A TWO-LEVEL APPROACH FOR THE CROWD-BASED COLLECTION OF VEHICLES FROM 3D POINT CLOUDS

Volker Walter[*], Michael Kölle, David Collmar, Yongxiang Zhang

Institute for Photogrammetry (ifp), University of Stuttgart, Germany
Geschwister-Scholl-Strasse 24D, 70174 Stuttgart
firstname.lastname@ifp.uni-stuttgart.de

Commission IV, WG 4

KEY WORDS: Crowd, Paid Crowdsourcing, Data Collection, Vehicle Detection, 3D Point Clouds, Quality Improvement

ABSTRACT

In this article, we present a two-level approach for the crowd-based collection of vehicles from 3D point clouds. In the first level, the crowdworkers are asked to identify the coarse positions of vehicles in 2D rasterized shadings that were derived from the 3D point cloud. In order to increase the quality of the results, we utilize the wisdom of the crowd principle which says that averaging multiple estimates of a group of individuals provides an outcome that is often better than most of the underlying estimates or even better than the best estimate. For this, each crowd job is duplicated 10 times and the multiple results are integrated with a DBSCAN cluster algorithm. In the second level, we use the integrated results as pre-information for extracting small subsets of the 3D point cloud that are then presented to crowdworkers for approximating the included vehicle by means of a Minimum Bounding Box (MBB). Again, the crowd jobs are duplicated 10 times and an average bounding box is calculated from the individual bounding boxes. We will discuss the quality of the results of both steps and show that the wisdom of the crowd significantly improves the completeness as well as the geometric quality. With a tenfold acquisition, we have achieve a completeness of 93.3 percent and a geometric deviation of less than 1 m for 95 percent of the collected vehicles.

1. INTRODUCTION

Crowdsourcing, a neologism of the words “crowd” and “outsourcing” (Howe 2006), describes the outsourcing of activities of companies to an indefinite mass of people through an open call via the internet. Compared to classic outsourcing to well-known third-party companies, it is a more flexible and faster form of task distribution. Applications of this modern form of work organization are broad, ranging from data collection (Shehan 2018) to product design (Niu et al. 2019) to software development (Dubey et al. 2016), to name a few.

Many crowdsourcing projects are based on the work of unpaid volunteers, such as Wikipedia (www.wikipedia.org) or Zooniverse (www.zooniverse.org). The voluntary collection of geospatial data is known as Volunteered Geographical Information - VGI (Goodchild 2007). The most popular VGI project is OpenStreetMap (OSM - www.openstreetmap.org), an open collaborative project to create a detailed map of the world that can be edited by anyone (Haklay and Weber 2008).

VGI projects require an active community that is intrinsically motivated to work together. The main factors, that user collect OSM data voluntarily are, that their contributions are free and that other users benefit from it in the shape of digital maps (Budhathoki and Haythornthwaite 2012). This does not work for all applications: In the field of geodata collection, there are many tasks that could in principle be solved with crowdsourcing, but the bottleneck is finding enough volunteers and building an active community.

The most common extrinsic motivation for crowdworkers to complete tasks, and the motivation that leads to the fastest results, is monetary incentives (Haralabopoulos et al. 2019). In paid crowdsourcing, tasks are published on online marketplaces, which are responsible for the recruitment and payment of workers. The workers are compensated financially for completing the tasks (Mao et al. 2013). Established marketplaces, such as microWorkers (www.microworkers.com) (Hirth et al. 2011) or Amazon Mechanical Turk (MTurk - www.mturk.com) (Ipeirotis 2010), have large numbers of registered crowdworkers. Beside monetary incentives, also other forms of motivation that result in economic gains can be used. For example, Juhász and Hochmair (2018) have demonstrated that an extra credit assignment for students in two GIS courses can be used for getting faster results.

The realization of paid crowdsourcing projects is also possible without marketplaces, but the recruitment of crowdworkers would involve much effort. The workers on a crowdsourcing marketplace are automatically notified when an employer offers a new job. Employers may limit jobs to certain groups of employees. For example, it is possible to offer crowd jobs only to workers who live in a certain country, or to workers who have already successfully performed a certain number of other jobs. Further qualifications are possible with specially developed tests that must be solved before the crowdjob.

Quality control is a challenge in paid and voluntary crowdsourcing (Leibovici et al. 2017; Liu et al. 2018), as the quality of crowdsourced jobs can vary widely (Vaughan 2017). The crowd is composed of people with unknown and diverse skills, abilities, interests, personal goals, and technical resources (Daniel et al. 2018). Another problem - especially in paid crowdsourcing - is dishonest workers who try to maximize their income by submitting as many tasks as possible, producing incomplete or sloppy results (Hirth et al. 2011). In addition, there may exist adversarial workers that could greatly harm the quality of the collected data (Zhang et al. 2016). A challenge in crowdsourcing is to derive high-quality ground truth from noisy data collected by non-experts (Zhou et al. 2012).

[*] Corresponding author

https://doi.org/10.5194/isprs-annals-V-4-2021-97-2021 | © Author(s) 2021. CC BY 4.0 License.

Basically, there are two approaches to control and improve the quality of paid crowdsourced data (Zhang et al. 2016): “Quality Control on Task Designing” and “Quality Improvement after Data Collection”. The first approach guides crowdworkers to provide high-quality data. There are many methods to do this, such as skill testing, reputation systems, task assignment, task and workflow optimization, training, real-time quality assurance, quality control points, or incentive payment mechanisms. A discussion of such techniques can be found in (Daniel et al., 2018).

