The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-019-03792-1

ORIGINAL ARTICLE

Surfel convolutional neural network for support detection in additive manufacturing

Jida Huang[1] · **Tsz-Ho Kwok**[2] · **Chi Zhou**[1] · **Wenyao Xu**[3]

Received: 3 August 2018 / Accepted: 16 April 2019

Abstract

Support generation is one of the crucial steps in 3D printing to make sure the overhang structures can be fabricated. The first step of support generation is to detect which regions need support structures. Normal-based methods can determine the support regions fast but find many unnecessary locations which could be potentially self-supported. Image-based methods conduct a layer-by-layer comparison to find support regions, which could make use of material self-support capability; however, it sacrifices the computational cost and may still fail in some applications due to the loss of topology information when conducting offset and boolean operations based on the image. In order to overcome the difficulties of image-based methods, this paper proposes a surfel convolutional neural network (SCNN)-based approach for support detection. In this method, the sampling point on the surface with normal information, named **surfel** (**surf**ace **el**ement), is defined through layered depth-normal image (LDNI) sampling method. A local surfel image which represents the local topology information of the sampling point in the solid model is then constructed. A set of models with ground-truth support regions is used to train the deep neural network. Experimental results show that the proposed method outperforms the normal-based method and image-based method in terms of accuracy, reliability, and computational cost.

Keywords Support detection · 3D printing · Additive manufacturing · Deep learning · Convolutional neural network · Surfel

1 Introduction

Three-dimensional (3D) printing is a disruptive technology that can fabricate complex objects effectively in cost and time. The process takes the Computer-Aided Design (CAD) model directly and builds the physical part layer-by-layer. However, each layer requires a previous layer to build upon, and if the geometry contains suspensions (cantilevered out from the main body) or islands (separated segments) in a layer, extra material is needed to support segments of material layers that do not have solid material underneath—so-called supporting structures or supports. Therefore, as an important pre-fabrication step, the detection of where needs to be supported and the generation of supports are crucial to make sure the CAD model can be properly fabricated. The goal is always to find the minimum amount of supports such that the model can be printed successfully because the supports are the waste materials that will be removed after the fabrication is completed. In this paper, our focus is on the detection of regions needing to be supported.

In the practice of 3D printing community, an overhang can usually be printed with no loss of quality up to a certain angle, e.g., 30°, 45°, or 60° depending on the printing process and the material. For example, the newly printed layer is supported by 50% of the previous layer at 45°, which may allow sufficient support and adhesion to build upon. When the angle is above the allowable degrees, the support is required. Therefore, an intuitive way for support detection is to check the normal along the surface of the CAD model [16]. If the normal is in the opposite direction of the printing direction and their angle is small, the position is detected as support-required. *Normal-based method* is fast and can find all the areas that needs support as illustrated in Fig. 1a. However, this method could lead to a large

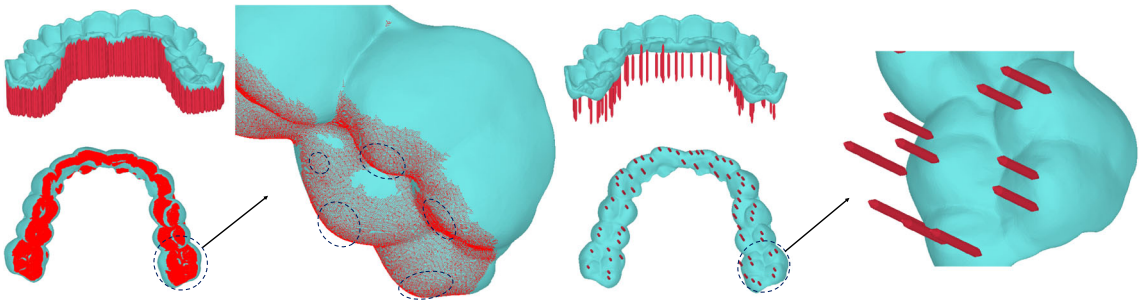
✉ Chi Zhou
chizhou@buffalo.edu

[1] Industrial and Systems Engineering, University at Buffalo, SUNY, Buffalo, NY 14260, USA

[2] Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

[3] Computer Science and Engineering, University at Buffalo, SUNY, Buffalo, NY 14260, USA

Published online: 06 May 2019



(a) Normal-based support detection

(b) Image-based support detection [3]

Fig. 1 Support region detected by normal-based method and image-based method. As is common practice, a threshold of 45° is used for normal checking. It shows that a large area of support regions is detected with many support locations unnecessary; that is, only the regions in black circles need support, and other regions can be self-supported

number of supports that may not be necessary, and the more supports are generated, the more printing time and cleaning efforts are needed. The main reason is that the normal-based method only takes the normal information into account and does not consider the neighboring structure, resulting in missing self-support information. Continuing with the previous 45° example, we know that a layer can be self-supported by 50% overlapping. Even a new layer cannot be fully supported by its previous layer at 50°, only the portion outside the 50% overlapping needs to be supported by extra material. There are some variants of normal-based methods trying to reduce the unnecessary supports through clustering or filtering [29], but it is still challenging to get the optimal amount of supports.

To make use of the partial self-support capability, another detection approach is conducted in a layer-by-layer mode. The support area is determined by comparing successive two layers after the model is sliced. For example, Chen et al. proposed an *image-based method* that makes use of the slice images computed for each layer [7]. This method first applies an offset operation on the slice image of the previous layer with a threshold to account for self-support capability, and then subtracts it from the slice image of the current layer using a boolean operation. The remaining pixels in the current layer are the regions needing to be supported. This method can compute the partially supported area directly on each layer, which follows the actual fabrication process well, and thus can eliminate many unnecessary supports that the normal-based method has. However, there are two drawbacks of the image-based method. First, since the detection requires the model to be sliced already, it is very time-consuming especially with the tiny layer thickness being used in the state-of-the-art 3D printing technology [18]. Moreover, the slicing might still need to be re-done after the supports are generated or adjustment is made by users. Second, the method detects support areas in the image domain by a number of offset and boolean operations, which would lose topology information between slices especially when small thin features and gaps in the offsetting area. For example, the image-based method reports the stair shape in Fig. 2 support-free even there are surfaces with normal opposite to the printing direction.

