Computers & Graphics 94 (2021) 164–190

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Survey Paper

Single-View 3D reconstruction: A Survey of deep learning methods ☆

George Fahim [a,b,*], Khalid Amin [a], Sameh Zarif [a]

[a] Department of Information Technology, Faculty of Computers and Information, Menofia University, Egypt
[b] Multimedia and Internet Department, International Academy for Engineering and Media Science, Giza, Egypt

ARTICLE INFO

Article history:
Received 19 September 2020
Revised 28 November 2020
Accepted 29 December 2020
Available online 5 January 2021

Keywords:
3D Machine learning
Geometric deep learning
3D Synthesis
3D Representations
Single-view 3D reconstruction

ABSTRACT

The field of single-view 3D shape reconstruction and generation using deep learning techniques has seen rapid growth in the past five years. As the field is reaching a stage of maturity, a plethora of methods has been continuously proposed with the aim of pushing the state of research further. This article focuses on surveying the literature by classifying these methods according to the shape representation they use as an output. Specifically, it covers each method's main contributions, degree of supervision, training paradigm, and its relation to the whole body of literature. Additionally, this survey discusses common 3D datasets, loss functions, and evaluation metrics used in the field. Finally, it provides a thorough analysis and reflections on the current state of research and provides a summary of the open problems and possible future directions. This work is an effort to introduce the field of data-driven single-view 3D reconstruction to interested researchers while being comprehensive enough to act as a reference to those who already do research in the field.

1. Introduction

Deep learning has brought about an encouraging performance in a multitude of tasks in computer vision, achieving impressive results over conventional methods that depend on handcrafted features. Tasks like image classification [1], object detection [2], and object localization and segmentation [3] have benefited a lot from applying deep learning algorithms. Most of the state-of-the-art performances in computer vision tasks are achieved by deep-learning-based methods[1].

There has been a rapidly growing interest in expanding the application of deep learning methods to otherwise unexplored venues in computer vision other than the traditional 2D tasks. Among these fields is 3D machine learning, an interdisciplinary field that integrates computer vision, machine learning, and computer graphics. Activities within this field can broadly be divided into two categories: 3D geometry analysis and 3D synthesis. Specific tasks in 3D machine learning include pose estimation [4], shape completion [5], 3D shape classification [6], retrieval [7], reconstruction, and generation.

This survey paper focuses on deep learning advances in 3D shape reconstruction and generation, specifically single-view reconstruction of 3D objects (scene reconstruction and organic shapes human faces and bodies reconstruction are beyond the scope of this paper), a topic that belongs to the 3D synthesis category and has attracted a lot of scientific attention in recent years. The problem of single-view 3D shape reconstruction can be framed as follows: Given a single image representing an object in 2D, how one can reconstruct a 3D representation of the depicted object as faithfully as possible. This problem of reconstructing a 3D shape from a 2D representation is ill-posed since there is no unique solution for it, especially because one has to hallucinate the information lost in the original 3D to 2D projection more so when there is an occlusion.

Conventionally, 3D reconstruction was approached traditionally using methods like Structure from Motion (SfM) [8] and Multi-View Stereo (MVS) [9]. These approaches require an adequate number of images with a small baseline viewpoint differences. They also either require known camera calibration (internal and external) or need to compute these camera properties. They also need to compute a correspondence between images to finally compute the representation of the 3D shape and lift these images collectively from 2D to 3D.

Single-view 3D reconstruction is a long-standing problem that has also been tackled using conventional computer vision algorithms [10,11]. These approaches require extensive user-specified constraints and/or strong assumptions concerning the image at

☆ This paper was recommended for publication by Yotam Gingold.

* Corresponding author.
E-mail addresses: george.fahim@iams.edu.eg (G. Fahim), k.amin@ci.menofia.edu.eg (K. Amin), sameh.shenoda@ci.menofia.edu.eg (S. Zarif).

[1] https://paperswithcode.com/sota

https://doi.org/10.1016/j.cag.2020.12.004
0097-8493/

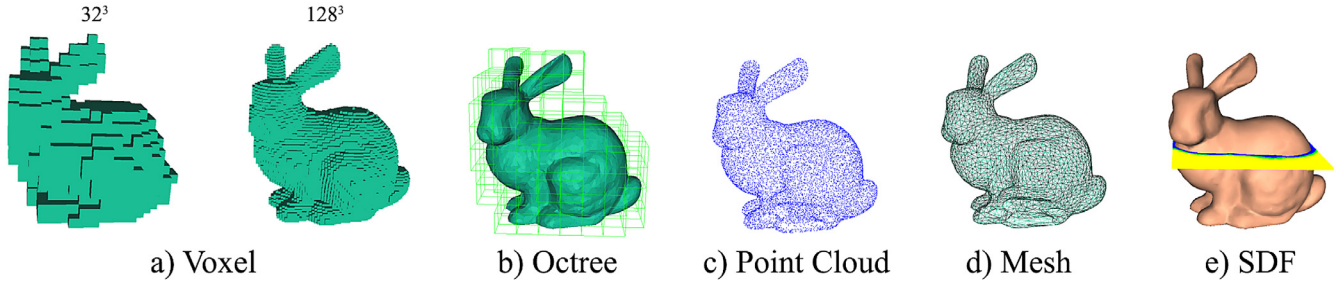

Fig. 1. The Stanford Bunny in different 3D representations.

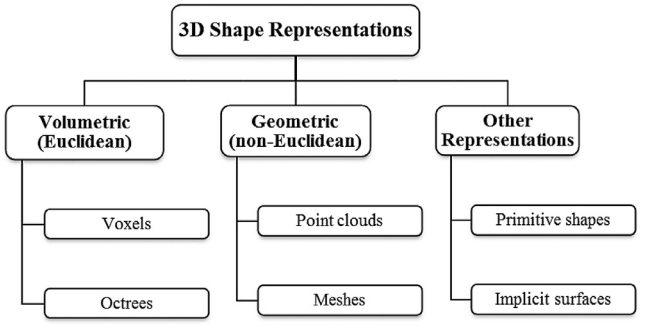

Fig. 2. 3D shape representations commonly used in single-view 3D reconstruction methods.

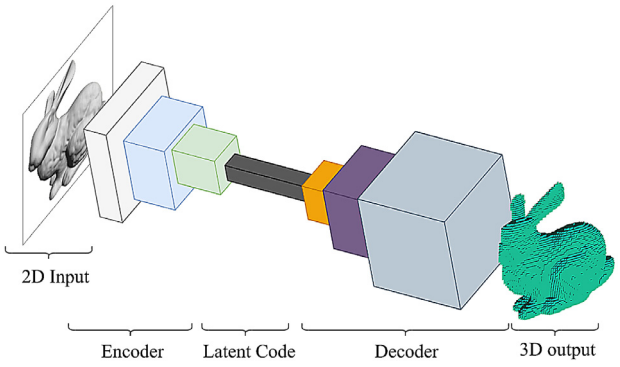

Fig. 3. Encoder-Decoder architecture for single-view 3D reconstruction.

hand. This resulted in sub-optimal results that were not generalizable. Researchers have been trying to mitigate such drawbacks using deep learning methods that are discussed in this survey paper.

In the earliest endeavors, many researchers did not work directly on 3D data but preferred to work on intermediate representations such as projections [12,13] and 2.5D (e.g. RGBD images) data [14–16]. The main reason behind this choice of representation was the availability of Convolutional Neural Network (CNN) architectures that were by design capable of working on 2D images directly. However, other researchers envisioned the modification of conventional CNNs to work directly on 3D data. This survey focuses mainly on these endeavors that work directly on 3D data.

3D deep learning makes use of several 3D shape data representations that can be broadly divided into Euclidean representations and non-Euclidean (also referred to as geometric) representations of data. In 3D reconstruction literature, shapes are usually represented as either voxel grids shown in Fig. 1a or octrees shown in Fig. 1b which belong to the Euclidean category, or as 3D point clouds shown in Fig. 1c or 3D meshes shown in Fig. 1d which belong to the non-Euclidean geometric category. Fig. 2 shows the 3D shape representations covered in this paper. For further discussion on different representations of 3D data and their use in deep learning one can refer to these excellent surveys [17]–[19].

The remainder of the paper is structured as follows: Section 2 presents different single-image 3D reconstruction methods that use Euclidean shape representations. Section 3 presents methods that use geometric shape representations. Section 4 covers methods based on implicit surfaces and geometric primitive representations. It also covers hybrid methods that make use of different shape representations in their pipelines. Commonly used 3D datasets, loss functions, and evaluation metrics are discussed in Section 5. Additionally, Section 5 provides a discussion on the current trends, challenges, and possible future directions in the field. Finally, Section 6 concludes this paper.

2. Euclidean volumetric approaches

The seminal work done by Wu et al. [20] and Maturana and Scherer [21] concurrently albeit separately is what paved the road for working on 3D volumetric data directly using 3D CNNs. Prior to their work, 3D CNNs were used in video analysis tasks assuming time as the third dimension. However, the spatially volumetric representation embedded in the architecture of 3D CNNs also fitted as a suitable representation for 3D objects as an occupancy grid of voxels. A 3D object can be represented as "a probabilistic distribution of binary variables on a 3D voxel grid. Each 3D mesh is represented as a binary tensor: 1 indicates the voxel is inside the mesh surface, and 0 indicates the voxel is outside the mesh" [20]. Despite that the purpose of these two projects was not 3D reconstruction but more generic 3D analysis tasks like object recognition and retrieval, still they were instrumental in directing the research towards more specialized tasks like 3D reconstruction using the introduced volumetric voxel grid representation.

2.1. Voxel grid as an output

Most of the methods that fall under this category follow the encoder-decoder pattern where an image is encoded into a learned feature vector or a latent space and then decoded in an opposite fashion into a voxel representation of the 3D shape as illustrated in Fig. 3. This is usually done under the supervision of synthetic data of voxelized 3D meshes and their respective renderings. Other methods relax this requirement by relying on 2D images only for supervision. Using voxel grids as an output is a quite popular approach and most of the body of literature on single-view 3D reconstruction can be categorized under this approach.

2.1.1. 3D Supervised methods

Methods that use 3D supervision require the availability of 3D ground truth shapes during training. They can be further di-

vided into generative and non-generative methods. Non-generative methods are only capable of reconstructing a shape given an input image. On the other hand, Generative models tend to learn the probabilistic distribution of a range of shapes thus are capable of synthesizing novel shapes in addition to recovering shapes from images.

Non-generative Methods

One of the first approaches (and widely used nowadays as a baseline for comparison) for 3D reconstruction using voxel representation was presented by Choy et al. [22]. The researchers, motivated by the need for an end-to-end automatically learned and data-driven method that is able to reconstruct 3D shapes from both single and multi-view images, proposed 3D-R2N2 a Recurrent Neural Network (RNN) to learn the mapping between images and their underlying 3D shapes in a unified approach that is capable of reconstructing the 3D shape from as few as a single image but also capable of refining the 3D representation when more views are available.

The main contribution of their work is the proposed 3D Convolutional Long Short-Term Memory (3D-LSTM) which resides between the encoder and the decoder. Several 3D-LSTM units are assembled in a volumetric grid structure, and each unit is restricted in its connections to enforce spatial locality for each unit so that it contributes to the reconstruction of only a small part of the whole 3D shape. The 3D-LSTM unit can update its hidden state selectively by opening or closing its input and forget gates depending on the new view fed into it through the encoder. The encoder and decoder are standard feedforward convolutional networks, and the researchers also experimented with deep residual networks. The encoder contained a series of convolution, pooling, leaky ReLU (Rectified Linear Unit) layers followed by a fully connected layer. The deep residual variation has identity matching connections after every two convolution layers except for the fourth pair. The decoder also follows the same pattern except for unpooling layers instead of pooling ones until the required output volume size is reached.

The 3D-R2N2 is credited with being the first to work on 3D voxel grids directly for 3D reconstruction. The researchers approached the problem in a way that made single and multi-view reconstruction possible in one architecture. Additionally, the experiments show that their approach achieved better results over conventional methods in multi-view reconstruction when the number of views is limited. In single-view reconstruction, their work outperformed the work done by Kar et al. [23] which is discussed in this survey under mesh representations.

Sun et al. [24] proposed Im2Avatar an architecture that reconstructs colorful 3D objects from single-view images. The problem of recovering surface color along with predicting a 3D shape was succinctly tackled in Tulsiani et al. work [25] as one form of weak supervision from RGB images as will be discussed below, however, Im2Avatar architecture is dedicated to recovering both shape and color simultaneously. The researchers proposed two independent encoder-decoder networks, one for shape reconstruction and the other for estimating surface color. The color learning process is further divided into two methods. The first method regresses the color from the image directly, and the other samples the color from a projected 2D view of the 3D shape. The two methods are then blended to recover the final appearance of the 3D shape.

Another contribution of Sun et al. work is the modified form of cross-entropy loss function for 3D shape learning which incorporates calculating both the false positive and false negative cross-entropies separately to compose the Mean Squared False Cross-Entropy Loss (MSFCEL) which is a variant of Mean Squared False Error (MSFE). The reason behind this new formulation is to balance the accuracies of predicting both the empty and occupied voxels since their losses are minimized together.

In an attempt to recover 3D shapes from user-drawn sketches, Delanoy et al. [26] proposed a novel method to seamlessly integrate interactive 2D sketching and 3D visualization. Their model is made of a single-view shape predictor network and an updater network. The two work together to reconstruct shapes from one or more user-drawn sketches. First, the single-view network takes one sketch and predicts an initial 3D shape as a voxel grid, then additional sketches even from different views can be used to refine the prediction using the updater network iteratively. To simplify the training procedure, the single-view network is trained first on line drawings and their respective ground truth 3D shapes. After training the single-view network, its predictions are used as training data for the updater network. The researchers also built a user interface that incorporates a 3D camera to aid the user in sketching from different views with the option to convert the drawing to a 3D prediction at any point of time during sketching.

To produce higher resolution reconstructions, Richter and Roth [27] proposed a novel two-stage approach towards voxel-based reconstructions. The main idea of the work is to encode the 3D shape more compactly by representing a shape as an n-channel image, where n is the number of voxels along the z-direction. This approach translates the problem into a 2D dense pixel prediction task, where each pixel represents a whole row of voxels (a voxel tube). The decoder then predicts entire voxel tubes instead of individual voxels using such representation. The second stage allows for further higher resolution reconstructions where voxel tubes are compressed by means of shape layers. Each shape layer is the result of fusing six depth maps orthogonally projected from both the positive and negative X, Y, and Z axes. Each pair belonging to a specific axis are fused to construct a shape from this specific view. Finally, the three shapes are fused through their intersection. The aforementioned shape layers can even be nested to compose more complex shapes through a series of additions/subtractions between shapes in different layers. This novel approach allows for single-view reconstructions with resolution up to 256^3, an unprecedented achievement using voxel representation.

Shape plausibility and consistency was tackled by Wu et al. [28] by introducing a network that learns shape priors. The researchers' work is an extension of their MarrNet [29] network (discussed below under 2D supervised methods). They added a Shape naturalness network which is basically a 3D-GAN [30] with Wasserstein distance and gradient penalty as a loss function. In this adversarially trained network the discriminator learns to distinguish between realistic shapes from unrealistic ones, thus can be used as a naturalness discriminator since it has the ability to model the distribution of real shapes.

Also exploiting shape priors to learn generic shape representation that is class-agnostic, Zhang et al. [31] proposed Generalizable Reconstruction (GenRe) framework. This framework divides the reconstruction process into three different modules: a single-view depth estimator, a spherical map inpainting network, and a voxel refinement network. These learnable modules are connected through geometric projections. The depth estimator takes an RGB image and estimates a depth map. The role of the spherical map inpainting network is to take a partial spherical map projected from the depth map and learns to fill the missing data to represent the shape's whole surface. The voxel refinement network takes two projections in voxel space, one projected from the estimated depth map, and the other projected from the inpainted spherical map, then it reconstructs the final shape by integrating the two projected shapes into one final output.

A different approach towards recovering shape from both single-view and multi-view inputs that is also context-aware was proposed by Xie et al. [32]. The researchers' work can be considered an attempt to overcome the drawbacks of RNN-based methods like 3D-R2N2 [22]. The main problems with RNN-based

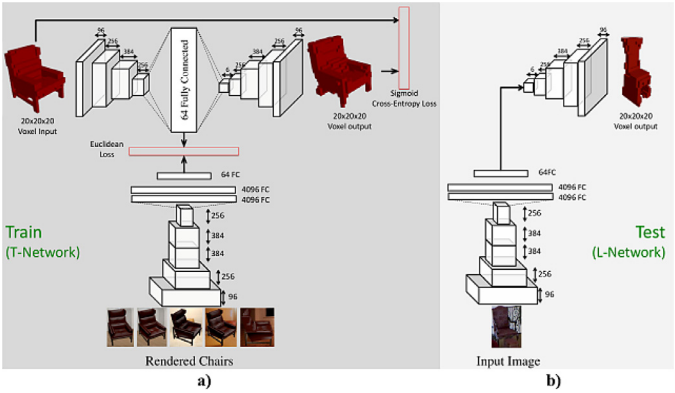

Fig. 4. TL-embedding network as originally shown in [34]. (a) The network structure for training. (b) The network structure for testing.

methods are their sensitivity to image order, their inability to fully utilize all the images due to long-term memory loss, and their time and calculation complexity. Alternatively, the researchers proposed a parallel approach to encode several image inputs and decode them into coarse volumes simultaneously. The coarse volumes are then fed into a context-aware fusion network that bases its fusion on a learned fusion score. Finally, the fused volume is further refined using the third component of the model, the refiner module.

In an attempt to address topology along with shape reconstruction, a recent approach [33] tackled the issue by incorporating two topological properties into the reconstruction pipeline, namely, connectivity and genus number (number of holes). To this end, the researchers proposed a topology-aware autoencoder called TPW-Coder. This autoencoder extends the MarrNet [29] architecture by introducing two loss functions - combined as a topological loss - that minimize the discrepancies in connectivity and genus between the ground truth shape and the reconstructed shape, where genus and connectivity prediction is posed as a classification problem. The TPWCoder is first pre-trained on the ground truth voxel, connectivity, and genus data, then the whole model along with MarrNet 3D shape estimator is fine-tuned together.

Generative Methods The first to propose a generative model for the 3D reconstruction problem was Girdhar et al. [34]. The researchers were motivated by the need to create an object representation that is both generative in 3D and predictable from 2D images. Thus, they divided their architecture (termed TL-embedding shown in Fig. 4) into two parts: one that deals with encoding and decoding the 3D shape using an autoencoder with a bottleneck of a 64-dimensional embedding, and the other part is a pre-trained network that learns the mapping of 2D images into the same 64D embedding space.

The proposed TL-embedding network architecture, was trained in three distinct stages. The autoencoder was trained under 3D supervision of voxelized shapes using cross-entropy loss. The image network is in fact a slightly modified AlexNet [35] initialized with ImageNet [36] weights and learned to regress to the autoencoder coefficients using Euclidean loss. The third stage was to jointly fine-tune the whole network using both loss functions. During testing, the encoder part is removed and the image goes through the image network to produce the 64D feature vector which is then fed into the decoder part of the autoencoder to produce the 3D voxel representation of the object.