The second approach (quality improvement after data collection) is based on methods that improve the quality of data after it has been collected. A commonly used approach is based on repeated data collection by different crowdworkers. After data collection, procedures are used to filter out noisy data and infer truth. Estimating truth from noisy, repeatedly collected data is referred to as "truth inference" (Zheng et al. 2017).

This follows the idea of the "Wisdom of the Crowd". Surowiecki (2004) has shown in his book “*The Wisdom of the Crowd – why many are smarter than the few and how collective wisdom shapes business, economics, societies and nations*” that averages of several guesses are often better than the best individual guess. Groups of people are smarter and can solve complicated problems even better than specialists. For this, we need multiple representations as input (which can be easily realized with paid crowdsourcing, but would be difficult to achieve with voluntary data collection) and an aggregation rule (e.g., averaging) to make a decision (Simons 2004). Sir Francis Galton first observed this principle in 1907 (Galton 1907). He found that the average estimate of the weight of an ox in a weight competition at a farmers' market exceeded the accuracy of expert opinions (butchers). The average estimation converged almost to an optimal result. On average, the fair audience estimated the weight of the ox at 1197 pounds. The actual weight was 1198 pounds.

In this article, we use the wisdom of the crowd principle to improve the completeness and the geometric quality of the collection of vehicles from 3D point clouds. The interpretation of 3D point clouds is a non-trivial task and can be challenging for non-experts. Most of the existing work in the field of crowd-based geodata collection concentrates on 2D image data and only few works use 3D point clouds as data basis. Herfort et al. (2018) describe the use of majority voting for the crowd-based detection of trees in 3D LiDAR point clouds. Koelle et al. (2020) discuss a “human-in-the-loop” system to classify 3D LiDAR point clouds where the machine in the shape of a machine-learning algorithm iteratively improves its performance by learning from paid crowdworkers. In a preliminary work (Walter et al. 2020), we discuss the crowd-based collection of trees from 3D point clouds by means of minimum enclosing cylinders. Each crowd job was duplicated 10 times. Integrated cylinders were calculated from the individual cylinders by averaging the centres and the heights of the individual cylinders. We demonstrated that the quality of the integrated cylinders is significantly higher than the average quality of the individual cylinders. The difference between the approach for tree collection and the approach we describe in this article is that in this approach we divide the data collection into two steps. First, crowdworkers are asked to identify only the coarse positions of the vehicles. These coarse positions are used as input for a second step in which the vehicles are to be approximated with MBBs. With this method, it is easier to collect data from areas with inhomogeneous distributed objects.

The detection of vehicles in remote sensing data can be important for many applications, e.g. traffic management, traffic monitoring, urban planning, parking lot analysis, etc. The new aspect of our approach is that we use not images as input but 3D point clouds. The advantage of using 3D point clouds is that we get a full 3D approximation of the vehicles (position, orientation, and dimension).

The remainder of this article is organized as follows. In section 2, we present an overview of the approach. The data from which the crowdworkers had to collect the data is presented in section 3. Section 4 is dedicated to the methodology and the results of the coarse positioning of vehicles. In section 5, we discuss the methodology and the results of the approximation of the vehicles with MBBs. A discussion of the overall approach and an outlook to future work can be found in section 6.

2. DATA COLLECTION

Most crowdworkers have no expert knowledge in the field of geospatial data collection and have never worked with 3D point clouds before. Therefore, it is necessary to design the data collection task as simple as possible in order that also non-specialists can solve them. A common approach for designing paid crowdsourcing tasks is to divide large problems into smaller sub-problems that can be solved quickly and easily. The typical working time for paid crowdsourcing tasks is in the range of some minutes and the payment is often only in the range of several cents (Hirth et al. 2011, Hitlin 2016). In principle, geospatial data collection tasks can be subdivided by splitting the working area geographically into many small tiles and assigning these tiles to individual crowdworkers. However, this is only reasonable if the objects, which should be collected, are homogeneously distributed in the working area, which is not the case in our data. If we would split the working area simply into tiles, we would produce many tiles that contain no vehicles at all.

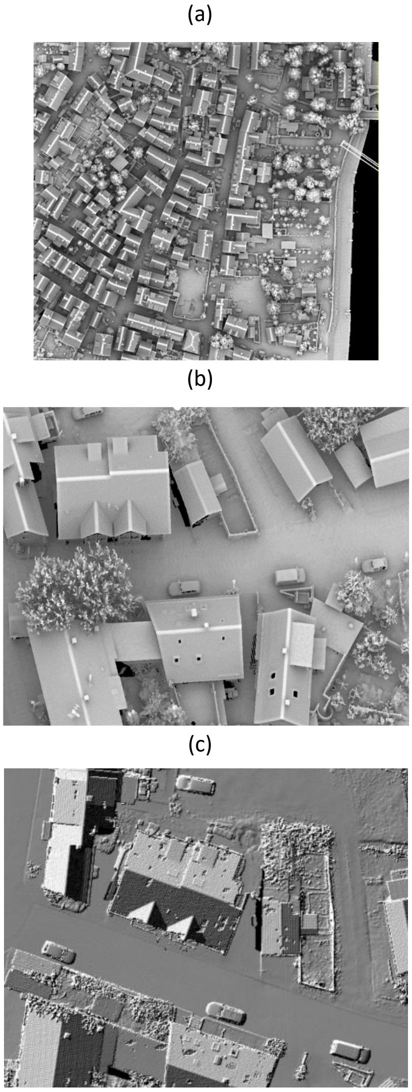
Therefore, we suggest a two-level approach. In the first level, we subdivide the working area into large strips and present them as 2D rasterized shadings to the crowdworkers in which they must identify the positions of all vehicles. In the second level, we use these positions as prior information to cut out small parts from the 3D point cloud for each vehicle. Each of these small 3D point clouds is then presented to a crowdworker to approximate the vehicle with a MBB. This incorporates the additional advantage that each crowdworker has to download only a small part of the 3D point cloud, which considerably reduces the download time.

