

FROM MULTIPLE POLYGONS TO SINGLE GEOMETRY: OPTIMIZATION OF POLYGON INTEGRATION FOR CROWDSOURCED DATA

D. Collmar*, V. Walter, M. Kölle, U. Sörgel

Institute for Photogrammetry (ifp), University of Stuttgart, Germany
Geschwister-Scholl-Strasse 24D, 70174 Stuttgart
{david.collmar, volker.walter, michael.koelle, uwe.soergel}@ifp.uni-stuttgart.de

KEY WORDS: Polygon Integration, Crowdsourcing, Raster Conversion, Data Filtering, Wisdom of the Crowd

ABSTRACT:

Paid crowdsourcing is a popular approach for creating training data in machine learning, but output quality can suffer from various drawbacks, such as noisy data. One solution is to obtain multiple acquisitions of the same dataset and perform integration steps, which can be challenging for geometries such as polygons. In this paper, we propose a raster-based polygon integration approach for the use of crowdsourced data, providing a solution for integrating multiple geometric shapes into single geometries. We analyze the effects of the choice of the integration threshold parameter for different sample sizes on the quality measures intersection over union (IoU) and Hausdorff distance, and provide a recommendation for its optimal selection based on empirical analysis. Additionally, further possibilities to improve integration results are explored, i.e., methods of filtering data before integration by outlier detection.

1. INTRODUCTION

1.1 Crowdsourcing

Training data are essential for the performance of machine learning (ML) methods such as Convolution Neural Networks (CNNs). Since low-quality training data cannot be compensated by even the best ML algorithms (Whang and Lee, 2020), the demand of valid training data is enormous (Stonebraker and Rezig, 2019). One method to collect training data is paid crowdsourcing, which is becoming increasingly popular in the field of remote sensing (Saralioglu and Gungor, 2020).

Crowdsourcing, a neologism of “crowd” and “outsourcing”, describes the practice of using collective intelligence in order to solve tasks (Howe, 2006), i.e., outsourcing a task to a (unknown) group of people, the crowdworkers. Crowdworkers are motivated by different factors, such as intrinsic motivations, often seen in voluntary crowdsourcing, or extrinsic motivations, such as monetary rewards (Hossain, 2012). Although offering payments may attract a large number of workers, it may also affect data quality and lead to inhomogeneous results due to the primarily extrinsic nature of motivation (Chandler et al., 2013).

Zhang et al. (2016) recommend improving the quality of crowdsourced data through “quality control on task design” or “quality improvement after data collection”. A common idea for improvement when using the second method is to have the same data set processed by different crowdworkers (Zhang et al., 2016). While coming at higher costs, this enables two methods: Firstly, it allows one to identify outliers without a necessary existence of ground truth data (Walter and Sörgel, 2018). Secondly, the multiple acquired data can be integrated, making it possible to eliminate or mitigate the influence of those outliers that were previously difficult to filter out (Zhang et al., 2016).

A simple but effective method for integrating multiple collected data is majority voting, where in a binary case, as long as more than half of the annotators provide correct results, the integrated result will equal the correct value (Zhang et al., 2016).

This follows the principle of “Wisdom of the Crowd”, which was already described in 1785 in Condorcet’s jury theorem: If the probability of a worker performing a task correctly is higher than 50%, then the theorem states that as the number of workers n increases, the collective competence will gradually approach the maximum possible value (Condorcet, 1785). Or, in other words: “[...] the output of the crowd can be greater than the sum of its parts” (Chandler et al., 2013).

1.2 Polygon integration

While majority voting can be a reliable method for the integration of multiple collected labels, the application may only seem obvious for simple cases. For binary problems, which involve only two possible options, an integrated solution can be obtained by choosing the option with the absolute majority. In cases with more than two possible solutions, but with only a limited number of solutions to choose from, such as classifications, the same approach can be used. Here, a simple majority instead of an absolute majority is sufficient.

When handling problems with an infinite or unknown number of solutions, an integration can no longer be performed by majority voting, and therefore other means like averaging through mean or median are resorted to. For more complicated data structures, such as multiple polygons with differing number of points, no possibility to apply a majority vote in the vector domain is known to the authors. Instead, (Walter 2018) provides a solution for the integration of multiple polygons using a conversion to raster data and a subsequent integration in raster domain.

This strategy appears to be a promising option for a large-scale generation of training data as needs to be performed for typical ML applications: Simple implementation, fast computation and results of high quality make this approach look attractive. However, the practical application of this integration approach has been tested with a rather small sample size, but not under crowdsourcing-typical circumstances, such as noisy input data due to factors like lack of motivation of crowdworkers (Chandler et al., 2013). Further processing steps such as smoothing operations are performed, meaning multiple input parameters are required.

* Corresponding author

Our goal is to adapt and modify this raster integration method for application in paid crowdsourcing, while minimizing the amount of input parameters and exploring different optimization methods to enhance the results. The optimization is performed by using a real-world dataset with a crowdsourcing-typical application: The acquisition of tree crown outlines in airborne images using polygons.

The rest of the paper is organized as follows: In section 2, the used dataset is described. Section 3 presents the integration method in detail, while section 4 describes the optimization of the threshold parameter. Section 5 analyzes the influence of data filtering before the integration, section 6 draws a conclusion, before section 7 closes the publication with possible future work.

