What’s the Situation With Intelligent Mesh Generation: A Survey and Perspectives

Na Lei, Zezeng Li, Zebin Xu, Ying Li, and Xianfeng Gu

(Survey Paper)

Abstract—Intelligent Mesh Generation (IMG) represents a novel and promising field of research, utilizing machine learning techniques to generate meshes. Despite its relative infancy, IMG has significantly broadened the adaptability and practicality of mesh generation techniques, delivering numerous breakthroughs and unveiling potential future pathways. However, a noticeable void exists in the contemporary literature concerning comprehensive surveys of IMG methods. This paper endeavors to fill this gap by providing a systematic and thorough overview of the current IMG landscape. With a focus on 113 preliminary IMG methods, we undertake a meticulous analysis from various angles, encompassing core algorithm techniques and their application scope, agent learning objectives, data types, targeted challenges, as well as advantages and limitations. We have curated and categorized the literature, proposing three unique taxonomies based on key techniques, output mesh unit elements, and relevant input data types. This paper also underscores several promising future research directions and challenges in IMG.

I. INTRODUCTION

A. Motivation

In this paper, we embark on a systematic review of publications centered on Intelligent Mesh Generation (IMG). Recognized as one of the six fundamental research directions in NASA’s Vision 2030 [1], mesh generation holds a pivotal role in computational geometry and is integral to numerical simulations. IMG amalgamates machine learning with mesh generation, a blend whose significance is gaining recognition as a growing corpus of research that explores the use of neural networks to generate high-quality meshes across diverse application scenarios. Driven by various meshing objectives, IMG research has seen a surge in interest. Notably, recent years have seen the advent of many innovative algorithms. Despite the substantial research on IMG, there is a conspicuous absence of comprehensive reviews that adopt a standardized methodology to ensure significance, completeness, and impartiality. To advance research in the IMG field, an urgent need exists for a systematic overview of the current state-of-the-art IMG methods. Such an overview will aid in distilling the essential commonalities in these works, identifying current research trends, and pinpointing promising future directions.

Herein, our aim is to present a comprehensive and systematic survey, complete with a detailed taxonomy and content oriented evaluation. We undertake a multi-faceted analysis, examining key algorithmic techniques and their application scope, agent learning objectives, data types, targeted challenges, along with advantages and limitations. As depicted in Fig. 1, we offer varied taxonomies for existing IMG methods, conceptualized from three perspectives: key techniques, output mesh unit elements, and relevant data types. Considering key techniques, we categorize IMG methods into deformation-based, classification-based, isosurface-based, Delaunay-triangulation-based, parametrization-based, and advancing-front-based.

Fig. 1. Taxonomies for existing IMG methods from three views.
classify application input data types into point cloud-based, image-based, voxel-based, mesh-based, boundary or sketch-based, and latent variable-based mesh generation. Additionally, we present a concise overview of IMG’s related areas, such as conventional neural networks for mesh learning, mesh quality metrics, and datasets. To summarize, we researched and reviewed 190 papers, which includes 113 papers focused on IMG and 77 papers on associated topics. Our paper makes four primary contributions:

- We have undertaken a comprehensive review of 113 seminal IMG papers, evaluating them based on several key aspects: underlying techniques, the scope of application, agent learning objectives, data types, targeted challenges, and advantages and limitations.
- We have categorized existing IMG methods from three distinct vantage points: key techniques, output mesh unit elements, and suitable input data types;
- We provide a broad overview of areas closely related to IMG, including conventional neural networks for mesh learning, mesh quality assessments, and datasets;
- We encapsulate the present challenges confronting IMG and outline potential promising avenues that researchers could explore to address these challenges in their future endeavors.

B. Related Surveys

A systematic literature review (SLR) [2], as utilized in software engineering, is conducted by using a predefined methodical series of steps. To the best of our knowledge, no SLR has been conducted in the IMG domain. In this section, we describe some related review articles [3], [4], [5], [6], [7].

Berger et al. [3] surveyed various point cloud-based surface reconstruction algorithms developed from 1992 to 2015. They classified the algorithms based on the type of priors that are employed, the ability to handle point cloud artifacts, input requirements, shape class, and reconstruction output. This review focuses on point cloud surface reconstruction whose output does not necessarily mesh. Note that mesh reconstruction and surface reconstruction are two different concepts. Mesh reconstruction refers to not only surface mesh but also volume mesh. However, surface reconstruction is not necessarily meshing but may also be voxel, RGBD image, signed distance field, etc. There have been other surveys covering only a specific surface reconstruction domain, such as image-based, 3D object reconstruction [4], surface remeshing [5], and advancing front-based methods [7]. All the surveys mentioned above focus on specific surface reconstruction domains but are not specialized for IMG. In addition, Xiao [6] et al. summarize geometric deep learning from a representational perspective. Thus, our paper is the first review of existing IMG methods.

II. APPROACH

The aim of this paper is to offer a comprehensive and systematic survey by employing a dual-pronged strategy. In the initial phase, we delve into the existing literature, sieving out IMG-relevant work using both an open-ended interpretive approach and a keyword-focused literature review. The former approach is grounded on the authors’ expertise, leading to a broad and exploratory collection of works, though it may introduce selection bias. In contrast, the latter targets specific IMG publications, offering a clear boundary, but its effectiveness relies heavily on the chosen keywords. In an effort to amalgamate the strengths of these two methods, we have subsequently applied both the keyword-centric and the open-ended interpretive approaches. In the subsequent phase, we compile and evaluate all articles gathered from the first phase based on the content extracted.

A. Search Strategy

The search strategy is paramount in assembling pertinent literature on a specific subject. The choice of the search string and the libraries are significant factors. We crafted a search string and selected five distinct libraries¹ to scout for relevant papers. Adhering to the SLR guidelines [2], we constructed a search string by amalgamating the keywords and their synonyms using Boolean operators.

**Keywords and Boolean Operators:**

\[ (\text{“computer graphics” OR “computational geometry”}) \text{ AND (“mesh reconstruction” OR “mesh generation” OR “Surface reconstruction” OR “3D object reconstruction” OR “triangulation” OR “quadilateral” OR “tetrahedral” OR “hexahedral”) AND (“intelligent” OR “learning” OR “data-based” OR “neural network” OR “ANN”) OR (“mesh evaluation” OR “3D object dataset” OR “mesh neural network” OR “mesh metric”).} \]

B. Content Extraction and Presentation of Findings

The existing literature offers a plethora of IMG techniques. We chose a total of 190 papers for a detailed analysis; 113 papers focus on IMG, while 77 papers delve into IMG’s related areas, encompassing classical neural networks for mesh, mesh quality metrics, and mesh datasets, etc. In light of our research questions, we extracted the following details from each selected paper: 1)The main challenges faced and significant breakthroughs achieved; 2)The fundamental concept, scope of applicability, along with pros and cons; 3)The type of input data, output mesh, and mesh quality. Furthermore, to appraise these IMG methods more objectively, we gathered the following additional information: 1)The frequency of citation for the article (number of citations per month as per Google Scholar citation); 2)The practicality of the algorithm proposed in the paper.

From the perspective of key techniques, output mesh types, and input data types, we first classified existing IMG papers and presented them in Table I. This table also encapsulates whether the IMG is end-to-end, its principal network structure, and the learning goals of the neural network model. Table I provides a clear categorization of the 113 IMG papers from three unique viewpoints, allowing an intuitive understanding of how intelligence is incorporated in each IMG method. Second, we delved into the detailed content of each IMG paper and organized Table II in chronological order. This table lists the targeted

---

¹) IEEE Xplore https://ieeexplore.ieee.org; 2) ACM https://dl.acm.org; 3) ScienceDirect https://www.sciencedirect.com; 4) SpringerLink https://link.springer.com; 5) Wiley https://onlinelibrary.wiley.com/.
| Article | DVMG | CLMG | ISMG | DMG | PAMG | AFMG | P | EM | V0 | ME | Bt | LA | D | Qsa | Hy | T | ERI | Note | Antisquare | Learning Goal |
|--------|------|------|------|-----|------|------|---|---|---|----|---|----|---|---|---|---|---|------|-------|-----------|-------------|
| So-organize [42] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Self-organizing network | Vertical location |
| So-Nerther [43] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Self-organizing network | Vertical location |
| Autoann [44] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Learning ANN | Node positions |
| Autoann [45] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Learning ANN | Node positions |
| Autoann [46] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Learning ANN | Node positions |
| Tang [47] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Multi-level Link | Coordinated point occupancy |
| Yao [48] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Self-organizing maps | Node positions |
| Autoann [49] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Learning ANN | Node positions |
| Agouris et al. [50] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Self-organizing maps | 3D coordinates |
| Xiong et al. [51] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Spine dictionary | Vertical positions and triangulation |
| DeepGun [52] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | DynamicNet | Vertex positions |
| DeepGun [53] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | DeepNet | Vertex positions |
| ResNet [54] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Res-Net | Vertex positions |
| SRFNet [55] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | SRFNet | Vertex positions |
| GC-Net [56] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | GC-Net | Vertex positions |
| Taxonomy of the various types of IMG employed in the Literature | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | GC-Net | Vertex positions |
| TABLE I | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | GC-Net | Vertex positions |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
| Challenge | Advantages | Limitations |
|----------|------------|-------------|
| Lossless compression | Unique to each image, minimal quality loss | Requires complex decoding algorithm, slower encoding and decoding process |
| Lossy compression | Faster encoding and decoding, lower storage requirements | Loss of information, potentially affecting visual quality |
| JPEG (Baseline) | Simple implementation, widely supported | Limited range of bit depth, resolution loss at high compression rates |
| JPEG 2000 | Higher quality images, better scalability | More complex implementation, slower encoding and decoding |
| JPEG XR | High performance, compatible with existing JPEG standards | Requires new hardware support, limited adoption |
| WebP | High quality, lossless and lossy compression | Requires new browser support |
| HEIF (Hierarchical Eoded Image Format) | High compression efficiency, flexible compression options | Requires new hardware support, limited adoption |

**Table II: Challenges, Advantages, and Limitations of All Selected IMG Methods**

**Challenges**

- **Lossless compression**
  - Unique to each image, minimal quality loss
  - Requires complex decoding algorithm, slower encoding and decoding process

- **Lossy compression**
  - Faster encoding and decoding, lower storage requirements
  - Loss of information, potentially affecting visual quality

- **JPEG (Baseline)**
  - Simple implementation, widely supported
  - Limited range of bit depth, resolution loss at high compression rates

- **JPEG 2000**
  - Higher quality images, better scalability
  - More complex implementation, slower encoding and decoding

- **JPEG XR**
  - High performance, compatible with existing JPEG standards
  - Requires new hardware support, limited adoption

- **WebP**
  - High quality, lossless and lossy compression
  - Requires new browser support

- **HEIF (Hierarchical Eoded Image Format)**
  - High compression efficiency, flexible compression options
  - Requires new hardware support, limited adoption

**Advantages**

- Faster encoding and decoding, lower storage requirements
- High compression efficiency, flexible compression options
- Unique to each image, minimal quality loss
- Simple implementation, widely supported
- Better scalability

**Limitations**

- Requires complex decoding algorithm, slower encoding and decoding process
- Loss of information, potentially affecting visual quality
- Limited range of bit depth, resolution loss at high compression rates
- More complex implementation, slower encoding and decoding
- Requires new hardware support, limited adoption
- Requires new browser support
- Requires new hardware support, limited adoption

