# Perceptual Models of Preference in 3D Printing Direction

Xiaoting Zhang[1] Xinyi Le[1*] Athina Panotopoulou[2*] Emily Whiting[2] Charlie C.L. Wang[1]

[1]The Chinese University of Hong Kong [2]Dartmouth College

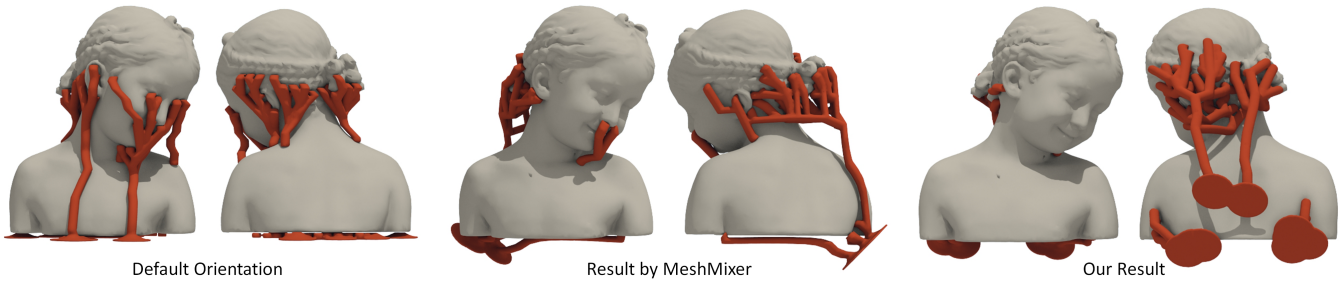


**Figure 1:** *Optimization of printing directions has mainly focused on structural stability and material cost. We develop a perceptual model to find optimal printing directions that preserve important visual features. In comparison to alternative methods, our technique avoids placing support structures on important regions to prevent damage in the final printed model. (Left) Support structures resulting from default orientation. (Middle) Result by Autodesk MeshMixer. (Right) Our result.*

## Abstract

This paper introduces a perceptual model for determining 3D printing orientations. Additive manufacturing methods involving low-cost 3D printers often require robust branching support structures to prevent material collapse at overhangs. Although the designed shape can successfully be made by adding supports, residual material remains at the contact points after the supports have been removed, resulting in unsightly surface artifacts. Moreover, fine surface details on the fabricated model can easily be damaged while removing supports. To prevent the visual impact of these artifacts, we present a method to find printing directions that avoid placing supports in perceptually significant regions. Our model for preference in 3D printing direction is formulated as a combination of metrics including area of support, visual saliency, preferred viewpoint and smoothness preservation. We develop a training-and-learning methodology to obtain a closed-form solution for our perceptual model and perform a large-scale study. We demonstrate the performance of this perceptual model on both natural and man-made objects.

**CR Categories:** I.3.5 [Computer Graphics]: Shape Modeling—Shape Analysis; J.6 [Computer-Aided Engineering]: Computer-aided design (CAD)

**Keywords:** orientation, human preference, perceptual model, supporting structure, 3D printing

*joint second author

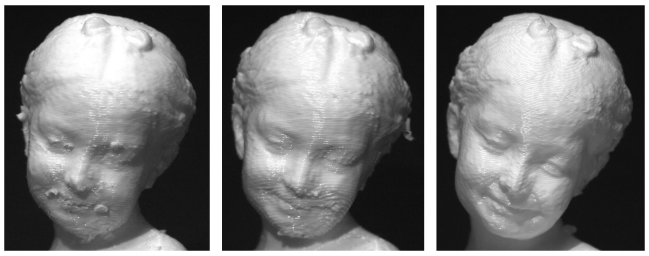

**Figure 2:** *Photographs of 3D printed model comparing artifacts from different support structure placement. (Left) Default orientation. (Middle) Autodesk MeshMixer. (Right) Our result.*

## 1 Introduction

3D printers have become widely used in small-scale prototyping and independent creative projects due to their flexibility and low cost. A crucial step in the 3D fabrication pipeline is the placement of support structures for printed objects: material is deposited incrementally in layers which fuse together to create the solid object. As a result of this bottom-up procedure, overhanging geometry requires additional supporting structures to be printed along with the object to prevent collapse caused by gravity – such additional material is referred to as *supporting structures* (or simply called *support*). An often ignored effect is that supports can lead to surface artifacts which damage the visual quality of the object. Unlike industrial machines for additive manufacturing which can use dissolvable materials to make supporting structures, many home-use 3D printers can only extrude a single type of material. Supporting structures have to be removed manually as a post-processing step after printing, introducing two damaging outcomes:

- *Visual Artifacts*: Even after support removal, anchor points remain at regions contacted by supporting structures. These artifacts can significantly affect the appearance of a printed model (see Figure 2).
- *Damage on small features*: When anchors of support are linked to small features, support removal becomes challenging as geometric details can be easily damaged.

Many automatic techniques exist for generating support structures, however these aim to optimize criteria focused on efficiency rather than appearance. Recent approaches optimize either the topology of support [Dumas et al. 2014; Vanek et al. 2014; Wang et al. 2013] or the shape of input model [Hu et al. 2015] to reduce the volume of support. Considerations of whether appearance distortion will occur is left to case-by-case assessment by the user usually followed by manual adjustment.

The volume and positioning of the supporting material depends on the chosen print orientation of the object. Determining a good printing direction is complex due to the many factors under consideration. For example, Autodesk® MeshMixer™ provides a semi-automatic orientation optimization tool to minimize support volume, support area, structural strength, or a combination of these three attributes. Users can weight each attribute according to their objective. Prior approaches on determining optimal printing directions all focus on similar criteria (e.g., [Cheng et al. 1995; Majhi et al. 1998; Padhye and Deb 2011; Hildebrand et al. 2013]). None of them studied the problem of different appearance on a model printed in different directions (especially the effectiveness of remained anchor dots after removing supporting structures). As the appearance of a printed model is a complex problem relating to multiple factors, we propose a perceptual model in this paper to find good printing directions. Training-and-learning methodology is developed to construct the model. Results have been validated on a variety of models.

We make three main contributions:

1. We introduce a perceptual model for finding preferable printing directions using a training-and-learning approach. To the best of our knowledge, this is the first approach using perceptual metrics to determine good orientations in 3D printing.
2. We conduct a study to determine relative importance of factors affecting appearance of 3D printed models after removing support. Our model is formulated as a combination of known metrics, including area of support, visual saliency, preferred viewpoints and smoothness preservation. Our methods for efficient evaluation are presented.
3. We introduce a *Double-Layered Extreme Learning Machine* (DL-ELM) to train the perceptual model. Our DL-ELM method is able to convert pairwise training sets into a closed-form formulation for evaluating the scores of a model printed in a range of orientations.

Our approach can be applied to different types of single material based 3D printing techniques, including Fused Deposition Modeling (FDM) and Stereolithographic (SLA) resin printing.

## 2 Related Work

Methods for generating support structures to prevent material collapse during layered fabrication have been studied for more than two decades [Sachs et al. 1992; Kulkarni et al. 2000; Levya et al. 2003; Liu et al. 2014; Strano et al. 2013]. Early work originates mainly in the CAD/CAM community to generate supports by deterministic algorithms. When dissolvable materials are used, variants of a region subtraction algorithm [Chalasani et al. 1995] are usually conducted to generate support structures. For FDM and SLA using single materials, supports are connected to the main surface by small pyramidal volumes, where methods similar to [Chen et al. 2013] are used in commercial systems.

In recent years, the problem of optimizing support structures has been studied for a variety of objectives, for example, to increase stability [Wang et al. 2013], decrease material, and improve print speed [Alexander et al. 1998; Dumas et al. 2014; Vanek et al. 2014]. However, none of these approaches consider the artifacts remaining on the surface of a printed object after removing support structures. To eliminate the usage of supporting structures altogether, segmentation methods have been investigated that subdivide a model into height field pieces, which can also be fabricated by mold casting [Herholz et al. 2015; Hu et al. 2014]. However when the final model is assembled from pieces fabricated separately, gaps and seams may also detract from the appearance.

The appearance of printed models has been considered previously in [Wang et al. 2015] together with printing time, but this approach was limited to adjusting layer thickness and segmenting the input model into subparts. Other approaches focus more on reflectance results of printed models, e.g., [Dong et al. 2010; Papas et al. 2013; Lan et al. 2013; Schüller et al. 2014]. In OpenFab [Vidimče et al. 2013], a programmable pipeline is proposed for synthesizing multi-materials to generate different appearances on 3D printed objects. These methods do not consider damage to surface quality due to support removal which is the problem to be solved here.

Our work is inspired by Secord et al. [2011], where optimal *viewpoints* were found for 3D models according to perceptual metrics. While a large variety of metrics exist for evaluating the goodness of views, Secord et al. [2011] conduct a learning-based method to find the combination of attributes that match human preference. The resulting perceptual model is formulated as a linear combination of attributes by weights obtained from the pair-based training. We apply their result as a central component of our model. However, the application domain has fundamental differences: fabricated models are not limited to a static view, and we are considering damage to the surface. Notably, we observe a non-linear nature in the metrics for finding good printing directions preferred by users. Therefore, a more advanced learning method – DL-ELM is investigated in our work. Our contribution is to find a representation of the weightings for perceptual metrics which encapsulate properties unique to fabricated 3D models.

