Geometric Shape Deviation Prediction with RNN Model for Additive Manufacturing

Mingdong Lyu

Viterbi School of Engineering, University of Southern California

# Abstract

With the increasing demand of the high-quality additive manufacturing (AM) product, the geometric shape deviation is a one of the most significant concerns due to the large-scaled customization. Considering the basic mechanism that the shape build-up is a layer by layer fabrication process, a slicing data generation is proposed to overcome the drawback of less training data in AM. Then a novel RNN based prediction model will be proposed to capture the layer-by-layer build-up effect and then give us a universal understanding and prediction for the shape deviation. Experimentation in   Experimentation in Fused Deposition Modeling (FDM) are conducted to demonstrate the proposed approach.

# 1. Introduction

## 1.1 Background

### 1.1.1 Additive Manufacturing

Additive Manufacturing (AM) is an advancing and increasingly popular manufacturing technology that adds a new dimension to the modern manufacturing process. It is a process in which a part is made by joining material, layer by layer, directly from 3D model data [1]. AM offers a competitive advantage over traditional manufacturing techniques by enabling fabrication of low volume, customized products with complex geometries and material properties, in a cost-effective and time-efficient way [2]. The rapid proliferation of AM technologies has resulted in the capability of producing metallic parts in various applications instead of fabricating prototypes only. Despite the growth of and advancements in the AM industry, achieving consistency with part quality and process reliability in AM remains a challenge. The fundamental reason for this situation is that both the shape and material properties of a part are formed during the AM process.

### 1.1.2 Geometric Deviation

In order to achieve more elaborate modeling of shape deviation for AM, Huang et al. divide the overall geometric deviations of a fabricated layer into two components: in-plane and out-of-plane [3]. In-plane deviation denotes transformation of the nominal 2D shape of the layer due to thermal shrinkage or machine errors. While the out-of-plane deviation results from the deformation along build direction due to the curling effects. Numerous works have been put on the mechanisms of geometric deviations. Currently, there are several mechanisms including geometric approximate errors, machine errors, and process-related errors as well as material-related errors. The geometry errors mainly result from chordal errors and staircase effects, which focus more on the pre-processing phase of AM processes. Efforts have been devoted to a better transfer for CAD to STL [4-5]. While machine errors means that the slightly different between each machine. When it comes to process-related errors, two aspects must be considered: shrinkage effect and warpage effect. In the quality control community, shape-specific shrinkage models have been proposed aiming at an analytical formulation of the shrinkage along shape boundary either in a single layer [6] or in the cross-sections parallel to the build direction [7].

## 1.2 Project Objectives

With increased availability of AM data, predicting AM built accuracy, or product shape accuracy in particular, has become a focal issue in Machine Learning for Additive Manufacturing. However, since AM enables individualized manufacturing of low-volume products with huge variety and geometric complexity, meeting the requirement of large quantities of data for normal machine learning models is a tough issue. For the purpose of predicting, learning, and compensating 3D shape deviations based on limited shape measurement, we propose a prediction model based on the Recurrent Neural Network (RNN).

## 1.3 Challenge

•       Large varieties of geometric shapes

•    Complicate printing process including inter-layer interactions, various materials and machines

•       Another problem is the low sample size due to one-of-a-kind manufacturing

## 1.4 Project plan

Due to the closing of the lab, we may use the existing scanning data to do the training and verification, instead of designing and printing our own shapes.

Table 1: Timeline for the project

|  |  |
| --- | --- |
| Activity | Date |
| Registration and deviation calculation | 03.30~04.05 |
| Data cleaning and parameter selecting | 04.06~04.12 |
| RNN model building and training | 04.13~04.26 |
| Writing reports | 04.27~05.03 |

# 2. Methodology

A Recurrent Neural Network (RNN) is a special type of artificial neural network, designed with the idea that the outcome of each neuron is dependent on its input and a history variable from past operations, which enables this structure to work with sequential data. Traditional feed-forward neural networks take in a fixed amount of input data all at the same time and produce a fixed amount of output each time, meaning that the first input will not alter the output of the second input. However, in many cases like understanding lyrics, translation, and pattern recognition, the input data interconnect with each other, so there is a need to study the relationship between the input element at the same time of training a neural network. Hence, instead of consuming all the input data at once, the RNN takes them in one at a time and in a sequence. At each step, the RNN does a series of calculations before producing an output. The output, known as the hidden state, is then combined with the next input in the sequence to produce another output. This process continues until the model is programmed to finish or the input sequence ends [8]. A typical many-to-one RNN structure is shown in Figure 1.

