PHASED ACCURACY ANALYSIS IN ROAD CONSTRUCTION: USING BIM AND PHOTOGRAMMETRIC OUTPUT

H. De Winter[1,2*], Maarten Bassier[1], Sam De Geyter[1,3], Maarten Vergauwen[1]

[1] Dept. of Civil Engineering – Geomatics, KU Leuven – Faculty of Engineering Technology, Ghent, Belgium
[2] DIRK BAUWENS NV, Evergem, Belgium
[3] MEET HET BV, Mariakerke, Belgium
(heinder.dewinter, maarten.bassier, sam.degeyter, maarten.vergauwen)@kuleuven.be

Commission IV, WG IV/4

KEY WORDS: BIM, Photogrammetry, Quality analysis, Road Construction, RTK-UAV

ABSTRACT:

The construction industry is experiencing a gradual shift towards digitization. Currently, this digitization is relatively limited, but the need for that digitization as well as for automation is large. Digitization in road construction in particular is even more dire than in the construction industry in general. It involves various tasks such as progress, quality and quantity analyses. In this paper a proof of concept is presented on the possible automation of accuracy analysis in road construction throughout the construction of a road. The used data is 3D as-design data on the one hand and multiple 3D as-built data on the other hand, with the former being modelled using the regulations formulated by the Flemish ”Departement van Mobiliteit en Openbare Werken” (MOW), and the latter being captured using a Real Time Kinematic Unmanned Aerial Vehicle (RTK-UAV). The feasibility of the proposed workflow is examined using a real test case. Conducting these quality checks regularly will provide a clear overview of the construction site progress. In this way, any delays or errors can be detected quickly, resulting in higher quality and reduced failure costs.

1. INTRODUCTION

It is apparent that throughout society digitization is taking on a more important role. This is not different for the road construction sector. Besides this digitization in the form of recording and assessment techniques, it can also be seen in the form of automation. For instance, certain analyses are open to automation and thus speeding up if the data is available in digital form. Analyses such as quantity, quality and progress monitoring are examples. In addition, profit margins in the construction industry are limited, as described in (Love et al., 2017), so failure and other unforeseen costs need to be limited to guarantee a profit. All this together shows that digital quality analysis in road construction is very important.

At the moment there are two main types of accuracy analyses that are conducted in road construction. The first analysis happens during the construction of the road, the second at the end when the construction site is completely finished. The analysis during the construction is typically performed manually and is a time-consuming task. This manual method is frequently performed using Global Navigation Satellite System (GNSS) devices and traditional surveying techniques, such as total station measurements (TS). These frequent analyses ensure that errors can be detected during construction and adjustments can be made in a timely manner. These analyses can reduce failure costs, by detecting errors in an earlier stage of the construction. The second analysis happens at the end of construction. This analysis is conducted in a destructive manner by performing drillings to check the thickness of the layers (Agentschap Wegen & Verkeer, 2019). In Flanders, the final analyses are required by ”Agentschap Wegen & Verkeer” (AWV) and if errors are detected there, they cause very high failure costs. Finally, similar measurements with TS or GNSS sensors are also conducted to get a full as-built plan of the construction site. It is clear that these analyses are selective whereby a limited number of points are measured.

The time-consuming and selective methods, in combination with the destructive method of control analyses, result in the need for other methods that are faster, holistic and preferably also non-destructive. The ideal solution for these needs is the regular and coordinated use of unmanned aerial vehicles (UAVs). Analogue to studies and experiments conducted by (De Winter et al., 2022), an RGB-camera is mounted on the UAV. Other sensors, such as LiDar, are also possible but are much more expensive than RGB cameras which create data with sufficient quality for further analyses. This makes it the ideal sensor for capturing medium-sized construction sites.

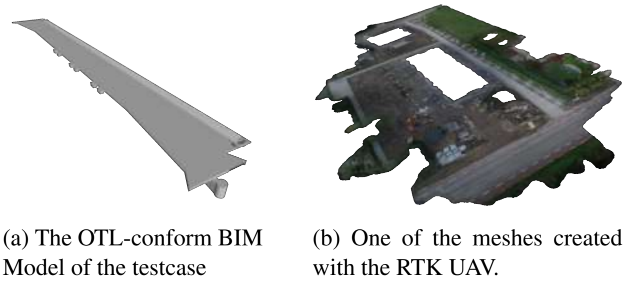
The testsite used in this research is a construction site located in Mariakerke, Belgium, and is snown in Figure 1. The as-design data has been created in accordance with the Object Type Library OTL of ”department mobiliteit en openbare werken” (MOW)[1]. See Figure 1a. The as-built data consists of multiple mesh models created with a real time kinematic (RTK) UAV, (Figure 1b). These different as-built datasets were created at different times throughout the construction of the site. In addition to the meshes created with photogrammetry, other output is created and used, such as georeferenced orthopictures and point clouds.

The remainder of this paper consists first of a literature review on related research, after which the methodology of the quality analysis is being discussed. Third, the experiments that are conducted are reviewed and the result of the analysis is compared with the ground truth. Finally, the conclusion of this work is discussed together with future work at the end.

[*] Corresponding author

[1] translation: ”mobility and public works department”

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-1627-2023 | © Author(s) 2023. CC BY 4.0 License.



(a) The OTL-conform BIM Model of the testcase

(b) One of the meshes created with the RTK UAV.

Figure 1: The testcase situated in Mariakerke, Belgium

2. RELATED WORK

This chapter reviews several researches related to quality analysis in construction and, in particular, road construction. First, the accuracy that can be achieved on the photogrammetric output of the data obtained with the RTK-UAV is discussed. Next other research on quality analysis is discussed, in two major parts: planimetric and altimetric.

