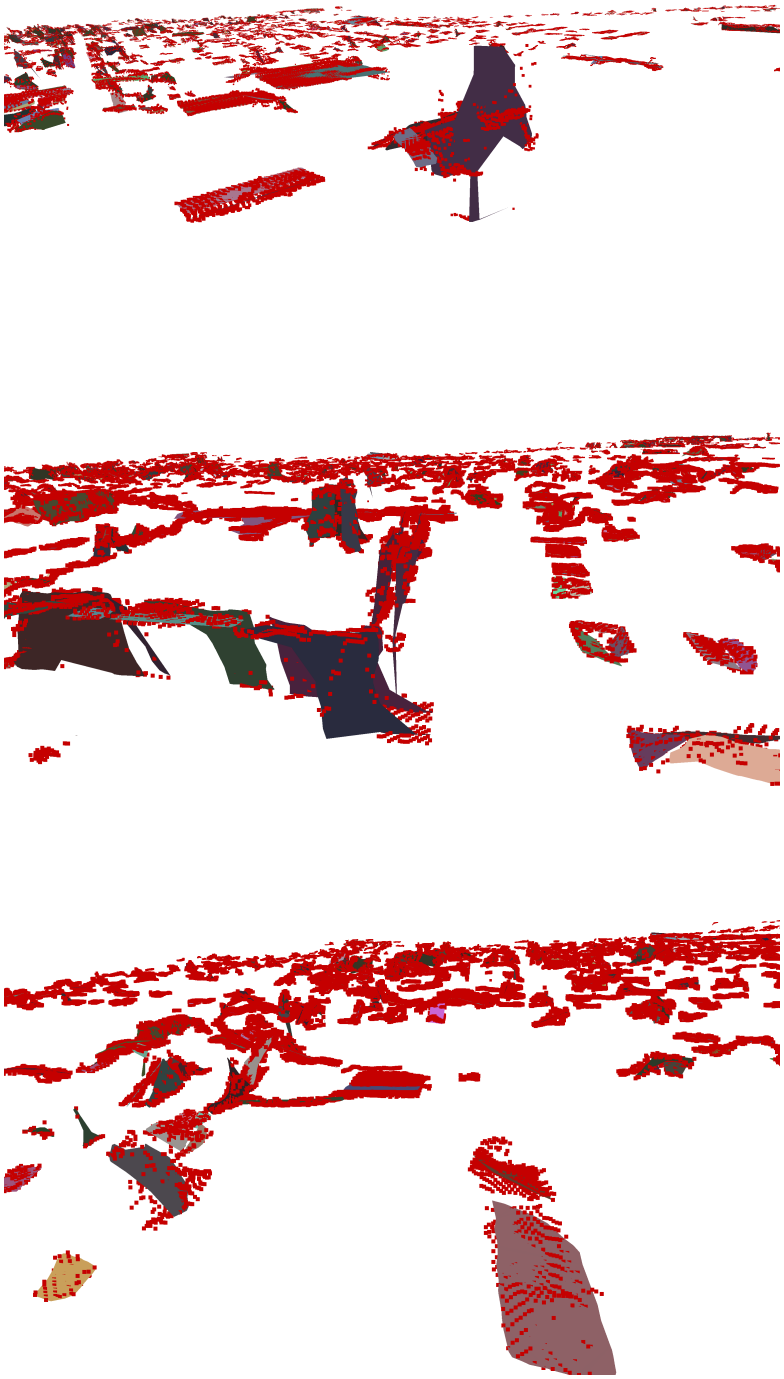

Figure 10: Comparison baseline based on the 2D segmentation of our KIBS method, followed by the interpolation of the elevation point cloud stemming from our DSM data (red points), courtesy of LuxCarta, with ensuing roof section 3D reconstruction (coloured planes).

heights' mean value for the roof section pixels correctly inferred by the 2D segmentation method of 1.60 m (Mourmelon-le-Grand) and 2.06 m (Sissonne). The KIBS method can thus perform 3D reconstruction of urban satellite raster images within the requirements of the LOD2 precision range.

As such, the authors posit that the weight played by deep learning methods in satellite and aerial data ground reconstruction, whether via end-to-end approaches or in complement of more procedural approaches, will only increase in coming years.

Bearing in mind the time-constrained optimization procedure of the method presented in this research work, the authors also posit that the performance results of the KIBS method may be easily enhanced at a little cost, notably by a further exploration of the hyperparameter space, and by use of deep learning architectures other than the Mask-RCNN neural networks here employed.

