Graph Cut Based Point-Cloud Segmentation for Polygonal Reconstruction

David Sedlacek and Jiri Zara

Czech Technical University in Prague, Faculty of Electrical Engineering

Abstract. The reconstruction of 3D objects from a point-cloud is based on sufficient separation of the points representing objects of interest from the points of other, unwanted objects. This operation called segmentation is discussed in this paper. We present an interactive unstructured point-cloud segmentation based on graph cut method where the cost function is derived from euclidean distance of point-cloud points. The graph topology and direct 3D point-cloud segmentation are the novel parts of our work. The segmentation is presented on real application, the terrain reconstruction of a complex miniature paper model, the Langweil model of Prague.

Keywords: segmentation, graph cut, point-cloud, terrain reconstruction, Langweil model.

1 Introduction

The 3D objects reconstruction from a set of photos often utilizes a 3D point-cloud to separate various objects of interest. If the shape of reconstructed objects is unknown an automatic segmentation does not provide satisfactory results. An interactive processing of input point-cloud is thus more efficient. Such a user-driven segmentation can be performed either in image space (by selecting areas in input photos) or in 3D space of the point-cloud. A set of segmented points is finally converted into a polygonal model representing boundaries of reconstructed objects (see Fig. 1).

The main contribution of this paper is in 3D point-cloud interactive segmentation performed completely in 3D space. The segmentation process derives a

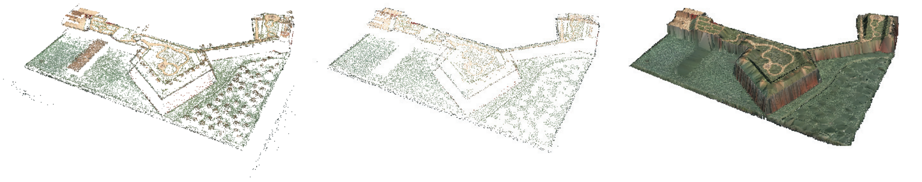

Fig. 1. Three steps of the reconstruction process (from left to right): input point-cloud, segmented point-cloud (ground points only), reconstructed polygonal ground

G. Bebis et al. (Eds.): ISVC 2009, Part II, LNCS 5876, pp. 218–227, 2009.

benefit from a graph cut algorithm, i.e. from searching the minimal cut or maximum float in the weighted graph. The weight function is based on euclidean distance and no other information is needed (e.g. normals, colours). The graph topology presented in this paper has not been used in graph cut algorithm so far.

Yuan et al. already describe graph cut for point-cloud segmentation at [1]. A hybrid approach to 3D segmentation is described in their paper. The point-cloud is pre-segmented into clusters based on certain properties (points distance, colours, normals). The user can freely manipulate with point-cloud into desired view position, then the point-cloud is rendered into an image. The graph cut algorithm is applied on this image, the segmentation criterion is derived from points colour and normal. Finally the image segmentation is projected back, to the pre-segmented 3D point-cloud clusters.

Contrary to this approach, our method is completely 3D based and does not require any additional information about point-cloud which results in more generic usage. The paper is organized as follows. Section 2 introduces previous work related to graph cut and segmentation. Section 3 is the core of this paper – it describes the point-cloud segmentation and graph construction. The application of segmentation is presented in Section 4 on the example of Langweil model ground reconstruction. The main contributions of our approach are summarized in Section 5.

2 Related Work

Our work combines two previously unconnected techniques, the unstructured point-cloud segmentation with powerful graph cut algorithm [2] in order to get interactive segmentation. The graph cut algorithm is designed to minimize energy function in weighted graph where the energy function defines segmentation. This technique is widely used in standard image segmentation [3,4,5] or in segmentation of range images or stereo matching [6,7]. Several works benefits from graph cuts for 3D mesh segmentation [8,9] or surface extraction [10,11,12].

Segmentation refers to the task of labelling a set of measurements in the 3D object space (point-cloud). The points with the same label should satisfy several similarity conditions (e.g. close distance, colour, normal) and segmentation methods differ in a way how this is achieved. Segmentation of unstructured point-cloud was discussed in several works: clustering [13,1], region growing [14,15], divide and conquer, mesh segmentation[15]. Our method is similar to the region growing but it provides globally optimized results thanks to the graph cut algorithm behaviour.

