sensors

MDPI

Article

Adaptive Points Sampling for Implicit Field Reconstruction of Industrial Digital Twin

Jiongchao Jin [1,2,*], Huanqiang Xu [1] and Biao Leng [1]

[1] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2] Beijing GlacierAI Technology Co., Ltd., Beijing 100084, China
* Correspondence: jinjiongchao@buaa.edu.cn

Abstract: Nowadays, the digital twin (DT) plays an important role in Industry 4.0. It aims to model reality in the digital space for further industrial maintenance, management, and optimization. Previously, many AI technologies have been applied in this field and provide strong tools to connect physical and virtual spaces. However, we found that single-view 3D reconstruction (SVR) for DT has not been thoroughly studied. SVR can generate 3D digital models of real industrial products from just a single image. The application of SVR technology would bring convenience, cheapness, and robustness to modeling physical objects in digital space. However, the existing SVR methods cannot perform well in the reconstruction of details, which is indispensable and challenging in industrial products. In this paper, we propose a new detail-aware feature extraction network based on a feature pyramid network (FPN) for better detail reconstruction. Then, an extra network is designed to combine convolutional feature maps from different levels. Moreover, we also propose a novel adaptive points-sampling strategy to adaptively change the learning difficulty according to the training status. This can accelerate the training process and improve the fine-tuned network performance as well. Finally, we conduct comprehensive experiments on both the general objects dataset ShapeNet and a collected industrial dataset to prove the effectiveness of our methods and the practicability of the SVR technology for DT.

Keywords: deep learning; digital twin; implicit field; 3D reconstruction

Citation: Jin, J.; Xu, H.; Leng, B. Adaptive Points Sampling for Implicit Field Reconstruction of Industrial Digital Twin. *Sensors* **2022**, *22*, 6630. https://doi.org/10.3390/s22176630

Academic Editors: Zhihan Lv, Kai Xu and Zhigeng Pan

Received: 23 July 2022
Accepted: 26 August 2022
Published: 2 September 2022

1. Introduction

With the rapid development of the Internet of things (IoT) in Industry 4.0, the interaction between physical and digital spaces has become an essential process in the industrial world. The digital twin (DT) [1–3] was brought out to bidirectionally bridge the virtual world and reality. DT is a complex concept, closely related to robotics technology, computer vision, computer graphics, artificial intelligence, and other areas in computer science. It enables real-time modeling, monitoring, analysis, prediction, and control of physical objects [4,5]. DT can also significantly improve industry chain collaboration, urban management, and industrial system optimization [6–9]. Importantly, the tasks of the digital twin cannot be accomplished without 3D models.

The 3D digital model plays an indispensable role throughout the different stages of industrial production, including industrial product design, manufacturing, and maintenance. However, the interaction between authentic products and 3D digital models is still a challenging problem. It is also an important research topic in DT. In existing systems, sophisticated sensors are needed to capture the 3D data of products to reconstruct 3D digital models. In our paper, we try to utilize single-view 3D reconstruction (SVR) technology to reconstruct 3D models from one 2D image of the real world. Compared with complex sensor-based model reconstruction, an image input can be easily obtained by mobile devices and processed with less computing power, which enables SVR to be applied in a wide range of scenarios.

Nowadays, the most popular SVR methods are based on implicit field [10–12]. Previous works have used meshes, point clouds, or voxels to discretely represent a 3D model, which makes it impossible for them to reconstruct a smooth surface for 3D model shapes. However, implicit-field-based SVR methods represent 3D models with a continuous SDF function. In the 3D decoder part, they train a neural network as a binary classifier with supervision by implicit fuction (i.e., SDF function). The decoder network takes the coordinates of 3D points as queries and outputs the predictions of whether the given points are inside or outside of the 3D object's surface. In this way, the network generates an implicit representation of the 3D objects. When the number of input points is large enough, the surface of the 3D object can be well reconstructed. Implicit representation avoids using neural networks to predict complex geometry structures, which is difficult to do. It also utilizes networks to fit the decision surface, which is an advantage of neural networks. In addition, implicit representation has no limit in resolution. Theoretically, the more query points we input, the more precise of a surface we generate. However, the existing implicit-field-based methods are not suitable for industrial production. Without sufficient utilization of image information and a careful data-sampling strategy, learning with implicit field cannot perform well on sharp edges or surface details [13], which are indispensable for industrial product reconstruction.

To better reconstruct the details of industrial products, the local information in the input image should be utilized. Most of the popular implicit-field-based methods [10–12] only use a convolutional neural network (CNN) to extract a single feature vector from the input image, ignoring the local information. Other methods, such as [14,15], obtain multi-level feature maps by CNN and obtain local information from a query point on these feature maps. However, the low-level feature maps are extracted by a tiny network and contain limited semantic information. To handle this problem, we propose a feature-extracting network based on feature pyramid networks (FPN) [16]. Then, an extra network is used to combine high-level feature maps with rich semantic information, and low-level feature maps containing texture information. This way, we can reserve enough image information to reconstruct the industrial products with local and detail information.

The data-sampling strategy is another significant part of training neural networks. Previous implicit-field-based reconstruction methods neglected it and used a default sampling strategy to train their networks. However, fitting the decision surface is still a challenging task, especially in those implicit-field-based networks with complex branches. If the training data is hard to learn at the beginning, it will take a lot of time to converge. However, if the training data is too simple to indicate the exact surface of products, it will hurt the accuracy of the reconstruction in the end. The existing methods have to face this problem because the same sampled data is used throughout the training process. It is difficult for them to adaptively find a balance between the training points. Therefore, we propose an adaptive data-sampling strategy that can dynamically change the learning difficulty by varying input points according to the status of the network. It turns out that this design is able to accelerate the training speed and also improve the performance of the final fine-tuned network. In our paper, we aim to build a connection between physical and virtual spaces using single-view reconstruction technology for industrial products. We reconstruct 3D models in a digital twin field with a convenient and robust image input instead of complex high-cost multi-sensors. However, the existing single-view reconstruction methods cannot perform well on detail reconstruction. We propose a new feature-extractor network and a novel adaptive data-sampling strategy to reconstruct the 3D models together with detail and edge optimization. Then, we perform comprehensive experiments on both ShapeNet and a collected industrial dataset, the results of which indicate that our proposed method can achieve better performance than other SOTAs. The main contributions of this paper are summarized as follows:

1. We leverage SVR technology to model physical industrial products in digital space using only one image. Experiments on industrial datasets show that implicit-field-based reconstruction methods have great potential for the digital twin field.