First of all we tackle the accuracy of the photogrammetric output. In recent research (Ferrer-González et al., 2020, Štroner et al., 2020, Bi et al., 2021), regardless of the possible test situations, it is stated that accuracy in planimetry (x and y) is greater than in altimetry (z). According to the manufacturer of the used drone, a DJI Phantom 4 RTK, it should have a GNSS position accuracy of 0,01 m in planimetry and 0,015 m in altimetry (Bi et al., 2021). Several possible set-ups are discussed in literature, along with their corresponding accuracy. The final accuracy of the photogrammetric output depends, on the one hand on the accuracy of the location of the images – this would basically be the accuracy given by the manufacturer– and on the other hand the error of the generation of the photogrammetric output from the images. Factors that can affect the former are the speed of the drone, the number of FNSS satellites, multipath etc. Factors that can effect the latter include the altitude of the flight, the overlap between the pictures, both in width and length, the ground sampling distance (GSD), etc. All three papers describe that the best accuracy can be achieved by combining the RTK data with some Ground Control Points (GCPs). In the work of (Štroner et al., 2020) pre-calibrating the camera improves accuracy results. In the research of (Bi et al., 2021) incorporating oblique images alongside nadir images can also enhance accuracy. Additionally, in the work of (Ferrer-González et al., 2020) the focus is on the importance of the distribution and accuracy of GCPs, as this can significantly impact the accuracy of photogrammetric output. These studies offer valuable insights into strategies for improving the accuracy of photogrammetric output in aerial imaging.

Since quality control in road construction on 3D data is relatively new, resources are scarce. For the quality analysis itself, we also make a distinction between planimetry and altimetry. This is for two reasons, first, the specific accuracy requirements imposed by MOW are different for planimetry and altimetry, second, different georeferenced outputs are used for both analyses which results in different methodologies. In the technical note of AWV (Agentschap Wegen & Verkeer, 2020) different elements receive different thresholds (σ) to check if it is build correctly. These threshold differ in planimetry and altimetry. When an element's deviation doesn't exceed σ, the element is considered to be built correctly. In other words, it does not need to be rebuilt or redrawn for the as-built plan, provided of course that other requirements, such as thickness and slopes adhere to other specs, such as (Ministerie van de Vlaamse Gemeenschap, 2000). The latter is not discussed further as it exceeds the scope of this study.

For the planimetric quality analysis, the edges of the elements from the as-design model and the as-built model are compared using line detection. Several different edge detection methods have been described and discussed in literature over the past years. One of the most widely used methods is the ”Canny edge detector”, first described in (Canny, 1986). This is an edge detector that can handle only gray-scale images. The detector has some adjustable parameters and is an old but accurate and reliable method to detect lines in images. Another highly promising edge detector is the Holistically-Nested Edge detector (HED) (Xie and Tu, 2017). HED is a convolutional neural-network-based edge detection system. In contrast with the other edge detectors it uses RGB information if availble. Another recent and promising detection method, originally developed for segmentation, is the Segment Anything Model (SAM) (Kirillov et al., 2023). The boundary between two segmented objects can be seen as an edge and in this way boundaries can also be found. Several other edge detection methods exist, but for the aim of this research the problem can be resolved with these detectors. After edges have been detected, lines are constructed from these edges, using a method based on the Hough transform (Hough, 1962). The Hough Line Transform is a popular technique in computer vision for detecting straight lines within an image. In addition to these two-part methods, there is also a method that directly yields lines as output, namely the Mobile Line Segment Detection (MLSD). In their recent work (Gu et al., 2022), a lightweight network for line segment detection is proposed. By incorporating a matching and geometric loss function during training, the model is able to seamlessly integrate geometric cues.

As a result, the designed and built edges are known and consequently the distances between the lines need to be converted to the actual world, using the ground sampling distance (GSD) (Saini and Singh, 2021). There is also the option to reproject the pixel in the image into the real world using raycasting, such as in (Bowman and Hodges, 1997) and (Bowman et al., 1999). This entails the creation of a ray from a point in a certain direction, and then computing the intersection of the ray with the desired geometry, here a mesh. Other output like the id of the hit geometry can be retrieved as well. The difficulty here is that there must be a translation between the image coordinates and the world coordinates before the raycasting onto the mesh is possible. Depending on the needed accuracy and the processing time available the one is preferred over the other.

In order to determine the quality in altimetry, there are several photogrammetric outputs that can be used: a point cloud, like in the research of (Maalek et al., 2019), or a mesh which has however, not often been used in this type of research. In an analogue methodology the same can be done with a mesh, if points are sampled on the mesh to create a point cloud. The advantage of a mesh over a point cloud is the possibility of interpolation where occlusions occur. This is almost never done for regular construction sites, but for road construction it makes more sense because of the high number of large surfaces. Either way, the methodology is the same independent from photogrammetric input, because the start is a raycasting from every point either from the original point cloud or the created one.

3. METHODOLOGY

This section discusses the methodology for determining the quality of elements in road construction. This will, once more, be done separately for planimetry and altimetry.