Especially, a direct natural extension of the KIBS approach should study whether one single neural network comprising this two-step approach (2D segmentation followed by 3D reconstruction) into one backbone architecture could be designed. The authors posit this general monocular or single-shot approach to 3D inference could find many promising applications reaching far beyond the satellite and aerial imagery segments of computer vision, and pertain to all 3D inference methods of machine learning in the largest sense, with other potential applications in autonomous driving, drones engineering, environmental monitoring, and virtual reality.

This said, our work also raises new questions, such as how to further improve the accuracy of 3D inference, how to handle taller structures, and how to apply our methods to other types of data. A crucial future prospect of the KIBS method pertains to its generalization, not only for very different data sets (e.g. dense city centers with tall buildings), but also wrt. to raster data sets of different satellite viewing angle α and solar zenith angle ω from those of our training set. Thanks to its short computational training and inference times, a suite of several KIBS algorithms could be trained on sets of data taken with different combinations of these two angle values' neighborhoods, so as to reach a practical generalisation threshold by modular learning, corresponding to the offers of satellite data vendors.

6. Supplementary material

6.1 Implementation details of the 2D segmentation part

The training data preprocessing for the Mask-RCNN model performing the 2D segmentation of the roof lines first relies on first slicing the overall 8687×9890 satellite raster image into individual tiles of 230×230 individual raster images. These are cut to overlap each other on all four sides by a margin of 10 pixels, to improve the future reconstruction at inference level.

Then, the ground truth shapefile of the polygons delimiting each roof section contours is extracted for each associated 230×230 tile raster image. For each such tile raster image, a set of 230×230 black and white images is generated for each roof section, where each white pixel belongs to one unique roof section per image, and all other pixels are set as black. Each roof section mask is given one same dummy class label at this stage.

All these generated black and white images associated with each tile raster image are given a unique file name that allows a specific `pycococreator` algorithm [55] to generate a .json file for these ground truth masks in the PYCOCO format [33].

The set of all such tile raster images, together with their associated ground truth images is then shuffled randomly according to a uniform distribution in order to build three disjoints sets: one for training (60% of the whole data set), one for validation (20% of the whole data set), and one for testing (20% of the whole data set).

Via `pycococreator`, a .json ground truth file associated with the training set is hence generated, and likewise for the validation and testing sets.

All the tile raster images and these generated .json ground truth files are then given as input to train a Mask-RCNN artificial neural network [20] from the Detectron2 suite [12] named `mask_rcnn_R_50_FPN_3x`. This network consists in a backbone combination of a ResNet-50 model [19] stacked with a Feature Pyramid Network [34] (FPN), comprising standard convolution and fully-connected heads for mask and box prediction, respectively. It is pretrained with a 3x schedule, corresponding to about 37 COCO epochs.

The results, presented in Section 4, are based on a six days training, on a Dell T630 GPU node of dual-Xeon E5-26xx with four GeForce GTX 1080 Ti GPUs cards, 3584 CUDA cores per card, and 11 GB of RAM capacity per card.

The training metrics are shown in Fig. 11-18 of Section 4.3 below, and were monitored online via TensorBoard [1] in order to limit regularization issues. The weights of the Mask-RCNN network trained for this 2D segmentation are available on the KIBS GitHub repository [37].

6.2 Implementation details of the 3D reconstruction part

From a deep learning perspective, the 3D reconstruction training relies on a Mask-RCNN model of exact same architecture as for the 2D segmentation, but designed for panoptic segmentation, i.e. both pixel segmentation and class inference. In our case, the pixel segmentation here consists in the model drawing a 15×15 pixels square over each roof section corner of a raster image blended with the output of the 2D segmentation, and the class inference consists in giving each such corner a unique class label allowing to retrieve the corner's height-to-ground in meters.

As said, after training, the output of the latter 2D segmentation is blended within the original associated RGB raster image, such that each segmented pixel (identifying a roof section) is given a value $\{0, 0, 200\}$ if belonging to an image from the training data set, $\{0, 0, 210\}$ if belonging to an

image from the validation data set, and $\{0, 0, 220\}$ if belonging to an image from the testing data set. Roof sections on the raster images hence appear in blue color. Ablation studies (see below) show this method allows the 3D reconstruction algorithm to identify much more efficiently the roof sections' corners, than if the ground truth was associated with the original raster images only.

In our code (for particular reasons related to the Detectron2 framework), class labels are: `hah` for a height of 1 m, `hbh` for 2 m, `hch` for 3 m, ..., `hsh` for 19 m. This range of 19 possible different classes is due to the maximum corner's height in our particular data set not exceeding 19 m above ground, but one can extend the number of these classes/heights much more in the Detectron2 framework to cope with the potential taller building structures of other data sets.

Hence, if our data set contained skyscrapers or buildings of greater heights, the KIBS method and training could remain similar by simply increasing the number of possible classes, and/or raising the height granularity above 1 m, and/or using non-linear graduations in the heights increments, etc.

