FRAMEWORK FOR COMPARING SEGMENTATION ALGORITHMS

George Sithole[a], Langalethu Majola[b]

[a]Geomatics Division, School of Architecture, Planning and Geomatics, University of Cape Town, Private Bag X3, Rondebosch, 7701, South Africa, Email: george.sithole@uct.ac.za
[b]Geomatics Division, School of Architecture, Planning and Geomatics, University of Cape Town, Private Bag X3, Rondebosch, 7701, South Africa, Email: mjllan001@myuct.ac.za

Commission IV/WG IV/7

KEY WORDS: Segmentation, Algorithms, Mapping, Point Clouds

ABSTRACT:

The notion of a ‘Best’ segmentation does not exist. A segmentation algorithm is chosen based on the features it yields, the properties of the segments (point sets) it generates, and the complexity of its algorithm. The segmentation is then assessed based on a variety of metrics such as homogeneity, heterogeneity, fragmentation, etc. Even after an algorithm is chosen its performance is still uncertain because the landscape/scenarios represented in a point cloud have a strong influence on the eventual segmentation. Thus selecting an appropriate segmentation algorithm is a process of trial and error.
Automating the selection of segmentation algorithms and their parameters first requires methods to evaluate segmentations. Three common approaches for evaluating segmentation algorithms are ‘goodness methods’, ‘discrepancy methods’ and ‘benchmarks’. Benchmarks are considered the most comprehensive method of evaluation. This paper shortcomings in current benchmark methods are identified and a framework is proposed that permits both a visual and numerical evaluation of segmentations for different algorithms, algorithm parameters and evaluation metrics. The concept of the framework is demonstrated on a real point cloud. Current results are promising and suggest that it can be used to predict the performance of segmentation algorithms

1. INTRODUCTION

1.1 Introduction

Mass data acquisition techniques that produce a cloud of points for indoor modelling are now widely used. The acquisition techniques used include photogrammetry and laser scanning deployed on mobile or static platforms. The point clouds generated are very large and pre-processing is required before they are in a form useful for modelling or reduction to a wireframe or parameterised model. Pre-processing includes the cleaning, thinning (decimation) and meshing of point clouds.

1.1.1 Segmentation: Often before pre-processing a point cloud has to be segmented, i.e., dividing the point cloud into disjoint point sets (called segments) that have homogeneous geometric properties, such as surface curvature.

Let V be a point cloud, then segmentation is an operation, F, on the point cloud which partitions it into n segments, S1, S2… S_n. Alternatively it can be thought of as a membership function that generates labels for points in the cloud such that points in the same segment receive the same label.

$$F(V) = \{f(v \in \mathbb{R}3) \mid \forall v \ \in V\}$$

Here f is a function, in this paper called a similarity function that generates a label for a point as a function of the neighbourhood of the point. Definitions of the quality of a segmentation vary; however the general definition given by (Hoover et al. 1996) is used here. According to them a segmentation has to satisfy the following conditions:

1. $V = \cup_{i=1}^{n} S_i$
2. $S_i\ is\ spatially\ connected \qquad for\ i = 1, 2, 3 \ldots n$
3. $S_i \cap S_j = \{\emptyset\} \qquad for\ i \neq j$
4. $P(S) = true \qquad for\ i = 1, 2, 3 \ldots n$
5. $P(S_i \cup S_j) = false \qquad for\ i \neq j$

Conditions 1 and 3 respectively state that all segments combined yield the original point cloud and no two segments have points in common. Condition 2 states that spatial connectivity must be maintained after the segmentation, i.e. points that are spatially connected are most likely to be on the same surface thus should fall in the same segment. Condition 4 defines a predicate, P, on S that requires points in a segment lie on or near the same geometric body (plane, surface etc.). Finally Condition 5 states that P applied to the combination of points from any two segments should be false, i.e., points from two or more geometric bodies should not be captured in the same segment.

Segmentation algorithms differ according to (a) the features they yield, (b) their similarity/dissimilarity functions, (c) the parameters used in the functions, and (d) the metrics used to describe their performance.

1.1.2 Segmentation Algorithms: The features that are often sought in a point cloud are, edges, surfaces, and objects. These can be further distinguished based on whether they are regular or irregular, e.g., a straight line vs a curve. This paper restricts itself to surface extracting segmentation algorithms, e.g. connected components, region growing, and clustering. These methods of segmentation will not be discussed further as they are beyond the scope of the paper.

