Review of Surface Reconstruction: From Classical Methods to Neural Radiance Fields

João Francisco S. S. Lemos[1], Igor P. Maurell[1], Kristofer S. Kappel[1], Gabriel A Dorneles[1], Stephanie L. Brião[1], Felipe G. Oliveira[1,2], Nelson L. Duarte Filho[1], Rodrigo da S. Guerra[1] and Paulo L. J. Drews-Jr[1]

***Abstract*—This paper provides an overview of surface reconstruction methods. It ranges from classical geometric algorithms to advanced deep learning techniques, which offer greater flexibility by leveraging neural networks to handle complex and high-dimensional data. It involves creating a digital 3D model from real-world data and capturing the geometric details of objects or scenes. The intersection with robotics comes by providing a world model for long-horizon tasks and a better understanding of the attributes of each object in the scene, enabling a more efficient data collection for open-vocabulary planning. In addition, we introduce key concepts that intersect with this topic, offering a better understanding of the techniques and the state of the field. Considering this, a significant focus is placed on methods that represent scenes in a five-dimensional space, enabling the synthesis of highly accurate new views from 2D input data. Considering this, the present paper aims to portray an overview of three-dimensional surface reconstruction, highlighting some of the main turning points in the field of research.**

***Index Terms*—Surface Reconstruction, Panoptic Segmentation, NeRF, Deep Learning.**

I. Introduction

Over the past few decades, the importance of 3D representation in computers has increased exponentially, as has the effectiveness of methods to achieve it. This 3D data representation is vital in multiple applications, including simulations, robotics, machine learning, and other tasks. Robotic simulations, in particular, rely heavily on accurate 3D representations to model real-world environments and behaviors, providing a crucial link between theoretical robotics and practical applications. Such simulations also serve as valuable tools for learning, offering safe and controlled environments for experimentation and skill acquisition. The diverse range of digital data types is essential for transmitting, storing, and processing information in these digital environments [1].

Surface reconstruction is a long-standing challenge in robotics, aiming to create accurate geometric representations of objects or scenes from a sample of real-world data. This process allows for the digital representation of an object's surface with high geometric and spatial precision [2].

*This work was partly supported by FINEP, ANP-PRH22, FAURG, FAPERGS, FAPEAM and CNPQ.

[1]Centro de Ciências Computacionais (C3). Universidade Federal do Rio Grande, Rio Grande - RS, Brasil. {jfsslemos, igormaurell, kris, stephanie.loi, dmtnldf, paulodrews, rodrigo.guerra}@furg.br

[2]Instituto de Ciências Exatas e Tecnologia – ICET. Universidade Federal do Amazonas – UFAM. Itacoatiara, AM, Brazil. felipeoliveira@ufam.edu.br

979-8-3315-5288-6/25/$31.00©2025 IEEE

Traditional surface reconstruction methods, such as Poisson- and geometry-based techniques, have proven effective in many scenarios. However, the advent of deep learning-based methods opened new possibilities by enabling more flexible reconstructions that can handle complex and high-dimensional data. These methods leverage the power of neural networks to learn intricate patterns and 3D data representations, leading to significant advancements in the quality and efficiency of surface reconstruction. The high applicability of these methods has helped the development of sim-to-real applications, further increasing the learning capabilities of robotic systems [3].

Considering this, the current paper aims to portray a review of surface reconstruction and its methods, highlighting some of the main points in the field of research, the applications in robotics and paving the way for future works.

The current paper is divided as follows: Section II presents an overview of the main formats for digital representation of 3D and surface data. Section III explains the concepts for surface reconstruction. Section IV describes surface reconstruction methods like classics and deep learning. Section V shows the applications of surface reconstruction in the robotics area. Finally, Section VI summarizes the discussion.

II. Formats for digital data representation

Digital data representation formats encompass various ways to represent information, from plain text to complex media like audio, video, and images. This diversity is crucial for transmitting, storing, and processing data.

The nature of the data determines the most suitable format for representation. Regarding geometry, there is a distinction between Euclidean and non-Euclidean data. Euclidean data adhere to classical geometric principles, with distances measured linearly using the Euclidean metric. Non-Euclidean data [4], on the other hand, deviate from these rules, employing alternative distance metrics, such as measuring distances along great-circle arcs on spherical surfaces. Euclidean data are often associated with simple geometric forms, while non-Euclidean data involve more complex contexts and high-dimensional spaces. Point clouds are collections of Cartesian coordinates representing positions in space. Their typical sources are LiDAR sensors or photogrammetry techniques [5]. Meshes [6] are fundamental structures in computer graphics and 3D modeling. They represent three-dimensional surfaces using polygons, particularly triangles. Triangular meshes are simple and efficient, storing information such as spatial position, color, texture, and normals.