The main contribution of Girdhar et al. other than the novel TL-embedding network architecture was the introduction of a latent space that is smooth and generative where one can interpolate between two shapes and get a new shape that the network has not seen before during training. Additionally, the latent representation allowed 3D arithmetic where shapes can be added or subtracted from each other to produce novel shapes. The researchers also experimented with 3D model retrieval which is a task that the network has not been specifically trained for but still managed to perform with quite good results.

In a similar vein but using a Generative Adversarial Network (GAN) [37] Wu et al. [30] proposed a generative-adversarial model called 3D-GAN. It uses a generator that is fed a latent vector sampled from a probabilistic latent space and maps it to a voxel grid representing the 3D object. It also uses a discriminator that learns to distinguish between objects generated from the generator and real 3D objects. The adversarial interaction between the generator and the discriminator lets them both learn better: the generator learns to generate objects that can fool the discriminator and the discriminator learns to enhance its discriminative ability to distinguish between real and generated objects.

To equip the 3D-GAN model to perform reconstruction from images, the researchers incorporated a Variational Autoencoder (VAE) [38] as an image encoder that maps a 2D image into the latent space described above. The extended model thus contains an image encoder, a decoder (which is the generator of the 3D-GAN), and a discriminator. In addition to the cross-entropy loss used to train the 3D-GAN, the extended model (called 3D-VAE-GAN) uses an additional reconstruction loss and Kullback Leibler (KL) divergence loss. The aim of the KL divergence loss is to restrict the variational distribution and sampling distribution (used as an input to the generator) to be from the same prior distribution of the latent space. The latent space has the same properties as in the TL-embedding network, it is smooth, allows interpolation between objects even from different classes, and supports 3D arithmetic.

Liu et al. [39] extended the 3D-GAN model to work in an interactive tool for 3D modeling. This was achieved by devising an additional projection operator that maps from the user input in a voxel grid form to a feature vector in the latent space of the 3D-GAN. Smith and Meger [40] also extended the 3D-GAN model. They employed the Wasserstein distance normalized with gradient penalization as a training objective, which made them achieve better results than the original 3D-GAN. Another recent improvement to the 3D-GAN model was proposed by Zhu et al. [41]. The researchers introduced an additional enhancer network that is trained with synthetic images in an adversarial manner. The learned high-level image features of the enhancer network are fed into the GAN generator during training for a better generation of shapes.

Along the same lines, Brock et al. [42] proposed a VAE for voxel modeling and they created a user interface to visualize the latent space interpolation. Additionally, they modified the cross-entropy loss function to reduce the vanishing gradient problem and to avoid the network from getting stuck in a local optimum. This was done by adding a hyperparameter that weighs the relative importance of false negatives against false positives. Also, they changed the range of the binary target value from {0,1} to {-1,2} and they clipped the output to be in the range $[0.1, 1)$ to increase the magnitude of the loss gradient thus reducing the probability of vanishing gradients.

In an attempt to produce higher quality shapes and to ease the learning process - by avoiding multi-stage training - Liu et al. [43] adopted a hierarchical approach towards learning the latent variable instead of learning a single latent representation of data, and proposed the Variational Shape Learner (VSL). The researchers' main idea was to use skip connections to combine a number of local latent variables to form a global latent variable. Each local variable represents a specific level of abstraction, where variables closer to the input represent lower-level features and the ones farther away from the input represent higher-level features in a coherent hierarchical manner. All local variables and the global

variable are concatenated at the end to represent the encoded shape. The proposed generative model was jointly trained and was used for single-view reconstruction as well as shape classification and retrieval.

To enforce structural consistency in a generative model, Balashova et al. [44] proposed an additional structure detector network to a VAE based architecture. Their work aimed to retain important structural elements within an object category (e.g. chairs) during reconstruction. The structure detector was used to learn semantic landmarks of shapes from a specific category and then was used as a guide for the generator to produce shapes consistent with the output of the structure detector. This was achieved by adding a structure consistency loss term to the generator's loss function. The encoder/generator and the structure detector were trained separately, then the encoder was fixed and the structure detector and the generator were trained together, and lastly, the whole model was fine-tuned together.

2.1.2. 2D Supervised methods

Yan et al. [45] approached the problem of single-view 3D reconstruction from a different perspective, from the learning agent's point of view. Two assumptions were made: first, the learning agent has a built-in camera to convert from 3D objects into 2D observations. Second, the learning agent can disentangle the object's intrinsic properties like geometry and material from external factors relating to view-point changes like orientation, position, and illumination. The proposed method then handles the problem as a dense prediction problem. The researchers devised two approaches: one required 3D supervision and the other depended on 2D silhouettes only for supervision, with the possibility of combining both approaches. The network architecture called Perspective Transformer Network (PTN) followed the encoder-decoder pattern with the decoder divided into a volume generator and a perspective transformer layer which is a differentiable dense sampling layer. The aim of this additional layer is to transform a 3D shape into a 2D observation from a particular point of view. An extensive ablation study was performed with single-class, multi-class, and out-of-category (unseen classes). Also, different loss functions were compared in each variation. The performance of the network with the perspective transformer layer proved to be superior even without 3D supervision and also generalized better than other variants in the out-of-category tests.

As an alternative to using full 3D supervision, Gwak et al. [46] further extended their work done on 3D-R2N2 architecture with an additional Raytrace Pooling layer that renders the reconstructed voxel shape into a 2D mask. The 2D masks are then used for 2D supervision against ground truth 2D masks in a similar manner to the perspective transformer layer described above in Yan et al. [45] work. However, the researchers also added an adversarial constraint to ensure the plausibility of the reconstructed shape. To this end, they framed the problem as a constrained optimization problem. The adversarial constraint aims to restrict the reconstruction to lie inside the manifold of the plausible and realistic shapes within a category or class. This constrained optimization problem was solved using the log barrier method, by formulating a function similar to the GAN discriminator and its loss function.

The researchers at MIT CSAIL[2] - noticing the noisy and blurry reconstructions of their 3D-GAN model - tried to tackle the 3D reconstruction problem differently. This time Wu et al. [29] proposed the MarrNet model to first estimate 2.5D sketches (depth and surface normal maps) from a 2D image then estimate the 3D shape from the 2.5D sketches. Although this method does not regress an image to a 3D voxel grid directly, the 2.5D sketch estimator is only considered a component for intermediate representation, and the method targets 3D shape reconstruction in the voxel grid format.

The MarrNet model contains two main modules (2.5D sketches estimator and 3D shape estimator) and an additional reprojection consistency function. Both modules follow the encoder-decoder pattern. Each module is trained separately on synthetic data then the whole model is fine-tuned with real data using the reprojection consistency loss function, but with fixing the decoder of the 3D estimator to prevent overfitting and to preserve the learned shape prior.

MarrNet outperformed the baselines and proved to adapt well to real data but still suffered from failure in images with complex shapes and thin structures. Another achievement attributed to MarrNet is its ability to reconstruct 3D shapes with a resolution of 128^3, an unprecedented result within the voxel representation category at the time except for the work done by Johnston et al. [47] where the researchers achieved the same result of 128^3 resolution through replacing the decoder with an inverse discrete cosine transform (IDCT) layer.

An approach similar to MarrNet and PTN but depends on weaker multi-view supervision instead of direct 3D supervision or reprojection was proposed by Tulsiani et al. [25]. The researchers motivated by how active agents perceive the 3D world proposed multi-view supervision from 2D images in different modalities (RGB - foreground masks - depth maps). This weaker form of supervision guides the network to learn geometric consistency between the predicted 3D shape and 2D observations. They formulated the Differentiable Ray Consistency (DRC) loss to achieve this geometric consistency.

However, instead of using the 2D image as a whole for computing the loss function, they reduced the loss to be the sum of consistency losses between rays traversing through the predicted voxel grid and the predicted 3D shape, where each ray corresponds to a pixel in the image and is associated with a specific observation with known camera intrinsic parameters. For each ray, the probabilities of the ray's termination at each voxel along its path are assigned a random variable each and grouped in a ray termination event. Then event probabilities are computed to determine the likelihood of termination at various voxels. For each termination point, event costs determine how inconsistent is stopping there with respect to the observation associated with the ray. This novel loss function was used to train the model and outperformed the baselines. One should notice that despite multi-view supervision, the DRC model is used to infer the 3D shape from a single image at test time.

Another attempt to decrease the reliance on expensive 3D supervision was investigated by Yang et al. [48]. The researches proposed a unified model that can handle different modes of supervision (e.g. images annotated with camera pose from a single category, images annotated with camera pose from multiple categories, unlabeled images). The model contained an encoder, a generator, a discriminator, and an additional projection module. The training paradigm was based on alternating between different modes of supervision in each training iteration. While each mode of supervision required a different loss function, the same model got updated in all iterations. In the iterations with pose annotation, the loss function was a combination of a reconstruction loss and a pose-invariance loss (for both encoded representations and voxels separately). The iterations that used unlabeled images the loss function was a typical adversarial loss.

A self-supervised method for voxel-based shape prediction was proposed by Mees et al. [49]. The researchers approach uses only image silhouettes as a supervisory signal. However, it also requires a class-specific mean shape. The proposed model has a convolutional image encoder, an upconvolutional shape decoder, a

[2] https://www.csail.mit.edu/.

viewpoint regressor, and a 3D to 2D projection module. The training is performed in multiple steps: first, the viewpoint estimator is trained for a number of epochs, then the whole network is trained on synthetic images. The second step involves fine-tuning the network using real images. The shape decoder predicts only the difference between the class-specific mean shape and the desired shape to be predicted. The predicted shape is then projected into camera space and the loss is computed between the ground truth silhouette and the projected shapes silhouette using squared Euclidean loss. The researchers validated the usefulness of their shape estimation model in the field of robotics by integrating it in a grasp planning experiment.

In general, voxel-based methods constitute a huge part of the work done in data-driven single-view 3D reconstruction. However, this approach suffers from a major drawback. Voxel-based approaches are quite popular because they are intuitive and straight forward, but they are not able to attain high-resolution reconstructions. The highest resolution attained from purely voxel-based approaches is 64^3 due to GPU memory constraints. This is mainly because the memory requirements grow cubically as the voxel resolution increases. Additionally, the representation space is inefficiently used since voxels inside the shape take up memory and computational resources without contributing to the final appearance which depends on boundary voxels only.

2.2. Octree as an output

Due to the mentioned limitations of voxel-based methods, the research community had to explore other representations for 3D shapes that might not be as straight forward as voxels but more memory and computationally efficient. Among these alternative representations is the octree representation [50]. The first to explore using an octree data structure as an alternative to regular voxel grids were Riegler et al. [51] then Wang et al. [52] briefly afterwards. An octree is a 3D space partitioning data structure, where each node can subdivide the space it represents into eight octants depending on the density of the space represented. Octrees are more efficient spatial representation than standard voxels as they can represent any arbitrarily shaped object and they can represent fine-grained details through recursively subdividing any node into eight more children nodes instead of resorting to the uniformly sized voxels in a regular voxel grid. The limitation of octrees is that they are still considered an approximation to the 3D shape regardless of the number of subdivisions since they cannot fully represent surface curvature.

Riegler et al. [51] made two critical observations regarding the standard voxel-based approaches. (1) Because of the sparsity of 3D data, and since the voxel occupancy percentage decreases as the voxel resolution increases, much of the computation power is wasted on empty voxels. (2) Highest activations occur at the object boundary thus it is more efficient to focus computation power and memory allocation near the surface of the object. The researchers proposed OctNet which incorporates space partitioning function within the network architecture. Specifically, they placed a number of shallow octrees, with a fixed depth of three, in a grid structure, where a large octree cell can cover a number of voxels away from the surface, and finer octree cells cover voxels near the surface. This efficient placement of different sized octree cells made working on high-resolution shapes possible up to 256^3, however, it required redefining the common network operations like convolution, deconvolution, pooling, and unpooling in a way that can work on this hybrid and irregular structure.

Motivated by the same observations, Wang et al. [52] proposed the octree-based O-CNN which varies slightly from OctNet. In O-CNN network operations are performed only on the surface boundary represented by the leaf octants at the finest level. Correspondence between features at different levels is managed through a label array, and a hash table is built to facilitate searching for the local neighborhood of octants for 3D convolution to be performed. It is worth mentioning that both OctNet and O-CNN use a fixed depth for the octree structure which makes these models suitable for 3D shape analysis tasks at a higher resolution but are not capable of performing reconstruction or generation of 3D shapes since the depth of the octree structure is not known beforehand in reconstruction tasks but needs to be predicted.

To perform 3D reconstruction using an octree-like structure, Hne et al. [53] proposed the Hierarchical Surface Prediction (HSP) method. The key idea was to extend the network prediction from binary labeling of free vs. occupied voxel to a three-label prediction adding the prediction for 'boundary' voxels. The HSP model's decoder is used then to predict a data structure, the researchers called 'voxel block octree', in a coarse to fine hierarchical manner. The coarse level is decoded in a straight forward manner, then the decoder predicts the new and finer level through three steps.

Given the feature block at the coarse level which contains information about all children nodes, the first step is to crop the feature space around the child octant of interest, then upsample the cropped feature map to a higher-resolution feature block, then lastly, generate a higher-resolution voxel block. The generation step is performed under the supervision of ground truth labels (i.e. boundary, empty, occupied) at this specific level of the tree, which means ground truth labels must be available for all levels of the tree. The decision to further subdivide a node or not is based on the number of predicted boundary labels at the current level, if it is above a certain threshold then children nodes are added, otherwise, no more children nodes are added and the final output is generated from the cropped features at the previous level.

Another method that predicts 3D shapes using octrees is the Octree Generating Network (OGN) proposed by Tatarchenko et al. [54]. The researchers proposed to start the OGN decoder with a standard voxel prediction block (called dense block) until a certain resolution is reached, after that the feature maps of the dense block, which already represent large uniform regions of the shape, are converted to an octree structure using a hash table for further processing to reconstruct finer details. The subsequent blocks of the decoder are responsible for handling the octree-based feature maps through custom-built convolution and up-convolution layers. Octree blocks are arranged in a sequence where each block is responsible for predictions at a single level of the generated octree. At each octree level, the block responsible for it convolves the previous features with learned weight filters then predict occupancies as either filled, empty, or mixed. The features of the empty and filled cells are discarded and the features of the mixed cells are propagated to the next octree block. The process continues until the desired output resolution is attained.

Wang et al. [55] further developed their previously proposed O-CNN into Adaptive O-CNN. The researchers adopted the same octree-based approach but proposed to represent the shape with adaptive planar patches instead. To this end, they designed a decoder to predict the occupancy probability (patch approximation status) of octants as either empty, surface-well-approximated, or surface-poorly-approximated. Only octants with surface-poorly-approximated labels are further subdivided into the next tree level while propagating their features to their children nodes. Two loss functions are incorporated in the network: a structure loss and a patch loss. The structure loss evaluates the difference between the predicted octree structure and the ground truth using cross-entropy loss, while the patch loss measures the difference between the predicted plane parameters at all leaf octants and the ground truth using squared distance error.

The novel idea of representing leaf octants as planar patches and terminating the subdivision process when these patches

adequately approximate the surface at the associated octant resulted in a generated octree representation that is more compact and adaptive. However, the resultant shapes are not seamless and require further post-processing. Additionally, since planar patches are used, surface curvature is not well represented nor adequately approximated.

In conclusion, Euclidean 3D shape representations provide a structured way to represent 3D shapes for 3D reconstruction tasks. Regular voxel grids are simple to process using 3D CNNs but are inefficient in terms of computation and memory consumption, thus are limited in their output resolution. On the other hand, Octrees provide a structured and hierarchical representation of 3D shapes and are more efficient. However, both representations cannot capture the smoothness of the represented surface, nor can they preserve its intrinsic geometrical properties. These limitations encouraged the researchers to experiment with geometric representations for 3D reconstruction tasks. Table 1 provides a summary of the discussed Euclidean-based methods.

3. Non-Euclidean/geometric approaches

The main types of non-Euclidean geometric data used in 3D analysis and synthesis tasks are point clouds and 3D meshes. Point clouds are sets of unstructured and unordered points scattered in a 3D space that represent the surface of 3D objects. 3D polygonal meshes represent surfaces through connected vertices with coordinates in 3D space. 3D meshes can also be represented as graphs, where nodes represent vertices and edges represents the connectivity between them. In comparison to Euclidean data, non-Euclidean data lack a rasterized grid structure thus there is no global parameterization which makes processing such data challenging. Additionally, since there is no intrinsic ordering of individual points or vertices, it is hard to extract neighborhood information from such points which makes convolutions nontrivial to implement.

3.1. Point clouds as an output

PointNet [56] and its successor PointNet++ [57] are the first architectures that were able to perform 3D analysis on point clouds directly. These models learn features directly from point cloud inputs and are capable of performing shape classification, part segmentation, and scene segmentation. PointNet solved two challenges inherent in point cloud representation. (1) The network needed to be invariant to permutations since points are unordered. (2) The network needed to be invariant to geometric transformations so that results are not altered by point cloud rotations for example. These two properties were achieved by proposing an architecture that consisted of a series of individual and identical layers of Multi-Layer Perceptrons (MLP), a symmetric max-pooling function, then another MLP to output a global feature vector. For transformation invariance, the researchers incorporated within the architecture a small network that predicts an affine transformation matrix and applies it to the coordinates of input points. PointNet++ introduced hierarchical feature learning by using PointNet on local regions then aggregating the local features.

Nash and Williams proposed ShapeVAE [58], a point cloud-based generative model for describing and generating part segmented 3D objects. This model is based on an autoencoder and a low-dimensional embedding space. Novel shapes can be sampled from the embedding space. Additionally, since points' orientations are part of the shape representation, 3D mesh reconstruction can be performed on the sampled points. However, this approach is capable only of synthesizing new shapes but not reconstructing shapes from images, since it depends on an input of oriented and part segmented point clouds. Another point cloud based autoencoder is FoldingNet proposed by Yang et al. [59]. FoldingNet's encoder is similar to PointNet. Its decoder works by deforming a 2D uniform grid of points into the desired output through a two-step-folding network of MLPs.

An extensive study on generative models learning latent representation using point clouds was done by Achlioptas et al. [60]. The researchers explored autoencoders, GANs, and Gaussian Mixture Models (GMM). They also experimented with different evaluation metrics: the Jensen-Shannon Divergence (JSD), coverage (COV), and Minimum Matching Distance (MMD). The coverage and MMD are calculated using both Chamfer Distance (CD) and Earth Mover's Distance (EMD). Their proposed model was tested on shape completion and classification tasks. A different approach toward using GANs as a generative model for point cloud sets was proposed by Li et al. [61] with the possibility of incorporating an image regressor for image to point cloud transformation. More recently, Sun et al. proposed PointGrow [62], an autoregressive model for generating point cloud shapes with the possibility of generating shapes conditioned on 2D images.

Closely related to the work done by Achlioptas et al. [60] is the generative model recently proposed by Valsesia et al. [63]. The researchers however proposed a graph-convolutional GAN for point cloud generation. Their main contribution is the generator that is capable of learning localized features using graph convolutions even when the structure of the graph is not known a priori. The researchers defined a graph convolution operation based on edge-conditioned convolution [64] for graphs, where at each layer and for each node in the graph, the feature vectors of the next layer are computed by performing a weighted local aggregation of the feature vectors of the node's neighbors. An upsampling operation was also defined based on local aggregation in a similar fashion to the graph convolution operation but with using diagonal weight matrices instead of dense matrices. The researchers incorporated the same discriminator proposed by Achlioptas et al. [60] and trained a Wasserstein GAN with gradient penalty.