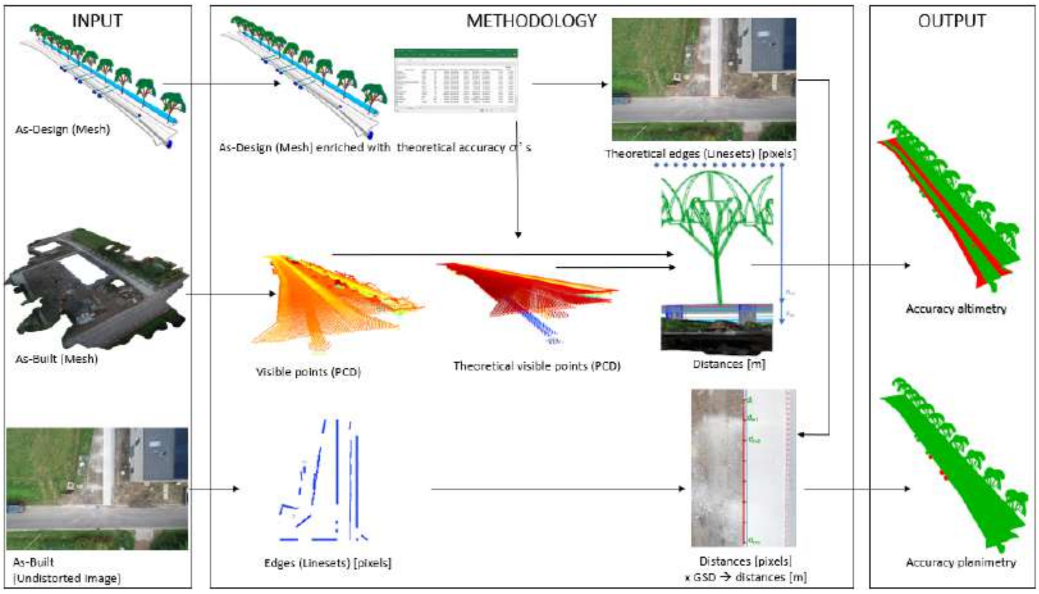


Figure 2: The proposed methodology of the accuracy analysis.

As shown in Figure 2, the algorithm consists of **4 general steps**. **(1)** Adding a theoretical σ, the accuracy needed determined by MOW, to each element of the as-design data. **(2)** Detecting linesets of the elements for both the as-build and as-design dataset. **(3)** Creating a visible point cloud both for the as-design and as-build dataset. This is a point cloud on top of both the as-design and as-built elements. **(4)** The final distance calculation between the detected linesets for planimetric accuracy and the visible point clouds for altimetric accuracy, and the conclusion for each element if an element is build correctly or not.

(1) The first step of the algorithm is to connect the σ_{plan} and σ_{alt}, determined by MOW, to the elements of the as-design model by means of the layer name. All metadata per element, i.e. Cartesian bounds, oriented bounds, timestamp, etc. of the as-built model is stored in graphs to optimise the calculation time, so the entire geometry doesn't need to be loaded. An example of a .ttl-file containing such graph in RDF format, can be seen in Figure 3.

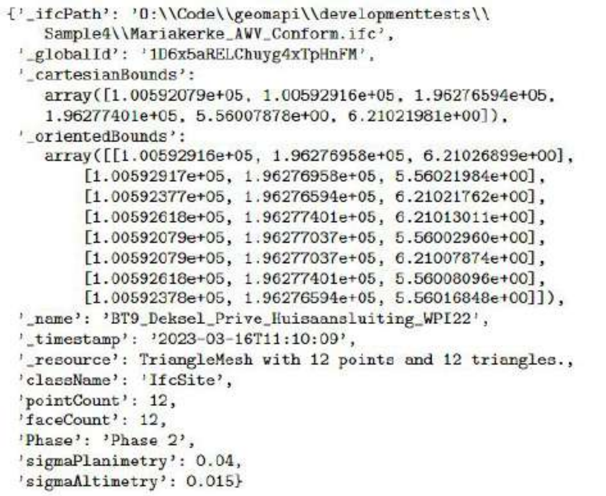

```
{'_ifcPath': 'O:\\Code\\geomapi\\developmenttests\\
    Sample4\\Mariakerke_AWV_Conform.ifc',
 '_globalId': '1D6x5aRELChuyg4xTpHnFM',
 '_cartesianBounds':
    array([1.00592079e+05, 1.00592916e+05, 1.96276594e+05,
    1.96277401e+05, 5.56007878e+00, 6.21021981e+00]),
 '_orientedBounds':
    array([[1.00592916e+05, 1.96276958e+05, 6.21026899e+00],
          [1.00592917e+05, 1.96276958e+05, 5.56021984e+00],
          [1.00592377e+05, 1.96276594e+05, 6.21021762e+00],
          [1.00592618e+05, 1.96277401e+05, 6.21013011e+00],
          [1.00592079e+05, 1.96277037e+05, 5.56002960e+00],
          [1.00592079e+05, 1.96277037e+05, 6.21007874e+00],
          [1.00592618e+05, 1.96277401e+05, 5.56008096e+00],
          [1.00592378e+05, 1.96276594e+05, 5.56016848e+00]]),
 '_name': 'BT9_Deksel_Prive_Huisaansluiting_WPI22',
 '_timestamp': '2023-03-16T11:10:09',
 '_resource': TriangleMesh with 12 points and 12 triangles.,
 'className': 'IfcSite',
 'pointCount': 12,
 'faceCount': 12,
 'Phase': 'Phase 2',
 'sigmaPlanimetry': 0.04,
 'sigmaAltimetry': 0.015}
```

Figure 3: An example of an RDF-graph of a household manhole.

(2) The second step of the algorithm is the detection of the edges of the elements in order to calculate the planimetric accuracy.

