

3DMSE: An Interactive 3D Media Search Engine

Maria Eirini Pegia
mpegia@iti.gr
CERTH-ITI
Thessaloniki, Greece
Reykjavík University
Reykjavík, Iceland

Dimitris Georgalis
dgeorgalis@iti.gr
CERTH-ITI
Thessaloniki, Greece

Nick Pantelidis
pantelidisnikos@iti.gr
CERTH-ITI
Thessaloniki, Greece

Björn Þór Jónsson
bjorn@ru.is
Reykjavík University
Reykjavík, Iceland

Anastasia Moutzidou
moutzid@iti.gr
CERTH-ITI
Thessaloniki, Greece

Sotiris Diplaris
diplaris@iti.gr
CERTH-ITI
Thessaloniki, Greece

Ilias Gialampoukidis
heliasgj@iti.gr
CERTH-ITI
Thessaloniki, Greece

Stefanos Vrochidis
stefanos@iti.gr
CERTH-ITI
Thessaloniki, Greece

Ioannis Kompatsiaris
ikom@iti.gr
CERTH-ITI
Thessaloniki, Greece

ABSTRACT

We present the 3D Media Search Engine (3DMSE), which is designed to facilitate the exploration and retrieval of 3D models and images. 3DMSE incorporates unimodal, cross-modal and multi-modal retrieval, using any combinations of mesh, point-cloud and multi-image representations. The 3DMSE system is built on the recently proposed MuseHash approach for multimodal representation, and offers a user-friendly web interface that enables formulating queries, presenting search results, and visualising 3D information in an accessible manner.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Information retrieval**; **Search Interfaces**; • **Human-centered computing** → **Visualization systems and tools**; *Visualization systems and tools*.

KEYWORDS

Multimodal Image Retrieval, 3D Data, Visualization

ACM Reference Format:

Maria Eirini Pegia, Dimitris Georgalis, Nick Pantelidis, Björn Þór Jónsson, Anastasia Moutzidou, Sotiris Diplaris, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2024. 3DMSE: An Interactive 3D Media Search Engine. In *Proceedings of the 2024 International Conference on Multimedia Retrieval (ICMR '24)*, June 10–14, 2024, Phuket, Thailand. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3652583.3657593>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICMR '24, June 10–14, 2024, Phuket, Thailand

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0619-6/24/06...\$15.00

<https://doi.org/10.1145/3652583.3657593>

1 INTRODUCTION

In recent years, advancements in 3D modeling tools [16], scanning technology [12], and consumer devices with 3D sensors [4] have transformed the accessibility of vast 3D content collections, influencing various domains, from entertainment and gaming to healthcare [25], archaeology [1], computer-aided design (CAD) [7] and autonomous systems [6]. These changes have not only made it easier for anyone to create 3D content but also radically changed industries by enabling faster design processes and helping make better decisions.

There are various 3D applications specifically designed for retrieval and management. CAD software (e.g., Autodesk AutoCAD [9], SolidWorks [15], and CATIA [21]) often include built-in features for managing and retrieving 3D data. Geospatial Information Systems (GIS, e.g., Esri ArcGIS [23] and QGIS [3]) also offer tools for managing and visualizing 3D geospatial data. Extended Reality (VR/AR/MR) platforms (e.g., Unity [14] and Unreal Engine [19]) offer built-in features for managing and retrieving 3D data. Efficiently navigating this abundance of models, however, remains a challenge.

Techniques for retrieving 3D models play a crucial role in various applications, allowing users to find relevant models based on classification or similarity criteria [5, 8, 13, 17, 20, 24]. The 3D data can be represented in different forms: as point clouds, which consist of points within the model's spatial domain; as meshes, represented by interconnected triangles to approximate the shape; or as multiview images, comprising a set of images captured from various viewpoints of the 3D shape. Effectively handling these diverse 3D representations is considered as modalities and has been addressed using the MuseHash approach. This demonstration paper described the 3DMSE search engine, which builds a user-friendly web-based interface on top of the MuseHash approach.