This said, the training, validation, and testing sets generation is done in a similar way as for the 2D segmentation: each aforementioned 230×230 blended raster image with a margin overlap of 10 pixels on all four sides, is associated with a set of ground truth images, each of them representing on a black background a square of 15×15 white pixels, in order to represent a roof corner on this image. Its class name (i.e. corner's height) is given in the image file name.

This is likewise proper and fed to the pycococreator framework, in order to produce .json annotation files for this ground truth data in a PYCOCO format that is understandable to the Detectron2 framework. A same Mask-RCNN model (`mask_rcnn_R_50_FPN_3x`) as before is thus trained on this training data set, so as to identify roof corners and their classes (i.e. heights). The results, presented in Section 4, are based on a six days training, also on the same hardware as before (four GeForce GTX 1080 Ti GPUs cards).

The training metrics are shown in Fig. 11-18 of Section 4.5 below, and can be monitored online in order to limit regularization issues. The learning weights of the Mask-RCNN model for this 3D reconstruction are available on the KIBS GitHub repository [37].

6.3 Training and validation metrics

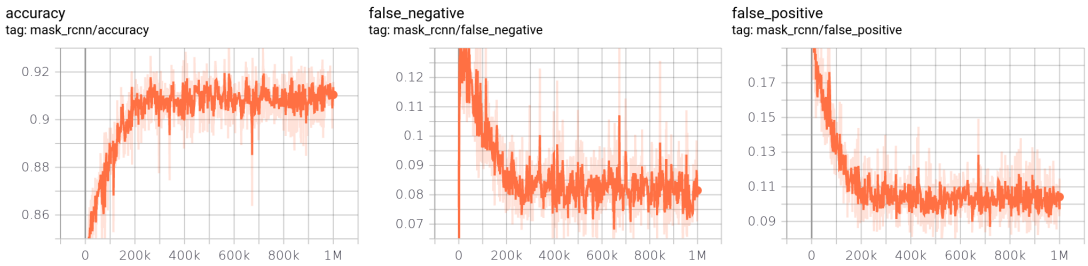

Figure 11: Training of the 2D segmentation with Mask-RCNN for Mourmelon-le-Grand, showing the model's precision (left), error rates for false negatives (center) and false positives (right), as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

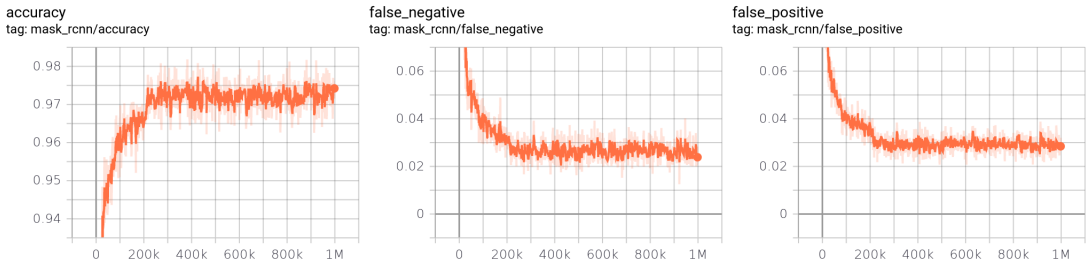

Figure 12: Training of the 2D segmentation with Mask-RCNN for Sissonne, showing the model's precision (left), error rates for false negatives (center) and false positives (right), as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

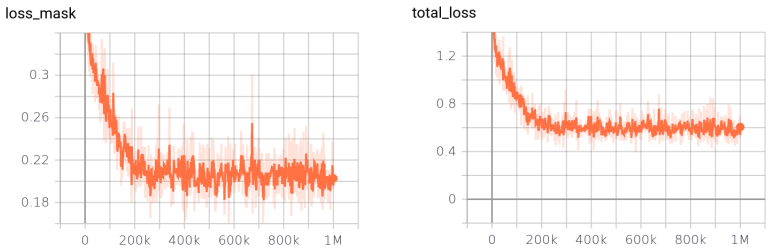

Figure 13: Training of the 2D segmentation with Mask-RCNN for Mourmelon-le-Grand, showing the model's mask loss (left) and total loss (right), as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

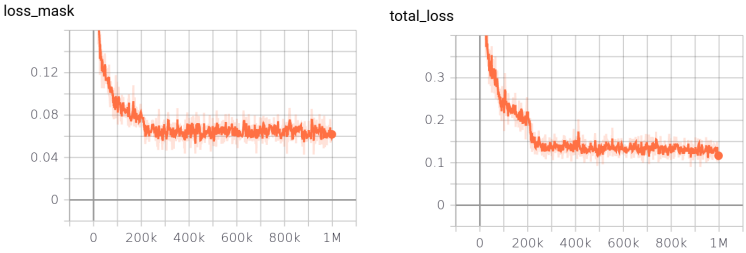

Figure 14: Training of the 2D segmentation with Mask-RCNN for Sissonne, showing the model's mask loss (left) and total loss (right), as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