The first part **(a)** of step **(2)** is the boundary edge detection of each element of the as-design model. By using pyvista's extract feature edges (Sullivan and Trainor-Guitton, 2019), the edges of the as-built model can be found. These are located in a georeferenced coordinate system, such as Lambert 72 in Belgium. The as-design model and the detected theoretical lines can be seen in Figure 4. A LineSet $LineSet_{ad}$ is created containing the edges of the as-design model. $LineSet_{ad}$ is the union of $LineSet_{i,ad}$, where i is the number of the element. Let $M_{i,ad}$ be a triangular mesh of an element from the as-design model, with vertex set V and face set F. Let E be the set of edges of $M_{i,ad}$, where each edge is a pair of vertices in V that are connected by a line segment. So $LineSet_{i,ad}$ can be defined as in Eq. 1.

$$LineSet_{i,ad} = \{(p,q) \mid p,q \in V, p,q \text{ is an edge in } E\} \tag{1}$$

So the $LineSet_{ad}$ can be defined as:

$$\text{LineSet}_{ad} = \bigcup_{i=1}^{|n|} \text{LineSet}_{i,ad} \tag{2}$$

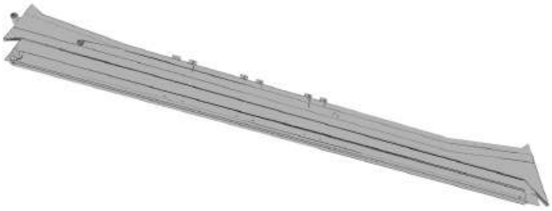
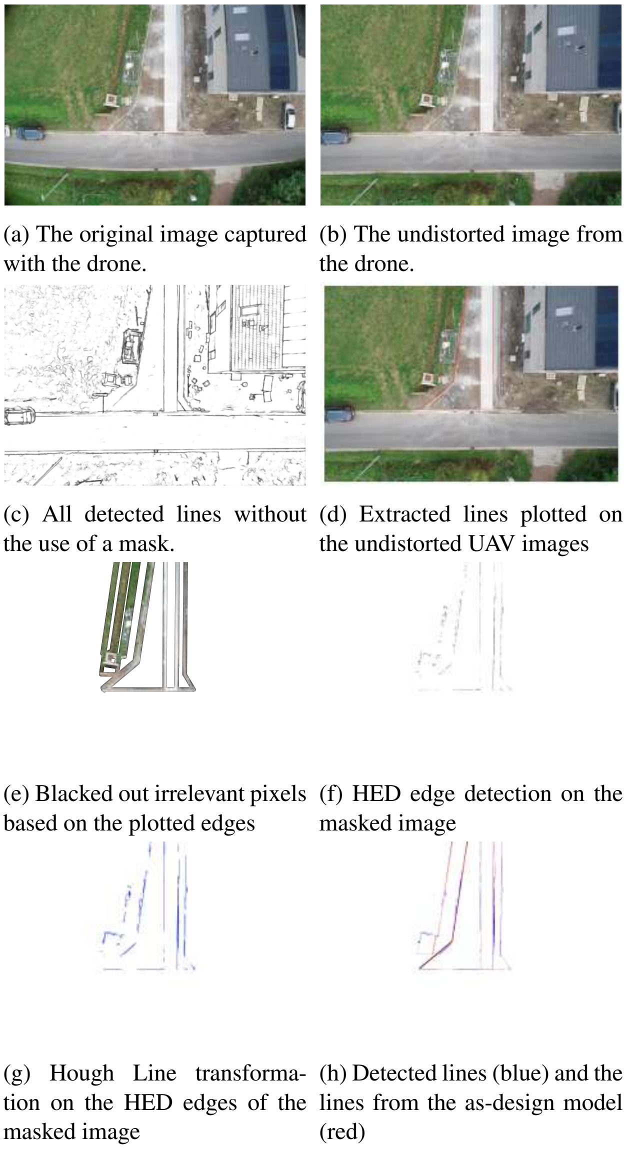


Figure 4: The extracted edges and the as-built model.

The second part **(b)** of step **(2)** is to use this lineset as an initial condition to detect the lines on as-built data, in the form of georeferenced undistorted images. Undoing the distortion is vital. Figure 5a shows an original, distorted images, whereas an undistorted image can be seen in Figure 5b. The choice is made to create a mask around the lineset created from the as-design model $LineSet_{ad}$ because the line detection method, here HED edge detection, detects plenty of lines from the environment that should not be taken into account for the accuracy calculations, such as lines from machinery, stacked materials, etc. See Figure 5c. First, the georeferenced lines from the as-design dataset $LineSet_{ad}$ are transformed into the coordinates of the undistorted image, and plotted, as shown in Figure 5d. All pixels that exceed the threshold distance d are blacked out, and hence a mask is created, as shown in Figure 5e. The remaing pixels are used for line detection. Then the linesets in pixels are found, using HED edge detection, see Figure 5f. The next step is to preform the Hough lines transformation, see Figure 5g which are shown in blue.

The output of step **(2)** is two-sided, on the one hand the $LineSet_{ab}$ for each image in pixels, i.e., the blue lines, and on the other hand the $LineSet_{ad}$ for each image in pixels, the red lines. This output is shown in Figure 5h.



(a) The original image captured with the drone.

(b) The undistorted image from the drone.

(c) All detected lines without the use of a mask.

(d) Extracted lines plotted on the undistorted UAV images

(e) Blacked out irrelevant pixels based on the plotted edges

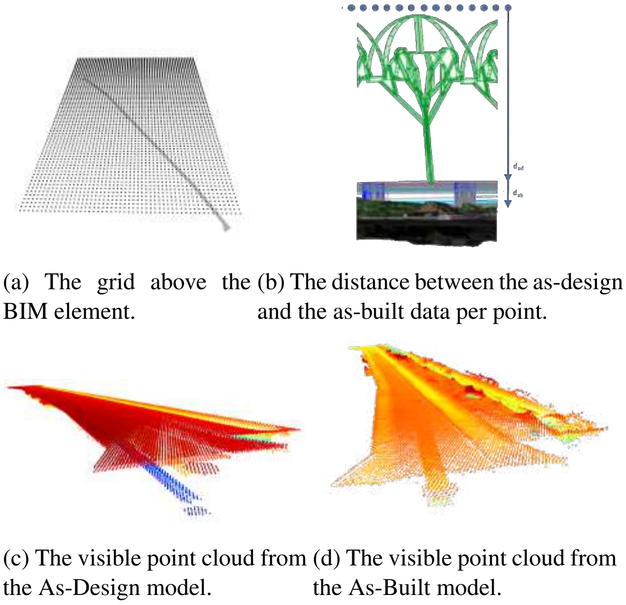
(f) HED edge detection on the masked image

(g) Hough Line transformation on the HED edges of the masked image

(h) Detected lines (blue) and the lines from the as-design model (red)

Figure 5: The different steps for the detection of the lines in the as-built data and the comparison with the as-design data.