The first to explicitly use point clouds for single-view 3D reconstruction were Fan et al. [65]. The researchers proposed the Point Set Generation (PSG) network, and in their work, they experimented with different network design alternatives and different loss functions. They proposed a simple vanilla architecture shown in Fig. 5(a), a two-branch version shown in Fig. 5(b), and an hourglass version shown in Fig. 5(c). The encoder was the same in the three architectures and was fed an input image and a random variable. The purpose of the random variable is to be incorporated in modeling shape uncertainty and in generating a distributional output instead of a single output. The predictor in the vanilla version is a fully connected network that produces a matrix where each row is the Cartesian coordinates of a single point. The two-branch predictor has the fully connected network of the vanilla version to predict the shape's intricate and fine structures and an upconvolution network to predict the shape's main body. The two results are combined through a set union to generate the final shape. The hourglass version is a deep network that performs encoding and decoding recurrently to make better use of local and global information. The researchers experimented with two loss functions the CD and EMD. The result of their work outperformed the 3D-R2N2 architecture using three different metrics: CD, EMD, and Intersection-over-Union (IoU).

The ShapeMVD proposed by Lun et al. [66] is a model that is capable of reconstructing 3D shapes from both single-view and multi-view sketches through intermediate depth and normal maps that are eventually fused into a point cloud output. The network has a separately trained convolutional encoder for different input view configurations (only front view sketch available, only side view sketch available, both front and side view sketches available,

Table 1
A summary of the methods that use Euclidean representations. First section: Non-generative 3D supervised methods. Second section: Generative 3D supervised methods. Third section: 2D supervised methods. Fourth section: Octree methods. Method name links to official code (digital version). IoU: Intersection over Union, AP: Average Precision, CD: Chamfer Distance, EMD: Earth Mover's Distance, PSNR: Peak Signal-to-Noise Ratio.

Method	Year	3D Representation	Model Architecture	Loss Function	Output Resolution	Dataset(s)	Evaluation Metric
https://github.com/chrischoy/3D-R2N2 3D-R2N2 [22]	2016	Voxel	Encoder - 3D convolutional LSTM - decoder	Voxel-wise cross entropy	32^3	ShapeNet Online Products Pascal3D+	IoU
IDCT [47]	2017	Voxel	Encoder - inverse discrete cosine transform decoder	Voxel-wise cross entropy	128^3	ShapeNet Pascal3D+	IoU
https://github.com/syb7573330/im2avatar Im2Avatar [24] (CVN)	2018	Voxel	Encoder - decoder (for both shape and color)	Mean Squared False Cross-Entropy Loss	64^3	ShapeNet Colorful Human	IoU (shape) Surface PSNR (color)
http://ns.inria.fr/d3/3DSketching/ 3D sketching [26]	2018	Voxel	Single view CNN: U-net encoder-decoder Updater CNN: unrolled recurrent network	Multinomial logistic loss	64^3	Shape COSEG Procedural shapes	IoU
https://bitbucket.org/visinf/projects-2018-matryoshka/src/master/ Matryoshka Network [27]	2018	Voxel	Encoder - decoder (with residual blocks)	BCE, L1, L2, cosine Similarity, approx. IoU	256^3	ShapeNet	IoU
https://github.com/xiumingzhang/GenRe-ShapeHD ShapeHD [28]	2018	Voxel	2.5D sketch estimator (encoder: ResNet-18 - decoder) 3D shape estimator (encoder - decoder) Shape Naturalness network (WGAN)	Voxel loss (BCE) Naturalness loss	128^3	ShapeNet Pascal3D+ Pix3D	IoU CD
https://github.com/xiumingzhang/GenRe-ShapeHD GenRe [31]	2018	Depth (intermediate) Voxel (output)	Depth estimator - spherical inpainting network - voxel refinement network	Voxel loss (BCE) Depth estimator (L2)	128^3	ShapeNet Pix3D	CD
https://infinitescript.com/project/pix2vox/ Pix2Vox [32]	2019	Voxel	Encoder - decoder - context-aware fusion - refiner	Voxel-wise cross entropy	32^3	ShapeNet	IoU
TPWCoder [33]	2020	2.5D depth/normal maps (intermediate) Voxel (output)	MarrNet + TPWCoder (encoder-decoder)	Reconstruction loss (BCE) Topology loss (CE)	128^3	Pascal3D+ ABC ShapeNet	IoU CD EMD
https://github.com/rohitgirdhar/GenerativePredictableVoxels T-L Embedding [34]	2016	Voxel	Encoder-decoder	Cross entropy loss	20^3	ShapeNet	AP
https://github.com/zck119/3dgan-release 3D-GAN [30]	2016	Voxel	Image network (Convnet) 3D GAN + VAE	Euclidean loss BCE KL divergence loss	64^3	ShapeNet ModelNet	AP

(*continued on next page*)

Table 1 (*continued*)

Method	Year	3D Representation	Model Architecture	Loss Function	Output Resolution	Dataset(s)	Evaluation Metric
https://github.com/ajbrock/Generative-and-Discriminative-Voxel-Modeling voxel-based VAE [42]	2016	Voxel	VAE	Modified BCE KL divergence loss L2 regularization	32^3	ModelNet	AP
https://github.com/EdwardSmith1884/3D-IWGAN 3D-IWGAN [40]	2017	Voxel	3D GAN + VAE	Wasserstein distance with gradient penalty	32^3	ModelNet ShapeNet Ikea dataset	AP
https://github.com/lorenmt/vsl VSL [43]	2018	Voxel	Image regressor: 2D convnet Encoder - decoder	Reconstruction loss L2 regularization - KL divergence	30^3	ModelNet Pascal3D+	IoU
Struct-aware [44]	2018	Voxel	VAE + structure detector	VAE: reconstruction loss + KL divergence loss Structure consistency loss	64^3	ShapeNet	IoU
https://github.com/xcyan/nips16_PTN PTN [45]	2016	Voxel	Encoder - volume generator - perspective transformer	Volumetric loss Silhouette based loss	32^3	ShapeNet	IoU
https://github.com/jgwak/McRecon McRecon [46]	2017	Voxel	Encoder - generator - discriminator	Reprojection error (cross entropy loss) Discriminator loss (log barrier method)	32^3	ObjectNet3D Ikea dataset	IoU AP
https://github.com/jiajunwu/marrnet MarrNet [29]	2017	2.5D depth/normal maps (intermediate) Voxel (output)	2.5D sketch estimator (encoder: ResNet-18 - decoder) 3D shape estimator (encoder-decoder)	BCE Depth/normal reprojection loss	128^3	ShapeNet Pascal3D+	IoU
https://github.com/shubhtuls/drc DRC [25]	2017	Voxel	Encoder - decoder	Ray consistency loss	32^3	ShapeNet Pascal VOC	IoU
https://github.com/stevenygd/3d-recon 3D-recon [48]	2018	Voxel	Encoder - generator - discriminator - projection module	Reconstruction loss Pose invariance loss	32^3	ShapeNet	IoU AP
Shape Estimation for robotics [49]	2019	Voxel	Image encoder shape decoder viewpoint regressor	Squared Euclidean loss	32^3	ShapeNet Pix3D	Shape estimation (IoU Hausdorff distance) Viewpoint estimation (Median Angular Error)
https://github.com/chaene/hsp HSP [53]	2017	Octree	Encoder - decoder	Cross entropy	256^3	ShapeNet	IoU CD
https://github.com/lmb-freiburg/ogn OGN [54]	2017	Octree	Encoder - decoder	Cross entropy	512^3	ShapeNet	IoU
https://github.com/Microsoft/O-CNN Adaptive O-CNN [55]	2018	Octree	Encoder - decoder	Structure loss (cross entropy) Patch loss (squared distance error)	128^3	MPI-FAUST ShapeNet	CD

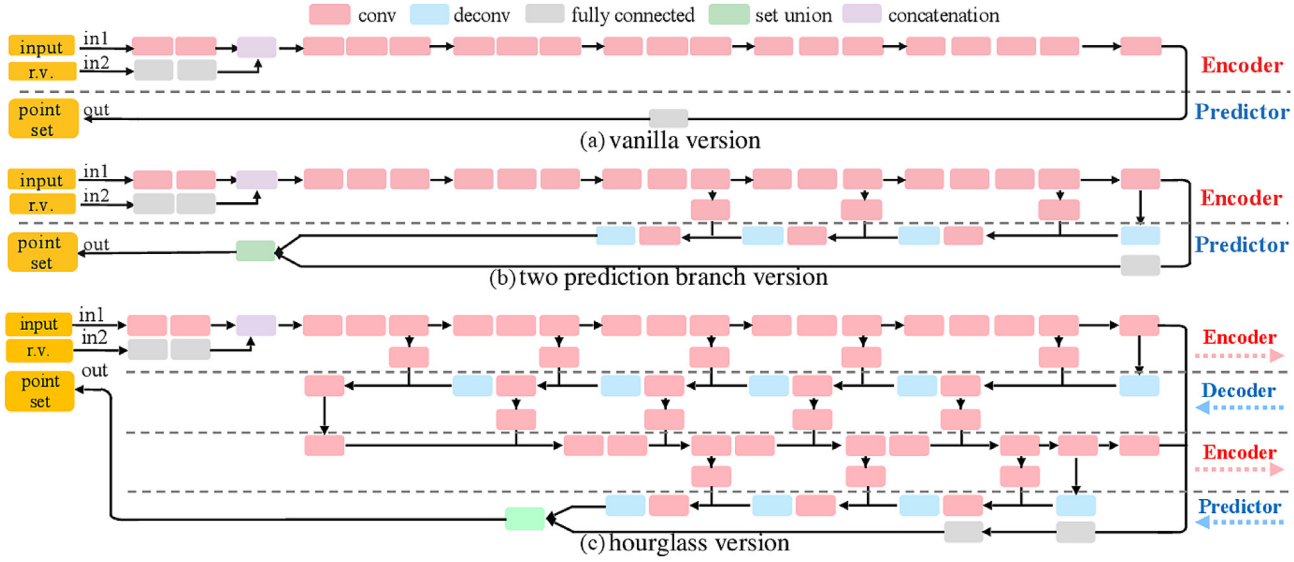

Fig. 5. The three different PSG versions as originally shown in [65]. The three versions have the same encoder architecture. (a) The vanilla version has a fully connected decoder. (b) The two-branch version has additional deconvolutional operations. (c) The hourglass version performs encoding and decoding simultaneously.

etc.). The encoded image features are then fed to a 12-branch decoder that outputs depth and normal maps from 12 different viewpoints. The encoder and decoder are linked in a U-Net [67] pattern where each layer in the decoder is connected with the corresponding layer in the encoder. To ensure output plausibility an adversarial training is also incorporated. The final outputs of normal and depth maps are fused into a consolidated point cloud. Further post-processing is proposed to reconstruct a mesh-based shape and refine it.

Kurenkov et al. [68] approached the problem of single-view 3D reconstruction through first performing an image-based shape retrieval, then by applying a learned deformation on the retrieved shape template. The proposed DeformNet model is composed of two encoders, a decoder, and a differentiable Free-Form Deformation (FFD) layer. The first encoder is a 2D image encoder for encoding the input image. The second encoder is a 3D-CNN that encodes the retrieved shape template in a voxel grid structure. The decoder is connected to the 3D encoder through skip connections in a U-Net hourglass pattern and is fed the combined input of the two encoders to predict a deformation vector field used as the offset for the control points of the subsequent FFD layer. Lastly, the FFD generates the final deformed shape in point cloud representation using the deformation vector field.

Mandikal et al. [69] explored creating a probabilistic latent space for reconstructing multiple possible shapes from a single image by proposing 3D-LMNet, a latent embedding matching network. The simple model incorporates a 3D point cloud autoencoder and an image encoder. A two-step training process is carried out. First, the 3D autoencoder is trained by minimizing a CD-based reconstruction loss, and in the second step the 3D encoder parameters are fixed and the difference between the two latent codes produced from the 3D encoder and 2D encoder is minimized using a latent matching loss (L1/L2 loss). The probabilistic variant of the model has the same 3D autoencoder and a modified image encoder that learns to map the image into a probabilistic representation by predicting the mean and standard deviation of the distribution just like VAEs but without constraining the mean of the distribution.

To enforce global geometric consistency, Jiang et al. [70] proposed a Geometric Adversarial Loss (GAL) model that incorporates an adversarial loss function in addition to the local pointwise CD-based loss function. The adversarial loss has two terms: a multi-view geometric loss that acts as a global shape constraint and a conditional adversarial loss that acts as a semantic constraint. The multi-view loss requires first projecting the predicted and ground truth shapes into multiple 2D images from different views using a projection function, then the loss is computed in two modes: a high-resolution mode and a low-resolution mode. The high-resolution mode loss aims to force the predicted points to lie in the manifold of ground truth shape. The low-resolution mode loss aims to enforce geometric consistency as a result of the bi-directional geometric constraint which makes the predicted points cover the whole shape of the ground truth model. The multi-view loss is then computed as the sum of the two modes' losses. To compute the conditional adversarial loss, a discriminator is built with two components: a PointNet [56] 3D point cloud feature extractor and a pre-trained VGG [71] 2D feature extractor. The extracted features from both networks are concatenated and the discriminator loss is computed by least squared error as in LSGAN [72]. The total objective function then incorporates all these losses to enforce both global and local consistency in the predicted point cloud shape.

Again using a generative model, Lin et al. [73] proposed a method that uses 2D convolutional networks to efficiently generate 3D shapes in dense point cloud representation. The proposed model has a convolutional image encoder, a convolutional structure generator, and a pseudo-renderer. Given the latent representation produced by the image encoder, the structure generator uses 2D convolutional operations to generate a number of multi-channel images representing the coordinates of points on the 3D shape surface from different viewpoints. All the generated views of the shape structure are then fused into canonical coordinate space through orthographic projection. The pseudo-renderer is then used to produce depth maps that are used along with silhouettes in calculating the loss using cross-entropy for silhouettes and L1 loss for depth maps.

Along the same lines, Insafutdinov and Dosovitskiy [74] proposed an architecture that jointly learns both pose and shape of 3D objects represented as point clouds also using only 2D projections for supervision. This category-specific approach learns from a multi-view unlabeled image collection. The proposed architecture contains a convolutional image encoder followed by two fully connected layers, then it branches into a shape predictor which

is an MLP, and a pose predictor which is also an MLP. The model also incorporates a differentiable point cloud renderer that projects the learned shape using the learned pose into a 2D view which is used in loss calculations using Mean Squared Error (MSE) against ground truth projection. To avoid the possibility of the pose predictor getting stuck in local minima caused by pose ambiguity, the researchers proposed a more robust way of estimating the pose by using an ensemble of pose regressors that learn to predict the pose individually and only updating the weights of the network that predicts the best matching pose to that of the ground truth projection. Additionally, the ensemble is distilled into a single regressor that is used at test time instead of the whole ensemble.

Mandikal and Radhakrishnan [75] proposed a hierarchical approach toward dense point cloud reconstruction using a deep pyramidal network. The proposed framework progressively scales up the resolution of an initial sparse point cloud generated by a separate sparse reconstruction network using EMD loss. This is considered the first step in a multi-stage reconstruction process. Then the sparse reconstruction is fed into a dense reconstruction network that has three modules: one for global feature learning, one for local feature learning, and one for feature aggregation and grid conditioning. The global feature learning module utilizes a number of shared MLPs that work on individual points on the shape followed by a point-wise max pooling operation to eventually learn global shape properties. To learn local shape features, a number of shared MLPs operate on a neighborhood of points to produce neighborhood features. These features are then pooled to produce local features for each individual point. To increase the resolution of the point cloud, the feature aggregation module is used to concatenate all the learned features and the sparse point cloud prediction and tile them, and to disperse the extra generated points a local 2D grid deformation is applied to help in feature propagation. The concatenated and tiled features along with the 2D grid are fed to yet another set of shared MLPs to predict a four times denser shape. The whole dense reconstruction network is applied twice to eventually obtain a 16 times denser point cloud. The output of each dense reconstruction step is compared to an equivalent sized ground truth shape using CD loss instead of EMD loss to ease memory consumption.

A self-supervised learning approach for single-view reconstruction was proposed by Sun et al. [76]. The researchers proposed SSL-Net which performs the final point cloud generation from a single image but uses components from two other pre-trained networks that autoencode point cloud shapes and binary image masks separately. The 3D point cloud autoencoder contributes the shape decoder and shape latent features to the final SSL-Net model, while the 2D binary image autoencoder contributes the image decoder and binary mask latent features. The end-to-end training process involves encoding the input image using a different image encoder to produce new shape latent features which are to be approximated to the shape latent features of the pre-trained 3D autoencoder (using L1 loss). The pre-trained shape decoder then uses the approximated shape latent features to generate a preliminary point cloud shape. After that, a different shape encoder is used to encode the preliminary point cloud shape into new binary mask latent features which are to be approximated to the binary mask latent features of the pre-trained 2D image autoencoder (using L1 loss). The pre-trained image decoder takes the approximated latent features and predicts a binary mask. The difference between the predicted mask and the mask of the input image is minimized using 2D MSE loss function and used to achieve self-supervision by minimizing the difference between the rendered image of the generated point cloud and the binary mask of the input image to the network. A pose estimation network is also needed to learn the pose of the input image so that the generated point cloud can be rendered in the same pose as the input image.

Lu et al. [77] recently proposed an attention-based approach toward dense point cloud generation from single-view inputs. The proposed framework that can be considered an attempt to improve [75] has also two separately trained networks that are fine-tuned together. The role of the first network, which follows an encoder-decoder pattern, is to generate a sparse point cloud shape from a single image. This network's encoder has a series of consecutive convolutions and attention modules with residual blocks. The attention mechanism ensures that the encoder learns specific details of the shape's structure (such as edges and legs in the chair category) which improves the learnability of the network for synthesis tasks over the regular image encoder networks used for classification tasks. The second network densifies the output of the sparse network to produce a 16 times denser shape. The dense point cloud generation network has two dense modules each contains a feature extraction module and a feature expansion module that are connected through a feature interpolation mechanism. The role of the feature extraction module is to learn both local and global features and concatenate them. The role of the feature expansion module is to increase the number of output points through a replication and concatenation process with a number of 2D grids which allows the module to learn how to distribute the dense points along the shape's surface and the final point cloud coordinates are achieved through a number of MLPs. In contrast to [75] which has a similar intention as mentioned before, this approach applies an attention mechanism, does not require intermediate ground truth shapes for loss calculations, and provides a communication mechanism between the dense modules.

Point clouds offer a convenient and efficient representation for 3D shapes since they capture the shapes surface only. They are also easy to acquire using depth-sensing devices, which makes it feasible to collect large datasets of ground truth point clouds. However, because of their unstructured nature, lack of connectivity, and irregularity, they still pose a challenge for tasks that go beyond pure 3D reconstruction. Table 2 provides a summary of the discussed point-cloud-based methods.

The point cloud output of 3D reconstruction models requires further post-processing to be usable in computer graphics, virtual reality, and robotics domains. These domains can directly utilize 3D meshes, which motivated the researchers to work on reconstructing shapes using 3D mesh representation.