All crowd jobs were published on the commercial platform microWorkers (www.microworkers.com) which handles the recruitment and the payment. According to their website, the platform has access to more than 2,200,000 registered crowdworkers (April 2021).

3. TEST AREA

For our test area, we focus on the western shore of the Hessigheim dataset presented in (Cramer et al. 2018, Haala et al. 2020). Hessigheim is located in the southern part of Germany. Our test area has a size of 0.25 km * 0.26 km (see

Figure 1). The point cloud was collected with a RIEGL VUX-1LR LiDAR sensor combined with two Sony Alpha 6000 oblique cameras using the RIEGL RiCopter octocopter in March 2018. The mean laser pulse density is 300-400 points/m² per strip and more than 800 points/m² for the entire flight block due to the nominal side overlap of 50%. The ranging accuracy, reported in the data sheet of the sensor is 10 mm (Riegl 2018).


(a)

(b)

(c)

Figure 1. Test area Hessigheim: (a) overview of the 3D point cloud, (b) zoom-in 3D point cloud, (c) zoom-in 2D rasterized shading

4. COARSE POSITIONING OF VEHICLES

For the coarse positioning, the test area was subdivided in nine east-west-oriented strips of 50 x 260 meters (using a 50% strip overlap). Since vehicles can be occluded by vegetation, we first filter out the vegetation points from the 3D point cloud. For this, we derive a Digital Terrain Model based on filtered ground points to calculate for every point an individual height above ground. Only those points are used for the calculation of the shading that have a maximum height above ground of 3 m. Shadings were calculated according to Tanaka's algorithm using a light source situated at 350 gon azimuth and 50 gon zenith angle. Due to the elimination of all points that are 3 m above ground, there may be raster cells which contain no 3D point at all (e.g. at buildings). In these cases, we use the maximum *z*-value of all points inside the cell to avoid holes in the shading. All calculations have been carried out using the Opals software (Pfeiffer et al. 2014).

The first task of the crowdworkers is to identify all vehicles in one strip by means of line segments reaching from the front to the back of the vehicle or vice versa. An example is shown in Figure 2.

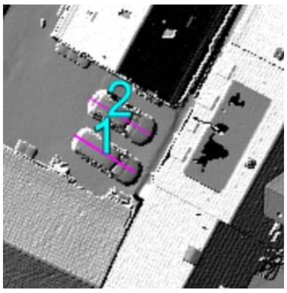


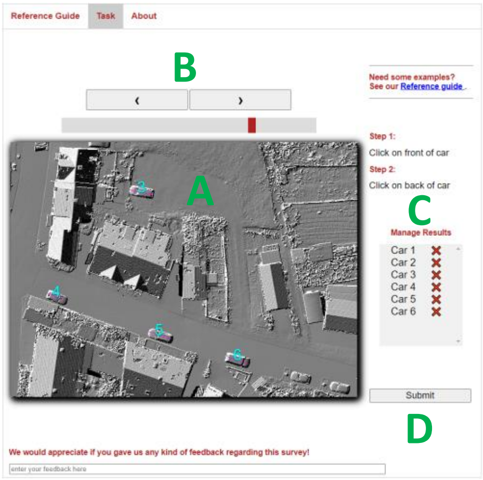
Figure 2. Example of two vehicles identified by crowdworkers in 2D rasterized shadings

4.1 GUI for coarse positioning of vehicles

The Graphical User Interface (GUI) for the coarse positioning of the vehicles is shown in Figure 3. The GUI is subdivided into four parts:

- Data view (A): Shows the 2D rasterized shadings and the already collected vehicles.
- Data control elements (B): The strips can be moved left and right with two control buttons.
- Management area (C): All collected vehicles are shown in a list. Erroneously collected vehicles can be deleted from the list. An already collected vehicle can be activated and edited by clicking on the corresponding element in the list.
- A submit button (D): The crowd job can be finished by clicking the *Submit* button. The collected data will then be submitted to the server.

The GUI was developed with HTML, Javascript, CSS and PHP. All crowdjobs were be published on the microWorkers marketplace which handles the worker recruitment and payment. The recruited crowdworkers are pointed via an URL to the GUI that was installed on our own servers.



Need some examples?
See our Reference guide.

Step 1:
Click on front of car

Step 2:
Click on back of car

Manage Results

Car 1 ✗
Car 2 ✗
Car 3 ✗
Car 4 ✗
Car 5 ✗
Car 6 ✗

We would appreciate if you gave us any kind of feedback regarding this survey!

Figure 3. GUI for the coarse identification of vehicles

4.2 Crowdsourcing campaign for coarse positioning

In each crowd job, the workers must collect all vehicles from one strip. Each crowd job was duplicated 10 times. The number of strips is 9. For each crowd job, we paid $0.10. The total cost is 9 * 10 * $0.10 = $9.00. The average working time for one job is 5.06 minutes. Altogether 1920 vehicles were collected in all crowd jobs together. After the data integration 116 vehicles remained. The average cost per vehicle is $0.08.

4.3 Data integration

The multiple collected data is integrated by clustering the centre points of each collected line. We use DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), which is a density-based algorithm for the detection of clusters and outliers (Ester et al. 1996). The advantages of DBSCAN are that it is not necessary to specify the number of clusters in prior - like in k-means - and that it is robust to outliers.