In the area of point-cloud segmentation, the graph cut has been used in several works. Similarly as previously mentioned [1], the other works use some kinds of pre-segmentation (they do not segment the point-cloud directly) and they require additional information from 2D images like color, gradient or depth. Quan et al. [16] use a ratio of 3D euclidean distance and color gradient from corresponding pictures as a segmentation criterion. They build a weighted graph, pre-segment it with thresholding and then the graph cut is applied on the wrongly pre-segmented parts, based on user will. Anguelov et al. applied the Markov Random Fields for point-cloud and range data segmentation in [17] where the object segmentation is based on local surface features. Trained Markov Random Fields can be solved through graph cut.

3 Segmentation

We present a point-cloud segmentation as a graph cut problem. It means a finding minimal cut at a weighted graph containing all point-cloud points. The graph must sufficiently represent an unstructured point-cloud and the segmentation process minimizes the energy function. The energy function reflects equilibrium between segmentation and graph topology. The graph construction is described first followed by segmentation description.

3.1 Graph Construction

The weighted graph G contains all 3D points (vertices) V from the input point-cloud and two imaginary vertices called terminals. The terminals represent assignment of points from V to two sets representing Object (S) or Background (T) points. Terminals corresponding to these sets are called source s and sink t. Every vertex from V is initially connected with N nearest neighbours from V where each edge E connecting two vertices is weighted with capacity C_E

$$C_E = \kappa * e^{-D_E^2/\sigma} \tag{1}$$

where D_E is Euclidean distance of two vertices and σ controls exponent function behaviour.

$$D_E^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \tag{2}$$

The constant κ is scaling factor from float to integer values $\langle 0; \kappa \rangle$. The simple graph is shown in figure 2a where every point is connected with $N = 4$ nearest neighbours.

The edges between vertices V and terminals are not set in initialization phase. This is a special behaviour in comparison with standard graph cut as presented in [3] where edges among point $p \in \{V\}$ and two terminals s and t exist and he edges capacity is set to probability of assigning the point to Object or Background set based on certain a priory known information. For example the probability may reflect how the point fits to known colour model of the object and background.

The omission of edges to terminals has two advantages for graph cut algorithm. First advantage is decreasing the edges amount in final graph which results in higher speed in max flow computing. Boykov in [2] proved that asymptotic complexity of their graph cut implementation is $O(mn^2|C|)$ where m is point count, n is edges count and $|C|$ is minimal cut size. The second advantage for point-cloud segmentation is in segmentation behaviour. The second advantage deals with the graph cut behaviour. Based on our observation, the algorithm generates too many small areas (either Object or Background labeled) when initialized with many edges connected to terminals [18]. Having no such initial edges allows us to better control the algorithm output interactively.

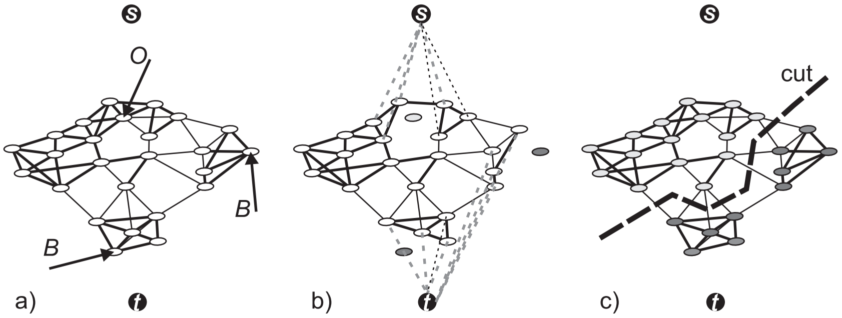

Fig. 2. Graph topology and segmentation example. Vertices (white dots) are connected with 4 nearest neighbours here. The edges cost are reflected by edge's thickness. a) Points specified interactively by a user (seeds) for Object, Background are shown as arrows with appropriate letter. b) Seeds are replaced by corresponding terminals, newly created edges to terminals inherit their capacities from corresponding edges previously connecting seeds with their neighbours. c) Segmented input graph. Light grey vertices are Objects, dark grey ones are Background.