3.2. 3D Mesh as an output

The methods that utilize a 3D mesh representation as an output can be further divided into two subcategories: 2D supervised methods and 3D supervised methods. Most 3D supervised methods rely on learning the deformation of a shape template or a base mesh to generate the desired output shape. On the other hand, most 2D supervised methods rely on integrating a differentiable neural renderer in their pipeline to compute an image-based loss function.

3.2.1. 2D Supervised methods

One of the pioneering works in learning mesh-based 3D reconstruction from a single image was proposed by Kar et al. [23]. The proposed model is class-specific and learns from a number of annotated images a deformable shape model that is capable of capturing intra-class shape variations. Annotated images are used to jointly estimate camera viewpoints for all instances within a class using a non-rigid SfM [78] model and Expectation-Maximization (EM) algorithm for maximizing the likelihood of the model. Then the deformable shape model is formulated by utilizing the camera parameters, keypoint correspondences, and image silhouettes. The objective function used enforces several forms of consistency such as silhouette consistency, silhouette coverage, and keypoint

Table 2
A summary of the methods that use point cloud representation. Method name links to official code (digital version). IoU: Intersection over Union, CD: Chamfer Distance, EMD: Earth Mover's Distance, JSD: Jensen-Shannon Divergence, COV: Coverage, MMD: Minimum Matching Distance.

Method	Year	Model Architecture	Loss Function	Output Resolution	Dataset(s)	Evaluation Metric
https://github.com/fanhqme/PointSetGeneration PSG [65]	2017	Encoder (convnet)	CD and EMD	1024	ShapeNet	CD
		Vanilla predictor (fully connected)				EMD
		2-branch predictor (deconvolution branch - fully connected branch)				IoU
https://github.com/happylun/SketchModeling ShapeMVD [66]	2017	U-net encoder - multi-view decoder	Per-pixel depth loss (L1)	NA	ShapeNet	CD
			Normal loss (angle cosine difference)		"The models resource"	IoU
			Mask loss (cross entropy) adversarial loss			Hausdorff distance
https://github.com/diegovalsesia/GraphCNN-GAN GraphCNN-GAN [63]	2018	Graph-convolutional GAN	Wasserstein distance with gradient penalty	2048	ShapeNet	JSD COV MMD
https://deformnet-site.github.io/DeformNet-website/ DeformNet [68]	2018	Image encoder - voxel encoder - voxel decoder - free-form deformation layer	L1, L2, CD, EMD	1024/16384	ShapeNet	CD
						EMD
https://github.com/val-iisc/3d-lmnet 3D-LMNet [69]	2018	Image encoder - autoencoder	Reconstruction loss (CD)	2048	ShapeNet	CD
		Image encoder - variational autoencoder	Latent matching loss (L1/L2)		Pix3D	EMD
			Diversity loss			
GAL [70]	2018	Generator (PSG - hourglass version)	Conditional adversarial (least square) loss	1024	ShapeNet	CD
		Conditional discriminator (PointNet + VGG CNN)	Multi-view geometric loss			IoU
			CD loss			
https://github.com/chenhsuanlin/3D-point-cloud-generation 3D-point-cloud-generation [73]	2018	Convolutional image encoder - convolutional structure generator - pseudo-renderer	Mask loss (cross entropy)	variable	ShapeNet	3D Euclidean distance
			Depth loss (L1)			
https://github.com/eldar/differentiable-point-clouds Differentiable Point Clouds [74]	2018	Convolutional image encoder - Shape predictor (MLP) - pose predictor (MLP) - differentiable point cloud renderer	MSE	Up to 16,000	ShapeNet	CD
https://github.com/val-iisc/densepcr DensePCR [75]	2019	Sparse: convolutional image encoder - fully connected layers	CD loss and EMD	16,384	ShapeNet	CD
		Dense: sequence of shared MLPs			Pix3D	EMD
SSL-Net [76]	2019	3D autoencoder 2D binary autoencoder pose estimator	Reconstruction loss (CD) 2D binary loss (MSE) Latent feature loss (L1)	2048	ShapeNet	CD EMD
https://github.com/VIM-Lab/AttentionDPCR AttentionDPCR [77]	2019	Sparse generator (encoder - decoder with attention module and skip connections)	CD loss and EMD	16,384	ShapeNet	CD
					Pix3D	EMD
		Dense generator (Feature extraction - feature expansion modules using MLPs)				

consistency. The researchers further refined their work [79] and proposed an alternative CNN based system for viewpoint prediction.

The work done by Sinha et al. [80] was an extension of their work done in [13] which was based on geometry images as an intermediate representation for 3D surfaces. Geometry images were used to map arbitrary surfaces into a regular grid structure with three different maps for x, y, and z axes. The researchers created the geometry images by first performing an authalic parameterization of a single shape per class, then establishing a correspondence between the other shapes and this base surface. This was done to create consistent geometry images to be fed to the network. The network itself is a deep residual network that follows the encoder-decoder pattern with upsampling, downsampling, and standard residual blocks. Each channel in the geometry image needed a separate network with the same architecture. The experiments done with this model included both rigid and non-rigid meshes but were limited to genus-0 surfaces only.

A slightly different approach towards 3D mesh reconstruction uses the inverse rendering method in which an approximate differentiable renderer is integrated into the neural network as shown in Fig. 6. Kato et al. [81] adopted this approach in a fashion

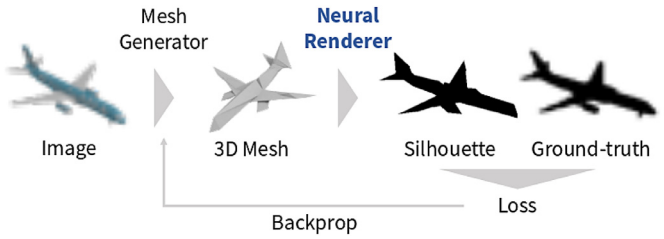

Fig. 6. The neural 3D mesh renderer, as originally shown in [81], takes a 3D mesh and renders it to produce a silhouette that is used in loss function computation.

similar to PTN [45] but for mesh reconstruction instead of voxel reconstruction. The main challenge facing such techniques is the discrete rasterization step in their pipeline which is by nature non-differentiable (since the derivative of a pixel with respect to coordinates of a vertex is always zero) thus preventing backpropagation. To overcome this problem, Kato et al. proposed an invisible gradual change instead of the sudden discrete change to allow for backpropagation. The proposed model deforms an isotropic sphere with 642 vertices under the supervision of silhouettes in an encoder-decoder network architecture. The researchers also experimented with 2D to 3D style transfer with successful results.

Kanazawa et al. [82] proposed to learn textured mesh reconstruction from image collections. This category-specific approach only requires 2D supervision using images, landmark keypoints, and silhouettes instead of the usual 3D shape or multi-view supervision. The proposed framework includes a ResNet image encoder [83] and three different prediction modules. The first prediction module estimates the camera pose by matching it to a previously obtained camera pose calculated by applying SfM on the landmark keypoints. The second predictor is a deformation module that learns to offset a category-specific learned mean shape (learned from all instances in the image collection thus instance-independent). The third prediction module predicts texture through texture flow. The framework also uses the neural mesh renderer proposed by Kato et al. [81] to render the predicted image masks and textured meshes for loss calculations. In addition to the loss functions that minimize the difference between the predicted mask, textured mesh rendering, and estimated keypoints and their respective ground truth values, the researchers incorporated several priors such as symmetry constraints and smoothness and deformation regularizations.

Henderson and Ferrari proposed the first probabilistic generative framework for 3D mesh reconstruction with only 2D supervision [84]. The framework consists of an encoder and a generative module that has a decoder and a differentiable renderer. The decoder learns to disentangle the shape and pose information extracted from a single unannotated color image. The decoder predicts the mesh parameters that are then fed to a fixed mesh parameterization function that produces the mesh's vertices. The mesh is then passed to the differentiable renderer which renders an image of the shape with the aid of the estimated pose from the encoder. The difference between the rendered image and the input image is minimized through the image log-likelihood loss function thus achieving mesh reconstruction through 2D supervision only. The researchers experimented with different parameterization functions and with different supervision modalities. They also incorporated shading information in calculating the loss by experimenting with different lighting conditions using both colored and white lighting.

Another work that integrates a differentiable renderer layer within network architecture was recently proposed by Liu et al. [85]. The researchers aimed to provide a more accurate differentiable renderer that truly approximates the discrete and non-differentiable rasterization step. In comparison to the Neural 3D Mesh Renderer [81] which only approximates the backward gradient computation, their proposed Soft Rasterizer can be used in both the forward and backward passes. Additionally, Soft Rasterizer provides an enhanced approximation function that is claimed to be more accurate than the linear function proposed by Kato et al. [81]. Soft Rasterizer renders silhouettes using a probabilistic procedure by creating a probability map for each triangle in the mesh. Each probability map describes the probability of a certain triangle covering each pixel in the rendered image. The probability maps of all triangles are fused using a differentiable aggregate function that approximates the logical OR operator used in discrete rasterization. The whole model uses a generator module with an encoder-decoder network similar to [81] that outputs a deformed mesh, and a Soft Rasterizer layer which produces a rendered silhouette of the deformed mesh to be used in 2D supervision against ground truth silhouette of the input image with loss computed using IoU.

Kato and Harada [86] utilized view-based training for 3D mesh reconstruction. The proposed model does not require 3D supervision, instead, it requires images with annotated viewpoints and silhouettes similar to the view-based methods discussed earlier in voxel-based reconstruction. The model has an image encoder for image features extraction, a shape decoder and a texture decoder for mesh reconstruction, a modified neural mesh renderer [81] for mesh rendering, and a discriminator for learning view priors through an adversarial interaction with the encoder and decoders. The researchers performed an extensive ablation study testing single-view training, multi-view training, and class conditioning with results showing better accuracies than methods that utilize 3D supervision especially using single-view training.

Petersen et al. [87] proposed another framework that integrates a differentiable renderer in the pipeline of 3D mesh reconstruction. The researchers based their work on the observation that the output of differentiable renderers is stylistically different from input images and images rendered using off-the-shelf renderers, so they proposed integrating two domain translation components to translate one style into the other. Additionally, the proposed differentiable renderer can handle occlusions and dis-occlusions without causing discontinuities thus allowing backpropagation of gradients. The framework contains a mesh reconstructor made of a ResNet image encoder followed by two fully connected layers. The output is used to deform a base mesh sphere through offsetting its vertices. Then the smooth renderer is used to render the mesh (into one style and yet to be translated into the other style). The reconstructor is part of a Reconstructive Adversarial Network (RAN) that performs self-supervision through a non-trivial series of training steps using 5 simpler sub-RANs. The discriminator in each sub-RAN has the task of discriminating between a pair of inputs to decide which is fake and which is real. In this context, being fake or real depends on the aim of the sub-RAN. Eventually, the whole process allows the output of the renderer and the two domain translation components to be trained together. The system requires multi-view images for training with the possibility of inferring shape from single-view inputs.

Another recent work that utilizes multi-view supervision was proposed by Xiang et al. [88] using normal maps for 2D supervision instead of the usual silhouettes. The proposed model is capable of generating a 3D mesh and estimating camera pose using a normal mismatch loss function. The framework is divided into a normal map generator and a mesh predictor. The normal map generator is based on a Conditional Generative Adversarial Network (CGAN) [89] that generates a normal map conditioned on the input image using adversarial loss. The mesh predictor follows the encoder-decoder pattern and uses for supervision multi-view normal maps that describe both the geometric surface details and the

silhouette of the shape. The neural mesh renderer [81] is again used to render the normal maps in specific viewpoints that are used in the normal mismatch objective function. In addition to the normal loss function, smoothness and edge length geometric constraints are incorporated to regularize the deformation of the output mesh.

Li et al. [90] tackled the problem of 3D mesh reconstruction using a self-supervised approach. The proposed work extends CMR [82], however, without the need for 2D keypoint annotations nor a class-specific mean shape as supervisory signals. To perform self-supervised training, the researchers divided the task into two objectives each is achieved using a separate module and both modules are trained using a progressive training paradigm using the Expectation-Maximization approach. The first module aims at reconstructing mesh instances along with textures and camera pose using only a collection of category-specific images and their silhouettes for supervision. The second module aims at learning a category-specific semantic mesh template. The proposed model also incorporates the Soft Rasterizer [85] differentiable renderer. The main contribution of this work is the semantic consistency constraint that allows for self-supervision without the need for 2D annotations while minimizing shape-camera ambiguities. The main intuition behind this is that instances across a category can share the same semantic part segmentation despite their differences in shape. The researchers proposed wrapping a learned category-specific canonical semantic UV map unto the learned shape template to describe the part segmentation shared by all instances of the category at hand. During training, the difference between the semantic rendering of the predicted shape and the semantically segmented input image is minimized thus achieving semantic consistency.

3.2.2. 3D Supervised methods

The works by Kong et al. [91] and Pontes et al. [92] explored the idea of graph embedding CAD models using local dense correspondence to eventually create a deformable dense model for single image 3D reconstruction. The core concept of these methods is creating a deformable model from local correspondences between a collection of shapes from a single category and fitting the best shape model to be similar to the image at hand. The input to the model is a segmented image from a known category with landmark annotations and silhouette. Landmark registration and silhouette fitting are performed to ensure the fitting of the best matching shape from the graph and the input image. The shape model is deformed using a linear combination between the best-selected CAD model and its neighbors in the correspondence graph through the non-rigid Iterative Closest Point (ICP) algorithm [93]. Pontes et al. extend the work by using FFD in addition to ICP to deform the shape thus resulting in more detailed, less distorted, and more realistic results.

To eliminate the requirement of landmarks and silhouettes, Pontes et al. [94] again extended their work to make use of their previously proposed graph embedding model within a learning framework. The framework has three distinct networks. The first one is a convolutional autoencoder that is fed the input image to learn the image latent space, a lower dimension representation that has the useful features extracted from the image. The latent representation is fed into two networks, a multi-label classifier, and a feedforward network. The role of the multi-label classifier is to learn the mapping between the latent representation and the different shapes that are present in the CAD graph embedding model, so this network predicts which 3D CAD model is closest to the image by giving a specific index from the graph embedding. The feedforward network estimates the FFD parameters and the sparse linear combination parameters. These parameters are finally used to deform the selected CAD model to produce the final 3D shape reconstruction.

Along the same lines of using the FFD approach, Jack et al. [95] proposed a simple lightweight CNN that simultaneously learns how to select and deform a shape template from single image input to reconstruct a 3D mesh that best fits the input image. The proposed CNN is a slightly modified MobileNet [96] with initial weights from ImageNet [36]. The Network learns to map the input image to a shared feature space, which in turn is mapped to the deformation parameters of each individual shape template. The researchers experimented with different training regimes for imposing diversity in model selection by introducing explicit entropy penalty and/or deformation regularization. They also evaluated the results using different metrics including IoU, CD, and EMD.

A novel approach proposed by Groueix et al. [97] targets directly learning to reconstruct 3D meshes from either a single image or point clouds. The main contribution of their work is their decoder that takes a latent representation and outputs parametric surface elements. The researchers built a decoder that is based on the PSG model [65] with an additional prior that points sampled should belong to a 2D surface. To enforce this prior, they proposed an architecture that constitutes a number of MLPs, each learns to deform a 2D surface by being fed the latent shape representation and the coordinates of a point sampled from a 2D patch and outputs the 3D coordinates of that point. When this process is repeated with several points from the same 2D patch, the resultant 3D points would also belong to an implicit surface with connectivity between these points preserved. The number of learned patches deformations resembles learning an atlas of charts which helps in meshing, tessellation, and texture mapping, hence the name of the proposed model AtlasNet.

Another work that targets direct 3D mesh reconstruction was proposed by Liao et al. [98]. The researchers integrated a Differentiable Marching Cube Layer (DMCL) to their decoder as an alternative to the otherwise non-differentiable Marching Cube (MC) algorithm [99] for mesh construction. This integration allows end-to-end surface learning and prediction instead of resorting to performing the MC algorithm as a post-process. The proposed decoder takes point cloud input (with the possibility of adapting it to be fed a volumetric or image representation) and outputs occupancy probabilities and vertex displacement that are comparable to the output of the traditional MC algorithm of topology estimation and vertices locations. The model incorporates geometric and occupancy losses in addition to smoothness and curvature losses to ensure a highly detailed and smooth surface prediction.

The first to propose the use of Graph Convolution Networks (GCN) [101,102] for mesh-based reconstruction were Wang et al. [103]. In this approach, dubbed Pixel2Mesh, a 3D mesh is represented as a graph, where nodes represent vertices and the connections between the nodes represent the edges between the vertices directly and straightforwardly. However, for this representation to be learned in an end-to-end manner from single color images, special consideration has to be given when designing the model to bridge the gap between the image representation and this graph representation. Wang et al. propose a perceptual feature pooling layer to bridge the two representations. The proposed model has a VGG-16 [71] image encoder to extract image features and a GCN to perform cascaded mesh deformation with the feature pooling layer in between to allow the extraction of features to be leveraged in mesh deformation. Given an image and an initial polygonal ellipsoid shape. The mesh deformation network, which has three deformation blocks, is initialized with the coordinates of the vertices of the ellipsoid. It then pools features from the image encoder and predicts the new vertices locations, then these features are fed to the next deformation block. Between each deformation block, there is a graph unpooling layer with the task of increasing the number

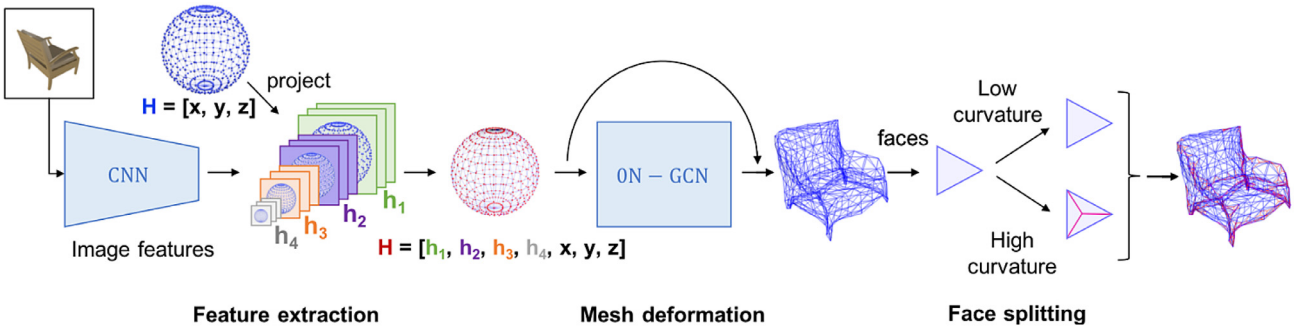

Fig. 7. The components of the GeoMetrics model as originally shown in [100]: Feature extraction (Left), zero-neighbor graph convolution mesh deformation (middle), and adaptive face splitting (Right).

of vertices of the shape to produce higher resolution mesh. The researchers adopted the Chamfer and normal loss functions with additional Laplacian and edge length regularization terms.