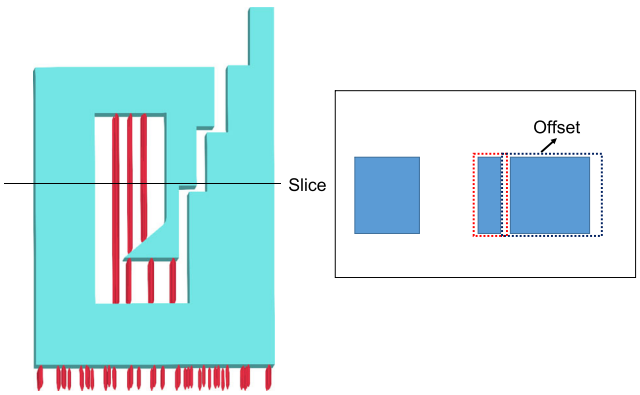


Fig. 2 Image-based method fails to detect support between the gap of the stair shape

Although the normal-based method is fast, the image-based method is more attractive as it can utilize the partial support capability to reduce the waste of material. The goal of this research is to overcome the difficulties of image-based method in terms of computational inefficiency and topology information missing. We observe that a slice image is a discretization result of a solid model and contains only local details, and processing the images layer-by-layer is inevitably time-consuming. Based on the observation, we hypothesize that if a richer representation is used, the topology information will be considered and there will be no need to process the images one-by-one. Layered depth-normal image (LDNI) is an implicit representation of a solid [30]. It is basically as simple as an image and contains an array of rays that are shot to intersect with the CAD model. It stores the intersection points on the rays with the depth and normal values. As compared to a slice image, it not only can represent the details in a

layer, but also can describe the topology along the rays. However, the research question here is how to make use of this rich representation in support detection for 3D printing? Therefore, our objective is to apply the LDNI and the convolutional neural network (CNN) to learn the topology information and directly determine if a point needs to be supported without the image processing operations. As a newly emerging area in machine learning, CNN has been widely used in image analysis, computer vision area and achieve remarkable breakthroughs. One reason for the success of deep neural networks is their ability to leverage the statistical properties of data through local statistics [23]. We believe that it is a good candidate to learn the hidden relationship between the topology information and the need for support. In addition, as in practice with the consideration of design preference, the support position usually requires manual adjustment, which should be learned as well. The main contributions of this work can be summarized as follows:

1. The LDNI representation is applied in the support detection, which not only can describe a solid model but also can reflect the topology information of a local area.
2. A surfel image is developed based on the LDNI sampling, which is a multi-channel image representation for **surf**ace **el**ements. The surfel image fits well to the existing CNN for image classification.
3. A novel surfel convolutional neural network (SCNN) is presented to learn the intrinsic property of support regions, and a new support detection pipeline is proposed for 3D printing, which can consider self-support capability without slicing the model.

Experimental results demonstrate the success of the proposed method when compared with the normal-based and image-based methods, and the physical fabrication also validates this method. To the best of our knowledge, this is the first time to apply the deep learning technique to the support detection for 3D printing.

The rest of the paper is organized as follows. Section 2 will briefly review the related works. The support learning pipeline will be discussed in Section 3. Section 4 will introduce the architecture of the proposed deep neural network, and it is followed by the experimental results in Section 5. Section 6 will conclude the paper.

2 Literature review

In this section, the related work of support detection, support generation, and deep learning, as well as its extension in 3D data will be discussed.

2.1 Support detection

It should be noted that the support detection problem is the first step of support generation, existing works are concentrated on the support structure generation, and few works only focused on the support detection. Most studies simply use intuitive methods to find the positions requiring support. This is also the motivation of this work to focus on the very first step of support generation, i.e., support detection problem.

For support detection, the most intuitive way is to check whether the angle between an overhang facet and horizontal plane (assuming the building direction is upward) is large enough to withstand the weight of the material. This can be easily performed by computing the dot product of the facet normal vector and the horizontal vector [13]. Other methods consider the self-support capability of material, the detection is based on a layer-by-layer comparison mode. Representative works are *image-based method* for stereolithography (SLA) process [7] and layer-by-layer comparison method for fused deposition modeling (FDM) process [9].

As discussed in previous section, *normal-based methods* suffer from redundant support generation and *image-based methods* have topology information missing problem. In this work, a learning-based method is investigated to overcome these drawbacks.

2.2 Support generation

Once the support positions are determined on a solid model, the following operation is to generate the support structure to hold the regions at these positions. Various methods have been proposed for different 3D printing processes.

For FDM process, Vanek et al. used a geometry-based approach that minimizes the support material while providing sufficient support [29]. Dumas et al. proposed a bridge structure to support overhangs [9]. Boyard et al. presented a method to minimize the volume of support and its impact on a part's surface finish for FDM [3]. For selective laser melting process (SLM), Calignano investigated the manufacturability of the overhanging structures in aluminum and titanium alloys by using optimized support parts [6]. Most of these works are based on simple normal-checking or process-specific parameters like G-code to detect the support positions. A general and effective support detection method is needed to find the support positions.

Other works focus on the generation methods for the support structure. For example, an orientation-driven shape optimizer is introduced in [12] to slim down the supporting

structures used in single material-based AM. Ezair et al. explored the effect of the model orientation on the volume of the needed support structure directly below the model [10]. Vaidya et al. introduced an optimum support structure generation method for AM using unit cell structures and support removal constraint [28]. Since the support structure is based on the detected support positions, this work mainly focused on the detection of support regions. It is remarked that the support generation does not affect the support detection.

2.3 Deep learning for 3D data

Deep learning refers to learning complicated concepts by representing them from simpler ones in a hierarchical or multi-layer manner [5]. In the past decade, many fields such as image and speech recognition have witnessed the re-emergence of deep learning especially the convolutional neural network (CNN) techniques [19]. With the computational power of modern graphics processing unit (GPU), many applications have been achieved qualitative breakthrough [20], and many successful deep convolutional neural networks have been trained such as AlexNet [17] and GoogleNet [26]. One of the key reasons for the success of the deep neural network is their capability of leaning the statistical properties of the data, and these statistical properties were related to intrinsic physics [23] and formalized with different types of CNNs [5, 27].

In order to extend the deep learning methods into 3D data, many attempts have been made to apply the convolution operation to 3D data. The most direct way is to use voxel representation for 3D model. For example, Wu et al. represented a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid, by using a convolutional deep belief network, to learn the distribution of complex 3D shapes for 3D object recognition problem [32]. Similarly, Brock et al. trained voxel-based variational auto-encoders for object classification [4]. Balu et al. used voxel data to train the network and learn salient features from a CAD model of a mechanical part, and then determine if the part can be manufactured or not [1].