III. Surface Reconstruction

Surface reconstruction [7] is a complex and long-standing challenge in the field of computer vision, aiming to create accurate and detailed geometric representations of objects or scenes from a set of real-world data using two-dimensional or three-dimensional information captured by sensors like 3D cameras or laser scanners. These data are processed to create detailed models of the object's geometry, enabling the digital representation of its surface with high geometric and spatial accuracy using the following general concepts.

A. Convex Hull

A significant concept for some methods that extend these algorithms for surface reconstruction is the so-called Convex Hull, which represents the smallest convex envelope containing all the points in a set [8]. To delineate a specific area, the algorithms implementing this concept draw lines between all neighboring points, creating edges, and subsequently connect these edges to neighboring edges, forming a hollow shell, as shown in Fig. 1.

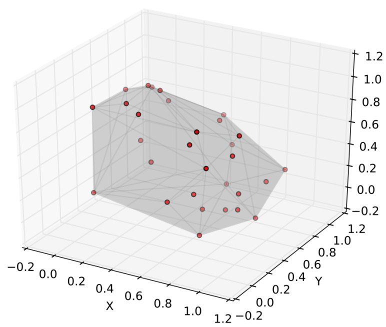

Fig. 1: Set of red points enclosed by the smallest possible solid in gray [9].

B. Normal Vector

Another essential concept for numerous methods is the surface normal vector around a point [10]. In three-dimensional geometry, the normal is the direction perpendicular to a surface around a given point. For each point, there is an associated orthogonal vector representation pointing outward. Fig. 2 shows the normal vector for multiple points in the center of each cell.

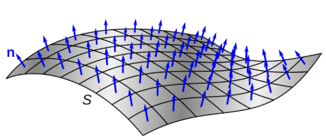

Fig. 2: Surface with computed normals [11].

IV. Surface Reconstruction Methods

This section explores methods for reconstructing surfaces like classical methods and learning approaches.

Classic Methods

The principle of surface reconstruction in mesh structures is one of the most significant and well-developed areas in computing, with various classical approaches aiming at broad generalization through algorithms such as Delaunay [12], Marching Cubes [13], and RANSAC [14], as noted by de Berg et al. [4].

A. Alpha Shapes

Building on this line of algorithms, methods like Alpha Shapes [15] emerge, a generalization of Convex Hull III-A based on the alpha parameter, denoted α. This parameter defines the radius or threshold of the maximum distance between points, which, once exceeded, becomes a boundary, as demonstrated in Fig. 3.

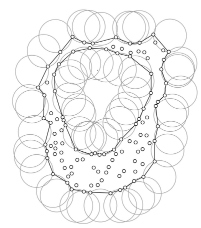

Fig. 3: Example where unfilled circles have their origin points determined as a boundary [16].

The advantages of Alpha Shapes are their low computational cost and ability to represent and extract complex topological information, such as cavities and holes. However, the definition of the value α can impact the resulting structure. Additionally, in datasets with uneven point distribution, the reconstruction quality tends to form imprecise or incomplete structures in certain areas, as demonstrated in Fig. 4.

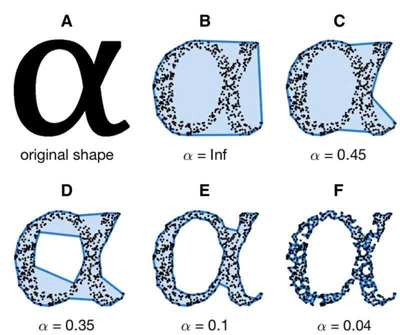

Fig. 4: Result variation as an alpha parameter function [5].

B. Ball Pivoting

Another relevant classical method is Ball Pivoting [17], an interpolation method for polygonal meshes based on a sphere of radius ρ and the iterative pivoting of this sphere until it is no longer possible.

In simple terms, the analogy for the algorithm's operation is that the set of points M represents the three-dimensional surface of an object, and N is a subset of points from M, assuming that N is dense enough so that a sphere of radius ρ does not pass through when placed on top. Consequently, a sphere placed on a set of three points, without losing contact with two of these points, is translated until it touches a third point again, as shown in Fig. 5.