2. DATASET

The dataset used in this study is a large-area orthomosaic covering cherry orchards. We extracted 115 image sections, each containing a single tree, with the aim of obtaining their outlines. Each tree geometry was acquired by 150 different workers on the crowdsourcing platform “microWorkers.com” by means of polygons, resulting in a total of 17,250 polygons. Figure 1 displays one of the 115 image sections, including a total of 150 crowd-sourced polygons. Crowdsworkers were paid \$0.10 for 5 acquisitions, resulting in a cost of \$0.02 per tree outline, \$3 per image section or \$345 in total. For quality evaluations, reference data collected by experts was used.

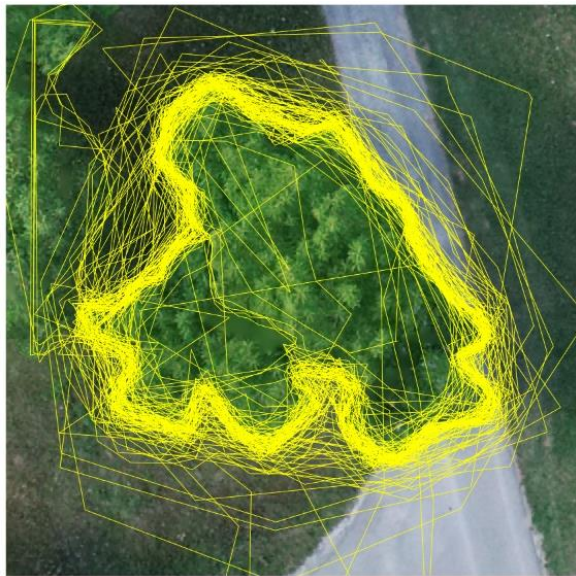


Figure 1. Image section with crowd-sourced tree outlines shown in yellow, collected by 150 different crowdsworkers.

3. INTEGRATION METHOD

The approach of (Walter, 2018) converts input vector data into raster data, processes and integrates them in the raster domain, and then converts the integrated results back to vector data. Our adaption is similar to this procedure: (vector) input data are first converted into the raster domain, followed by a pixel-wise binarization. The cell size is set to match the pixel size, which determines the output resolution, i.e., the pixel size of the integrated shape. We are focusing mainly on the binarization and

the respective threshold value. In that case, the binarization threshold becomes the threshold for a binary vote on pixel-level, and thereby determines whether a pixel is included in the integrated polygon. This enables a data integration by majority voting.

Figure 2 shows an exemplary pixel-level integration of 15 input polygons with a chosen threshold value $t_{15}=8$, including every pixel with 8 or more votes in the integrated shape. The numbers in Figure 2a indicate the number of votes per pixel, i.e., the amount of polygon shapes which included those pixels. While Figure 2a visualizes the majority voting, Figure 2b shows the integrated polygon shape visualized as a skeleton line.

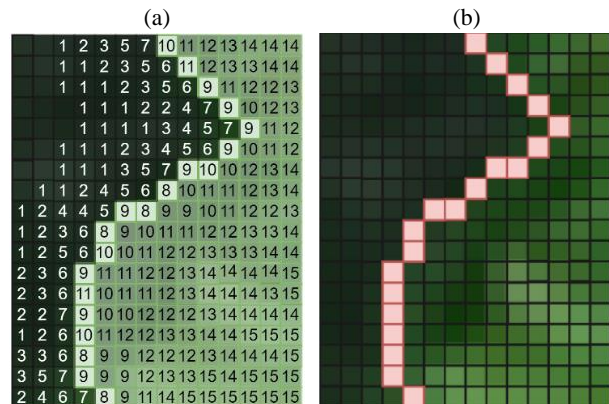


Figure 2. Exemplary pixel-level integration for 15 input polygons. (a) Visualization of majority voting. (b) Integrated polygon shape as skeleton line.

As can be seen from Figure 2, the choice of the threshold parameter has a direct influence on the shape and therefore the geometric accuracy of the integrated polygon. Increasing the threshold in Figure 2a would lead to some pixels being not included in the integrated polygon, while lowering the threshold would lead to the opposite, a larger polygon.

Both these cases can be seen in Figure 3, which illustrates the importance of the correct choice of the threshold value on the example of tree outlines: In Figure 3a, a too low threshold was chosen, allowing noisy acquisitions to gain too much influence, and therefore resulting in a too large integrated polygon. In Figure 3c, the threshold was set too high, resulting in a decrease in the number of pixels that remained after integration, and therefore in a too small integrated polygon. Both cases, i.e., too small or too large thresholds, have a large impact on the resulting polygons and therefore on data quality and should be avoided. The integrated results shown in Figure 3b seem to be a good fit and include most pixels containing the central tree. The choice of the threshold parameter for Figure 3b was not obvious for the authors and was set manually by “trial and error”.

To overcome the issue of choosing a correct threshold parameter, we are aiming for a general approach that is independent of the exact number of annotations or acquisitions. We will study the behavior of the threshold parameter t_n for different numbers of observations n .