The remainder of this paper is organised as follows: Section 2 presents MuseHash, the state-of-the-art multimodal retrieval method underlying 3DMSE, while Section 3 details the UI and its

features. Section 4 presents two example retrieval scenarios using the 3DMSE system. Finally, Section 5 provides the conclusion.

2 SIMILARITY SEARCH WITH MUSEHASH

The 3DMSE system integrates the advanced multimodal retrieval method MuseHash with an interactive Graphical Interface (GUI), enabling browsing in supported 3D collections and various data and retrieval types. The current system, 3DMSE, boasts compatibility with a wide array of data types, including meshes (consisting of interconnected triangles to approximate the shape), point clouds (comprised of data points of significance within the model's spatial domain), and images, offering versatility for users' needs. The retrieval types in 3DMSE encompass uni-modal, cross-modal, and multimodal approaches, providing users with diverse options for searching and interacting with 3D models. Uni-modal retrieval focuses on a single data modality, cross-modal retrieval enables searching one data modality based on a query in a different data modality, while multimodal retrieval integrates multiple data modalities for comprehensive results.

The retrieval module utilises MuseHash [17, 18], a Bayesian-based method for retrieving visually similar content across various modalities. It employs Bayesian techniques to learn hash functions, enabling retrieval across unimodal, cross-modal ($M = 1$), and multimodal scenarios ($M > 1$). The visual modality involves averaging 180 feature vectors from ResNet50's fc-7 layer [11], resulting in a 2048-D vector. For point cloud and mesh modalities, 256-D vectors are directly obtained from DGCNN [24] and MeshNet [5]. These feature vectors are transformed into hash codes using learned hash functions. In the multimodal case, hash codes across different modalities are fused into a single code C_i using the Equation below, where $H_{i,j}$ is the hash code of i -th instance and j -th modality, with $\oplus$ denoting the XOR operation between hash codes. Finally, retrieved items are ranked based on Hamming distance:

$$C_i = f(H_{i,1}, H_{i,2}, \dots, H_{i,M}) = \sum_{s=1}^{s=M} \sum_{t=1}^{t=M} H_{i,t} \oplus H_{i,s}$$

3 USER INTERFACE AND SEARCH MODES

3DMSE system integrates two public benchmark 3D datasets; the BuildingNet_v0 dataset [22] that contains different building models (e.g., church, hotel) and the ShapeNetCore dataset [2] that includes different object models (e.g., airplane, car). The system provides users with the capability of locally storing data (Section 3.1), empowers users to seamlessly navigate through the stored data (Section 3.2) and retrieve specific information directly from the database (Section 3.3).

In 3DMSE, a robust technology stack ensures seamless functionality and user experience. Back-end data processing is implemented in PHP, using MongoDB [10] to securely store model information. Each MongoDB collection corresponds to a specific dataset (e.g., ModelNet40, BuildingNet_v0), organizing documents that represent individual 3D models. These documents include information such as the file name, the category, and metadata for the types of models (mesh, point cloud, image) each dataset supports. The intuitive front-end interface is implemented with HTML/CSS/JavaScript, using Google's model-viewer.js to render mesh models for an immersive

viewing experience, while Three.js facilitates dynamic interaction with point cloud models.

Figure 1 shows the starting screen of the interface. Upon entering 3DMSE, users see the initial results panel that displays snapshots of the models. At the top-right of this component is a drop-down list from which users can navigate to any of the available datasets (currently, BuildingNet_v0 and ShapeNetCore) and view their respective models. In the same corner is another drop-down list that enables users to select among the three modality types available for each dataset: Mesh, Point Cloud and Image. A pagination control is also provided, in the top-left corner, which ensures easy navigation, enabling transitions between pages that contain 100 models each. These three dropdown lists are featured in Figure 1.

3.1 Local Data Storage

When hovering over a snapshot, users can view the model's name and two buttons, as shown in Figure 2. The download button enables users to choose and download either the snapshot (in PNG format) or the model itself (in GLTF format for mesh models or PLY format for point cloud models). Another method for downloading data while browsing is discussed in Section 3.2. The left button is a part of the retrieval process and is explained in Section 3.3. These details are illustrated in Figure 2, where the model of interest is enclosed within a red square, aiding users in navigating the interface effectively.

