Variational Shape Approximation of Point Set Surfaces

Martin Skrodzki [1], Eric Zimmermann [1], and Konrad Polthier [1]

[1] *Freie Universität Berlin, Germany*
[1] *martin.skrodzki@fu-berlin.de, eric.zimmermann@fu-berlin.de, konrad.polthier@fu-berlin.de*

Abstract

This work proposes an algorithm for point set segmentation based on the concept of Variational Shape Approximation (VSA), which uses the k-means approach. It iteratively selects seeds, grows flat planar proxy regions according to normal similarity, and updates the proxies. It is known that this algorithm does not converge in general. We provide a concrete example showing that the utilized error measure can indeed grow during the run of the algorithm. To reach convergence, we propose a modification of the original VSA. Further, we provide two new operations applied to the proxy regions, namely split *and* merge*, which enqueue in the pipeline and act according to a user-given parameter. The advantages over regular VSA are independence of both a prescribed number of proxies and a (manual) selection of seeds. Especially the latter is a common drawback of region-growing approaches in segmentation.*

1 Introduction

Point sets arise naturally in almost all kinds of three-dimensional acquisition processes, like 3D laser-scanning and have been recognized over 30 years ago as fundamental shape for representation in computer graphics. In comparison to meshes, they have a decreased demand in storage and have the advantage to be the direct representation of the object as obtained from acquisition devices, whilst lacking connectivity information.

However, in many applications, large parts of the point set carry redundant information. For example, a flat area of a surface can be sampled sparsely compared to an area of high curvature. The identification of such flat areas can be achieved e.g. via segmentation. Cohen-Steiner et al. proposed the Variational Shape Approximation (VSA), [1]. The procedure segments a given mesh into a given number of regions approximated by proxies, which can be used for a simplified model. A translation to point sets was done by Lee et al. [4] with a focus on feature extraction.

In a survey of simple geometric primitives detection methods for captured 3d data, the authors of [3] find VSA to be a method in particular suited to be run on individual algorithms as opposed to in- or outdoor scene data. It is summarized as an "automatic clustering" approach with low abstraction level and medium data fidelity, which attains a good balance in terms of e.g. speed, scalability, simplicity, and generality when compared to other methods, see [3] for details.

Despite its advantages, the VSA procedure and its translations have several downsides. First, as [1] also states, the procedure is not convergent in general, which is the same for meshes and point sets alike. Second, the available variants assume a prescribed number of proxies. Third, the quality of the final segmentation depends on the choice of starting seeds for the proxies. Our main contributions are:

- Provide an example of a growing error during the run of the VSA algorithm which applies to meshes [1] and point sets [4] alike.
- Presentation of a modified VSA version and proof of its convergence.
- Description of two new operations, *split* and *merge*, in the VSA pipeline, making the initial choice of a fixed proxy number and manual seed selection unnecessary.

2 VSA Procedure

The VSA procedure partitions a surface $S \subseteq \mathbb{R}^3$ into $m \in \mathbb{N}$ disjoint regions $R_i \subseteq S$, $\sqcup R_i = S$, where each region is associated a linear proxy $P_i = (C_i, N_i) \in \mathbb{R}^3 \times \mathbb{S}^2$, where C_i denotes the center and N_i denotes an associated unit-length normal, i.e. every proxy appears as a plane. After an initial seed selection every region grows w.r.t. a metric given by

$$\mathcal{L}^{2,1}(R_i, P_i) = \int_{x \in R_i} \|n(x) - N_i\|^2 \, dx,$$

where $n(x)$ denotes the surface normal at point $x \in S$. Throughout the whole paper, with $\|\cdot\|$ we refer to the Euclidean norm. Observe that this is the second proposed metric in [1] and the first one considers only the point positions. We focus on the one driven by normals as the authors found it to be favorable. In the discrete setting, where S is given as a (triangulated) mesh with elements t_j, the error metric simplifies to

$$\mathcal{L}^{2,1}(R_i, P_i) = \sum_{t_j} \|n(t_j) - N_i\|^2 \, |t_j| \,, \tag{2.1}$$

with $n(t_j)$ the element normal and $|t_j|$ its area. In their adaption to point sets, Lee et al. replaced the element normals and area term in Equation (2.1) by vertex normals and weighted all points equally with value 1. Here, it is possible to introduce more complex weighting terms in the point set setting (e.g. [6]), since we do not have an adequate equivalent to the area of an element. Afterwards, in both the mesh and the point set setting, the error measure

$$E(\{(R_i, P_i) \mid i = 1, \ldots, m\}) = \sum_{i=1}^{m} \mathcal{L}^{2,1}(R_i, P_i). \tag{2.2}$$

is minimized.