Since voxel representation of the 3D model is not intrinsic and suffers from deformation sensitive issues, to extend convolution operation to intrinsic geometric deep learning, Bronstein et al. proposed geometric deep learning which goes beyond Euclidean data [5]. Masci et al. firstly extended convolutional neural networks to non-Euclidean domains (surfaces) by using the geodesic CNN model [22], the method is improved by Boscaini et al. [2] and further generalized by Monti et al. [24]. At the same time, Maron et al. applied convolutional neural networks for sphere-type shapes by using a global seamless parameterization to a planar flat-torus [21].

3 Support learning pipeline

It can be seen from the discussion in Section 1 that although the *normal-based method* is fast, the redundant support structures heavily impede material utilization and surface quality. While the *image-based method* can make full use of self-support capability, but misses topology information which may lead to incorrect support generation. Considering the issues of the normal-based and image-based method for support position detection, we propose to use an information-richer representation for the local topology of a surface element on a solid model and introduce a new deep learning technique for support detection problem. The overview flowchart of the proposed method is shown in Fig. 3.

As can be seen from Fig. 3, the 3D model is firstly pre-processed and the surfel image with the local topology information is extracted and serves as the input of the neural network. Here, the LDNI [30] is used to construct surfel for an input 3D model, and each element and its neighboring elements are extracted. The topology information encoded by the surfels is then represented as a multi-channel image. In the training stage, all of the multi-channel images are fed to the input of the deep neural network, which will be introduced in Section 4. The ground-truth support positions are used for the output of the network. The ground-truth support positions selection is based on the supports detected by the image-based method, and then through an interactive adjustment process for physical printing. By using deep learning technique, the relationship between the local topology information of a surfel and its status of whether a support is needed or not will be learned.

3.1 Surfel construction

LDNI is a semi-implicit representation of a solid model [8, 31]. For a solid model, a LDNI with a specified viewing direction is a two-dimensional (2D) image with $w \times w$ pixels, where each pixel contains a sequence of numbers which specify: (a) the depths from the intersections (between a ray passing through the center of a pixel along the viewing direction and the boundary surface) to the viewing plane; (b) the normal vectors of the boundary surfaces at the intersections. Therefore, each sample on the rays in a LDNI is a Hermite data. The samples on a ray are sorted in ascending order by their depth values. For simplicity, we define this ***surf*face *el*ement** on the ray as ***surfel*** in this paper.

For an implicit solid, firstly we represent it with LDNI. Taking the flower model in Fig. 4 as an example, the model can be represented by a 2D image in XY plane with $w_X \times w_Y$ pixels by using Z-LDNI. For each pixel (i, j), a ray is shot from its center along Z-axis and then the

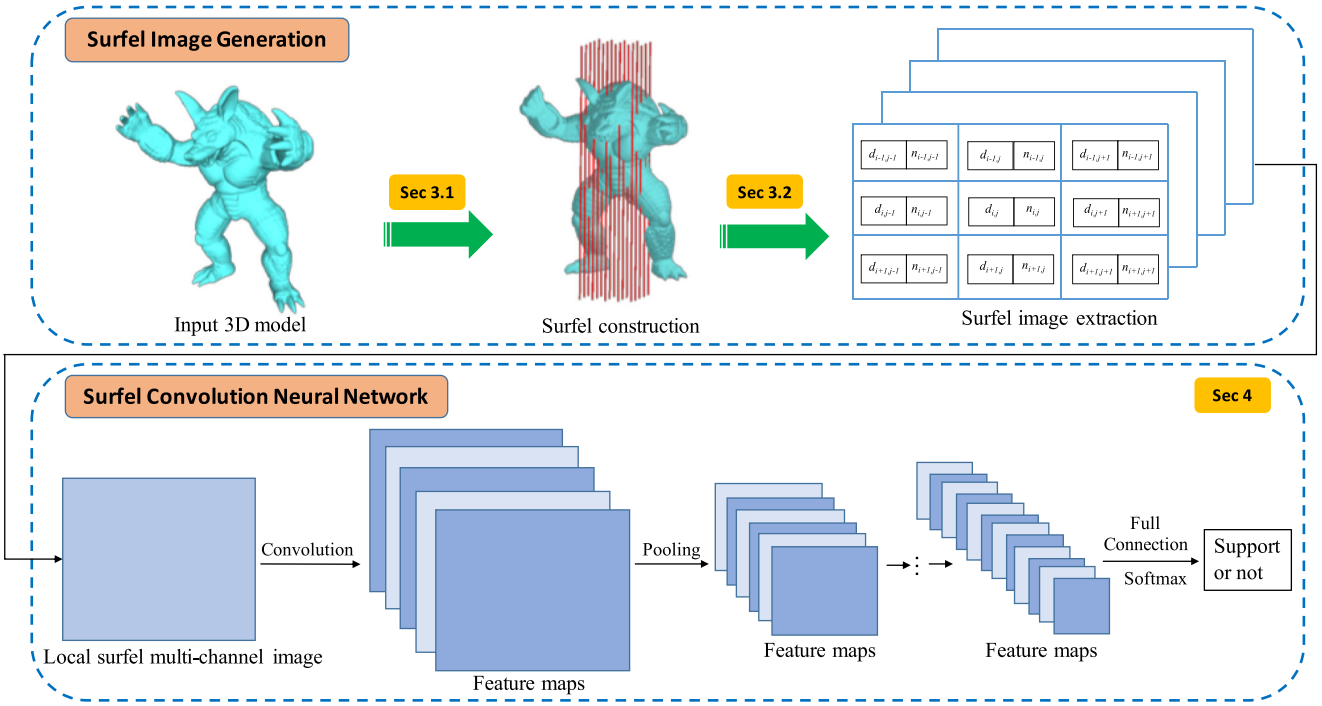


Fig. 3 Flowchart of support learning procedure

intersections of the ray and the surface of the solid model can be calculated. After that, a sequence of intersections (surfels) can be built. Each surfel contains four components: (d, n_x, n_y, n_z), where d specifies the depth from a surfel S to the viewing plane (XY plane in the figure), and $N_S(n_x, n_y, n_z)$ is the surface normal at S. According to the normal direction n_z, we can determine whether the ray shoots in or out from the surface, based on which whether a given point is inside or outside the solid model can be determined.