3.2 Browsing

By clicking on the snapshot of a model, a model viewer appears. This component provides a dynamic space where users can interact with 3D models (rotate and zoom), or view images in full-screen (in case the selected type from the dropdown is 'Image'). It also contains a button for downloading the model (with similar options as before). Examples of the 3D model viewer for a mesh model and for a point cloud model are illustrated in Figures 4 and 3, respectively.

3.3 Retrieval

By clicking the left button mentioned earlier (Figure 2 in Section 3.1), users initiate the appearance of a popup window, as depicted in Figure 5. This window offers users the option to select the types of models for retrieval, whether it be uni-modal, cross-modal, or multimodal. For instance, if a user selects Mesh for similarity, the system will retrieve mesh models. Alternatively, choosing Mesh for uni-modal retrieval, Point cloud for cross-modal retrieval, or more than two modalities for multimodal retrieval is also possible. Upon clicking the search button, a query is dispatched to the MuseHash service (Section 2). This service then returns models of the selected type(s) similar to the chosen one, ranked from most to least relevant. Subsequently, the system opens the similar results panel (Figure 6) for further exploration by the user.

In the similar results panel, users can explore models that closely relate to the selected one. Each displayed model within this panel provides users with access to the same set of buttons and functionalities available in the initial results panel, ensuring consistent interaction capabilities. Positioned prominently at the top of the panel is a snapshot of the selected model, allowing users to quickly

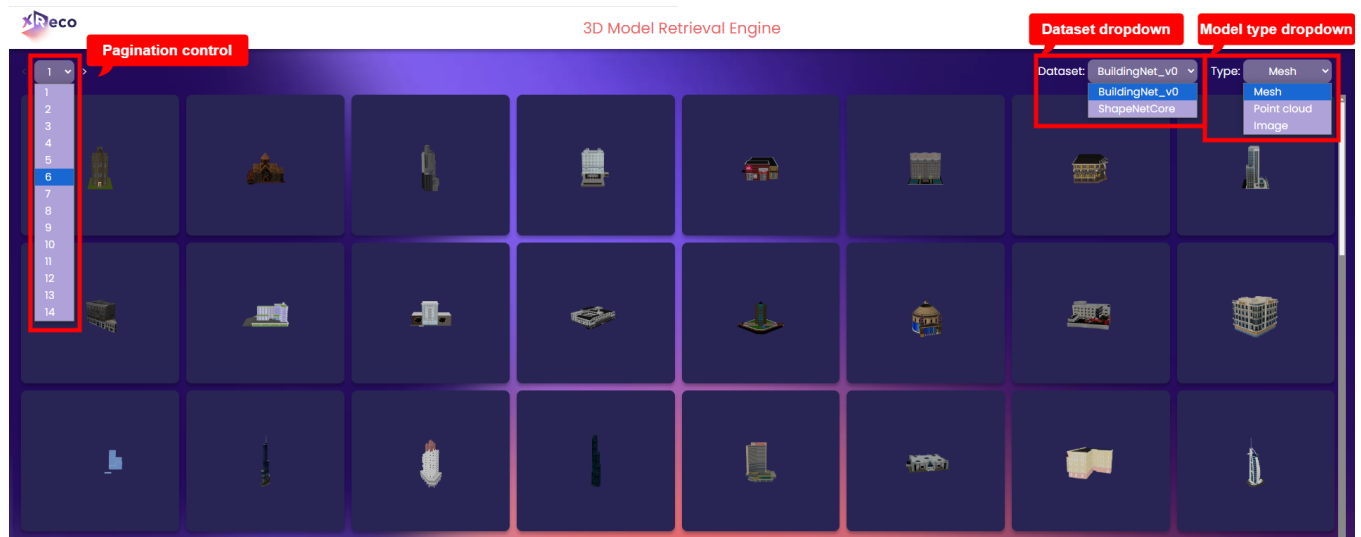


Figure 1: The initial results panel and its dropdown lists.



Figure 2: Hovering over a snapshot and available buttons.