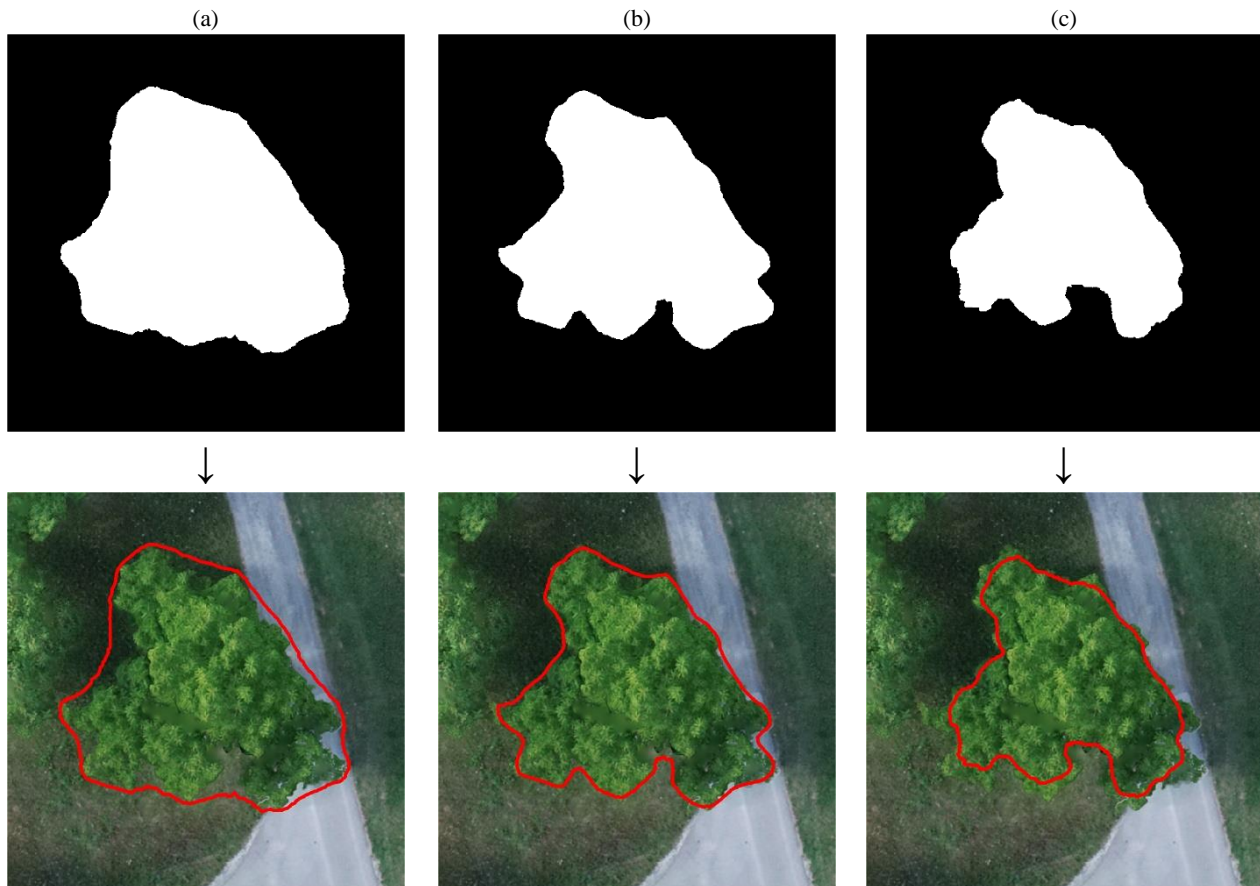


Figure 3. Visualization of the results of a pixel-based integration of tree outlines using different threshold values (a) - (c), using increasingly higher thresholds. Top row: Binary masks. Bottom row: Derived integrated polygons.

4. INTEGRATION THRESHOLD

4.1 Quality measures

In order to determine the optimal threshold value t_n , it is necessary to measure the quality of the integrated results for evaluation purposes. We are using the Jaccard index or intersection over union (IoU) parameter to evaluate the integrated results. The IoU calculates the similarity of two polygons on a scale from 0 to 1, which we use to compare the integrated polygons to our ground truth polygons. A score of 1 would be a perfect fit, while a score of 0 would denote that the polygons do not overlap (Jaccard, 1901). Minor local errors or single outliers may cause only slight changes in IoU, and may therefore be hard to detect. In order to be able to detect such inaccuracies, we add the Hausdorff distance as metric of quality. The Hausdorff distance can be used to measure the distance between two sets of points, or polygons, allowing to accurately detect local outliers (Hausdorff, 1914). A Hausdorff distance of zero describes two equal sets, indicating a perfect match. The higher the Hausdorff distance value, the lower the similarity between the two sets.

Therefore, both quality measures, namely intersection over union and Hausdorff distance, will be taken into consideration when trying to find an optimal threshold for the majority vote of the integration.

4.2 Optimization & results

By performing an integration for any number of observations n , the result will always be one single polygon. For the trivial case $n=1$, the input polygon equals the integrated polygon. For all other cases, the choice of $t_n=1$ describes the union of all input polygons, resulting in a large, inflated polygon shape. Setting $t_n=n$ equals the intersection of all input polygons, resulting in only a tiny fraction of the actual solution, or no solution at all (if no pixel was included in all n input polygons).

The dataset consists of 115 image sections, resulting in 115 integrated polygons. For each of those polygons, single IoU values and Hausdorff distances can be calculated. For easier comparison and simpler representation, both quality measures are substituted with their mean over all image sections, effectively reducing the 115 values per measure with a single value.