In order to find a minimum of the above error functional, the VSA procedure relies on a variation of Lloyd's k-means algorithm [5]. It works on both meshes and point sets, while the latter just uses the points, its normals, and a proper notion of neighborhoods. From all respective elements, m are chosen randomly to build up the proxies, with C_i as a proxy's barycenter and N_i its normal. The neighbors of selected elements are collected into a priority queue $\mathcal{Q}$ and sorted increasingly with growing $\mathcal{L}^{2,1}$. Afterwards the following three steps are performed iteratively until convergence:

1. *Flood:* As long as $\mathcal{Q}$ is not empty, pop the first element. Ignore it, if it has already been assigned to a proxy. If not, assign it to the proxy that pushed it into $\mathcal{Q}$ and collect all neighboring elements into the queue with proxy label they got pushed by.
2. *Proxy Update:* Update all proxy normals as averaged sum of the normals of their associated elements.
3. *Seeds:* For each proxy respectively, find an element in each region which is most similar according to the associated proxy normal and use it as seed element for the next flooding step.

In the work of Lee et al. [4], the authors use the k nearest neighbors as their neighborhood notion, with $k \in \{15, \dots, 20\}$.

3 Example for a Growing Error Functional

Although the authors of [1] state that they cannot guarantee global convergence, they do not provide a concrete example. In this work, we contribute to the understanding of the algorithm by describing a setup in which the error function (2.1) does grow during the run of VSA.

Consider the 2-dimensional setup shown in Figure 1(a) with n points given connected on a line with normal $\binom{-1}{1}$ next to a line of n points with normal $\binom{0}{1}$. At the right end of the second line, there is a single point with normal $\binom{-1}{0}$ and another single point with normal given by

$$N = \frac{1}{n+2}\left(n \cdot \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}-1\\0\end{pmatrix} + N\right).$$

Now, two proxies will act on this example, with their initial seeds shown in yellow and blue in Figure 1(a). They each start on one of the two lines of n points respectively. The result after a flood is shown in Figure 1(b), where each line is completely covered by the proxy starting on it and the two single points are associated to the proxy with normal $\binom{0}{1}$. After updating the proxy normals, the yellow proxy has normal $\binom{-1}{1}$ while the blue proxy has normal N given by the equation above. Thus, the yellow proxy starts from an arbitrary point on its line while the blue proxy starts from the rightmost point. The error after this first flood and proxy update is given by

$$E_1 = n \cdot \left\| \begin{pmatrix}0\\1\end{pmatrix} - N \right\|^2 + \left\| \begin{pmatrix}-1\\0\end{pmatrix} - N \right\|^2.$$

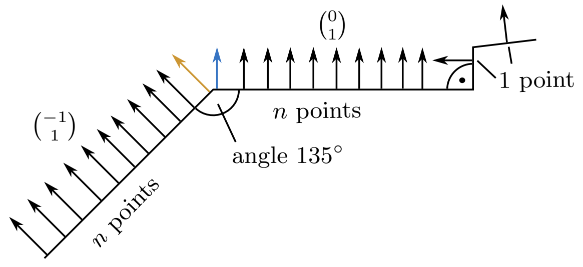
Starting from the new seed points, a second flood results in the situation shown in Figure 1(c). Here, almost all points except for the rightmost one are associated to the yellow proxy with normal $\binom{-1}{1}$. Its new normal after a proxy update is

$$N' = \frac{1}{2n+1}\left(n \cdot \begin{pmatrix}-1\\1\end{pmatrix} + n \cdot \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}-1\\0\end{pmatrix}\right),$$

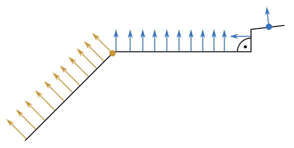
which amounts to an error after the second flood and proxy update given by

$$E_2 = n \cdot \left\| \begin{pmatrix}-1\\1\end{pmatrix} - N' \right\|^2 + n \cdot \left\| \begin{pmatrix}0\\1\end{pmatrix} - N' \right\|^2 + \left\| \begin{pmatrix}-1\\0\end{pmatrix} - N' \right\|^2.$$

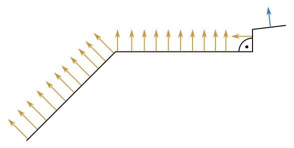
Choosing e.g. $n = 100$, we obtain $E_1 \approx 1.9802$, but $E_2 \approx 39.395$. Furthermore, the corresponding error value after the flood is also growing.