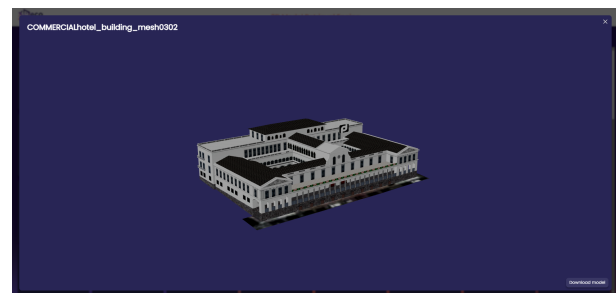


Figure 4: The 3D model viewer for a mesh model.

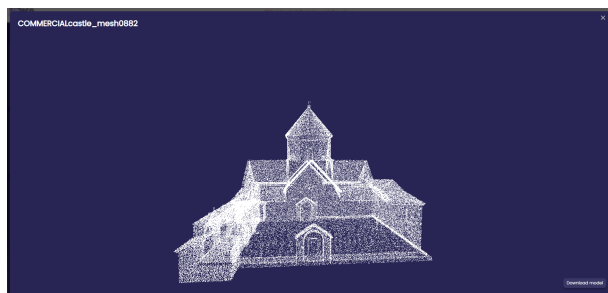


Figure 3: The 3D model viewer for a point cloud model.

identify it. Alongside the snapshot, the model's name and associated category or categories are provided, offering additional context and information to the user.

4 EXPLORING SMALL SCENARIOS

To illustrate the functionality of 3DMSE, we present two scenarios; one *uni-modal* and one *multimodal*. We provide a comprehensive explanation of the uni-modal scenario and a brief description of the multimodal scenario.

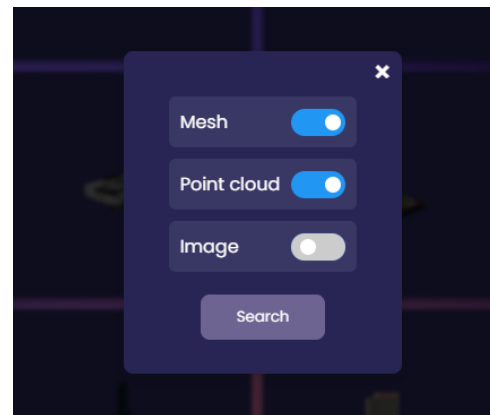


Figure 5: The types' options popup window.

4.1 Uni-modal Scenario

In this scenario, the objective is to pinpoint a particular building (identified as "COMMERCIALhotel_building_mesh1286"). This specific structure is described as being tall and slender, composed of