---

**Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.**
| Article | Challenge | Advantage | Limitation | AMC |
|---------|-----------|-----------|------------|-----|
| HandbookA [23] | Mesh generation with sparse point cloud, material modeling | No self-intersection, does not need ground-truth segmentation | Requires a reference template with the same topology | 4.60 |
| SKNA [4] | Large-scale point cloud mesh reconstruction | Good scalability; small errors in constructing geometry details | Not an end-to-end differentiable process; the partition method is crucial | 3.20 |
| MaskVoxel [6] | An effective propagation of mesh | Propagation speed is based on mesh simplification; can generate details | Only for planar meshes; lacks non-leaf or asymmetrical mesh | 4.55 |
|法mesh [17] | Local mesh for mesh generation | Robust to untrusted entities and noise; produces watertight meshes, unexpected learning | Initial mesh needed; high computational complexity and low reconstruction accuracy | 3.68 |
| NF [8] | Manifold mesh generation | Generates two-manifold mesh | Only for planar shapes | 3.24 |
| SDF+ [9] | A differentiable way to produce surface models | Robust and differentiable; can generate watertight tetrahedra | Need pre-processing for making | 3.67 |
| SDF+ [9] | A differentiable way to produce surface models | Robust and differentiable; can generate watertight tetrahedra | Need pre-processing for making | 3.67 |
| SDF+ [9] | A differentiable way to produce surface models | Robust and differentiable; can generate watertight tetrahedra | Need pre-processing for making | 3.67 |
| Lasso [16] | Non-rigid surface mesh generation | Does not need category-specific mesh templates; generalization | Faster as it increases efficiency needs improvement | 3.47 |
| TransFormation [48] | Efficient coding and surface reconstruction | Online reconstruction approach; runtime as inverselystrained | Cannot correct details for some features | 2.54 |
| CSSNet [9] | Efficient and low memory representation for complex surfaces | Can represent complex surfaces of any topology; efficient computation | Not an end-to-end differentiable process | 2.82 |
| DSSM [26] | High-quality and detailed mesh from DF | Can generate meshes with arbitrary topology, fine geometric details, fewer artifacts | Global uniform, resolution, suitable for local minimums, tends to produce double/blank surfaces | 1.62 |
| VisMesh [15] | Large-scale point cloud mesh generation | Generates robust, small details, as complex | The generated mesh can be incomplete | 1.60 |
| MLASS [14] | Transforms the discrete point set into smooth mesh | Overlaps can be handled on point level | Not an end-to-end differentiable process | 1.60 |
| Repro-all [9] | Mesh reconstruction of large scene | Robust to incomplete information | Does not support real-time | 1.27 |
| Deep [6] | 3D/4D classification, cannot obtain dense meshes | Without ground truth labels of tetrahedra or visibility information | Unstable, can handle extraneous large input | 1.80 |
| 6D-poses [9] | DFMs for noisy and incomplete input | Further convergence and smoother accuracy of details and topology | Not exploitably model the appearance | 1.97 |
| DST [9] | The differentiable structure of triangular mesh | Control both the vertex positions and the topology, conformality to the number of vertices | Surface partition needed; visible artifacts exist | 1.95 |
| SAP [9] | Efficient and differentiable point to mesh layer | Output watertight manifold mesh; interpretable, light-weight and short development time; robustness to outliers | Limited to small scenes; robust memory requirements | 1.80 |
| Borgeaud et al. [27] | Generalization to variuous geometry and topology | Allows continuous evolution with differential geometry | Mining generated details in the final mesh | 6.55 |
| NeuralGP [16] | Mesh generation with extremely sparse point cloud | A supervised method; output can be sparse point clouds | Requires pre-processing to generate mesh | 1.80 |
| AnalyzeMesh [102] | Accurate mesh reconstruction | Does not depend on the details of the implicit field created for the blending cube | Pre-processing, such as smoothing or noise filtering is needed | 1.80 |
| LMR [28] | Mesh inference with 3D/4D videos | Local-dynamical-modeling approach to video mesh recovery | Can generate mesh that cannot match the human body | 5.00 |
| Brief [18] | Mesh generation with sparse point cloud | A simple and efficient model for non-rigid detailed mesh generation | The generated mesh may overlap at point boundaries | 1.81 |
| DSHF [112] | Mesh generation with sparse colored point cloud, multi-point integration | Generates mesh with high-resolution texture; robust to noise and noisy point clouds | Complex structure and surfaces; only single mesh with simple topology can be processed | 1.80 |
| Neural-Net [38] | Learning high-quality DVRs to generic mesh | Learn DVRs without the signed distance values, robust to noise | Gluing sharp features in the reconstruction mesh | 1.80 |
| LSLR [105] | The tradeoff between memory and mesh quality | Randomly encoded geometric and uncertain information | Unstable to provide spatially continuous sharp features | 1.80 |
| DHP [11] | Generate manifold meshes | Presents novel geometric properties; robust to outliers | Not differentiable; only required for simple geometry | 1.80 |
| DHPQ [11] | Most of the DM methods of quadrilaterals rely on a dense user-provided directional field | A robust data-driven approach to the computation of direction and thickness | The way the growth of the frame mesh is constructed determines the quality of the mesh | 1.80 |
| DGR [114] | 3D MR with large-scale scene and incomplete point cloud input | Mesh reconstruction for large, dense level point clouds, take into account visibility information | Lower generalization, low precision when the acquisition is noisy, making reconstruction details | 1.77 |
| Hsu et al. [34] | Generative design of geometric barcodes | Attaches a mesh of through-bottom density and human representation | The final mesh has more overlap | 1.80 |
| AMC [15] | AMC method that preserves sharp features | Can maintain sharp geometric features and learning local topology | Sensitivity to rotation; possible self-intersection in the generated mesh | 1.89 |
| SalvationSM [32] | Lacks of constraints on the topology | Topology-preserved, high-quality skeletal volume | Cannot deal with natural images in the wild; high algorithmic complexity | 1.80 |
| NodFD [114] | Joint optimization of topology and material and lightness | An appearance-aware and end-to-end mesh generator with materials | High evaluation and memory consumption; simplified shading model | 1.80 |
| NodFD [114] | Combines the advantages of imprint and explicit representations | Produces high-quality surface for arbitrarily defined human body with self-supervised optimization | Shape optimization; mainly suitable for self-orienting meshes | 2.00 |
| Autoform [10] | Powerful forces for mesh generation | Provides the mesh generation for a diverse set of tasks | Sensitivity to alignment; CAD models only; not end-to-end differentially | 3.67 |
| NK2 [110] | Generalizes large scene mesh with sparse point cloud, generalization | Can reconstruct shapes according to existing training set; can reconstruct large scene mesh | Kermit implementation requires a dataset linear solver, requirement of optimized points | 2.28 |
| Sketch2Q [111] | Real-time SDF | Real-time mesh generation with a dense direction field | Not differentiable; only for disk topology surfaces | 5.00 |
| SRMACE [112] | Good generalization | Can handle meshes in different axes, suitable for unstructured mesh | Ciferreto pre-processing, underutilization of global information | 2.00 |
| Sfinesse [110] | With point cloud data and valid input | ICPF and normals are not used, applicable to sparse point clouds | Not an end-to-end differentiable process | 1.80 |
| Li et al. [16] | Intelligent advancing-front method | An automatic advancing-front based ICM method | Valid only for 2D mesh; no control over angularity | 4.00 |
| McNer [113] | Enriched structural mesh generation | Unsupervised structured quad mesh generation; requires no prior knowledge or measured datum | Boundary accuracy determines the quality of the mesh; only for planar mesh | 1.80 |
| RLQCM [114] | Efficient and intelligent quad-mesh generation | Automatic quad-mesh generation without extra clean-up operations | Lack of control over angularity; only for planar mesh | 1.80 |
| Rockne[6] [38] | Topology generalization | Not limited by the topology of the template; better geometry-examined mesh | The reconstructed mesh may be used | 5.00 |
| Shape [41] | Topology generation of animal heads | Can generate meshes with geometric details | Fixed to create shapes with thin structures and poor generalization ability | 3.33 |
| 5DQ [11] | Difficult to obtain gradient sufficient for Dual Consensus methods | Good reconstruction quality applicable to various inputs | Produces non-manifold mesh, not completely invariant to orientation | 3.33 |
| PEGAN [114] | Process topology similarity among meshes by CNN | Presents topology information and the spatial structuring | Need pre-processing and post-processing for mesh generation | 3.33 |
| Nine-Island [11] | Large sparse multi-mesh generation | Can fit small holes and extrapolate scene geometry; real-time processing | Cracks representation only | 3.33 |
| NEGR [116] | Scene high-quality mesh reconstruction of scene scene reconstruction | Effective combination of depth observations and neural methods for real-time video scene reconstruction | Need pre-processing, lack of details slow convergence | 4.47 |
| FPOC [121] | Scene high-quality mesh reconstruction of scene scene reconstruction | Effective combination of depth observations and neural methods for real-time video scene reconstruction | Cannot compute shape when missing large parts; with no normal, yield stabilization failures | 2.00 |

AMC denotes average monthly citations.
challenges, advantages, limitations, and average monthly citations for each study. We also showcased some representative methods in Fig. 2, collated existing IMG papers into a line chart according to the number of papers published per year, and displayed this in Fig. 3. Given that our collection concluded in the first half of 2022, the number of IMG papers for the year has seen a notable decrease. However, Fig. 3 illustrates that IMG research is currently at its peak. Furthermore, we displayed the number of papers published per year for each technology as a stacked bar chart. For papers belonging to multiple categories, we counted them separately; hence, the sum of bar values will exceed the total number of papers.

C. Fundamental Concepts

In general, meshes are divided into surface meshes and volume meshes. A surface mesh is a set of polygonal faces, and the goal is to form an approximation of an object’s surface. A polygonal surface mesh has three different combined elements: vertices, edges, and faces. This mesh can also be considered the combination of geometry and topology, where the geometry provides the positions of all its vertices, and the topology provides the information between different adjacent vertices. Mathematically, a polygonal surface mesh $\mathcal{M}$ that contains $V$ vertices and $F$ faces is formed as

$$\mathcal{M} = \{(V, F)|V = \{v_i\}_{i=1,2,...,V}, F = \{f_i\}_{i=1,2,...,F}, f_i \in V^d\}$$  

(1)

where $V$ is the vertex set, $F$ is the face set, and $d$ denotes the number of vertices of each facet. Similarly, a volume mesh $\mathcal{M}$ is formed as

$$\mathcal{M} = \{(V, F, T)|T = \{t_i\}_{i=1,2,...,T}, t_i \in F^k\}$$  

(2)

where $k$ represents the number of faces of each voxel $t_i$.

To avoid ambiguity, the definition of mesh generation is given as follows:

**Definition 1: Mesh Generation** Mesh generation is such a task: given the input data $X$, the mapping $g : X \rightarrow \mathcal{M}$ acts on $X$ and outputs the surface mesh $\mathcal{M} = \{V, F, T\}$.

In essence, the goal of mesh generation is to restore the missing geometry information or topology information in $X$. Through our survey, the input data types of existing IMG methods do not extend beyond point clouds, images, voxels, boundaries or sketches, meshes, and latent variables.