DBSCAN requires two parameters: (1) *Epsilon* defines the maximum distance between two points to be considered as neighbours and (2) *MinPts* defines the minimum number of points in a cluster. Based on empirical tests we use *Epsilon* = 2 m and *MinPts* = 6.

Figure 4 shows the 2D coordinates of the centres of the collected vehicles of a part of our test area. The outliers detected with DBSAN are marked with red colour. The total number of outliers in the whole test area is 113. The remaining 1807 points are subdivided into clusters (Figure 5). The total number of clusters in the whole test area is 116. The final positions of the clusters are calculated by averaging the *x*- and *y*-positions (Figure 6).

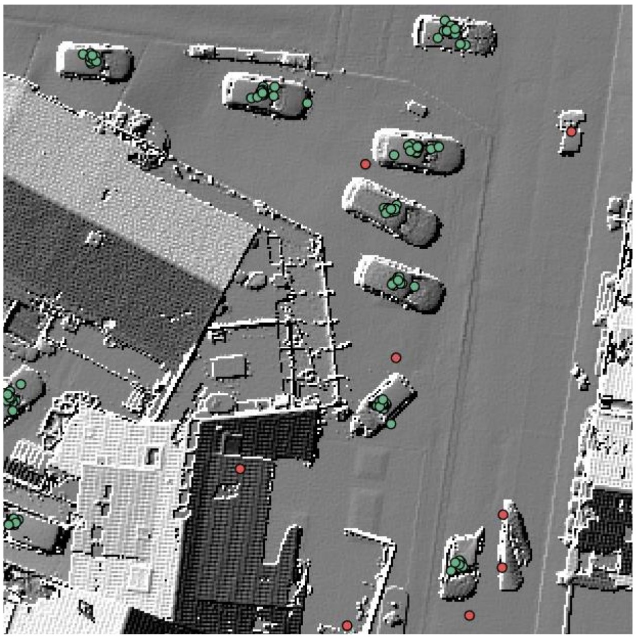


Figure 4. Collected centres (*x*- and *y*-position) of the vehicles for an exemplary subset of the test area. Outliers detected with DBSCAN are marked with red colour

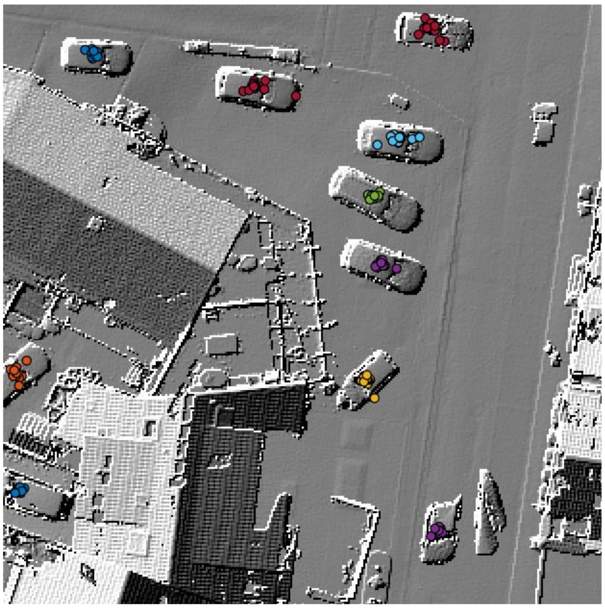


Figure 5. Remaining centres (*x*- and *y*-position) of the vehicles for an exemplary subset of the test area are subdivided with DBSCAN into clusters that are marked with different colours

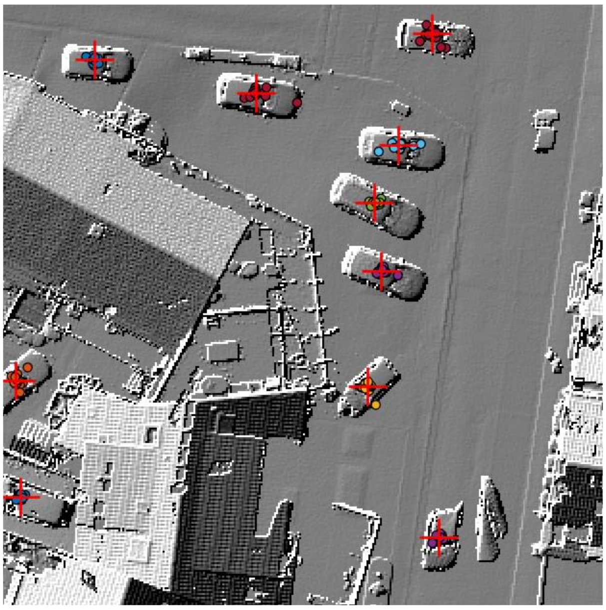


Figure 6. Result of integration for an exemplary subset of the test area

4.4 Quality analysis

The centres of the clusters represent the centres of the vehicles. The geometric quality of these centres is of minor importance since we only require approximated values as input for the following step. For the quality evaluation of the coarse positioning, we are mainly interested in the completeness and correctness. A common approach is to subdivide all collected data into the categories: *True Positive TP*, *False Negative FN*, *False Positive FP* and *True Negative TN* (compare Heipke et al. 1997). From that, we can calculate *Completeness* = *TP* / (*TP* + *FN*), *Correctness* = *TP* / (*TP* + *FP*) and *Quality* = *TP* / (*TP* + *FP* + *FN*) where *TP*, *FP* and *FN* are the numbers of vehicles of each category.

Table 1 shows a comparison between these quality measures averaged over the ten data collections and the quality measures after the integration. The average quality measures describe the quality that we can expect when we collect the data only once. It can be seen that multiple data collection with subsequent integration significantly improves the results - especially the completeness.