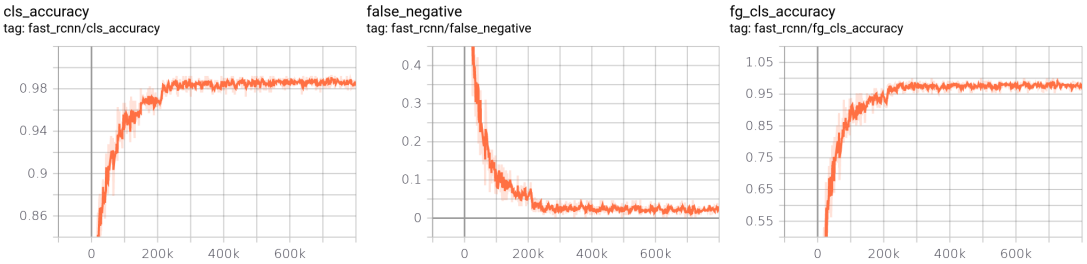

Figure 15: Training of the 3D reconstruction with Mask-RCNN for Mourmelon-le-Grand, showing the model's class accuracy (left), error rates for false negatives (center) and class accuracy for foreground proposals, as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

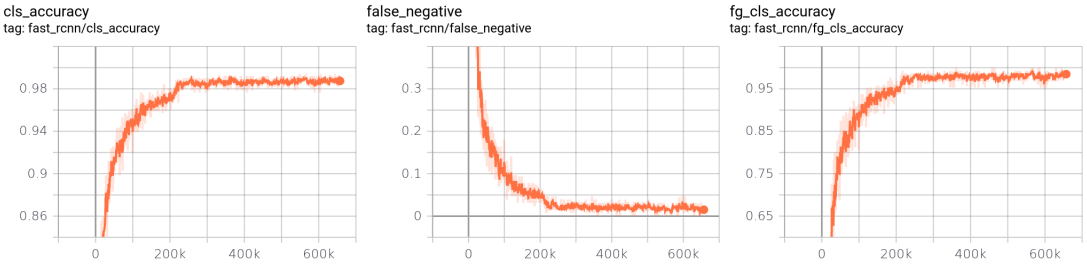

Figure 16: Training of the 3D reconstruction with Mask-RCNN for Sissonne, showing the model's class accuracy (left), error rates for false negatives (center) and class accuracy for foreground proposals, as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

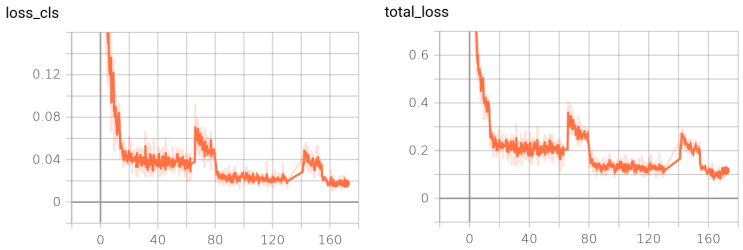

Figure 17: Training of the 3D reconstruction with Mask-RCNN for Mourmelon-le-Grand, showing the model's class loss (left) and total loss (right), as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

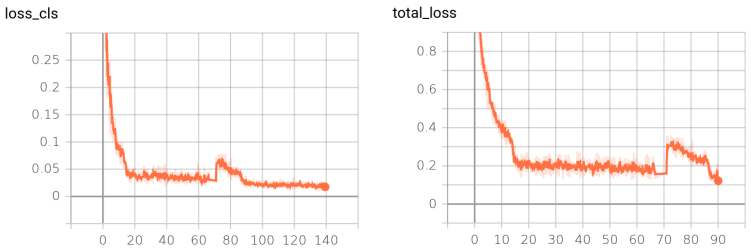

Figure 18: Training of the 3D reconstruction with Mask-RCNN for Sissonne, showing the model's class loss (left) and total loss (right), as a function of the number of epochs (x-axis). These plots were monitored online via TensorBoard [1] in order to limit regularization issues.

6.4 Ablation studies

Figure 19: Example of roof section corners inference directly on a raster image of size 768×768, with ground truth on the left, and result output on the right. The large number of roof sections in the ground truth overloads the Mask-RCNN algorithm.