III. CLASSIFICATION BASED ON TECHNIQUE

In this section, we categorize existing IMG methods into six groups, namely deformation-based, classification-based, isosurface-based, Delaunay triangulation-based, parameterization-based, and advancing front-based mesh generation. This classification is grounded on the key techniques and guiding principles inherent in each method.

In this classification, we illustrate how predominant IMG methodologies adeptly amalgamate traditional mesh generation techniques with novel deep learning modules, highlighting the merits and drawbacks of each approach. We anticipate this taxonomy will help readers grasp the function of neural networks in mesh generation more intuitively. In deformation-based IMG algorithms, neural network modules primarily predict vertex locations. For classification-based IMGs, neural networks primarily function as classifiers determining the connection or occupancy of vertices. Isosurface-based IMGs employ neural networks for implicit function fitting and zero isosurface extraction. In Delaunay triangulation-based IMGs, neural networks are variably employed to predict vertex coordinates or weights. For parameterization-based IMG algorithms, neural networks learn parameterized mapping or process parameterized data. In advancing front-based IMG algorithms, neural networks are utilized to predict the forward direction of the wave and the connection type of the new node. Further elaborations are provided in the corresponding subsections below.

A. Deformation-Based Mesh Generation

A mesh consists of polygons and is essentially a discrete representation of a continuous surface. The combinatorial nature of the polygons prevents taking derivatives over the space of possible meshing of any given surface. Thus, it is difficult for mesh processing and optimization techniques to take advantage
of modular gradient descent components of modern optimization frameworks. To circumvent this problem, deformation-based IMG has attracted more attention. As shown in Fig. 4, a distinguishing feature of this technology is the need for an initial mesh such as spherical or ellipsoidal template mesh. Deformation-based IMG is applicable to different data types of inputs. A target mesh based on geometric information is usually generated from images [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], point clouds [35], [36], [37], [38], [39] or voxels [40], [41].

Due to the presence of an initial mesh, this type of method reduces the difficulty of mesh generation to some extent. The neural network only needs to predict the position of the vertex because the connection relationship already exists. The oldest deformation-based IMG work dates to 1991, and Ahn et al. [42] proposed a self-organizing neural network to automatically generate a nonuniform density mesh. This is a pioneering work, albeit only plane meshes with simple geometry. In recent years, a sheer volume of practical methods has emerged. Most of the early deformation-based IMG works involved the deformation of the basic model to obtain the target mesh, such as clothing [8], face [9], and body [13]. These methods only generate deformation meshes of specific target objects. Later, a series of algorithms represented by Pixel2Mesh [10] introduced templates as the basic mesh, which greatly improved the practicality and generalization. The deformation-based approach is tightly integrated with GCN and MLP for better prediction of vertex positions.

However, all deformation-based models have a common drawback: they only generate meshes with the same topology as the initial mesh. Therefore, another research direction is how to adapt it to a variety of topologies. Rios et al. [36] proposed a very direct solution by providing multiple basic template meshes for the algorithm to automatically select. Notably, this method does not fundamentally solve the above problems but only alleviates them. To proceed, Charrada et al. [34] introduced a face-pruning mechanism, which iteratively adapted the topology of the mesh, using face-pruning operations, while keeping the main properties of the template, i.e., the appealing visual aspect and uniform mesh connectivity. Admittedly, the face-pruning mechanism can make the method adaptable to mesh generation for more complex topologies.

B. Classification-Based Mesh Generation

Inspired and encouraged by the classification model in the structured data domain, many researchers have designed various effective IMG algorithms by combining classification model-based neural networks with mesh generation. These algorithms are divided into two categories: occupancy prediction [14], [32], [47], [58], [61], [65], [69], [76], [77], [78], [80], [81], [88], [89], [90], [91], [93], [94], [96], [100], [101], [106], [107], [110], [117], [121] and mesh basic element prediction [66], [79], [82], [83]. The former voxelized the three-dimensional space and decides whether the voxel belongs to an object, which is a binary classification problem. Alternatively, occupancy prediction can be further regarded as the three classification problems of the interior, surface and exterior of the object. The latter regards mesh generation as a combination of basic elements, so the generation model only needs to judge whether the basic elements to be detected should exist, which is also a binary classification problem in essence. Examples include the classification of facet existence and local connectivity. Besides binary classification, multiclassification can also be designed to generate meshes, such as IER [79]. Figs. 5 and 6 show the IGM methods based on occupancy and basic element classification, respectively.

IMG algorithms based on occupancy prediction are represented by occupancy networks [61], which are mainly used to address the task of reconstructing mesh by giving single or multiple images. Early works locate the object surface by judging the occupancy of each grid and then reconstruct the mesh by marching cubes [122]. In the high-resolution case, the floating point operation proportional to the cube of the voxel resolution is unbearable. The grid occupied by the object often consists of only a small number of cubes gathered in the same area, which means that many calculations are wasted. To overcome the problem of high computational complexity, IM-NET [65] proposed a more efficient sampling approach that samples more points near shape surfaces and disregards most points far away. When addressing the task of generating meshes from point clouds, subsequent works [80], [121] focused on judging the occupancy of the point cloud instead of the whole 3D space, which directly reduces the floating-point computation.

Compared with the former, these IMG methods based on mesh basic element prediction generate meshes end-to-end. Yao [48] proposed the first algorithm to generate a mesh using a classification model, which is employed to determine whether to add new vertices and in what way. IER [79] proposed a set of candidate triangle faces by constructing a k-nearest neighbor graph on the input point cloud, where the neural network was utilized to filter...
out the incorrect candidates. The flowchart of IER is provided in Fig. 6. Scan2Mesh [66] and REIN [83] generated meshes by classifying the existence of edges. PointTriNet [82] iteratively applied two neural networks: a classification network predicts whether a candidate triangle should appear in the mesh, while a proposal network suggested additional candidates. The proposal network of PointTriNet avoids the need to define candidate faces in advance and removes the assumption that all points must be derived from triangular mesh vertices. Although the application of PointTriNet has expanded, the generated mesh still has many problems, such as nonmanifold and many holes.

C. Isosurface-Based Mesh Generation

It is assumed that each point \( X \in \mathbb{R}^3 \) has an attribute value and that \( H: X \rightarrow \mathbb{R} \) denotes implicit functions in space \( X \). Then, the surface composed of continuous spaces with equal attribute values is referred to as an isosurface, which is defined as

\[
\text{Isosurface}(h) = \{ X | H(X) = h \}
\]

Existing works on isosurface-based IMG leverage neural networks to fit implicit functions and extract \( \text{Isosurface}(0) \). Triangle meshes are usually reconstructed by marching cubes [122], marching tetrahedra [123], Poisson surface reconstruction [124] algorithms or neural networks [58], [92], [101], [107], [117]. According to the different selected implicit functions, these IMG methods are divided into four categories: radial basis functions (RBFs) [52], occupancy fields [32], [54], [57], [58], [61], [73], [78], [84], [88], [91], [110], [119], [121], signed distance functions (SDFs) [39], [60], [64], [65], [69], [76], [77], [80], [87], [90], [92], [94], [95], [96], [97], [99], [101], [103], [107], [108], [109], [113], [117], [120], and unsigned distance function (UDFs) [74], [86], [89], [100]. The different types of implicit functions are shown in Fig. 7.

Isosurface-based IMG has the benefit of representing complex geometry and topology and is not limited to a predefined resolution. Therefore, isosurface-based IMG is favored by researchers. In 2009, Wen et al. [52] utilized the least squares radial basis function-based neural network to estimate the coefficient on each surface sample and constructed a continuous implicit function to represent a 3D surface. The network also overcame the problem of numerical ill-conditioning and overfitting of traditional RBF reconstruction and offered a tool to utilize fewer samples to reconstruct models and make geometry processing feasible and practical. Afterward, Lun et al. [54] used a decoder network to fit the foreground probability function with the occupancy fields; it is defined as follows:

\[
O : X \rightarrow \{0, 1\}, \quad X \in \mathbb{R}^3.
\]

Recently, many works based on occupancy prediction emerged. Liao et al. [58] proposed differentiable Marching Cubes by predicting the probability of occupancy for each voxel. To improve voxel occupancy prediction’s calculation and memory efficiency, Mescheder et al. proposed occupancy networks [61]. Currently, various occupancy-based methods with different network architectures and training strategies have been proposed to continuously improve the efficiency, robustness, and accuracy [32], [57], [73], [78], [84], [88], [91], [110], [119], [121]. The other two commonly utilized implicit functions are SDF and UDF. Cao et al. [60] introduced a cascaded, 3D, convolutional network architecture that learned SDF from noisy and incomplete depth maps in a progressive, coarse-to-fine manner. Wang et al. [64] presented a deep implicit surface network that generated a high-quality, detail-rich, 3D mesh from a 2D image by predicting the underlying SDF. Among these methods, IMG methods of utilizing Neural Radiance Fields (NeRF) show promising results in reproducing the appearance of an object or scene [120], [125], [126]. For them, NeRF is used to predicate SDF or density fields. Then the mesh can be extracted from the SDF using the marching cubes algorithm.

Although neural implicit representations gained popularity in 3D reconstruction due to their expressiveness and flexibility, the nature of neural implicit representations produces slow inference time and requires careful initialization. To solve these problems, Peng et al. [99] introduced a differentiable point-to-mesh layer using a differentiable PSR [124], which bridged the explicit 3D point representation with the 3D mesh via the implicit indicator field, enabling end-to-end optimization. Its flowchart is shown in Fig. 8. For UDF, Atzmon et al. [74] introduced a sign-agnostic learning approach for learning implicit shape representations directly from raw and unsigned geometric data. In [86], neural distance fields were proposed to predict the unsigned distance field for arbitrary 3D shapes given sparse point clouds. To estimate local geometric such as the normals and tangent planes by implicit representation, Venkatesh et al. [89] utilized an efficient implicit representation referred to as the closest surface point representation, which was utilized to solve the PDE-related problems on surfaces [127], [128], [129], [130], [131], [132]. To process highly sparse input point clouds, NeeDrop [100] introduced a statistically based self-supervised approach to estimate occupancy functions directly from point clouds.

The family of isosurface approaches is sometimes limited by their sensitivity to noise, outliers, and nonuniform sampling. In addition, solving equations on a large scale in implicit methods can be time-consuming. There is a cubic or quadratic relation

\[
\text{Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.}
\]

Fig. 7. Different types of implicit functions [100].

Fig. 8. SAP [99]: an isosurface-based IMG method.
between resolution and run time as well as memory usage, which limits its application value.