Artificial neural network methods have been widely used in many engineering problems, where neural networks can map the relations between the input data and output performance directly [Cochocki and Unbehauen 1993]. *Support Vector Machine* (SVM) and its extension – *Least Square Support Vector Machine* (LS-SVM) are two important techniques for training and optimization. However, [Chorowski et al. 2014] has shown that both SVM and LS-SVM face many challenges, such as slow learning speed, intensive human intervention, etc. Recently, *Extreme Learning Machine* (ELM) [Huang et al. 2015] aiming at overcoming these challenges has received much attention. The universal approximation capability and classification capability of ELM has been shown by Huang et al. [2006a; 2012]. The key feature of ELM is that the hidden neurons of generalized feedforward networks do not need to be tuned and can be generated randomly, which is considered more closely analogous to biological learning in animal brains. As a result, it can achieve much faster learning speed. ELM learning has been used in [Xie et al. 2014; Xie et al. 2015] for fast 3D segmentation and labeling. In our perceptual model, the ELM learning is further extended to a DL-ELM based training-and-learning methodology, where ELM classification is used in the first layer to evaluate relative-score between different printing directions and ELM regression is investigated in the second layer to construct a global score for all printing directions.

## 3 Metrics

In this section we describe the metrics constituting our measure of goodness for printing directions. Contrary to previous work focus-

ing on efficiency objectives, such as material cost and print time which may lead to poor feature preservation (Figs. 1,2), we analyze factors relating to appearance. We assess properties of contact area, visual saliency, viewpoint preference, and smoothness entropy.

$M_1$: **Contact area**. We consider the total surface area of regions connecting to supporting structures, $A_s$. Given a printing direction $\mathbf{d}$, the total area of contact $A_s(\mathbf{d})$ is determined by considering overhangs as well as potential connections at the base of supports. Many existing approaches calculate overhangs by detecting facing-down regions (e.g., [Vanek et al. 2014; Hu et al. 2015]), however, this is not representative. Surface regions need support only if the angle $\alpha_1$ between its tangent plane and the printing direction $\mathbf{d}$ is larger than the critical angle $\alpha_{\max}$ (red regions in Fig. 3). Faces with angle $\alpha_2 < \alpha_{\max}$ are self-supported (green regions in Fig. 3). Moreover, supporting structures can also attach to up-facing regions when "shadowed" by overhangs (blue regions in Fig. 3). We develop an efficient method to compute $A_s$ using *Layered Depth Images* (LDI), which can be accelerated by graphics hardware [Leung and Wang 2013; Faure et al. 2008]. Details are presented in Section 6.1. A distribution of $A_s(\mathbf{d})$ can be computed and visualized on the Gauss sphere (see Fig. 5). The support contacting region is denoted by $\mathcal{T}(\mathbf{d})$ in the rest of our paper, which corresponds to surface area $A_s$ and depends on the printing direction $\mathbf{d}$. Note that our study uses the same self-supporting angle as the default in MeshMixer.

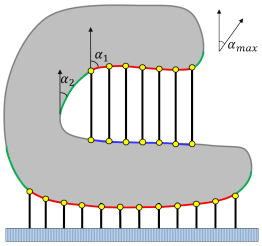


**Figure 3:** *Contact area consists of both the red and the blue regions. Green regions are self-supported.*

$M_2$: **Visual saliency**. We compute mesh saliency of the input mesh model at each vertex by the method of Lee et al. [2005]. The metric is a scalar quantity indicating the importance of a local region in low-level human visual attention. Let $\chi(x)$ denote the mesh saliency of a point $x$ on the input model, the metric $M_2$ is defined as the integral of mesh saliency of all points in the support contacting region, $\mathcal{T}(\mathbf{d})$.

$$M_2(\mathbf{d}) = \frac{1}{A_s}\int_{x\in\mathcal{T}(\mathbf{d})} \chi(x)d\mathbf{A} \tag{1}$$

$M_3$: **Viewpoint preference**. This metric measures the goodness of a viewpoint based on the perceptual model introduced by Secord et al. [2011]. The five most dominant attributes are employed in our approach, which include projected area ($a_1$), surface visibility ($a_2$), silhouette length ($a_4$), max depth ($a_7$) and above preference ($a_{12}$). The final score of $M_3$ is obtained by linear blending of these five attributes using the weights provided [Secord et al. 2011]. An interactive tool is also developed to further incorporate the viewing preference of a user into this metric (see Section 6.2).

$M_4$: **Smoothness entropy**. To prevent adding anchor dots at smooth regions where they will be highly visible, the metric of smoothness entropy is also introduced in our model. For a point $x$ on the surface $S$ of an input model, define $\mathcal{N}(x)$ as $x$'s Euclidean neighboring region on $S$ and $\hat{\mathbf{n}}(\cdot)$ as the unit normal. The smoothness at $x$ can be evaluated as:

$$\varpi(x) = \inf\{p, q \in \mathcal{N}(x) : \langle \hat{\mathbf{n}}(p), \hat{\mathbf{n}}(q)\rangle\} \tag{2}$$

Given a threshold $\tau$, the region $\mathcal{S} = \{x : \varpi(x) > \tau\}$ is considered smooth. The value of $\tau$ is selected by Otsu's method [Otsu 1979]. Briefly, the histogram of $\varpi$ is computed and $\tau$ is selected at the location that minimizes the weighted within-class variance. This turns out to be the same as maximizing the between-class variance, which means a minimal probability of misclassification. An example can be found in Figure 4. The smaller the overlap between $\mathcal{S}$ and the support contacting area $\mathcal{T}(\mathbf{d})$, the better the appearance of the fabricated model. Thus we have

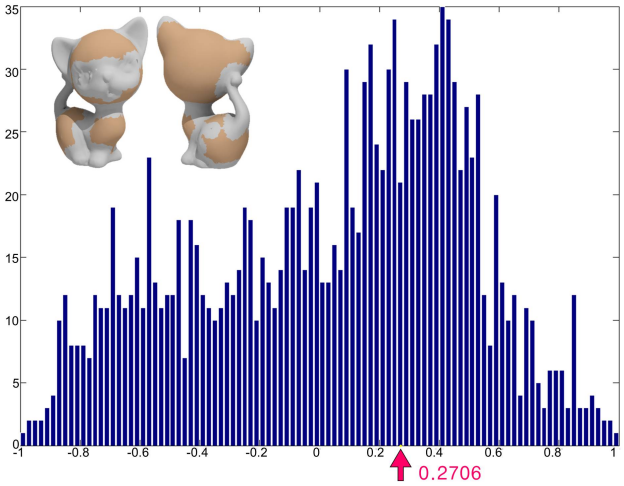


**Figure 4:** *Histogram of $\varpi(x)$ is used to determine the value of $\tau$ by Otsu's method. $\tau = 0.2706$ is computed in this example. The detected smooth region is shown on the model.*

$$M_4(\mathbf{d}) = \frac{1}{A_s}\int_{x\in(\mathcal{S}\cap\mathcal{T}(\mathbf{d}))} dA \tag{3}$$

In the following sections we devise a method to optimize printing direction based on these multiple metrics. Note all metrics are normalized into the range $[0, 1]$ to serve as the input parameters for the perceptual model (Section 4). Details on how to evaluate these metrics efficiently can be found in Section 6.

## 4 Perceptual model

The goodness of a printing direction for a model is commonly determined by the four factors analyzed above. To construct the objective function $F(M_{i,i=1,\ldots,4}(\mathbf{d}))$, we conduct a training-and-learning strategy. A study is first performed to collect data from trainers about the relative goodness of printing directions according to human preferences. Next, the dataset obtained from training is passed to a machine-learning model – i.e., a mathematical formulation of $F(\cdots)$. As a result, the values evaluated from $F(\cdots)$ can be used to determine good printing directions.

### 4.1 Collecting human preferences

The purpose of the training stage in our study is to learn which printing directions are preferred. Ideally we seek a direct quantitative measurement on preference. However, as the amount of data to be trained is large, personal variations will occur, e.g., a score '75' given by a trainer at the end of training may be better than a score '90' given at the beginning. Further, equivalent scores given by two people may have different interpretations. Inspired by Secord et al. [2011], the method of paired comparison is conducted which asks people to select between two printing directions. This is the well-known *2-Alternative Forced Choice* experiment (2AFC). The overall rank for all printing directions is established from many of

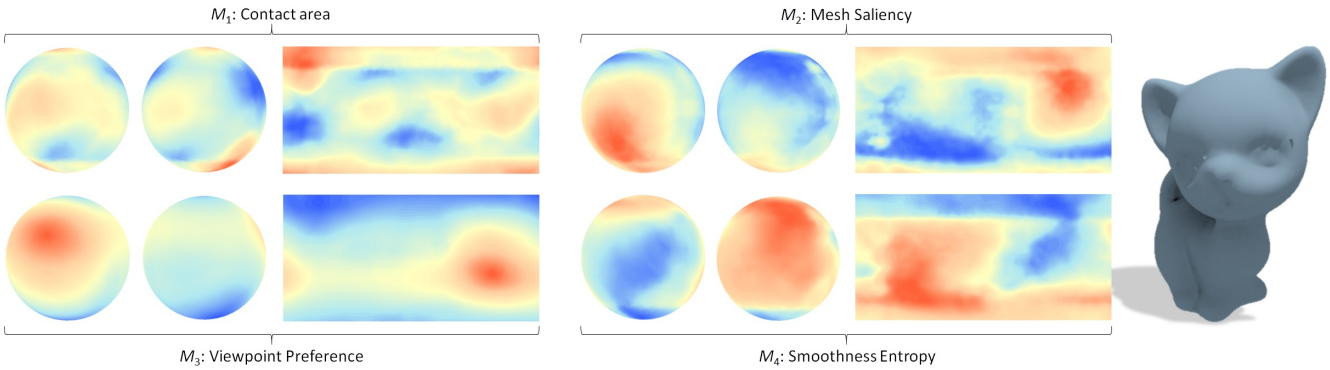


**Figure 5:** *Metrics for measuring four different factors of preference in 3D printing directions. The distribution of each metric is visualized on the Gauss sphere, both the front-view (left) and back-view (right) are displayed. Blue denotes zero and red denotes one (highest preference). To display the overall distribution, the Gauss spheres are also unfolded onto a rectangular region.*