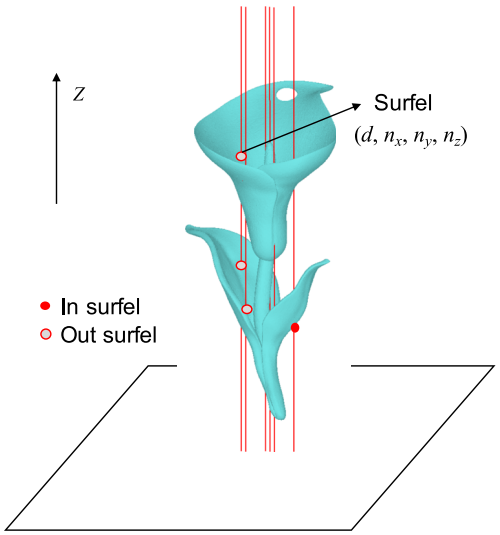


Fig. 4 Surfel construction with LDNI

3.2 Surfel image extraction

The surfel and its neighboring topology information will be extracted after the LDNI sampling is conducted. As illustrated in Fig. 5, for a given surfel (the red diamond marker), its neighboring surfels can be easily extracted from the neighboring rays. Since whether a surfel needs a support is determined by the above topology, we choose the nearest surfels above the given one on the neighboring rays. In this paper, we use such neighboring surfel information to represent the local topology of a sampling point. Since the support is determined by a local area, a sufficient number of neighboring surfels can properly represent the topology information.

Based on each surfel and its neighboring surfels, we can retrieve the local topology of the surfel based on their depth value and the normal information. For the sake of computational simplicity, we store such information into a multi-channel image. For each pixel of the image, the depth value of a surfel and its normal is stored. The whole image represents the local topology of the center surfel. The illustration of such extracted local surfel image is shown in Fig. 4. A 3×3 surfel image is used as an example to illustrate the concept of surfel multi-channel image. In practice, the image size can be chosen according to the self-support capability and other factors. It can be seen from Fig. 4, for a given surfel, its depth value and normal information (d_i, n_i) is recorded at the center pixel of the image, and

the information of its neighboring surfels can be retrieved through looking up the LDNI data.

Since each pixel represents a set of samples on a ray shooting through the solid model, it is possible that there are no surfels located at a neighboring pixel if the ray has no intersections with the model. Consideration of the empty pixels will reduce the computational efficiency. What's more, the support status is also related to the slice height. In order to integrate the slice information into the surfel image, the depth value needs to be unified. To address these issues and depict the surfel information precisely related to the support detection problem, a local surfel image extraction algorithm is proposed and shown in Algorithm 1.

Algorithm 1: Local surfel image extraction.

```
Input  : LDNI image representation of a solid model,
         slice thickness t, local image size s
Output: Local surfel images
Calculate the height of each slice based on t;
for All pixels in the LDNI image do
    if Number of surfels in current pixel > 0 then
        for All surfels in current pixel do
            Determine the surfel height h_s and its slice
            interval [s_k, s_{k+1}];
            for All pixels in I do
                Find the nearest surfel above h_s;
                Calculate the height difference
                d = h_s − s_k;
                Calculate the regulated depth d_reg;
                Store the depth value d and surfel
                normal in this pixel;
            end
        end
    end
end
```

In practice, the depth value ranges from millimeter to inches. In order to eliminate the height variance effect, we applied a depth regulation function on the depth value.

$$d_{reg} = \ln(1 + d) \tag{1}$$

Through Algorithm 1, we extract a local multi-channel image for each surfel which represents its local topology information in the solid model. By using such representation, the local topology information of a surfel and its neighboring surfels can be represented concisely, which can be seamlessly integrated with the state-of-the-art deep learning framework.

3.3 Support learning pipeline

By passing the extracted local surfel image as the input to the deep neural network, we expect to learn the underlying intrinsic properties of the local topology and the status of whether a support is needed at the surfel. The next step is to provide the ground-truth support result for a surfel to guide the training process for the network. As discussed in the introduction section, the image-based method may fail to detect a position in some cases due to topology information missing caused by the offsetting and boolean operations on the images.

For ground-truth surfel support, firstly, the image-based method in [7] is used to determine the surfel support. Then based on the practical printing requirement, we correct the failure cases by removing unwanted supports and adjusting its positions to ensure the physical printing. After all of the ground-truth support positions are determined, it is passed into the network as ground-truth output. Through the deep neural network, the relation between a local surfel topology and its support status is learned. The training network architecture is introduced in the next section.

4 Surfel convolutional neural network

In this section, we move to the bottom part of the flowchart in Fig. 3, the learning framework—surfel convolutional neural network (SCNN) for the relation of a local surfel image and support status. The input of the network is a local surfel multi-channel image which represents local topology information of a surfel, and the output layer is the status of the support for that surfel. Different types of layers are included in the network framework, which will be introduced in the following sections.

4.1 SCNN architecture

The SCNN consists of several subsequent layers. SCNN is applied to the local surfel image defined in Section 3 and produces a category output (need support or not). The architecture of the deep network consists of the following different layers.

Convolution(Conv) layer is specified by a certain number of filters, a_{qp}, along with additive biases b per each kernel, and it operates by computing the convolution of the previous layer with each of those filters, afterward adding the biases. Finally, an activation function, σ, will be applied to all of the pixels of the output images. The convolution layer contains PQ filters arranged in banks (P filters in Q bank), each bank corresponds to an output dimension.

$$g_q^{out}(x) = \sigma\left(\sum_{p=1}^{P}\left(f_p^{in} \circledast a_{qp}\right)(x) + b_q\right); \quad p = 1, ..., P; q = 1, ..., Q \tag{2}$$