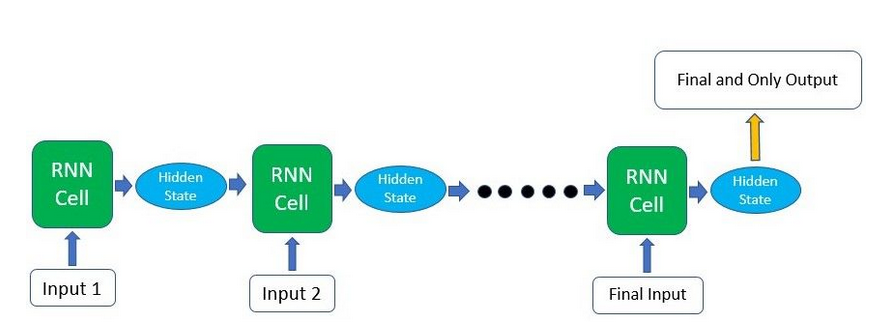


Fig. 1: Many-to-one RNN structure [8]

In our study, since the shape build-up is layer by layer printing process, the current layer will not only be influenced by the previous layers but also be affected by the future layers due to the heat exchange and thermal conduction. Thus, there will be a strong interaction between layers. Meanwhile, some of the geometric shape characters, such as normal, curvature and roughness, show the pattern of layer-wise similarity. More specifically, a sharp curvature of a point in the current layer tends to show a sharp curvature in the next layer.  Hence, to capture these layer-wise interactions between the points, we choose the RNN model to learn the deviation pattern.

However, since the hidden output of each layer only depends on the Hidden state of the previous layer and the new input, RNN network has a difficulty in learning long-term temporal dependencies [9]. This is because the gradient of the loss function decays exponentially with time. In our project, the changes in curvature or normal requires a long-term memory to figure their trend. Thus, it is essential for the model to learn the previous changes in features in the long term so as to get the precise prediction. To optimize the result, we adopt LSTM, a more complicated structure of RNN, as the prediction model. However, LSTM units include a 'memory cell' that can maintain information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it is output, and when it is forgotten. This architecture lets them learn longer-term dependencies. A typical LSTM RNN neuron consists of three gates: forget gate, update gate, and output gate, as shown in Figure 2.

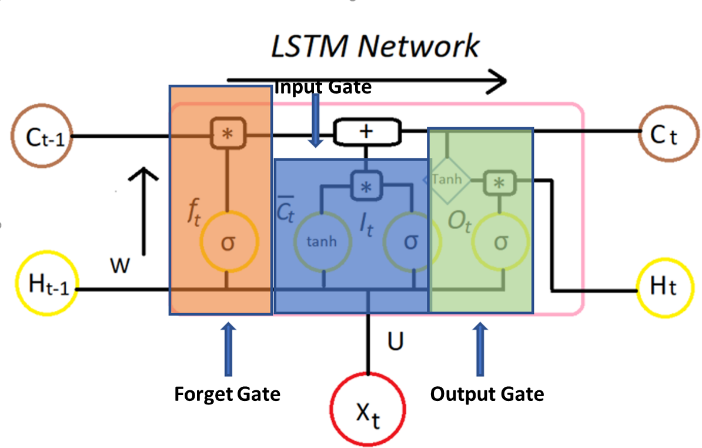


Fig. 2: LSTM neural structure

1. Forget Gate: This gate Decides which information to be omitted in from the cell in that particular time stamp. It is decided by the sigmoid function. it looks at the previous state () and the content input() and outputs a number between 0**(***omit this*)and 1**(***keep this***)**for each number in the cell state .
2. Update Gate: Decides how much of this unit is added to the current state.
3. Output Gate: Decides which part of the current cell makes it to the output.