D. Delaunay Triangulation-Based Mesh Generation

Delaunay triangulation [133] is a widely employed mesh reconstruction technology that can connect point clouds into triangular meshes. Delaunay triangulation has many excellent properties, such as maximizing the minimum angle characteristic. Regardless of which region the Delaunay triangulation is constructed from, the final generated triangular mesh is unique. Inspired by Delaunay triangulation, some researchers have tried to integrate it into IMG to generate high-quality and manifold meshes [43], [44], [45], [46], [49], [98], [104], [106]. Among these methods, early works [43], [44], [45], [49] predict the position of vertices using neural networks and then generate high-quality meshes using Delaunay triangulation [133]. These methods generate 2D planar meshes with given boundaries or coarse meshes. However, in applications, a three-dimensional surface often needs to be addressed. Alfonzetti [46] utilized the 3D Delaunay algorithm to generate a tetrahedral mesh from a 3D point cloud. Song et al. [93] combined the classification model and adaptive Delaunay algorithm to realize the mesh reconstruction of large indoor and outdoor scenes. Rakotosaona et al. [104] projected a 3D point into the 2D parameter space by a neural network, triangulated the 2D plane point using the Delaunay algorithm, and then pulled the point cloud back to the 3D surface by inverse mapping. The algorithm pipeline is shown in Fig. 9. Furthermore, to realize end-to-end differentiable triangulation, Rakotosaona et al. [98] proposed a differentiable weighted Delaunay triangulation.

E. Parametrization-Based Mesh Generation

As [134] remark, parameterization is a correspondence between a surface mesh embedded in 3D and a simple 2D domain referred to as the parameter space. Generally, parameterization is expected to be bijective, which is a classic problem in computer graphics and geometry processing with multifarious applications, such as mesh generation, texture mapping, and shape correspondence. Considering the convenience of mesh generation in parameter space, some researchers have tried to combine parameterization with machine learning modules and then invented a series of IMG methods based on parameterization. According to the different participation modes of parameterization, we broadly divide them into two categories. The first category is geometry image-based IMG. The authors map the surface to the parameter plane by the traditional parametric method [135] to obtain a geometry image, process the geometry image using the deep learning model, and then convert the geometry image to a mesh. The second category is parameterization learning-based IMG, which learns the mapping from the parameter space to the target space. Then, meshes are generated in the parameter space by the traditional or machine learning method.

For the first category, Sinha et al. [55] developed a procedure to create geometry images that represent the shape surface of a category of 3D objects. Then, the authors utilized these geometry images for category-specific surface generation by developing a variant of deep residual networks. Li et al. [63], [118] proposed a prediction generative adversarial network and prediction-compensation generative adversarial network to learn a joint distribution of geometry and normal images for generating meshes. They aimed to generate meshes with two generative adversarial networks (GANs) [136] and achieved good performance in face mesh generation. The advantage of this type of approach is that 3D objects can be directly processed by 2D grid convolution. However, angle and area distortion is often disregarded. Surfnet [55] reduced the effect of area distortion to some extent by making different objects of the same class have the same geometry image. For the second category, the main idea was to fit a mapping from the parameter space to the object space or vice versa by a neural network. Early work [50] promoted the planar mesh to the surface by fitting the bivariate function. In [51], the mapping from a 2D triangular lattice to a surface in 3D space is learned to lift the mesh in parameter space to the target surface. Other researchers try to project the 3D point cloud onto a 2D plane, generate meshes on the plane, and then pull it back. Considering the complexity and difference in surface topology, part of the work chooses to fit local homeomorphic mapping [26], [56], [67], [75], [98], [104], [105]. The other part directly fits the global mapping by assuming the surface type [53], [102], [111], [115]. Fig. 10 shows a local parameter learning method, DST [98], which decomposed the source into local patches and then performed per-patch meshing. The method employed parameterization $m$, a bijective and piecewise differentiable mapping, to project 3D surfaces into 2D parametric space. Then, the 2D vertices were lifted to form a soft 3D triangulation on the surface using $m^{-1}$. The advantage of DST is to ensure that the generated mesh is 2-manifold, and the limitation is the boundary artifact caused by partitioning, which is a common problem for all local parameterization methods.

F. Advancing Front-Based Mesh Generation

The advancing front method is a greedy algorithm that gradually generates mesh nodes from the boundaries to the interior and recursively executes. Each recursive process is divided into three steps: first, a line segment is selected from the generation segment front set that splits the meshed domain and unmeshed domain, where the selected segment is referred
to as the base segment, as it will form the basis for the creation of a new triangle element; second, a new mesh node or an existing mesh node is connected to the base segment to generate a new triangle element; last, the generation segment front set and triangle elements are updated.

Attracted by the classical advancing front method, some researchers have tried to combine it with machine learning modules to design new practical algorithms. Early advancing front method can generate simple meshes, but it involves many inefficient calculations [137], [138]. Yao et al. [48] combined new vertex position prediction and classification to generate quadrilateral meshes. RLQMG [116] combined the advancing front method with reinforcement learning, which generated wavefronts and new points by state representation and action representation, respectively. Lu et al. [114] used deep learning to learn to advance direction and step. The main limitation of this type of method is that the quality of the generated mesh cannot be guaranteed (e.g., too many singularities). In addition, the existing advancing front-based methods only generate planar meshes.

IV. CLASSIFICATION BASED ON UNIT ELEMENTS

Based on the type of output mesh primitive, we categorize IMGs into triangular, quadrilateral, hybrid polygon, and tetrahedral mesh generation. This classification allows us to identify the type of mesh generated by each method, thus enabling users to select the content that best suits their needs. As demonstrated in Table I, the majority of current IMG methodologies produce triangular meshes. This trend is logical given that triangular meshes are still the prevailing format, supported by a wealth of tools and theoretical knowledge. This observation also underscores that fields other than triangular meshing, such as quadrilateral and volumetric meshing, are areas ripe for further exploration.

A. Triangular Mesh Generation

Triangle meshes are arguably the most predominant surface mesh representation. Its popularity is derived from its simplicity, flexibility, and the existence of many data structures for efficient mesh navigation and manipulation. With an increase in publicly available triangular mesh datasets and further development of deep learning techniques, many methods have been proposed to generate triangular meshes of given point clouds, images or other forms of data.

Notably, intelligent triangular mesh generation algorithms attracted the interest of researchers as early as the 1990 s. The main focus of this period was density-adaptive triangular mesh generation algorithms [42], [43], [44], [45], [46], [49], whose main feature is the generation of triangular meshes of desired nonuniform density on an initial coarse mesh following the guidance of the density function learned by neural networks. However, the above methods mainly focus on planar meshes and rarely consider the surface mesh generation of 3D objects. Therefore, to improve the practicability of the algorithm to adapt to real-world inputs, more researchers have dedicated themselves to investigating 3D mesh generation. Peng et al. [47] explored the feasibility of neural networks in triangular mesh reconstruction and 3D object representation. Although this approach is very simple, it realizes lifting from 2D to 3D by predicting the $z$ coordinate.

As 3D data acquisition equipment rapidly develops, triangle mesh generation with 3D data as input is becoming an important branch of IMG. Triangle mesh generation broadly falls into two types: (1) explicit methods, which directly recover a triangle mesh from the input points, and (2) implicit methods, which are aimed at recovering a volumetric function whose zero level-set encodes the surface. For the former, we consider the point cloud as a degenerate mesh (i.e., a mesh without the connection relationship between two points). Therefore, the selection of vertices and prediction of the correct adjacency is needed. Specifically, Scan2mesh [66] generated triangle meshes by joint prediction of point positions and connection relations. However, using the fully connected graph as the candidate set of edges makes the edge information redundant. Daroya et al. [83] presented REIN, an RNN-based network to generate meshes with varying numbers of vertices by sequential edge prediction. Some methods based on template deformation, as mentioned in Section III-A, are also equivalent to initializing the point connection relations. For the latter, surface triangle meshes are often reconstructed with the marching cubes [122], marching tetrahedra [123] or Poisson surface reconstruction [124] algorithms. However, these methods cannot maintain sharp features and are not end-to-end. Therefore, some researchers have explored differentiable marching cube algorithms to maintain sharp features. Liao et al. [58] demonstrated a differentiable alternative to achieve end-to-end training. Liu et al. [94] proposed a deep moving least squares method to generate triangular meshes based on iso-surfaces. Chen et al. [107] redesigned the marching cube algorithm and constructed some tessellation templates, allowing better preservation of sharp features. Chen et al. [117] improved the dual contouring algorithm [139] to eventually allow for end-to-end training.

Another branch is the triangular mesh generation method based on 2D images or features, which is more ambiguous than the method based on 3D data and demands more robustness prior to assist mesh generation. The reconstruction of occlusion information and the estimation of depth information become particularly important. Some deformation-based methods and parameterization-based methods have become the mainstream methods to address the abovementioned cases. For the former, the image is considered the information that guides the deformation of the template. The typical representative is the Pixel2Mesh series [10], [15], [17], [31]. For the latter, we understand 2D features as a parameterization of 3D object surfaces. The geometry image-based [55], [118] approach deserves to be noted.

B. Quadrilateral Mesh Generation

Due to the appealing tensor-product nature and smooth surface approximation of quad meshes, quad meshing techniques are usually preferred over triangle meshes in many applications, such as texturing, simulation with finite elements, and B-spline fitting. To take advantage of the powerful representation and generalization capabilities of deep learning, some researchers...
have also tried to combine deep learning with quad mesh generation and have obtained good solutions [26], [29], [42], [48], [105], [111], [115], [116].

Following the traditional method of classification, we categorize deep learning-based quadrilateral mesh generation into direct [29], [42], [48], [115], [116] and indirect [26], [105], [111] methods, according to whether intermediate model representation is needed. In these direct IMG methods, Ahn et al. [42] proposed a self-organizing neural network that achieves automatic nonuniform density quad mesh generation by deforming the initial mesh. Yao et al. [48] proposed an artificial neural network-based element extraction method for automatic finite element quad mesh generation. The authors designed reasonable node insertion types and then employed neural networks to predict the node positions and insertion types. The mesh boundary is continuously advanced by several iterations of updates. Pedone et al. [29] generated a database of small deforming rectangular meshes. Chen et al. [115] introduced a differential method for structured mesh generation in an unsupervised manner. The method takes boundary curves as input, employs a well-designed neural network to analyze the potential meshing rules, and outputs the mesh with a desired number of cells. To overcome the difficulties of conventional methods in achieving the balance between high-quality mesh and computational complexity, Pan et al. [116] proposed a reinforcement learning-based method for automatic quadrilateral mesh generation. Unfortunately, all these methods can only handle simple planes or surfaces.

For indirect methods, Smirnov et al. [26] learned a special shape representation: a deformable parametric template composed of Coons patches. Given a raster image, first, the system infers a set of parametric surfaces that realize the input in 3D. Then, quadrilateral meshes are generated using a template. Deng et al. [111] presented the Sketch2PQ system, which uses stroke lines, depth samples, and the visible and occluded region masks induced from the sketch as input. First, Sketch2PQ calculates a direction field and B-spline surface as the intermediate model representation. Then, a quad mesh is extracted from the B-spline surface and CDF via geometry optimization. Another interesting work is LDFQ [105], which learns cross fields from a triangular mesh to generate a quadrilateral mesh. Their network can infer frame fields that resemble the alignment of quads. This rich guidance information is able to guarantee the generation of correct and high-quality quad meshes. The pipeline of LDFQ is depicted in Fig. 11.