A valuable feature of Ball Pivoting is that it handles noise and data gaps. The method accounts for the normals of the points, allowing it to check the consistency of a triangle's direction to determine whether or not to form the polygon, as shown in Fig. 5.

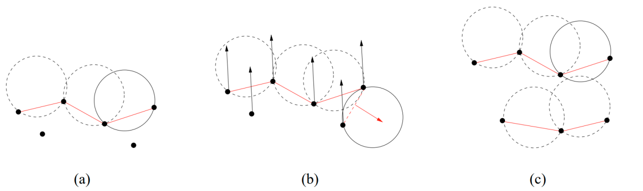

Fig. 5: (a) Sparse points resulting from noise are not considered if the overlapping set is dense enough. (b) The sphere's movement creates a link inconsistent with the adjacent normals, and the method prevents the face from closing. (c) The lower layer is sufficiently distant from the upper layer, allowing two surfaces to be formed in the end [17].

Ball Pivoting performs well in various cases, but it encounters difficulties in forming meshes with sharp corners, point clouds with drastic differences in sparsity, or surface roughness, as the sphere's ρ coefficient can make certain areas inaccessible for pivoting, resulting in a loss of detail in the final mesh.

C. Poisson

Finally, one of the later methods in the classical literature is the Poisson method [18], which solves a regularized optimization problem to obtain a smooth surface. This quality makes it preferable compared to the previously described methods, which convert the points of a cloud directly into mesh vertices without any modification.

The distinguishing feature of the Poisson method lies in recognizing the integral relationship between the oriented points in the surface sample of a model and its indicator function. Specifically, the gradient of the indicator function is a matrix predominantly populated by zeros (since the function is constant in most of the matrix), except at points relatively close to surfaces, where the function's value corresponds to the inward normal. Fig. 6 illustrates this analysis.

The primary strength of the Poisson method is that the reconstruction aims to be fully closed, assuming that the structure has no holes or empty regions. This characteristic makes it less sensitive to incomplete scans and ineffective for applications with hollow spaces or sharp edges.

Segmentation Methods

In this context, segmentation refers to dividing or grouping elements into smaller subsets, identifying regions of interest

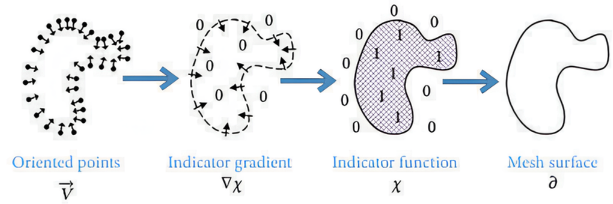

Fig. 6: Illustration of Poisson reconstruction in 2D [18].

within an image, a point cloud, or any other data representation.

D. Semantic Segmentation

Semantic segmentation [19], detailed in Fig. 7, is a process that assigns labels to each pixel in an image, indicating the semantic set to which it belongs. For example, in a scene, semantic segmentation can distinguish between objects such as cars, trees, roads, and people, assigning a label to each corresponding region in the image.

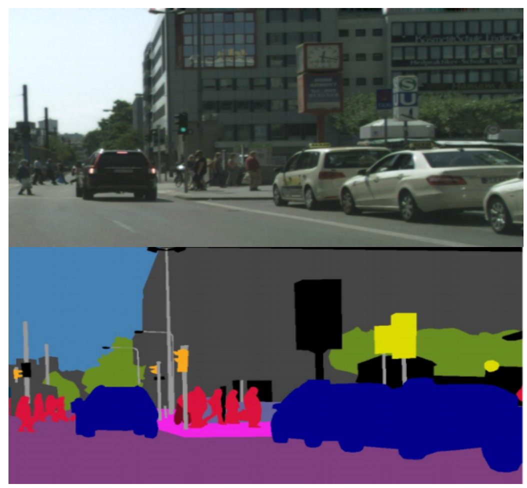

Fig. 7: Base image (above); Semantic Segmentation (below) [20].

E. Instance Segmentation

Instance segmentation [21], detailed in Fig. 8, goes beyond semantic segmentation by categorizing the semantic class and distinguishing each unique instance within those categories. Uniquely, only the classes described as relevant will have a category assigned to them. For example, in a scene with several cars, instance segmentation would assign unique identifiers to each one, allowing for the individual tracking of these objects over time.