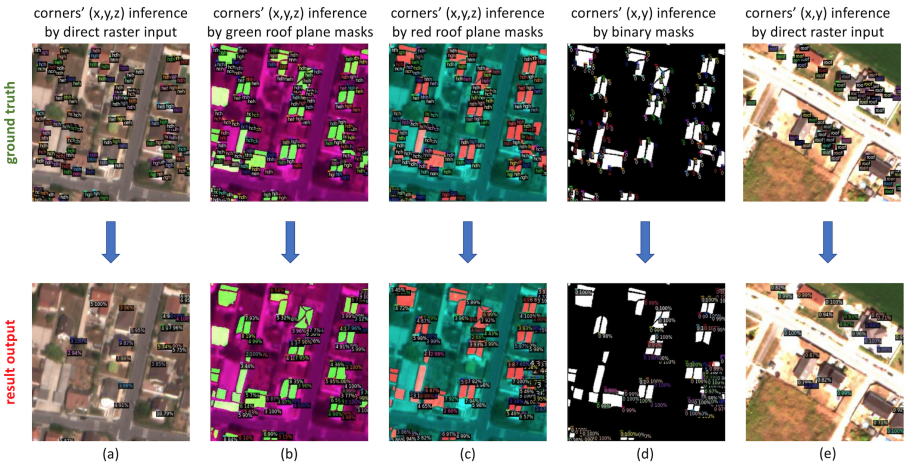

Figure 20: Sum up of the ablation studies and other research approaches. If one tries to infer the roof section corners' position and heights (x, y, z) directly through the raster satellite image (a), the result output shows a poor performance. In fact, even the simple inference of the roof section corners' positions (x, y) is unsatisfying, whether the image input consists in the direct raster satellite image or the binary masks of the roof sections (d and e). Trying to perform the KIBS method through other RGB channels, e.g. green or red (b and c), doesn't change the performance of the algorithm noticeably.

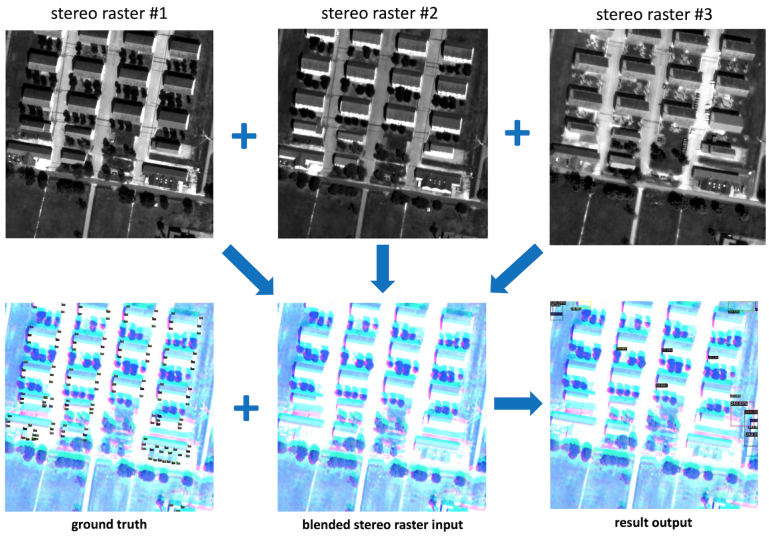

Figure 21: Ablation study based on inferring the roof section corners' height out of a blending of three stereo raster pictures of the same geographical area, taken with different satellite viewing angles and solar zenith angles. As one can see, the output results show the poor performance of this approach.

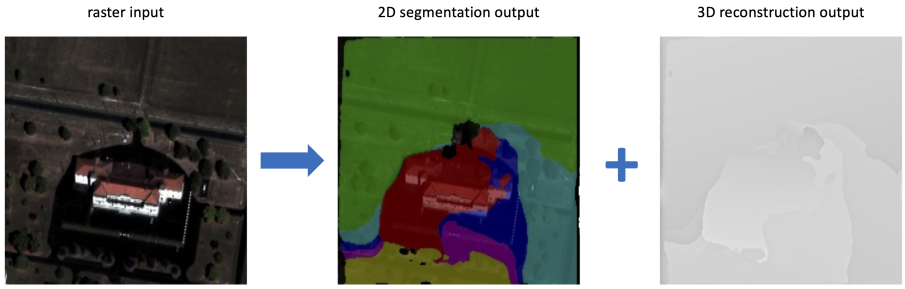

Figure 22: Ablation study based on the work of [61, 28] instead of the Mask-RCNN algorithms used in the KIBS method, where a 2D segmentation is applied to the satellite raster input, and a 3D map is reconstructed out of it.

6.5 Data postprocessing

There are six general comments one can make wrt. the data postprocessing of the KIBS method. We also refer the reader to Fig. 2 for a general recap of this procedure.