2. THE PROBLEM

Because segmentation is a fit for purpose partitioning of a point cloud into homogeneous point sets (segments) with user defined geometric properties (e.g., surface curvature), the notion of a ‘Best’ segmentation does not exist. Therefore, no segmentation can be said to be better than the other.

However, if segmentation is to be included in a point cloud processing work flow, then (1) the selection of the ideal segmentation algorithm has to be automated, (2) the selection of optimal parameters of the chosen segmentation algorithm has to be automated, and (3) the quality of the result of a segmentation has to be quantified/evaluated to test for fitness for purpose.

The purpose of this paper is to provide a framework capable of answering these three questions. The problem is non-trivial because the optimality of segmentation is specific to the scene and its geometric qualities. For example a high density point cloud of a scene will likely provide a greater number of segments than its low density equivalent. The next section looks at how others have attempted to answer the above questions.

2.1 Previous work

Most of the work done on this subject resides in the image processing and computer vision domain. Nonetheless the concepts are transferrable to point cloud segmentation.

2.1.1 Automated algorithm selection: The selection of ideal segmentation algorithms depends on the features being extracted (e.g., surfaces, edges, etc.,) and the properties of the point cloud (e.g., point density, noise, point attributes, etc.). The selection of an algorithm is thus not as problematic as the selection of algorithm parameters and the evaluation of the performance of the algorithm.

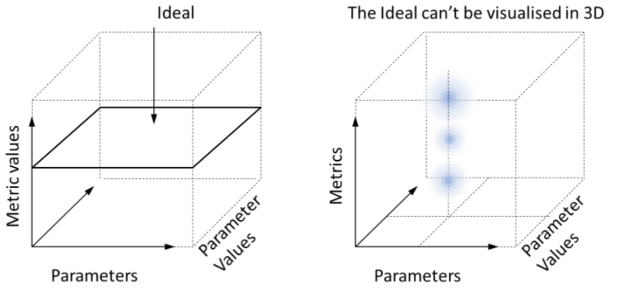
2.1.2 Parameter selection: A common approach for automatically selecting an algorithm's optimal parameter values is to segment a data set for many different values of the algorithm's parameters. For each segmentation the number of segments generated is counted. Typically, small segments i.e., those that contain less than a user defined number of points, are not counted. An assumption is then made that the segmentation that yields the greatest number of segments is the ideal segmentation.

2.1.3 Evaluation of Segmentation Algorithms: Segmentation evaluation methods fall into two general categories, Analytical methods and the Empirical methods.

Analytical evaluation methods: These assess segmentation algorithms by analysing their fundamental principles; this is done without actually implementing these algorithms. In this approach the requirements, utilities and complexities of each algorithm are studied without implementing these algorithms (Zhang 1996). Due to the lack of a general theory for segmentation, the analytical approach can fail to obtain all the required properties of the segmentation algorithm (Zhang 1996). For the purpose of this research analytical methods will not be further considered because they do not adequately account for the variability and complexities of different scenes.

Empirical methods: These methods quantify the performance of a segmentation algorithm by directly applying them to test point clouds and then evaluating their result. Empirical methods fall into two categories: goodness methods, discrepancy methods and benchmarking.

- **Goodness methods**: This approach evaluates the internal quality of the segmentation algorithm. Statistical information within and between segments is evaluated. These methods do not require prior knowledge about the ideal segmentation. These techniques evaluate performance by quantifying homogeneity, heterogeneity or both in the segments. The first measure evaluates the local variance within a segment, ideally a "good" segmentation should maintain high intra-region uniformity (inversely proportional to variance) and the second measure evaluates the variance of the segments with respect to adjacent or all other segments, ideally the segmentation results should have high inter-region heterogeneity. Examples of goodness methods are presented by Levine & Nazif (1985), Liu & Yang (1994), Huang & Dom (1995), and Borsotti et al. (1998). The application of these measures will be discussed in section 3. Figure 1 exemplifies the concept of goodness evaluation methods. An N-dimensional space is defined for parameters, their values and different metrics. Ideal segmentations exist for certain regions of this space. If a segmentation maps into the 'Ideal' regions, it is then considered optimal. The problem with this method is that it assumes that the definition of 'Ideal' applies for all types of scenes.