(3) The third step of the algorithm is to create a visible point cloud for each element of the as-design model and the as-built model in order to calculate the quality in altimetry. Due to the existence of different layers or phases, the visible point cloud is limited to the top plane of each element. A grid G with resolution r is created for each element, shown in Figure 6a. A downwards ray is then cast from each point of this grid G. In this way, each point is assigned two distances: one where it intersects the as-design model (d_{ad}) and one where the ray intersects the as-built model (d_{ab}), see Figure 6b. Based on these values and the height of the grid, h_G, the top for every element of both as-design and as-built data can be created in the form of a point cloud. These calculations can be found respectively in Eq. 3 and Eq. 4, the visible point cloud from the full as-design model can be seen in Figure 6c and the visible point cloud from one of the as-built models can be seen in Figure 6d.



(a) The grid above the BIM element.

(b) The distance between the as-design and the as-built data per point.

(c) The visible point cloud from the As-Design model.

(d) The visible point cloud from the As-Built model.

Figure 6: The different steps for the creation of the visible point clouds.

$$P_{ad} = \{(x, y, z) \mid (x, y) \in G, z = G(x, y) - d_{ad}\}. \quad (3)$$

$$P_{ab} = \{(x, y, z) \mid (x, y) \in G, z = G(x, y) - d_{ab}\}. \quad (4)$$

(4) The fourth and final step of the algorithm is the distance calculation between the as-design and the as-built model.

The first calculation **(a)** yields the planimetric accuracy. The linesets based on the as-design model $LineSet_{ad}$ and the linesets based on the as-built model $LineSet_{ab}$ are in the same image pixel system, for each image. Each line of $LineSet_{ab}$ is associated with a line from the $LineSet_{ad}$, this is based on the element, the parallelism of the lines and the shortest distance. So a line of $LineSet_{i,ab}$ is associated to the closest line from $LineSet_{i,ad}$ for the same i, if the parallelism is within a threshold t. Multiple points are plotted on the $LineSet_{ab}$, with a certain distance from each other, and the perpendicular distance d_p, distance in pixels to $LineSet_{ad}$ is then calculated. This is formulated in Eq. 5, where $LineSet_{ad}$ is $ax + by + c = 0$ and x_i and y_i are the coordinates of each point on the $LineSet_{ab}$, that is just plotted.

$$d_p = \frac{|\sum_{i=1}^{n}(ax_i + by_i + c)|}{\sqrt{a^2 + b^2}} \quad (5)$$

Since the distance in pixels d_p is known and the GSD of each image can be calculated, because the sensor of the camera on the drone is known, as well as the height of the drone, the distance d

can be converted to an actual distance between each point of the $LineSet_{ab}$ and the $LineSet_{ad}$, as follows:

$$d = d_p \cdot GSD \tag{6}$$

The second calculation **(b)** is the method to calculate accuracy in altitude, Eq. 7 . In previous steps of the algorithm the outputs P_{ad} and P_{ab} were created with the same x,y coordinates, where each point (x_i, y_i, z_i) in P_{ad} and P_{ab} have the same x and y coordinates. Then the difference in height Δz between the two point clouds can be calculated as:

$$\Delta z = \sum_{i=1}^{n} (z_i^{(ad)} - z_i^{(ab)}) \tag{7}$$

Where n is the number of points in each point cloud, $z_i^{(ad)}$ is the z-coordinate of the i-th point in P_{ad}, and $z_i^{(ab)}$ is the z-coordinate of the i-th point in P_{ab}. This shows that both distances in altimetry Δz and distances in planimetry d are a set of multiple values per element from which a conclusion about the quality must be made.

There is an output for each element that reflects whether the element is built correctly or not, both for planimetry and altimetry. Ultimately, if desired, both these analyses can be done together to make a final decision on overall accuracy. But the final way to determine whether an element is built correctly is to visualize all values of d and Δz for each element in a separate histogram. Then, for both situations the percentage of inliers within interval between σ, 2σ, 3σ, and so on is determined. The σ to be used for this test depends on the σ of the particular element and whether it is a planimetric or altimetric control. The percentage of inliers that is needed depends on planimetry, 2D, and altimetry, 1D. Based on these percentages, it can be determined whether an element is built correctly.

4. EXPERIMENTS

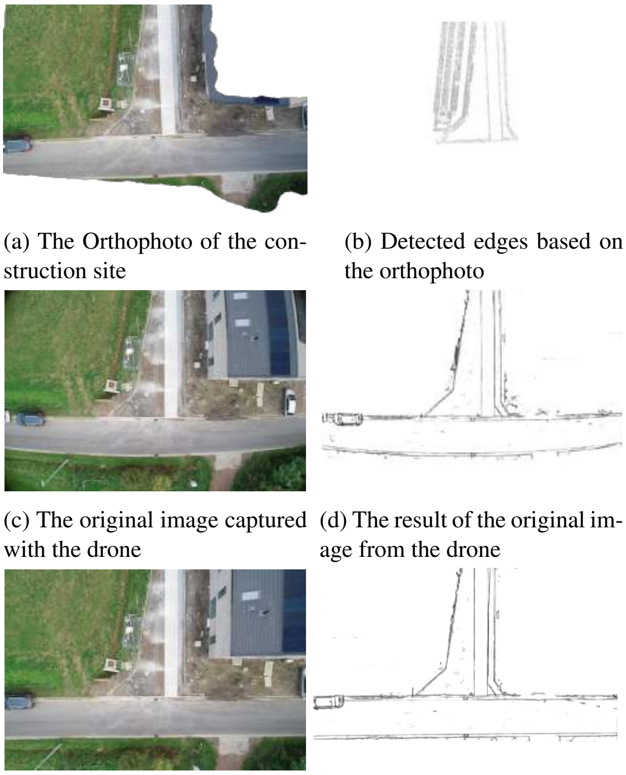
In this section several experiments are discussed. These experiments are conducted to determine the optimal values of the parameters in the methodology, such as which photogrammetric output, which line detection method and which decision framework to use.