Closely related to Pixel2Mesh is the method proposed by Smith et al. [100] called GEOMetrics, which introduces a number of improvements to mesh-based reconstruction methods using GCN. The main insight behind their modifications is that reconstructed meshes need to be adaptive, where the density of the vertices can vary instead of being uniform depending on the surface's curvature. This insight led the researchers to apply an adaptive face splitting algorithm based on the curvature of the surface at a specific region in it. Another improvement proposed is their composite loss function that incorporates novel point-to-point and point-to-surface losses instead of the traditional CD loss function. An additional loss component is a latent loss that requires a pre-trained mesh-to-voxel mapping network where the learned latent code is used in the global loss computation. One final modification is the zero-neighbor update method for vertices during graph convolutions which ensures that each vertex maintains enough information about itself without being smoothed out by information from its neighbors. Fig. 7 shows the end-to-end model with its three components: feature extraction, mesh deformation, and adaptive face splitting.

Similar to AtlasNet, Pan et al. [104] also proposed using MLPs but in a cascaded hierarchical manner for 3D mesh reconstruction. The proposed architecture includes a ResNet-18 [83] image encoder for feature extraction and three blocks of stacked MLPs for hierarchical mesh deformation. The first block contains one MLP and is fed the coordinates of a 2D mesh primitive and image features to perform the primary shape deformation. The consequent blocks have several stacked MLPs that perform deformations on the previously deformed mesh in parallel. The reason behind using parallel paths for mesh deformations is to enrich the mesh representation and parameters using this deep architecture. The output points of the MLPs in each block are concatenated to produce a higher resolution mesh. The model makes use of shortcut connections that form the residual net structure. The researchers adopted CD loss with an additional pair-wise consistency loss between the final deformed mesh outputs to enforce consistency across the meshes.

To address the limitations of template-based deformation techniques, especially being restricted to a predefined topology and their inability to recover meshes beyond genus-0, Pan et al. [105] proposed a novel network architecture that is capable of performing learning-based topology modifications dynamically within the network. The proposed encoder-decoder architecture has a ResNet [83] image encoder and three subnets that act as the decoder to progressively reconstruct the mesh from a single image. The first and second subnets contain a mesh deformation module followed by a topology modification module. The third subnet contains a boundary refinement module. The first mesh deformation module is an MLP network that learns to deform a spherical base mesh, and the second deforms the mesh modified by the topology modification module. The role of the topology modification module is to modify the coordinates and the connectivity of the vertices by removing the faces that produce high reconstruction error above a certain threshold, which dynamically changes the overall mesh topology and allows meshes to go beyond genus-0 despite that the input base mesh is genus-0. The last module enhances boundary edges and smoothes them. One drawback of this approach is that it produces non-closed meshes due to face pruning, which can be managed afterwards by a post-process.

In general, using 3D mesh representation as an output for 3D reconstruction models has drawn more attention recently. Models that output 3D meshes are producing significant results in terms of shape resolution, smoothness, and overall quality. However, several open problems require further attention from the research community. These open problems are discussed extensively in the discussion section. Table 3 provides a summary of the discussed mesh-based methods.

Generally, geometric approaches towards 3D reconstruction provide a more fitting alternative to Euclidean approaches. These approaches can be more efficient in their memory and computation consumption. Also, their outputs can provide a better approximation to ground truth shapes. They handle curvature and smoothness better than the Euclidean approaches. Moreover, with the addition of shape regularization terms to the loss functions, better results can be achieved. The research community is actively engaged in coming up with better models and training paradigms that will help push the current state of research further. See Section 5.4 for further discussion on these efforts.

4. Approaches based on other representations

While the previously covered approaches based on voxel, point cloud, and mesh 3D representations comprise the majority of the work done in the 3D analysis and synthesis domains, researchers endeavored to explore and propose other approaches and relevant network architectures for representing 3D data for such tasks. With respect to 3D reconstruction, two notable approaches can be included to permit for more comprehensive coverage of the subject. These approaches represent 3D shapes as either implicit surfaces such as Signed Distance Functions (SDF) or as a collection of geometric primitives (e.g. cuboids). This section also includes hybrid methods that utilize more than one 3D shape representation in their pipeline.

4.1. Implicit surface representations

The notable work by Li et al. [106] utilized 3D fields as the underlying representation for 3D shapes. The researches proposed a

Table 3
A summary of the methods that use 3D mesh representation. Top section: 2D supervised methods. Bottom section: 3D supervised methods. Method name links to official code (digital version). IoU: Intersection over Union, CD: Chamfer Distance, EMD: Earth Mover's Distance, MSE: Mean Squared Error.

Method	Year	Model Architecture	Loss Function	Output Resolution	Dataset(s)	Evaluation Metric
https://github.com/sinhayan/surfnet SurfNet [80]	2017	Encoder - decoder with deep residual blocks	Euclidean loss	NA	ShapeNet	Euclidean distance
https://github.com/hiroharu-kato/mesh_reconstruction Neural renderer [81]	2018	Encoder- decoder	Shape aware loss Smoothness loss - silhouette loss	642 verts.	Pascal 3D+ ShapeNet	IoU
https://github.com/akanazawa/cmr CMR [82]	2018	ResNet image encoder + camera, deformation, texture predictors	Keypoint reprojection loss + silhouette mask loss + camera regression loss	642 verts.	CUB	IoU (masks)
Mesh Generation & Reconstruction [84]	2018	Encoder - decoder - differentiable renderer	Image likelihood loss	98 verts.	Pascal 3D+ ShapeNet	IoU Pose estimation error and accuracy
https://github.com/ShichenLiu/SoftRas SoftRas [85]	2019	Mesh generator (encoder - decoder) - Soft Rasterizer layer	IoU loss + Laplacian loss + flattening loss	642 verts	ShapeNet	IoU
VPL [86]	2019	ResNet image encoder - shape decoder - texture decoder - discriminator	Reconstruction loss + view discrimination loss + internal pressure loss	1352 verts	ShapeNet Pascal3D+	IoU CD EMD
Pix2Vex [87]	2019	Reconstructor (ResNet + 2FC layers) - smooth renderer - discriminator (RAN)	Binary cross entropy + L2 & L1 regularization	162 verts	ShapeNet	
Generative Normal Map [88]	2019	Normal map (CGAN) Mesh reconstruction (encoder-decoder)	GAN adversarial loss Normal map loss (L1 loss - angular distance loss) Geometric Loss (Smoothness & edge length regularization)	642 verts	ShapeNet	IoU L1 distance
https://github.com/NVlabs/UMR UMR[90]	2020	Reconstruction network (CMR) semantic template module Differentiable renderer (Soft Rasterizer)	Reconstruction network (-IOU perceptual loss) Semantic consistency (CD MSE)		Pascal3D+ CUB ImageNet OpenImages	Mask reprojection accuracy (IoU) Keypoint transfer accuracy
https://github.com/jhonykaesemodel/image2mesh Image2Mesh [94]	2018	Convolutional autoencoder - multi-label classifier - feedforward network	Classifier: soft margin loss function CAE: MSE FFN: MSE	NA	ShapeNet PASCAL3D+	Classifier: accuracy, precision, recall FFN: MSE Reconstruction: symmetric surface distance, IoU
https://github.com/jackd/template_ffd Template FFD [95]	2018	Modified MobileNet CNN	Weighted chamfer loss + entropy penalty / deformation regularization	NA	ShapeNet	CD EMD IoU
https://github.com/ThibaultGROUEIX/AtlasNet AtlasNet [97]	2018	Image/point cloud encoder - decoder (MLPs)	CD Loss	Up to 125 patches	ShapeNet	CD Hausdorff distance (METRO)
https://github.com/yiyiliao/deep_marching_cubes DMC [98]	2018	Encoder (PointNet++) - decoder with skip connections	Geometric loss + occupancy loss + smoothness and curvature loss	NA	ShapeNet	CD - accuracy - completeness
https://github.com/nywang16/Pixel2Mesh Pixel2Mesh [103]	2018	Image encoder (VGG) - graph convolution network	Chamfer loss + normal loss + Laplacian regularization + edge length regularization	2466 verts.	ShapeNet	F-score CD EMD
ResMeshNet [104]	2018	ResNet image encoder - stacked blocks of MLPs	CD + pair-wise consistency regularizer	Up to 30,000 verts	ShapeNet	CD
https://github.com/EdwardSmith1884/GEOMetrics GEOMetrics [100]	2019	Mesh-to-voxel network (encoder-decoder) Mesh reconstruction (CNN image encoder - GCN)	Surface sampling loss (point-to-point loss / point-to-surface loss) + latent loss + Laplacian regularization + edge length regularization	Mean 574 verts.	ShapeNet	F-score
Topology Modification Network [105]	2019	ResNet image encoder - decoder (MLP based subnets)	CD loss - quadratic loss Smoothness - normal - edge length regularization	2562 verts	ShapeNet Pix3D	CD EMD

Field Probing Neural Network (FPNN) that operates on 3D fields for the extraction of 3D shape features for 3D analysis tasks. The proposed model, motivated by the inefficiency of the voxel-based approaches, samples the input 3D fields using a number of probing filters instead of 3D convolutions in a more computationally tractable manner. 3D distance fields are first obtained by applying a distance transform on the cells of a binary voxel grid of the shape. The features are then extracted through the field probing layers to produce an intermediate representation of the shape. These layers can be integrated/extended with task-specific inference layers to have a complete inference model based on 3D distance fields. This discriminative model despite being efficient, it lacked the capability of capturing finer shape details, but still was a step toward using distance fields in discriminative 3D learning tasks.

To learn signed distance functions for shape representation, Park et al. [107] proposed DeepSDF a network architecture that learns to represent shapes as a continuous 3D field where a shape's boundary is encoded as the zero-level set of the learned function and the whole surrounding space is encoded as either positive or negative signed distance denoting wither a specific point is either outside or inside the shape's boundary. This generative model is capable of representing a whole class of shapes as opposed to classical SDFs that represent single shapes. The researchers investigated several architectures, a deep feedforward network for single shape representation, an autoencoder, and for the first time in the 3D learning literature, proposed an auto-decoder (encoder-less architecture). The proposed model produced exceptional results for shape completion tasks and latent shape interpolations with regard to its representational ability and surface quality. However, DeepSDF is best suited for shape completion tasks not for single-view reconstruction.

Mescheder et al. [108] proposed Occupancy Networks (OccNet), a network architecture that is capable of reconstructing 3D shapes from diverse input modalities in function space. The researchers formulated the reconstruction task as a binary classification problem where the shape's surface is implicitly represented as the continuous decision boundary of the classifier. To train the model for single-view reconstruction, the input images are fed into a pre-trained ResNet image encoder to produce an embedding of the input. The network also requires the input shapes to be sampled into points. The researchers used a uniform sampling technique to produce a number of points sampled from inside the bounding box of each input shape with small additional padding on the sides. A fully connected layer is fed the sampled points representing the shape to produce a feature vector for each point. Both the image embeddings and the points' features are passed to a number of fully connected ResNet blocks which use Conditional Batch Normalization (CBN) to condition the network on the image embedding. The output of the network is projected to a one-dimensional vector, and after applying the sigmoid function, the occupancy probabilities are obtained. For inference, the researchers proposed a Multi-resolution IsoSurface Extraction (MISE) algorithm that allows for the efficient extraction of high-resolution meshes from the predicted occupancy network.

A concurrent and closely related work was proposed by Chen and Zhang [109] where the researchers focused on introducing a generative model for implicit-field-based shape generation and reconstruction by building an implicit decoder that classifies whether a point is outside the shape or not using a simple structure of only MLPs and ReLU activations, essentially resembling the structure of a binary classifier. The implicit decoder is fed a concatenated input of both sampled points coordinates and encoded shape features and outputs the classification of a point with regard to the given shape. It is worthy to note that these output decisions can be used to extract a shape at any desired resolution regardless of the resolution of the shapes in the training set. The researchers evaluated the proposed implicit decoder by embedding it in several task-dependent networks: an autoencoder, a GAN for shape generation and interpolation, and a single-view reconstruction network. The researchers also proposed using the Light Field Descriptor (LFD) [110] as a visual similarity evaluation metric in addition to the commonly used CD and IoU metrics.

Instead of posing SDF prediction as a binary classification problem like in [108,109], Xu et al. [111] proposed to directly regress continuous SDFs values. To this end, the researchers proposed the Deep Implicit Surface Network (DISN), a network architecture that has individual modules for camera pose estimation, global feature extraction, local feature extraction, and for point location to feature space mapping. Global feature extraction is performed via a VGG-16 [71] image encoder. To extract local features, first camera pose has to be predicted using the camera pose estimation network, then each query point is projected into the image space using the predicted camera parameters, then through a process of locating and retrieving local features at each layer of the global feature map, multi-scale local features are extracted and concatenated. Points are also mapped to a higher-dimensional feature space and concatenated with both the local and global features. Each local/global feature vector is fed into a separate decoder, and the results are added together to finally regress the SDF value of each query point. Final explicit surface reconstruction is then achieved through identifying the isosurface and applying the Marching Cubes algorithm [99].

Michalkiewicz et al. [112] proposed to integrate Level Sets implicit representation into a learning framework in which individual layers in the neural network are used to output a level set of a continuous embedding function that implicitly represents a 3D surface. The proposed architecture contains two modules: an autoencoder for 3D shape generation and a CNN for image encoding. The two modules are connected through a 64-dimensional embedding space. The researchers formulated a variational loss function that is based on energy functionals that quantify both the differences between points and normals at these points in the predicted shape and their respective counterparts in the ground truth shape. They also proposed a number of shape priors such as surface area and volume as regularization terms to the loss function. To handle numerical instabilities, the researchers added another regularization term that promotes the unit gradient property similar to that of SDFs.

Genova et al. [113] explored the idea of representing 3D shapes as a collection of oriented 3D Gaussians. This representation (termed Structured Implicit Functions) provides several advantages over other implicit representations including the ability to be used in shape correspondence, interpolation, and segmentation applications. The researchers proposed a model that learns to fit a number of axis-aligned anisotropic 3D Gaussians (geometrically, can be viewed as a number of ellipsoids) to form a shape template that can approximate objects of varying geometry and topology. The shape template is acquired by fusing a number of depth maps that are then fed to an encoder-decoder network that directly regresses to the parameters of the shape elements that constitute the shape template approximating the original shape. Among other tasks, the researchers investigated the task of single-view 3D reconstruction using a distilled network that regresses from an input image to the shape elements parameters under the guidance of the original template-learning model. This shape representation was further extended [114] to include localized shape information in the form of local latent codes that are acquired by sampling points within the region of each shape element thus increasing its representational capacity to capture local details of shapes.

Similar to Genova et al. [113], Yamashita et al. [115] also proposed to use 3D Gaussians to analytically and implicitly represent

a 3D shape. However, Yamashita et al. proposed to use a Gaussian mixture distribution in 3D space instead of axis-aligned 3D Gaussians. To this end, the researchers proposed an image encoder that contains a series of convolution layers followed by a series of fully connected layers and a final layer that outputs the Gaussian mixture parameters. Additionally, the researchers incorporated a differentiable para-perspective projection module [116] that projects the 3D Gaussians into 2D Gaussians to be able to leverage an additional multi-view 2D loss in addition to a 3D loss based on KL divergence. The analytical properties of the Gaussian mixture representation can also be leveraged to solve the problems of relative pose estimation and multiple level-of-details reconstruction.

One of the few methods that extensively explored the effect of the object coordinate frame on the generalization abilities of 3D reconstruction models is the SDFNet proposed by Thai et. al [117]. The proposed method involves several properties that contribute to better generalization capabilities towards unseen objects, unseen classes, and even completely different datasets. The researchers argue that using a 3-DOF viewer centered coordinate frame is conducive to better generalizations. Additionally, they incorporate an intermediate 2.5D sketch estimator, where the shape features are encoded based on depth, normal, and silhouette information, then the SDF values are regressed in a similar fashion to [108]. A noteworthy observation made by the researchers is regarding the need to train on images with high visual variability, where images vary in lighting, reflectance, and backgrounds, in addition to pose variability offered by the 3-DOF viewer centered frame. All of this variability in pose and appearance contribute to the generalization capabilities of 3D reconstruction models.

4.2. Geometric primitives

The rather classical approach of approximating 3D shapes using geometric primitives has recently resurfaced, however, using modern data-driven and learning approaches. This line of work aims to learn shapes' semantic structures and/or generate compact representations and descriptions. This representation has different potential applications such as learning shape abstractions, learning part-level shape similarity and correspondences, and performing structure-aware shape manipulations.

Tulsiani et al. [118] proposed to use a CNN for learning shape abstractions from both volumetric shapes and from images. The main idea is to generate a number of cuboids that can efficiently and accurately approximate a shape's structure. The researchers experimented with representing the input shape using both a fixed and a variable number of primitives. The proposed model predicts for each primitive shape its probability of existence (for the variable number of primitives), its shape, and its rotation and translation. This prediction is governed by two loss functions: a coverage loss that forces the predicted shapes to completely cover the input shape, and a consistency loss which forces the predicted shapes to completely lie inside the input shape.

To learn a generative model for hierarchical shape structures, Li et al. [119] proposed Generative Recursive Autoencoder for Shape Structures (GRASS), a framework that can learn a topological, geometric, and hierarchical structure of 3D shapes in a single, generative, and fixed-dimensional representation. To this end, the researchers introduced a Recursive Neural Network (RvNN) that parses the layout hierarchy of shape parts (represented by oriented bounding boxes) in a bottom-up fashion and merges the children nodes to form parent a node moving up the hierarchy to finally produce a fixed-length code representing the whole structure. This parsing mechanism takes into consideration encoding both the symmetrical and adjacency properties of the parts' structures in relation to each other. To recover/generate shape structures, the researchers implemented a VAE-GAN architecture that combinedly learns not only reconstruction and interpolation of structures but also a plausible generation of novel ones. In addition to box structure synthesis, the researchers also introduced a network that learns voxelized part geometry synthesis from part codes.

Zou et al. [120] proposed a generative recurrent network, named 3D-PRNN, that is capable of reconstructing 3D shapes from single depth maps by approximating the shape using a number of cuboids. To obtain ground truth shape primitives required for training, the researchers proposed a method that takes 3D point clouds and fits a number of cuboids to them using Gaussian fields and energy minimization consecutively. Given an input depth map during inference, the proposed network encodes the input into a feature vector then it feeds the feature vector to a recurrent generator which is composed of a number of LSTM and Mixture Density Network (MDN) units. At each step, the network predicts a set of shape primitives where each primitive has shape, translation, and rotation attributes to be predicted. In addition, the network also predicts a binary signal denoting whether more shape primitives are needed to be predicted or not.

To recover shape structure from single RGB images, Niu et al. [121] proposed a framework, named Im2Struct, which has two separate networks: a structure masking network and a structure recovery network. The structure masking network is a two-scale CNN with the task of predicting a binary mask representing the shape's silhouette given an input of a single image. The features of the intermediate layers of the two scales of the network (full resolution scale and quarter resolution scale) are fused using jump connections, and the output of the smaller scale network is the binary mask prediction. The structure recovery network first fuses two sets of features: one coming from a VGG-16 image encoder encoding the input image, and the other is the last feature map produced by the structure masking network just before the mask prediction layer. The fused features are then fed to an RvNN decoder that was introduced in the GRASS framework [119] to finally predict the hierarchical box structure of the object depicted in the input image. Some highlighting is due concerning the internal work of this decoder. At each node in the recursive decoder, a decision has to be made regarding the status of each part code, whether it represents a leaf node, an adjacency relationship between two parts, or a symmetry. This decision is made utilizing a classifier and based on the classification the part code is fed into a corresponding adjacency/symmetry/box decoder. The adjacency decoder splits the part code into two child codes. The symmetry decoder recovers the symmetrical properties of a part. The leaf decoder recovers the box properties with regard to its dimensions and rotation.