2. We propose a feature extractor network for implicit field learning. An extra network is used to take advantage of FPN architecture that combines multi-level feature maps to extract feature vectors with rich semantics and texture information.
3. A novel adaptive data-sampling strategy is proposed to accelerate and stabilize the training of the network and improve the performance of the reconstruction.

The rest of the article is organized as follows: Section 2 summarizes the related work. Section 3 provides definitions of symbols for implicit field learning and describes their usage in DT. In Section 4, the proposed network architecture and data-sampling strategy are introduced. Then, experimental comparisons are provided in Section 5. At last, Section 6 concludes this article.

2. Related Work

2.1. Digital Twin Technology

The digital twin represents a digital system of natural objects or subjects, including their data, function, and communication capabilities [1]. Digital representation brings convenience for analysis, optimization, verification, and validation, and therefore it can reduce the cost of industrial production. Simulation technology is key to the application of DT. The authors of [17] claim that machine tools can be simulated as virtual machine tools in a safe and cost-effective way. They propose an integration of manufacturing data and sensory data for developing a digital twin of machine tools to improve accountability and capabilities for cyber-physical manufacturing. A similar view is presented by the authors of [18], who emphasize the importance of simulation more. They propose an experimental digital twin (EDT), which means the application of simulation techniques brings digital twins to life and makes them experimental. In addition to virtual and physical industrial equipment, [19] also takes a human into consideration and presents a human-cyber-physical system in intelligent manufacturing. They also consider that the new generation of digital manufacturing consists of three main factors: human, network, and physical system.

Another critical challenge is digitalization technology, which constructs and maintains the digital twin of the existing physical system. Based on deep learning technology, [20] proposes a hybrid neural network model and a small object detection algorithm. Then, a cyber-physical system is built, with the aim of realizing dynamic synchronization between a physical manufacturing system and its virtual representation. The concern of [21] is the poor flexibility of current intelligent manufacturing systems caused by their centralized architecture. They propose applying blockchain technology to build a decentralized industrial Internet of things. The same idea is shared by [22], which presents an iterative bi-level hybrid intelligence model combining a permission blockchain with a holistic optimization model. They use smart contracts to decentralize task execution among machine tools and a digital twin model to apply coarse-grained holistic optimization.

2.2. Implicit-Field-Based SVR Methods

The existing SVR methods can be classified into two categories: geometry-based and implicit-field-based methods. The geometry-based methods can reconstruct 3D objects by generating visible geometry representations, such as meshes [23,24], voxels [25,26], and points [27,28]. However, meshed points are sparse and irregular, which makes them difficult to analyze. The storage occupied by a voxel is cubic of its resolution, so we would face a difficult balance between the quality and the storage cost of the voxels.

Implicit field learning avoids the above problems by representing the 3D shape implicitly and has brought significant improvement for SVR. A similar idea is presented in [10–12], which make predictions for each 3D point to reconstruct 3D shapes. This design avoids complex geometric representations of 3D shapes and is beneficial for producing contiguous surfaces. However, due to the flaws in the perception of objects in these works, the reconstructed shapes are too coarse [13]. Some critical parts of shapes, such as edges and corners, are disconnected or linked up at the wrong scale. To address this problem, refs. [14,15] extract feature vectors, combining global and local 2D features. Furthermore, the work in [29]

independently extracts the hierarchy of local elements at different levels and performs recombination of these partial shapes in various sizes. These methods can capture richer local information from the input image and perform a more accurate reconstruction. However, these methods have to introduce a complex architecture and coarse results, which make them hardly expanded or embedded. In our paper, we avoid this problem by simplifying the camera assumption and presenting a concise reconstruction pipeline.

2.3. Detail Reconstruction with SVR

Compared to multi-view reconstruction (MVR), there is much less useful visual information in SVR. Though SVR methods can reconstruct 3D shapes successfully, some details are usually neglected. With the development of SVR, there has been a tendency towards reconstruction focusing on details. Local feature encoding is a common idea to address this problem. In [30], an hourglass network is used to extract feature maps from input images. Then, 3D points are mapped onto the 2D feature maps to extract the point-wise feature vector. In [15], 3D shape reconstruction is decoupled into shape reconstruction and residual reconstruction. The former produces a smooth main body according to the global feature vector, and the latter generates the residual shape to reconstruct the details. Aiming to learn the 3D shape from a global perspective, the result is projected onto a 2D plane, and the difference between the projection and the ground-truth shape is used as a part of the loss function. Besides local feature encoding, many other technologies have been tried. For example, [31] proposes a minimum circumference loss that trains the network in an easy-to-hard way. At the beginning of training, the loss function has a high tolerance, and the network focuses on the reconstruction of the main body. Then, the penalty for false prediction is increased, which helps supervise the learning of detail reconstruction.

3. Overview of Implicit Field Learning

3.1. Problem Definition

Given an image I of a product, the goal of SVR is to reconstruct a 3D digital shape O that captures not only the overall structure but also fine-grained details of the product. Implicit-field-based methods use an implicit function f to represent the digital shape.

$$f(p,z) = \begin{cases} 1, & \text{if } p \text{ is occupied by } O, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where p is any point in 3D digital space, and z is called the feature vector and represents the reconstructed object.

The task of implicit field learning is to train a feature extractor network E and a reconstruction network D. Network E is trained to extract feature vector z, and network D is designed to fit f.

$$\begin{aligned} z &= E(I) \\ D(p,z) &= \begin{cases} 1, & \text{if } p \text{ is occupied by } O, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

To train the networks, we need to sample a number of point–value pairs (p, v) as training data from the ground-truth shape O, where v is the label indicating whether the point p is occupied by O or not. Given a batch of point–value pairs (P, V) and the corresponding image I, we use the binary cross-entropy (BCE) function as the loss function and formulate it as follows:

$$\begin{aligned} L(I,P,V) &= \sum_{p \in P, v \in V} v \log v' + (1-v)\log(1-v') \\ v' &= D(p,z) \end{aligned} \tag{3}$$

Then, gradient descend algorithms, such as the Adam optimizer [32], can update the parameters of D and E.