F. Panoptic Segmentation

Finally, panoptic segmentation [22, 23], detailed in Fig. 9, unifies semantic and instance segmentation into a single result, providing both semantic and instance labels for all pixels in the

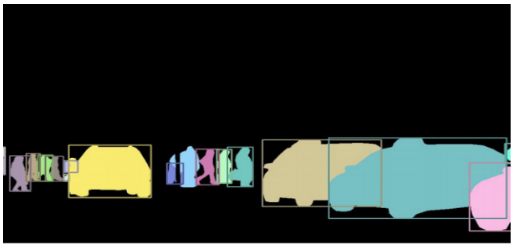

Fig. 8: Instance Segmentation [20].

image. Moreover, similar to instance segmentation, the concept of using relevant classes to determine whether a pixel should be classified exists. However, panoptic segmentation classifies all pixels semantically. Those not belonging to relevant classes do not receive a unique instance label. For example, in a domestic context, in a kitchen, chairs, which are relevant classes, would be labeled as ”chair 1”, ”chair 2,” and ”chair 3,” whereas pixels of the floor, which are not a class of interest, would only receive the label ”floor”.

Fig. 9: Panoptic Segmentation [20].

Deep Learning-based Methods

In recent years, the advancement and application of Deep Learning techniques have significantly revolutionized various fields of computer science, including computer vision and three-dimensional surface reconstruction. In particular, the use of Deep Learning methods for surface reconstruction from sparse data, as discussed by Chen in [24], has gained prominence due to their ability to handle complex and high-dimensional information.

Traditional surface reconstruction methods and geometry-based techniques have proven effective in many scenarios. However, Deep Learning offers an alternative and promising approach to surface reconstruction, enabling the learning of intrinsic data representations without heavily relying on heuristics or explicit modeling.

G. ShapeFlow

ShapeFlow [25] is a method based on the ”Retrieve and deform template” approach, which finds a similar template and deforms it accordingly.

The ShapeFlow model has three major stages: forming and learning a deformation-sensitive space by vectorizing various objects considered elemental for the application (embedding), loading the input data into this space, and identifying the most similar shape to the object using a nearest-neighbor classifier, and finally, a deformation neural network operates on the already learned structure, seeking the best geometric correspondence. Fig. 10 illustrates this process.

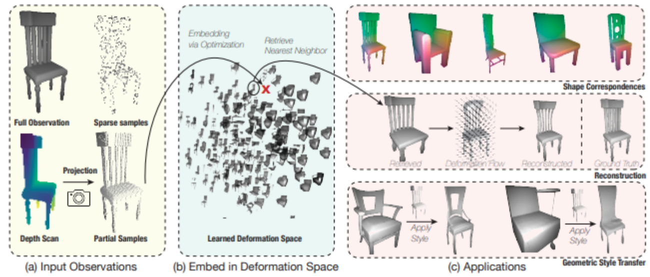

Fig. 10: Illustration of the chronological sequence of the deformation space learning process. (a) Input is a sparse point cloud or the conversion of a depth map into a point cloud. (b) Visualization of the base object library with the estimation of the input data's position. (c) The unsupervised deformation network acting on the reconstruction [25].

ShapeFlow offers a flow-based model capable of reconstructing objects with fidelity regarding volume, isometry, and symmetry. However, it encounters issues due to the limitation of the base structure group and the lack of semantic supervision, as the input cloud will always be tied to a template, regardless of how different they may be.

H. Scan2Mesh

Scan2Mesh [1] is a neural network-based reconstruction method that is not limited to point clouds, as it focuses on the generation and connection of vertices, which means there is no direct translation from a point in the cloud to a vertex in the mesh.

The procedure starts with an MLP-type point cloud generator to produce a set of points. Next, the method creates a fully connected graph between these points. Then, a graph neural network predicts which edges should exist in the final mesh. Finally, accounting for the predicted edges, it calculates all possible triangular faces, constructing a dual graph, and a second graph neural network determines which faces belong to the final mesh. Fig. 11 illustrates the process.