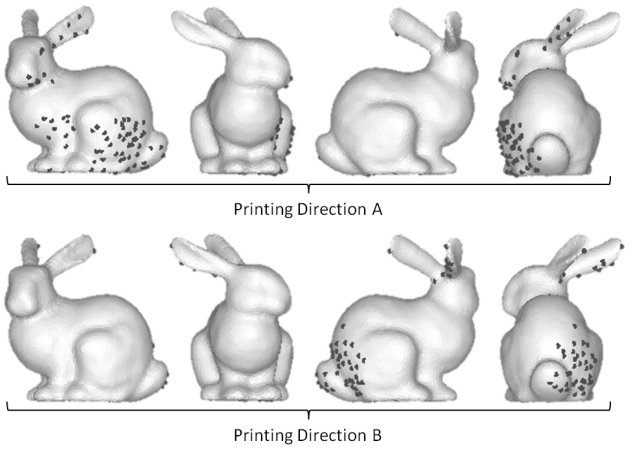


**Figure 6:** *Pairs for comparison in Amazon study. Anchor points are represented as black dots on the model surface. In the study, ten views are provided for each printing direction (only four views displayed here).*

such 2AFC experiments. The service of *Amazon Mechanical Turk* (AMT) is applied in our study. AMT hires anonymous workers (known as *Turker*) to perform tasks (called HITs in AMT) that require human intelligence, which now is heavily used in machine learning research.

Ten models are employed in our study to generate HITs of AMT (see Figure 12). We uniformly sample the Gauss sphere to obtain 1,500 possible printing directions. For each model, 500 pairs of printing directions are randomly picked to generate HITs. Among them, 16 random pairs of each model occur more than once for testing the honesty of Turkers. Nearby orientations can result in supporting structures which are too similar. We therefore exclude the pairs with less than $10^\circ$ difference. To show Turkers the results of printed models after removing supports, we simulate anchor dots on the models. Images for each HIT contain 10 different views for each model in the simulated pair. An example is shown in Figure 6. These images were precomputed in our system and then uploaded to AMT.

Before training, the instructions are provided to each Turker to build up the basic concept about single material based 3D printing and the artificial anchor points left on the model. In addition, an example pair with simulated and physical results is also given to better explain the instructions. The provided instructions are:

- *When 3D printing a complicated shape, printers often require extra structures to support the model. These accessory structures temporarily support overhanging parts of the shape, and need to be manually removed after printing. However, anchor dots (that is, the black dots in images) will be left on the surface of a printed object and affect the appearance.*
- *To minimize the damage caused by supporting structures, a preferred printing direction is selected so that important features of a model can be well preserved.*
- *Criteria of preference selection includes:*
  *1. No dot is located at important regions (e.g. eyes, nose and sharp features); 2. No dot (or only a few) can be found in a human-preferred view.*

Considering the human preference, we hired 8 Turkers for each HIT. Note that, according to the running mechanism of AMT, the number of trainers involved is much larger than that (e.g., 32 in our training). After the training, a total number of $40{,}000 = 8 \times 500 \times 10$ choices are obtained. A trainer who gives more than 4 inconsistent choices among any of the repeated pairs is considered as taking random picks. All his or her choices are excluded from the dataset. As a result 22,134 out of 40,000 choices remained in the 'honest' dataset to be used in our learning practice below.

### 4.2 Linear or nonlinear

The most straightforward formulation to score a problem with multiple factors is to linearly blend the metrics. In our problem,

$$F(\mathbf{d}) = \sum_{i=1}^{4} w_i M_i(\mathbf{d}) \tag{4}$$

with $w_i$ to be determined. From each HIT of the trained dataset given two printing directions $\mathbf{d}_a$ and $\mathbf{d}_b$, the result of training indicates $F(\mathbf{d}_a) > F(\mathbf{d}_b)$ (or $F(\mathbf{d}_a) < F(\mathbf{d}_b)$). Reformulating the above equation, each HIT can be considered as inputting a class label $y$ for the 4-dimensional patterns with

$$\triangle M_{i,i=1,\ldots,4} = (M_i(\mathbf{d}_a) - M_i(\mathbf{d}_b))$$

as the four components for a pattern $\mathbf{x}$. The training dataset with $m$ HITs gives

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m) \in \Re^4 \times \{\pm 1\}. \tag{5}$$

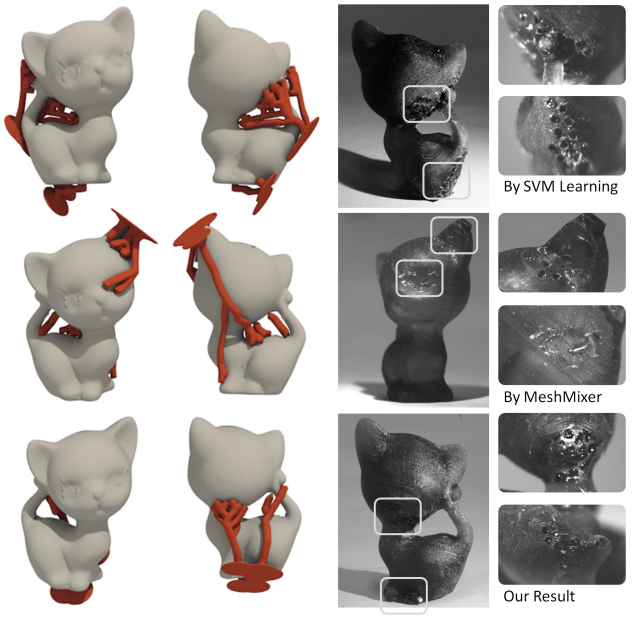


**Figure 7:** *Comparison on the 'best' printing directions determined by a linear SVM learning (top), Autodesk MeshMixer (middle) and our learning model (bottom). Our nonlinear model results in less material cost and better feature preservation.*

Classifiers such as linear *Support Vector Machines* (SVM) can then be used to determine the values of $\mathbf{w} = (w_1, w_2, w_3, w_4)$ via a decision function

$$y(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{4} \mathbf{w} \cdot \mathbf{x} + b\right). \quad (6)$$

The solution can be obtained by solving a quadratic programming problem [Hearst et al. 1998].

Now we study the configuration space spanned by the dataset of AMT training. As visualized in Figure 8, the training set is hard to classify into two groups by a hyperplane:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \ \ (\mathbf{w} \in \Re^4, b \in \Re).$$

Specifically, the function to find the optimized printing direction should be complex and cannot be determined by a linear classification. When using a least-square solution to find the value of $\mathbf{w}$, the resultant $F(\cdots)$ gives poorer results than our DL-ELM method below. Although nonlinear SVM can be used to conduct the classification, it may result in a form with too many support vectors so that the evaluation of $F(\cdots)$ becomes time-consuming. We propose a DL-ELM method in the following section to formulate the nonlinear function $F(\cdots)$. The difference of results obtained from a linear SVM vs. the nonlinear learning model can be found in Figure 7.

## 5 Learning from human preferences

ELM is a learning technique [Huang et al. 2006b] aimed at overcoming the problems of slow learning speed and intensive human intervention in conventional machine learning techniques (e.g., SVM). An ELM neural network is composed of one input layer, one hidden layer and one output layer as illustrated in Figure 9. For a learning problem, nodes in the input layer are entries of the training dataset. All the nodes in the hidden layer are independent of training data, and they are randomly generated. In the output layer, the coefficients, $\beta_i$, are problem based, which are variables to be determined by the training dataset.

Considering $m$ input-output samples $(\mathbf{x}_j, y_j) \in \Re^4 \times \Re$, ELM formulates the output of a single layer feed-forward neural network with $n$ hidden nodes as

$$f(\mathbf{x}) = \sum_{i=1}^{L} \beta_i \phi(\mathbf{a}_i, b_i, \mathbf{x}) \quad (7)$$

where $\mathbf{a}_i$ and $b_i$ are learning parameters of hidden nodes and $\beta_i$ is the weight connecting the $i$-th hidden node to the output node. $\phi(\mathbf{a}_i, b_i, \mathbf{x})$ is the output of the $i$-th hidden node with respect to the input $\mathbf{x}$. There are many choices for activation functions used in ELM, such as sigmoid, sine, hard-limit, radial basis functions, and complex activation functions, etc. Sigmoid activation function is employed here, which gives

$$\phi(\mathbf{a}_i, b_i, \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{a}_i \cdot \mathbf{x} + b_i)}}. \quad (8)$$

If a single layer feed-forward neural network with $L$ hidden nodes can approximate $m$ samples with zero error, it implies that there exist $\beta_i$, $\mathbf{a}_i$, and $b_i$ such that (see Appendix for more details)

$$f(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i \phi(\mathbf{a}_i, b_i, \mathbf{x}_j) = y_j \ \ j = 1, \ldots, m. \quad (9)$$

This ELM neural network is conducted in the rest of this section to model our perceptual model of preference in 3D printing. ELM is first used to give the relative goodness between two printing directions. Next, a DL-ELM method is investigated to score all printing directions.