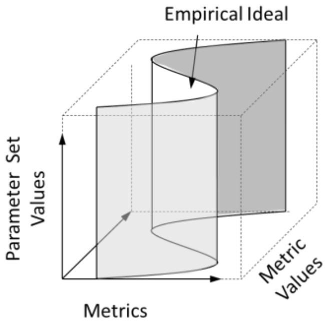


Evaluation of A single Segmentation: A single metric tested against a spectrum of parameters. The result is a histogram

Evaluation of A single Segmentation: Multiple metrics tested against a spectrum of parameters. The result is a scalar field

Figure 1 Goodness Methods, two possible variations. Parameters, their values and metrics are defined in an N-dimensional space. The ideal segmentation exists in particular regions of this space.

- **Discrepancy methods**: This approach evaluates the disparity between computed segments and ground-truth segmentations, with the ground-truth being representative of an "ideal" segmentation. The larger the disparity the lower the performance of the segmentation algorithm. A number of measures are used in these methods for example, number of mis-segmented points, position of mis-segmented points, feature values of segmented regions etc. Examples of discrepancy methods are presented by Yasnoff et al. (1977), Huang & Dom (1995), and Chen et al. (2009).



Evaluation of A single Segmentation:
A single metric tested against the values of multiple parameter sets

Figure 2 Discrepancy methods. Metrics, their values and parameter sets for a single segmentation are defined in an N-dimensional space. The metrics for a segmented reference point cloud (the empirical ideal) is mapped into this space. The parameter sets that yield segmentations that map into the

neighbourhood of the empirical ideal are treated as being optimal.

Figure 2 shows the concept of the discrepancy method. The metrics for a segmented reference point cloud (the empirical ideal) is mapped into an N-dimensional space defined by the Metrics, their values and parameter sets for a single segmentation. Parameter sets that yield segmentations that map into the neighbourhood of the empirical ideal are treated as being optimal. The limitation of this method is that typically a single algorithm is tested against a single metric at a time.

- **Benchmarking**: The results of segmentation can be quantified using a variety of metrics. Typically metrics have to be considered together because a single metric on its own is not enough for an evaluation. For this reason evaluation methods use benchmarks. Benchmarking expands on the principle of empirical discrepancy evaluation, by testing algorithms against a range of 'ideal' segmentations and a range of metrics. Benchmarks are used to rank segmentation algorithms according to performance metrics and the types of scenes being segmented. Examples of benchmarking for segmentation are presented by Hoover et al. (1996), Chen et al. (2009), Estrada & Jepson (2009), Zhao et al. (2011) and Li et al. (2013).

Mersmann et al. (2010) describe the process of benchmarking as follows:

1. Define the problem domain; restrict the domain of the benchmarking process to a specific application, e.g., benchmarking of point cloud segmentation algorithms.
2. Define the algorithms and input parameters to be benchmarked. Also select a set of performance indicators.
3. Select a standard algorithm, method or data set.
4. Lastly, rank the algorithms according to their performance, against the standard algorithm or data set.

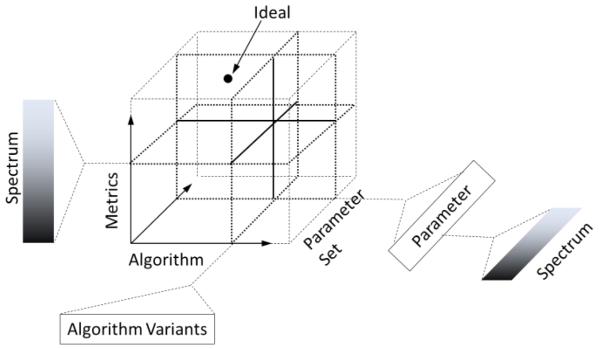


Figure 3 Benchmarking. Many algorithms are compared against many parameter sets, many metrics and reference segmentations (the empirical ideal).

Figure 3 shows the concept of benchmarking. For a single problem domain, many algorithms are compared against many parameter sets and many metrics. The problem here is that because of the multiplicity of algorithms, parameter sets, and metrics, finding the ideal segmentation within this N-dimensional space is not obvious. Moreover, as already mentioned, the benchmark has to be applicable across multiple domains (scene types). In the next section a benchmarking framework is proposed to address this problem.

3. PROPOSED BENCHMARKING FRAMEWORK

The benchmarking framework proposed here is designed with the intention of answering the following question: "Given point clouds of different scenes, what are ideal segmentation algorithms to use and what are their ideal parameters?"

