ENTAC 2024
XX ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO
Maceió, Brasil, 9 a 11 de outubro de 2024

Utilização do algoritmo DBSCAN para segmentação de nuvens de pontos de ambientes internos

DBSCAN algorithm for segmenting indoor point clouds

Bruna Brito Liberal
Universidade Federal de Pernambuco | Recife | Brasil | bruna.liberal@ufpe.br
Gustavo de Hollanda Cavalcanti Soares
Universidade Federal de Pernambuco | Recife | Brasil | ghcs@cin.ufpe.br
Arthur Henrique da Costa e Silva
Departamento Nacional de Infraestrutura de Transportes | Recife | Brasil | arthursilva@outlook.com
Rachel Perez Palha
Universidade Federal de Pernambuco | Recife | Brasil | rachel.palha@ufpe.br

Resumo

Levantamentos feitos por *laser scanners* vêm sendo requisitados em um contexto de inovação e economia de recursos dentro do setor da Arquitetura, Engenharia, Construção e Operação (AECO), especialmente no campo *BIM* (*Building Information Modeling*), pois são capazes de gerar nuvens de pontos que representam com precisão o ambiente real. Apesar de facilitar os levantamentos, trabalhar com nuvens de pontos pode ser uma tarefa complexa, pois podem exigir poder de processamento computacional elevado, além da modelagem de modelos *BIM* a partir dessas nuvens não ser intuitiva devido à sobreposição de pontos de diversos elementos. Com o objetivo de facilitar a modelagem *BIM* a partir de nuvens de pontos, o algoritmo de segmentação *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*) foi utilizado neste estudo. Foi elaborado um script em linguagem *Python* com utilização do *DBSCAN* para segmentar uma nuvem de pontos de um ambiente interno. O algoritmo mostrou ser útil para separar elementos como paredes, pisos e telhados, permitindo a exportação das nuvens segmentadas de cada um desses elementos para softwares como o *Revit* e facilitando a modelagem em *BIM*.

Palavras-chave: Nuvens de pontos. *DBSCAN*. *BIM*.

Abstract

Surveys carried out by laser scanners have been requested in a context of innovation and resource savings within the Architecture, Engineering, Construction and Operation (AECO) sector, especially in the BIM (Building Information Modeling) field, as they are capable of generating point clouds that accurately represent the real environment. Despite making surveys easier, working with point clouds can be a complex task, as they can require high computational processing power, in addition to modeling BIM models from these clouds being non-intuitive

Como citar:
LIBERAL, B.; SOARES, G.; SILVA, A.; PALHA, R. Utilização do algoritmo DBSCAN para segmentação de nuvens de pontos de ambientes internos. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 20., 2024, Maceió. **Anais**... Maceió: ANTAC, 2024.

due to the overlapping of points from different elements. With the aim of facilitating BIM modeling from point clouds, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) segmentation algorithm was used in this study. An algorithm was developed in Python language using DBSCAN to segment a point cloud from an indoor environment. The algorithm proved to be useful for separating elements such as walls, floors, and roofs, allowing the export of segmented clouds of each of these elements to software such as Revit and facilitating BIM modeling.

Keywords: Point clouds. DBSCAN. BIM.

INTRODUÇÃO

As nuvens de pontos vêm sendo utilizadas com frequência e com os mais diversos objetivos dentro do campo da Arquitetura, Engenharia, Construção e Operação (AECO) e são normalmente obtidas por meio do mapeamento fotogramétrico com VANTs (veículos aéreos não tripulados) ou por meio de *LiDAR* (*Light Detection and Ranging*). Elas são representações de áreas ou objetos em formato de pontos distribuídos em coordenadas tridimensionais, e são capazes de recriar a superfície externa do elemento escaneado em formato 3D. Por meio dessas nuvens, é possível levantar diversas informações como profundidade, elevação, geometria e localização. Quanto mais densa for uma nuvem de pontos, mais informações são obtidas por meio dela [1].