3.2. Overview of the SVR Pipeline

Figure 1 shows the network architecture of common implicit-field-based SVR methods. It consists of feature extractor network E and reconstruction network D. The network E is usually implemented by CNN and is responsible for extracting the feature vector z from the input image I. The reconstruction network E, typically a multi-layer perceptron (MLP), takes the feature vector z and a 3D point p as input and predicts the label v of the input point. To generate surfaces from the implicit field in the inference phase, we can construct a 3D grid of the specified resolution, where $256 \times 256 \times 256$ is commonly used. Then, the fine-tuned networks are used to make a prediction for each grid point so that the 3D grid contains the voxelized reconstruction result. Finally, the marching cube algorithm [33] is applied to extract the surface from the voxelized shape. Thus, we can get various geometric representations from implicit field learning.

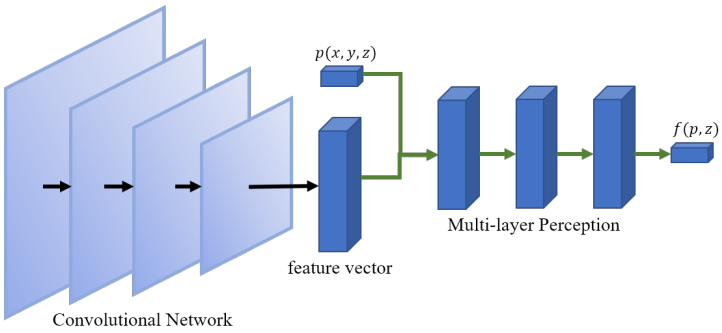


Figure 1. Common network architecture of implicit-field-based SVR.

4. Method

Based on the existing SVR methods, we propose two improvements. First, we propose an FPN-based feature extractor network to obtain rich information in the feature vector. Figure 2 shows the overview of our network architecture. Then, we design a novel adaptive data sampling strategy to get point–value pair (p, v) to train the networks efficiently and stably.

4.1. Feature Extractor Network

In implicit field learning, z is the only basis of network E for learning about the reconstructed product. It is natural that one feature vector z represents one product. Figure 3a shows the ordinary feature extraction, where the final feature map from CNN is average-pooled into a feature vector z. However, it also could be a bottleneck of the reconstruction algorithm. To alleviate the burden on z, we can change the form of network E as follows, which means one feature vector represents just one point of the product

$$z = E(I, p) \tag{4}$$

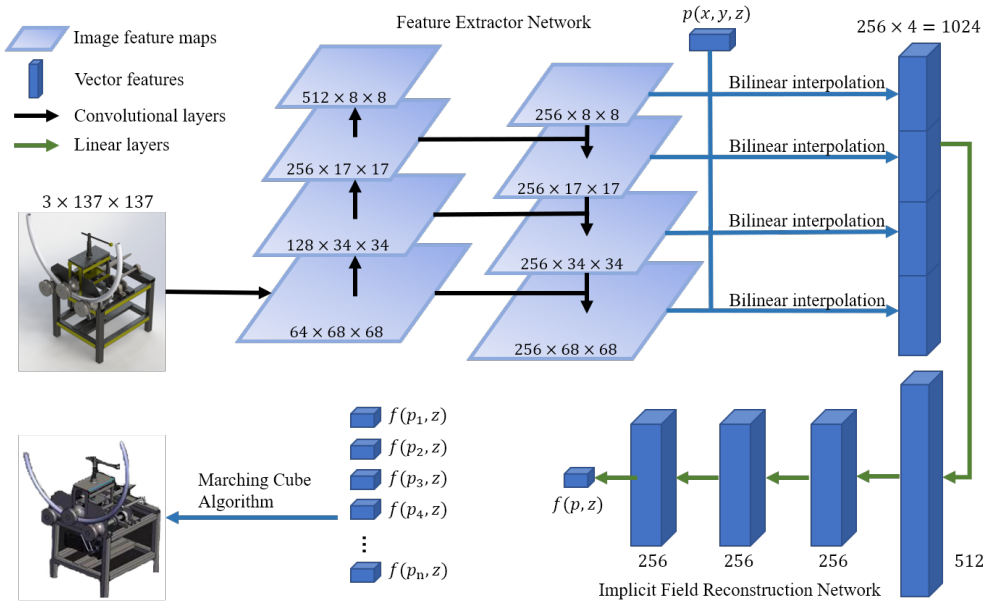


Figure 2. Pipeline of our proposed method.

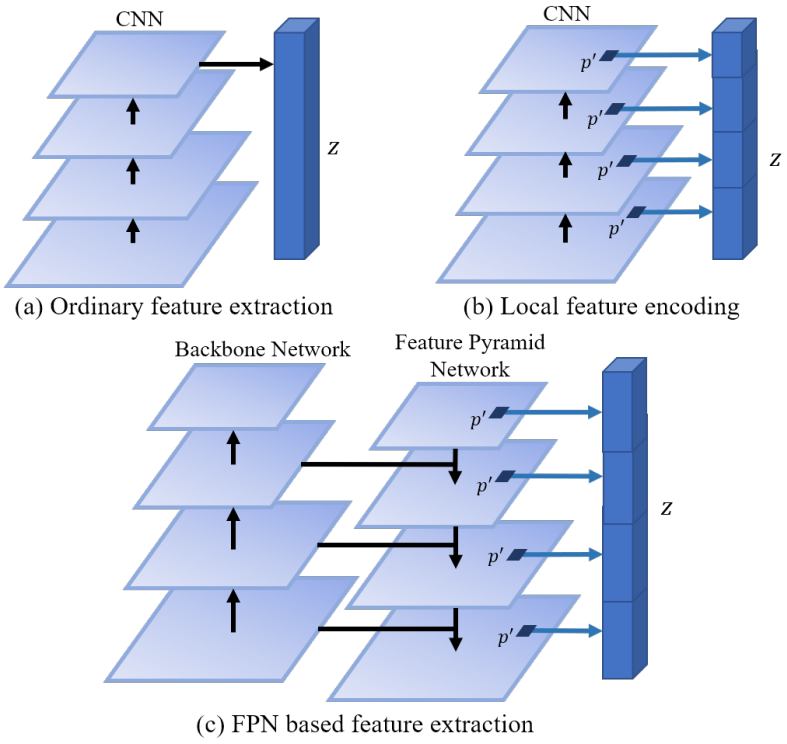


Figure 3. Different structures of feature extractors.