(a) Setup for growing error functional.



(b) Segmentation after first flood.



(c) Segmentation after second flood.

Figure 1: Example for a growth in the error measure after a flood and proxy update.

4 Modification for Converging VSA

In order to obtain an algorithm with guaranteed convergence, we propose to alter the steps of the algorithm as follows. First, we perform an initial *seeding*, one *flood* step, and a *proxy update* as explained above. Instead of the *seeding* step, we perform the following procedure:

4. *Switch*: For all points $p \in P$, consider their k nearest neighborhoods $\mathcal{N}_k(p)$. Assume that p is assigned to proxy P_i. If any point $p_\ell \in \mathcal{N}_k(p)$ is assigned to another proxy P_j, compute the change of the error measure in Equation (2.2) resulting from reassigning p from P_i to P_j. Compare it to the current best known reassignment. After iterating through all points $p \in P$, reassign the point such that the error measure is reduced maximally.

This new *switch* step replaces the *seed* step and the *flood* step described above. That is, it is only iterated together with the *proxy update*. The iteration is continued until no further *switch* operations can be performed. Although we describe the *switch* for point sets, it can easily be adopted for the mesh setting. For this alternate procedure, we can prove the following statement.

Theorem 1 (Error reduction by switch and proxy update, M. S. and E. Z.)**.** *Given a point set $P = \{p_1, \dots, p_{n'}\}$ with a neighborhood structure, such that the neighborhood graph on P is connected and normals $n_1, \dots, n_{n'}$ on P, with n' denoting the number of points in P, then each proxy update step and each switch step as defined above leads to proxies (R_i, P_i) with a smaller error measure in Equation (2.2).*

Proof. Concerning the proxy update step, consider

$$\begin{aligned}\nabla E(\{R_i, P_i\}) &= \nabla \sum_{i=1}^{m} \mathcal{L}^{2,1}(R_i, P_i)\\ &= \sum_{i=1}^{m} \sum_{p_j \in R_i} \nabla \omega_j \left\| n_j - N_i \right\|_2^2\\ &= \sum_{i=1}^{m} \sum_{p_j \in R_i} 2\omega_j (n_j - N_i).\end{aligned}$$

Setting $N_i = \frac{\sum_{p_\ell \in R_i} \omega_\ell n_\ell}{\sum_{p_\ell \in R_i} \omega_\ell}$, that is updating the proxy normal as weighted average of its assigned normals, we obtain

$$\begin{aligned}\sum_{p_j \in R_i} 2\omega_j (n_j - N_i) &= \sum_{p_j \in R_i} 2\omega_j n_j - \sum_{p_j \in R_i} 2\omega_j \left(\frac{\sum_{p_\ell \in R_i} \omega_\ell n_\ell}{\sum_{p_\ell \in R_i} \omega_\ell} \right) \\ &= \sum_{p_j \in R_i} 2\omega_j n_j - \left(\frac{\sum_{p_\ell \in R_i} 2\omega_\ell n_\ell}{\sum_{p_\ell \in R_i} \omega_\ell} \right) \cdot \sum_{p_j \in R_i} \omega_j \\ &= \sum_{p_j \in R_i} 2\omega_j n_j - \sum_{p_\ell \in R_i} 2\omega_\ell n_\ell = 0.\end{aligned}$$

Thus, at the chosen updated proxy normal, the energy reaches a (local) minimum. As the energy is convex as sum of norms, which are convex, the found minimum is indeed its global minimum for the current choice of segmentation.

Concerning the *switch* step, only those points are reassigned which reduce the value of error measure (2.2). Therefore, trivially, after a *switch* operation the error is smaller. □

This theorem proves the convergence of our modified VSA procedure.