Uma das vantagens de utilizar nuvens de pontos é o apoio na construção de modelos *BIM* (*Building Information Modeling*), cada vez mais requisitados, dado o Decreto 9.377 [2], que estabeleceu a Estratégia BIM BR. O processo de se trabalhar com nuvem de pontos, apesar da facilidade no levantamento de áreas existentes, pode se tornar pouco intuitivo na fase do pós-processamento, pela dificuldade de se trabalhar com um grande volume de pontos que só possuem informação de localização e cor (*RGB - Red, Green, Blue*). Por esse motivo, modelar em *softwares BIM* utilizando como base as nuvens de pontos não é tão fácil, e busca-se alternativas para facilitar o manuseamento dessas nuvens, como a segmentação de nuvem de pontos.

A segmentação de nuvens de pontos 3D é uma tarefa que envolve a identificação e separação de pontos dentro de *clusters* em um espaço tridimensional. O uso de algoritmos de clusterização, como o *DBSCAN*, tem mostrado resultados promissores nesta área, permitindo a descoberta de *clusters* de formato arbitrários dentro da nuvem de pontos [3]. A aplicação de algoritmos para segmentação de nuvens de pontos tem o potencial de transformar a forma como os engenheiros civis realizam suas tarefas. Com a capacidade de processar e analisar grandes volumes de dados de forma rápida e precisa, os engenheiros arquitetos podem obter insights valiosos que podem ajudar na tomada de decisões informadas e na otimização de processos.

No entanto, apesar do potencial dos algoritmos de segmentação, ainda existem alguns desafios a serem superados. A qualidade e a precisão dos dados, a escolha do algoritmo adequado, a interpretação dos resultados e a integração dos insights obtidos no processo de tomada de decisão são algumas das questões que precisam ser abordadas para maximizar o potencial da automatização desses processos dentro da

indústria AECO. O objetivo deste estudo é propor um script capaz de segmentar uma nuvem de pontos de um ambiente interno utilizando algoritmos de segmentação, para facilitar o uso de nuvens de pontos dentro de um contexto BIM. Nesse contexto, este artigo é dividido nas seguintes seções: Referencial Teórico, onde serão apresentados os conceitos ligados ao estudo; Metodologia, onde está apresentada a instrumentação e os parâmetros utilizados no desenvolvimento do estudo; Resultados, onde são apresentados os *output*s dos algoritmos; e Conclusão, com as considerações finais.

REFERENCIAL TEÓRICO

Este estudo se baseia na aplicação de dois algoritmos fundamentais, o *RANSAC* (*Random Sample Consensus*) e o *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*), para a análise de dados de nuvens de pontos. Esses algoritmos foram escolhidos com base em suas aplicações bem-sucedidas em estudos anteriores [4], em que foram utilizados para a segmentação semântica de nuvens de pontos 3D baseada em *voxels*. A nuvem de pontos utilizada neste estudo foi obtida por meio da tecnologia *LiDAR* e, nesta seção, estão apresentados os conceitos referentes a cada um desses elementos para que haja o entendimento necessário.

ESCANEAMENTO A LASER

Inicialmente, a tecnologia *LiDAR* era utilizada para levantamento de dados para elaboração de Modelos Digitais de Elevação (MDE) [5], que representa as altitudes da superfície topográfica de uma região, em um período em que os métodos tradicionais não eram suficientes, principalmente em áreas de difícil acesso. Hoje, o uso da tecnologia se expandiu para outras aplicações, e o *LiDAR* é utilizado com frequência para levantamentos *as-built* dentro da engenharia civil, dada a sua precisão na captura de nuvens de pontos. Tendo em vista que são um dos sistemas de sensoriamento remoto mais apropriados para gerar informações espaciais precisas e diretamente georreferenciadas para características de forma e superfície [6].

Apesar das vantagens de se utilizar a tecnologia *LiDAR* dentro da engenharia, a implementação de dados 3D para a criação precisa de *as-builts* ainda é um desafio, especialmente para detalhes finos dos objetos de construção num ambiente interior [7]. Por esse motivo, buscam-se alternativas para facilitar o manuseio de nuvens de pontos para criação de modelos *BIM*, como é o caso dos algoritmos de segmentação. Os algoritmos podem ser utilizados para segmentar os diferentes elementos presentes nas nuvens de pontos em *clusters*, ou seja, em nuvens de pontos menores para cada objeto, que posteriormente podem ser exportados para os *softwares BIM*.