The first experiment is on the type of photogrammetric output that is used for the quality analysis in planimetry. Several options are tested. Since we are dealing with 2D data, three real options are considered, namely **(a)** orthophotos (Figure 7a) **(b)**, raw images from the drone (Figure 7c), and **(c)** undistorted images from the drone (Figure 7e).

The first photogrammetric output tested was **(a)** the orthophoto. This looks like a promising option because all detected lines are already in the correct coordinate system. However, several problems are encountered, such as the fact that a .tiff files are very large, resulting in long processing times, unless one splits the image into smaller sub-orthophotos. Furthermore, an orthophoto is created, merging several photos together, which sometimes causes imperfect stitches between photos, so that clear lines in the construction site are not always perfectly visible on the orthophoto. As a result, we found that this was not the ideal input for line detection. Results can be seen in Figure 7b.

The second option for a photogrammetric output is the drone image **(b)**. Unlike the orthophoto, this image is not too large and thus simple to process. On the other hand, such a drone image does have the disadvantage of distorion, especially near the edges of the image. This causes the straight lines to become rounded. As a result, this is not the ideal photogrammetric output for line detection. Results can be seen in Figure 7d.

The third and final option is the undistorted drone image **(c)** Figure 7e. This option has the same advantages as the original image with the added advantage that the distortions are removed. The result of the linedetection on this undistorted image can be seen in Figure 7f. We conclude from this first experiment that the ideal photogrammetric output for planimetric quality control is undistorted drone images.



(a) The Orthophoto of the construction site

(b) Detected edges based on the orthophoto

(c) The original image captured with the drone

(d) The result of the original image from the drone

(e) The undistorted image from the drone

(f) Detected edges based on the undistorted image

Figure 7: The different options of photogrammetric outputs for planimetric quality control and their results

The second experiment is to determine the optimal line detection algorithm on the previously chosen photogrammetric output, namely the undistorted drone images. Four different ways to detect the lines are considered, three two-step algorithms and one one-step algorithm. The first method *(1)* is based on the Canny Edge line detection. The second method *(2)* is based on HED edge detection. The third option *(3)* is based on SAM. These three options are followed with the Hough line detection in order to get lines instead of edges. The fourth *(4)* and last option is a one step method named MLSD.

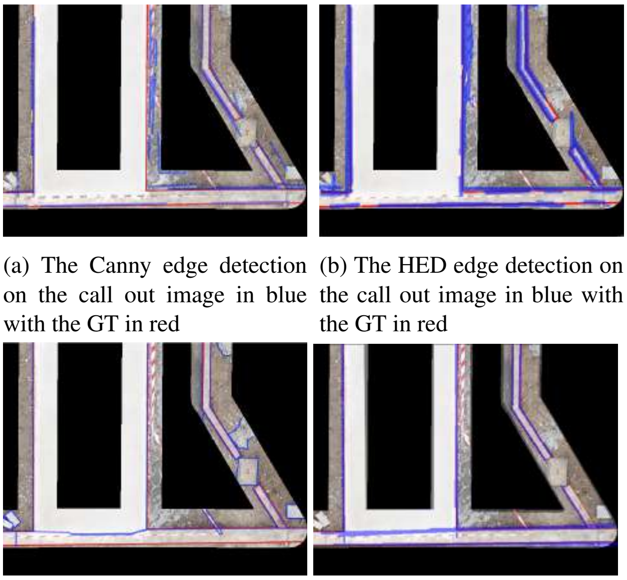
The first method *(1)* that is tested is the Canny edge detector in combination with Hough line transform. This is a fast method, see Table 1 but it creates many irrelevant lines, even when tweaking the parameters. Many of these irrelevant lines can be removed by the Hough line transformation, as can be seen in Figure 8a. Some lines, like the edge of the concrete, are not detected because of the more visible edge of the safety ribbon close to the edge. It appears that Canny is not an ideal method for planimetric control of elements.

The second method *(2)* is the HED edge detector in combination with Hough line transform. This is a slightly slower method, see Table 1; than the previous one, but it creates less irrelevant lines. The edges detected by HED after Hough are shown in Figure 8b. All desired lines are detected but the lines are thicker than in the previous method.

The third method *(3)* is the SAM method in combination with Hough line transform. This method is very slow, see Table 1, especially when calculating on the CPU. This is an issue if we need to process hundreds of images. The edges detected by SAM are shown in Figure 8c. Irrelevant lines are still detected.

The fourth and last method *(4)* is the only one-step method, i.e. MLSD. This method is comparable in speed to the HED method, see Table 1. The quality is also in the same order of magnitude as HED. The MLSD result can be seen in Figure 8d.

From this experiment, it appears that in this application the preferred method is the HED edge detector combined with Hough lines, on the one hand, and the MLSD method, on the other. Both have similar processing time and similar quality. For this research, the HED combined with Hough will be chosen.