This can be implemented as shown in Figure 3b. Several feature vectors of the point p are extracted from different levels' feature maps and then concatenated into the final feature vector z. In this architecture, feature maps from different levels are leveraged. However, the feature maps from low levels are obtained by a few convolutional layers and so contain limited semantic information.

To tackle the above problems, we propose the feature extractor network shown in Figure 3c. The backbone network consists of four stages, the same as common CNN designs, and so will produce four feature maps from different levels. The low-level feature map is obtained by a few layers of convolutions and contains rich texture information but little

semantic information. To address this problem, an extra network is used to combine the feature map from different levels in reverse order. Then, four combined feature maps are produced, and we can get z corresponding to the point p from them.

To gather z from 2D feature maps, the 3D point $p(p_x, p_y, p_z)$ in space should be mapped onto 2D point $p'(p'_x, p'_y)$ in the image plane, where x and y are axes in the 2D image plane and z is the axis perpendicular to them. The commonly used method [14,15] uses a camera pose estimation network to predict the parameters of the projective transformation, which leads to more computations and errors. We notice that when the distances between each part of the product and the camera do not change too much, this transformation can be treated as an orthographic projection. This is usually the case with industrial products. Therefore, the 2D point p' can be computed as follows:

$$p' = (p'_x, p'_y) = (a \cdot p_x, b \cdot p_y) \tag{5}$$

where a and b are parameters representing the ratio between the product sizes in the 2D image and 3D space and can be computed easily. Then, we can locate p' in the feature maps and apply a bilinear interpolation algorithm to get the corresponding feature vector. The feature vectors from four feature maps are concatenated into the final feature vector z.

4.2. Adaptive Data Sampling Strategy

As mentioned in Section 3, the training data of the networks is in the form of point–value pairs (p, v). The value $v \in \{0, 1\}$ indicates whether the point p is occupied by the product or not. The binary labels carry limited information and lead to inefficient training. We notice that properties of the function $y = s \log x$ are useful to improve binary classification learning. Firstly, y is a strict convex function of x. This property can be leveraged to produce different weights for training different samples. Secondly, s is a hyperparameter controlling a function that is closer to a linear function or a piecewise function.

To apply the function to implicit field learning, we redefine v as follows:

$$v(d, s) = -s \log(m \cdot d + n) \tag{6}$$

where d denotes the distance between the point p and the product surface, while m and n are constant values and can be solved according to the following constraints:

$$\forall s \in (0, 1), \begin{cases} v(0, s) & = 1 \\ v(d^*, s) & = 0 \end{cases}. \tag{7}$$

The first constraint means when the point p is on the product surface ($d = 0$), the label v should be a positive label. The second one means when the point p is far away from the surface ($d \geq d^*$), the label v should be a negative label. In addition, $d^* = \sqrt{5^3}$ is a hyperparameter representing that we only care about the $5 \times 5 \times 5$ cube neighborhood of p. Then, we can solve the following:

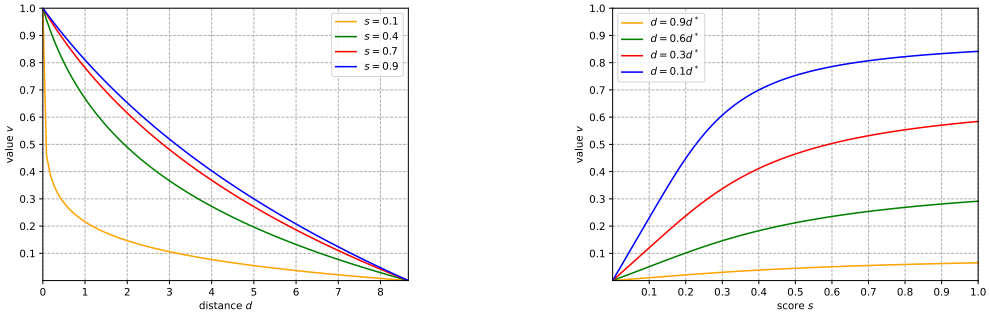
$$\begin{cases} m & = \frac{1 - e^{-\frac{1}{s}}}{d^*} \\ n & = e^{-\frac{1}{s}} \end{cases} \tag{8}$$

Correspondingly, v can be defined as follows:

$$v(d, s) = \begin{cases} -s \log(\frac{1 - e^{-\frac{1}{s}}}{d^*} d + e^{-\frac{1}{s}}) & \text{if } d \leq d^*, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where s is a value indicating the training status of the networks (the smaller, the better). As shown in Figure 4a, v is negatively correlated with d, which means the closer a point is to

the surface, the more we expect the network to predict it as an occupied point. In addition, v is a strict convex function of d. This property will produce a much larger penalty for points closer to the surface and a relatively smaller penalty for farther ones. This mechanism can also be understood as adding a larger weight to hard samples and will help the networks to learn more efficiently.



(**a**) $v - d$ curve. (**b**) $v - s$ curve.

Figure 4. Visualization of $v(d, s)$.

The new definition of v changes it from a binary label to a contiguous value, which makes it so that each (p, v) can reveal more information about the surface. This way, the learning difficulty is reduced at the beginning of training. However, it will be hard for the networks to fit the exact surface, because v will be ambiguous if the point p is close to the surface, and so the networks will only reconstruct a rough and bloated shape. To tackle this problem, we set s dynamically in the training process. As shown in Figure 4b, when s is close to 1, the curve is smooth and descends slowly. However, when s is small (e.g., 0.1), the curve becomes steep, and v is close to the binary label. Based on this phenomenon, we can claim that the value of s controls the learning difficulty. Intuitively, at the beginning of the training process, s should be assigned a relatively large value. The training samples' (p, v)s contain information about p's neighborhood. Therefore, the networks can learn efficiently and arrive at a stable status. Then, s should be set larger so that the networks can fit the product's exact surface.