3.2 Graph Cut Based Segmentation

Segmentation in our approach is presented as a minimal graph cut problem. We want to separate the V set into two disjunctive sets labelled S and T while the energy function $E(L)$ describing equilibrium between segmentation and point-cloud behaviour is minimized. The energy function is defined as

$$E(L) = \lambda \sum_{p \in V} R_p(L_p) + \sum_{\langle p,q \rangle \in M} B_{\langle p,q \rangle}(L_p, L_q) \tag{3}$$

where $L = \{L_p | p \in V\}$ is labelling of point-cloud V. The coefficient $\lambda \geq 0$ specifies relative importance of the region term $R_p()$ versus the boundaries properties term $B_{\langle p,q \rangle}()$. The set of all N nearest neighbouring pairs from V is M.

The boundary properties term reflects a distance function between neighbour points and should be taken into account only in case of different labelling between neighbour pixels.

$$B_{p,q}(L_p, L_q) = C_E * \delta(L_p, L_q) \tag{4}$$

Where C_E is edge capacity between points p,q from equation 1 and

$$\delta(L_p, L_q) = \{ {}^{1 \; if L_p \neq L_q}_{0 \;\; othervise} \tag{5}$$

The region term $R_p()$ influences a process of segmentation through an interaction. A user specifies point-cloud labelling by selecting several points. We connect these points either to terminal s or t (see Fig. 2a). For the case of Object terminal s, capacities from point p to terminals will be

$$\begin{aligned} C_{sp} &= \infty \\ C_{pt} &= 0 \end{aligned} \tag{6}$$

and similarly for Background terminal t.

Since we do not want to increase a number of edges during interactive work, we apply the following simplifications on the flow graph. First, we do not construct any edge to terminal when the edge capacity is 0. Secondly, the point connected to a terminal with maximum (infinite) edge capacity can be considered as a terminal itself. In this case, we can disconnect this point from the graph, reconnect its edges to adequate terminal, and keep the capacities of these edges unchanged (see fig. 2b).

The segmentation is computed using max flow algorithm, implemented as in [2]. After each user input (a stroke in our case) the graph cut is computed and visualised, so the user can decide to add more strokes. The segmentation process is illustrated in figure 3.

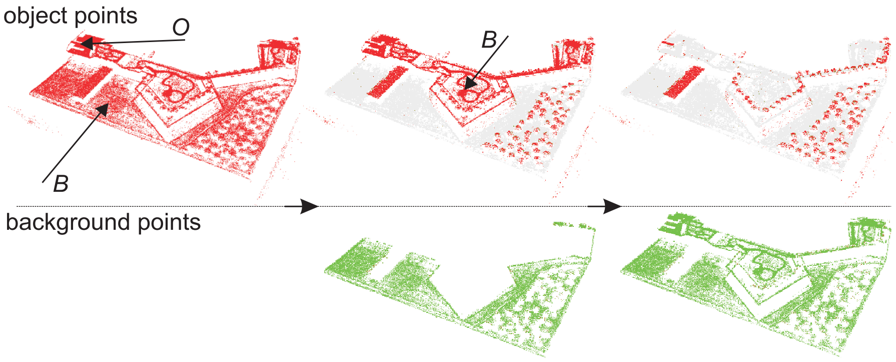

Fig. 3. Segmentation process. The first row represents points in S set, the T set points are in bottom row. From left to right, initially points are interpreted as members of S set, the seeds are represented by arrows with appropriate letter. At second column the points are segmented depending on seeds from the first image, additional background seed is specified. The final segmentation is on the right.

4 Application-Terrain Reconstruction

The proposed segmentation method was intensively used in terrain reconstruction of Langweil model of Prague. Langweil model is described in Section 4.1. The terrain reconstruction with details about point-cloud segmentation and final triangulation are presented in Section 4.2. The other tests were carried out with

LiDAR data from [http://opentopography.org], Northern San Andreas Fault subset. We choose LiDAR data because they are well customizable for our application and prove that the algorithms are applicable to more common problems than was the main application purpose.