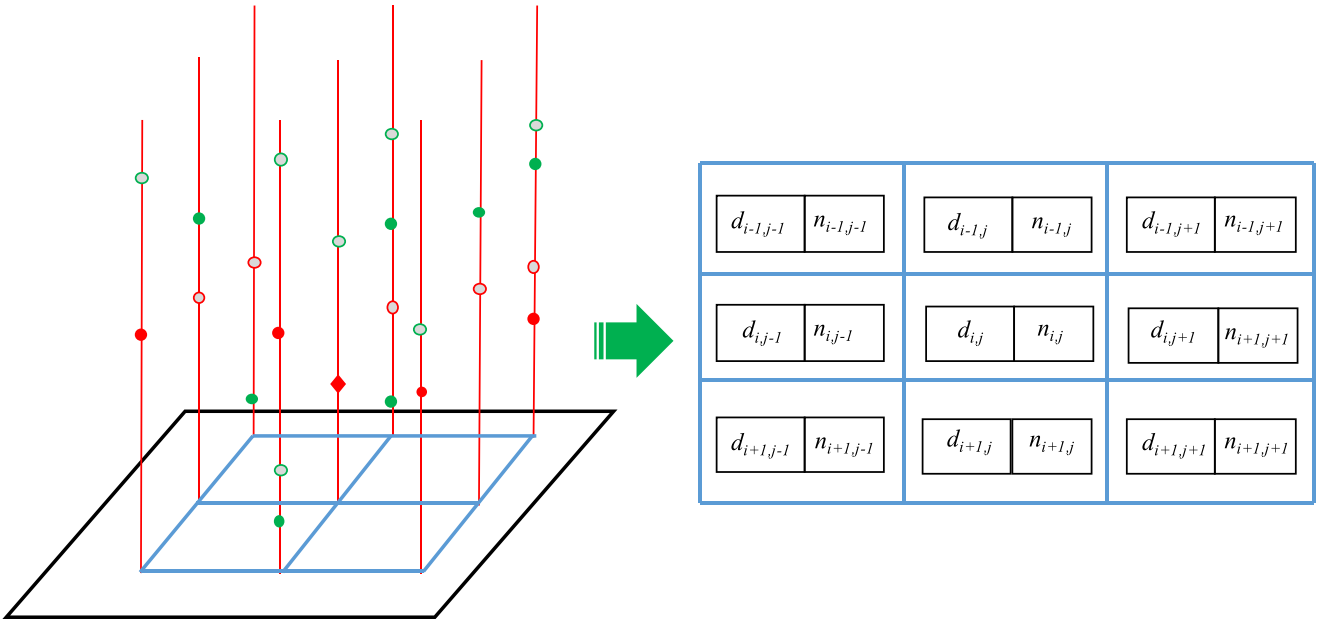


Fig. 5 Local surfel multi-channel image extraction. Left: the red diamond surfel and its local surfel information (Solid is in surfel and empty is out surfel, the red surfel is the nearest one above the given diamond surfel). Right: Surfel topology is stored in a multi-channel image

where a_{qp} is the learnable coefficients of the p^{th} filter in the q^{th} filter bank.

Fully connected (FC) layer is a linearly connected layer to adjust the input and output dimensions. Given a P-dimensional input $X^{in} = (x_1^{in}, ..., x_P^{in})$, the fully connected layer produces a Q-dimensional output $Y^{out} = (y_1^{out}, ..., y_Q^{out})$ by using a learnable weights w,

$$g_q^{out}(x) = \eta\left(\sum_{p=1}^{P} w_{qp} f_p^{in}(x)\right); q = 1, ..., Q \tag{3}$$

The output is optionally passed through a non-linear function such as the ReLU [25], $\eta(t) = max\{0, t\}$.

Softmax layer is used to classify the output from the previous layer,

$$g_p^{out} = softmax(f_p^{in}) = \frac{\exp(f_p^{in})}{\sum_{p=1}^{P} \exp(f_p^{in})} \tag{4}$$

Dropout(π) layer is a fixed layer to prevent over-fitting [11]. It injects binomial noise into each computational unit of the network. During the training stage, an independent and identically distributed binary mask $m_p \sim Binomial(\pi_{drop})$ is generated for each input dimension, each element is 1 with the probability of $1 - \pi_{drop}$,

$$g_p^{out} = m_p f_p^{in} \tag{5}$$

During testing stage, all possible binary masks have to be integrated over. In practice, an approximation method is re-scaling the input by the drop probability of the layer:

$$g_p^{out} = \pi_{drop} f_p^{in} \tag{6}$$

Pooling layer is also a fixed layer which combines the outputs of neuron clusters at one layer into a single neuron in the next layer; it is used to reduce the input dimension.

Batch normalization layer is another fixed layer to reduce the training time of a large network [14]. It normalizes each mini-batch during stochastic optimization to have zero mean and unit variance, and then performs a linear transformation of the form:

$$g_p^{out} = \frac{f_p^{in} - \mu}{\sqrt{\sigma^2 + \varepsilon}}\gamma + \beta \tag{7}$$

where μ and σ^2 are the mean and the variance of the training dataset by using exponential moving average. To avoid numerical errors, a small positive constant ε is used.

4.2 Learning surfel support

For a surfel x on a solid model, the output of SCNN $f_\Theta(x)$ is a binary classification which is interpreted as the support status. Let $y^*(x)$ denote the ground-truth support of surfel x, and samples of surfel and their ground truth support are collected: $\mathcal{T} = \{(x, y^*(x))\}$, the optimal parameters of the

network were determined by minimizing the cross-entropy loss:

$$\ell(\Theta) = -\sum_{(x,y^*(x))\in\mathcal{T}} y^*(x)\log f_\Theta(x) + (1-y^*(x))\log(1-f_\Theta(x)) \quad (8)$$

5 Experimental results

In this section, we evaluate the performance of the proposed SCNN method for support detection problem in additive manufacturing. For experimental settings, the LDNI of the solid model is computed using the method described in [8, 30, 31]. The image resolution is set as 1024×768. For local surfel image generation, the slice thickness is 0.04 mm, and LDNI sampling size is set as $0.1mm \times 0.1mm$.

The convolutional neural network is implemented in Keras. The ADAM stochastic optimization algorithm [15] is used with initial learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. As the input of the network, the two-channel images for the surfels are generated by using Algorithm 1 from the given 3D models. The ground-truth support positions are determined through image-based method followed by the post-interactive correction, fine-tuning and physical testing. Here, a home-made SLA printer is used for physical testing, it should be noted that the proposed method is general and not limited to a specific printing process since the ground-truth supports position can be pre-defined according to certain printing process (FDM, SLS, etc.). For all experiments, training is done by minimizing the loss function in Eq. 8.

5.1 Single batch model support detection

In this experiment, a set of models belonging to the same category is studied. The models in this batch are similar but different from each other. The purpose of the experiment is to test the leaning capability of the network, that is, whether the proposed method can distinguish similar models with local deformations. Figure 6 shows such a batch of teeth aligners used for training.

Table 1 Effect of input surfel image size on computation time and accuracy

Input surfel image size	Training time (min)	Support prediction (sec)	Prediction accuracy (%)
5×5	*11.2*	*0.75*	98.2
10×10	16.3	1.20	98.5
20×20	27.8	2.34	*98.7*

In this experiment, the following network architecture is used: Conv64+Conv32+MaxPooling+Dropout+FC64+FC32+softmax. In all convolution layers, the 3×3 filters are used. In pooling layer, a 2×2 pool size is used. For each teeth aligner model, around 100K surfels with down-facing normals are extracted; thus, around 100K surfel images are extracted for each model. Ten teeth aligner models are studied in the experiment, which provides us around one million surfels (samples) in total for deep learning process. In this study, 75% of the samples are used for training, and the remaining 25% is used for testing. Table 1 is the learning results by using the proposed SCNN method.