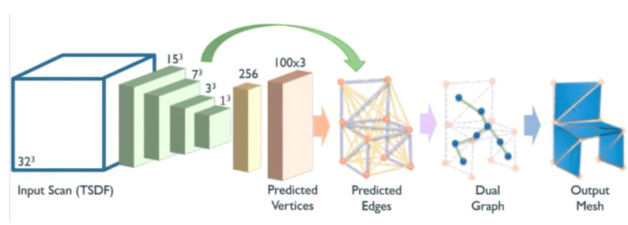

Fig. 11: Visual representation of Scan2Mesh [1].

Therefore, avoid hollow structures and handle sharp edges well, as shown in Fig. 12. However, the high computational cost due to the different stages of the process significantly limits the maximum number of vertices.

Fig. 12: Input data (left), Scan2Mesh inference (center), expected result (right) [1].

I. Neural Radiance Field

Neural Radiance Field, introduced by Mildenhall et al. 2021, is a deep learning-based method for reconstructing a three-dimensional representation of a scene from sparse two-dimensional images. The NeRF model learns the geometry of the scene, camera poses, and reflective properties, achieving state-of-the-art results in synthesizing new views of complex scenes by optimizing a continuous volumetric function using only a sparse set of input images.

The method renders a scene using a fully connected deep network that takes only a continuous coordinate, composed of the spatial location (x, y, z) and the viewing direction (θ, ϕ), outputting the volumetric density and the radiance emitted from the respective 3D point. Figure 13 illustrates this simplified logic.

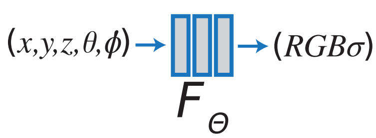

Fig. 13: Logic of the NeRF operation [26].

The views are synthesized using classical volume rendering techniques by queuing the 5D coordinates along the camera's orthogonal rays, projecting the output colors and densities onto an image, as exemplified in Fig. 14. Given that volumetric rendering is differentiable, the input required to optimize the function is the five-dimensional coordinates.

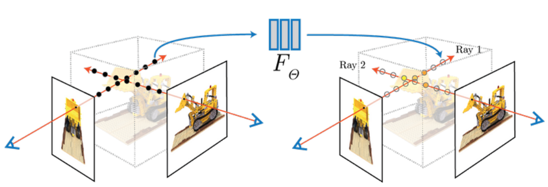

Fig. 14: Projection of coordinates along the camera rays [26].

Results like the one in Fig. 15 demonstrate that the Neural Radiance Field, with its approach to 3D reconstruction through the representation of the scene in five dimensions (three spatial dimensions and two view-related dimensions), produces significantly more accurate results, addressing issues previously discussed in this section.

V. Applications in Robotics

Digital twins, virtual replicas of physical systems, benefit from NeRF by creating accurate 3D models of environments in which robots can interact with their surroundings in a more informed manner. In applications for industrial robots, accurate 3D reconstructions, based on images and poses gathered by the robot, can streamline workflows and improve operational efficiency [27]. Additionally, the ability of NeRF to synthesize novel views from sparse input data allows the continuous updating of digital twins as the physical environment changes, thus maintaining their relevance and accuracy over time [2].

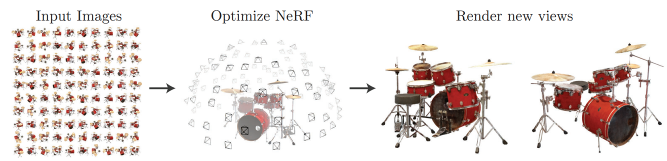

Fig. 15: Complete NeRF process [26].

Regarding sim-to-real transfer in robotics, where the goal is to bridge the gap between simulated environments and real-world scenarios, realistic simulations to train robotic systems can be generated. For 3D scene representations, NeRF facilitates alignment between simulated and real environments through a bundle adjustment approach [3]. Furthermore, integrating uncertainty quantification into NeRF enhances the robustness of these models, making them more suitable for real-world applications where variability is a factor [28].

Moreover, surface reconstruction can enhance the development and evaluation of robotic systems in controlled environments before deployment in the real world. Thus, for example, it can be employed to optimize the grasping of transparent objects, showcasing its utility in training robots for complex manipulation tasks [29]. Furthermore, advances in NeRF, such as the introduction of Complex-Motion NeRF, allow simultaneous optimization of camera poses and scene representations, which is crucial for dynamic environments where robots must adapt to changing conditions [30].