To avoid setting s manually in the training process, we choose *F*1-*score* [34] to evaluate the training status and assign s as follows:

$$s = 1 - \lambda \cdot f1\_score \tag{10}$$

where $\lambda = 1.5$ is used to adjust the value to an appropriate range. We call point–value pairs $(p, 1)$ positive samples and the others negative ones. We also use tp (true positive) and fn (false negative) to denote the number of positive samples that the networks predict correctly and incorrectly, respectively, while fp (false positive) denotes that of negative samples predicted incorrectly by the networks. Therefore, the *F*1-*score* can be defined as follows:

$$\begin{aligned} precision &= \frac{tp}{tp + fp} \\ recall &= \frac{tp}{tp + fn} \\ f1\_score &= 2\frac{precision \cdot recall}{precision + recall} \end{aligned} \tag{11}$$

We compute the *F*1-*score* every five epochs and assign it to s if it is larger than the previous score. At the beginning of training, we compute the ratio r of positive samples

over all samples and assume that the networks perform prediction randomly. Therefore, the initial s is computed as follows:

$$\begin{aligned} precision &= r \\ recall &= \frac{1}{2} \\ f1\_score &= 2\frac{r \cdot \frac{1}{2}}{r + \frac{1}{2}} = \frac{2r}{1+2r} \end{aligned} \tag{12}$$

5. Experiments

To evaluate the proposed method, we first compare it with the state-of-the-art algorithms on a general 3D shapes dataset [35]. Then, we conduct experiments on industrial products to prove the practicability of our SVR technology for DT.

5.1. Implementation Details

To implement our method, ResNet-18 [36] is used as the backbone network, and the implementation of FPN is followed [16]. The implicit field reconstruction network consists of five linear layers, each followed by a Leaky-ReLU [37] activation and a batch normalization [38] layer. To sample the training point–value pairs, we first sample 9872 points from the surface of objects and add random offsets to the points. We also generate 128 random points in the 3D spaces. Then, we assign values for the 10,000 points by the strategy proposed in Section 4.2 and obtain the training point–value pairs. We use the binary cross-entropy function as the loss function and Adam optimizer with learning rate $1e^{-4}$. We train the networks for 50 epochs. For training stability, we adjust the value of s every 10 epoch instead of every epoch. At the inference phase, the threshold of the marching cube algorithm is 0.5.

5.2. Dataset and Metrics

For comparison with other SVR methods, we conduct experiments on 13 categories of the ShapeNet [35] dataset, including 43,781 objects. The input images are provided by 3D-R2N2 [25]. As is common, 23 images per object are used in training, and the last image is used for testing. The voxel dataset provided by HSP [39] is used for training the data sampling. We follow the same train/test split as IM-NET.

The quantitative metrics we used are intersection of union (IoU), Chamfer-L_1 distance (CD), edge Chamfer distance (ECD), and DR-KFS. Edge Chamfer distance [10] is computed in the same way as Chamfer-L_1 distance, but only takes edge points into consideration. For a point p, we define its sharpness $\sigma(p)$ as follows:

$$\sigma(p) = \min_{p' \in \mathcal{N}_\epsilon(p)} |n_p \cdot n'_p| \tag{13}$$

where $\mathcal{N}_\epsilon(p)$ denotes neighbor points of p within distance ϵ, while n_p and n'_p are the unit normal vectors of p and p'. We set ϵ to 0.01, and only points with $\sigma(p) < 0.1$ are treated as edge points. DR-KFS is implemented according to [40], and the results are normalized into $[0, 1]$.

5.3. Experiments on General Objects

Our method is compared with Pixel2Mesh [24], AtlasNet [41], OccNet [10], IM-Net [11], and DISN [14]. OccNet and IM-Net both use the most common network architecture, shown in Figure 1. OccNet trains the whole pipeline in a end-to-end way, while IM-Net trains the CNN and MLP separately. Based on OccNet, DISN improves the feature extraction network, as shown in Figure 3b. All three methods use binary labels to train the networks.

Qualitative Results. Figure 5 shows the qualitative results. OccNet is more likely to produce thick meshes. This is a benefit in generating the main body of the 3D object but harmful to reconstruction of the details and edges. For example, OccNet can reconstruct the

body of a handgun, but fails to recover the trigger. In contrast, IM-Net tends to produce thin meshes, but also generates more fragments. As we can see, it cannot reconstruct the complete connections for chair legs and airplane engines. DISN makes accurate reconstructions for some categories, such as chairs and tables. However, occasionally it cannot generate complete planes and fails to generate flat edges. The results of our methods are shown in the last column. Not only the main body and structure of 3D objects are reconstructed, but more details are also reserved.

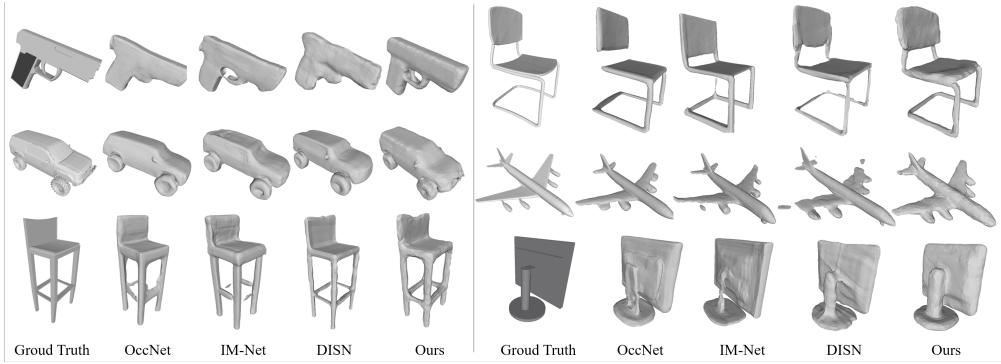


Figure 5. Reconstruction results on industrial machines.

Quantitative Results. Table 1 shows a quantitative comparison of the seven categories with the most shapes in the ShapNet-Core dataset [35]. The mean value of each metric in computed over all samples. The $\downarrow$ means the lower, the better; while $\uparrow$ is the opposite. As shown in Table 1, our method shows the best performance on most categories. Especially on chairs and tables, our method significantly outperforms the other methods. As mentioned above, OccNet does well in the reconstruction of objects' main body, but not in generating edges and details. The main body of an airplane or a car takes a large part of the whole shape, so the OccNet has outstanding results. However, over all the categories of ShapeNet-Core, our method shows a better performance.

Ablation Study. We conduct ablation studies on the proposed design, and the results are shown in Table 2. The first row describes the used feature extractor network. *CNN* represents when only the ordinary CNN, ResNet-18, is used. *CNN + FPN* represents the proposed design where the extra FPN is also applied. The second row describes which data sampling method is used; *binary* means v is assigned a binary label, as described in Section 3. In addition, $v(d, 0.5)$ means that the proposed function $v(d, s)$ is applied but s is fixed to 0.5, so that the learning difficulty can be dynamically adjusted, while $v(d, s)$ denotes the adaptive data-sampling strategy. It turns out that the application of FPN can bring significant improvements over ordinary CNN. Therefore, we claim that the simple feature vector mechanism can severely limit the performance of SVR methods. The value function $v(d, s)$ is also effective. Compared with binary values, our $v(d, s)$ design can carry more information and apply an appropriate penalty on the false predictions.