4.3. Hybrid methods

Tang et al. [122] proposed an elaborate model that makes use of different representations in order to recover complex structures and topologies. The main intuition behind this setup is that each representation suffers from some drawbacks that can be mitigated by augmenting it with other representations. More specifically, the proposed setup makes use of point clouds, volumetric, and mesh representations in three different modules each is built on the output of the previous and refines it. Given an input image, the first stage aims at recovering the main topological structure of the shape in the form of skeletal points (skeletal curves and skeletal sheets). These skeletal points are then voxelized into a coarse volume and a high-resolution volume. The two volumes are then progressively refined under the guidance of the original image. Afterwards, a base mesh is extracted from the refined volume using Marching Cubes [99]. Finally, the base mesh is refined using a mesh deformation network that again uses the guidance of the

input image to add geometric details. The main strength of this setup is that the output mesh is capable of capturing complex topologies and thin structures and can go beyond genus-0 without the need for a shape template unlike most of the other approaches that output mesh-based reconstructions.

Deng et al. [123] proposed a novel hybrid 3D representation based on convex decomposition that is simultaneously implicit and explicit in order to inherit the advantages of both representations. The main intuition behind this new representation is that a shape can be represented implicitly during training and at inference time can be represented explicitly as a polygonal mesh directly without the need for further post-processing steps (for example iso-surface extraction). In the proposed representation, a generic non-convex shape is implicitly represented as a composition of convex polytopes and each convex polytope is constructed by a collection of learned hyperplanes parameters that specify its indicator function. On the other hand, a convex polytope can also be interpreted explicitly after applying a line to point duality transform on its hyperplanes, then computing a convex hull, then applying another duality transform and another convex hull computation to achieve the final corresponding polygonal mesh. The network architecture that the researchers proposed is capable of encoding both depth images and single RGB images, thus capable of performing single-view shape reconstruction with results on par with supervised methods in addition to its inherent shape decomposition capability.

In order to reconstruct low-poly compact meshes with sharp details, Chen et al. [124] proposed a hybrid representation that is based on Binary Space Partitioning (BSP). The main idea behind this representation is to dynamically learn the tree structure of a collection of binary partitions of space and their connectivity to produce convex polytopes that can eventually be merged to produce the final polygonal shape via Constructive Solid Geometry (CSG) operations. To achieve this, the researchers proposed a decoder that has three main modules. Given a shape/image features code, the first step is to extract hyperplane parameters, then for any given point, a signed distance vector can represent the location of the point in relation to each of the planes. Then, these hyperplanes are to be grouped into convex polytopes, which are finally grouped to form more complex and possibly non-convex shapes. The training is done in two stages: first with continuous but bounded weights then training is fine-tuned using discretized weights and an additional loss function that discourages overlapping convexes. An image encoder can be integrated to achieve single-view reconstruction using this representation. This method bears some similarity to CvxNet [123] in that both methods decompose a shape into convex polytopes. However, CvxNet has a fixed structure and number of hyperplanes and produces smoother meshes. On the other hand, BSP-Net learns the structure and tends to produce sharper more compact shapes.

Another approach that makes use of different shape representations is the one proposed by Poursaeed et al. [125]. This hybrid approach is built upon coupling the explicit AtlasNet [97] with the implicit OccupancyNet [108] through consistency losses. The main idea is to jointly train both networks and use the output of each network to enhance the output of the other through encouraging surface and normal consistencies between the outputs of each network. The proposed hybrid model has two branches with two image encoders, one for each branch. Given the output of AtlasNet, a cross-entropy-based consistency loss encourages the surface points of AtlasNet to align with the level set of the implicit function of OccupancyNet. Normal consistency is achieved by penalizing the misalignment between the direction of the normals of AtlasNet surface points and the gradient of the implicit function at the same points. An additional image-based loss is incorporated through the use of a neural renderer based on Soft Rasterizer [85]. The output of each network branch showed to be superior to the output of its original vanilla counterpart which proves the effectiveness of this hybrid training approach.

The methods that use implicit surfaces and geometric primitives vary in nature and their use cases may differ from the previously covered Euclidean and geometric methods. It is noteworthy that recovering the structure of shapes from images is a whole discipline on its own. Despite this, the previously covered methods are closely related to the topic of single-view 3D reconstruction and they overlap in their approach towards the problem of recovering shape properties from images. A summary of the discussed methods using implicit surface representation and geometric primitives is provided in Table 4.

5. Discussion

In the previous sections, we covered the methods based on the 3D representation used for their output, highlighting the details of each method. This section focuses more on what is common between these methods. Section 5.1 discusses the common 3D datasets used in training and validating these methods. Section 5.2 covers the common loss function and regularization terms. Section 5.3 covers common evaluation metrics used in the field. We finish this section with a discussion about the current challenges, trends, and possible future directions in Section 5.4.

5.1. Datasets

Training and evaluating data-driven learning-based models require a huge amount of data. Deep learning methods are data-hungry and their success depends on the availability of large amounts of high-quality training data. In the task of image-based 3D reconstruction and generation, the learning-based approaches require datasets that contain 3D models and their corresponding ground truth images. This is especially true in methods that adopt 3D supervision in their training paradigm. Table 5 provides a summary of the common 3D shape datasets used in single-view 3D reconstruction.

The main dataset used in training most of the discussed methods is ShapeNetCore [126]. ShapeNetCore is a subset of the ShapeNet dataset which does not seem to be publicly available up till the time of writing this paper. ShapeNetCore V1 has more than 51,000 3D models in 55 object categories. The 3D models are provided in the OBJ file format and their corresponding materials in MTL files. All the 3D models are object-centered, pre-aligned, and normalized to the unit cube. Renderings are not readily provided in this first version of ShapeNetCore, however, researchers usually relied on the renderings provided by Choy et al. [22] which provide renderings of all the models in ShapeNetCore V1 from 24 different views and follow the same categorization and naming conventions of the original dataset. The second version of ShapeNetCore provides an update to the first version with improved mesh and texture qualities. In addition, ShapeNetCore V2 has voxelized representations of the meshes and renderings from different views. However, the currently available dataset is incomplete as it is in its preliminary stage and what is provided at the time of writing is only a part of an upcoming release.

ModelNet [20] is another widely used dataset in 3D reconstruction and generation tasks. The full dataset which contains over 660 categories and almost 128,000 models is rarely used. Two subsets are however used: the ModelNet10 and ModelNet40 datasets, each has 3D models belonging to 10 and 40 categories respectively. The 3D models are provided as OFF (Object File Format) files. The ModelNet10 3D models come readily aligned, while the ModelNet40 models do not. An aligned version of ModelNet40 is provided by Sedaghat et al. [127]. The renderings of ModelNet40 are provided

Table 4
A summary of the methods that use implicit surface (first section), geometric primitives (second section), and hybrid (third section) representations. Method name links to official code (digital version). IoU: Intersection over Union, CD: Chamfer Distance, EMD: Earth Mover's Distance, MSE: Mean Squared Error, LFD: Light Field Descriptor.

Method	Year	3D Representation	Model Architecture	Loss Function	Dataset(s)	Evaluation Metric
https://github.com/autonomousvision/occupancy_networks OccNet [108]	2019	SDF	Image encoder - fully connected ResNet blocks	Cross entropy loss	ShapeNet	IoU CD
				Generative model (negative log likelihood + KL divergence)		Normal Consistency
https://github.com/czq142857/IM-NET IM-NET [109]	2019	SDF	ResNet image encoder - decoder (MLPs- ReLU activation)	Weighted MSE	ShapeNet	MSE CD IoU LFD
https://github.com/Xharlie/DISN DISN [111]	2019	SDF	Camera pose estimation (VGG image encoder - Two FC branches)	SDF (Weighted L1-norm)	ShapeNet	CD EMD IoU
			SDF prediction (VGG image encoder - two branches of 1x1 convs)	Camera pose (MSE)		
LevelSets [112]	2019	Level Sets	Image encoder (CNN)	Variational loss (points and normals) +	ShapeNet	IoU CD
			Shape generator (3D autoencoder)	surface area, volume, & unit gradient regularization		
https://github.com/google/ldif SIF [113]	2019	Oriented Gaussians	ResNet50 V2 + 2 FC layers	L2 (via network distillation)	ShapeNet	F-score
3D-GMNet [115]	2020	Gaussian Mixture	Image encoder (CNN + FC layers)	Gaussian mixture loss (KL divergence)	ShapeNet	CD EMD IoU
			para-perspective projection module	2D multi-view loss (L2)	Pix3D	
https://github.com/rehg-lab/3DShapeGen SDFNet [117]	2020	2.5D depth/normal maps (intermediate)	Depth, normal, silhouette estimation (U-ResNet18)	Depth, normal, silhouette estimation (MSE)	ABC	CD EMD
		SDF (output)	Shape feature encoding (ResNet18)	SDF estimation (L1)	ShapeNet	IoU F-score
			SDF estimation (MLP with CBN)			
https://github.com/shubhtuls/volumetricPrimitives Volumetric Primitives [118]	2017	Geometric Primitives (cuboids)	CNN	Coverage loss (L1)	ShapeNet	Qualitative
				Consistency loss (L2)		
https://github.com/junli-lj/grass GRASS [119]	2017	Geometric Primitives (OBB)	RvNN VAE-GAN	VAE (reconstruction loss - KL divergence)	ShapeNet	
				GAN (Standard generator - discriminator losses)	ModelNet	
https://github.com/zouchuhang/3D-PRNN 3D-PRNN [120]	2017	Geometric Primitives (cuboids)	Depth map encoder - recurrent generator (LSTMs - MDNs)	MDN loss (log likelihood)	ModelNet	IoU
				Rotation loss (MSE)		Surface-to-surface distance
https://github.com/chengjieniu/Im2Struct Im2Struct [121]	2018	Geometric Primitives	Structure Masking (Two-scale CNN connected by jump connections)	Structure Masking (SoftMax loss)	ShapeNet	Mask (Accuracy)
			Structure Recovery (VGG image encoder - RvNN decoder)	Structure Recovery (reconstruction loss - Cross entropy loss for classifier)		Reconstruction (Hausdorff Distance)
https://github.com/tangjiapeng/SkeletonBridgeRecon SkeletonBridgeRecon [122]	2019	Hybrid	Skeleton learning (ResNet18 image encoder two-stream MLPs)	Skeleton learning (CD Laplacian regularization)	ShapeNet	CD EMD
			Base mesh network (ResNet18 3D CNNs)	Mesh refinement (CD edge & normal regularizations)	ShapeNet-Skeleton	
			Mesh refinement network (VGG-16 image encoder graph CNN)			
https://github.com/tensorflow/graphics/tree/master/tensorflow_graphics/projects/cvxnet CvxNet [123]	2020	Hybrid	ResNet 18 image encoder MLP sequential decoder	Stochastic approximation loss + auxiliary losses	ShapeNet	IoU CD F-score
https://github.com/czq142857/BSP-NET-original BSP-Net [124]	2020	Hybrid	ResNet18 image encoder Dense MLP	Reconstruction loss (least squares) + overlap loss	ShapeNet	CD Edge-CD LFD
Hybrid-AN-ON [125]	2020	Hybrid	2 ResNet18 image encoders AtlasNet	Occupancy loss + Chamfer loss + Consistency loss (cross entropy) +	ShapeNet	CD
			OccupancyNet Neural renderer	normal consistency loss + image loss (L2)		Normal Consistency Score

Table 5
Common datasets used in single-view 3D reconstruction models.

Dataset	Image Type	No. of images	No. of categories	Total No. of models
Ikea Dataset [130]	Real	759	7	Total: 219 w/ matching images: 90
Pascal3D+ [131]	Real	30,899	12	79
ShapeNetCore [126]	Synthetic (rendered)	-	55	51,300
ModelNet [20]	Synthetic (rendered)	-	662	127,915
ObjectNet3D [129]	Real	90,127	100	44,147
Pix3D [132]	Real	10,069	9	395
ABC [133]	Synthetic (rendered)	-	NA	1,000,000+

by Wang et al. [128]. ModelNet has been used extensively for evaluation purposes to test the generalization capabilities of models trained with the ShapeNet dataset.

Another large-scale dataset that is used for testing and evaluation purposes is ObjectNet3D [129]. This dataset was compiled for 3D object recognition and pose estimation tasks. The ground truth images are natural images that contain more than one object depicted, and the 3D models are aligned with these objects in the images. The dataset contains more than 90,000 images depicting more than 200,000 objects.

It also contains more than 44,000 unique 3D models in 100 categories. Despite that ObjectNet3D is not tailored towards 3D reconstruction tasks, 2D images-3D models pairs can be used in evaluating 3D reconstruction models. However, the authors of ObjectNet3D advise against using ObjectNet3D for training such models since the 3D shapes are used to annotate both the training and test image sets which would result in data leakage that can produce biased results.

Similarly, other datasets that serve the same purpose as ObjectNet3D have been used in evaluating 3D reconstruction models. The Ikea [130] and Pascal3D+ [131] datasets were curated specifically for fine pose estimation and 3D detection/recognition. These datasets contain a limited number of 3D models. However, they also provide natural images for these 3D models. As with ObjectNet3D these datasets are suitable for testing the behavior of 3D reconstruction models with natural images rather than the synthetic images that have been used in training such models. Researchers have used these datasets to evaluate the generalizability of their trained models to unseen classes and/or cluttered natural images as inputs.

Sun et al. [132] released Pix3D, a dataset that is specifically built for image-based 3D reconstruction tasks. The researchers had in mind the limitations of the previously discussed datasets. Pascal3D+ and ObjectNet3D have natural images but the alignment between the 2D-3D pairs is rough and the 3D models do not 100% match the image objects. ModelNet and ShapeNet have a large number and diversity of 3D models but do not have ground truth natural images. Ikea dataset has natural images and precise 2D-3D alignment, but lack in number and diversity of 3D models. In Pix3D the researchers used a large number of natural images and aligned exactly matching 3D models to these images with high precision. The Pix3D dataset comes with rich information about each image-shape pair: 2D and 3D keypoints, voxel representation, image mask, rendering camera intrinsic and extrinsic parameters, and flags stating whether a specific object in an image is occluded, slightly occluded, or truncated.

A database named ABC (A Big CAD) [133] was recently released specifically for geometric deep learning tasks including shape reconstruction. It is a huge collection of over one million CAD (Computer-Aided Design) models explicitly parameterized as curves and surfaces in B-rep (Boundary representation). While this representation is not readily suitable for deep learning models, it allows for generating 3D models in mesh formats at varying resolutions through a sampling process. The authors of the ABC dataset also provide the 3D models as OBJ and STL files, renderings of these 3D models from canonical viewpoints, and several features and statistical information.

5.2. Loss functions

In the task of shape reconstruction from images, the learning models are usually trained to minimize the difference between the ground truth shape and the predicted shape. This is done through the use of different cost or loss functions that must be differentiable. The choice of a loss function depends on the type of 3D shape representation used and the training paradigm. As discussed above, training shape reconstruction models can be done using either 3D supervision only, 2D supervision only, or a combination of both 2D and 3D supervision.

5.2.1. 3D Loss functions

When 3D supervision is used, the loss function directly measures the difference between two 3D shapes: the predicted and the ground truth shapes. This can be done using an appropriate distance metric. In Euclidean representations such as when using voxels, the volumetric loss or reconstruction loss can be calculated using Euclidean distance measure as L2 distance. In probabilistic predictions of occupancy of voxel grids, the Binary Cross Entropy loss is often used which is defined as:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(p(y_i)\right) + (1-y_i)\log\left(1-p(y_i)\right), \tag{1}$$

where y_i is the ground truth occupancy of voxel i, $p(y_i)$ is the predicted occupancy probability, and N is the total number of voxels. Note that only the sum can be used instead of the mean as in [22].

In geometric representations, such as point cloud representations, the distance between the ground truth and predicted shapes can be measured by Chamfer distance, which is defined as [65]:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x-y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x-y\|_2^2, \tag{2}$$

where S_1, S_2 are the two point cloud sets representing the predicted and ground truth shapes, x and y are two corresponding points belonging to S_1 and S_2 respectively. The correspondence is based on the nearest neighbor search.

Alternatively, the distance between two point cloud sets can be measured by the Earth Mover Distance, which is defined as [65]:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2, \tag{3}$$

where S_1, S_2 are the two point cloud sets of equal size ($|S_1| = |S_2|$) representing the predicted and ground truth shapes, and $\phi : S_1 \to S_2$ is a bijection. This one-to-one correspondence between points

is based on an optimal assignment. In practice, the computation of EMD is relaxed through an approximation algorithm.

When 3D mesh representation is used, CD can be applied to measure the distance between the vertices of predicted and ground truth meshes. Another technique used is sampling points from both the ground truth mesh and the predicted mesh, then calculate the distance using CD on the sampled points. However, using CD only in computing the loss function is not sufficient since mesh connectivity and neighborhood information are not taken into consideration. Researchers proposed adding more regularization terms to the loss function to take advantage of the readily available connectivity, face, and edge information.

A normal loss can be incorporated in the loss function, which aims at enforcing the consistency of surface normal. This is achieved by enforcing the orthogonality between each edge formed between a vertex in the predicted mesh and its neighboring vertices and the normal at the closest vertex in the ground truth mesh. Normal loss is defined as [103]:

$$\mathcal{L}_n = \sum_p \sum_{q=\arg\min_q \left(\|p-q\|_2^2\right)} \|\langle p-k, \mathbf{n}_q\rangle\|_2^2, \text{ s.t. } k \in \mathcal{N}(p), \tag{4}$$

where p is a vertex in the predicted mesh, q is the closest vertex to p in the ground truth mesh, $N(p)$ is the set of p's neighboring vertices, and k is a vertex that belongs to $N(p)$, n_q is the normal at q, and $\langle\cdot,\cdot\rangle$ is the inner product of two vectors. Note that this loss term does not reach zero unless on a planar surface, but optimizing it leads to a more consistent surface normal.

Another regularization term that can be added to the loss function is Laplacian regularization [103], which aims at giving a smoother predicted mesh through restraining the movement of its vertices during deformation. Laplacian regularization is applied between the mesh before deformation and the mesh after applying the deformation in each deformation block. First, graph Laplacian is computed on each mesh to calculate the average difference between each vertex and its neighbors:

$$\delta_p = p - \sum_{k\in\mathcal{N}(p)} \frac{1}{\|\mathcal{N}(p)\|} k. \tag{5}$$

Then, the Laplacian loss can be calculated through:

$$\mathcal{L}_{lap} = \sum_p \left\|\delta'_p - \delta_p\right\|_2^2, \tag{6}$$

where δ'_p and δ_p are the Laplacian coordinate of a vertex after and before a deformation block.

Additionally, an edge length regularization [103] can also be incorporated in the loss function which restricts the length of edges thus reduces flying vertices. Edge regularization can be computed through:

$$\mathcal{L}_{edge} = \sum_p \sum_{k\in\mathcal{N}(p)} \|p-k\|_2^2. \tag{7}$$

5.2.2. 2D Loss functions

When learning models are trained under 2D supervision and the 3D ground truth is not available, the loss functions become a distance metric within a reprojection loss. The distance measure is done between silhouette masks, normal maps, and/or depth maps of the input image and the projected image that the model learns. The reprojection loss depends on the projection operator embedded in the model's architecture. The projection operators range from basic non-learnable orthographic/perspective projection modules and differentiable renderers to learnable projection operators that learn camera parameters.