### 5.1 Pairwise learning by ELM

Suppose the relative goodness between two printing directions, $\mathbf{d}_a$ and $\mathbf{d}_b$, can be evaluated by a function

$$G(\mathbf{d}_a :: \mathbf{d}_b) = G(M_1^a, M_2^a, M_3^a, M_4^a, M_1^b, M_2^b, M_3^b, M_4^b) \quad (10)$$

by using their corresponding metrics as input. Here we denote $M_i(\mathbf{d}_a)$ by $M_i^a$ (also $M_i^b$ similarly). As the printing preference between $\mathbf{d}_a$ and $\mathbf{d}_b$ has been provided by Turkers in the AMT training, the relative score at a pair $(\mathbf{d}_a :: \mathbf{d}_b)$ can be obtained by voting as follows:

- Starting from zero, one preference in $\mathbf{d}_a$ will increase the score by one while one preference in $\mathbf{d}_b$ will decrease one.
- After updating the score by all Turkers at this HIT, the final score is divided by the number of Turkers (i.e., normalizing into the range $[-1, 1]$).

As a result, a higher score indicates that more people prefer the printing direction $\mathbf{d}_a$.

Using the metrics in each of the $m$ training pairs as the pattern vector $\mathbf{x}$ and the score as the target function value $y$, the function $G(\cdots)$ for evaluating the relative score between any pair of printing directions can be learned from the dataset by ELM. Here we have $\mathbf{x} \in \Re^8$. Note that $G(\mathbf{d}_a :: \mathbf{d}_b) = 0$ is enforced when $a = b$. The type of classifier with 15 neurons is used in ELM learning, and the resultant function $G(\cdots)$ will return one of the values $\{-1, 0, 1\}$.

**Finding 'Best':** With the help of the relative goodness function, we can now search for the most preferred printing direction on a

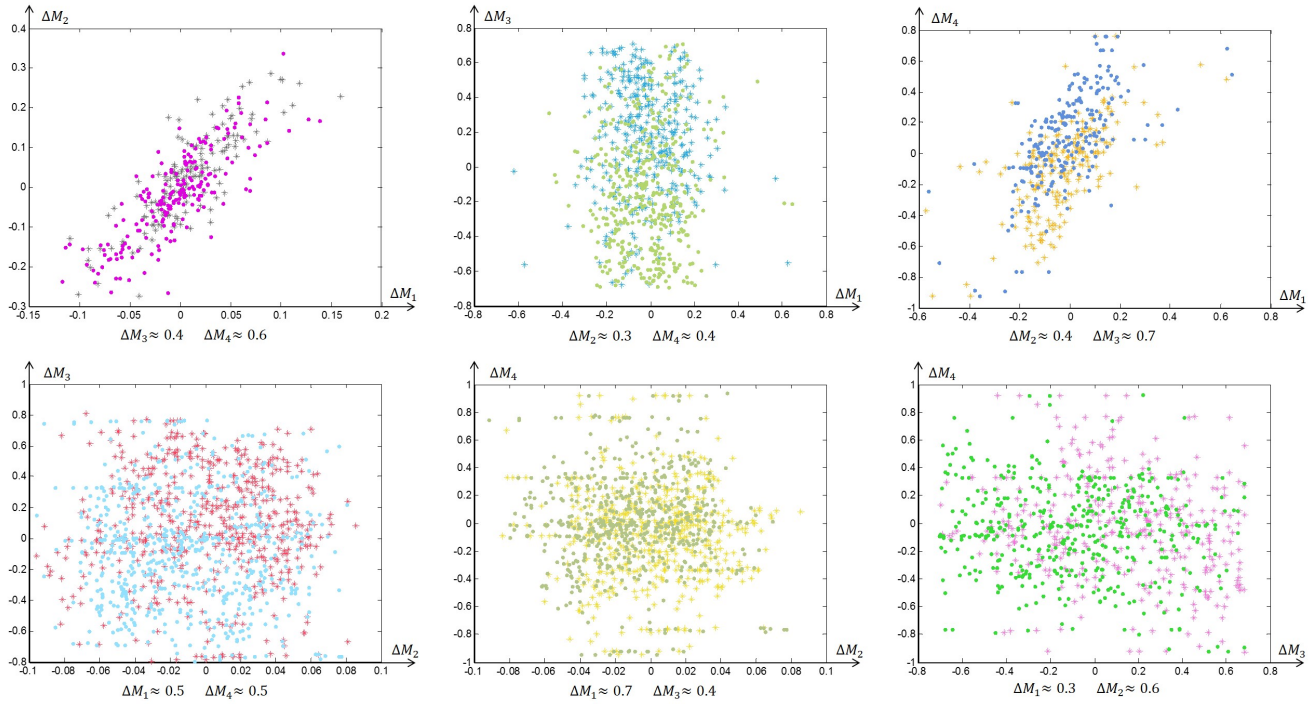


**Figure 8:** *Visualization for the configuration space spanned by the dataset of AMT training. The classification boundary between* $\{+1\}$ *(star) and* $\{-1\}$ *(dot) groups of training samples is nonlinear in the space of 4-dimensional patterns:* $(\triangle M_1, \triangle M_2, \triangle M_3, \triangle M_4)$.

given model. First, sampling $N$ directions (points) on the Gauss sphere. For each $\mathbf{d}_k$, we evaluate values of the relative score function $G(\mathbf{d}_k :: \mathbf{d}_j)$ with all other directions. The 'best' one should be better than most of the other printing directions. We then define a ranking function

$$R(\mathbf{d}_k) = \sum_{j=1}^{N} G(\mathbf{d}_k :: \mathbf{d}_j), \tag{11}$$

with which a larger returned value indicates a more preferable printing direction.

### 5.2 DL-ELM for evaluation

Using relative scores to find good printing directions requires exploring all possible pairs of printing directions, with a slow computational complexity of $O(N^2)$. A new function $F(\cdots)$ is trained in the second layer of our learning model to directly score a printing direction $\mathbf{d}$ according to its metrics, $M_i(\mathbf{d})$ $(i = 1, \cdots, 4)$, which forms a 4-dimensional feature vector $\mathbf{m}$. This learning model is called DL-ELM as double layers of ELM neural networks are conducted (see Fig. 9).

Ideally, the newly trained function $F(\cdots)$ satisfies the following constraint over the entire configuration space spanned by 4-dimensional patterns.

$$\text{sign}\left(F(\mathbf{m}_a) - F(\mathbf{m}_b)\right)) = G(\mathbf{d}_a :: \mathbf{d}_b)$$

Using the sign function will result in formulas with inequalities, which is hard to solve. Therefore, we employ a weak form of the above constraint in our training process:

$$F(\mathbf{m}_a) - F(\mathbf{m}_b) = G(\mathbf{d}_a :: \mathbf{d}_b) \ (\forall \mathbf{m}_a, \mathbf{m}_b \in \Re^4). \tag{12}$$

Substituting into Eq.(7), the function $F(\cdots)$ can be learned by ELM using:

$$\sum_{i}^{L} \beta_i g(\mathbf{a}_i, b_i, \mathbf{m}_a, \mathbf{m}_b) = G(\mathbf{d}_a :: \mathbf{d}_b). \tag{13}$$

with $g(\mathbf{a}_i, b_i, \mathbf{m}_a, \mathbf{m}_b) = \phi(\mathbf{a}_i, b_i, \mathbf{m}_a) - \phi(\mathbf{a}_i, b_i, \mathbf{m}_b)$. The hidden layer output matrix of the network can then be constructed, and its Moore-Penrose generalized inverse is employed to determine the values of $\{\beta_i\}$.

In this second layer of ELM, the bounding box of all training samples in the dataset is first constructed. Uniform samples of the configuration space are then generated inside the bounding box, where each dimension is sampled into 100 points – generating $100M$ samples in total. 100,000 pairs of samples are randomly selected to train the function $F(\cdots)$. Through ELM for regression, we can obtain a function that is able to evaluate the goodness of a printing direction according to the input metrics. The correctness of this function has been verified on all samples of our preference dataset trained by AMT. We found 92% of evaluations given by our nonlinear approximation $F(\cdots)$ are consistent with Turkers' choices, compared with only 86% using ELM method. When using this function to find preferred printing directions, the orientations leading to higher values of $F(\cdots)$ are selected.

## 6 Implementation

This section presents issues relating to implementation details. We employ sampling based methods and computation in image space that can be accelerated by graphics hardware (i.e., GPUs) to evaluate the values of metrics in different printing directions. We sample the Gauss sphere into 1,500 directions and the continuous distribution of metrics is generated by interpolation. Methods for evaluating different metrics are discussed below.