Furthermore, to clarify the influence of $v(d, s)$ on the reconstruction results, we fix the s to different values through the training phase and repeat the network training. The qualitative results are shown in Figure 6. When s is fixed to 0.9 or 0.7, the network produces bloated meshes. The main shape is reconstructed, but most details are lost. When s is set to a small value, such as 0.1, the network generates slim meshes. Sometimes the reconstruction results are delicate, but occasionally the surfaces are broken, and even the main shapes cannot be reconstructed. According to Figure 4a, this makes sense for the results shown in Figure 6. Even though the points are far away from objects' surface, as long as they are in the neighborhood controlled by d^*, they can get a positive value. This is quite different from the binary labels. When s is set to a large value, such as 0.9 or 0.7, the points can be assigned to relatively large values. After training with these point–value pairs, the network is more likely to predict input points as positive. With these predictions, the bloated meshes are generated by the marching cubes algorithm. When s is set to a

small value, the predictions tend to be negative, and slim meshes are produced. Therefore, it is intuitive that the proposed adaptive data-sampling strategy works by choosing the appropriate value of s dynamically according to the training status.

Table 1. Quantitative comparison between different SVR methods. Bold means the best result in current table item.

	Method	Airplane	Car	Chair	Display	Lamp	Rifle	Table	Mean
IOU(↑)	Pixel2Mesh	0.423	0.524	0.311	0.475	0.238	0.429	0.408	0.401
	AtlasNet	0.451	0.535	0.366	0.480	0.217	0.455	0.430	0.419
	OccNET	**0.480**	0.570	0.358	0.439	0.254	0.427	0.461	0.461
	IM-NET	0.379	0.674	0.487	0.514	**0.336**	0.468	0.484	0.527
	DISN	0.328	0.672	0.301	0.358	0.189	0.197	0.105	0.360
	Ours	0.443	**0.710**	**0.532**	**0.552**	0.295	**0.671**	**0.505**	**0.592**
CD(↓)	Pixel2Mesh	0.587	0.414	0.662	0.641	0.702	0.521	0.796	0.617
	AtlasNet	0.592	0.440	0.651	0.632	0.695	0.438	0.701	0.593
	OccNET	**0.461**	0.368	0.639	0.636	0.683	0.414	0.763	0.587
	IM-NET	0.574	0.650	0.919	0.907	0.802	0.556	0.979	0.797
	DISN	0.572	0.645	0.907	0.906	0.800	0.578	0.972	0.794
	Ours	0.562	**0.344**	**0.597**	**0.613**	**0.637**	**0.325**	**0.653**	**0.552**
ECD(↓)	Pixel2Mesh	0.565	0.477	0.589	0.582	0.677	0.430	0.742	0.580
	AtlasNet	0.601	0.397	0.506	0.658	0.679	0.426	0.688	0.565
	OccNET	**0.423**	**0.288**	0.465	0.475	0.581	0.321	0.594	0.473
	IM-NET	0.522	0.370	0.627	0.641	0.695	0.478	0.750	0.589
	DISN	0.554	0.412	0.732	0.653	0.733	0.565	0.844	0.655
	Ours	0.447	0.313	**0.385**	**0.469**	**0.547**	**0.296**	**0.536**	**0.452**
DR-KFS(↓)	Pixel2Mesh	0.338	0.490	0.551	0.568	0.592	0.487	0.605	0.519
	AtlasNet	0.299	0.401	0.499	0.524	0.559	0.463	0.592	0.425
	OccNET	**0.296**	**0.239**	0.325	0.366	0.402	0.291	0.438	0.337
	IM-NET	0.337	0.308	0.375	0.386	0.398	0.269	0.512	0.367
	DISN	0.324	0.313	0.392	0.403	0.422	0.401	0.524	0.397
	Ours	0.320	0.274	**0.314**	**0.361**	**0.374**	**0.258**	**0.373**	**0.288**

Table 2. Quantitative results of ablation study.

	CNN Binary	CNN + FPN Binary	CNN + FPN v(d,0.5)	CNN + FPN v(d,s)
IOU	0.524	0.553	0.567	0.592
CD	0.813	0.664	0.560	0.552
ECD	0.576	0.538	0.457	0.452

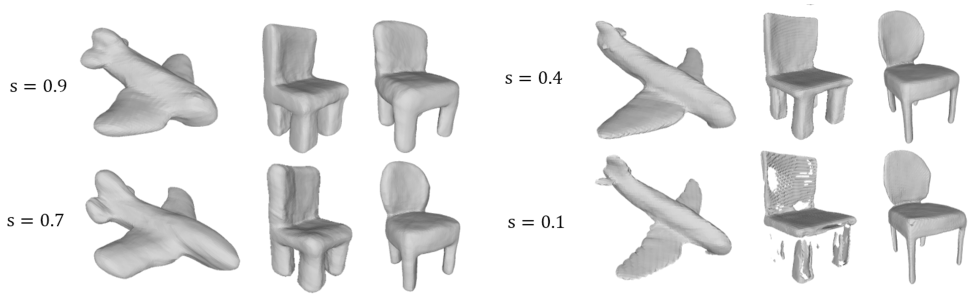


Figure 6. Reconstruction results trained with s different values.

5.4. Experiments on Industrial Machines

We collect 400 3D models of different categories and their 2D rendered images on public 3D model websites, such as TurboSquid, SketchFab, and cgmodel, and paid 3D models on Taobao. We divide the 3D models into four categories, including 195 industrial machines, 138 industrial components, 45 metaverse buildings, and 22 cartoon characters. Here, we use seven out of ten for training, two out of ten for testing, and one out of ten for validation in each category.

Then, we apply our SVR method to prove the practicability of this technology. Unlike the general 3D objects dataset, the number of industrial machines is relatively small, but each machine has a more complex structure and richer details. However, there are not enough models in the categories of metaverse buildings and cartoon characters. Therefore, we only show the qualitative reconstruction results of industrial components and machines in Figure 7. The sophisticated components and machines can be fully reconstructed, but the main structures are basically generated. Considering the reconstruction results, we believe the SVR technology has its own place in the field of industrial DT.