4.1 Langweil Model of Prague

The oldest model of Prague was created by Antonín Langweil in years 1826 - 1837 and is placed at City of Prague Museum [www.muzeumprahy.cz]. It is made from paper and illustrated by pen-and-ink drawings. The model size is about 3.5m × 6m in scale 1:480, corresponding area of the real city is about 1.6km × 2.6km. There are more than 2000 buildings corresponding to land register and almost 7000 other unique objects like shelters, small walls, statues, and trees. The ground varies throughout the city. The Old Town is mainly planar going down to the Vltava river and then to the hilly part of Prague, the Prague Castle and gardens around. The model itself consists of 60 parts, each with different number of objects and a complexity, see Fig. 4.

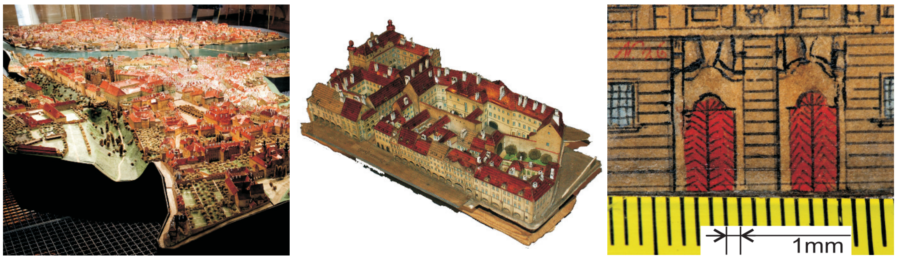

Fig. 4. From left to right: Langweil model of Prague at City of Prague Museum, one model part, detail photo - a wall with millimetre ruler

Because of paper nature of the model, museum experts prohibited standard types of scanners, lasers, high-powered flashing, and model touching. For this reason, a special robot was developed for the reconstruction purposes. The robot automatically took photos of one model part from several camera positions and orientations. The whole scanning took two months of non-stop running. Almost 300 000 photos with the *4K* resolution were obtained.

4.2 Terrain Reconstruction

The terrain reconstruction is performed in a three-stage process. The first stage is a point-cloud generation from top view photos. The point-cloud is then interactively segmented using segmentation described in Section 3 - we present implementation details in the following text. The last step is a polygonal terrain reconstruction preceded by point-cloud filtration.

Segmentation: The aim of the application of our novel method has been to separate ground points from other objects in paper model like walls or trees. For this reason we decide to adjust a distance function from equation 2 to

$$D_E^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 + \gamma * (z_1 - z_2)^2 \tag{7}$$

where γ is penalisation of height in Euclidean distance.

In equations 1 and 7 we pre-set experimentally the values $\sigma = 0.25$ and $\gamma = 200$. These values depend on the point-cloud density and real ground behaviour. If the ground is almost planar γ should be increased for faster segmentation convergence and contrariwise for hilly terrain. Similarly with σ value. If the point-cloud density is low σ should be higher so the distance function do not penalties distant points. The neighbour count was set to $N = 16$, scaling factor was set to $\kappa = 4096$ and we did not experiment with those values. Several σ values were tested, 0.05, 0.07, 0.1, 0.25, 0.5. The values from 0.05 to 0.1 inclusive were to low, the segmentation can't cross over little discontinuities between points and the user need to use more strokes. On the other hand $\sigma = 0.5$ was to high and segmentation process connect distant points together so the user need to add more opposite strokes. The γ value is more dependent on point-cloud behaviour than σ. The γ value should be lowered for hilly terrain (tested values 50, 100 works well) and for flat parts should be higher (tested values 200, 250).

User interaction is necessary during segmentation process. The point-cloud is shown to the user in 2D form. User looks from orthographic top view to the terrain and selects which part should be a ground. The points are separated into two sets - ground points T and the other points S representing buildings, roofs, walls and trees. The consecutive segmentation is visualised by green and red pixels (T and S sets respectively) and the points depth is reflected through the colour intensity. The user add strokes to the image what is interpreted as adding edges between selected point-cloud points and terminals s or t as described in Section 3.2.