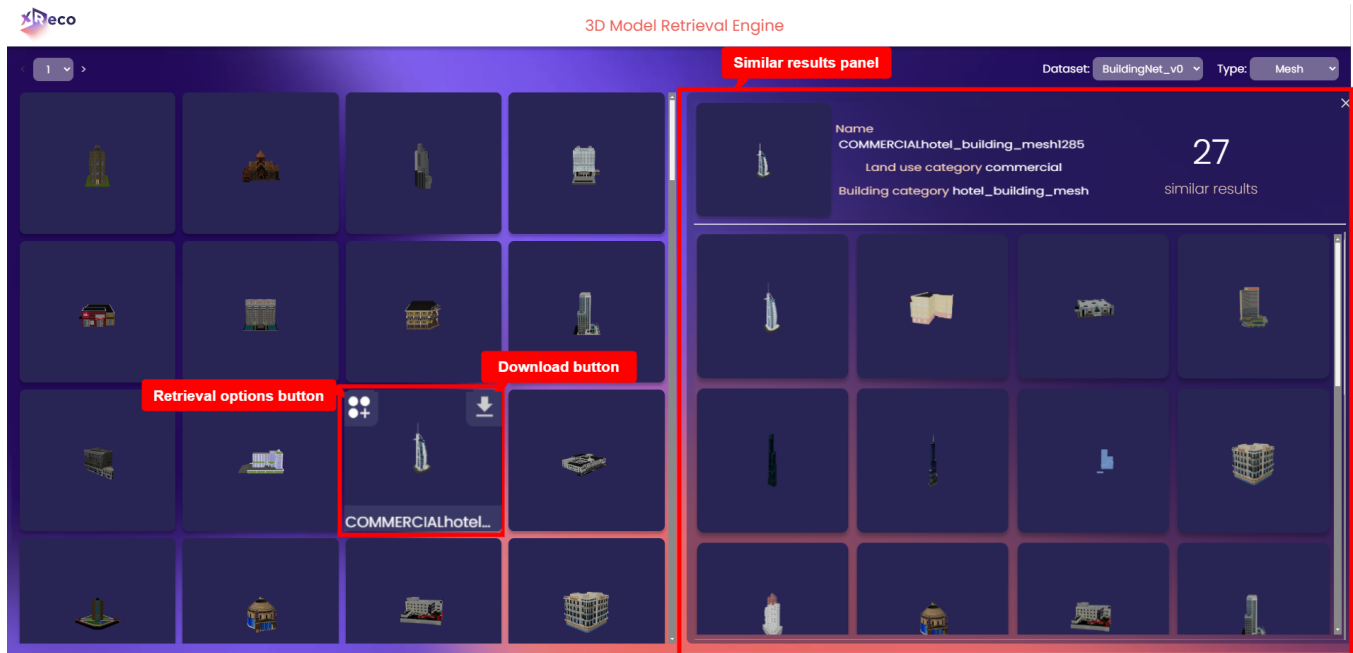


Figure 6: The similar results panel and a model's details.

mesh. Below, we provide an exhaustive, detailed guide delineating each step necessary to successfully accomplish this task:

- (1) Begin by accessing the interface and changing the dataset to BuildingNet_v0 and the model type to Mesh using the respective popup lists.
- (2) Navigate to the left popup menu and select page 1 to modify the interface's display.
- (3) Utilize the selection mechanism on the UI's interface to pick a tall and narrow building from the list, such as the building with name "COMMERCIALhotel_building_mesh0555".
- (4) Click the icon positioned at the top left corner of the interface to initiate a potential search and choose Mesh from the types' option popup window.
- (5) A list of relevant results will be presented. Use your mouse to explore this list and locate the building image named "COMMERCIALhotel_building_mesh1285" in the 1st row, 1st column.
- (6) Click on the image "COMMERCIALhotel_building_mesh1285" to set it as your query and choose again Mesh from the types' option popup window.
- (7) In the returned list, find the target image with the name "COMMERCIALhotel_building_mesh1286" in the 1st row, 1st column, and click on it to access additional details or related information.

This detailed guide is intended to facilitate a smooth navigation experience through the interface, assisting users in efficiently locating the designated mesh hotel building. Given that a uni-modal scenario has been detailed provided, we offer for completeness a concise multimodal scenario below.

4.2 Multimodal Scenario

Your objective now is to find the same building as in *Uni-modal scenario*, but now taking advantage of the mesh and point cloud modalities. Similarly as before, we pick the building with name "COMMERCIALhotel_building_mesh0555" and choose Image and Mesh from the types' option popup window. A list of relevant results will be presented. In the 4th row and 2nd column you will find the target "COMMERCIALhotel_building_mesh1286". Therefore, by utilising two modalities instead of one, we have reached the target in one step fewer.

5 CONCLUSIONS

In summary, our investigation demonstrates that the 3DMSE system effectively assists users in retrieving and managing 3D data. By combining the MuseHash retrieval approach with a user-friendly interface, 3DMSE offers a straightforward way for users to explore and interact with different types of 3D models. Its versatility in handling various data formats makes it a potentially valuable tool across different application domains.

6 ACKNOWLEDGMENTS

This work was supported by the Horizon research and innovation programme under grant agreement HE-101070250 XRECO.

REFERENCES

- [1] Mauro Lo Brutto and Paola Meli. 2012. Computer Vision Tools for 3D Modelling in Archaeology. In *hde*.
- [2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. ShapeNet: An Information-rich 3D Model Repository. *arXiv preprint arXiv:1512.03012* (2015).