The solution proposed here is to devise a benchmarking frame work for comparing the performance of segmentation algorithms. The framework is composed of three elements (1) a parameter spectrum, (2) a performance metric spectrum, and (3) a set of segmented reference point clouds. The framework provides both a quantitative and visual comparison of segmentation algorithms.

3.1 Elements of the framework

3.1.1 Parameter Spectrum: In a segmentation algorithm, a similarity function is used to test whether two neighbouring points belong to the same segment. If the function yields a value less than a given threshold the points are judged to belong to the same surface. The similarity function is here given by f(v, ϕ), where v is a point in the cloud and $\phi = \{\phi_1, \phi_2, \phi_3, \dots \phi_p\}$ is the parameter set, Domain:$\{\phi \in \Re\ \}$, of the function f. The parameter set, ϕ, contains p number of parameters, ϕ_i. Typical parameters used in segmentation algorithms are:

- **Proximity:** Points that belong to the same surface are often assumed to be close to each other. Proximity is often computed using the Euclidean distance between two points.

- **Curvature**: Curvature is the amount by which the surface of an object deviates from being planar. The curvature at a point is computed using a number of methods, e.g., mean curvature and Gaussian curvature. An example of computing curvature is presented by Rabbani et al. (2006) who estimate surface curvature using the variation of point normals.

- **Colour difference**: Neighbouring points on the same surface are often expected to have the same colour, i.e., the difference in their colours is expected to be small. Colour difference is typically computed as the Euclidean distance between the colour of two points in an RGB colour cube.

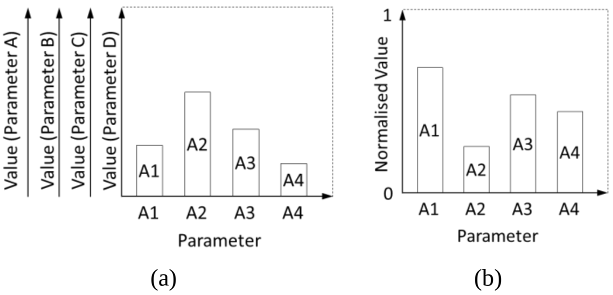


Figure 4 A segmentation done with four parameters A1 to A4 (e.g., curvature, etc). (a) The parameters have different useable ranges. (b) For visual convenience the parameters are normalised against their useable ranges.

Associated with every parameter is a useable range, $\phi_{i.a} < \phi_i < \phi_{i.b}$, i.e., that range for which a parameter yields segments (see Figure 4a). The values $\phi_{i.a}$ and $\phi_{i.b}$ are the lower and upper bounds of this range respectively. For example if curvature is expressed as an absolute angle, then this angle has a range from 0 to π, i.e., $\phi_{i.a} = 0$ and $\phi_{i.b.} = \pi$. Because the parameters have different useable ranges it becomes convenient to normalise them over the useable range, i.e., re-range them between 0 and 1 (see Figure 4b).

The p-dimensional space formed by the p parameters and their normalised ranges now constitute the parameter spectrum. For visual evaluation it is convenient to represent this p dimensional space in 2 dimensions. The approach favoured here is a 100% stacked bar chart of the normalised parameter sets as shown in Figure 4b. An example of this is shown in Figure 5. Here multiple parameter sets of a segmentation algorithm can be shown (in the case of Figure 5 it's four, namely A, B, C and D). The relative size of the parameter values in a parameter set indicate the relative strength of each parameter used for that segmentation.

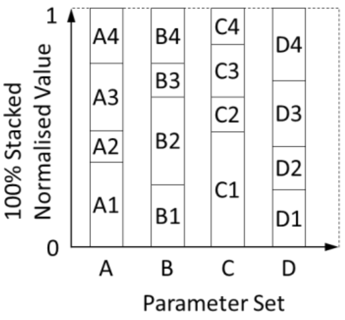


Figure 5 Parameter spectrum as 100% stacked normalised parameter sets.

3.1.2 Performance Metric Spectrum: As mentioned in section 2 algorithms have to be compared across a spectrum of metrics. Typical Metrics are:

- **Segment count**: This is the number of segments generated after segmentation. Typically segments containing less than a user defined number of points are not counted.