C. Hybrid Polygon Mesh Generation

In practical applications, due to various nonideal conditions, it is often impossible to generate only a single basic facet. Therefore, some researchers have designed hybrid mesh generation algorithms out of active or passive intention. However, due to the limitation of their application, there are only a few types of IMG with hybrid meshes as the generation target. In the literature that we searched, only BP-ANN [114], AnalyticMesh [101] and PolyGen [70] belong to this category. BP-ANN generates anisotropic quadrilateral and isotropic triangular meshes by the advancing front method, which improves the level of automation and efficiency of hybrid mesh generation. AnalyticMesh marches among analytic cells to recover the exact mesh of the closed, piecewise planar surface captured by an implicit surface network. The algorithm is applicable to more advanced MLP architectures, including those with shortcut connections and max pooling operations, which support a richer set of architectural designs for learning and exactly meshing complex surface shapes. Polygen more compactly represents the surface of the object with different polygons.

D. Tetrahedral Mesh Generation

Tetrahedral mesh generation is an important branch of mesh generation. Unlike the surface mesh, tetrahedral mesh generation simultaneously divides the surface and the interior of the object. Tetrahedral mesh plays an irreplaceable role in numerical simulation, so it has received the attention of many researchers. However, due to the complexity of the problem and the incompleteness of the basic theory, intelligent tetrahedral mesh generation is still lacking. In the literature that we selected, the tetrahedral mesh generation process has been partially [46] or fully [81] replaced by deep learning modules. In [46], Alfonzetti et al. proposed a neural network generator for tetrahedral meshes. Starting from an initial moderately coarse mesh, the generator grows the mesh up to the user-specified number of nodes by a node probability density function. DefTet [81] is optimized for both vertex placement and occupancy and is differentiable with respect to standard 3D reconstruction loss functions. DefTet offers several advantages over prior work and can output shapes with arbitrary topology, using the occupancy of the tetrahedrons to differentiate the interior of the object from the exterior of the object. DefTet also represents local geometric details by deforming the triangular faces in these tetrahedrons to better align with the object surface, thus achieving high-fidelity reconstruction at a significantly lower memory footprint than existing volumetric approaches. DefTet accepts point clouds or images as input data to generate meshes. The algorithm flowchart of DefTet is shown in Fig. 12.

V. CLASSIFICATION BASED ON DATA TYPES

According to the type of applicable input data, we categorize IMG into categories such as point cloud-based, image-based,
used the focused on the problem 13
I
was the
simultaneously optimized topology, materials, and

IV-A
and Ma et al. reconstructed
is mainly achieved by combining Delaunay triangulation
problems, researchers have proposed a variety of approaches
often sparse and noisy or even incomplete. To address these chal-

A. Point Cloud-Based Mesh Generation
Point clouds, as a major representation of 3D data, have
been widely employed in fields such as autonomous driving
and AR. Point clouds naturally have depth information and
are not affected by ambient lighting conditions compared to
RGB images. Reconstructing a mesh from a point cloud is a
long-standing problem in computer graphics. Recently, various
approaches have been developed to reconstruct shapes for an
array of applications [3]. These methods are divided into two
categories: direct estimation of point locations and connection
relationships, as shown in Fig. 13, and implicit surface recon-
construction based on isosurfaces. This division is illustrated in
section IV-A. Next, we introduce these methods in two aspects:
(1) the challenges of point cloud-based mesh generation and (2)
the learnable priors and related methods proposed to cope with
these challenges.

The challenges of point cloud-based IMG methods are de-

A voxel is a data structure that uses a fixed-size cube as the
minimum unit to represent a 3D object. A voxel is a traditional
method for storing volume data, providing density, opacity,

C. Voxel-Based Mesh Generation
A voxel is a data structure that uses a fixed-size cube as the
minimum unit to represent a 3D object. A voxel is a traditional
method for storing volume data, providing density, opacity,
normals, and other information. The index of the voxel grid

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
proposed a novel framework for data-driven mesh generation that tended to take a coarse mesh as input and refine it to meet the geometric and topological information of the object by neural networks. In [44], Alfonzetti proposed an automatic mesh generator that is the Let-It-Grow neural network. Starting from a rough mesh of triangles with a small number of vertices, this algorithm increases the number of vertices until a user-selected value is reached. The vertex growth is driven by a predefined probability density function. In Alfonzetti’s follow-up work [45, 46, 49], he followed the idea in [44] and proposed different solutions. Liu et al. [72] proposed a novel framework for data-driven coarse-to-fine geometry modeling. Hahner et al. [112] proposed an autoencoder that handled semiregular meshes of different sizes and topologies. This algorithm can reconstruct meshes with high quality and generalize to the dynamics of unseen time sequences. Dielen et al. [105] proposed a neural network that inferred frame fields from unstructured triangle meshes. Afterward, a quad mesh is reconstructed from the frame field by an existing parametrization based quadrangulation method.

E. Boundary- or Sketch-Based Mesh Generation

Compared to point clouds or images, boundaries and sketches are two types of data that are more concise and easier to obtain. Therefore, boundaries and sketches are popular input data forms for mesh generation tasks. The boundary defines the area covered by the mesh, gives the initial position and growth direction for the mesh and is a common input for advancing front-based mesh generation methods. Sketches depict rich geometric features of 3D shapes, such as silhouettes, occluding and suggestive contours, ridges and valleys and hatching lines, and thus provide a succinct and intuitive method for mesh generation. In the IMG that we collected, boundary-based mesh generation methods [21, 42, 43, 48, 114, 115, 116] are usually employed for 2D planar meshes, while sketch-based methods [9, 26, 41, 54, 57, 78, 111] are generally utilized for 3D object surface meshes.

For boundary-based IMG, most methods generate 2D meshes by the advancing front technique [48, 114, 116] or by solving partial differential equations with given initial boundary conditions [21, 115]. In addition, A. Chang-Hoi et al. [42] and D. Lowther et al. [43] generated meshes by deformation and point prediction, respectively. A sketch can be considered a very sparse image. Therefore, additional information (e.g., depth information) or global and local constraints are needed. Lun et al. [54] divided sketch-based generation into two stages: first, generating depth and normal images from a multiview sketch and, second, generating a surface from the depth and normal images. Li et al. [57] used CNNs to infer the depth and normal maps that represent the surface, with an intermediate layer that models the curvature direction field and produces a confidence map to improve robustness. Deng et al. [111] generated quadrilateral meshes using visible and occluded boundaries, contour, and feature lines, and some depth samples. The boundary and the feature lines collectively determine the style of mesh, as shown in Fig. 16.

D. Mesh-Based Mesh Generation

In IMG, mesh-based input is mainly utilized in two tasks: mesh optimization [44, 45, 46, 49, 112] and quadrilateral mesh generation [105]. Early deep learning-based mesh generation
and the target selected the VAE model. The harmonic mean of precision and recall, where recall is a fraction of points on \( M_{gt} \) that lie within a certain distance to \( M_{gt} \), and precision is the fraction of points on \( M_{pred} \) that lie within a certain distance to \( M_{gt} \). The F score mainly captures significant mistakes and disregards small mistakes that significantly contribute to visual artifacts. However, different threshold and distance selection strategies directly lead to different F score values. The normal consistency score [61] measures the consistency of the normal of the 3D model surface, that is, the normals of adjacent faces should be smooth and continuous, rather than obviously discontinuous or

### F. Latent Variable-Based Mesh Generation

As the two dominant generation models, GANs and variational autoencoders (VAEs) [140] naturally exert their power in IMG. It is well known that both models can generate images directly from latent variables (random noise) after training. Inspired by the success in image generation, some researchers have conducted mesh generation research based on GANs and VAEs. Considering that only a latent variable of latent space is needed to generate mesh after model training, this paper classifies this kind of method as latent variable-based mesh generation [59], [62], [63], [69], [85], [118]. Specifically, Li et al. [63], [118] used the GAN model to generate geometry and normal images from a latent variable and then generated a mesh by postprocessing, and PGAN’s flowchart is shown in Fig. 17. Some other works [59], [62], [85] selected the VAE model to reconstruct, deform, or interpolate the mesh. Pq-Net [69] enabled the sequence generation of parts using a conditional variational autoencoder.

### VI. Evaluation

Generally, evaluation methodologies for various IMGs fall into two major categories: qualitative and quantitative. Our qualitative analysis focuses on the content perspective of all IMG methods, including targeting challenges, advantages, limitations, and average monthly citations. On the other hand, quantitative evaluations typically emphasize specific metrics such as time complexity and mesh quality. In this section, we undertake a thorough and comprehensive qualitative analysis based on the extracted content from all selected methods. However, due to the vastly different target tasks and datasets across various IMG methods, the metrics they employ differ significantly. Consequently, we refrain from a quantitative comparison of all IMG methods, instead providing a list of commonly used metrics.

#### A. Content Evaluation

We extracted the key challenges, advantages, and limitations of the proposed algorithm from the reviewed papers and sorted them in Table II in chronological order. The key challenges are often the goal of a paper and embody its value to a certain extent. Advantages are often reflected in effective solutions to challenges. The limitations point out the shortcomings of the algorithm and indicate directions for future research. In specific applications, these three aspects also provide guidance for us to choose suitable algorithms.

In the summary of advantages and limitations, we did not use phrases such as higher accuracy, faster speed, or fewer memory requirements. With the development of IMG technology, the advantages claimed in the original paper are often untenable. Therefore, we try to give a more objective evaluation based on the mission objectives and application scenarios. In addition, we calculated and displayed the average monthly citations (AMC) of each paper, which to some extent reflects the impact of the corresponding work.

#### B. Metrics

In our early attempt at quantitative analysis, we experienced complicated and arduous data collection and analysis. Although we ultimately failed, we also obtained an unexpected discovery that the lack of benchmarks is a challenge hindering the development of IMG. The benchmark here includes unified test data and metrics. For quantitative analysis, most of the methods create their own test data. Even if the dataset is the same, the test data will vary due to different preprocessing. Moreover, the difference in metric selection also directly leads to the inability of quantitative comparisons. Due to the above insurmountable issues, we abandoned the attempt to make a comprehensive quantitative comparison of existing IMG methods. Next, we list and explain the commonly employed metrics in IMG so that readers can understand the challenges of comprehensive comparison in this field. Based on what metrics evaluate, we classified these metrics into error and feature preservation metrics, topological metrics, and perceptual metrics.

**Error and feature preservation metrics:** In IMG, the chamfer distance (CD) [141] is the most prevalent metric to measure the error between the reconstructed mesh \( M_{gt} \) and the target surface \( M_{pred} \). The CD is defined as the average of the minimum distance between two point clouds. The coordinate range of points, density of points, and definition of the distance norm will directly affect the value of the CD. Whether or not the coordinates are normalized, the number of sampling points on the mesh, and whether the distance is the \( L_1 \) or \( L_2 \) norm, yield different CD values, making comparative analysis impossible. The F score [10] is defined as the harmonic mean of precision and recall, where recall is a fraction of points on \( M_{gt} \) that lie within a certain distance to \( M_{pred} \), and precision is the fraction of points on \( M_{pred} \) that lie within a certain distance to \( M_{gt} \). The F score mainly captures significant mistakes and disregards small mistakes that significantly contribute to visual artifacts. However, different threshold and distance selection strategies directly lead to different F score values. The normal consistency score [61] measures the consistency of the normal of the 3D model surface, that is, the normals of adjacent faces should be smooth and continuous, rather than obviously discontinuous or
inconsistent. The test data must contain accurate normal vectors and involve complex projection, so the application scenario is limited.

**Topological metrics:** Watertightness and manifoldness are two topological metrics. Watertightness [82] denotes the percentage of edges that have exactly two incident triangles, and manifoldness [82] represents the percentage of edges with one or two incident triangles. These two metrics measure the mesh from the perspective of topology but disregard the geometry and features of the target surface.