An example of a projection loss that uses silhouette loss is provided in PTN [45]. The loss is defined as:

$$\begin{aligned}\mathcal{L}_{proj}\left(I^{(k)}\right) &= \sum_{j=1}^{n} \mathcal{L}_{proj}^{(j)}\left(I^{(k)}; S^{(j)}, \alpha^{(j)}\right) \\ &= \frac{1}{n}\sum_{j=1}^{n} \left\|P\left(f\left(I^{(k)}\right); \alpha^{(j)}\right) - S^{(j)}\right\|_2^2,\end{aligned}$$

where $f(I^{(k)})$ is the learned volume from image k, j is the index of output 2D silhouettes, $\alpha^{(j)}$ is the camera viewpoint corresponding to j, $P(\cdot)$ is a 3D-2D projection function, $S^{(j)}$ is the j^{th} ground truth silhouette. Here L2 distance is used as the distance measure.

Another distance metric that can also be used with silhouette-based losses is the negative IoU as in [81]. Negative IoU is defined as:

$$\mathcal{L}_{sl}(x \mid \phi_i, s_i) = -\frac{\left|\hat{s}_i \odot s_i\right|_1}{\left|\hat{s}_i + s_i - \hat{s}_i \odot s_i\right|_1}, \tag{8}$$

where s_i is ground truth silhouette, $\hat{s}_i$ is reconstructed silhouette, and $\odot$ is an element-wise product.

5.3. Evaluation metrics

The role of evaluation metrics is to assess the performance of the learning models and to compare different reconstruction models. The most commonly used evaluation metric in 3D reconstruction tasks is the Intersection over Union metric. This is especially true in volumetric approaches that use voxels as shape representation. The IoU is defined as [22]:

$$IoU = \frac{\sum_{i,j,k}\left[I\left(p_{(i,j,k)} > t\right) I\left(y_{(i,j,k)}\right)\right]}{\sum_{i,j,k}\left[I\left(I\left(p_{(i,j,k)} > t\right) + I\left(y_{(i,j,k)}\right)\right)\right]}, \tag{9}$$

where t is a voxelization threshold, $I(\cdot)$ is an indicator function, $y_{(i,j,k)}$ is the ground truth voxel occupancy, and $p_{(i,j,k)}$ is the predicted voxel occupancy.

For 3D meshes and point clouds, both CD and EMD can be used for evaluation purposes in the same way they are used as distance functions in calculating loss functions (see Section 5.2.1). Similarly, Hausdorff distance [134] can also be used in evaluating mesh and point clouds, however, it differs from CD and EMD in that it does not require a point-to-point correspondence while still being able to measure the similarity between the two shapes. Hausdorff distance between two surfaces $\mathcal{S}$ and $\mathcal{S}'$ is computed as follows: First, we compute a point to surface distance between point p and surface $\mathcal{S}'$, where p' is point in $\mathcal{S}'$: $d\left(p, \mathcal{S}'\right) = \min_{p'\in\mathcal{S}'} \left\|p - p'\right\|_2$, then the distance between surfaces $\mathcal{S}$ and $\mathcal{S}'$ becomes: $d\left(\mathcal{S}, \mathcal{S}'\right) = \max_{p\in\mathcal{S}} d\left(p, \mathcal{S}'\right)$. However, this distance is generally not symmetrical. The symmetrical Hausdorff distance can then be defined as:

$$d_{symHausdorff}\left(\mathcal{S}, \mathcal{S}'\right) = \max\left[d\left(\mathcal{S}, \mathcal{S}'\right), d\left(\mathcal{S}', \mathcal{S}\right)\right]. \tag{10}$$

Light Field Descriptor (LFD) [110] has been proposed to evaluate the visual similarity between 3D shapes. A light field is a function that represents the radiance at a given 3D point along a given direction. To obtain LFD for a 3D shape, a number of renderings of this shape are acquired from different angles (usually the rendering camera is positioned at the vertices of a dodecahedron surrounding the 3D shape). Then from these silhouette renderings, two shape descriptors are used in combination. Zernike Moment Descriptors (ZMD) are used as region-based descriptors and Fourier Descriptors (FD) are used as boundary-based descriptors. It is argued that using LFD as an evaluation metric on reconstructed shapes can give a better intuition regarding visual similarity since it is inspired by the human vision [109].

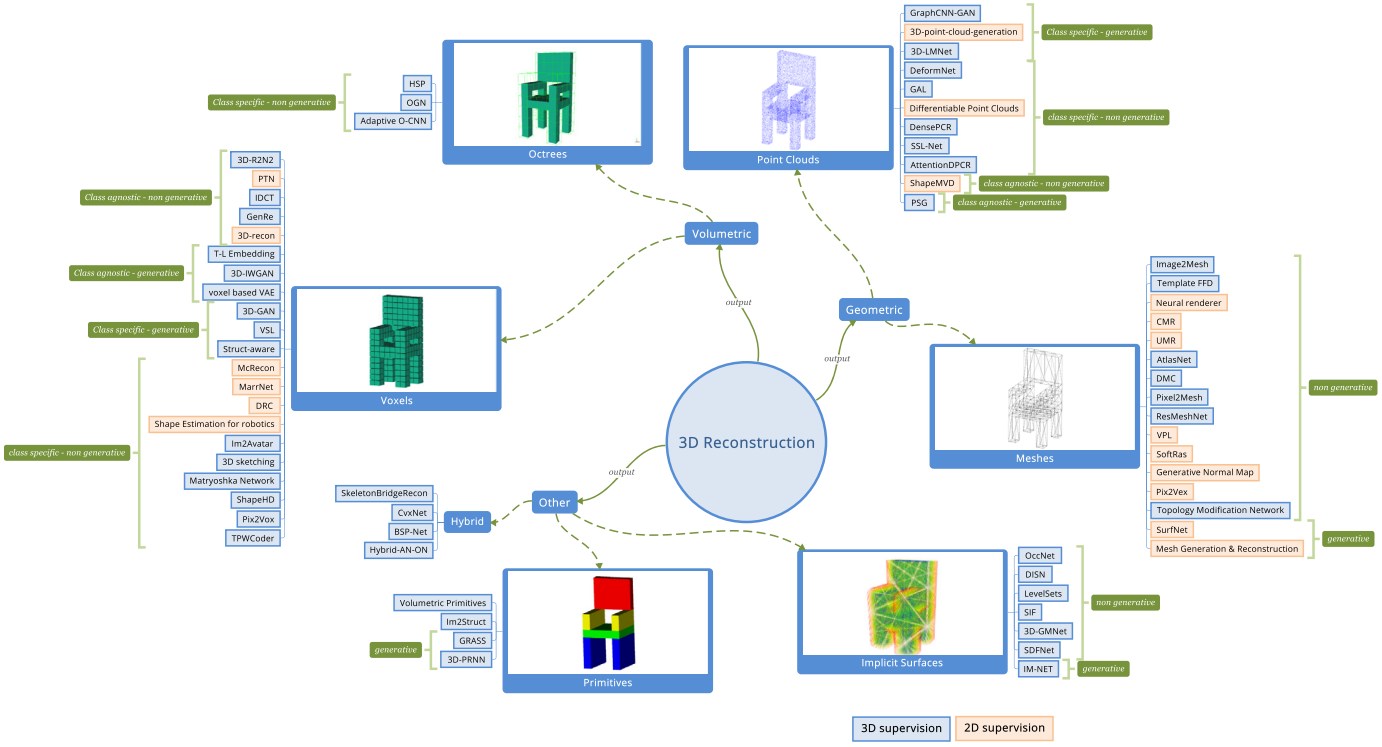

Fig. 8. Visual Summary of selected methods categorized according to output representation and showing the supervision paradigm, being generative vs. non-generative, and being class-specific vs. class-agnostic.

The F1 score or F-score has been proposed as a more robust indicator of the quality of reconstruction [135]. The F-score works on points and calculates the harmonic mean between precision and recall. In this case, the precision quantifies the accuracy of the reconstruction and the recall quantifies the completeness of the reconstruction. To compute the recall term, we compute first a point to surface distance between point p and surface $\mathcal{S}'$, where p' is point in the reconstructed $\mathcal{S}'$: $d(p, \mathcal{S}') = \min_{p' \in \mathcal{S}'} \left\| p - p' \right\|_2$, and the recall value becomes:

$$Recall_r = \frac{100}{|S|} \sum_{p \in S} \left[d(p, \mathcal{S}') < r \right], \tag{11}$$

where r is the threshold radius from a point that defines a hit. To compute the precision term, the same calculation is repeated in the opposite direction, in other words, we compute a point to surface distance between point p' and the ground truth surface $\mathcal{S}$: $d(p', \mathcal{S}) = \min_{p \in S} \left\| p' - p \right\|_2$, and the precision value becomes:

$$Precision_r = \frac{100}{|\mathcal{S}'|} \sum_{p' \in \mathcal{S}'} \left[d(p', \mathcal{S}) < r \right]. \tag{12}$$

The final F-score is calculated as follows:

$$Fscore_r = 2 \cdot \frac{Precision_r \cdot Recall_r}{Precision_r + Recall_r}. \tag{13}$$

5.4. Current trends and future directions

The discussed methods above show the significant progress that the field of single-view 3D reconstruction has achieved. Fig. 8 presents a visual summary of these methods and for each method, the figure shows its category according to its output, whether it uses 3D or 2D supervision, whether it is generative or non-generative in nature, and whether it is considered class-specific or class-agnostic.

Despite the significant progress, the variety of approaches, and the nuances in how these methods handle the problem of single-view 3D reconstruction, it is reported that a simple image classification and nearest neighbor shape retrieval method can outperform the state-of-the-art methods of single-view 3D reconstruction [135]. While this may not be necessarily true and applicable to all methods, it is always a good idea to analyze the current research trends and pinpoint those areas that need more experimentation where there is room for improvement to advance the state of research.

The choice of representation is clearly crucial to the quality of shape reconstructions. While volume-based methods comprise most of the work done in single-view 3D reconstruction, these methods lack the scalability needed to reconstruct high-resolution and detailed shapes due to its memory consumption constraints. Point-cloud-based methods have less memory footprint, but points lack connectivity information, and they require post-processing to acquire a shape from these points. Mesh-based methods utilize connectivity information but hugely depend on the base model that they deform. Generally speaking, there is no straightforward method to change the topology of the base mesh during reconstruction to achieve better edge and mesh flows which are conducive to a better shape quality. Additionally, going beyond genus-0 mesh reconstruction is still a matter that needs to be tackled in future research. Implicit-surface-based methods usually result in overly smoothed reconstructions due to the sampling method inefficiency. A solution to this problem has been very recently proposed by Tancik et al. [136] by incorporating Fourier feature mapping to control the frequencies that an MLP can learn which results in increasing the network's capability to learn high frequency details of 3D shapes.

Implicit-surface-based methods have been gaining traction recently. This is because of their desirable properties in the context of single-view 3D reconstruction. However, isosurface extraction is

Table 6
A summary of the properties of implicit vs. different explicit representations in single-view shape reconstruction context.

Property	Euclidean		Geormetric		Implicit
	Voxel	Octree	Point Cloud	Mesh	
Regularity	Yes	Yes	No	No	Yes
Suited for deep learning	Yes	No	No	No	Yes
Compact representation	No	Yes	Yes	Yes	Yes
Supports topological changes	Yes	Yes	No	No	Yes
Geometric information	No	No	Yes	Yes	Yes
Supports deformation	No	No	Yes	Yes	Yes
Surface information	No	Yes	Yes	Yes	Yes
Ready for applications	No	No	No	Yes	No
Shapes correspondence	No	No	Yes	Yes	No

still a huge drawback because it is computationally expensive and hinders the integration of implicit surfaces into other applications that consume 3D shapes. Also, implicit surfaces methods lacked the ability to learn shape correspondences, but solutions have also been recently proposed to this issue by using 3D Gaussians [113] and Gaussian mixture [115] based techniques. Another recent trend gaining traction is the hybrid approach towards single-view 3D reconstruction which tries to make the best of both the implicit and explicit representations (Section 4.3). The properties of implicit and explicit surfaces in relation to single-view reconstruction are summarized in Table 6. It is worth noting that new implicit representations are being continuously proposed. One of the most promising recent representations is the Neural Radiance Fields [137]. One has yet to see their adoption in single-view 3D shape reconstruction pipelines.

A successful data-driven 3D shape reconstruction model presumably needs to learn low-level image cues, structural knowledge, and high-level shape understanding, then combine these features in its inference workflow. Learning image features has reached an unprecedented level of robustness through architectures like VGG [71] and ResNet [83], that is why we see a lot of VGG-based and ResNet-based image encoders incorporated in 3D shape reconstruction models as, for example, in [82,103].

Incorporating shape structure within the scope of shape reconstruction has also been tackled in [44,100,121] and within the scope of shape generation in general in [138,139]. However, there is room for improvement and this is an important future research direction. Integrating shape structure will result in a more plausible shape generation. This has to be integrated with learning both global shape features and local shape features to be able to avoid reconstruction artifacts such as holes and the inability to reconstruct thin shapes (e.g. thin legs of a chair).

There is also a pressing need to create models that are genuinely generalizable and/or able to scale to novel shape categories using natural images captured in-the-wild. Some of the previously discussed methods are considered category-specific in which the learned model is trained on one shape category only. This decision is usually made to be able to leverage prior knowledge pertaining to a specific object category. However, there is a chance that the model memorizes the training shapes since there is a small intra-class variation between the shapes in the training set. Additionally, these models may not be able to handle out-of-class reconstructions. On the other hand, some category-agnostic models learn from many object classes. While in theory these models are supposed to generalize gracefully to unseen categories, there is still no straight-forward way to incorporate shape structure and high-level shape understanding in the learning process. In both cases, whether the model is class-specific or class-agnostic, it is required to be capable of reconstructing the 3D shape out of natural images leveraging its generalization capability.

The issue of using an object-centered versus a viewer-centered coordinate frame is also related to the generalizability of 3D reconstruction models. This issue has been studied [140] and validated in [135]. It is reported that a viewer-centered coordinate frame is better for reconstructing novel shapes, while an object-centered coordinate frame performs better with familiar shapes. This interesting discovery requires more investigation since almost all of the methods depend on the object-centered coordinate system during training as the prediction problem is simplified when all training data is aligned to a canonical pose. However, since generalizability is a desired quality in reconstruction models, more research needs to experiment to leverage both the generalizability of the viewer-centered coordinate system and the better performance of the object-centered coordinate system.

A major factor contributing to the success of data-driven methods is the availability of large amounts of quality training data. Due to this demand and because the research in 3D and geometric deep learning is growing, the available 3D shape datasets are getting bigger and new datasets are being introduced. However, Pix3D [132] is the only dataset that is created with shape reconstruction in mind. The lack of reconstruction-focused 3D datasets has been partially addressed by relying on other suitable 3D shape datasets, by using 2D supervision mechanisms, and by applying data augmentation techniques. However, there exists a need for bigger datasets with natural images along with 3D annotations. An optimal 3D shape dataset for shape reconstruction may contain a large number of shapes, a balanced number of shapes in each represented category, and in each category, it is desirable to have a degree of intra-class variations.

With regard to 2D supervision as an alternative to full 3D supervision, it has achieved interesting results and has relaxed the training process. However, this training paradigm inherently relies on visual hull reconstruction-from-silhouettes and space carving techniques, so the emphasis is basically on the boundary information, and dependent on the number of views used in training, which might not be enough to recover the whole 3D shape faithfully. One solution to mitigate shape-camera ambiguities has been recently proposed in [90] as discussed above. Additionally, with differentiable rendering techniques, realistic renderings, and domain adaptation techniques, 2D supervision can achieve superb results while benefiting from the relaxed training requirements.

One last issue that requires further work is seeking a more representative shape quality metrics for quantitative evaluation purposes. While there is already a number of shape evaluation metrics, they suffer from some drawbacks that make them fall short of being optimum for shape reconstruction evaluation and benchmarking. CD metric is sensitive to outliers, EMD is computationally intensive. IoU is not an accurate indication of shape discrepancies below a mid-range value [135]. Thus, F-score has been proposed as a more indicative evaluation metric. Still, there is room for coming

up with a measure that can better capture topological and visual discrepancies between shapes. A good candidate can come from the computer graphics field just like LFD and Hausdorff measures.

Finally, a noteworthy step taken by the researchers in the field of 3D deep learning is the introduction of libraries that can potentially facilitate and accelerate the state of research by providing tools for 3D data loading, conversion between representations, common evaluation metrics, and even readily available models and architectures. All of these features can help with standardizing the workflow and benchmarking tasks, thus pushing the field forward. Two notable examples of such libraries are Nvidia's Kaolin [141] and Facebook research's PyTorch3D [142].