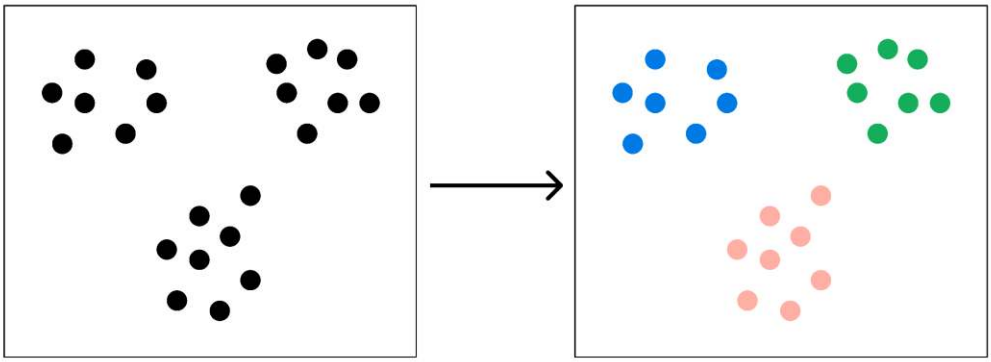
ALGORITMOS DE SEGMENTAÇÃO

O algoritmo *RANdom SAmple Consensus* (*RANSAC*) proposto por [8], é uma técnica usada para estimar parâmetros de um modelo matemático, mesmo quando há muitos dados espúrios ou *outliers*. Ele seleciona aleatoriamente um pequeno conjunto de dados, ajusta um modelo a esses dados e, em seguida, verifica quantos outros pontos

se encaixam bem nesse modelo. Se suficientes pontos se encaixarem, o modelo é considerado bom. O processo é repetido várias vezes para garantir a melhor estimativa. É uma abordagem robusta e amplamente utilizada em visão computacional e outras áreas que lidam com dados ruidosos.

Já o algoritmo *DBSCAN*, desenvolvido por [9], é um método de agrupamento baseado em densidade. Este algoritmo define *clusters* como regiões de alta densidade separadas por regiões de baixa densidade. Diferentemente de outros algoritmos de agrupamento, o *DBSCAN* não requer que o usuário especifique o número de *clusters*, podendo descobrir *clusters* de forma arbitrária. Este algoritmo tem se mostrado particularmente eficaz ao lidar com grandes conjuntos de dados e *clusters* de formas variáveis [10].

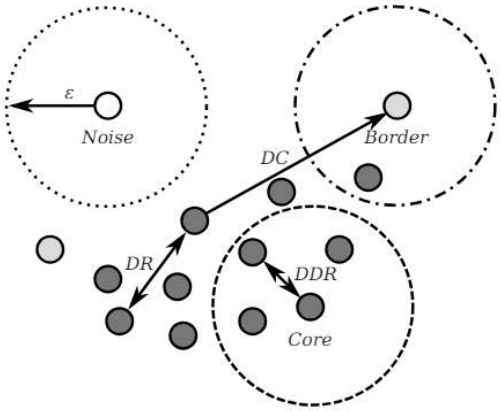
Figura 1: Exemplo prático do algoritmo *DBSCAN*.



Fonte: Os autores.

Neste algoritmo, devem ser definidos dois parâmetros primordiais: *min_cluster_points* e *epsilon* [9]. O *min_cluster_points* é o número mínimo de pontos que devem estar próximos para formar um *cluster* denso. Por outro lado, *epsilon* é a distância máxima entre dois pontos para que eles sejam considerados vizinhos. Se um ponto tem pelo menos *min_cluster_points* dentro de sua vizinhança *epsilon*, ele é considerado um ponto central do *cluster*. A relação entre *min_cluster_points* e *epsilon* determina a forma e o tamanho dos *clusters*. Um *epsilon* pequeno pode resultar em muitos, porém pequenos *clusters*, enquanto um *epsilon* grande pode fundir vários *clusters*. Da mesma forma, um *min_cluster_points* grande pode resultar em menos, mas maiores *clusters*, enquanto um *min_cluster_points* pequeno pode criar muitos *clusters* pequenos. Portanto, a escolha apropriada desses parâmetros é crucial para capturar a estrutura de densidade dos dados de maneira eficaz [11].

Figura 2: *Cluster DBSCAN* com *min_cluster_points* = 4. ε é referente ao *epsilon*.



Fonte: [12].