VI. Conclusion

The paper reviews three-dimensional surface reconstruction methods, covering classical geometric approaches and advanced deep-learning techniques. Traditional methods have proven effective in various scenarios but are limited by their reliance on specific geometric assumptions and their sensitivity to incomplete or noisy data. In contrast, deep learning-based methods, such as the NeRF method, have opened new possibilities, enabling more flexible reconstructions to handle complex high-dimensional data. While classical methods remain relevant, surface reconstruction is shifting towards deep learning techniques. These methods offer the ability to learn intrinsic data representations, leading to more accurate and robust reconstructions, especially from sparse or noisy data.

Moreover, surface reconstruction can be employed to optimize training for complex manipulation tasks. Furthermore, advances in NeRF allow for simultaneous optimization of camera poses and scene representations, enabling robots to adapt to changing conditions. These advancements further strengthen sim-to-real applications in robotics, facilitating more accurate scene understanding and improving the transferability of learned models to physical systems.

References

[1] Angela Dai and Matthias NieBner. Scan2Mesh: From Unstructured Range Scans to 3D Meshes . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5569–5578, Los Alamitos, CA, USA, June 2019. IEEE Computer Society.

[2] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pages 112–123. PMLR, 2022.

[3] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021.

[4] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, Heidelberg, 2008.

[5] J.D. Gardiner, J. Behnsen, and C.A. Brassey. Alpha shapes: determining 3d shape complexity across morphologically diverse structures. *BMC Evolutionary Biology*, 18:184, 2018.

[6] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1996.

[7] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual Conference on CGIT*, pages 71–78, 1992.

[8] Donald R Chand and Sham S Kapur. An algorithm for convex polytopes. *Journal of the ACM*, 1970.

[9] Dominik Moritz. Convex hull in 3d. https://commons.wikimedia.org/wiki/File:Convex_hull_in_3D.svg, 2013. Accessed: 26th March 2025.

[10] Grit Thürmer and Charles A Wüthrich. Normal computation for discrete surfaces in 3d space. In *CGF*, volume 16, pages C15–C26. Wiley Online Library, 1997.

[11] John Leonard. Normals: What are they and how they help us pick better. https://blog.zivid.com/normals, 2021. Accessed: 26th March 2025.

[12] Boris Delaunay. Sur la sphère vide. *Izvestia Akademii Nauk SSSR*, 7:793–800, 1934.

[13] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[14] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[15] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE TIT*, 29(4): 551–559, 1983.

[16] Articque. C&d and articque platform. https://www.articque.eu/news/new-features-february-18th-2020/, 2020. Accessed: 26th March 2025.

[17] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE TVCG*, 5(4):349–359, 1999.

[18] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *4th Eurographics SGP*, volume 7, 2006.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[20] Michal Oleszak. Introduction to image segmentation. https://projector-video-pdf-converter.datacamp.com/32718/chapter3.pdf, 2025. Accessed: 26th March 2025.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *In IEEE ICCV*, pages 2961–2969, 2017.

[22] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE Conference on CVPR*, pages 9404–9413, 2019.

[23] João F. da S. S. Lemos, Gabriel A. Dorneles, Stephanie L. Brião, Paulo L. J. Drews-Jr, and Rodrigo S. Guerra. Digital environment description and reconstruction using panoptic segmentation. *RoboCup Symposium*, 2024.

[24] Zhiqin Chen. A review of deep learning-powered mesh reconstruction methods. *preprint arXiv:2303.02879*, 2023.

[25] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, and Leonidas J Guibas. Shapeflow: Learnable deformation flows among 3d shapes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9745–9757. Curran Associates, Inc., 2020.

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65, 2021.

[27] Markus Hillemann, Robert Langendörfer, Max Heiken, Max Mehltretter, Andreas Schenk, Martin Weinmann, Stefan Hinz, Christian Heipke, and Markus Ulrich. Novel view synthesis with neural radiance fields for industrial robot applications. *preprint arXiv:2405.04345*, 2024.

[28] Jianxiong Shen, Antonio Agudo, Francesc Moreno-Noguer, and Adria Ruiz. Conditional-flow NeRF: Accurate 3d modelling with reliable uncertainty quantification. In *European Conference on Computer Vision*, 2022.

[29] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. *preprint arXiv:2110.14217*, 2021.

[30] Hyunjin Kim, Daekyeong Lee, Suyoung Kang, and Pyojin Kim. Complex-motion NeRF: Joint reconstruction and pose optimization with motion and depth priors. *IEEE Access*, 2023.