	Average over 10 data collections	After integration
Completeness	79.9 %	93.3 %
Correctness	94.2 %	96.9 %
Quality	73.5 %	90.7 %

Table 1. Quality evaluation of the coarse positioning of vehicles

5. APPROXIMATION WITH MBBS

The coarse positions are used for a second crowd campaign in which each vehicle is to be approximated with a MBB that is defined by 9 parameters: box dimensions *a*, *b*, c, position of the centre point *x*, *y*, *z*, and orientation *α*, *β*, *γ*. An example of a vehicle approximated with such a MBB is shown in Figure 7.

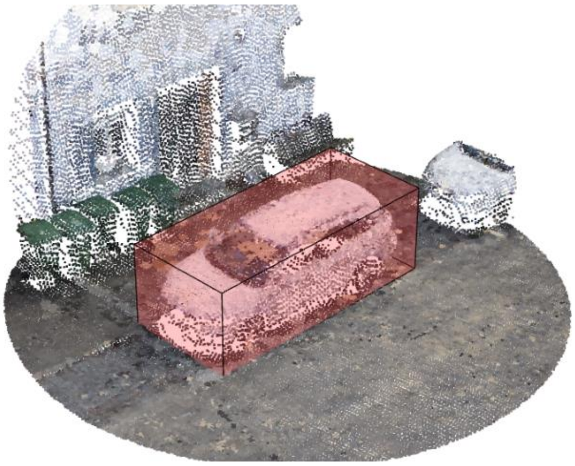


Figure 7. Example of a vehicle approximated with a MBB

5.1 Graphical User Interface (GUI)

The GUI for the crowd-based collection of MBBs is shown in Figure 8. The GUI is subdivided into four parts:

- Main data view (A): Shows a circular area of the 3D point cloud around the coarse positon of the vehicle and the MBB. The point cloud can be rotated and zoomed by mouse controls. The MBB can be moved and scaled in *x*, *y* and *z*.
- Three sub-views (B): Show the 3D point cloud and the MBB as 2D projections in *x*-, *y*- and *z*-direction. It is possible to change the width and the height of the two-dimesional projection of the MBB.
- A button area (C): Contains six rotation buttons for rotating the MBB, a 'reset view' button, a 'reset model' button and a 'help' button.
- A submit button (D): Used to submit the result and switch to the next vehicle until all vehicles of a job are processed.

5.2 Quality control on task design

In order to control and improve the quality during the data collection, we implemented three methods:

Qualification Test: Each crowdworker must first collect a MBB of a reference vehicle. We calculate the position difference Δ_P and the scale difference Δ_S:

$$\Delta_P = \sqrt{(x_R - x_C)^2 + (y_R - y_C)^2 + (z_R - z_C)^2} \tag{1.1}$$

$$\Delta_S = \sqrt{(a_R - a_C)^2 + (b_R - b_C)^2 + (c_R - c_C)^2} \tag{1.2}$$

with: x_R, y_R, z_R, a_R, b_R, c_R are the parameters of the reference MBB and x_C, y_C, z_C, a_C, b_C, c_C are the parameters of the MBB collected by a crowdworker. The maximum difference allowed is 0.4 m for both parameters. Each crowdworker can try maximum five times to reach this accuracy. Crowdworkers, who fail more than five times, are rejected.

Plausibility Control: We check that all parameters of the initial MBB (size: 1m x 1m x 1m) have been changed in order to avoid that crowdworkers just click on the submit button without collecting a vehicle.

Bonus payment: Each crowd job contains one point cloud with a reference vehicle. The crowdworkers get a bonus payment of \$0.05 if Δ_P and Δ_S of the reference vehicle are smaller than 0.1 m. The crowdworkers do not know which of the point clouds contains the reference vehicle.

5.3 Crowdsourcing campaign

Each crowd job contains five vehicles plus one reference vehicle (see section 5.2) for which the MBBs have to be collected. Each crowd job was duplicated 10 times. The number of vehicles is 116. The number of crowd jobs is: *ceil*(116/5) * 10 = 240.

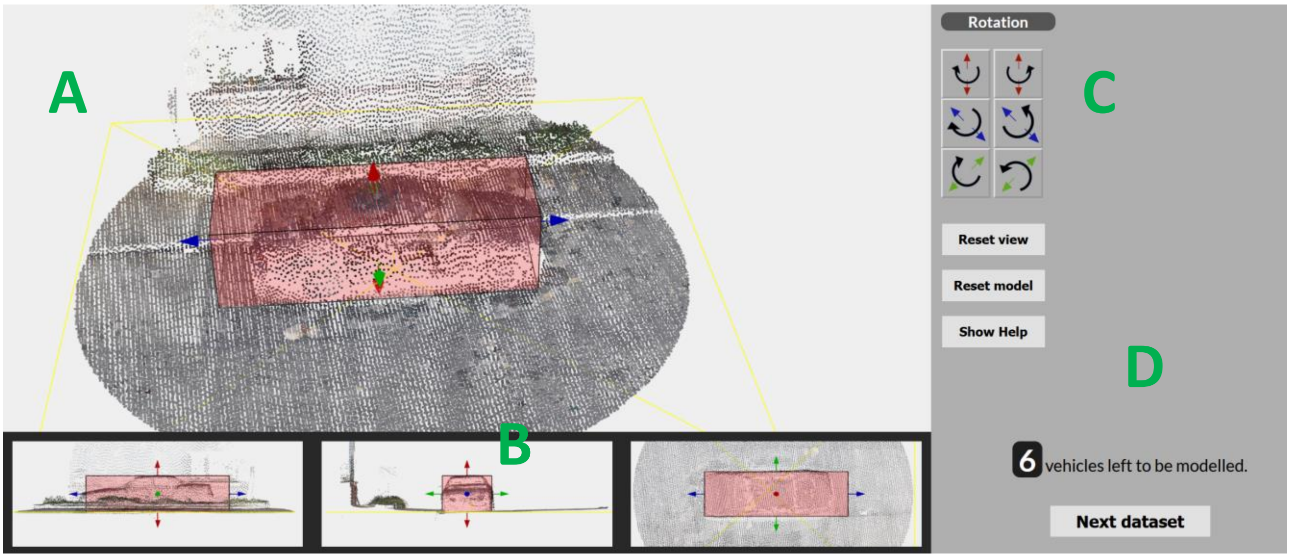
For each crowd job we paid \$0.20 plus a bonus of \$0.05 that was paid out to 32 crowdworkers. The total cost is 240 * \$0.20 + 32 * \$0.05 = \$49.60. The average working time for one job is 11.8 minutes. The average cost per vehicle is \$0.42.