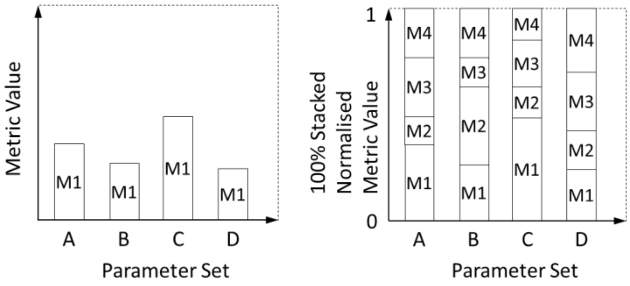
- **Homogeneity**: This is a function of the similarity of points within a segment across a segmentation. Two coomon forms of homogeneity are Intra-segment homogeneity and Inter-segment homogeneity. Intra-segment homogeneity is a measure of the disparity within a segment. Inter-segment homogeneity is a measure of the disparity between neighbouring segments. More on this can be found in Levine & Nazif (1985), Liu & Yang (1994), Huang & Dom (1995), Borsotti et al. (1998) and Radoux and Defourny (2008).

- **Heterogeneity**: This is a function of the di-similarity of points within a segment across a segmentation

- **Fragmentation metric**: After segmentation points belonging to the same surface maybe segmented into n_f number of segments. Here n_f is called the fragmentation of a segment. Statics of the distribution of n_f across all the segments yields a set of metrics for fragmentation, e.g., mean, standard deviation, percentile of n_f.

- **Deviations from a modelled surface**: The points in a segment can be used to model a surface. The distribution of the orthogonal distance/deviation of the segment points from this surface can be determined. For ideal segments the range of this distribution should be small.

Each segmentation algorithm has a parameter set. Each segmentation is performed for given parameter set values. A given segmentation is also evaluated by a set of metrics. Therefore, associated with each segmentation is a set of metrics and corresponding parameter set values (see Figure 6). Figure 6a shows the metric value for four parameter sets. This however is restrictive because it does not permit the visual comparison of multiple metric values against multiple parameter sets. To overcome this restriction the multiple metrics are represented as a 100% stacked bar chart of the normalised metric sets as shown in Figure 6b. Note here that by visualising Figure 5 and Figure 6b side by side it is now possible to compare segmentation parameter sets against their corresponding metrics.



(a) (b)

Figure 6 Metric spectrum. (a) A single metric value for multiple parameter sets A, B, C and D. (b) Metric spectrum as 100% stacked normalised metric sets.

3.1.3 Segmented Reference Point Clouds: As mentioned already segmentations are fit for purpose. The purpose changes with the scene and the intended application. The segmented reference point clouds are fit for purpose segmentations for different types of application (e.g., extracting planes, edges, etc.). Segmented reference point clouds can be generated manually or they can be points that have been segmented by trial and error and found to be optimal.
By calculating metrics for each scene type the optimal metrics for each scene type are obtained, Figure 7.

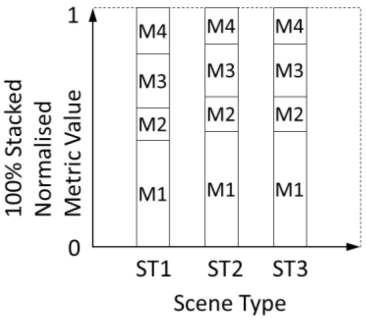


Figure 7 Metric spectrum for different scene types.

3.1.4 Using the Benchmark: By viewing the different charts side by side (Figure 5, Figure 6b, and Figure 7) it is now possible to compare segmentation parameters against their corresponding metrics and optimal metric sets. The problem of visually evaluating the parameter sets, parameter values, metrics and metric values in N-dimensional space has been reduced to 2 dimensions.

Figure 8 shows an example of one way of using the framework. In this example the best segmentation parameters for a scene of Type ST1(e.g., urban-residential) are being sought. The metrics for the reference ST1 are compared to the metric spectrum for the parameter sets of the segmentation algorithm. The closest match is parameter C. Therefore, this is chosen as the 'Algorithm – Parameter Set' combination likely to yield the best segmentation result.

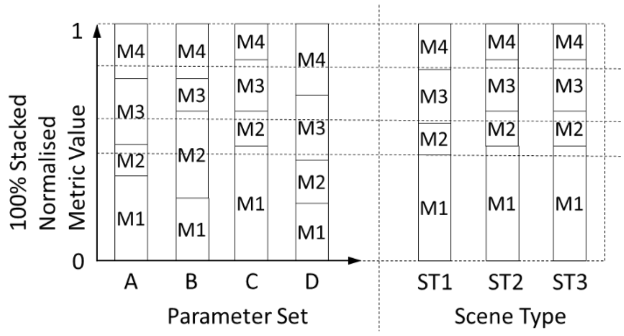


Figure 8 Example of finding the optimal parameter set for a scene type. Here the parameter set C has the metrics that best correspond to scene ST1. Therefore if a scene is of type ST1, for this algorithm the parameter set C is the optimal choice.