As stated before, our interest lies in finding the optimal solution for t_n , independent of the number of observations n . Therefore, the steps described are performed for all possible values of n , resulting in n values per measure. The optimal threshold t_n can then be determined by minimizing the mean Hausdorff distance (mHd) or maximizing the mean IoU ($mIoU$). Table 1 shows those results for selected n .

n	t_n by IoU		$mIoU$ [%]	t_n by Hd		mHd [px]
10	5	50%	81.31	6	60%	37.93
15	8	53%	82.36	8	53%	36.62
20	10	50%	82.96	11	55%	36.04
25	12	48%	83.07	14	56%	35.37
50	24	48%	83.97	25	50%	34.57
75	36	48%	84.12	40	53%	34.31
100	49	49%	84.30	51	51%	34.07
125	60	48%	84.32	64	53%	33.79
150	72	48%	84.36	78	52%	33.78

Table 1. Mean intersection over union values ($mIoU$), mean Hausdorff distances (mHd) and optimal majority vote thresholds (t_n) after integration for selected n .

In Table 1, n describes the number of observations per image section taken as input for each integration. t_n by IoU and t_n by Hd specify the optimal choice of the threshold parameter t_n , if the mean intersection over union is maximized or the mean Hausdorff distances are minimized. The columns $mIoU$ and mHd show the maximum and minimum values, respectively. Interestingly, an increase in n does result in better $mIoU$ and mHd values for all considered n . While the improvements are more notable for small n , the impact of increasing n appears to significantly reduce for larger n . Apparently, both the $mIoU$ and the mHd values seem to be converging to a saturation point, which is around 84% for $mIoU$ and 33.8 px for mHd .

Figure 4 shows a visualization of the saturation effect: The distances between the curves become smaller for larger n , with only very slight differences for $n=150$. The saturation effect is more noticeable in the case of $mIoU$, as the distance of lines decreases rapidly with increasing n , as can be observed in Figure 4a. In the case of mHd however, while not being as apparent for smaller n values, the lines of $n=125$ and $n=150$ are nearly identical at their minimum (Figure 4b). Therefore, our results indicate that the saturation point has been reached, which can be seen as a convergence limit for increasing n .

Another point that might be noteworthy is that a choice of $n=10$ already delivers very good results with a $mIoU$ of 81% and a mHd of around 38 pixels as can be seen in Table 1, both of which are

already close to the apparent saturation point. These observations might seem surprising, but are consistent with previous research findings: (Walter et al., 2021) suggest that data acquired via paid crowdsourcing can achieve optimal solutions even with a small sample size. Small n do not only deliver good results according to our chosen evaluation metrics but also seem to reliably approximate the optimal threshold t_n , as is indicated by Table 1 and Figure 4.

Refocusing on the original question, i.e., the selection of the optimal threshold parameter independent of the number of observations n : All results listed in Table 1 suggest the choice of t_n to be chosen at around 50 percent of n . When going by IoU, the ideal threshold appears to be slightly below $0.5n$, whereas the Hausdorff distance implies an optimal threshold slightly above $0.5n$. This similarity of results, measured by the two evaluation parameters intersection over union and Hausdorff distances, reinforces the validity of our results and supports the theory of “Wisdom of the Crowd”. The optimal threshold varies depending on the task and the significance of outliers or minor inaccuracies in the dataset, making it impossible to provide a universal answer. Still, the authors suggest a threshold of 50% as a general guideline, which is the middle ground between the results of $mIoU$ and mHd . The processing in the following sections will be performed using this recommended threshold of $0.5n$.

5. FILTERING

5.1 General Methodology

In the previous section, it was shown that opting for a small value of n can already produce results with good IoU values and small Hausdorff distances. Still, increasing the number of observations n seems to produce better outputs until reaching a saturation point, where the results seem to converge and no further improvement can be reached by increasing the sample size. It is also not possible to further optimize the integration by adjusting parameters, since the threshold is the only parameter that can be optimized. However, it is possible to optimize the input data used for the integration, e.g. through prior filtering. A common approach is filtering input data for outliers. Since the existence of