As can be seen from Table 1 that with different input image size to represent the local topology of a surfel, the trained model has different prediction accuracy. The larger the input image size, the higher accuracy of the prediction. Specifically, 20×20 input size obtained 98.7% of prediction accuracy; however, it needs the longest training time (27.8 minutes) to train the network. But the difference of the accuracy among the three input size is very small, a 5×5 input surfel image can achieve a 98.2% prediction accuracy. This is mainly because the support of a surfel is determined by the local topology of surrounding surfels on the model, a 5×5 input surfel image is enough to represent this neighboring topology information. Hence, a 5×5 input image size is good enough for a support detection.

Once the network is trained, the prediction process is fairly fast. Table 1 shows that it only takes 0.75 to 2.34 s to predict all of the supports. While it takes about 20 min

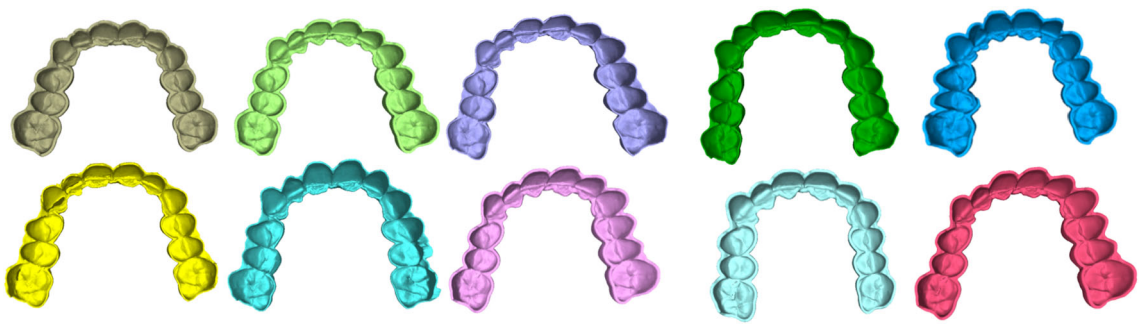


Fig. 6 A batch of teeth aligner models used for training

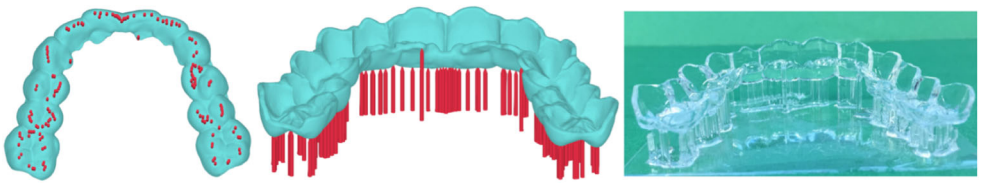


Fig. 7 The prediction of support using the proposed SCNN and the physical fabricated model with the predicted supports

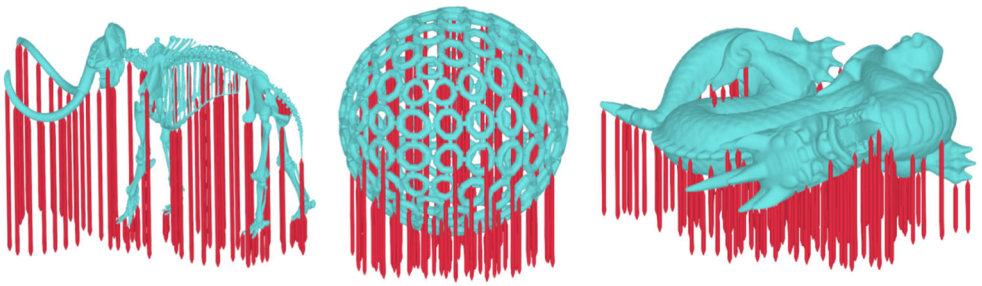
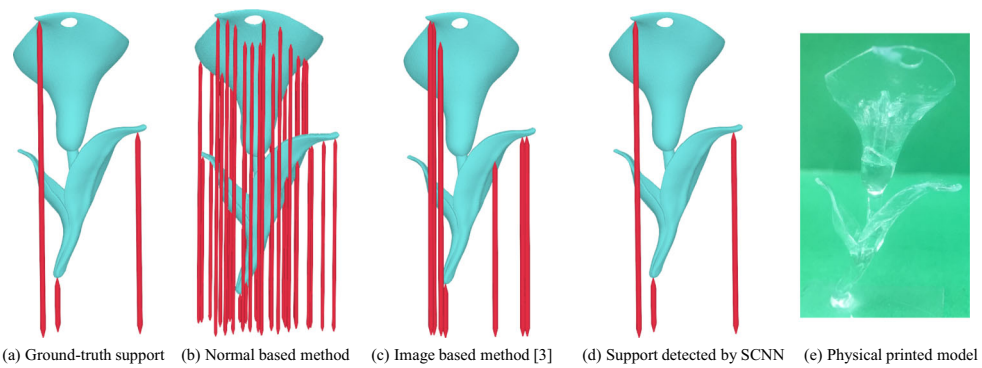


Fig. 8 3D models and their ground-truth supports used for training

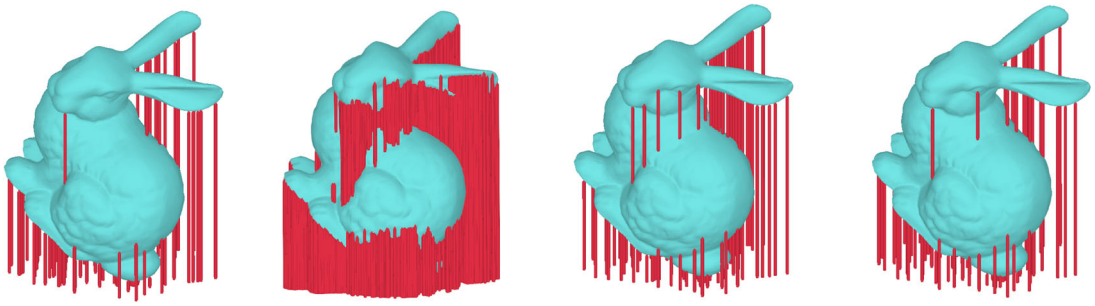
Table 2 Comparison of support detection for different methods