5.4 Data integration

Before the data integration, we first identify the outliers with two subsequent DBSCAN runs. The input for the first DBSCAN are the centre points x_k, y_k, z_k and for the second DBSCAN the side lengths a_k, b_k, c_k of the MBBs for each vehicle. Based on empirical tests we defined *MinPts* = 4 for both DBSCANs and *Epsilon* = 0.5 m for the first DBSCAN and *Epsilon* = 1.5 m for the second DBSCAN. If a MBB is identified as outlier in one of the DBSCANs, it is removed from the integration. The final MBB is calculated by averaging. Table 2 shows the number of outliers. Figure 9 shows the outlier detection and the integration on an example.

Number MBBs	Outliers centre points	Outliers box dimensions	Outliers centre points or box dimensions
1160	154 (13.3%)	135 (11.6%)	214 (18.4%)

Table 2. Number of outliers detected by two DBSCANs



6 vehicles left to be modelled.

Figure 8. GUI for the crowd-based collection of MBB

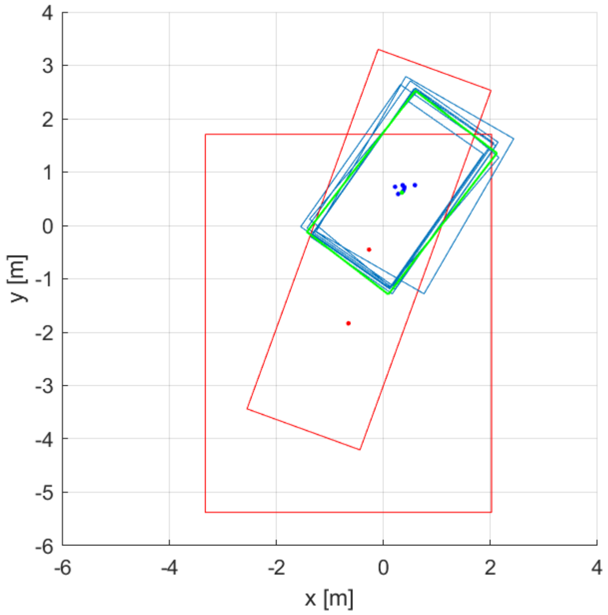


Figure 9. Example of outlier detection and integration of the MBBs (2D-view from above, red: outliers, blue: remaining MBBs, green integrated MBB)

5.5 Quality analysis

We evaluate the geometric quality of the MBBs with two quality indicators: (1) the position difference Δ_P between crowdsourced and reference MBBs and the volume ratio V_c/V_R of crowdsourced to reference MBBs.

Figure 10 shows the position difference Δ_P before and after outlier detection and integration. It can be seen that more than 50% of all crowdsourced MBBs have a Δ_P smaller than 1 m, which shows that most of the crowdworkers work very accurate. However, there is a significant amount of MBBs with a Δ_P up to 6 m. The outlier detection and integration clearly improves the results: 95% of all integrated MBBs have a Δ_P smaller than 1 m. The maximum Δ_P is less than 3 m.

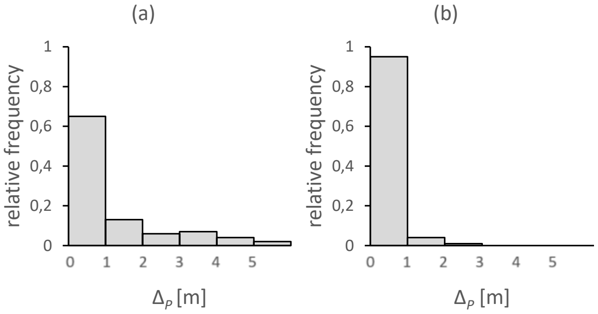


Figure 10. Position difference Δ_P between crowdsourced and reference MBBs: (a) before outlier detection and integration, (b) after outlier detection and integration

Figure 11 shows the volume ratio V_c/V_R of crowdsourced to reference MBB before and after outlier detection and integration. A ratio of 1 indicates that the volumes of the crowdsourced MBB and the reference MBB are identical. It is interesting to see that the majority of the ratios are larger than 1. The reason for this is that crowdworkers tend to collect oversized MBBs (i.e. excessive box dimensions are chosen), because it is easy to see that some points of a vehicle are outside an MBB, but it is difficult to position an MBB exactly so that there is absolutely no empty space between the points and the MBB. Again, it can be seen that the outlier detection and integration significantly improves the quality of the results.