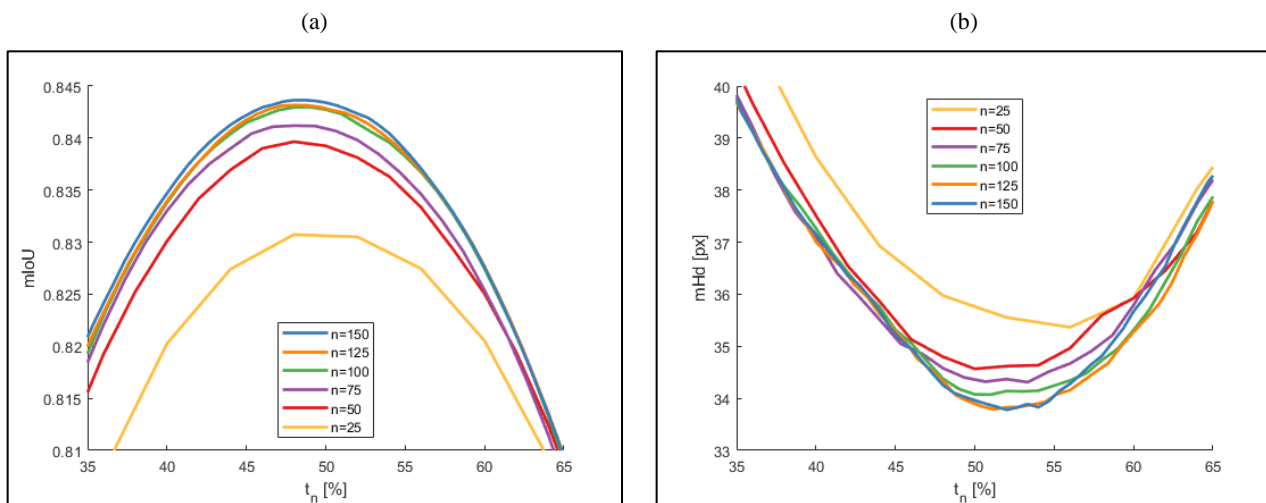


Figure 4. Visualization of quality evaluation parameters for selected n in relation to chosen majority vote thresholds. (a) Mean intersection over union ($mIoU$). (b) Mean Hausdorff distance (mHd).

ground truth data cannot always be assumed, we will highlight filtering methods not relying on the existence of a ground truth. We therefore try to apply the central limit theorem, which states that for a sufficiently large sample size, random variables will follow a normal distribution if certain criteria are met.

Assuming the input data are normally distributed, the geometric properties of normal distributions can be used for simple filtering. This can be done by calculating shape features such as area or circumference of the polygons, and subsequently omitting those polygons, which are the furthest from the center of the distributions. It is worth mentioning that a skew might be added to the normal distributions, depending on which shape features are observed. However, this still allows filtering in a practical way.

Figure 5 visualizes an example for such a skew normal distribution, containing all 150 polygon areas calculated for one of the image sections. The histograms for all other image sections followed similar distributions, validating our assumptions.

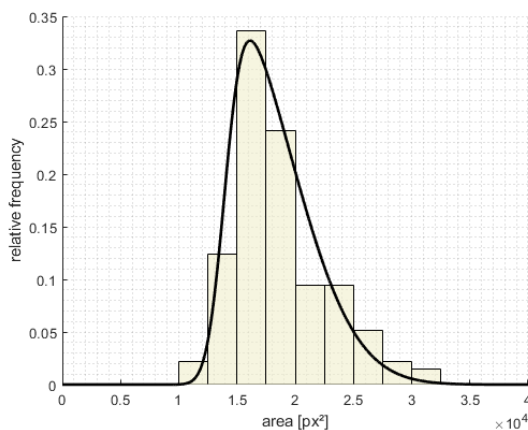


Figure 5. Histogram of 150 calculated polygon areas for one image section and derived skew normal distribution.

Prior to performing filtering, specific shape features must be selected to serve as the basis for the filtering process. There is a wide range of features to describe two-dimensional shapes such as the crowd-acquired polygons. One possibility are moments, which provide a way to measure the spatial distribution of a shape in relation to an axis. They enable the determination of physical properties such as object orientation, eccentricity, area, and centroid, or can be important shape features themselves (Steger, 1996). An example application is in automatic text recognition, where moments can be used to distinguish between similar letters such as I and T, by analyzing the mass distribution of the individual pixels.

Each moment can be calculated for every acquisition and result in a single value, allowing for effective outlier detection. We are using the central moments up to the 4th order for our filtering process, making the following adjustments: The central moment of 0th order, μ_{00} , is equal to the raw moment of 0th order, M_{00} , which describes the surface area of a shape. The central moments of 1st order, μ_{10} and μ_{01} , are 0 per definition for all input data, making filtering impossible. Therefore, the center of gravity can be used instead, which effectively combines the raw moments M_{10} and M_{01} . The three central moments of 2nd order, the four of 3rd order and the five central moments of 4th order can be used as they are.

5.2 Simple filtering

We calculated the central moments μ_{ij} up to the 4th order for 100 acquisitions per image section, using the explicit method proposed by (Steger, 1996). Only $n=100$ acquisitions are used since we assume that the saturation point has been reached around this number, as Figure 4 indicates. Furthermore, we want to draw comparisons between the filtered results and those with a larger sample size, namely the results for $n=150$. We therefore performed an outlier filtering on the polygons acquired by the crowd for each of those moments. For the filtering, we used a quantile value $p \in [0,1]$, which describes the relative number of observations to keep, e.g., $p=0.9$ will keep 90% of observations and sort out the 10% being the furthest from the distribution's center. After filtering, the integration described in the previous sections was performed, taking the observations that survived the filtering process as input data, and using a majority vote threshold of $0.5 \cdot p \cdot n$, following our recommendation of the previous section to use a threshold value of 50%.

Figure 6 shows the quality evaluation parameters $mIoU$ (Figure 6a) and mHd (Figure 6c) of the integrated results after filtering in relation to p , in form of mean values per order over all acquisitions and images. The unfiltered results are visualized as dotted lines for comparison, both for $n=100$ and $n=150$, using the recommended thresholds of $0.5n$. Figure 6b adds the $mIoU$ ranges of all central moments per 2nd, 3rd and 4th order, depicting the mean value per order as a dashed line, while Figure 6d shows those for mHd , respectively.