6. Conclusion

In this survey paper, we focused on the advances in single-view shape 3D reconstruction using data-driven deep learning methods. Methods are classified based on the 3D representation used as an output. For each method, we discussed its major contributions, degree of supervision, and highlighted its training paradigm. Additionally, we discussed the 3D shape datasets, loss functions, and evaluation metrics currently used in research. We also reflected on the current trends, challenges, and possible future directions to comprehensively cover the topic of data-driven single-view 3D reconstruction and provide a thorough introduction to those who want to delve into this fast-growing field of research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Druzhkov P, Kustikova V. A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognit Image Anal 2016;26(1):9–15.
[2] Zhao Z-Q, Zheng P, Xu S-t, Wu X. Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 2019.
[3] Zhao B, Feng J, Wu X, Yan S. A survey on deep learning-based fine–grained object classification and semantic segmentation. Int J Autom Comput 2017;14(2):119–35.
[4] Mousavian A, Anguelov D, Flynn J, Kosecka J. 3D bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 7074–82.
[5] Dai A, Ruizhongtai Qi C, Nießner M. Shape completion using 3D-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 5868–77.
[6] Hegde V, Zadeh R. Fusionnet: 3D object classification using multiple data representations 2016 arXiv preprint arXiv:160705695.
[7] Grabner A, Roth PM, Lepetit V. 3D pose estimation and 3D model retrieval for objects in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 3022–31.
[8] Schonberger JL, Frahm J-M. Structure-from-motion revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 4104–13.
[9] Seitz SM, Curless B, Diebel J, Scharstein D, Szeliski R. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1. IEEE; 2006. p. 519–28.
[10] Zhang L, Dugas-Phocion G, Samson J-S, Seitz SM. Single-view modelling of free-form scenes. The Journal of Visualization and Computer Animation 2002;13(4):225–35.
[11] Prasad M, Fitzgibbon A. Single view reconstruction of curved surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2. IEEE; 2006. p. 1345–54.
[12] Shi B, Bai S, Zhou Z, Bai X. Deeppano: deep panoramic representation for 3-D shape recognition. IEEE Signal Process Lett 2015;22(12):2339–43.
[13] Sinha A, Bai J, Ramani K. Deep learning 3D shape surfaces using geometry images. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer; 2016. p. 223–40.
[14] Tatarchenko M, Dosovitskiy A, Brox T. Multi-view 3D models from single images with a convolutional network. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer; 2016. p. 322–37.
[15] Arsalan Soltani A, Huang H, Wu J, Kulkarni TD, Tenenbaum JB. Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 1511–19.
[16] Chen W, Fu Z, Yang D, Deng J. Single-image depth perception in the wild. In: Advances in Neural Information Processing Systems; 2016. p. 730–8.
[17] Ahmed E, Saint A, Shabayek AER, Cherenkova K, Das R, Gusev G, et al. Deep learning advances on different 3D data representations: a survey 2018 arXiv preprint arXiv:180801462.
[18] Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I. Deep learning advances in computer vision with 3D data: a survey. ACM Computing Surveys (CSUR) 2017;50(2):20.
[19] Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: going beyond euclidean data. IEEE Signal Process Mag 2017;34(4):18–42.
[20] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, et al. 3D ShapeNets: A deep representation for volumetric shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 1912–20.
[21] Maturana D, Scherer S. VoxNet: A 3D convolutional neural network for real–time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2015. p. 922–8.
[22] Choy CB, Xu D, Gwak J, Chen K, Savarese S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer; 2016. p. 628–44.
[23] Kar A, Tulsiani S, Carreira J, Malik J. Category-specific object reconstruction from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 1966–74.
[24] Sun Y, Liu Z, Wang Y, Sarma SE. Im2avatar: colorful 3D reconstruction from a single image 2018 arXiv preprint arXiv:180406375.
[25] Tulsiani S, Zhou T, Efros AA, Malik J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 2626–34.
[26] Delanoy J, Aubry M, Isola P, Efros AA, Bousseau A. 3D Sketching using multi-view deep volumetric prediction. Proceedings of the ACM on Computer Graphics and Interactive Techniques 2018;1(1):21.
[27] Richter SR, Roth S. Matryoshka networks: Predicting 3D geometry via nested shape layers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 1936–44.
[28] Wu J, Zhang C, Zhang X, Zhang Z, Freeman WT, Tenenbaum JB. Learning shape priors for single-view 3D completion and reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 646–62.
[29] Wu J, Wang Y, Xue T, Sun X, Freeman B, Tenenbaum J. MarrNet: 3D shape reconstruction via 2.5D sketches. In: Advances in Neural Information Processing Systems; 2017. p. 540–50.
[30] Wu J, Zhang C, Xue T, Freeman B, Tenenbaum J. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in Neural Information Processing Systems; 2016. p. 82–90.
[31] Zhang X, Zhang Z, Zhang C, Tenenbaum J, Freeman B, Wu J. Learning to reconstruct shapes from unseen classes. In: Advances in Neural Information Processing Systems; 2018. p. 2263–74.
[32] Xie H, Yao H, Sun X, Zhou S, Zhang S. Pix2Vox: Context-aware 3D reconstruction from single and multi-view images. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2019. p. 2690–8.
[33] Chen Q, Nguyen V, Han F, Kiveris R, Tu Z. Topology-aware single-image 3D shape reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 270–1.
[34] Girdhar R, Fouhey DF, Rodriguez M, Gupta A. Learning a predictable and generative vector representation for objects. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer; 2016. p. 484–99.
[35] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems; 2012. p. 1097–105.
[36] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Ieee; 2009. p. 248–55.
[37] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in Neural Information Processing Systems; 2014. p. 2672–80.
[38] Kingma DP, Welling M. Auto-encoding variational bayes 2013 arXiv preprint arXiv:13126114.
[39] Liu J, Yu F, Funkhouser T. Interactive 3D modeling with a generative adversarial network. In: 2017 International Conference on 3D Vision (3DV). IEEE; 2017. p. 126–34.
[40] Smith E, Meger D. Improved adversarial systems for 3D object generation and reconstruction 2017 arXiv preprint arXiv:170709557.
[41] Zhu J, Xie J, Fang Y. Learning adversarial 3D model generation with 2D image enhancer. In: Thirty-Second AAAI Conference on Artificial Intelligence; 2018. p. 7615–22.
[42] Brock A, Lim T, Ritchie JM, Weston N. Generative and discriminative voxel modeling with convolutional neural networks 2016 arXiv preprint arXiv:160804236.
[43] Liu S, Giles L, Ororbia A. Learning a hierarchical latent-variable model of 3D shapes. In: 2018 International Conference on 3D Vision (3DV). IEEE; 2018. p. 542–51.

[44] Balashova E, Singh V, Wang J, Teixeira B, Chen T, Funkhouser T. Structure-aware shape synthesis. In: 2018 International Conference on 3D Vision (3DV). IEEE; 2018. p. 140–9.
[45] Yan X, Yang J, Yumer E, Guo Y, Lee H. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In: Advances in Neural Information Processing Systems; 2016. p. 1696–704.
[46] Gwak J, Choy CB, Chandraker M, Garg A, Savarese S. Weakly supervised 3D reconstruction with adversarial constraint. In: 2017 International Conference on 3D Vision (3DV). IEEE; 2017. p. 263–72.
[47] Johnston A, Garg R, Carneiro G, Reid I, van den Hengel A. Scaling cnns for high resolution volumetric reconstruction from a single image. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017. p. 939–48.
[48] Yang G, Cui Y, Belongie S, Hariharan B. Learning single-view 3D reconstruction with limited pose supervision. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 86–101.
[49] Mees O, Tatarchenko M, Brox T, Burgard W. Self-supervised 3D shape and viewpoint estimation from single images for robotics. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2019. p. 6083–9. doi:10.1109/IROS40897.2019.8967916.
[50] Meagher D. Geometric modeling using octree encoding. Computer graphics and image processing 1982;19(2):129–47.
[51] Riegler G, Osman Ulusoy A, Geiger A. OctNet: Learning deep 3D representations at high resolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 3577–86.
[52] Wang P-S, Liu Y, Guo Y-X, Sun C-Y, Tong X. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. ACM Transactions on Graphics (TOG) 2017;36(4):72.
[53] Häne C, Tulsiani S, Malik J. Hierarchical surface prediction for 3D object reconstruction. In: 2017 International Conference on 3D Vision (3DV). IEEE; 2017. p. 412–20.
[54] Tatarchenko M, Dosovitskiy A, Brox T. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017. p. 2088–96.
[55] Wang P-S, Sun C-Y, Liu Y, Tong X. Adaptive O-CNN: a patch-based deep representation of 3D shapes. In: SIGGRAPH Asia 2018 Technical Papers. ACM; 2018. p. 217.
[56] Qi CR, Su H, Mo K, Guibas LJ. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 652–60.
[57] Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems; 2017. p. 5099–108.
[58] Nash C, Williams CK. The shape variational autoencoder: a deep generative model of part-segmented 3D objects. Comput Graphics Forum 2017;36(5):1–12.
[59] Yang Y, Feng C, Shen Y, Tian D. FoldingNet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 206–15.
[60] Achlioptas P, Diamanti O, Mitliagkas I, Guibas L. Learning representations and generative models for 3D point clouds. In: Proceedings of the 35th International Conference on Machine Learning; 2018. p. 40–9.
[61] Li C-L, Zaheer M, Zhang Y, Poczos B, Salakhutdinov R. Point cloud GAN 2018 arXiv preprint arXiv:181005795.
[62] Sun Y, Wang Y, Liu Z, Siegel JE, Sarma SE. PointGrow: Autoregressively learned point cloud generation with self-attention. In: Winter Conference on Applications of Computer Vision; 2020.
[63] Valsesia D, Fracastoro G, Magli E. Learning localized generative models for 3D point clouds via graph convolution. In: International Conference on Learning Representations (ICLR); 2019 https://openreview.net/forum?id=SJeXSo09FQ.
[64] Simonovsky M, Komodakis N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 3693–702.
[65] Fan H, Su H, Guibas LJ. A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 605–13.
[66] Lun Z, Gadelha M, Kalogerakis E, Maji S, Wang R. 3D shape reconstruction from sketches via multi-view convolutional networks. In: 2017 International Conference on 3D Vision (3DV). IEEE; 2017. p. 67–77.
[67] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). In: LNCS, 9351. Springer; 2015. p. 234–41. (available on arXiv: 1505.04597[cs.CV])
[68] Kurenkov A, Ji J, Garg A, Mehta V, Gwak J, Choy C, et al. DeformNet: Free-form deformation network for 3D shape reconstruction from a single image. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE; 2018. p. 858–66.
[69] Mandikal P, Navaneet KL, Agarwal M, Babu RV. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. In: Proceedings of the British Machine Vision Conference (BMVC); 2018.
[70] Jiang L, Shi S, Qi X, Jia J. GAL: Geometric adversarial loss for single-view 3D-object reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 802–16.
[71] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition 2014 arXiv preprint arXiv:14091556.
[72] Mao X, Li Q, Xie H, Lau RY, Wang Z, Paul Smolley S. Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017. p. 2794–802.
[73] Lin C-H, Kong C, Lucey S. Learning efficient point cloud generation for dense 3D object reconstruction. In: Thirty-Second AAAI Conference on Artificial Intelligence; 2018. p. 7114–21.
[74] Insafutdinov E, Dosovitskiy A. Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in Neural Information Processing Systems; 2018. p. 2802–12.
[75] Mandikal P, Radhakrishnan VB. Dense 3D point cloud reconstruction using a deep pyramid network. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE; 2019. p. 1052–60.
[76] Sun R, Gao Y, Fang Z, Wang A, Zhong C. SSL-Net: Point-cloud generation network with self-supervised learning. IEEE Access 2019.
[77] Lu Q, Xiao M, Lu Y, Yuan X, Yu Y. Attention-based dense point cloud reconstruction from a single image. IEEE Access 2019;7:137420–31.
[78] Bregler C, Hertzmann A, Biermann H. Recovering non-rigid 3D shape from image streams. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2. Citeseer; 2000. p. 690–6.
[79] Tulsiani S, Kar A, Carreira J, Malik J. Learning category-specific deformable 3D models for object reconstruction. IEEE Trans Pattern Anal Mach Intell 2016;39(4):719–31.
[80] Sinha A, Unmesh A, Huang Q, Ramani K. SurfNet: Generating 3D shape surfaces using deep residual networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 6040–9.
[81] Kato H, Ushiku Y, Harada T. Neural 3D mesh renderer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 3907–16.
[82] Kanazawa A, Tulsiani S, Efros AA, Malik J. Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 371–86.
[83] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 770–8.
[84] Henderson P, Ferrari V. Learning to generate and reconstruct 3D meshes with only 2D supervision. In: Proceedings of the British Machine Vision Conference (BMVC); 2018.
[85] Liu S, Li T, Chen W, Li H. Soft rasterizer: a differentiable renderer for image-based 3D reasoning. Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2019.
[86] Kato H, Harada T. Learning view priors for single-view 3D reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 9778–87.
[87] Petersen F, Bermano AH, Deussen O, Cohen-Or D. Pix2vex: image-to-geometry reconstruction using a smooth differentiable renderer 2019 arXiv preprint arXiv:190311149.
[88] Xiang N, Wang L, Jiang T, Li Y, Yang X, Zhang J. Single-image mesh reconstruction and pose estimation via generative normal map. In: Proceedings of the 32nd International Conference on Computer Animation and Social Agents. ACM; 2019. p. 79–84.
[89] Mirza M, Osindero S. Conditional generative adversarial nets 2014 arXiv preprint arXiv:14111784.
[90] Li X, Liu S, Kim K, De Mello S, Jampani V, Yang M-H, et al. Self-supervised single-view 3D reconstruction via semantic consistency. In: Proceedings of the European Conference on Computer Vision (ECCV); 2020. p. 677–93.
[91] Kong C, Lin C-H, Lucey S. Using locally corresponding cad models for dense 3D reconstructions from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 4857–65.
[92] Pontes JK, Kong C, Eriksson A, Fookes C, Sridharan S, Lucey S. Compact model representation for 3D reconstruction. 2017 International Conference on 3D Vision (3DV) 2017:88–96.
[93] Amberg B, Romdhani S, Vetter T. Optimal step nonrigid icp algorithms for surface registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2007. p. 1–8.
[94] Pontes JK, Kong C, Sridharan S, Lucey S, Eriksson A, Fookes C. Image2Mesh: A learning framework for single image 3D reconstruction. In: Asian Conference on Computer Vision. Springer; 2018. p. 365–81.
[95] Jack D, Pontes JK, Sridharan S, Fookes C, Shirazi S, Maire F, et al. Learning free-form deformations for 3D object reconstruction. In: Asian Conference on Computer Vision. Springer; 2018. p. 317–33.
[96] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications 2017 arXiv preprint arXiv:170404861.
[97] Groueix T, Fisher M, Kim VG, Russell B, Aubry M. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 216– 224.
[98] Liao Y, Donne S, Geiger A. Deep marching cubes: Learning explicit surface representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 2916–25.
[99] Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. ACM siggraph computer graphics 1987;21(4):163–9.

[100] Smith E, Fujimoto S, Romero A, Meger D. GEOMetrics: Exploiting geometric structure for graph-encoded objects. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, 97. Long Beach, California, USA: PMLR; 2019. p. 5866–76.
[101] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems; 2016. p. 3844–52.
[102] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR); 2017.
[103] Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang Y-G. Pixel2Mesh: Generating 3D mesh models from single RGB images. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 52–67.
[104] Pan J, Li J, Han X, Jia K. Residual MeshNet: Learning to deform meshes for single-view 3D reconstruction. In: 2018 International Conference on 3D Vision (3DV). IEEE; 2018. p. 719–27.
[105] Pan J, Han X, Chen W, Tang J, Jia K. Deep mesh reconstruction from single RGB images via topology modification networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2019. p. 9964–73.
[106] Li Y, Pirk S, Su H, Qi CR, Guibas LJ. FPNN: Field probing neural networks for 3D data. In: Advances in Neural Information Processing Systems; 2016. p. 307–15.
[107] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 165–74.
[108] Mescheder L, Oechsle M, Niemeyer M, Nowozin S, Geiger A. Occupancy networks: Learning 3D reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 4460–70.
[109] Chen Z, Zhang H. Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 5939–48.
[110] Chen D-Y, Tian X-P, Shen Y-T, Ouhyoung M. On visual similarity based 3D model retrieval. Comput Graphics Forum 2003;22(3):223–32.
[111] Xu Q, Wang W, Ceylan D, Mech R, Neumann U. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In: Advances in Neural Information Processing Systems; 2019. p. 492–502.
[112] Michalkiewicz M, Pontes JK, Jack D, Baktashmotlagh M, Eriksson A. Implicit surface representations as layers in neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2019. p. 4743–52.
[113] Genova K, Cole F, Vlasic D, Sarna A, Freeman WT, Funkhouser T. Learning shape templates with structured implicit functions. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2019. p. 7154–64.
[114] Genova K, Cole F, Sud A, Sarna A, Funkhouser T. Local deep implicit functions for 3D shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 4857–66.
[115] Yamashita K, Nobuhara S, Nishino K. 3D-GMNet: Single-view 3D shape recovery as a gaussian mixture. In: Proceedings of the British Machine Vision Conference (BMVC). BMVA Press; 2020.
[116] Hartley R, Zisserman A. Multiple view geometry in computer vision. Cambridge university press; 2003.
[117] Thai A, Stojanov S, Upadhya V, Rehg JM. 3D Reconstruction of novel object shapes from single images 2020 arXiv preprint arXiv:200607752.
[118] Tulsiani S, Su H, Guibas LJ, Efros AA, Malik J. Learning shape abstractions by assembling volumetric primitives. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 2635–43.
[119] Li J, Xu K, Chaudhuri S, Yumer E, Zhang H, Guibas L. GRASS: Generative recursive autoencoders for shape structures. ACM Transactions on Graphics (TOG) 2017;36(4):52.
[120] Zou C, Yumer E, Yang J, Ceylan D, Hoiem D. 3D-PRNN: Generating shape primitives with recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017. p. 900–9.
[121] Niu C, Li J, Xu K. Im2Struct: Recovering 3D shape structure from a single RGB image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 4521–9.
[122] Tang J, Han X, Pan J, Jia K, Tong X. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 4541–50.
[123] Deng B, Genova K, Yazdani S, Bouaziz S, Hinton G, Tagliasacchi A. CvxNet: Learnable convex decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 31–41.
[124] Chen Z, Tagliasacchi A, Zhang H. BSP-Net: Generating compact meshes via binary space partitioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 45–54.
[125] Poursaeed O, Fisher M, Aigerman N, Kim VG. Coupling explicit and implicit surface representations for generative 3D modeling. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer International Publishing; 2020. p. 667–83.
[126] Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, et al. ShapeNet: An information-rich 3D model repository Tech. Rep.. Stanford University — Princeton University — Toyota Technological Institute at Chicago; 2015.
[127] Sedaghat N, Zolfaghari M, Amiri E, Brox T. Orientation-boosted voxel nets for 3D object recognition. In: Proceedings of the British Machine Vision Conference (BMVC); 2017. p. 97.1–97.13.
[128] Chu Wang MP, Siddiqi K. Dominant set clustering and pooling for multi-view 3D object recognition. In: Proceedings of the British Machine Vision Conference (BMVC). BMVA Press; 2017. p. 64.1–64.12.
[129] Xiang Y, Kim W, Chen W, Ji J, Choy C, Su H, et al. ObjectNet3D: A large scale database for 3D object recognition. In: Proceedings of the European Conference on Computer Vision (ECCV); 2016. p. 160–76.
[130] Lim JJ, Pirsiavash H, Torralba A. Parsing IKEA objects: Fine pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2013. p. 2992–9.
[131] Xiang Y, Mottaghi R, Savarese S. Beyond PASCAL: A benchmark for 3D object detection in the wild. In: IEEE Winter Conference on Applications of Computer Vision (WACV); 2014. p. 75–82.
[132] Sun X, Wu J, Zhang X, Zhang Z, Zhang C, Xue T, et al. Pix3D: Dataset and methods for single-image 3D shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 2974– 2983.
[133] Koch S, Matveev A, Jiang Z, Williams F, Artemov A, Burnaev E, et al. ABC: A big CAD model dataset for geometric deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 9601–11.
[134] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. Comput Graphics Forum 1998;17(2):167–74.
[135] Tatarchenko M, Richter SR, Ranftl R, Li Z, Koltun V, Brox T. What do single-view 3D reconstruction networks learn?. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 3405–14.
[136] Tancik M, Srinivasan PP, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, et al. Fourier features let networks learn high frequency functions in low dimensional domains. Adv Neural Inf Process Syst 2020.
[137] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV); 2020.
[138] Li J, Niu C, Xu K. Learning part generation and assembly for structure-aware shape synthesis. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence. AAAI Press; 2020. p. 11362–9.
[139] Deprelle T, Groueix T, Fisher M, Kim V, Russell B, Aubry M. Learning elementary structures for 3D shape generation and matching. In: Advances in Neural Information Processing Systems, 32; 2019. p. 7435–45.
[140] Shin D, Fowlkes CC, Hoiem D. Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 3061–9.
[141] J K, Smith E, Lafleche J-F, Fuji Tsang C, Rozantsev A, Chen W, et al. Kaolin: a pytorch library for accelerating 3D deep learning research 2019 arXiv preprint arXiv:191105063.
[142] Ravi N, Reizenstein J, Novotny D, Gordon T, Lo W-Y, Johnson J, et al. Accelerating 3D deep learning with pytorch3d 2020 arXiv preprint arXiv:200708501.