- [3] Luca Congedo. 2021. Semi-Automatic Classification Plugin: A Python Tool for the Download and Processing of Remote Sensing Images in QGIS. *Journal of Open Source Software* 6, 64 (2021), 3172.
- [4] Matthew M. Dummer, K. L. Johnson, Sara Rothwell, Karim Tatah, and Mary K. Hibbs-Brenner. 2021. The Role of VCSELS in 3D Sensing and LiDAR. In *OPTO*.
- [5] Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao. 2019. MeshNet: Mesh Neural Network for 3D Shape Representation. In *AAAI*. Honolulu, Hawaii.
- [6] QP Ha, L. Yen, and C Balaguer. 2019. Robotic Autonomous Systems for Earth-moving in Military Applications. *Automation in Construction* 107 (2019).
- [7] Young-Soo Han, Jaejoon Lee, Jungmin Lee, Wonhyuk Lee, and Kyungcho Lee. 2019. 3D CAD Data Extraction and Conversion for Application of Augmented/Virtual Reality to the Construction of Ships and Offshore Structures. *International Journal of Computer Integrated Manufacturing* 32, 7 (2019), 658–668.
- [8] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: A Network with an Edge. *ACM Transactions on Graphics (ToG)* 38, 4 (2019), 1–12.
- [9] L Scott Hansen. 2023. Autodesk Inventor 2024: A Tutorial Introduction. (2023).
- [10] Guy Harrison and Michael Harrison. 2021. *MongoDB Performance Tuning: Optimizing MongoDB Databases and Their Applications*. Springer.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
- [12] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, and Rajiv Suman. 2021. Industrial Perspectives of 3D Scanning: Features, Roles and it's Analytical Applications. *Sensors International* 2 (2021).
- [13] Longlong Jing, Elahe Vahdani, Jiaxing Tan, and Yingli Tian. 2021. Cross-Modal Center Loss for 3D Cross-Modal Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3142–3151.
- [14] Kaushal Kishor, Rupa Rani, Atul Kumar Rai, and Varsha Sharma. 2023. 3D Application Development Using Unity Real Time Platform. In *Doctoral Symposium on Computational Intelligence*. Springer, 665–675.
- [15] John Matsson. 2023. *An Introduction to SOLIDWORKS Flow Simulation 2023*. SDC publications.
- [16] Elisa Mohr, Thomas Thum, and Christian Bär. 2022. Accelerating Cardiovascular Research: Recent Advances in Translational 2D and 3D Heart Models. *European Journal of Heart Failure* 24, 10 (2022), 1778–1791.
- [17] Maria Pegia, Björn Þór Jónsson, Anastasia Moutzidou, Sotiris Diplaris, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2024. Multimodal 3D Object Retrieval. In *International Conference on Multimedia Modeling*. Springer, 188–201.
- [18] Maria Pegia, Björn Þór Jónsson, Anastasia Moutzidou, Ilias Gialampoukidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2023. MuseHash: Supervised Bayesian Hashing for Multimodal Image Representation. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*. 434–442.
- [19] Fabian Plum, René Bulla, Hendrik K Beck, Natalie Imirzian, and David Labonte. 2023. replicAnt: A Pipeline for Generating Annotated Images of Animals in Complex Environments using Unreal Engine. *Nature Communications* 14, 1 (2023), 7195.
- [20] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*. Honolulu, HI, USA.
- [21] José Ignacio Rojas-Sola, Gloria del Río-Cidoncha, Rafael Ortiz-Marín, and José Antonio Moya-Ocaña. 2023. Design and Development of a Macro to Compare Sections of Planes to Parts Using Programming with Visual Basic for Applications in CATIA. *Symmetry* 15, 1 (2023), 242.
- [22] Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2021. BuildingNet: Learning to Label 3D Buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10397–10407.
- [23] Ellen Soward and Jianling Li. 2021. ArcGIS Urban: An Application for Plan Assessment. *Computational Urban Science* 1 (2021), 1–10.
- [24] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (tog)* 38, 5 (2019), 1–12.
- [25] Martin J Willemink, Wojciech A Koszek, Cailin Hardell, Jie Wu, Dominik Fleischmann, Hugh Harvey, Les R Folio, Ronald M Summers, Daniel L Rubin, and Matthew P Lungren. 2020. Preparing Medical Imaging Data for Machine Learning. *Radiology* 295, 1 (2020), 4–15.