5 New Operations: Split and Merge

Two drawbacks of region growing approaches are the prescribed number of proxies to be chosen and the proper placement of seed points. The latter is often done manually in order to enhance results. In this section, we want to propose two new operations, which also adds adaptability of the algorithm to input and desired outcome. Also, they give the user the possibility to control the level of detail, i.e. how fine the segmentation should be in the end. For this, we introduce a user-given parameter $\kappa \in \mathbb{R}_{\geq 0}$ which controls the maximum deviation within a proxy region R_i from a corresponding completely flat approximation. This parameter is used in the following two additional steps:

(a) *Split*: Given a proxy P_i with its region R_i such that $\mathcal{L}^{2,1}(R_i, P_i) > \kappa$. We use weighted principal component analysis [2] to compute the most spread direction of R_i. The set R_i is then split at the center of this direction into two new regions $R_i = R_i^1 \sqcup R_i^2$. The new normals are chosen as $N_i^1 = \sum_{p_j \in R_i^1} \frac{\omega_j n_j}{\sum_{p_j \in R_i^1} \omega_j}$ and a corresponding N_i^2 respectively. The new centers C_i^1 and C_i^2 are then placed at those points of R_i^1, R_i^2 that have least varying normals from N_i^1 and N_i^2 respectively.
Note that the reasoning of Theorem 1 holds for this case, too. Thus, the modified VSA procedure outlined above, enriched with an additional split step does continue to converge.

(b) *Merge*: Consider a pair P_i, P_j of neighboring proxies with their respective normals N_i, N_j. If the region $R' = R_i \sqcup R_j$ with normal $N' = \frac{N_i + N_j}{2}$ achieves an error measure (2.2) strictly less than κ, the two regions are replaced by their union R', with normal N' and a chosen center $C' \in R'$ with its normal least deviating from N'.
Note that we could allow only those pairs of neighboring regions to merge such that

$$\mathcal{L}^{2,1}(R_i, P_i) + \mathcal{L}^{2,1}(R_j, P_j) \leq \mathcal{L}^{2,1}(R', P').$$

Then, the error measure would not increase and termination of the algorithm would be guaranteed. However, this would result in neighboring regions not observing the user-given κ threshold. Therefore, we accept an increase of the global error measure in favor of a better region layout. In all experiments performed, the algorithm still converged.

Both operations alter the number m of proxies. Thereby, a significant disadvantage of the algorithm is eliminated as the user does not have to choose m a priori. It is replaced by the user's choice of κ, providing a semantic guarantee on the regions being built by the algorithm. The user can prescribe a value of κ computed from the desired curvature within a proxy.

In the *merge* process outlined above, we asked for two neighboring regions. However, we have not defined any relation on the regions yet. In the meshed case discussed above, two regions are neighbors if and only if they share an edge in the mesh. In the context of point sets, we cannot rely on this, thus we propose the following definition. During the *flood* step described above, an element p is popped from the priority queue $\mathcal{Q}$ together with a possible assignment to a region R_i. However, it is ignored if p has already been assigned to a region R_j. In this case, we denote R_i and R_j to be neighbors, if $i \neq j$. This can be extended to the *switch* simply by considering two proxies to be neighbors, if in the k nearest neighborhood of a point $p \in P_i$, there exists a point $q \in P_j$, $i \neq j$.

This finishes the whole VSA pipeline, including the additional two steps *merge* and *split*. In the following, we show several results of the altered VSA on point sets.

6 Experimental Results

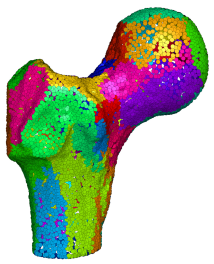
As our proposed extension of the VSA procedure gives some possibilities to arrange the steps, for the following experiments we used the pipeline:

Iterate(*seeding - flood - proxy update - one split - one merge*).

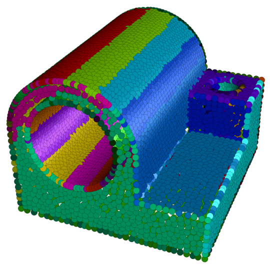
The displayed models with respective number of points in brackets are the CAD models Joint (9,998), Rockerarm (10,000), and Fandisk (6,475) as well as the real-world models Balljoint (10,002), Max Planck Bust (10,112), and Bunny (4,899), shown in Figures 2 and 3. The used parameters are given in the following table:

Model	m	m'	k	κ
Joint	15	34	8	20
Rockerarm	15	45	11	50
Fandisk	15	27	8	20
Balljoint	20	44	10	50
MaxPlanckBust	20	41	8	20
Bunny	15	36	8	20

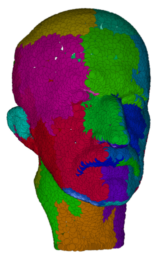
All these models visually give nice segmentation results, especially when we consider that all of them started with randomly selected seeds, where the number of seeds increased by the *merge* step in all cases to reflect the local geometries' behavior. A further visual comparison can be seen in Figure 3. The first row shows the segmentation gained with our approach, resulting in 27 (Fandisk) and 36 (Bunny) proxies. The second row shows results obtained by applying the VSA procedure with a corresponding number of seeds chosen randomly. Finally, there third row has been started with 15 triangle seeds for both models, which where selected manually for improved guidance of the algorithm, while additional seeds where added randomly to have the final seed numbers 27 (Fandisk) and 36 (Bunny). Here, we can see that our algorithm produces comparable results, despite the fact of not having manually placed seeds.