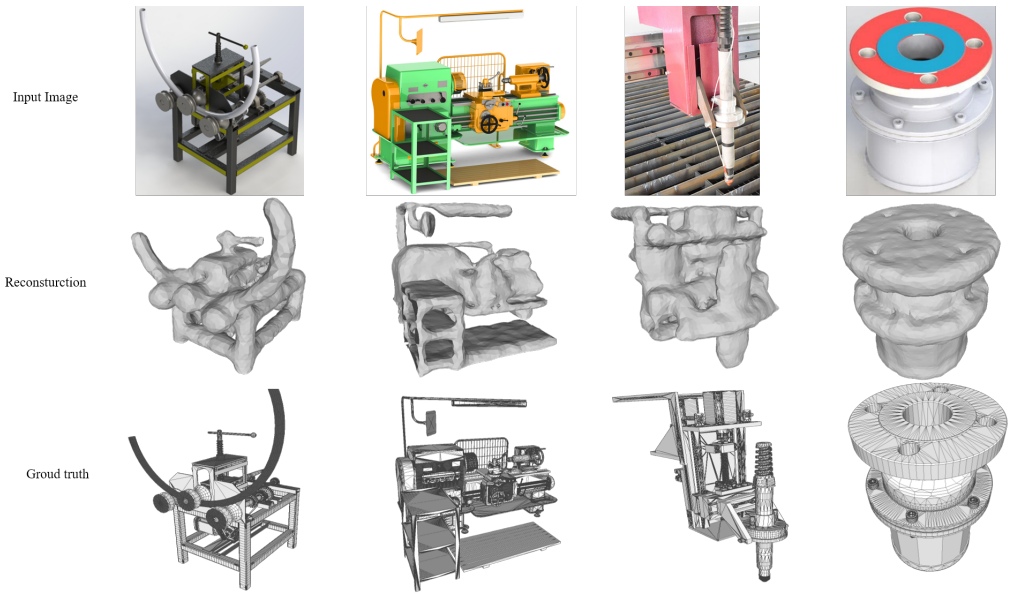


Figure 7. Reconstruction results on industrial machines.

We also compare our methods with Pixel2Mesh, OccNet, IM-NET, and DISN on our collected data to show the effectiveness of our methods quantitatively in Table 3. We evaluate each method with ECD and DR-KFS. ECD can leverage the edge and detail reconstruction quality, and DR-KFS can provide a global view of reconstructed shapes. It is obvious that our method outperforms the other methods on our collected data. In our collected data, there are rich details and edge information with hard topology. Using our methods can optimize the detail and edge reconstruction, which makes our methods perform better than others.

Table 3. Quantitative comparison between different SVR methods on our collected data. The bold means the best result when comparing with others.

	Method	Machines	Components	Buildings	Cartoons	Mean
ECD(↓)	Pixel2Mesh	0.828	0.801	0.874	0.853	0.839
	AtlasNet	0.802	0.783	0.859	0.848	0.823
	OccNET	0.795	0.763	0.848	0.820	0.806
	IM-NET	0.789	0.771	0.852	0.810	0.805
	DISN	0.798	0.784	0.851	0.816	0.812
	Ours	**0.724**	**0.712**	**0.801**	**0.790**	**0.757**
DR-KFS(↓)	Pixel2Mesh	0.742	0.709	0.885	0.838	0.793
	AtlasNet	0.757	0.701	0.873	0.850	0.795
	OccNET	0.695	0.673	0.794	0.771	0.733
	IM-NET	0.662	0.649	0.781	0.744	0.709
	DISN	0.678	0.641	0.790	0.724	0.724
	Ours	**0.589**	**0.576**	**0.703**	**0.699**	**0.641**

6. Conclusions

In this paper, we first introduce single-view reconstruction technology to build a connection from the physical space to the digital space. Through SVR technology, we can generate the digital twins of industrial productions just by relying on a single image. Compared with complex multi-sensor-based reconstruction, the convenience and cheapness of 2D images can be leveraged in the intelligence industry. It is still challenging for SVR to be utilized in the industrial world, because even if the existing SVR methods can reconstruct the shapes, they all suffer from over-smooth surface and a lack of details and edges.

To address this problem, we propose a feature extractor network and a novel data-sampling strategy in our paper. We first use CNN to extract feature maps instead of a single feature vector. Inspired by FPN, we also design an extra convolutional network to combine high-level feature maps and low-level feature maps. Then, we use a billinear interpolation algorithm to extract the feature vector corresponding to the input point. We also design an adaptive point–value (p, v) pairs sampling strategy. This strategy treats v as the function $v(d, s)$. This way, each point–value pair can carry more information about the surface. By setting s to different values according to the training status, the learning difficulty can be automatically adjusted.

For the methods proposed in this paper, there are still some improvements that can be made. For example, when we extract the feature vector of the point, we simplify the perspective transformation, which limits the usage scenario and may be designed in a more elegant way. Furthermore, the technology of the model ensemble is also a direction worth trying.

In the future, we will explore the SVR technology for industrial DT deeper, aiming to improve the quality of the reconstructed digital model. Besides, a larger 3D industrial dataset should be collected and reorganized so that more research on 3D reconstruction for DT can be conducted.