METODOLOGIA

Para aplicação do algoritmo, foi realizado um levantamento por *LiDAR* estacionário em uma sala de escritório de uma empresa. A nuvem gerada possui equipamentos e pessoas trabalhando, tendo em vista que a ideia é que a realização do levantamento não atrapalhe a produtividade e que, caso seja necessário excluir as pessoas da cena, será feito em um pós-processamento. Como o objetivo do artigo é justamente segmentar os elementos presentes, não foi necessário excluir nenhum objeto.

Figura 3: Nuvem de pontos utilizada no estudo.

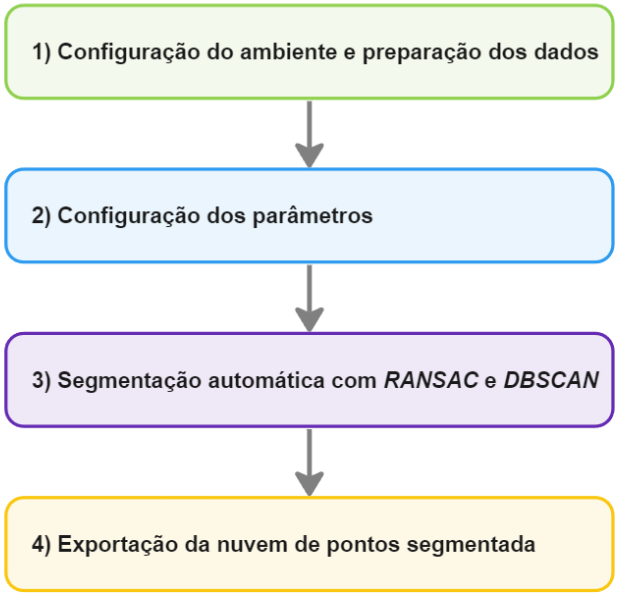


Fonte: Os autores.

O algoritmo foi aplicado por meio do *software VSCode* com base no estudo realizado por [13], utilizando a linguagem de programação *Python*. O estudo baseou-se em 3 etapas principais, que podem ser observadas na Figura 4, sendo a primeira etapa a configuração do ambiente de desenvolvimento, com exportação de todas as bibliotecas necessárias e preparação dos dados da nuvem de pontos. Na segunda etapa foi realizada a configuração das variáveis do algoritmo, que pode ser vista na Figura 5. Já na terceira etapa, foi realizada a segmentação automática pelo *RANSAC* e *DBSCAN* por meio do código e, com os resultados obtidos, foram feitos ajustes para que se chegasse aos parâmetros desejados. Na quarta e última etapa, foi incluído no

algoritmo a exportação da nuvem de pontos segmentada, para que ela possa ser utilizada em outros *softwares.*

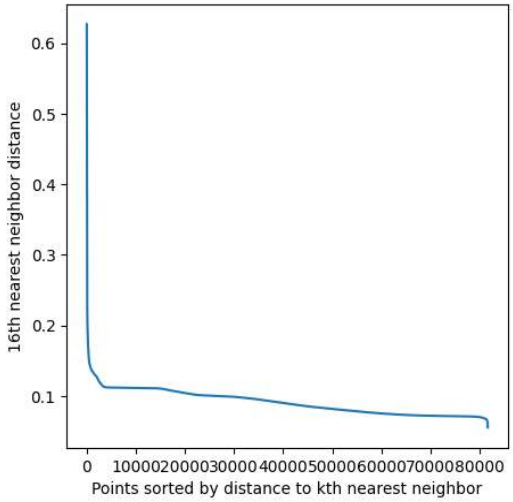
Figura 4: Fluxograma adotado no desenvolvimento do estudo.



Fonte: Adaptado de [13].

Para configuração das variáveis e otimização da escolha dos parâmetros a serem utilizados pelo *DBSCAN*, foi produzido um gráfico de correlação entre o *epsilon* e o *min_cluster_points*. Para determinar o valor do *epsilon*, calcula-se a distância média entre cada ponto e seus vizinhos mais próximos e em seguida, é traçada uma distância *k* e é escolhido o valor *epsilon* no "cotovelo" do gráfico. No eixo das ordenadas são plotadas as distâncias médias e no eixo das abscissas todos os pontos de dados no conjunto de dados. Sendo assim, foi observado que os valores ideais do *epsilon* a ser utilizado foi de aproximadamente 0.16.