Considering the large geometric information and features saved in the design file, we are going to extract more features from the STL file, such as the roughness, curvature, and normal, et al. First, we compute the neighbors of each point and calculate each point curvature. We assume that the deviation pattern is related to the curvature and the normal of a plane. When using the same machine to fabricate models with the same materials, the same pattern in the change of these two features indicates the same pattern in shrinkage and curling. Thus, it is quite necessary to add curvature and normal to the data set. The curvature at each point is estimated by best fitting a quadratic around it. The fitting quadric is computed by the neighbors of this point.

K denotes the curvature of the point, and y’, y’’ denotes the first and second derivative of the point respectively.

Roughness is also a key feature to define the quality of a face. It is defined by the deviations in the direction of the normal vector of a real surface from its ideal form [10]. As for point cloud, the roughness is computed by the distance between this point and the best fitting plane computed on its nearest neighbors. Therefore, we add roughness using the following equation:

After calculating the geometric shape features, we are going to slice the design file into layers and treat each layer as each input to the RNN model. Hence the total input structure will be a tensor, i.e. [layer\_number, layerpoint\_number, feature\_number]. Then the training data of the different shape will be integrated together to feed into the RNN. At the last step, we are going to verify the prediction result with the test set.

# 3. Case Study

## 3.1 Shape Design

To catch the change of features layer-wise as much as possible, we design three models with different shapes. As shown in Fig 3, these three models include Primary model, Egg model as well as Tear drop model. The Primary model is a tetrahedron, whose faces consist of four flat triangles. And the egg model is an oval in the top view with continuous changes in curvature in vertical direction. Tear drop model is a model in which its upper part is a cone, while the lower part is a hemisphere. It has both linear change and quadratic change of curvature in the build direction. In this application, we would like to use the Pyramid model and Egg model as the training set and use the Tear Drop model as the testing set, which is difficult for traditional neural networks, since the prediction needs to combine the knowledge from the previous two models.

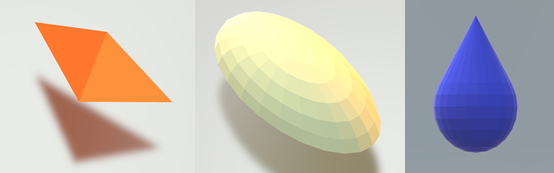


Fig. 3: shape design

## 3.2 Data Preprocessing

We fabricate the three models mentioned above using Makerbot machine. And our raw data is collected by laser 3D scanner. Then we use CloudCompare, a 3D point cloud processing software, to finish our data preprocessing. All the features are shown in Figure 4.

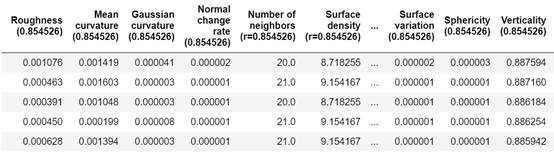


Fig. 4: features extracted from  the dataset

### 3.2.2 Slicing

To use RNN to the prediction model, there is a strict requirement on the input size, which must be in the same shape. Therefore, how to slice the model and form a batch makes a great difference to the result. In this project, we consider each layer as one input and slice the model vertically and horizontally. Figure 5 shows the distribution of point cloud in the build direction. The maximum number in one layer is less than 600 while some layers may have no sample point. The number of point cloud in each layer fluctuate dramatically. The point cloud distribution is scattered since the sample point of scanning is scattered.