Author Contributions: Methodology, H.X.; Project administration, J.J.; Supervision, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data that support the findings of this study are openly available in the following links: https://shapenet.org/, https://ai.taobao.com/ and https://www.turbosquid.com/, accessed on 22 July 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [CrossRef]
2. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [CrossRef]
3. Gehrmann, C.; Gunnarsson, M. A Digital Twin Based Industrial Automation and Control System Security Architecture. *IEEE Trans. Ind. Inform.* **2020**, *16*, 669–680. [CrossRef]
4. Tao, F.; Zhang, M. Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *IEEE Access* **2017**, *5*, 20418–20427. [CrossRef]
5. Yang, H.; Li, Y.; Yao, K.; Sun, T.; Zhou, C. A Systematic Network Traffic Emulation Framework for Digital Twin Network. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; pp. 94–97. [CrossRef]
6. Viola, J.; Chen, Y. Parallel Self Optimizing Control Framework for Digital Twin Enabled Smart Control Engineering. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; pp. 358–361. [CrossRef]
7. An, D.; Chen, Y. Digital Twin Enabled Methane Emission Abatement Using Networked Mobile Sensing and Mobile Actuation. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; pp. 354–357. [CrossRef]
8. Zhang, Z.; Lu, J.; Xia, L.; Wang, S.; Zhang, H.; Zhao, R. Digital twin system design for dual-manipulator cooperation unit. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; Volume 1, pp. 1431–1434. [CrossRef]
9. Ji, G.; Hao, J.g.; Gao, J.l.; Lu, C.z. Digital Twin Modeling Method for Individual Combat Quadrotor UAV. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; pp. 1–4. [CrossRef]
10. Mescheder, L.M.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; Geiger, A. Occupancy Networks: Learning 3D Reconstruction in Function Space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 4460–4470. [CrossRef]
11. Chen, Z.; Zhang, H. Learning Implicit Fields for Generative Shape Modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 5939–5948. [CrossRef]
12. Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.A.; Lovegrove, S. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 165–174. [CrossRef]
13. Chen, Z.; Tagliasacchi, A.; Zhang, H. BSP-Net: Generating Compact Meshes via Binary Space Partitioning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; pp. 42–51. [CrossRef]
14. Xu, Q.; Wang, W.; Ceylan, D.; Mech, R.; Neumann, U. DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; MIT Press: Cambridge, MA, USA, 2019; pp. 490–500.
15. Li, M.; Zhang, H. D2IM-Net: Learning Detail Disentangled Implicit Fields From Single Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; pp. 10246–10255. [CrossRef]
16. Lin, T.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 936–944. [CrossRef]
17. Cai, Y.; Starly, B.; Cohen, P.; Lee, Y.S. Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing. *Procedia Manuf.* **2017**, *10*, 1031–1042. [CrossRef]
18. Schluse, M.; Priggemeyer, M.; Atorf, L.; Rossmann, J. Experimentable Digital Twins—Streamlining Simulation-Based Systems Engineering for Industry 4.0. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1722–1731. [CrossRef]
19. Zhou, J.; Zhou, Y.; Wang, B.; Zang, J. Human–Cyber–Physical Systems (HCPSs) in the Context of New-Generation Intelligent Manufacturing. *Engineering* **2019**, *5*, 624–636. [CrossRef]
20. Zhou, X.; Xu, X.; Liang, W.; Zeng, Z.; Shimizu, S.; Yang, L.T.; Jin, Q. Intelligent Small Object Detection for Digital Twin in Smart Manufacturing With Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1377–1386. [CrossRef]
21. Zhang, C.; Zhou, G.; Li, H.; Cao, Y. Manufacturing Blockchain of Things for the Configuration of a Data- and Knowledge-Driven Digital Twin Manufacturing Cell. *IEEE Internet Things J.* **2020**, *7*, 11884–11894. [CrossRef]
22. Leng, J.; Yan, D.; Liu, Q.; Xu, K.; Zhao, J.L.; Shi, R.; Wei, L.; Zhang, D.; Chen, X. ManuChain: Combining Permissioned Blockchain With a Holistic Optimization Model as Bi-Level Intelligence for Smart Manufacturing. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 182–192. [CrossRef]

23. Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. *CoRR* **2018**. Available online: http://xxx.lanl.gov/abs/1802.05384 (accessed on 15 February 2018).
24. 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), Munich, Germany, 8–14 September 2018; pp. 52–67.
25. Choy, C.B.; 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 Computer Vision—ECCV 2016—14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VIII; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9912, Lecture Notes in Computer Science, pp. 628–644. [CrossRef]
26. 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 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 2690–2698. [CrossRef]
27. Yang, G.; Huang, X.; Hao, Z.; Liu, M.; Belongie, S.J.; Hariharan, B. PointFlow: 3D Point Cloud Generation With Continuous Normalizing Flows. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 4540–4549. [CrossRef]
28. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.D.; Savarese, S. TopNet: Structural Point Cloud Decoder. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 383–392. [CrossRef]
29. Bechtold, J.; Tatarchenko, M.; Fischer, V.; Brox, T. Fostering Generalization in Single-View 3D Reconstruction by Learning a Hierarchy of Local and Global Shape Priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; Computer Vision Foundation/IEEE, pp. 15880–15889. [CrossRef]
30. Saito, S.; Huang, Z.; Natsume, R.; Morishima, S.; Li, H.; Kanazawa, A. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 2304–2314. [CrossRef]
31. Duan, Y.; Zhu, H.; Wang, H.; Yi, L.; Nevatia, R.; Guibas, L.J. Curriculum DeepSDF. In Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part VIII; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12353, Lecture Notes in Computer Science, pp. 51–67. [CrossRef]
32. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings; Bengio, Y., LeCun, Y., Eds.
33. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. In Proceedings of the Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, CA, USA, 27–31 July 1987; Stone, M.C., Ed.; ACM: New York, NY, USA, 1987; pp. 163–169. [CrossRef]
34. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 26.
35. Chang, A.X.; Funkhouser, T.A.; Guibas, L.J.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *CoRR* **2015**, abs/1512.03012. Available online: http://xxx.lanl.gov/abs/1512.03012 (accessed on 9 December 2015).
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 770–778. [CrossRef]
37. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *CoRR* **2015**, abs/1505.00853. Available online: http://xxx.lanl.gov/abs/1505.00853 (accessed on 5 May 2015).
38. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015; Bach, F.R., Blei, D.M., Eds.; JMLR.org, JMLR Workshop and Conference Proceedings, 2015; Volume 37, pp. 448–456.
39. Hane, C.; Tulsiani, S.; Malik, J. Hierarchical Surface Prediction for 3D Object Reconstruction. In Proceedings of the 2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, 10–12 October 2017; IEEE Computer Society: Washington, DC, USA; pp. 412–420. [CrossRef]
40. Jin, J.; Patil, A.G.; Xiong, Z.; Zhang, H.R. DR-KFS: A Differentiable Visual Similarity Metric for 3D Shape Reconstruction. In Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXI; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12366, Lecture Notes in Computer Science, pp. 295–311. [CrossRef]
41. Vakalopoulou, M.; Chassagnon, G.; Bus, N.; Marini, R.; Zacharaki, E.I.; Revel, M.P.; Paragios, N. AtlasNet: Multi-atlas non-linear deep networks for medical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Granada, Spain, 16–20 September 2018, Springer: Berlin/Heidelberg, Germany, 2018; pp. 658–666.