Firstly, one needs to beware the procedure sometimes fails to correctly infer one or several roof section corners. For a number $N \geqslant 4$ of segmented roof section corners, the algorithm proceeds to select three corners among these forming the largest triangle area by a Delaunay triangulation, so as to increase 3D reconstruction accuracy. For $N = 1$ or $N = 2$, the model considers for simplification purposes the roof section to be parallel to the ground, and at a height equal to that of this corner, or the average of these two corners, respectively. Finally, if $N = 0$ and no corner is detected by the algorithm, the roof section is also considered to be parallel to the ground, but assigned a height equal to the mean of all roof corners heights of the training set (which in the case of our data set amounts to 6.11 m).

Secondly, another basic postprocessing consists in "filing" the heights of the corners used for the roof section 3D reconstruction, based on the assumption that virtually no real roof section contains three corners of different heights. Let's assume these three corner heights z_1, z_2, z_3, in ascending order, are all unequal to each other: then, if $z_2 < (z_1 + z_3)/2$, the algorithm hence sets z_1 and z_2 to the value of their average $(z_1 + z_2)/2$; and if $z_2 \geqslant (z_1 + z_3)/2$, the algorithm sets z_2 and z_3 to the value of their average $(z_2 + z_3)/2$.

Thirdly, the output of the this Mask-RCNN model for 3D reconstruction gives after some postprocessing (and some changes to the native Detectron2 code [37]), 15×15 pixels red squares representing the roof section corners, and the class of each of these corners (i.e. their height-to-ground in meters) is embedded in RGB format by assigning these pixels a value $200 + z$ in the red channel, where $z \in \mathbb{N}^*$ is their height in meters (as shown in Fig. 6 for Mourmelon-le-Grand and Fig. 7 for Sissonne). This 3D reconstruction output of red squares over a black background is then blended over its associated 2D segmentation output (i.e. the blue roof sections over a black background). This blending process must be done cautiously for several reasons, and the KIBS code contains several postprocessing methods to ensure no data is lost or mismatched at this stage. The reasons are the following: i- in certain complex roof structures, some of these 15×15 corner squares can overlap other roof section segmentation pixels they don't belong to, and hence assign a wrong height to them; ii- some of these corner squares can sometimes be placed at inference sufficiently far away from the segmented roof section, so that no match is made and the whole roof section is ill-reconstructed; iii- the Delaunay triangulation will have to chose for each 15×15 pixels square only one pixel overlapping the segmented roof section rim, and hence a dedicated method must find and select this pixel among many others overlapping the plane. This data postprocessing pipeline of the 3D reconstruction output is shown in Fig. 3.

Fourthly, the segmented roof sections need to be perfectly distinguished (i.e. pixel-separated) from each other at the 3D reconstruction stage, since each has its own 3D plane coefficients inferred by the model.

Fifthly, when these individual blended tile raster images are put back together to form the large 8687×9890 original image corresponding to the full satellite view, some roof sections and corners may be found to intersect two or more of these former tile images. Hence, some parts of a given reconstructed building may spread over several former tile images, and thus belong to different data sets (training, validation, or testing), with different associated blue pixel values. A `flood_fill()` function from the scikit-image collection for image processing with right `tolerance` parameter can correct this by assigning one single value via the blue channel of each spread roof section ($200, 210$, or 220 for training, testing, or validation, respectively). This flood-fill is done on a first come, first served basis, with no particular priority from the training, validation, or testing set queues.

Sixthly, the data postprocessing methods ultimately writes a text file containing for each line, each roof section points' 3D coordinates $\{x, y, z\}$ and ID (0 for training set origin, 1 for validation set origin, 2 for testing set origin). This is fed to the KSR method reconstruction [60] of the whole city in 3D for visualisation.

Acknowledgement

We graciously thank LuxCarta for providing the satellite raster data with its hand-annotated ground truth.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[2] S. Agarwala, L. Jin, C. Rockwell, and D. Fouhey. Planeformers: From sparse view planes to 3d reconstruction. In *ECCV*, 2022.

[3] J.-P. Bauchet and F. Lafarge. City reconstruction from airborne lidar: A computational geometry approach. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019.

[4] CGAL. Computational geometry algorithms library, 2022. URL www.cgal.org. Accessed: 2022-11-18.

[5] B. Chatterjee. Urban feature classification from remote sensor imagery using deep neural networks. Master's thesis, Concordia University, November 2019. URL https://spectrum.library.concordia.ca/id/eprint/986139/. Unpublished.

[6] W. Chen, J. Gao, H. Ling, E. J. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. Learning to predict 3d objects with an interpolation- based differentiable renderer. In *NeurIPS*, 2019.

[7] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *Computer Vision – ECCV 2016*, 2016.