Figura 5: Relação entre o *epsilon* e o *min_cluster_points*.



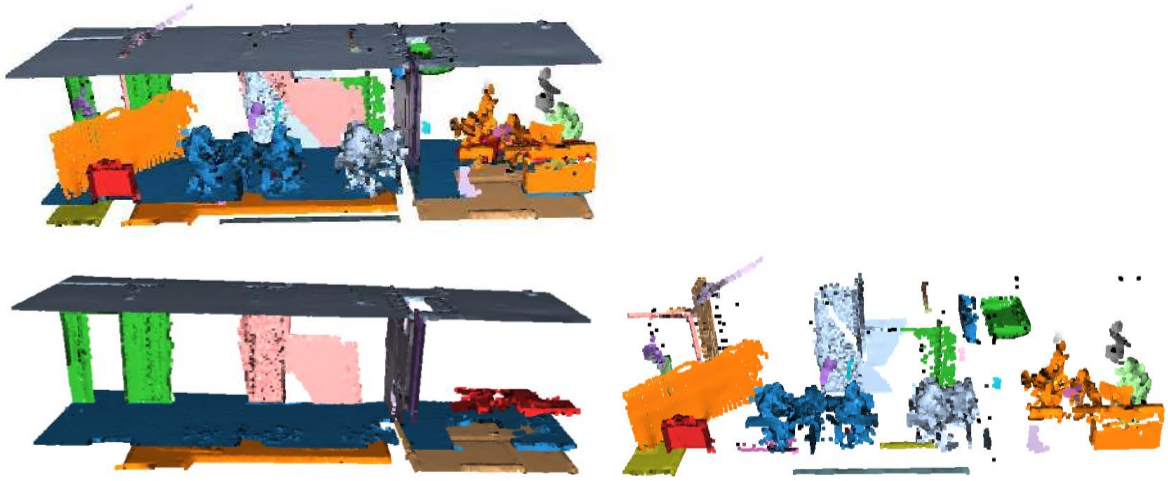
Fonte: Os autores.

Foram utilizados os seguintes parâmetros no algoritmo: *principal_min_cluster_points* = 20; *rest_epsilon* = 0.16; e *rest_min_cluster_points* = 6. Isso indica que, para a nuvem principal a ser gerada, foi escolhido um número mínimo de pontos no *cluster* de 20, enquanto na nuvem dos demais elementos (*rest*), foi escolhido um mínimo de pontos nos *clusters* de 6, pois quanto menor o número do parâmetro, mais *clusters* são gerados.

RESULTADOS

Nas nuvens de pontos obtidas a partir da clusterização da nuvem original, foram separados os elementos principais e os elementos restantes. As nuvens de pontos segmentadas foram visualizadas dentro do próprio VSCode, em uma interface que permite rotacionar e movimentar a nuvem gerada em todos os sentidos. Na nuvem principal (à esquerda abaixo), foram segmentadas as paredes presentes na nuvem, ainda que em alguns trechos a parede tenha sido cortada e tratada como elemento secundário. Isso pode indicar que algum tipo de diferença na iluminação do levantamento tenha feito o algoritmo reconhecer a mesma parede como mais de um elemento diferente, como pode ser observado na Figura 5, na parede em rosa.

Figura 5: Nuvem de pontos completa segmentada (à esquerda acima); elementos principais obtidos pela segmentação (à esquerda abaixo) e o "resto" da nuvem de pontos (à direita abaixo).



Fonte: Os autores.

Também foram segmentados de forma satisfatória os pisos, incluindo um forro do andar de baixo remanescente do levantamento; e o teto do escritório. Pode ser observado que havia a presença de alguns ruídos na nuvem de pontos na parte superior do teto, em roxo e verde, que não foram mantidos na nuvem principal. Um ponto interessante a ser observado é que o *cluster* vermelho na nuvem principal é na realidade uma mesa, que possivelmente foi interpretado pelo algoritmo como uma superfície, assim como o piso e o teto. Apesar desse erro, a maior parte dos elementos secundários foi separado da nuvem principal, indicando um bom resultado nos algoritmos utilizados.