Model	Number of vertices	Normal-based accuracy	Image-based accuracy	SCNN prediction accuracy	Normal-based method (sec)	Image-based method (sec)	SCNN prediction (sec)
Teeth aligner	120K	53.2%	95.2%	*96.4%*	*0.5*	1232.4	1.2
Hearing aid	163K	46.5%	92.0%	*97.3%*	*0.7*	1458.5	1.7
Flower	297K	35.1%	82.3%	*95.8%*	*0.8*	3970.3	2.9
Armadillo	43K	42.8%	97.8%	*98.1%*	*0.2*	53.6	0.9
Bunny	35K	60.4%	97.4%	*98.5%*	*0.2*	38.2	0.5



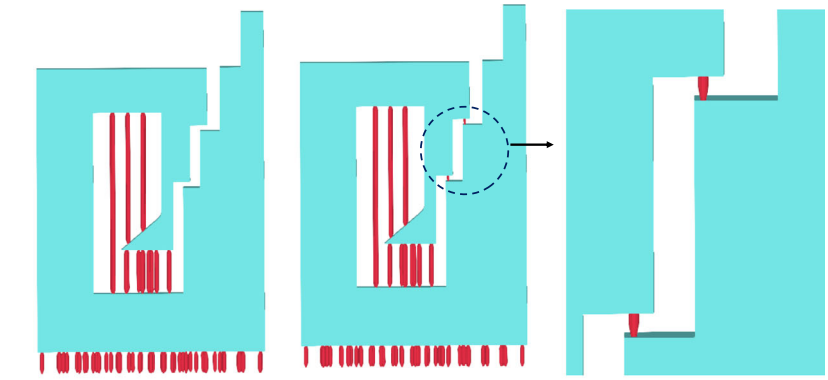
(a) Ground-truth support (b) Normal based method (c) Image based method [3] (d) Support detected by SCNN (e) Physical printed model

Fig. 9 Support positions of flower model detected by different methods



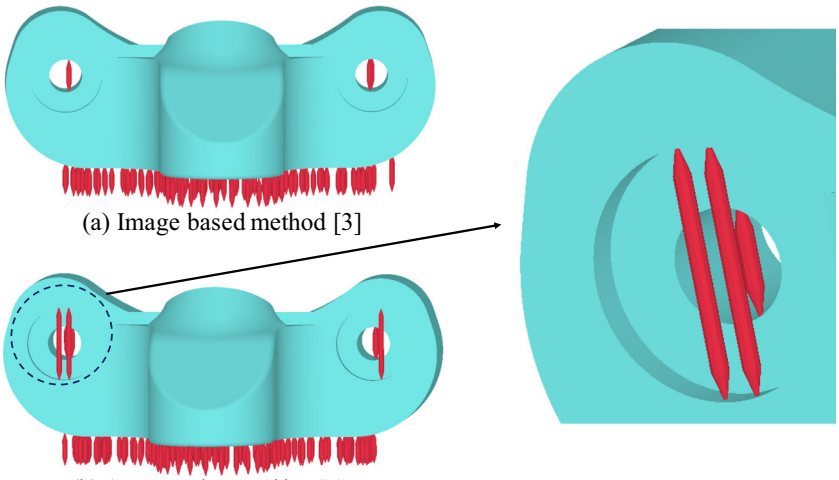
(a) Ground-truth support (b) Normal based method (c) Image based method [3] (d) Support detected by SCNN

Fig. 10 Support positions of bunny model detected by different methods



(a) Image based method [3] (b) Support detected by SCNN

Case 1



(a) Image based method [3]

(b) Support detected by SCNN

Case 2

Fig. 11 Support detection of applications with different methods

for detecting all of the support anchors by the image-based method. Hence, the proposed method is more efficient especially for the applications where a large set of models need to detect support structures. Figure 7 shows the support prediction results by using the trained model, through physical fabrication we can see the predicted supports can successfully withstand the model. The experimental results demonstrate the effectiveness of the proposed method.

5.2 Cross-model support detection

In order to test the effectiveness of the proposed learning method for cross-model support detection, a cross-model support detection experiment is conducted in this section. In this experiment, each model has different topology and features, the leaning capability of the proposed method can be tested. A set of models in different categories and their ground-truth support positions are collected to train and test the network.

For this experiment, the following network architecture is used: Conv128 + Conv64 + MaxPooling + Dropout + FC128 + FC64 + softmax. In the first and second convolution layers, the 4×4 filters and 3×3 filters are used respectively. In the pooling layer, a 2×2 pool size is used. The input image size is set as 10×10. In total, twenty shapes are used for training, and Fig. 8 shows the examples of the training shapes and their ground-truth support positions.

Table 2 shows the comparison of the proposed SCNN with other methods for support detection in terms of prediction accuracy and computational time. It can be seen that the proposed method achieves better performance on the support detection in this cross-model experiment. Compared to the normal-based method and image-based method, the SCNN accomplishes the highest prediction accuracy. For example, the teeth aligner model is used to test the performance of the trained model. In the test, the teeth aligner has 94064 down-facing normals, and the trained model predicts 90678 surfels with an accuracy of 96.4%.

Figures 9 and 10 show examples of support detection with the comparison of different methods, from which we can see that the result of SCNN is closest to the ground truth support. The normal-based method detects a large amount of unnecessary positions. While the image-based method detects fewer supports; however, with post-interactive design, the ground-truth supports have a certain number of adjusted positions, and the SCNN can learn such information and predict a better result. This shows that the proposed SCNN can learn not only the relation between surfel topology and the support status, but also the user interaction design preference, which reveals the proposed method is more effective than other two methods.

Table 2 shows the comparison of the statistical time of support detection for various shapes. It can be seen that the normal-based method has least computational time; however, the detection accuracy is poor due to a large number of unnecessary support positions detected. Compared to the image-based method, the SCNN prediction of support positions is much faster with higher accuracy. Since image-based method needs to detect support positions layer-by-layer, while the SCNN only needs to predict with a trained network; hence, for a trained network, the proposed method is much more efficient for practical support detection especially for mass production.

5.3 Robustness verification

It can be seen from Sections 5.1 and 5.2 that the proposed SCNN outperforms the normal-based method and image-based method. In the real application, a robust support detection method is critical. Hence, in this section, the robustness of the proposed method is tested. We test the performance of the proposed method on some extreme features where image-based method always failed to detect the support. Figure 11 shows the experimental result of such typical cases.