One final problem remains, and this is to quantify and rank parameter sets and segmentation algorithms for given scene types. The concept behind the proposed ranking is shown in Figure 9. The absolute difference of corresponding metric values for the reference scene and a parameter set are summed. The sum for a given parameter set, r_P, is given in equation 1, where m is the number of metrics in the metric spectrum.

$$r_P = \sum_{i=0}^{m} \left| M_{i,P} - M_{i,ST} \right| \qquad 1$$

If the parameter set matches the scene type, then r_P should be small. The value r_P can be calculated for a number of parameter sets. The values can then be ranked from smallest to largest. The best match is the parameter set with the smallest r_P.

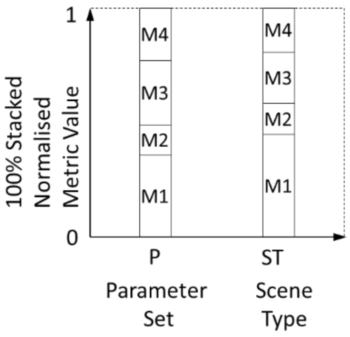


Figure 9 Comparison of two sets of metrics. A parameter set and a scene type match if the difference between the values of their corresponding metric values is small.

4. RESULTS

To test the concept of the framework a point cloud of a part of the Cape Town harbour was used. The point cloud has an average spacing of about 25 cm. This point cloud was manually segmented. This segmentation yielded 157 surfaces which included roof facets, domes and the ground, shown in Figure 10.

The point cloud was segmented using a three parameter (angle between neighbouring point normal, neighbourhood size, and minim segment size) region growing algorithm. Twenty seven different segmentations were done using different combinations of values (parameter sets). For each segmentation five metrics were calculated, absolute deviation, homogeneity (three variants, namely homogeneity, NPSSD and ratio p), and heterogeneity. Figure 11 shows the chart in which the parameter sets and their corresponding metric spectrum (compare this to Figure 8) are mapped. Also show in the figure is the metrics for the reference segmentation (ST, this is at the far right of the chart).

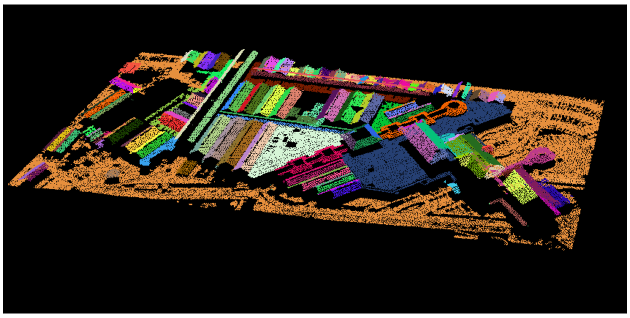


Figure 10 Manually segmented point cloud of part of the Cape Town harbour.

In Figure 11 it is apparent that parameter sets 1 to 9 are comparable to the reference segmentation. It's therefore expected that a region growing segmentation done with parameter sets 1 to 9 should yield fairly acceptable segments. Visual inspection showed that parameter set 3 yielded the best results while the other eight segmentations where tolerable.

Identification of the optimal parameter set should improve by using more metrics. Here only five were used and this may explain why the other eight parameter sets identified as ideal.

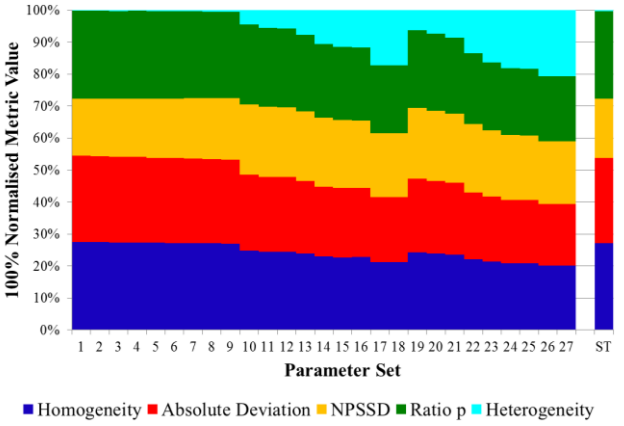


Figure 11 Performance Metric Spectrum for a particular algorithm. The metrics used to assess the algorithm are absolute

deviation, homogeneity (three variants, namely homogeneity, NPSSD and ratio p), and heterogeneity. The right most column is the metric spectrum for the reference scene (ST). The vertical-axis shows the 100% stacked normalised metrics values and the horizontal-axis shows the different parameter sets (here 27 different sets of parameters sets were used)