(a) The Canny edge detection on the call out image in blue with the GT in red

(b) The HED edge detection on the call out image in blue with the GT in red

(c) The SAM detection on the call out image in blue with the GT in red

(d) The MLSD detection on the call out image in blue with the GT in red

Figure 8: The different options of edge detection on a call out of an image

	Processing time [s]
Canny + Hough	0.2
HED + Hough	1.5
SAM + Hough	195
MLSD	0.6

Table 1: Processing time of the different Line detection methods for one picture.

The third experiment is on the decision framework, using the results from the proposed methodology in 3. The two methods, planimetry **(a)** and altimetry **(b)**, are discussed separately. In planimetry the results boils down to computing the distance between two lines, while in altimetry it is the distance between two points.

For the accuracy in planimetry **(a)** each line of the as-design model has an assigned line from the as-built model. A first method *(1)* to make the decision about quality in planimetry is to set up the equation of the lines and then calculate the distance between them based on the formula of parallel lines. The problem is multi-sided: on the one hand the lines are not perfectly parallel, so the distance is not exactly the same across the entire line while the formula yields only one distance. On the other hand, it is impossible to compare this distance to a statistical value, such as the σ_{plan}, provided by MOW. A second method *(2)* is to plot points on the line and calculate the perpendicular distance to the associated line. The advantage of this is that this yields several distances for a line, unlike the previous method. The average can be calculated and compared to the theoretical value. However, once more MOW expects a statistical analysis with respect to a standard deviation, so this is not feasible either. A third and final option *(3)* is to perform a statistical (2D) comparison. We calculate the percentage of inliers that fall within 1.51σ and 2.45σ. If these percentages are greater than 68% and 95%, respectively, then the element is built correctly. This final option is chosen.

For accuracy decisions in altimetry **(b)**, each point on the grid per element has a value that represents the difference between the as-design model and the as-built model, as described in section 3. Different methods are compared to find the best option for a conclusion over the accuracy in altimetry. The first *(1)* option is to calculate the average of the deviations. Just like the method of the planimetry, one value can not be compared with a standard deviation. A second option *(2)* is to calculate the percentage of inliers that fall within σ and 2σ and check if this percentage is greater than 68% and 95%, respectively. If so, the element is built correctly. This ensures that the σ value and signed distances are taken into account. This option is chosen.

5. RESULTS

The results from our experiments are compared to external manual methods to check the accuracy of our algorithm. The quality in altimetry is checked using CloudCompare and the quality in planimetry is checked by manually drawing lines on the images which are processed further using Rhino and Grasshopper. The as-built data are images captured during the works of phase 5 of the construction site. The mesh and isolated elements from this phase are shown in Figure 9.

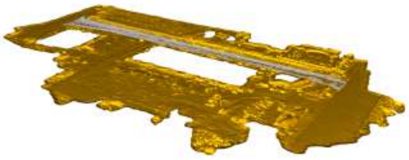
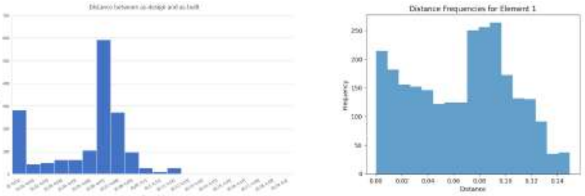


Figure 9: The mesh (in yellow) and the elements from the as-design model from Phase 5 (in grey).

The first comparison with the ground truth is the planimetric comparison. The different values calculated by our algorithm are compared to an external calculation of the same elements via Rhino and Grasshopper. For the ground truth, lines are drawn manually on the photos. In addition, lines from the BIM-model are projected to 2D. Both linesets are imported into Rhino and

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-1627-2023 | © Author(s) 2023. CC BY 4.0 License.

by using Grasshopper, points are sampled on these lines. Distances are then calculated based on a selected line and the sampled points, also in grasshopper. All calculated distances are transferred to an excel file and a histogram is made. An example of the histogram of one of the elements can be seen in Figure 10a.The histogram created with our own algorithm is shown in Figure 10b. These histograms and values, mean and σ, are compared with the proposed method to evaluate the quality.



(a) Planimetric differences with Rhino and Grasshopper (b) Planimetric differences with own methodology

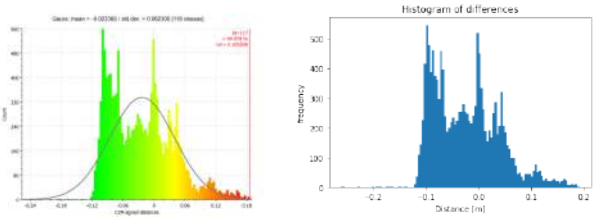
Figure 10: The differences between the planimetric histograms

The histograms have similar shapes but there is more noise on the proposed method. This can be explained by the overlap in the detected lines in one image, on the one hand, and by the overlap in the detected lines in consecutive images, on the other hand, as this overlap is situated slightly next to each other. The other values of this element can be seen in Table 2. There is little difference in these values: the mean is ± 0,01 m and the σ_{plan} is ± 0,003 m. This is in line with other elements with a max of ± 0,03 m in mean and ± 0,01 m in σ_{plan}. These are elements with less cleaner lines. It should be noted that in our method as-built lines are assigned to the closest as-design line, so if an element (e.g., a curbstone) is built wrong more than half the width of the element, the left as-built line will be paired with the right as-design line. This should be checked in future research.

	analogue [m]	proposed method [m]	difference [m]
mean	0,053862512	0,065606008	0,011745
σ	0,029864716	0,033280295	0,0034156

Table 2: Differences between the values of the altimetric method.

The second comparison to the ground truth is the altimetric comparison. The different values calculated by the algorithm are compared with an external calculation of the same elements via CloudCompare. Analogue to the planimetric control the same element is discussed. The other elements are treated the same way. The histogram created by CloudCompare is shown in Figure 11a and the one according to our algorithm in Figure 10b.