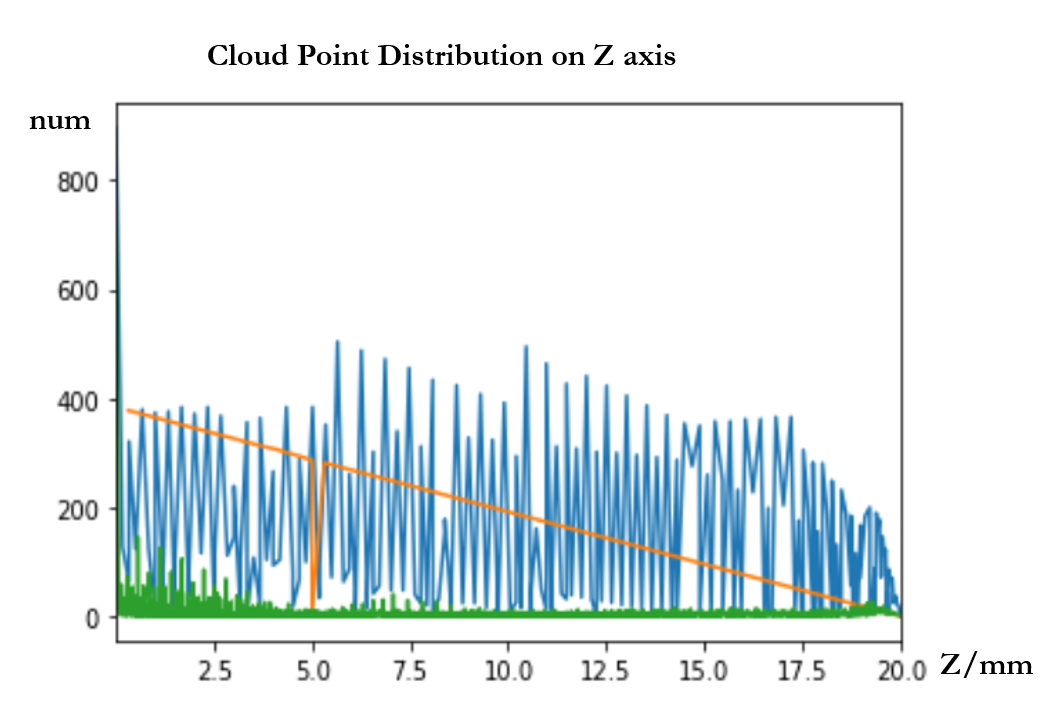


Fig. 5:  point cloud distribution on Z-axis

To tackle the problem of sparsity, in the build direction, the value of z-axis is rounded to 0.1 mm and take a sample layer in a step size of 0.5 mm. This approximate in z-axis is only used for classification, the real input is the original value of the z-axis. As for the in-plane subdivision, polar coordinates are adopted and the sample layer is sliced into 500 pieces. The point that has the nearest angle to the target angle is taken as the sample point. Then we form a batch that contains all this information in the L×M×N matrix. L means the vertical slice number; M means horizontal slice number and N means the length of features.

## 3.3 Model training (include preparing of the training data)

As been mentioned, two models, Pyramid Model and Egg Model are chosen as the training set. Figure 6 shows the training set in CloudCompare. We first choose a traditional RNN as the training model. We define the 32 hidden layers, 500 steps and a learning rate of 0.005. The input of the training model is stored as a tensor which contains the location of X, Y, Z as well as a list of features. The output of the model is the deviation of each point.

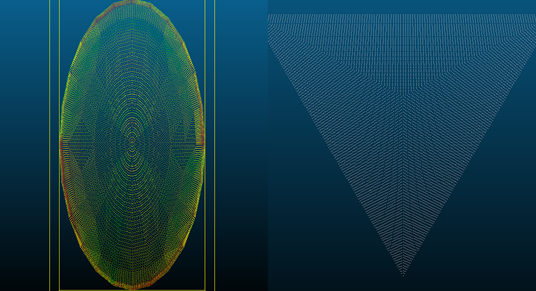


Fig. 6:  point cloud of testing data

Figure 7 shows the prediction result of the RNN training model. the red parts show the deviation of the target value, while the blue parts show the prediction value.  Expect some extreme deviation, the blue parts fit almost the whole red parts, which means that RNN network makes a quite good prediction. Also, you can see that the mean deviation of the blue part is smaller than the red part. This is good for the compensation.