Initially, all point-cloud points are interpreted and visualised as members of S set, i.e. objects. User adds strokes to the places where the points are assigned to the wrong set. After each stroke the minimal-cut is computed and the segmentation is visualised, so the user can decide if it is necessary to add more strokes or if the point-cloud is well-segmented, see Fig. 3. When the user is satisfied with segmentation, all points of the S set are deleted and only T points are used in the next step, the filtration.

Filtration and polygonal ground reconstruction: The segmented point-cloud contains all points signed as a ground but the points are still too dense and are not suitable for a polygon reconstruction, i.e. the polygon ground reconstructed from those points will have to much triangles. For this reason we should perform point-cloud filtration before geometry reconstruction. The second possibility is to decimate triangles after polygon reconstruction [19]. The point-cloud structure from previous step direct to filtration before reconstruction.

The final step of our process is point-cloud transformation into polygonal representation. There exist processes mainly based on marching cubes [20] or voronoi diagrams [21,22]. Those approaches try to represent 3D point-cloud as a solid 3D polygonal object. In case of terrain model data behaviour is more 2D than 3D. The segmented and filtered point-cloud looks like height map and for this reason the 2D voronoi triangulation is enough.

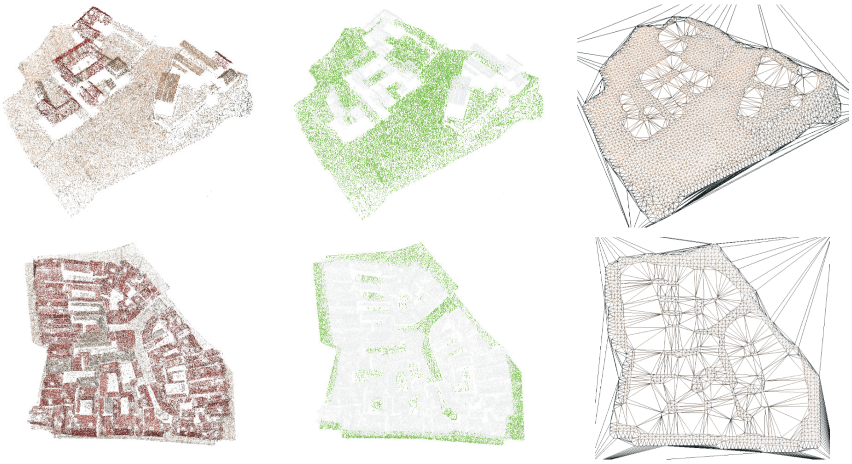

Fig. 5. Point-cloud segmentation and polygon terrain reconstruction examples. From left to right: input point-cloud, segmented points, reconstructed polygonal ground.

Table 1. Point-cloud sizes, initialization times, user interaction counts. The initialization phase is preprocessed off-line. Parts 10 and 13 are in figure 5, part 23 is in pictures 1 and 3. Part 39 and LiDAR data are not shown.

model	points	init time	update time	background seeds	object seeds
part 10	235 672	2m 40s	⟨ 1s	18	0
part 13	86 225	1m 10s	⟨ 1s	2	0
part 23	147 490	1m 52s	⟨ 1s	11	5
part 39	754 998	6m 36s	⟨ 1s	46	7
LiDAR	500 000	5m 27s	⟨ 1s	10	2

5 Results

We have successfully applied graph cut method for unstructured 3D point-cloud segmentation. We have utilized this novel algorithm on the large set of input data. In the case of the Langweil model of Prague, point-clouds were of typical size of half million points. For each such a point-cloud, human operators were able to separate ground data using several strokes and/or clicks, usually from 10 to 50. All these user inputs were processed immediately at interactive speed (less than 2 seconds) detailed results are in table 1. The final reconstructed polygonal terrain has become a solid and precise base, on which the other objects of the whole 3D digital Langweil model were positioned. Examples taken from the project [www.langweil.eu] are shown in fig. 6.