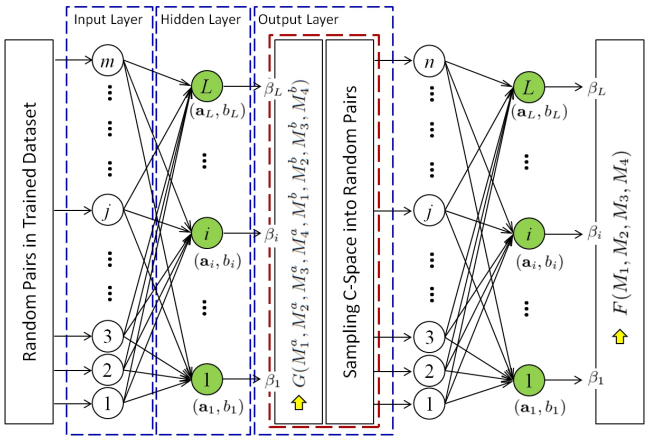


**Figure 9:** *Overview of the DL-ELM method we propose to evaluate the function $F(\cdots)$ indicating preference in 3D printing directions. Three layers of an ELM model (circled by blue dashed lines): input layer, hidden neuron layer and output layer.*

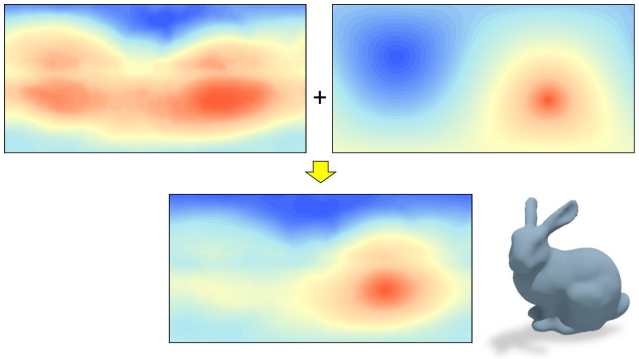

**Figure 10:** *Interactive modification of viewpoint preference – (top-left) automatic computation $M_3(\mathbf{d})$, (top-right) Harmonic field $H(\mathbf{d})$ according to user specification and (bottom) the modified field by average and normalization.*

### 6.1 Contact area

The metrics $M_1$, $M_2$ and $M_4$ each depend on the support contacting region, $\mathcal{T}(\mathbf{d})$, according to a given direction $\mathbf{d}$. An efficient method is crucial since searching for an optimal printing direction could require exploring over one thousand possible printing directions. Facing-down regions in $\mathcal{T}(\mathbf{d})$ can be found easily on polygonal mesh models. The difficult task is the facing-up part which depends on whether there are regions 'shadowed' by overhang areas (e.g., blue regions in Fig. 3). As direct computation on a mesh model is costly, we conduct a rasterization based method to compute a discrete version of $\mathcal{T}(\mathbf{d})$. Specifically, a LDI decomposition is conducted along a printing direction $\mathbf{d}$ for mesh $\mathcal{M}$, which results in a set of intersections stored on parallel rays passing through the centers of pixels in LDI. After sorting the intersections along $\mathbf{d}$, the samples for entering (facing-up) and leaving (facing down) a solid can be known by the order of samples. Moreover, borrowing the idea of Wang et al. [2010], the surface normal is encoded into color values and can be obtained simultaneously during rasterization accelerated by graphics hardware. Points belonging to self-supported regions can be easily detected from the angle between their surface normals and the printing direction $\mathbf{d}$. Here we use the same self-supporting angle as the default in MeshMixer of 25 degrees. By this method, the points belonging to $\mathcal{T}(\mathbf{d})$ can easily be found. Each sample represents a surface region with the same area as a pixel. A discrete integration on the sample gives a good approximation of $A_s(\mathbf{d})$.

The LDI-based decomposition may miss point overhangs and edge overhangs. We compensate for such cases by first detecting point and edge overhangs on meshes and then searching for rays around these overhangs. Samples below these overhangs are added into $\mathcal{T}(\mathbf{d})$. This simple method works nicely in practice. Note that edge overhangs are also sampled into points and inserted into $\mathcal{T}(\mathbf{d})$ together with the point overhangs. The resultant set of points are employed to evaluate all three metrics, $M_1$, $M_2$ and $M_4$.

### 6.2 Viewpoint preference

Computing viewpoint preference is based on evaluating five attributes originally listed in [Secord et al. 2011]. We compute the attributes in image space for fast computation. The *projected area* ($a_1$) of an object is defined on the viewing plane, and can be obtained by counting pixels in the stencil buffer. The *visible surface* attribute ($a_2$) is defined as the surface area of visible triangles over the total surface of object. When an input model has been sampled into LDI points, this can be approximated by the ratio between the number of rays containing samples and the total number of samples. The silhouette of a rendered model can be easily detected in the rendered image by edge extraction filters. The number of pixels belonging to the silhouette is counted to approximate the *silhouette length* ($a_4$) in the image plane. The *maximum depth* is computed with the help of $z$-buffer in the rendering pipeline. The *above preference* attribute ($a_{12}$) is computed exactly by the method of [Secord et al. 2011].

In some scenarios designers may wish to modify the viewpoint preferences via interactive selection. Our system provides a function to modify the distribution of $M_3(\mathbf{d})$ for this purpose. By selecting a strongly intended viewing direction $\mathbf{v}_a$, a Harmonic field $H(\mathbf{d})$ is computed on the Gauss sphere as

$$\begin{aligned} &\nabla^2 H(\mathbf{d}) = 0 \\ s.t. \quad &H(\mathbf{v}_a) = 1,\ H(-\mathbf{v}_a) = 0. \end{aligned}$$

The new metric of viewpoint preference is then updated by the average of $M_3(\mathbf{d})$ and $H(\mathbf{d})$ and then normalized (see Figure 10).

### 6.3 Other details

**Mesh saliency:** To evaluate the metric of visual saliency, we need to compute the distribution of mesh saliency, $\chi(x)$, on a given model. The method of Lee et al. [2005] is conducted. Mean curvature is first evaluated at all vertices. Then, saliency at a vertex is computed as the difference between mean curvatures filtered with a narrow and a broad Gaussian. Different standard deviations are employed to obtain a distribution of saliency, and the final result is the aggregate of saliency at all scales with a non-linear normalization. When a model with large number of vertices is used, the computation could be very time-consuming. However, for each new model, this $\xi(x)$ only needs to be computed once and can be repeatedly used to evaluate $M_2(\mathbf{d})$ in different directions. Moreover, the computation of saliency on mesh vertices is independent and easily sped up on many-core CPUs. In practice, $M_2$ is computed by using sample points in $\mathcal{T}(\mathbf{d})$ as: $M_2(\mathbf{d}) = \sum_{\mathbf{p}\in\mathcal{T}(\mathbf{d})} \chi(\mathbf{p})$.

**ELM learning:** According to the theorem of ELM learning [Huang et al. 2006b], trials are applied to generate different sets of random parameters $(\mathbf{a}_i, b_i)$ for the neurons: $\phi(\mathbf{a}_i, b_i, \mathbf{x})$. In each trial, a

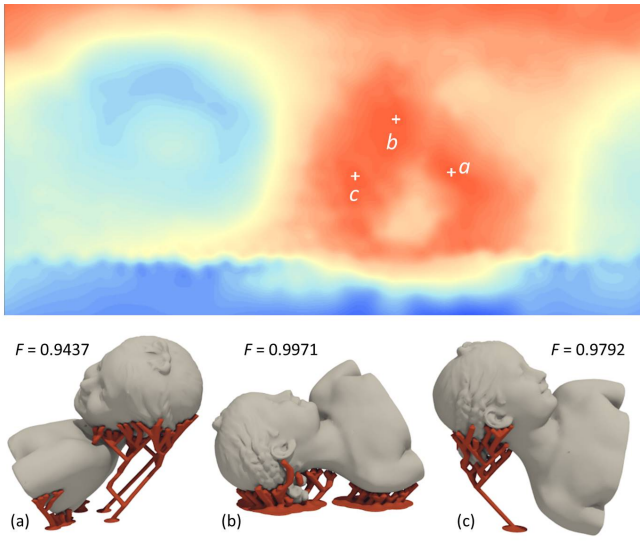


**Figure 11:** *The distribution of $F(\cdots)$ values for a sculptural object – three local maximums can be found. We leave the final selection to the end-users.*

newly learned function $f(\mathbf{x})$ is determined. We then evaluate the accuracy of this function by the $L^2$-norm

$$\sum_j (f(\mathbf{x}_j) - y_j)^2$$

according to Eq.(9). Among all trials, the one with smallest $L^2$-norm is kept as the best result of learning. Also, the number of neurons needs to be determined before training. When more neurons are used, it is easier to get a more accurate result. However, unwanted numerical oscillation will come together as a byproduct. A smaller number of neurons but still giving accurate result are demanded. $L = 15$ is a good tradeoff obtained by our experimental tests.

**'Best' printing direction:** Due to variations in personal preference, the printing direction with highest score in $F(\cdots)$ may not always correspond with the top pick by a user. Therefore, we find all local maximums on the map of $F(\cdots)$ and ask users to make the final decision on which one he or she prefers to use. One example can be found in Figure 11, where there are three local maximums.

## 7 Results

We now describe experimental validation and analysis of our approach. The metrics evaluation is developed as a standalone program in C++ with the use of OpenGL API to speed up the computation on graphics hardware. The machine learning part is conducted in MATLAB. All tests have been performed on a PC equipped with Intel Core i7 3.4 GHZ with 8GB RAM and nVidia GTX 660 Ti with 2GB VRAM. In all the tests, $1,500$ directions are randomly selected on the Gauss sphere to evaluate the metrics.

For all examples, supports are generated using Autodesk MeshMixer, and models are printed with a MakerBot Replicator 2X. In our comparisons with MeshMixer, we specify only *support area weight* when automatically selecting printing orientation. Support volume and strength parameters are given zero weight since they do not relate to surface appearance.