CONCLUSÃO

O algoritmo de segmentação foi aplicado em uma nuvem de pontos "crua" e se mostrou eficaz para separar em *clusters* alguns objetos presentes no escaneamento. Foi obtido um bom resultado na segmentação de paredes, pisos e telhados, tendo em vista que, a nível de necessidades específicas para engenharia civil, a identificação desses elementos é bastante útil para a elaboração de *as-builts*. Foi possível observar que em um trecho mais ruidoso, em que havia presença de mesas e pessoas, a parte superior da mesa foi segmentada na mesma nuvem dos elementos principais, isso pode ter se dado porque o algoritmo interpretou uma grande concentração de pontos na horizontal como um piso. Apesar da falha de interpretação, a maior parte dos objetos que não seriam úteis para um *as-built* foi excluído da nuvem principal e tratado como o "resto".

Como contribuição realizada no estudo, a clusterização feita na nuvem de pontos original é transformada em uma nuvem de pontos que contém a informação de cada *cluster* obtido, permitindo que essa nuvem e seus elementos segmentados sejam visualizados separadamente em *softwares* específicos de nuvens de pontos.

REFERÊNCIAS

[1] KIM, Minju; LEE, Dongmin. Automated two-dimensional geometric model reconstruction from point cloud data for construction quality inspection and maintenance. **Automation in Construction**, v. 154, p. 105024, 2023. ISSN 0926-5805. Disponível em https://doi.org/10.1016/j.autcon.2023.105024. Acesso em 23 de Janeiro de 2024.

[2] BRASIL. Presidência da República. **Decreto nº 9.377, de 17 de maio de 2018**. Estratégia Nacional de Disseminação do Building Information Modelling no Brasil - Estratégia BIM BR. Diário Oficial da União, Brasília, DF, 17 maio de 2018. Disponível em https://www2.camara.leg.br/legin/fed/decret/2018/decreto-9377-17-maio-2018-786731-publicacaooriginal-155623-pe.html. Acesso em 09 de abril de 2024.

[3] CHEN, Hui et al. An approach to boundary detection for 3D point clouds based on DBSCAN clustering. **Pattern Recognition**, v. 124, p. 108431, 2022.

[4] Poux, F., & Billen, R. (2019). Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featuring vs deep learning methods. ISPRS I**nternational Journal of Geo-Information**. 8(5), 213; https://doi.org/10.3390/ijgi8050213

[5] GIONGO, Marcos et al. LiDAR: princípios e aplicações florestais. **Pesquisa Florestal Brasileira**, v. 30, n. 63, p. 231-231, 2010.

[6] KOTB, Ahmed; HASSAN, Safaa; HASSAN, Hesham. A comparative study among various algorithms for lossless airborne LiDAR data compression. In: 2018 14th International **Computer Engineering Conference (ICENCO)**. IEEE, 2018. p. 17-21.

[7] SEPASGOZAR, Samad ME; LIM, Samsung; SHIROWZHAN, Sara. Implementation of Rapid As-built Building Information Modeling Using Mobile LiDAR. In: **Construction Research Congress 2014: Construction in a Global Network.** 2014. p. 209-218.

[8] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. C**ommunications of the ACM**, 24(6):381–395, 1981. DOI: https://doi.org/10.1145/358669.358692

[9] ESTER, Martin et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **kdd**. 1996. p. 226-231.

[10] H. V. Singh, A. Girdhar and S. Dahiya, "A Literature survey based on DBSCAN algorithms," 2022 6th **International Conference on Intelligent Computing and Control Systems (ICICCS)**, Madurai, India, 2022, pp. 751-758, doi: https://doi.org/10.1109/ICICCS53718.2022.9788440.

[11] Lehtola, V., Nikoohemat, S., & Nüchter, A. (2020). Indoor 3D: Overview on scanning and reconstruction methods. **Handbook of Big Geospatial Data**, 55–97. https://doi.org/10.1007/978-3-030-55462-0_3

[12] Markus Götz, Christian Bodenstein, and Morris Riedel. 2015. HPDBSCAN: highly parallel DBSCAN. In Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments (MLHPC '15). **Association for Computing Machinery**, New York, NY, USA, Article 2, 1–10. https://doi.org/10.1145/2834892.2834894

[13] Poux, F.; Mattes, C.; Kobelt, L. Unsupervised Segmentation of Indoor 3D Point Cloud: Application to Object-Based Classification. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,** Volume XLIV-4/W1-2020, 2020.