Filtering by higher order moments seems to have a rather mild effect. While still outperforming the unfiltered results for many p values, the improvements seem rather moderate, especially in the case of $mIoU$. The impact of the lower-order moments on the other hand is significant: Even when 50% of acquisitions are filtered out, both $mIoU$ and mHd improve substantially compared to no filtering. Although the choice of moment to filter obviously has a large impact on the results, filtering by any moment still outperforms no filtering for most cases: Indicated by the lower dotted line in Figure 6a, a choice of p between 75% and 90% leads to an improvement in average $mIoU$ compared to an integration without previous filtering, no matter which moment order was chosen for filtering. For mHd , the improvements are even more evident, as can be seen in Figure 6c: A choice of p between 60% and 95% leads to improved results, no matter which filtering method is picked. In general, the best results can be achieved for a value of p around 70% to 75%, depending on the evaluation parameter and moments considered. A value of p greater than 95% is not recommended, as this leads to very few observations to be filtered out, making the differences to a non-filtering approach marginally small.

While it could be expected that filtering input data outperforms an unfiltered approach for the same sample size, it is interesting to see that filtering pre-integration can even outperform a much larger sample size: The dotted lines in Figure 6a and Figure 6c also show the integrated results for a sample size of $n=150$, which is outperformed by most filtering approaches. Again, a threshold of $0.5 \cdot n \cdot p$ was used for the integrations. Thus, it can be concluded that simple filtering by moments not only enhances both $mIoU$ and mHd , but also outperforms an increased number of observations. As a result, this delivers a simple method to surpass the previously observed saturation point. This allows for integrations of higher quality while reducing the sample size, therefore saving cost and time.

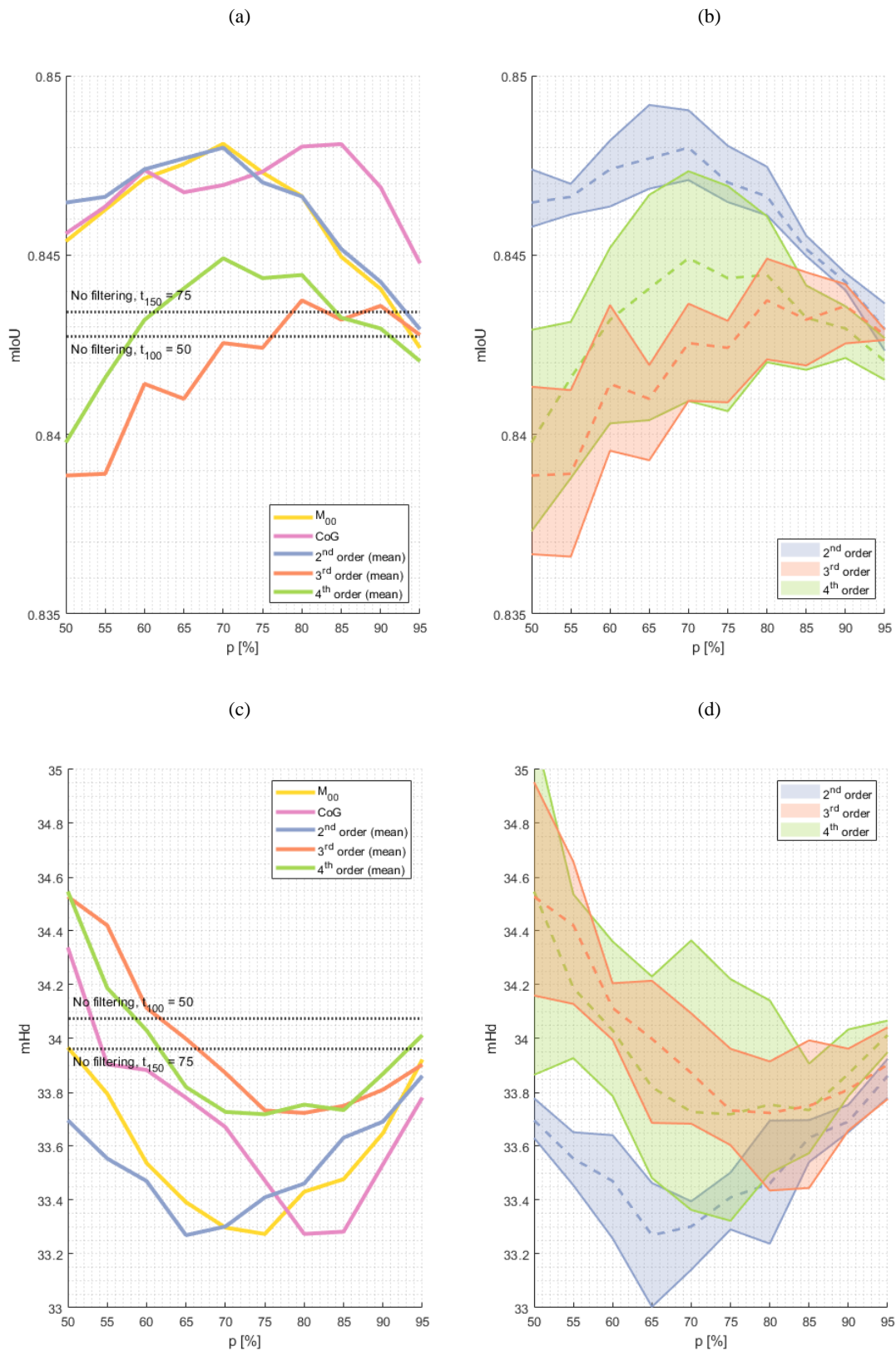


Figure 6. Quality evaluation parameters for integrated results after filtering by quantile parameter p with $t_n=0.5 \cdot p \cdot n$. (a) Mean $mIoU$ values for each order. (b) Range of $mIoU$ for higher order moments, mean values from (a) as dashed line. (c) Mean mHd values for each order. (d) Range of mHd for higher order moments, mean values from (c) as dashed line.