[8] D. Y. D. Kudinov, C. Lechot and H. Frank. 3d buildings from imagery with ai. part 1: From elevation rasters, 2021. URL https://medium.com/geoai/3d-buildings-from-imagery-with-ai-fbbc1852e4dd. Accessed: 2022-11-16.

[9] L. Duan and F. Lafarge. Towards large-scale city reconstruction from satellites. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[10] G. C. Emmanuel Maggiori, Yuliya Tarabalka and P. Alliez. Convolutional neural networks for large-scale remote-sensing image classification. In *IEEE Transactions on geoscience and remote sensing*, volume 55(2), pages 645–657, 2016.

[11] G. C. Emmanuel Maggiori, Yuliya Tarabalka and P. Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229, 2017.

[12] FAIR. Detectron2 code repository, 2022. URL https://github.com/facebookresearch/detectron2. Accessed: 2022-11-18.

[13] H. Fang, F. Lafarge, and M. Desbrun. Planar Shape Detection at Structural Scales. In *CVPR*, 2018.

[14] K. Fu, J. Peng, Q. He, and H. Zhang. Single image 3d object reconstruction based on deep learning: A review. *Multimedia Tools and Applications*, 80:463–498, 2021.

[15] M. B. Gaetan Bahl and F. Lafarge. Single-shot end-to-end road graph extraction. *CVPR 2022 : IEEE Conference on Computer Vision and Pattern Recognition EarthVision Workshop*, 2022.

[16] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, and et. al. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geoscience and Remote Sensing Magazine*, 6(3):10–43, 2018. doi: 10.1109/MGRS.2018.2854840.

[17] H. Goldberg, M. Brown, and S. Wang. A benchmark for building footprint classification using orthorectified rgb imagery and digital surface models from commercial satellites. In *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. doi: 10.1109/AIPR.2017.8457973. URL https://doi.ieeecomputersociety.org/10.1109/AIPR.2017.8457973.

[18] S. Gui and R. Qin. Automated lod-2 model reconstruction from very-high-resolution satellite-derived digital surface model and orthophoto. *ISPRS Journal of Photogrammetry and Remote Sensing*, 181, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.90. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

[21] B. Hosseiny, A. M. Abdi, and S. Jamali. Urban land use and land cover classification with interpretable machine learning: A case study using sentinel-2 and auxiliary data. *Remote Sensing Applications: Society and Environment*, 28:100843, 2022. ISSN 2352-9385. doi: https://doi.org/10.1016/j.rsase.2022.100843. URL https://www.sciencedirect.com/science/article/pii/S2352938522001513.

[22] J. Huang, Y. Zhang, and M. Sun. Primitivenet: Primitive instance segmentation with local primitive embedding under adversarial metric. In *ICCV*, 2021.

[23] D. Huh. Curvature-corrected learning dynamics in deep neural networks. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4552–4560. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/huh20a.html.

[24] S. Ji, S. Wei, and M. Lu. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1):574–586, 2019. doi: 10.1109/TGRS.2018.2858817.

[25] Y. Jia, Y. Ge, Y. Chen, S. Li, G. B. Heuvelink, and F. Ling. Super-resolution land cover mapping based on the convolutional neural network. *Remote Sensing*, 11(15), 2019. ISSN 2072-4292. doi: 10.3390/rs11151815. URL https://www.mdpi.com/2072-4292/11/15/1815.

[26] F. Lafarge and G. Bahl. Scanner Neural Network for On-board Segmentation of Satellite Images. In *IGARSS 2022 – IEEE International Geoscience and Remote Sensing Symposium*, Kuala Lumpur, Malaysia, July 2022. URL https://hal.inria.fr/hal-03664644.

[27] T. Lejemble, C. Mura, L. Barthe, and N. Mellado. Persistence analysis of multi-scale planar structure graph in point clouds. *Computer Graphics Forum*, 39(2), 2020.

[28] J. E. Lenssen, C. Osendorfer, and J. Masci. Deep iterative surface normal estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11244–11253, 2020.

[29] M. J. Leotta, C. Long, B. Jacquet, M. Zins, D. Lipsa, J. Shan, B. Xu, Z. Li, X. Zhang, S.-F. Chang, M. Purri, J. Xue, and K. Dana. Urban semantic 3d reconstruction from multiview satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[30] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *CVPR*, 2019.

[31] Z. Li, J. Dirk Wegner, and A. Lucchi. Topological map extraction from overhead images. In *ICCV*, 2019.

[32] Z. Li, B. Xu, and J. Shan. Geometric object based building reconstruction from satellite imagery derived point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13:73–78, 06 2019. doi: 10.5194/isprs-archives-XLII-2-W13-73-2019.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[34] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[35] C. Liu, D. Yang, D. Ceylan, E. Yumer, and Y. Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, 2018.