As can be seen from Fig. 11, the image-based method is failed to detect the small overhang feature between small gaps in case 1. The main reason is that the offset operation of an image on the right portion will merge with the left and then lose the topology information of the gap; however, the SCNN method can predict the supports on such a feature by its topology preserving capability. For case 2, the image-based method failed to detect the supports at protruded holes. This is because the offsetting operation of image assumes the position is covered; however, in practical printing, this position is salient and a support should be added. The SCNN can predict correct support positions due to its learning capability can extract the salient topology features and the support status.

From these two cases, the experimental results reveal that the proposed SCNN is more robust for support detection on extreme features.

6 Conclusions

In this paper, a deep learning technique is applied to the problem of support detection in the additive manufacturing area. The support detection problem is investigated as a sub-problem of support generation. The issues of current methods are analyzed and discussed, then the deep learning-based method for support detection is proposed. In this method, a richer representation for local topology information surfel image is developed, and a surfel convolutional neural network is constructed for support learning. Experimental results show that the proposed method achieves high prediction accuracy and outperforms previous methods in terms of support detection. With the learning capability of the deep neural network, the interactive design information can be extracted. To the best of our knowledge, this is the first time deep learning method is introduced in support generation in additive manufacturing.

Future work would be on the following two aspects. Firstly, the support structure generation could be incorporated with deep learning technique, as the application of such machine learning technique can help the designer to embrace intelligence knowledge to achieve automotive modeling. Secondly, the deep learning techniques can be used for other prefabrication and fabrication applications in the additive manufacturing field.

Funding information This study received financial support from the National Science Foundation (NSF) # CNS-1547167 and Natural Sciences & Engineering Research Council of Canada (NSERC) grant # RGPIN-2017-06707.

References

1. Balu A, Lore KG, Young G, Krishnamurthy A, Sarkar S (2016) A deep 3d convolutional neural network based design for manufacturability framework. arXiv:1612.02141
2. Boscaini D, Masci J, Rodoià E, Bronstein M (2016) Learning shape correspondence with anisotropic convolutional neural networks. In: Proceedings of the 30th international conference on neural information processing systems, Curran Associates Inc., USA, NIPS'16, pp 3197–3205
3. Boyard N, Christmann O, Rivette M, Kerbrat O, Richir S (2018) Support optimization for additive manufacturing: application to fdm. Rapid Prototyp J 24(1):69–79
4. Brock A, Lim T, Ritchie JM, Weston N (2016) Generative and discriminative voxel modeling with convolutional neural networks. arXiv:1608.04236

5. Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2016) Geometric deep learning: going beyond euclidean data. arXiv:1611.08097
6. Calignano F (2014) Design optimization of supports for overhanging structures in aluminum and titanium alloys by selective laser melting. Mater Des 64:203–213
7. Chen Y, Li K, Qian X (2013) Direct geometry processing for telefabrication. J Comput Inf Sci Eng 13(4):041002
8. Chen Y, Wang CC (2008) Layered depth-normal images for complex geometries: part one: accurate sampling and adaptive modeling. In: ASME 2008 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers, pp 717–728
9. Dumas J, Hergel J, Lefebvre S (2014) Bridging the gap: automated steady scaffoldings for 3d printing. ACM Trans Graph (TOG) 33(4):98
10. Ezair B, Massarwi F, Elber G (2015) Orientation analysis of 3d objects toward minimal support volume in 3d-printing. Comput Graph 51:117–124. International Conference Shape Modeling International
11. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:12070580
12. Hu K, Jin S, Wang CC (2015) Support slimming for single material based additive manufacturing. Comput Aided Des 65:1–10
13. Huang X, Ye C, Wu S, Guo K, Mo J (2008) Sloping wall structure support generation for fused deposition modeling. Int J Adv Manuf Technol 42(11):1074
14. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:150203167
15. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
16. Kirschman C, Jara-Almonte C, Bagchi A, Dooley R, Ogale A (1991) Computer aided design of support structures for stereolithographic components. In: Proceedings of the 1991 ASME computers in engineering conference, Santa Clara, CA, pp 443–448
17. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th international conference on neural information processing systems - Volume 1, Curran Associates Inc., USA, NIPS'12, pp 1097–1105
18. Kwok TH, Ye H, Chen Y, Zhou C, Xu W (2017) Mass customization: reuse of digital slicing for additive manufacturing. J Comput Inf Sci Eng 17(2):021009
19. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
20. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436
21. Maron H, Galun M, Aigerman N, Trope M, Dym N, Yumer E, Kim VG, Lipman Y (2017) Convolutional neural networks on surfaces via seamless toric covers. ACM Trans Graph 36(4):71:1–71:10
22. Masci J, Boscaini D, Bronstein MM, Vandergheynst P (2015) Geodesic convolutional neural networks on riemannian manifolds. In: Proceedings of the 2015 IEEE international conference on computer vision workshop (ICCVW). IEEE Computer Society, Washington, pp 832–840. ICCVW '15
23. Mehta P, Schwab DJ (2014) An exact mapping between the variational renormalization group and deep learning. arXiv:1410.3831
24. Monti F, Boscaini D, Masci J, Rodolà E, Svoboda J, Bronstein MM (2016) Geometric deep learning on graphs and manifolds using mixture model cnns. arXiv:1611.08402
25. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on international conference on machine learning, ICML'10
26. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9
27. Tygert M, Bruna J, Chintala S, LeCun Y, Piantino S, Szlam A (2016) A mathematical motivation for complex-valued convolutional networks. Neural Comput 28(5):815–825
28. Vaidya R, Anand S (2016) Optimum support structure generation for additive manufacturing using unit cell structures and support removal constraint. Procedia Manufacturing 5:1043–1059. 44th North American manufacturing research conference, NAMRC 44, June 27-July 1, 2016 Blacksburg, Virginia, United States
29. Vanek J, Galicia JAG, Benes B (2014) Clever support: efficient support structure generation for digital fabrication. Computer graphics forum, Wiley Online Library 33:117-125
30. Wang CCL, Leung YS, Chen Y (2010) Solid modeling of polyhedral objects by layered depth-normal images on the gpu. Comput Aided Des 42(6):535–544
31. Wang CC, Chen Y (2008) Layered depth-normal images for complex geometries: part two—manifold-preserved adaptive contouring. In: ASME 2008 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers, pp 729–739
32. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3d shapenets: a deep representation for volumetric shapes. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1912–1920