Dotted lines in (a) and (c) depict $mIoU$ or mHd values without filtering for $n=100$ with $t_{100}=50$ and for $n=150$ with $t_{150}=75$.

5.3 Combined filtering

Filtering by different moments before integration appears to have different impacts on the integrated results, as Figure 6 highlights. Still, Figure 6 only visualizes the results of filtering by single moments. What was not considered in the previous section is combined filtering based on multiple filter parameters (i.e., moments), which raises the question, if the results can be improved further by combining the moments with the highest impact. The parameters providing the highest impact are the raw moment M_{00} , i.e., the surface area, the center of gravity (CoG), i.e., a combination of the first order moments, and the central moments of second order. One of those moments of second order, μ_{20} , appeared to have the largest influence on the quality of results, not only for the central moments of 2nd order, but for all filtering parameters considered.

Therefore, we performed a combined filtering of M_{00} , CoG and μ_{20} by connecting the results using a logical AND. This means only acquisitions within the boundaries specified by p for all three parameters remain, i.e., those acquisitions, that are amongst the best p percent for M_{00} , CoG and μ_{20} . If a single acquisition fulfills the constraint for only one or two of the parameters, but not for all three of them, the acquisition is omitted. This leads to a generally smaller pool of acquisitions for the filtering, which can no longer be calculated by $n \cdot p$, as was the case for the simple filtering. Still, further improvements can be reached, as illustrated by Figure 7.

While μ_{20} appears to be the better filtering method for lower p , i.e., for a stricter filtering, the center of gravity appears to perform best for larger p , i.e., a less strict filtering. Additionally, M_{00} seems to provide a compromise between both filtering approaches. Although the three parameters all have strengths of their own, a combined filtering using those parameters appears to combine their strengths, resulting in the best method and peaking with a choice of p around 75% (*mIoU*) or 80% (*mHd*). Also, the approach using combined filtering appears to be superior for all p equal or greater than 55% in comparison to the non-combined parameters. Figure 7 further visualizes the magnitude of improvements compared to the results obtained by performing no previous filtering, i.e., the results for the same sample size (no filtering, $t_{100}=50$), and those for a larger sample size after the saturation point is reached (no filtering, $t_{150}=75$).

There are several implications to this: Using this filtering approach can enhance the integrated data beyond a saturation point, where further increases in sample size provide no additional benefits, or when increasing the sample size is not feasible. Furthermore, even minor improvements achieved through filtering can equal a substantial increase in sample size n . This can allow for the same output quality with a smaller sample size, saving cost and time. Further, this demonstrates that an improvement of quality of integrated data can be accomplished through simple methods, and a combination of filtering parameters can result in an additive effect.