**Perception metric:** As [65], [142] remarks, metrics such as the mean squared error and CD do not account for the visual quality of the object surfaces. For visual perception, low-frequency errors in shapes are less noticeable than high-frequency errors. To overcome the shortcomings of traditional quality metrics, deep learning measurement methods based on visual similarity or perceptual quality [143], [144], [145], [146], [147], [148], [149] have emerged. However, the generalization and applicability of these models need to be improved, so they are rarely employed in quantitative comparative analysis.

As mentioned above, we cannot use the existing results to compare all these IMG methods, nor can we compare them directly by experiments. It is very challenging to conduct a comparative analysis of these IMG methods. Nevertheless, to facilitate researchers in choosing papers of their own interest, we have provided some imperfect qualitative comparison results and comparative relationship diagrams in the supplementary materials and the project repository, available online.

** VII. Peripheral **

**A. Common Datasets**

Data serves as the cornerstone of IMG development. The variety of inputs underscores the diversity of the datasets employed for IMG methodologies. Regrettably, public datasets for volume meshes are currently non-existent. Thus, our focus here is limited to commonly utilized datasets for triangle and quadrilateral meshes.

**Triangle mesh:** These commonly employed triangle mesh datasets include Princeton ModelNet2, ShapeNet [150], TOSCA [151], COSEG [152], surface reconstruction benchmark of Berger [153] and Williams [67], Thingi10K [154], D-FAUST [155], Famous [80], and the CAD dataset ABC [156]; a dataset for single image 3D shape modeling: Pix3d [157]; facial expression datasets: COMA [59] and Meln3D [158]; human body shapes datasets: MGN [11] and MultiHuman [159]; clothed body meshes with real texture: RenderPeople [160], Axyz [161], and Digit Wardrobe [11]; and indoor scenes datasets: ScanNet [162], Scenenet [163], Matterport3d [164], and Synthetic Rooms [77]. The two most frequently used datasets are ShapeNet and ModelNet. ShapeNet contains 3D models from a multitude of semantic categories and organizes them under the Word-Net taxonomy. ShapeNet has indexed more than 3,000,000 3D models, 220,000 of which are classified into 3,135 categories. ModelNet has 662 object categories and 127,915 CAD models. It contains three subsets: Modelnet10 with 10 categories; Modelnet40 with 40 categories; and Aligned40 with 40 classes aligned 3D model.

**Quadrilateral mesh:** The commonly used quadrilateral mesh datasets include QuadWild [165]; Clothed body meshes with real texture: RenderPeople [160], Axyz [161]. Notably, the quadrilateral mesh datasets are not sufficient, which also hinders the development of this field to a large extent.

**B. Commonly Used Prior**

In IMG, commonly employed underlying priors include smoothness, primitives, distribution, user-driven, orientation, visibility and regularity priors. In the absence of supervision information, these priors provide effective guidance for model optimization. Below, we introduce them in detail.

The surface smoothness prior constrains the reconstructed surface to satisfy a certain level of smoothness. The most general form is local smoothness, e.g., the Laplace smooth constraint, which often serves as a regular term of the objective function [10], [14], [15], [16], [17], [20], [25], [40], [81]. Geometric regularization [111], [113] attempts to pull points to their neighbor. With smooth prior constraints, the model often filters out noise but also loses some high-frequency details.

The geometric primitive prior assumes that the scene geometry may be explained by a compact set of simple geometric shapes, i.e., planes, boxes, spheres, and cylinders.

The orientation prior is an important prior in traditional mesh generation and is applicable to deep learning-based mesh generation tasks. The normal consistency prior guarantees the consistency of orientation when generating object surfaces based on distance fields [74].

The visibility prior makes assumptions about the exterior space of the reconstructed scene and how this can provide cues for combating noise, nonuniform sampling, and missing data. Specifically, Deng et al. used visibility in Sketch2PQ [111] by a depth-normal compatibility term and depth sample term. Sulzer et al. [106] skillfully utilized visibility information derived from camera positions and produced watertight meshes. Song et al. [93] explicitly employs depth completion for the visibility prediction of 3D points.

The global regularity prior takes advantage of the notion that many shapes possess a certain level of regularity in their higher-level composition. Regularity refers to the relationship between the whole and the part [69], [76]. The same kind of object is composed of similar basic subparts in a similar arrangement. For example, cars are composed of wheels, chassis, and cover from bottom to top. In addition, symmetry is also a common regularity.

**C. Classical Mesh-Based Learning Architectures**

With the success of deep learning methods in computer vision, many neural network models have been introduced to conduct 3D shape representation using volumetric grids [68], [166], [167], [168], [169] and point clouds [170], [171], [172], [173], [174], [175], [176]. However, due to the complexity and irregularity of mesh data, there are only a few neural networks that

---

[2]http://modelnet.cs.princeton.edu/
simultaneously use the geometric and topological information as inputs [177], [178], [179], [180]. MeshCNN [177], a convolutional neural network designed for triangular meshes, defined edge-centric convolution and pooling operations. Convolutions are applied on edges and the four edges of their incident triangles, and pooling is applied via an edge collapse operation that retains the surface topology. These operations facilitate a direct analysis of mesh in its native form. MeshNet [178] defined face-centric convolution and pooling operations for triangular meshes. To synthesize geometric information and topological information, the network provides an effective data preprocessing and network framework. PD-MeshNet [179] utilized features for both edges and faces of a 3D mesh as input and dynamically aggregated them using an attention mechanism. Singh [180] designed surface correlation blocks that capture local features at various scales.

Although only a few of the network models mentioned above can simultaneously process geometry and topology, many outstanding works have emerged in the research of mesh analysis and feature representation. Convolutions that exploit mesh connectivity structures were introduced by Masci et al. [181]. Since then, several approaches have addressed the irregularity of the mesh structure and sampling rate, proposing ideas such as uniformly sampling the neighborhood of each vertex or achieving regularity by spectral decomposition [182], [183], [184], [185], [186]. Recently, combined with a learned heat diffusion operation, DiffusionNet [187] offered a unified perspective across representations of surface geometry. Another line of work seeks to understand the structure and connectivity of the mesh [38], [85], [185]. The authors implemented convolutional layers over point features related to vertices, for example, in EdgeConv [174]. By combining neighborhood or edge information, these methods can achieve a better comprehensive representation of the geometric and topological information of the mesh surface.

VIII. DISCUSSION AND CONCLUSION

IMG significantly enhances the generalizability, robustness, and practical utility of traditional mesh generation techniques. However, it comes with its own set of challenges: dealing with non-ideal data, algorithm practicality, and complex object generation. The non-ideal data challenge entails working with 2D images lacking 3D information, sparse and non-uniform point clouds, occluded or incomplete point clouds, noisy data, and large-scale input data. Algorithm practicality presents hurdles such as ensuring broad generalization, achieving high computational efficiency, maintaining low memory requirements, exhibiting robustness against non-ideal inputs, and guaranteeing end-to-end differentiability. The challenge of generating meshes for complex objects pertains to intricate objects or scenes. This includes the creation of meshes for large and dynamic scenes, high density detail, texture based meshes, water splash meshes, non-uniform density meshes, and structured meshes.

A. Summary of Open Challenges

In this review, we have highlighted the limitations found across the selected articles. These limitations not only represent the challenges within the IMG field, but also help identify potential future research directions. The primary challenges we’ve recognized are as follows:

1. Generalization: Generalization means the ability for an algorithm to handle objects not observed during training - is a crucial aspect demonstrating the practical utility of the algorithm. The lack of generalization is a prevalent issue with current IMG methods, particularly those using deformation-based mesh generation techniques. These methods often produce meshes specific to a particular object type or topology, limiting their wider applicability.

2. Interpretability: The interpretability of IMG is critically important, especially within the aerospace industry. It is our suggestion that melding IMG with fundamental mathematical and physical models, as evidenced by studies such as Zheng et al. [188] and Lei et al. [189], can not only effectively govern model optimization but also impart enhanced interpretability to the model. However, thus far, no research has directly addressed this particular issue.

3. Lack of Benchmark: As discussed in Section VI-B, performing a comprehensive comparative analysis of all these IMG methods poses significant challenges. The lack of standardized testing procedures, coupled with variations in test data and metrics, has significantly impeded IMG’s advancement. Therefore, the establishment of benchmarking standards would substantially expedite the progress of IMG.

4. Watertight and manifold: Watertight and manifold meshes are frequently required in many computer graphics applications. However, current element classification based IMG methods cannot generate watertight or manifold meshes. In attempting to generate a mesh from scratch, these methods overlook certain local constraints. Furthermore, due to the discrete nature of watertight and manifold meshes, they fail to provide gradients essential for model optimization.

5. Structured mesh: Structured meshes offer numerous advantages that unstructured meshes cannot match, including highly efficient storage and access. Unfortunately, due to their inherent complexity, MGNet [115] is currently the only IMG method that considers structured mesh generation. Thus, it is critical to explore more practical structured or semi-structured IMG algorithms capable of handling real-world objects. Additionally, there is a noticeable scarcity of quadrilateral and volume mesh datasets, which hinders the advancement of IMG in areas of structured quadrilateral mesh and volume mesh generation. Creating open-source datasets in these fields could significantly expedite the development of IMG in the context of structured meshes.

6. Textured mesh: Textured mesh is a highly sought-after representation for 3D objects and finds extensive application in numerous fields, including industrial design and digital entertainment. Existing IMG research has primarily concentrated on the challenges of reconstructing geometry and topology. In contrast, only a handful of studies have focused on the generation of textured meshes [23], [108], [190]. Furthermore, the generation of textured meshes undeniably enhances the interpretability of the model, making this an important area for future exploration in the IMG field.
Large or dynamic scene mesh generation: Scene mesh generation holds immense potential in diverse fields, such as autonomous driving, indoor robotics, and mixed reality. For IMG methods to truly prove useful in real-world applications, several critical properties must be addressed. First, real-time functionality of the algorithm is desired. Second, the algorithm should have the capacity to make plausible predictions for regions devoid of observations. Furthermore, the system should demonstrate scalability to accommodate large scenes. Lastly, the robustness against noisy or incomplete observations is essential.

Existing methodologies [13], [28], [73], [90], [91], [93], [95], [110], [119], however, fall short of satisfying these criteria simultaneously, resulting in scene meshes that tend to be overly smooth and lacking in detail.

B. Conclusion

In this article, we have provided a systematic and comprehensive review of Intelligent Mesh Generation (IMG) methodologies, highlighting core techniques, application scopes, learning goals, data types, targeted challenges, strengths, and limitations. We scrutinized 113 research papers for a detailed analysis and data extraction. Our primary points of extraction included 1) the addressed challenges, 2) the fundamental concepts, application range, advantages, and disadvantages, 3) the type of input data, the nature of output mesh, and mesh quality, and 4) potential directions for future research. Articles were classified based on their techniques, unit elements, and applicable data types. While there have been substantial advancements in IMG in recent years, we also identified numerous issues and challenges that pave the way for future exploration and investigation in this field.