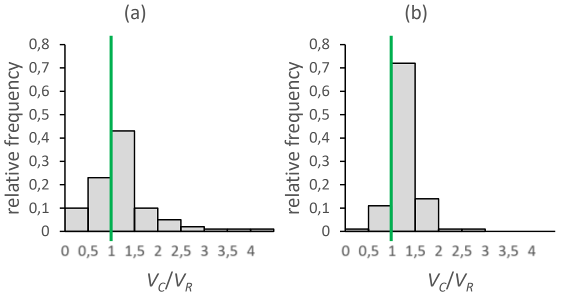


Figure 11. Volume ratio V_c/V_R of crowdsourced to reference MBBs: (a) before outlier detection and integration, (b) after outlier detection and integration

6. DISCUSSION

The problem with geospatial data collection by paid crowdworkers is that very heterogeneous results are to be expected. Even when experienced experts collect geospatial data, the results can be very heterogeneous due to the subjective nature of geospatial data collection (Walter and Soergel 2018). When non-experts collect geospatial data, this effect is even stronger because individuals with completely different backgrounds work together (Senaratne et al. 2017). While the majority of the workers try to solve the tasks as well as possible, a significant percentage of crowdworkers only produce data with poor quality. Without quality control, it is not possible to infer high quality data.

At the example of crowd-based collection of vehicles from 3D point clouds, we have shown that outliers can automatically be detected by multiple data collection and subsequent averaging, and that a high degree of completeness and geometric quality can be achieved. With a tenfold acquisition, we have achieved a completeness of 93.3 percent and a geometric deviation of less than 1 m for 95 percent of the collected vehicles. The average cost per vehicle are $0.08 for the coarse positioning plus $0.42 for the approximation with a MBB. The average total cost is $0.50 per vehicle.

Figure 12 shows the obtained results on a section of the test data. Although the described method was developed especially for the collection of vehicles, it can also be adapted to the collection of other object classes from 3D point clouds.

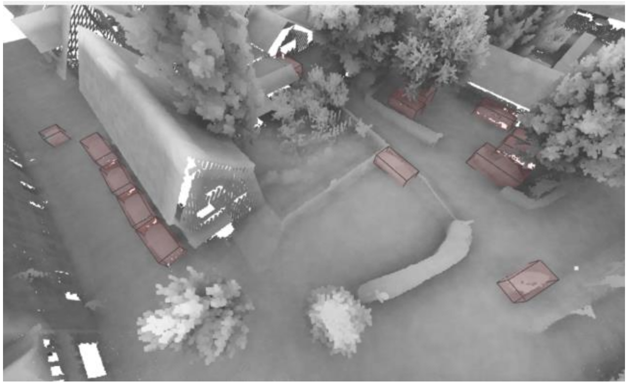


Figure 12. Result of crowd-based vehicle collection projected into the original point cloud

The number of how often the data is collected influences the quality of the results and the amount of cost. Higher quality can be achieved by a higher number of multiple collections, but this also leads to higher cost. If the cost are to be minimized, the number of multiple collections can be reduced, but this has a negative effect on the quality.

In the end, the optimal number of multiple collections is a compromise between data quality and cost. In our future work, we will investigate this relationship in more detail to gain a better understanding of how the number of multiple collections affects the results and how we can optimize this process. We will also investigate the limits of this process and the maximum quality that can be achieved.

REFERENCES

Budhathoki R., Haythornthwaite C., 2012. Motivation for Open Collaboration: Crowd and Community Models and the Case of OpenStreetMap. *American Behavioral Scientist*, 57(5), 548-575. doi:10.1177/0002764212469364

Cramer, M., Haala, N., Laupheimer, D., Mandlburger, G., Havel, P., 2018. Ultra-high precision UAV-based LiDAR and Dense Image Matching. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-1, 115-120. doi:10.5194/isprs-archives-XLII-1-115-2018

Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., Allahbakhsh, M., 2018. Quality Control in Crowdsourcing. A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Comput. Surv*., 51(1), Article 7, 40 pages. doi:10.1145/3148148

Dubey, A, Abhinav, K., Taneja, S., Virdi, G., Dwarakanath, A., Kass, A., Suma Kuriakose, M.S., 2016. Dynamics of Software Development Crowdsourcing. IEEE 11th International Conference on Global Software Engineering (ICGSE), Irvine, CA, 49-58. doi:10.1109/ICGSE.2016.13

Ester, M., Kriegel, H.P., Sander, J., Xu, X, 1996. A density-based algorithm for discovering cluster in large spatial databases with noise. In: Evangelos Simoudis, Jiawei Han, Usama M. Fayyad (Eds.). Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, 1996, 226-231. doi:10.1023/a:1009745219419

Galton, F., 1907. Vox Populi (the wisdom of the crowds). *Nature* 75, 450-451. doi:10.1038/075450a0

Goodchild, M.F., 2007. Citizens as sensors. The world of volunteered geography. *GeoJournal*, 69(4), 211–221. doi:10.1007/s10708-007-9111-y

Haala, N., Kölle, M., Cramer, M., Laupheimer, D., Mandlburger, G., Glira, P., 2020. Hybrid Georeferencing, Enhancement and Classification of Ultra-High Resolution UAV Lidar and Image Point Clouds for Monitoring Applications. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-2-2020, 727–734. doi:10.5194/isprs-annals-V-2-2020-727-2020

Haklay, M., Weber, P., 2008. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4), 12–18. doi:10.1109/MPRV.2008.80

Haralabopoulos G., Wagner C., McAuley D., Anagnostopoulos I., 2019. Paid Crowdsourcing, Low Income Contributors, and Subjectivity. In: MacIntyre J., Maglogiannis I., Iliadis L., Pimenidis E. (eds.). Artificial Intelligence Applications and Innovations. AIAI 2019, IFIP Advances in Information and Communication Technology, Vol 560, Springer. doi:10.1007/978-3-030-19909-8_20

Heipke, C., Mayer, H.M. Wiedemann, C., Jamet, O, 1997. Evaluation of Automatic Road Extraction. International Archives of Photogrammetry and Remote Sensing 32, 47-56.