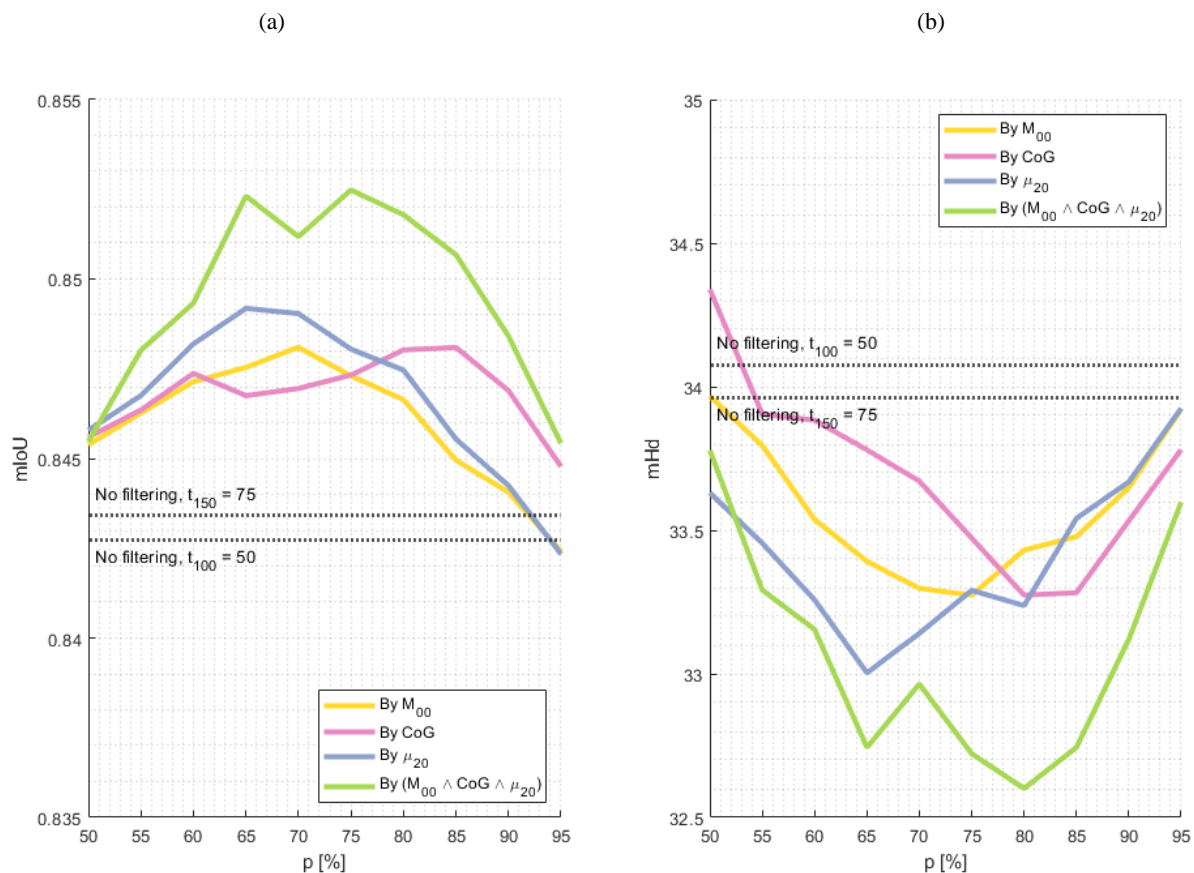


Figure 7. Quality evaluation parameters for integrated results after filtering by M_{00} , CoG and μ_{20} individually and combined. (a) Mean intersection over union (*mIoU*). (b) Mean Hausdorff distance (*mHd*).

6. CONCLUSIONS

We presented the possibility to integrate multiple collected crowdsourced polygons by using a raster-based integration. In a first step, we analyzed and optimized the threshold value for different sample sizes. We gave a general recommendation for the choice of this parameter in the context of crowdsourced data, which allows to boost integration performance with simple means.

Increasing the sample size can lead to improvements in data quality, which can be measured by mean intersection over union and mean Hausdorff distance. However, improvements can only be seen until a saturation point is reached. In a second step, we investigated ways to surpass this saturation point. We were able to show that this limitation can be overcome by filtering the input data by shape features such as their central moments before the integration, resulting in integrated data of higher quality, outperforming larger sample sizes. This allows for smaller sample sizes, saving time and resources on one hand, and can also be used in cases, where an increase of sample size is not feasible. When using a combined filtering approach by filtering multiple shape features at once, their beneficial effects can combine, resulting in even better data after integration.

Both filtering approaches, simple and combined filtering, were processed by performing outlier detections without the need for reference data. This makes our proposed strategy attractive for real-world applications where no ground truth is provided, such as the generation of training data for use in machine learning.

7. FUTURE WORK

While it was shown that filtering datasets before integration provides a benefit, our analysis centered on polygon moments and features derived from those moments. It seems likely that filtering by other parameters can also lead to a certain improvement, however more testing and verification are necessary to determine the different effects. Also, we only focused on a single combination of parameters in the step of combined filtering. It is plausible, that different combinations can lead to even further improvement. Furthermore, only simple and combined filtering were examined. Another possibility would be subsequent filtering, i.e., applying a secondary filtering step to the data once a first filtering has been performed.

We focused on intersection over union (IoU) and Hausdorff distance as measures of quality. Since these cannot be compared directly to each other, both measures had to be evaluated separately. An integrated quality measure could prove to be of high importance for future research, simplifying the evaluation process. Furthermore, this integrated measure could be combined with other quality measures, providing a more comprehensive and precise description of every aspect of a polygon.

8. ACKNOWLEDGEMENT

Partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2120/1 – 390831618.

REFERENCES

- Chandler, J., Paolacci, G., Mueller, P., 2013: Risks and rewards of crowdsourcing marketplaces. *Handbook of human computation*, Springer Verlag, New York, 377-392.
- Condorcet, N., 1785: *Essai sur l'application de l'analyse a la probabilité des décisions rendues 'a la pluralité de voix*. Imprimerie royale, Paris.
- Hausdorff, F., 1914: *Grundzüge der Mengenlehre*. Veit & Comp., Leipzig.
- Hossain, M., 2012: Users' motivation to participate in online crowdsourcing platforms. *2012 International Conference on Innovation Management and Technology Research*, 310-315.
- Howe, J., 2006: The rise of crowdsourcing. *Wired magazine*, 14(6), 1-4.
- Jaccard, P., 1901: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37, 547-579.
- Saralioglu, E., Gungor, O., 2020: Crowdsourcing in remote sensing: A review of applications and future directions. *IEEE Geoscience and Remote Sensing Magazine*, 8(4), 89-110.
- Steger, C., 1996: On the calculation of arbitrary moments of polygons. Technical report FGBV-96-05, Technische Universität München, Germany.
- Stonebraker, M., Rezig, E., 2019: Machine learning and big data: What is important?. *IEEE Data Engineering Bulletin*, 42.4, 3-7.
- Walter, V., 2018: Integration of multiple collected polygons with a raster-based approach. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*. XLII-4, 679-685.
- Walter, V., Sörgel, U., 2018: Implementation, results, and problems of paid crowd-based geospatial data collection. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 86, 187-197.
- Walter, V., Kölle, M., Collmar, D., 2022: Measuring the Wisdom of the Crowd: How Many is Enough?. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 90(3), 269-291.
- Wang, S. E., Lee, J. G., 2020: Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment*, 13(12), 3429-3432.
- Zhang, J., Wu, X., Sheng, V. S., 2016: Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review*, 46(4), 543-576.