To the best of our knowledge, this is the most recent and comprehensive survey of existing IMG methods. This survey provides a holistic perspective and extensive research resources for scholars in the field of IMG. However, it is worth noting that our review has certain limitations. Specifically, our focus was solely on IMG; non-machine-learning mesh generation methods were beyond our purview. As a result, numerous valuable papers on traditional mesh generation were not included. Furthermore, despite the application of systematic literature review methodology and a manual search to ensure comprehensive inclusion, there may have been relevant papers that were overlooked in the initial selection due to the sheer volume of existing literature.

REFERENCES

[1] J. P. Slotnick et al., “CFD vision 2030 study: A path to revolutionary computational aerosciences,” Tech. Rep., 2014.
[2] S. Keele et al., “Guidelines for performing systematic literature reviews in software engineering,” Version 2.3, Tech. Rep., 2007.
[3] M. Berger et al., “A survey of surface reconstruction from point clouds,” in Comput. Graph. Forum, vol. 36, no. 1, 2017, pp. 301–329.
[4] X.-F. Han, H. Laga, and M. Bennamoun, “Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 5, pp. 1578–1604, May 2019.
[5] D. Khan et al., “Surface remeshing: A systematic literature review of methods and research directions,” IEEE Trans. Vis. Comput. Graph., vol. 28, no. 3, pp. 1680–1713, Mar. 2022.
[6] Y.-P. Xiao, Y.-K. Lai, F.-L. Zhang, C. Li, and L. Gao, “A survey on deep geometry learning: From a representation perspective,” Comput. Vis. Media, vol. 6, no. 2, pp. 113–133, 2020.
[7] W. Nianhua, L. Peng, C. Xinghua, and Z. Laiping, “Preliminary investigation on unstructured mesh generation technique based on advancing front method and machine learning methods,” J. Theor. Appl. Mech., vol. 53, no. 3, pp. 740–751, 2021.
[8] R. Danielek, E. Dijkstra, C. Özüreli, R. Ziegler, and M. Gross, “DeepGarment: 3D garment shape estimation from a single image,” Comput. Graph. Forum, vol. 36, pp. 269–280, May 2017.
[9] X. Han, C. Gao, and Y. Yu, “DeepSketch2Face: A deep learning based sketching system for 3D face and caricature modeling,” ACM Trans. Graph., vol. 36, no. 4, pp. 1–12, 2017.
[10] M. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, “Pixel2mesh: Generating 3D mesh models from single RGB images,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 52–67.
[11] B. L. Bhattacharjee, G. Tiwari, C. Theobalt, and G. Pons-Moll, “Multi-garment net: Learning to dress 3D people from images,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 5420–5430.
[12] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, “Deep mesh reconstruction from single RGB images via topology modification networks,” in IEEE Int. Conf. Comput. Vis., 2019, pp. 9964–9973.
[13] A. Venkat, C. Patel, Y. Agrawal, and A. Sharma, “HumanMeshNet: Polygonal mesh recovery of humans,” in Proc. IEEE Int. Conf. Comput. Vis. Workshop, 2019, pp. 2178–2187.
[14] G. Gloeys, I. Johnson, and J. Malik, “Mesh R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 9784–9794.
[15] C. Wen, Y. Zhang, Z. Li, and Y. Fu, “Pixel2Mesh++: Multi-view 3D mesh generation via deformation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 1042–1051.
[16] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao, “BCNet: Learning body and cloth shape from a single image,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 18–35.
[17] N. Wang et al., “Pixel2Mesh: 3D mesh model generation via image guided deformation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3600–3613, Oct. 2021.
[18] F. Tong, M. Nakao, S. Wu, M. Nakamura, and T. Matsuda, “X-ray2Shape: Reconstruction of 3D liver shape from a single 2D projection image,” in Proc. IEEE 42nd Annu. Int. Conf. Eng. Med. Biol. Soc., 2020, pp. 1608–1611.
[19] X. Li, K. Ping, X. Gu, and M. He, “3D shape reconstruction of furniture object from a single real indoor image,” in Proc. 17th Int. Conf. Comput. Wavelet Act. Media Technol. Inf. Process., 2020, pp. 101–104.
[20] Y. Dongsheng, K. Ping, and X. Gu, “3D reconstruction based on GAT from a single image,” in Int. Conf. Comput. Wavelet Act. Media Technol. Inf. Process., 2021, pp. 122–125.
[21] Z. Zhang, Y. Wang, P. K. Jimack, and H. Wang, “MeshingNet: A new mesh generation method based on deep learning,” in Proc. Int. Conf. Comput. Sci., 2020, pp. 186–198.
[22] Z. Zhang, W. Isler, and D. D. Lee, “Surface Hof: Surface reconstruction from a single image using higher order function networks,” in Proc. IEEE Int. Conf. Image Process., 2020, pp. 2666–2670.
[23] P. Henderson, V. Tsiminaki, and C. H. Lampert, “Leveraging 2D data to learn textured 3D mesh generation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 7495–7504.
[24] G. Yang et al., “LASR: Learning articulated shape reconstruction from a monocular video,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 15980–15989.
[25] U. Wickramasinghe, P. Fua, and G. Knott, “Deep active surface models,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 11652–11661.
[26] D. Sminov, M. Bessmeltsev, and J. Solomon, “Learning manifold patch-based representations of man-made shapes,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–13.
[27] H. Bertiche, M. Madadi, and S. Escalera, “Deep parametric surfaces for 3D outfit reconstruction from single view image,” in Proc. IEEE 16th Int. Conf. Autom. Face Gesture Recognit., 2021, pp. 1–8.
[28] R. Li, S. Karanam, R. Li, T. Chen, B. Bhanu, and Z. Wu, “Learning local recurrent models for human mesh recovery,” in Proc. Int. Conf. 3D Vis., 2019, pp. 555–564.
[29] M. Pedone, A. Mostafa, and J. Heikkilä, “Learning non-rigid surface reconstruction from spatio-temporal image patches,” in Proc. Int. Conf. Pattern Recognit., 2021, pp. 10134–10140.
[30] M. Hu, J. Rutqvist, and C. I. Steefel, “Learning local recurrent models for human mesh recovery,” in Proc. Int. Conf. 3D Vis., 2020, pp. 3600–3613, Oct. 2021.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
[32] J. Tang, X. Han, M. Tan, X. Tong, and K. Jia, “SkeletonNet: A topology-preserving solution for learning mesh reconstruction of object surfaces from RGB images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6454–6471, Oct. 2022.

[33] B. Jiang, Y. H. Biao, and J. Zhang, “SelfRecon: Self reconstruction your digital avatar from monocular video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5605–5615.

[34] T. Ben Charaida, H. Tabia, A. Chetouani, and H. Laga, “TopoNet: Topology learning for 3D reconstruction of objects of arbitrary genus,” *Comput. Graph. Forum*, vol. 41, no. 6, pp. 336–347, Mar. 2022.

[35] W. Wang, D. Ceylan, R. Mech, and U. Neumann, “3DN: 3D deformation network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1038–1046.

[36] T. Rios et al., “Back to meshes: Optimal simulation-ready mesh prototypes for autoencoder-based 3D car point clouds,” in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2020, pp. 942–949.

[37] R. Hancock, G. Metzer, R. Giryes, and D. Cohen-Or, “Point2Mesh: A self-prior for deformable meshes,” *ACM Trans. Graph.*, vol. 39, no. 4, pp. 126:1–126:12, Aug. 2020.

[38] K. Gupta and M. Chandraker, “Neural mesh flow: 3D manifold mesh generation via diffeomorphic flows,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1747–1758.

[39] B. Ma, Z. Han, Y.-S. Liu, and M. Zwickler, “Neural-pull: Learning signed distance functions from point clouds by learning to pull point surfaces onto surfaces,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7246–7257.

[40] U. Wickramasinghe, E. Remelli, G. Knott, and P. Fua, “Voxel2Mesh: 3D implicit surface network for high-quality single-view 3D reconstruction,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 490–500.

[41] Z. Chen and H. Zhang, “Learning implicit fields for generative shape modeling,” in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5932–5941.

[42] A. Dai and M. Nießner, “Scan2Mesh: From unstructured range scans to 3D meshes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5569–5578.

[43] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, “Deep geometric prior for surface reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1012–1013.

[44] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[45] R. Wu, Y. Zhan, K. Xu, H. Zhang, and B. Chen, “PO-Net: A generative part seq2seq network for 3D shapes,” in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 829–838.

[46] C. E. Nash, Y. Gani, S. M. A. Eslami, and P. Battaglia, “PolyGen: An autoregressive generative model of 3D meshes,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 7220–7229.

[47] A. Hertz, R. Han, Y. H. Biao, and D. Cohen-Or, “Deep geometric prior for surface reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 108:1–108:11, 2020.

[48] H.-T. D. Liu, V. G. Kim, S. Chaudhuri, N. Aigerman, and A. Jacobson, “Deep implicit field learning via convolutional networks,” in *3D Shape Reconstruction from Point Clouds*, Springer, 2019, pp. 7213–7222.

[49] T. Li, Y. Shi, X. Sun, J. Wang, and B. Yin, “PGAN: Prediction generative adversarial nets for meshes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7213–7222.

[50] M. Liu, X. Zhang, and H. Su, “Meshing point clouds via learned mesh prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6001–6010.

[51] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, and T. Funkhouser, “Local implicit grid representations for 3D scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6001–6010.

[52] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.

[53] G. Yan, Z. Chen, J. Yang, and H. Wang, “Interactive liquid splash modeling by user sketches,” *ACM Trans. Graph.*, vol. 39, pp. 1–13, 2020.

[54] M. Liu, X. Zhang, and H. Su, “Meshing point clouds with predicted intrinsic-extrinsic ratio guidance,” in *Proc. Eur. Conf. Comput. Vis.*, Springer International Publishing, 2020, pp. 68–84.

[55] M. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Winzer, “Points2Surf learning implicit surfaces from point clouds,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 108–124.
A. Dielen, I. Lim, M. Lyon, and L. Kobbelt, “Learning direction fields for quad mesh generation,” in *Proc. Comput. Graph.,* vol. 40, no. 5, pp. 181–191, Aug. 2021.

J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, “SA-ConvONet: Signification learning from unstructured point clouds of large scenes with learned virtual view generation,” in *ACM Trans. Graph.,* vol. 40, no. 12, pp. 10068–10086, 2021.

Y. Luo, Z. Mi, and W. Tao, “DeepDT: Learning geometry from delaunay triangulation for surface reconstruction,” in *Proc. AAAI Conf. Artif. Intell.,* 2021, pp. 2277–2285.

A. Bozic, P. Pafalox, J. Thies, A. Dai, and M. Nießner, “Transformationfusion: Monocular RGB scene reconstruction using transformers,” in *Proc. Int. Conf. Neural Inf. Process. Syst.,* 2021, pp. 1403–1414.

R. Venkatatesh et al., “Deep implicit surface point prediction networks,” in *Proc. IEEE Int. Conf. Comput. Vis.,* 2021, pp. 12563–12662.

J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, “NeuralRecon: Real-time coherent 3D reconstruction from monocular video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 15598–15607.

J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, “SA-ConvONet: Signagnostic optimization of convolutional occupancy networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 6504–6513.

T. Shen, J. Gao, K. Yin, M.-Y. Liu, and S. Fidler, “Deep marching tetrahedra: A hybrid representation for high-resolution 3D shape synthesis,” in *Proc. Int. Conf. Neural Inf. Process. Syst.,* 2021, pp. 6087–6101.