**Training set:** Ten models are used in the AMT training to obtain the votes of preference on pairs of different printing directions.

**Figure 12:** *Ten models used in the AMT training – both natural and man-made objects are assessed.*

Both natural and man-made models are included (see Fig. 12). The performance of the perceptual model developed in our research was first verified on the three models that are part of the training set. As shown in Figure 13, pairs are randomly selected for these models from the training set. The function values from our perceptual model are compared with the votes from trainers. In each case, the direction with higher function values in our perceptual model is consistent with a higher voting.

**Verification of DL-ELM:** We carried out a study to verify the performance of our DL-ELM learning and the advantage of the global scoring function $F(\cdots)$. The Kitten model shown in Figure 12 is used here to generate the set of data for verification. We randomly selected 14 printing directions. A trainer is asked to provide his or her preferences for each of the 91 unordered pairs, which results in a symmetric pattern as shown in Figure 14 (left). Specifically, a grid location $(i, j)$, is black when direction $\mathbf{d}_i$ is preferred over $\mathbf{d}_j$; otherwise square $(i, j)$ is white. A gray color is used for equal preference, e.g., the diagonal. These pairs of preference are then added into our AMT dataset to learn new functions $G(\dots)$ and $F(\cdots)$. The pattern of signs of $G(\mathbf{d}_i :: \mathbf{d}_j)$ on all $14^2$ ordered pairs can be obtained as shown in Figure 14 (middle) – the result of ELM learning. The pattern obtained from ELM learning may be asymmetric which is caused by a result of $G(\mathbf{d}_a :: \mathbf{d}_b)$ and $G(\mathbf{d}_b :: \mathbf{d}_a)$ that is not always additive inverse (especially when there is no significant perference between $\mathbf{d}_a$ and $\mathbf{d}_b$). Learning both $G(\mathbf{d}_a :: \mathbf{d}_b)$ and $G(\mathbf{d}_b :: \mathbf{d}_a)$ can help improve this aspect of ELM learning. The pattern obtained from DL-ELM learning, i.e. sign of $(F(\mathbf{d}_i) > F(\mathbf{d}_j))$, is symmetric. It can be seen from Figure 14 (right) that results obtained by DL-ELM are more accurate than the results from the relative scores by single layer ELM learning.

**Perceptual model:** The outcome of this work can be directly applied using the coefficients of our perceptual model in Table 1. Note that although two learning functions need to be computed in two layers of ELM learning, the goodness of a printing direction $\mathbf{d}$ on a

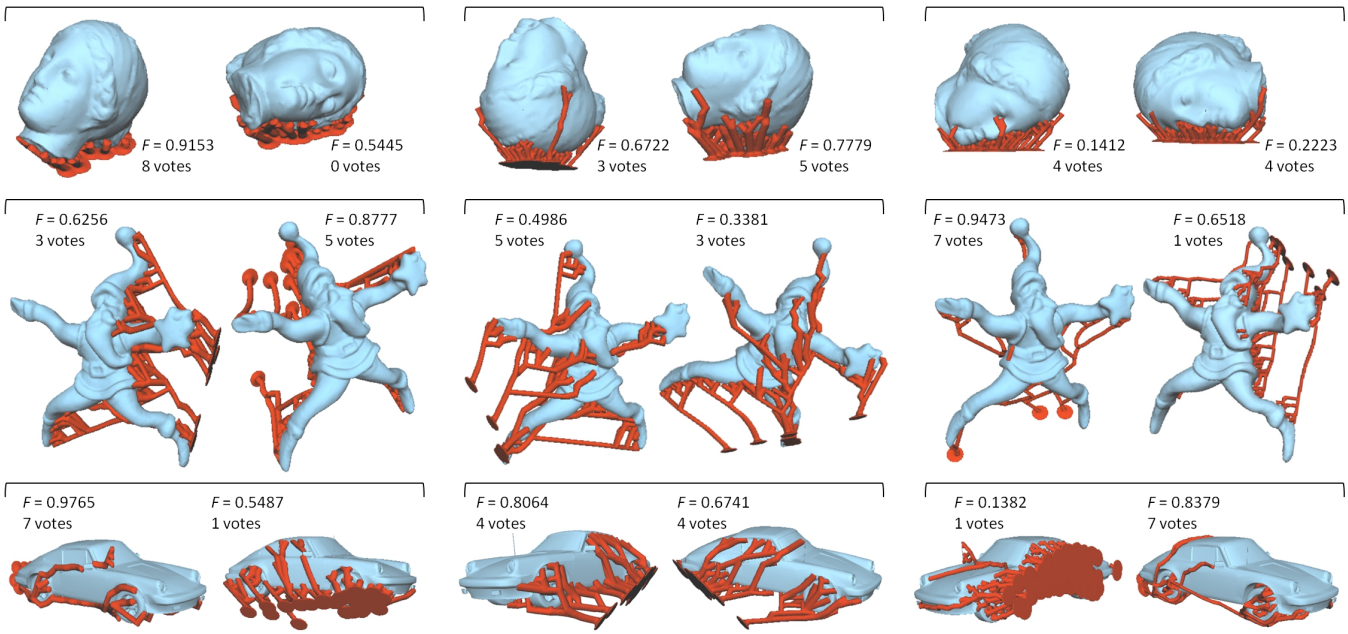


**Figure 13:** *Function values of our perceptual model are consistent with the votes given by Turkers in the AMT training – the direction with higher $F$ value has more votes than the other in the same pair.*

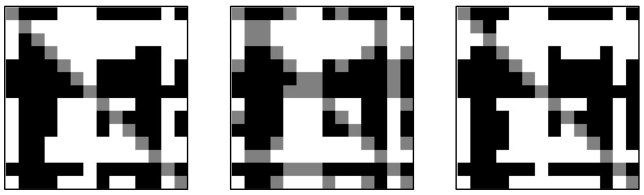

**Figure 14:** *Verifying the accuracy of our DL-ELM learning compared to conventional ELM: (left) test samples from Turkers' preference, (middle) ELM results and (right) our DL-ELM results. A black grid square indicates direction $\mathbf{d}_i$ is preferred over $\mathbf{d}_j$.*

model $M$ can be directly evaluated by using the function in Eq.(7), where an input $\mathbf{x}$ is formed by the four metrics $\{M_i, i = 1, \ldots, 4\}$ as described in Section 3. For a training set of 24k pairs of preference input, the function of our DL-ELM method can be obtained in 9.6 minutes. Note that, 10.5M pairs of samples are employed in the ELM regression of the second layer of DL-ELM.

**Novel input:** We studied performance on novel 3D meshes to further verify the function of our perceptual model. Given a new object that is not in the training set, e.g., with the armadillo in Figure 15, a preferable printing direction is computed by finding a point with maximal score in $F$. An example of a man-made model is given in Figure 17. Our results are compared with Autodesk MeshMixer and also the 'optimal' results obtained by considering only a single factor ($M_2$ and $M_3$ are tested in these examples). These can be obtained by searching for the maximum of $M_2(\mathbf{d})$ or $M_3(\mathbf{d})$ on their corresponding distribution. Contradictions occur between factors (e.g., minimizing material leads to bad feature preservation), thus results obtained from a single metric are less optimal than our multiple factor approach. More examples can be found in Figure 16.

Our perceptual model generates results with better appearance by placing fewer support structures at visually important regions. Run times for the whole procedure are shown in Table 2. The computation time of $M_1$ is related to the LDNI resolution and scale of model and can be accelerated by computing in parallel. Running time of $M_2$ can be greatly improved by using a spatial data-structure such as a grid or octree [Lee et al. 2005]. Similarly, the computing time of $M_3$ can be sped up by parallel computing since it depends only on screen resolution and model scale.

**Table 1:** *Coefficients of Perceptual Model*

| $F(\mathbf{x}) = \sum_i \beta_i \phi(\mathbf{a}_i, b_i, \mathbf{x})$ | | | |
|---|---|---|---|
| $i$ | $\beta_i$ | $\mathbf{a}_i$ | $b_i$ |
| 1 | 0.170 | $(0.913, -0.672, 0.092, 0.503)$ | 0.635 |
| 2 | 1.728 | $(0.871, 0.332, -0.202, -0.542)$ | 0.950 |
| 3 | 0.540 | $(-0.084, 0.788, -0.169, -0.871)$ | 0.444 |
| 4 | −3.472 | $(-0.519, 0.033, -0.638, 0.534)$ | 0.060 |
| 5 | −0.393 | $(0.527, 0.405, -0.489, 0.342)$ | 0.866 |
| 6 | −0.238 | $(0.518, -0.692, -0.958, 0.430)$ | 0.631 |
| 7 | −0.995 | $(0.481, 0.906, 0.847, 0.284)$ | 0.355 |
| 8 | −4.020 | $(0.487, 0.0818, 0.307, -0.161)$ | 0.997 |
| 9 | −2.752 | $(-0.788, 0.359, 0.865, -0.218)$ | 0.224 |
| 10 | −0.780 | $(0.363, -0.926, -0.673, 0.632)$ | 0.652 |
| 11 | 2.144 | $(-0.073, 0.618, 0.842, -0.365)$ | 0.605 |
| 12 | 2.293 | $(-0.575, 0.497, 0.589, 0.629)$ | 0.387 |
| 13 | 0.933 | $(-0.803, -0.759, 0.154, 0.578)$ | 0.142 |
| 14 | 2.239 | $(0.647, 0.050, -0.119, 0.704)$ | 0.025 |
| 15 | 2.838 | $(-0.650, -0.348, -0.484, 0.011)$ | 0.421 |

**Table 2:** *Run times for sample models (see Fig. 17).*

| **model**† | **# faces** | **time (sec.)** | | | | | |
|---|---|---|---|---|---|---|---|
| | | $M_1$ | $M_2$ | $M_3$ | $M_4$ | supports | DL-ELM |
| cow | 5.8K | 40.2 | 17 | 63 | 1.2 | 1.2 | 0.07 |
| duck | 19.3K | 83.4 | 51.5 | 70.6 | 5.2 | 2.5 | 0.07 |
| gargoyle | 71.5K | 267 | 363.8 | 85.4 | 77.9 | 11 | 0.08 |

†All the models are normalized to 40mm height.