(a) Altimetric differences with CloudCompare (b) Altimetric differences with own methodology

Figure 11: The differences between the altimetric histograms

Analogue to the planimetric ground truth control the same three elements are taken into account, the mean, the σ_{alt} and the histogram.

The histograms clearly have almost the exact similar shapes. The other values, the mean and σ, of this element, are shown in table 3. The difference in the mean and the σ_{alt} are both too small to mention. Other elements show results in the same magnitude, so it can be concluded that the control in planimetry of the self-developped method has the same accuracy as CloudCompare.

	Altimetry		
	analogue [m]	proposed method [m]	difference [m]
mean	- 0.023389	- 0.02339366	0.0000047
σ	0.062300	0.0624876	0.000186

Table 3: Differences between the values of the planimeric method.

The results of comparing the ground truth with the self-developed method show that both options are reliable, the planimetric and the altimetric quality control, but need more extensive testing in the future to provide more clarity on this accuracy such as multiple phases, as well as multiple different construction sites.

6. CONCLUSION

In this paper, a method for assessing the overall 3D quality of an element based on a BIM model and RTK-UAV data has been proposed. The method of assessing and ensuring this quality is split into two parts, planimetry on the one hand and altimetry on the other. Both methods have their own methodology and different photogrammetric inputs. Based on the average sigmas given by MOW we conclude that a HED edge detection combined with Hough yields sufficiently accurate and fast results in planimetry. For the altimetric quality control there is only one proposed method, based on a grid and raycasting. When the results created by the proposed workflow are compared to the ground truth, the results are promising.

In the future, this algorithm needs to be applied to multiple datasets to check, as well as improve, its robustness. If it turns out that accuracy is not sufficient with the HED combined with Hough lines, other options can be better explored, with the consequence that the speed of processing might be lower. The issue in planimetry, where a wrong assumption is made for narrow elements, if such element is built wrong more than half of the width of the element, should be fixed in the future. A next step would be to adapt the as-design model based on this check. If the deviation is within limits given by MOW, adjustments are not needed. But if an element is built incorrectly, and other regulations such as gradient are respected, the element could be moved or manipulated.

ACKNOWLEDGEMENTS

This project has received funding from the VLAIO COOCK project (grant agreement HBC.2019.2509),942 the VLAIO BAEKELAND programme (grant agreement HBC.2022.0153) in collaboration with BAUWENS NV, the943 FWO Postdoc grant (grant agreement 1251522N) and the Geomatics research group of the Department of Civil Engineering, at the KU Leuven in Belgium

REFERENCES

Agentschap Wegen & Verkeer, 2019. BIM-protocol Infrastructuurprojecten.

Agentschap Wegen & Verkeer, 2020. Instructiebundel voor opmaak en aanlevering van technische documentatie - Versie 2.2. p. 152.

Bi, R., Gan, S., Yuan, X., Li, R., Gao, S., Luo, W. and Hu, L., 2021. Studies on three-dimensional (3d) accuracy optimization and repeatability of uav in complex pit-rim landforms as assisted by oblique imaging and rtk positioning. Sensors.

Bowman, D. A. and Hodges, L. F., 1997. Evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. Proceedings of the Symposium on Interactive 3D Graphics pp. 35–38.

Bowman, D. A., Johnson, D. B. and Hodges, L. F., 1999. Testbed evaluation of virtual environment interaction techniques. ACM Symposium on Virtual Reality Software and Technology, Proceedings, VRST pp. 26–33.

Canny, J., 1986. A Computational Approach to Edge Detection. IEEE Transactions On Pattern Analysis And Machine Intelligence,.

De Winter, H., Bassier, M., De Geyter, S. and Vergauwen, M., 2022. Automatic Calculation of Volume Changes in Road Construction. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 48(2/W2-2022), pp. 23–30.

Ferrer-González, E., Agüera-Vega, F., Carvajal-Ramírez, F. and Martínez-Carricondo, P., 2020. UAV photogrammetry accuracy assessment for corridor mapping based on the number and distribution of ground control points. Remote Sensing.

Gu, G., Ko, B., Go, S., Lee, S.-h., Lee, J. and Shin, M., 2022. Towards Light-weight and Real-time Line Segment Detection. Proceedings of the 36th AAAI Conference on Artificial Intelligence 36, pp. 1052–1059.

Hough, P. V., 1962. Method and means for recognizing complex patterns. Machine Vision.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P. and Girshick, R., 2023. Segment Anything. arXiv:2304.02643.

Love, P. E. D., Teo, P. and Morrison, J., 2017. Revisiting Quality Failure Costs in Construction. Journal of Construction Engineering and Management 144(2), pp. 05017020.

Maalek, R., Lichti, D. D. and Ruwanpura, J. Y., 2019. Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction. Remote Sensing.

Ministerie van de Vlaamse Gemeenschap, 2000. Standaardbestek 250 voor de wegenbouw. Standaardbestek 250 p. 1285.

Saini, A. and Singh, D., 2021. DroneRTEF:development of a novel adaptive framework for railroad track extraction in drone images. Pattern Analysis and Applications 24(4), pp. 1549–1568.

Štroner, M., Urban, R., Reindl, T., Seidl, J. and Brouček, J., 2020. Evaluation of the georeferencing accuracy of a photogrammetric model using a quadrocopter with onboard GNSS RTK. Sensors (Switzerland).

Sullivan, C. and Trainor-Guitton, W., 2019. PVGeo: an open-source Python package for geoscientific visualization in VTK and ParaView. Journal of Open Source Software 4(38), pp. 1451.

Xie, S. and Tu, Z., 2017. Holistically-Nested Edge Detection. International Journal of Computer Vision 125(1), pp. 3–18.