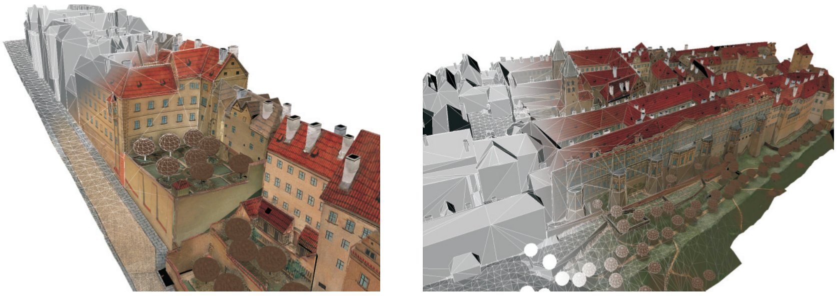

Fig. 6. Final reconstruction of two different parts showed as wire-frame and textured model. For better illustration the ground is displayed together with other objects as is in final model.

Acknowledgements

This research has been partially supported by MSMT under the research programs MSM 6840770014 and LC-06008 (Center for Computer Graphics). We also appreciate a lot of scientific suggestions given by Daniel Sýkora.

References

1. Joaquim, T.I.: In: Yuan, J.A.J., Xu, X., Nguyen, H., Shesh, A., Chen, B. (eds.) Eurographics workshop on sketch-based interfaces and modeling (2005)
2. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (2004)
3. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images, vol. 1, pp. 105–112 (2001)
4. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
5. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 15, 1101–1113 (1993)
6. Kahler, O., Rodner, E., Denzler, J.: On fusion of range and intensity information using graph cut for planar patch segmentation. Int. J. Intell. Syst. Technol. Appl. 5, 365–373 (2008)
7. Bleyer, M., Gelautz, M.: Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. Image Commun 22, 127–143 (2007)
8. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. In: SIGGRAPH 2003: ACM SIGGRAPH, pp. 954–961. ACM, New York (2003)

9. Golovinskiy, A., Funkhouser, T.: Randomized cuts for 3d mesh analysis. In: SIGGRAPH Asia 2008: ACM SIGGRAPH, Asia, pp. 1–12. ACM, New York (2008)
10. Hornung, A., Kobbelt, L.: Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In: SGP 2006: Proceedings of the fourth Eurographics symposium on Geometry processing, Aire-la-Ville, pp. 41–50. Eurographics Association, Switzerland (2006)
11. Sinha, S., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation. In: Tenth IEEE International Conference on Computer Vision, ICCV 2005., vol. 1, pp. 349–356 (2005)
12. Labatut, P., Pons, J.P., Keriven, R.: Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In: IEEE 11th International Conference on Computer Vision. ICCV 2007, pp. 1–8 (2007)
13. Dorninger, P., Nothegger, C.: 3d segmentation of unstructured point clouds for building modelling, p. 191 (2007)
14. Rabbani, T., van den Heuvel, F., Vosselmann, G.: Segmentation of point clouds using smoothness constraint, pp. xx–yy (2006)
15. Jiang, X.Y., Bunke, H.: Fast segmentation of range images into planar regions by scan line grouping. Machine Vision and Applications, 115–122 (1994)
16. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B.: Image-based plant modeling. ACM Trans. Graph. 25(3), 599–604 (2006)
17. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A.: Discriminative learning of markov random fields for segmentation of 3d scan data. In: CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, pp. 169–176. IEEE Computer Society, Los Alamitos (2005)
18. Sýkora, D., Dingliana, J., Collins, S.: Lazybrush: Flexible painting tool for hand-drawn cartoons. Comput. Graph. Forum 28(2), 599–608 (2009)
19. Y., H.M.O., K., N.: Efficient and feature-preserving triangular mesh decimation. Journal of WSCG, 167–174 (2004)
20. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. Computer Graphics 21(1987)
21. Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. ACM Comput. Surv. 23(3), 345–405 (1991)
22. Amenta, N., Bern, M., Kamvysselis, M.: A new voronoi-based surface reconstruction algorithm. In: SIGGRAPH 1998: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 415–421. ACM, New York (1998)