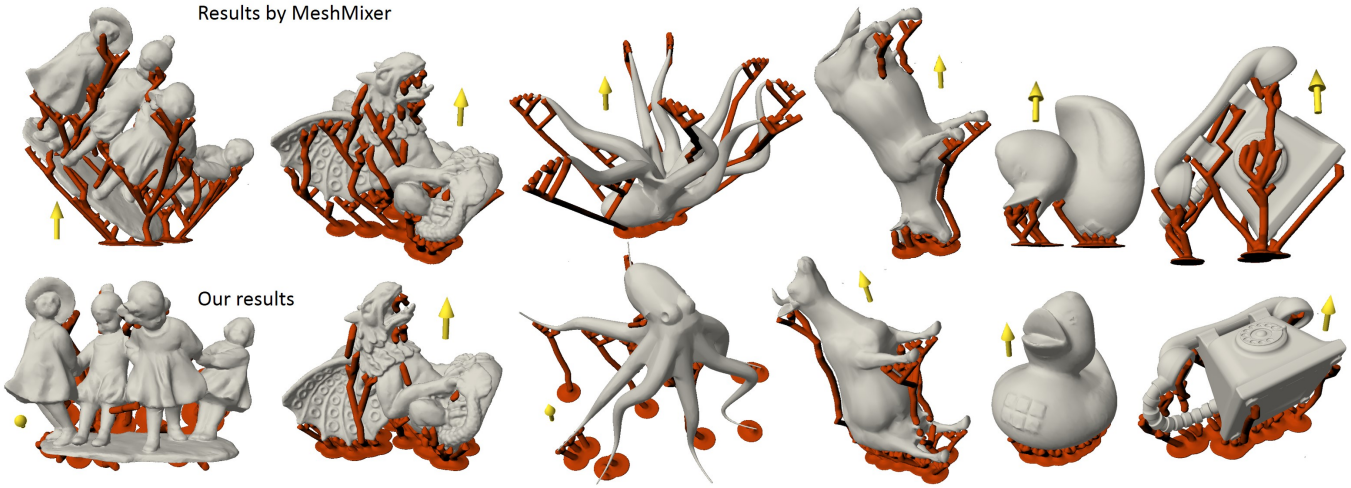


**Figure 16:** *Additional results, novel meshes not in the training set. (Top) generated by MeshMixer, (bottom) our results. Arrows indicate the printing direction.*

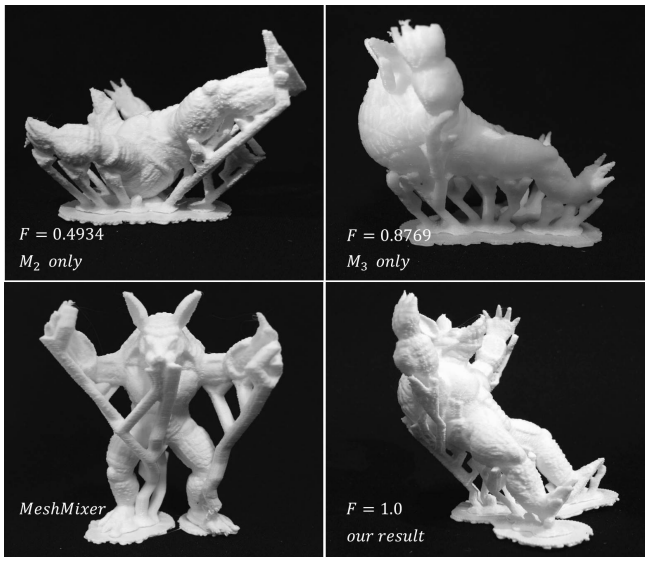


**Figure 15:** *The result of a sculptural object. (Top-left) $M_2$ only, (top-right) $M_3$ only, (bottom-left) MeshMixer result, and (bottom-right) our result. The values of $F(\cdots)$ are provided.*

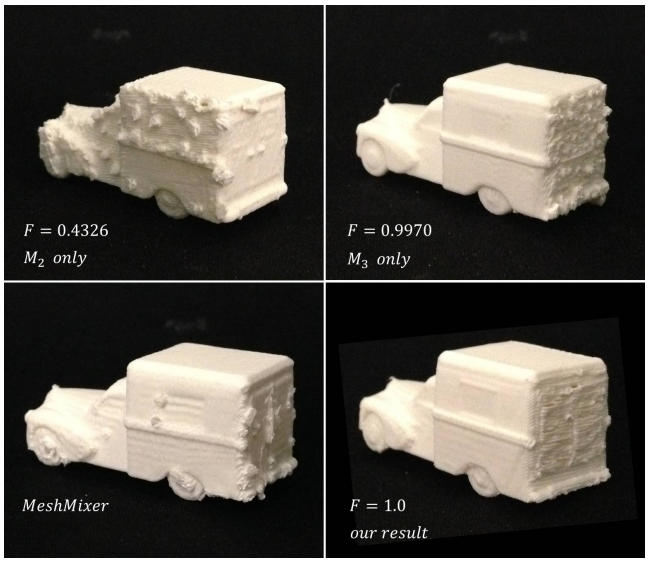


**Figure 17:** *The result of a man-made object. (Top-left) $M_2$ only, (top-right) $M_3$ only, (bottom-left) MeshMixer result, and (bottom-right) our result. The values of $F$ are provided.*

## 8 Conclusion

We present a perceptual model to determine good directions for 3D printing which avoid placing support structures on user-preferred features. Our method minimizes the visual impact of artifacts remaining on a model after support structures are removed. Our perceptual model is established by a training-and-learning process on a large-scale set of samples obtained from the service of Amazon Mechanical Turk. Four attributes, including viewpoint preference, visual saliency, smoothness entropy and area of support, are formulated into a non-linear combination by a novel DL-ELM method, which is able to convert pairwise training sets into a closed-form formulation for evaluating the scores of a model printed in different orientations. Our approach has been implemented on a training set built from ten models (both natural and man-made) and verified on a variety of novel meshes. Experimental results are encouraging – although our results are similar to other methods in some cases, overall we find improved printing directions with reduced visual artifacts, allowing higher quality printed models.

*Limitations and future work:* While the perceptual model proposed in this paper is promising, there are several limitations which could inspire future work. First, the attributes considered may not be comprehensive. For example, symmetry has recently caught attention for enhancing the beauty of 3D models [Podolak et al. 2006; Rhodes et al. 1998]. Future work could study how to incorporate a factor of symmetry in optimal printing directions. Second, the perceptual model learned from large-scale training sets relies heavily on personal preferences. As preferences can vary based on age, gender and culture background, it is worth studying how such dif-

ferences affect the accuracy of our model. Lastly, the computation of mesh saliency and area-of-support is the bottleneck in our pipeline. We plan to further improve performance by parallelizing the computation on multi-core CPUs.

## Acknowledgments

We thank the anonymous reviewers for their comments, and the authors of [Secord et al. 2011] for clarifications. Models provided courtesy of the AIM@SHAPE Shape Repository: Bimda, kitten, dancing children, gargoyle, Max-Planck, armadillo, cow, bunny, octopus, egea, sheep, duck. Remaining models provided by TF3DM and Thingiverse. This material is based upon work partially supported by the HKSAR RGC General Research Fund (GRF) CUHK/14207414, and the National Science Foundation under Grant No. 1464267.

## References

ALEXANDER, P., ALLEN, S., AND DUTTA, D. 1998. Part orientation and build cost determination in layered manufacturing. *Computer-Aided Design 2*, 3, 343–356.

CHALASANI, K., JONES, L., AND ROSCOE, L. 1995. Support generation for fused deposition modeling. In *Proceedings of 1995 Symposium on Solid Freeform Fabrication*, 229–241.

CHEN, Y., LI, K., AND QIAN, X. 2013. Direct geometry processing for tele-fabrication. *ASME Journal of Computing and Information Science in Engineering 13*, 4, 041002.

CHENG, W., FUH, J., NEE, A., WONG, Y., LOH, H., AND MIYAZAWA, T. 1995. Multi-objective optimization of part-building orientation in stereolithography. *Rapid Prototyping Journal 1*, 4, 12–23.

CHOROWSKI, J., WANG, J., AND ZURADA, J. M. 2014. Review and performance comparison of svm-and elm-based classifiers. *Neurocomputing 128*, 507–516.

COCHOCKI, A., AND UNBEHAUEN, R. 1993. *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, Inc.

DONG, Y., WANG, J., PELLACINI, F., TONG, X., AND GUO, B. 2010. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph. 29*, 4, 62:1–62:10.

DUMAS, J., HERGEL, J., AND LEFEBVRE, S. 2014. Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Trans. Graph. 33*, 4 (July), 98:1–98:10.