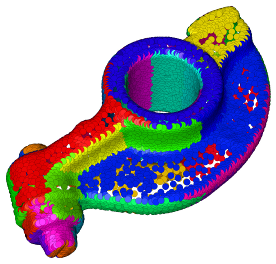
(a) Balljoint



(b) Joint



(c) Max Planck Bust



(d) Rockerarm

Figure 2: Our VSA adaption applied to four point-based models.

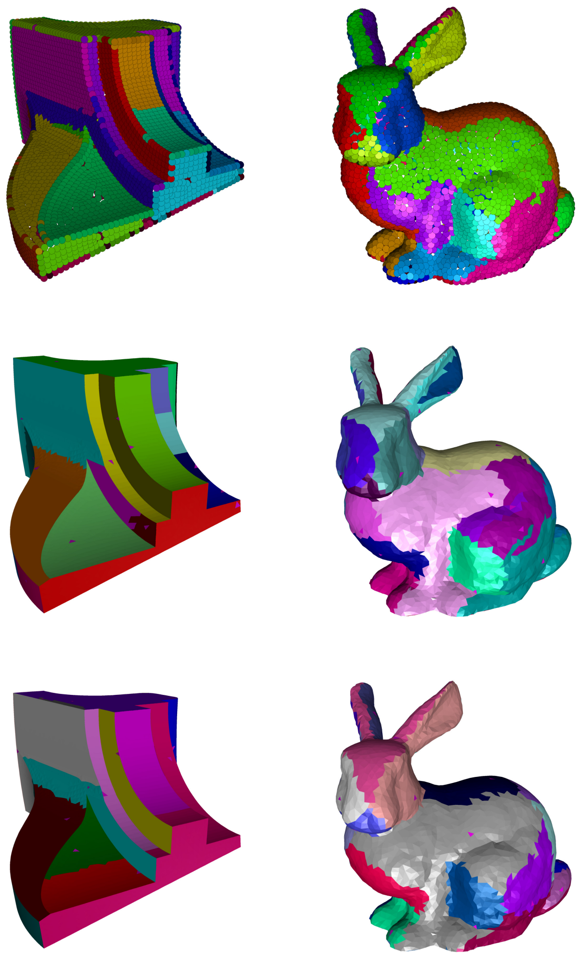


Figure 3: Segmentation applied to the Fandisk (left) and Bunny (right) models with our approach (first row), the VSA [1] with automatic (second row), and manual selected seeds (third row).

7 Conclusion

We have explained the VSA procedure, created an example with a growing error measure (2.1), proposed an alternate pipeline with the new operation *switch*, and gave proof of its convergence. Furthermore, we presented the two new operations *split* and *merge*, which make the procedure independent of a prescribed number of proxies and a (manual) initial selection of seeds as shown in our experiments.

For future work, we want to compare the quality of our segmentation approach with state-of-the-art methods and investigate a novel simplification algorithm based on the model approximating proxies.

References

[1] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational Shape Approximation. In *ACM Transactions on Graphics (TOG)*, 2004.

[2] Paul Harris, Chris Brunsdon, and Martin Charlton. Geographically weighted principal components analysis. *International Journal of Geographical Information Science*, 2011.

[3] Adrien Kaiser, Jose Alonso Ybanez Zepeda, and Tamy Boubekeur. A survey of simple geometric primitives detection methods for captured 3d data. In *Computer Graphics Forum*, volume 38, pages 167–196. Wiley Online Library, 2019.

[4] Kai Wah Lee and Pengbo Bo. Feature curve extraction from point clouds via developable strip intersection. *Journal of Computational Design and Engineering*, 2016.

[5] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[6] Martin Skrodzki, Johanna Jansen, and Konrad Polthier. Directional density measure to intrinsically estimate and counteract non-uniformity in point clouds. *Computer Aided Geometric Design*, 64:73–89, 2018.