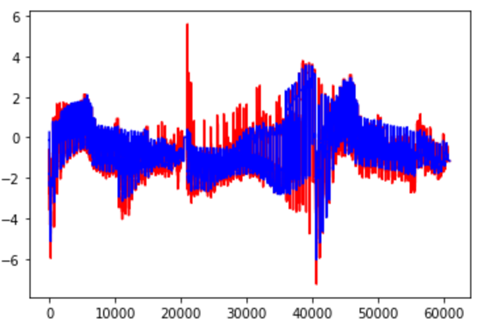


Fig. 7:  training result of  RNN model

Using the mean square error as the loss function, the performance is given by the following formula:

where is the actual deviation and is the predicted deviation. The total loss of this model is 0.02.

In LSTM model, we define 64 hidden unit, 350 steps and a learning rate of 0.005. We also use the MSE as the loss function. The result can be seen in Figure 8: As we can see, the blue parts have almost the similar shape of the red part, which means that our training prediction value fits the target values well. The total loss of the 0.009.

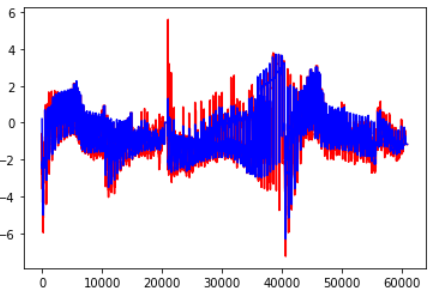


Fig. 8:  training result of LSTM model

## 3.5 Testing result

To valid our model, we choose the Tear Drop Model as our testing data. The point cloud of the model can be seen in Figure 9. The reason why we choose teardrop is that it is a combination of cone and hemisphere, a combination of linear change and quadratic change in curvature, which is a completely new pattern for the prediction model. This is difficult for a traditional prediction since that a traditional prediction model will view the whole model as one input. It will have a poor performance on a new model without training. While in this RNN model, we view one layer as an input. The change of model does not mean the change in the pattern of curvature and normal vector. Thus, the RNN prediction model can easily give a prediction based on the previous training.

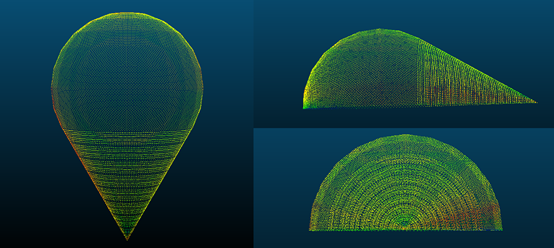


Fig. 9:  testing model of  LSTM model

Figure 10 shows the prediction result and the real deviation of the Tear Drop Model.  Even though the prediction points are slightly sparser than the real deviation model, the prediction model gives a similar result to the real one. In real deviation, the deviation is large on the left side of the hemisphere and the cone. There is also a slight deviation on the top and the right of the hemisphere. And the prediction model clearly points out the deviation on the left and the top of the hemisphere. However, it returns a larger deviation prediction of the right side of the cone. This is because the top of the cone has less valid points which makes it hard for the model to make a correct prediction.

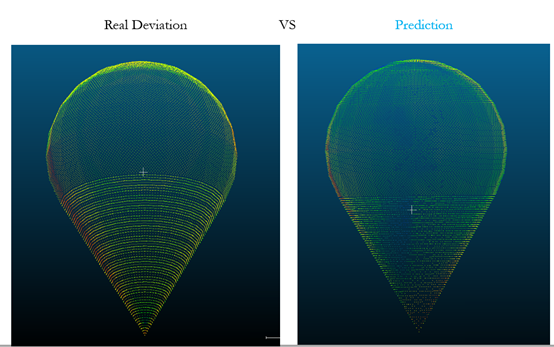


Fig. 10:  testing result of  LSTM model

# 4. Conclusion

This paper has proposed a novel RNN based shape deviation prediction method for Additive Manufacturing, providing a fundamental framework for the geometric quality analysis and control. The proposed layer-wise slicing method combined with the RNN model could capture the intrinsic relationship between points from layer to layer, which enables the prediction of deviation upon the shape features of each point. As it is shown in the case study, the model could predict the deviation of the new shape with a good accuracy.  With the decent prediction and a complete compensation plan developed in the future, the model framework could enhance the overall geometric accuracy of AM for various designs, which will hasten the adoption of Industry 4.0.