S. Song, Z. Cui, and R. Qin, “Vis2Mesh: Efficient mesh reconstruction from unstructured point clouds of large scenes with learned virtual view visibility,” in *Proc. IEEE Int. Conf. Comput. Vis.,* 2021, pp. 6514–6524.

S.-L. Liu, H.-X. Guo, H. Pan, P.-S. Wang, X. Tong, and Y. Liu, “Deep implicit moving least-squares functions for 3D reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 1788–1797.

Y. Sidiqi, J. Thies, F. Ma, Q. Shan, M. Nießner, and A. Dai, “Retrieval-fuse: Neural 3D scene reconstruction with a database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 12568–12577.

Y. Luo, Z. Mi, and W. Tao, “DeepDT: Learning geometry from delaunay triangulation for surface reconstruction,” in *Proc. AAAI Conf. Artif. Intell.,* 2021, pp. 2277–2285.

W. Yifan, S. Wu, C. Oztireli, and O. Sorkine-Hornung, “Iso-points: Optimizing implicit triangular surfaces with hybrid representations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 374–383.

M.-J. Rakotosona, N. Aigerman, N. J. Mitra, M. Ovsjanikov, and P. Guerrero, “Differentiable surface triangulation,” *ACM Trans. Graph.,* vol. 40, no. 6, pp. 1–13, 2021.

S. Feng, C. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger, “Shape as a differentiable poison solver,” in *Proc. Int. Conf. Neural Inf. Process. Syst.,* 2021, pp. 13032–13044.

A. Boulch, P.-A. Langlois, G. Puy, and R. Marlet, “NeeDrop: Self-supervised shape representation from sparse point clouds using needle dropping,” in *Proc. Int. Conf. 3D Vis.,* 2021, pp. 940–950.

J. Lei, K. Jia, and Y. Ma, “Learning and meshing from deep implicit surface networks using an efficient implementation of analytic marching,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 44, no. 12, pp. 10068–10086, Dec. 2022.

X. Wei, Z. Chen, Y. Fu, Z. Cui, and Y. Zhang, “Deep hybrid self-prior for full 3D mesh generation,” in *Proc. IEEE Int. Conf. Comput. Vis.,* 2021, pp. 5758–5794.

J. Huang, S.-S. Huang, H. Song, and S.-M. Hu, “DI-Fusion: Online implicit 3D reconstruction with deep priors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 8928–8937.

M.-J. Rakotosona, P. Guerrero, N. Aigerman, N. Mitra, and M. Ovsjanikov, “Learning delaunay surface elements for mesh reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2021, pp. 22–31.

A. Dielen, I. Lim, M. Lyon, and L. Kobbelt, “Learning direction fields for quad mesh generation,” *Comput. Graph. Forum,* vol. 40, no. 5, pp. 181–191, Aug. 2021.

R. Salzer, L. Landrieu, R. Marlet, and B. Vallet, “Scalable surface reconstruction with delaunay-graph neural networks,” *Comput. Graph. Forum,* vol. 40, no. 5, pp. 157–167, Aug. 2021.

Z. Chen and H. Zhang, “Neural marching cubes,” *ACM Trans. Graph.,* vol. 40, no. 6, pp. 1–15, Dec. 2021.

J. Munkberg et al., “Extracting triangular 3D models, materials, and lighting from images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2022, pp. 8280–8290.

P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani, “AutoSDF: Shape priors for 3D completion, reconstruction and generation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2022, pp. 306–315.

F. Williams et al., “Neural fields as learnable kernels for 3D reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2022, pp. 18500–18510.

Z. Deng, Y. Liu, H. Pan, W. Jabi, J. Zhang, and B. Deng, “Sketch2PQ: Freeform planar quadrilateral mesh design via a single sketch,” *IEEE Trans. Vis. Comput. Graph.,* to be published, doi: 10.1109/TVCG.2022.3170853.
[131] L. Tian, C. B. Macdonald, and S. J. Ruther, “Segmentation on surfaces with the closest point method,” in Proc. IEEE Int. Conf. Image Process., 2009, pp. 3009–3012.

[132] C. B. Macdonald, J. Brandman, and S. J. Ruther, “Solving eigenvalue problems on curved surfaces using the closest point method,” J. Comput. Phys., vol. 230, no. 22, pp. 7944–7956, 2011.

[133] D.-T. Lee and B. J. Schachter, “Two algorithms for constructing a delaunay triangulation,” Int. J. Comput. Inf. Sci., vol. 9, no. 3, pp. 219–242, 1980.

[134] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez, “Periodic global parameterization,” ACM Trans. Graph., vol. 25, no. 4, pp. 1460–1485, 2006.

[135] X. Gu, S. J. Gortler, and H. Hoppe, “Geometry images,” in Proc. 29th Annua. Conf. Comput. Graph. Interactice Techn., 2002, pp. 355–361.

[136] J. Goodfellow et al., “Generative adversarial networks,” Commun. ACM, vol. 63, pp. 139–144, 2020.

[137] R. Lührn and P. Parikh, “Generation of three-dimensional unstructured grids by the advancing-front method,” Int. J. Numer. Methods Fluids, vol. 8, no. 10, pp. 1135–1149, 1988.

[138] Y. Guo, X. Huang, Z. Ma, Y. Hai, R. Zhao, and K. Sun, “An improved advancing-front-delaunay method for triangular mesh generation,” in Proc. Comput. Aided Des., vol. 41, no. 4, pp. 1–10, 2009.

[139] X. Sun et al., “Pix3D: Dataset and methods for single-image 3D shape modeling,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2974–2983.

[140] I. Abouelaziz, A. Chetouani, M. El Hassouni, and H. Cherifi, “Reduced reference mesh visual quality assessment based on convolutional neural networks,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2017, pp. 656–665.

[141] I. Abouelaziz, M. El Hassouni, and H. Cherifi, “A curvature based method for blind mesh visual quality assessment using a general regression neural network,” in Proc. 12th Int. Conf. Signal-Image Technol. Internet-Based Syst., 2013, pp. 793–797.

[142] I. Abouelaziz, M. E. Hassouni, and H. Cherifi, “A convolutional neural network framework for blind mesh visual quality assessment,” in Proc. IEEE Int. Conf. Image Process., 2017, pp. 755–759.

[143] I. Abouelaziz, A. Chetouani, M. El Hassouni, and H. Cherifi, “Reduced reference mesh visual quality assessment based on convolutional neural network,” in Proc. Int. Conf. Signal-Image Technol. Internet-Based Syst., 2018, pp. 617–620.

[144] I. Abouelaziz, A. Chetouani, M. E. Hassouni, L. J. Latecki, and H. Cherifi, “Convolutional neural network for blind mesh visual quality assessment using 3D visual saliency,” in Proc. IEEE Int. Conf. Image Process., 2018, pp. 3553–3557.

[145] I. Abouelaziz, A. Chetouani, M. El Hassouni, H. Cherifi, and L. J. Latecki, “Learning graph convolutional network for blind mesh visual quality assessment,” in Proc. IEEE Int. Conf. Image Process., 2021, pp. 1–11.

[146] I. Abouelaziz, A. Chetouani, M. El Hassouni, H. Cherifi, and L. J. Latecki, “Feature-line driven quad-remeshing,” in Proc. Int. Conf. Robot. Automat., 2017, pp. 103104.

[147] X. Sun et al., “ABC: A big cad model dataset for geometric deep learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9601–9611.

[148] Y. Yang, C. Feng, Y. Shen, and D. Tian, “FoldingNet: Point cloud autoencoder via deep grid deformation,” IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2018, pp. 206–215.

[149] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 652–660.

[150] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 5105–5114.

[151] X. Zheng et al., “Deepmulticap: Performance capture of multiple characters using sparse multi-view cameras,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 6239–6249.

[152] X. Sun et al., “Pix3D: Dataset and methods for single-image 3D shape modeling,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2974–2983.

[153] A. Handa, V. Pätzlauceanu, S. Stent, and R. Cipolla, “SceneNet: An annotated model generator for indoor scene understanding,” in Proc. IEEE Int. Conf. Robot. Automat., 2016, pp. 5737–5743.

[154] N. Pietroni, S. Nuvoli, T. Alderighi, P. Cignoni, and M. Tarini, “Reducible feature-line driven quad-remeshing,” in ACM Trans. Graph., vol. 40, no. 4, pp. 1–17, 2021.

[155] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black, “Dynamic faust: Registering human bodies in motion,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6233–6242.

[156] S. Koch et al., “ABC: A big cad model dataset for geometric deep learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9601–9611.
[182] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, “Learning shape correspondence with anisotropic convolutional neural networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2016, pp. 3189–3197.

[183] I. Lim, A. Dielen, M. Campen, and L. Kobbelt, “A simple approach to intrinsic correspondence learning on unstructured 3D meshes,” in Proc. Eur. Conf. Comput. Vis. Workshop, 2018, pp. 349–362.

[184] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe, “DualConvMeshNet: Joint geodesic and euclidean convolutions on 3D meshes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8612–8622.

[185] Y. Zhou et al., “Fully convolutional mesh autoencoder using efficient spatially varying kernels,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 9251–9262.

[186] D. Smirnov and J. Solomon, “HodgeNet: Learning spectral geometry on triangle meshes,” ACM Trans. Graph., vol. 40, pp. 1–11, 2021.

[187] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov, “DiffusionNet: Discretization agnostic learning on surfaces,” ACM Trans. Graph., vol. 41, no. 3, pp. 1–16, 2022.

[188] X. Zheng, Y. Zhu, W. Chen, N. Lei, Z. Luo, and X. Gu, “Quadrilateral mesh generation III: Optimizing singularity configuration based on abel-jacobi theory,” Comput. Methods Appl. Mech. Eng., vol. 387, 2021, pp. 114146.

[189] N. Lei, K. Su, L. Cui, S.-T. Yau, and X. D. Gu, “A geometric view of optimal transportation and generative model,” Comput. Aided Geometric Des., vol. 68, pp. 1–21, 2019.

[190] L. Gao, T. Wu, Y.-J. Yuan, M.-X. Lin, Y.-K. Lai, and H. Zhang, “TM-Net: Deep generative networks for textured meshes,” ACM Trans. Graph., vol. 40, no. 6, pp. 1–15, 2021.

Na Lei received the BS and PhD degrees from Jilin University, in 1998 and 2002, respectively. Currently, she is a professor with the Dalian University of Technology. Her research interest is the application of modern differential geometry and algebraic geometry to solve problems in engineering and medical fields. She mainly focuses on computational conformal geometry, computer mathematics, and its applications in computer vision and geometric modeling.

Zebin Xu received the BS degree from Dalian University of Technology (DUT), in 2020. He is currently working toward the PhD degree from the Dalian University of Technology. His research interests include image processing and mesh generation.

Zebin Xu received the BS degree from the Dalian University of Technology (DUT), in 2020. He is currently working toward the PhD degree from the Dalian University of Technology. His research interests include image processing and mesh generation.

Zebin Xu received the BS degree from the Dalian University of Technology (DUT), in 2020. He is currently working toward the PhD degree from the Dalian University of Technology. His research interests include image processing and mesh generation.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.