[36] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *CVPR*, 2019.

[37] J. Lussange. Biyolo code repository, 2022. URL https://github.com/johannlussange/biyolo. Accessed: 2022-11-18.

[38] LuxCarta. Procédé de reconstruction d'un modèle 3d d'un toit d'un bâtiment par analyse d'images acquises par télédétection, 2022. URL https://data.inpi.fr/brevets/FR3123753?q=#FR3123753. Accessed: 2023-03-18.

[39] E.-T. Lê, M. Sung, D. Ceylan, R. Mech, T. Boubekeur, and N. J. Mitra. Cpfn: Cascaded primitive fitting networks for high-resolution point clouds. In *ICCV*, 2021.

[40] M. S. Minhas. Transfer learning for semantic segmentation using pytorch deeplab v3, Sep 2019. URL https://github.com/msminhas93/DeepLabv3FineTuning.

[41] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra. Rapter: rebuilding man-made scenes with regular arrangements of planes. *Trans. on Graphics*, 34(4), 2015.

[42] J. S. Nicolas Girard, Dmitriy Smirnov and Y. Tarabalka. Polygonal building extraction by frame field learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5891–5900, 2021.

[43] T. Partovi, F. Fraundorfer, R. Bahmanyar, H. Huang, and P. Reinartz. Automatic 3-d building model reconstruction from very high resolution stereo satellite imagery. *Remote Sensing*, 11(14), 2019.

[44] J.-S. Proulx-Bourque and M. Turgeon-Pelchat. Toward the use of deep learning for topographic feature extraction from high resolution optical satellite imagery. In *IGARSS 2018*, pages 3441–3444, 07 2018. doi: 10.1109/IGARSS.2018.8519171.

[45] Y. Qian, H. Zhang, and Y. Furukawa. Roof-gan: Learning to generate roof geometry and relations for residential houses. In *CVPR*, 2021.

[46] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5), 2006.

[47] J. Ren, B. Zhang, B. Wu, J. Huang, L. Fan, M. Ovsjanikov, and P. Wonka. Intuitive and efficient roof modeling for reconstruction and synthesis. *ACM Trans. on Graphics*, 40(6), 2021.

[48] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer graphics forum*, 26(2), 2007.

[49] G. Sharma, D. Liu, S. Maji, E. Kalogerakis, C. Siddhartha, and R. Mech. Parsenet: A parametric surface fitting network for 3d point clouds. In *ECCV*, 2020.

[50] C. Stucker, B. Ke, Y. Yue, S. Huang, I. Armeni, and K. Schindler. ImpliCity: City modeling from satellite images with deep implicit occupancy fields. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2022.

[51] B. Sun and P. Mordohai. Oriented point sampling for plane detection in unorganized point clouds. In *ICRA*, 2019.

[52] S. Tripodi, L. Duan, F. Trastour, V. Poujad, L. Laurore, and Y. Tarabalka. Automated chain for large-scale 3d reconstruction of urban scenes from satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019.

[53] A. Vali, S. Comai, and M. Matteucci. Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review. *Remote Sensing*, 12 (15), 2020. ISSN 2072-4292. doi: 10.3390/rs12152495. URL https://www.mdpi.com/2072-4292/12/15/2495.

[54] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, 2015.

[55] waspinator. pycococreator code repository, 2021. URL https://github.com/waspinator/pycococreator. Accessed: 2022-11-18.

[56] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, and Z. Xu. Regnet: Self-regulated network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–6, 2022. doi: 10.1109/TNNLS.2022.3158966.

[57] S. Yan, Z. Yang, C. Ma, H. Huang, E. Vouga, and Q. Huang. Hpnet: Deep primitive segmentation using hybrid representations. In *ICCV*, 2021.

[58] F. Yang and Z. Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *ECCV*, 2018.

[59] D. Yu, S. Ji, J. Liu, and S. Wei. Automatic 3d building reconstruction from multi-view aerial images with deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 171, 2021.

[60] M. Yu and F. Lafarge. Finding Good Configurations of Planar Primitives in Unorganized Point Clouds. In *Proc. of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, US, 2022.

[61] Z. Yu, J. Zheng, D. Lian, Z. Zhou, and S. Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1029–1037, 2019.

[62] H. Zeng, J. Wu, and Y. Furukawa. Neural procedural reconstruction for residential buildings. In *ECCV*, 2018.

[63] W. Zhao, C. Persello, and A. Stein. Extracting planar roof structures from very high resolution images using graph neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187, 2022.

[64] X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A review. *IEEE Geoscience and Remote Sensing Magazine (GRSM)*, 10 2017.

[65] S. Zorzi, S. Bazrafkan, S. Habenschuss, and F. Fraundorfer. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *CVPR*, 2022.