# 5. Future Work

Even though the model has a small loss of the test set, we are still uncertain about to what extent we could trust this model to any free-form shape. Therefore, we are going to study the shape similarity and give the confidence level of our prediction. Meanwhile, the sampling and training method we use in this paper is more adaptive to the convex shape. For the complicated non-convex shape, we will try to find some conformal geometric method to transform the non-convex shape to the convex shape and then commit the similar study. In addition, due to the quarantine policy, we can’t select and print a new shape that is most suitable for training. So in the future, we are going to study the most effective shape for machine learning, which contains more information in less number of prints. Finally, after we predict the shape deviation, the next step is how to reduce the deviation by compensating the shape design.

# 6. Lesson learned

Although the coronavirus has made a great challenge on the implementation of the project, it has been fortunate enough to finish it by the end of this semester. Technically, this project went through much knowledge we have learned in class, especially the transformation and vector calculation. After scanning the 3D model, we need to do some transformation to the dataset so that the scanning point cloud has the same coordinates as the design model. While we are processing data, we mainly try to deal with problems in point presentation of a model. We need tantalize the knowledge of how to  compute the normal vector of a face and the curvature. Working through this project, we also learned how to work virtually with high efficiency. And zoom plays an important role in communication. This project also tells us that learning from doing is always the greatest way to make progress.

# 7 References

[1]  B. Berman, “3-D printing: The new industrial revolution,” Bus. Horiz., vol. 55, no. 2, pp. 155–162, Mar. 2012, doi: 10.1016/j.bushor.2011.11.003.

[2] W. Gao et al., “The status, challenges, and future of additive manufacturing in engineering,” CAD Comput. Aided Des., vol. 69, pp. 65–89, 2015, doi: 10.1016/j.cad.2015.04.001.

[3] Huang, Q., Zhang, J., Sabbaghi, A., and Dasgupta, T., 2015, “Optimal Offline Compensation of Shape Shrinkage for Three-Dimensional Printing Processes,” IIE Trans., 47(5), pp. 431–441.

[4] Taufik, Mohammad, and Prashant K. Jain. 2016. “Estimation and Simulation of Shape Deviation for Additive Manufacturing Prototypes.” In Proceedings of the ASME Design Engineering Technical Conference,.

[5] Zhu, Zuowei, Nabil Anwer, and Luc Mathieu. 2017. “Deviation Modeling and Shape Transformation in Design for Additive Manufacturing.” In Procedia CIRP, , 211–16.

[6] Q., 2016. An analytical foundation for optimal compensation of three-dimensional shape deformation in additive manufacturing. Journal of Manufacturing Science and Engineering 138, 061010.

[7] Zhu, Zuowei, Nabil Anwer, and Luc Mathieu. 2019. “Geometric Deviation Modeling with Statistical Shape Analysis in Design for Additive Manufacturing.” Procedia CIRP 84: 496–501. <https://doi.org/10.1016/j.procir.2019.04.251>.

[8] Ren, K., et al. "Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning." *Computer Methods in Applied Mechanics and Engineering* 362 (2020): 112734.

[9] Lin C., Chi M. (2017) A Comparisons of BKT, RNN and LSTM for Learning Gain Prediction. In: André E., Baker R., Hu X., Rodrigo M., du Boulay B. (eds) Artificial Intelligence in Education. AIED 2017. Lecture Notes in Computer Science, vol 10331. Springer, Cham

[10] Degarmo, E. Paul; Black, J.; Kohser, Ronald A. (2003), Materials and Processes in Manufacturing (9th ed.), Wiley, p. 223, ISBN 0-471-65653-4.