Herfort, B., Hoefle, B., Klonner, C., 2018. 3D micro-mapping: Towards assessing the quality of crowdsourcing to support 3D point cloud analysis. *ISPRS Journal of Photogrammetry and Remote Sensing,* 137, 73-83. doi: 10.1016/j.isprsjprs. 2018.01.009

Hirth, M., Hoßfeld, T., Tran-Gia, P. 2011. Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com. Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Seoul, 322-329. doi: 10.1109/IMIS.2011.89

Hitlin, P. 2016. Crowdsourcing Age, a Case Study. Pew Research Center. July 2016. Available at: https://www.pewresearch.org/internet/2016/07/11/research-in-the-crowdsourcing-age-a-case-study/

Howe, J., 2006. The rise of crowdsourcing. *Wired Magazine,* 14(6), 1–4.

Ipeirotis, P.G., 2010. Analyzing the Amazon Mechanical Turk marketplace. XRDS: Crossroads, The ACM Magazine for Students, December 2010. doi:10.1145/1869086.1869094

Juhász, L., Hochmair, H.H., 2018: OSM Data Import as an Outreach Tool to Trigger Community Growth? A Case Study in Miami. *ISPRS Int. J. Geo-Inf.,* 7(113). doi:10.3390/ijgi7030113

Kölle, M., Walter, V., Schmohl, S., Soergel, U., 2020. Hybrid acquisition of high quality training data for semantic segmentation of 3D point clouds using crowd-based active learning. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-2-2020, 501–508. doi:10.5194/isprs-annals-V-2-2020-501-2020

Leibovici, D.G., Rosser, J.F., Hodges, C., Evans, B., Jackson, M.J., Higgins, C.I., 2017. On Data Quality Assurance and Conflation Entanglement in Crowdsourcing for Environmental Studies. *ISPRS International Journal for Geo-Information,* 6(3), 78 pages. doi:10.3390/ijgi6030078

Liu, Z., Shabani, S., Glassey, N., Sokhn, M., Cretton, F., 2018. How to motivate participation and improve quality of crowdsourcing when building accessibility maps. 3th International Workshop on Accessible Devices and Services in the 15th Conference of IEEE Consumer Communications & Networking Conference (CCNC), Las Vegas, USA, 6 pages. doi:10.1109/CCNC.2018.8319237

Mao, A., Kamar, E., Chen, Y., Horvitz, E., Schwamb, M.E., Lintott, C.J., Smith, A.M., 2013. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. Proc. 1st AAAI Conf. Human Computat. Crowdsourcing, 1-9.

Niu, X.J., Qin, S.F., Vines, J., Wong, R., Lu, H., 2019. Key Crowdsourcing Technologies for Product Design and Development. *International Journal of Automation and Computing*, 1751-8520. doi:10.1007/s11633-018-1138-7

Pfeifer, N., Mandlburger, J., Otepka, J., Karel, W., 2014. OPALS - A framework for Airborne Laser Scanning data analysis. *Computers, Environment and Urban Systems,* 45, 125-136. doi:10.1016/j.compenvurbsys.2013.11.002

Riegl, 2018. RIEGL VUX-1UAV product specifications, http://www.riegl.com/uploads/tx_pxpriegldownloads/RIEGL_VUX-1UAV_Datasheet_2017-09-01_01.pdf

Senaratne, H., Mobasheri, A., Ali, A.L., Capineri, C., Haklay, M., 2017. A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, 31(1), 139-167. doi: 10.1080/13658816.2016.1189556

Sheehan, K.B., 2018. Crowdsourcing research: Data collection with Amazon's Mechanical Turk. *Communication Monographs* 25, 140-156. doi: 10.1080/03637751.2017.1342043

Simons, A., 2004. Many wrongs: The advantage of group navigation. *Trends in Ecology and Evolution*, 19, 453-458. doi:0.1016/j.tree.2004.07.001

Surowiecki, J., 2004. The Wisdom of Crowd – why many are smarter than the few and how collective wisdom shapes business, economics, societies and nations. Doubleday, New York. doi: 10.1111/j.1744-6570.2006.00060_10.x

Vaughan, J.W., 2017. Making Better Use of the Crowd. How Crowdsourcing Can Advance Machine Learning Research. *Journal of Machine Learning Research*, 18(193). 1-46.

Walter, V., Soergel, U., 2018. Implementation, Results and Problems of paid Crowd-based geospatial Data Collection. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 86(3-4), 87-197. doi:10.1007/s41064-018-0058-z

Walter, V., Kölle, M., Yin, Y., 2020. Evaluation and optimisation of crowd-based collection of trees from 3D point clouds. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-4-2020, 49–56. doi:10.5194/isprs-annals-V-4-2020-49-2020

Zhang, J., Wu, X., Sheng, V., 2016. Learning from crowdsourced labelled data. A survey. *Artif. Intell. Rev.* , 46(4), 543-576. doi:10.1007/s10462-016-9491-9

Zheng, Y., Li, G., Yuanbing. L., Shan, C., Cheng. R., 2017. Truth inference in crowdsourcing: is the problem solved? Proceedings of the VLDB Endowment. doi:10.14778/3055540.3055547

Zhou, D., Platt, J.C., Basu, S., Mao, Y., 2012. Learning from the Wisdom of Crowds by Minimax Entropy. *Advances in Neural Information Processing Systems (NIPS)*, 25, 2204-2212.