FAURE, F., BARBIER, S., ALLARD, J., AND FALIPOU, F. 2008. Image-based collision detection and response between arbitrary volume objects. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 155–162.

HEARST, M., DUMAIS, S., OSMAN, E., PLATT, J., AND SCHOLKOPF, B. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications 13*, 4, 18–28.

HERHOLZ, P., MATUSIK, W., AND ALEXA, M. 2015. Approximating free-form geometry with height fields for manufacturing. *Computer Graphics Forum (Proc. of Eurographics) 34*, 2, 239–251.

HILDEBRAND, K., BICKEL, B., AND ALEXA, M. 2013. Orthogonal slicing for additive manufacturing. *Computers & Graphics 37*, 6, 669–675.

HU, R., LI, H., ZHANG, H., AND COHEN-OR, D. 2014. Approximate pyramidal shape decomposition. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2014) 33*, 6, 213:1–213:12.

HU, K., JIN, S., AND WANG, C. 2015. Support slimming for single material based additive manufacturing. *Computer-Aided Design 65*, 1–10.

HUANG, G.-B., CHEN, L., AND SIEW, C.-K. 2006. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *Neural Networks, IEEE Transactions on 17*, 4, 879–892.

HUANG, G.-B., ZHU, Q.-Y., AND SIEW, C.-K. 2006. Extreme learning machine: theory and applications. *Neurocomputing 70*, 1, 489–501.

HUANG, G.-B., DING, X., AND ZHOU, H. 2010. Optimization method based extreme learning machine for classification. *Neurocomputing 74*, 1, 155–163.

HUANG, G.-B., WANG, D. H., AND LAN, Y. 2011. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics 2*, 2, 107–122.

HUANG, G.-B., ZHOU, H., DING, X., AND ZHANG, R. 2012. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42*, 2, 513–529.

HUANG, G., HUANG, G.-B., SONG, S., AND YOU, K. 2015. Trends in extreme learning machines: A review. *Neural Networks 61*, 32–48.

KULKARNI, P., MARSAN, A., AND DUTTA, D. 2000. A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal 6*, 1, 18–35.

LAN, P.-T., CHOU, S.-Y., CHEN, L.-L., AND GEMMILL, D. 1997. Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer-Aided Design 29*, 1, 53–62.

LAN, Y., DONG, Y., PELLACINI, F., AND TONG, X. 2013. Bi-scale appearance fabrication. *ACM Trans. Graph. 32*, 4, 145:1–145:12.

LEE, C. H., VARSHNEY, A., AND JACOBS, D. W. 2005. Mesh saliency. *ACM Trans. Graph. 24*, 3 (July), 659–666.

LEUNG, Y.-S., AND WANG, C. 2013. Conservative sampling of solids in image space. *IEEE Comput. Graph. Appl. 33*, 1 (Jan.), 32–43.

LEVYA, G. N., SCHINDELA, R., AND KRUTH, J. P. 2003. Rapid manufacturing and rapid tooling with layer manufacturing {(LM)} technologies, state of the art and future perspectives. {*CIRP*} *Annals - Manufacturing Technology 52*, 2, 589–609.

LIU, L., SHAMIR, A., WANG, C., AND WHITING, E. 2014. 3d printing oriented design: Geometry and optimization. In *SIGGRAPH Asia 2014 Courses*, ACM, New York, NY, USA, SA '14.

LUO, L., BARAN, I., RUSINKIEWICZ, S., AND MATUSIK, W. 2012. Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Dec.).

MAJHI, J., JANARDAN, R., SMID, M., AND SCHWERDT, J. 1998. Multi-criteria geometric optimization problems in layered manufacturing. In *Proceedings of the fourteenth annual symposium on Computational geometry*, ACM, 19–28.

MASOOD, S., AND RATTANAWONG, W. 2002. A generic part orientation system based on volumetric error in rapid prototyping. *The International Journal of Advanced Manufacturing Technology 19*, 3, 209–216.

OTSU, N. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics 9*, 1, 62–66.

PADHYE, N., AND DEB, K. 2011. Multi-objective optimisation and multi-criteria decision making in SLS using evolutionary approaches. *Rapid Prototyping Journal 17*, 6, 458–478.

PAPAS, M., REGG, C., JAROSZ, W., BICKEL, B., JACKSON, P., MATUSIK, W., MARSCHNER, S., AND GROSS, M. 2013. Fabricating translucent materials using continuous pigment mixtures. *ACM Trans. Graph. 32*, 4, 146:1–146:12.

PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3d shapes. *ACM Trans. Graph. 25*, 3, 549–559.

RHODES, G., PROFFITT, F., GRADY, J. M., AND SUMICH, A. 1998. Facial symmetry and the perception of beauty. *Psychonomic Bulletin and Review 5*, 4, 659–669.

SACHS, E., CIMA, M., WILLIAMS, P., BRANCAZIO, D., AND CORNIE, J. 1992. Three dimensional printing: Rapid tooling and prototypes directly from a CAD model. *ASME Journal of Engineering for Industry 114*, 4, 481–488.

SCHÜLLER, C., PANOZZO, D., AND SORKINE-HORNUNG, O. 2014. Appearance-mimicking surfaces. *ACM Trans. Graph. 33*, 6, 216:1–216:10.

SECORD, A., LU, J., FINKELSTEIN, A., SINGH, M., AND NEALEN, A. 2011. Perceptual models of viewpoint preference. *ACM Trans. Graph. 30*, 5 (Oct.), 109:1–109:12.

STRANO, G., HAO, L., EVERSON, R., AND EVANS, K. 2013. A new approach to the design and optimisation of support structures in additive manufacturing. *The International Journal of Advanced Manufacturing Technology 66*, 9-12, 1247–1254.

UMETANI, N., AND SCHMIDT, R. 2013. Cross-sectional structural analysis for 3d printing optimization. In *SIGGRAPH Asia 2013 Technical Briefs*, 5:1–5:4.

VANEK, J., GALICIA, J. A. G., AND BENES, B. 2014. Clever support: Efficient support structure generation for digital fabrication. *Computer Graphics Forum 33*, 5, 117–125.

VIDIMČE, K., WANG, S.-P., RAGAN-KELLEY, J., AND MATUSIK, W. 2013. Openfab: A programmable pipeline for multi-material fabrication. *ACM Trans. Graph. 32*, 4, 136:1–136:12.

WANG, C. C. L., LEUNG, Y.-S., AND CHEN, Y. 2010. Solid modeling of polyhedral objects by layered depth-normal images on the gpu. *Comput. Aided Des. 42*, 6 (June), 535–544.

WANG, W., WANG, T. Y., YANG, Z., LIU, L., TONG, X., TONG, W., DENG, J., CHEN, F., AND LIU, X. 2013. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph. 32*, 6 (Nov.), 177:1–177:10.

WANG, W., CHAO, H., TONG, J., YANG, Z., TONG, X., LI, H., LIU, X., AND LIU, L. 2015. Saliency-preserving slicing optimization for effective 3d printing. *Computer Graphics Forum*.

XIE, Z., XU, K., LIU, L., AND XIONG, Y. 2014. 3d shape segmentation and labeling via extreme learning machine. *Computer Graphics Forum 33*, 5, 85–95.

XIE, Z., XU, K., SHAN, W., LIU, L., XIONG, Y., AND HUANG, H. 2015. Projective feature learning for 3d shapes with multi-view depth images. *Computer Graphics Forum (Proc. of Pacific Graphics 2015) 34*, 6, to appear.

## Appendix: Computation for ELM

The learning parameters $\mathbf{a}_i$, $b_i$ and $\beta_i$ can be obtained from the constitution equations enforced on $m$ input-output training samples. In other words, Eq.(9) can be rewritten into

$$\mathbf{H}\mathbf{b} = \mathbf{y} \tag{14}$$

where

$$\begin{aligned}&\mathbf{H}(\mathbf{a}_1, \ldots, \mathbf{a}_n, b_1, \ldots, b_n, \mathbf{x}_1, \ldots, \mathbf{x}_m)\\ &= \begin{bmatrix} \phi(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & \phi(\mathbf{a}_n, b_n, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ \phi(\mathbf{a}_1, b_1, \mathbf{x}_m) & \cdots & \phi(\mathbf{a}_n, b_n, \mathbf{x}_m) \end{bmatrix}_{m\times n}, \end{aligned} \tag{15}$$

$$\mathbf{b} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}. \tag{16}$$

$\mathbf{H}$ is called the hidden layer output matrix of the network [Huang et al. 2006b]. The $i$-th column of $\mathbf{H}$ represents the $i$-th hidden node's output vector with respect to inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, and the $j$-th row of $\mathbf{H}$ is the output vector of the hidden layer with respect to input $\mathbf{x}_j$. Note that $n \ll m$, the training error cannot be made exactly zero but can approach a minimum. Input weights $\mathbf{a}_i$ and biases $b_i$ of the hidden layer may simply be assigned with random values according to any continuous sampling distribution without tuning [Huang et al. 2006b]. Eq.(14) then becomes a linear system and the output weights $\mathbf{b}$ are estimated as

$$\hat{\mathbf{b}} = \mathbf{H}^{\dagger}\mathbf{y}, \tag{17}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the hidden layer output matrix $\mathbf{H}$.

The universal approximation capability has been analyzed in [Huang et al. 2006b] that ELM neural network with randomly generated additive and a wide range of activation functions can universally approximate any continuous target functions in any compact subset of the Euclidean space $\Re^n$.