5. DISCUSSION

Early results are promising and suggest that the framework is sound. However, the framework does have shortcomings:

- **Too few metrics**: At the moment only five metrics were tested in the framework. Two important metrics were not considered in the tests, and these are metrics for over segmentation (fragmentation), and under segmentation (fusion). As the framework is developed these metrics will be included as metrics in the framework.

- **Geometry of the point cloud**: The framework doesn't account for variations in the geometry of a point cloud, e.g., point spacing. How this will be built into the framework will have to be considered.

- **Surface Extraction:** The framework has only been tested for surface extraction. Segmentation algorithms also extract edges, objects and so forth. The framework will have to be tested for edge and object extraction algorithms.

6. CONCLUSION

As more feature extraction algorithms use segmentation, so too will automated segmentation gain in importance. This paper proposes a framework for evaluating segmentation algorithms with a view to providing a mechanism for automatically selecting both optimal algorithms and optimal parameters for different types of scenes.

The framework extends the concept of benchmarking and attempts to reduce the selection problem from an N-dimensional domain to a 2-dimensional domain. The purpose of this is to permit visual inspection.

The framework has been tested on a real data set and early results are promising although much more work needs to be done. The framework still needs to be expanded to account for varying geometries of point clouds, additional metrics, and the extraction of other types of features (e.g., edges, objects).

When the framework is complete an attempt will be made to build a database of reference segments for many different types of scenes. The framework will then be tested to see if automated algorithm and parameter selection is possible with the framework. If this is successful, then this type of algorithm and parameter selection will be tried on other types of problems to see if it is general in application.

6.1 References

Borsotti, M., Campadelli, P., Schettini, R., 1998: Quantitative evaluation of color image segmentation results. *Pattern recognition letters*. 19(8), pp. 741-747.

Chen, X., Golovinskiy, A., Funkhouser, T., 2009. A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics (TOG)*. Vol. 28, p. 73.

Estrada, F. J., Jepson, A. D., 2009. Benchmarking image segmentation algorithms. *International Journal of Computer Vision*. 85(2), pp. 167-181.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., Fisher, R. B., 1996. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 18(7), pp. 673-689.

Huang, Q., Dom, B., 1995. Quantitative methods of evaluating image segmentation. *Proceedings, International Conference on Image Processing, IEEE*. 3. pp. 53-56.

Levine, M. D., Nazif, A. M., 1985. Dynamic measurement of computer generated image segmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (2), pp. 155-164.

Li, H., Cai, J., Nguyen, T. N. A., Zheng, J., 2013. A benchmark for semantic image segmentation. *IEEE International Conference on Multimedia and Expo (ICME), IEEE*. pp. 1-6.

Liu, J. & Yang, Y. H., 1994. Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 16(7), pp. 689-700.

Mersmann, O., Preuss, M., Trautmann, H., 2010. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis, Springer.

Rabbani, T., van den Heuvel, F. & Vosselmann, G., 2006: Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. 36(5), pp. 248-253.

Radoux, J., Defourny, P., 2008. Quality assessment of segmentation results devoted to object-based classification. *Object-Based Image Analysis, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg. Blaschke, T., Lang, S., Hay, G*. pp. 257-271.

Yasnoff, W. A., Mui, J. K., Bacus, J. W., 1977. Error measures for scene segmentation. *Pattern Recognition*. 9(4), pp. 217-231.

Zhang, Y. J. 1996. A survey on evaluation methods for image segmentation. *Pattern recognition*. 29(8), pp. 1335-1346.

Zhao, Y., Nie, X., Duan, Y., Huang, Y, Luo, S., 2011, A benchmark for interactive image segmentation algorithms. *IEEE Workshop on Person-Oriented Vision (POV), IEEE*. pp. 33-38.