A Machine Learning Approach to Modeling Electrochemical Plating Process Variation in RDL Fabrication

Abraham Gonzalez, Chris Lang, and Duane Boning

Massachusetts Institute of Technology

MSRP MIT Summer Research Program

MTL microsystems technology laboratories massachusetts institute of technology

Background and Objective

- A **redistribution layer (RDL)** in integrated circuit (IC) fabrication is a packaging layer that allows multiple different IC chips or die to be integrated together (fig. 1).
- Within RDL, the copper interconnects are fabricated with an **electrochemical plating (ECP)**, a process in which one metal is plated onto another (fig. 2).
- Prone to pattern dependent variations in height (fig. 3), ECP must be planarized using **chemical mechanical polishing (CMP)** in order to prevent subsequent lithography issues.
- **Objective**: Predict these ECP variations with machine learning models such that the CMP step can be avoided or replaced by a simpler and lower cost alternative.

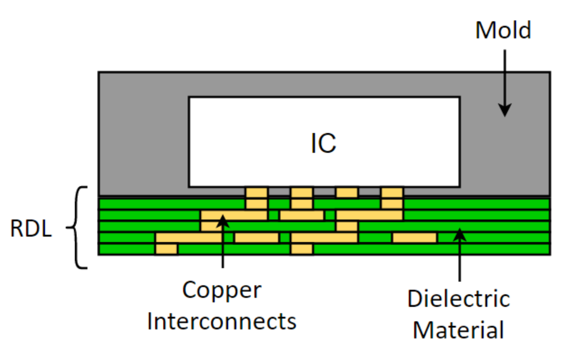

Figure 1: IC Packaging

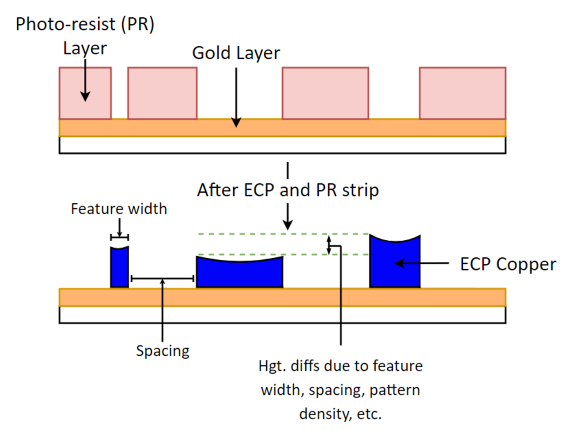

Figure 2: Electroplating Process (ECP)

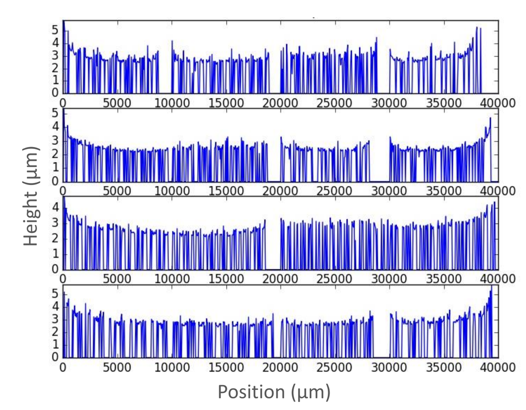

Figure 3: Height variation in copper interconnects

Research Overview

- Creating and optimizing **neural networks (NN)**, emerging machine learning models, to predict ECP RDL variation.
- Create NN structure based on a previous statistical and physically-motivated model and explore similar model structures.

Methods

- A series of models were created to compare performance between a generic NN and the statistical/physically motivated model in TensorFlow, a machine learning API.
- Training data was 3 different layouts that were experimentally grown and measured (fig. 4-7). This dataset was expanded with rotated and mirrored versions as a form of regularization.
- Models were trained by reducing the **root mean squared error (RMSE)** between predictions with the training data and experimental results.
- Performance was measured by testing the error on differing (or similar) layouts after training.

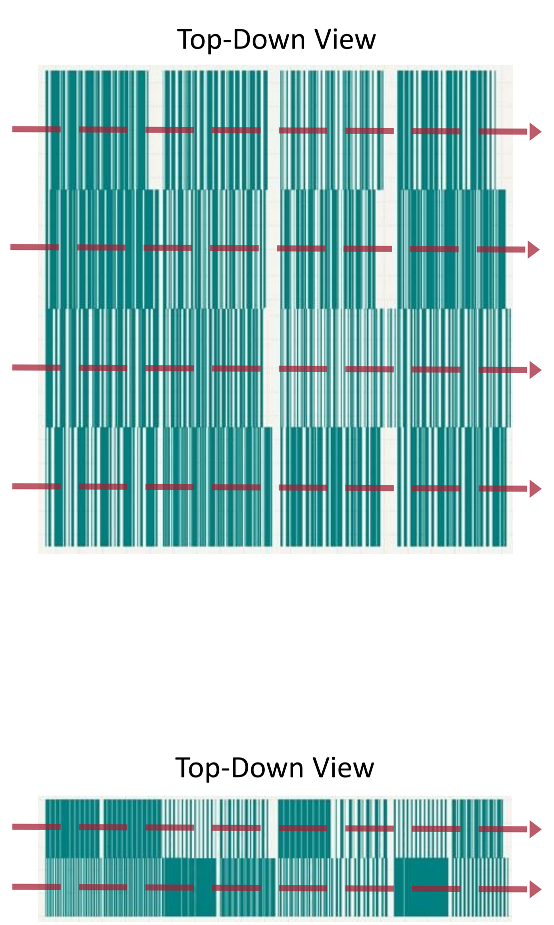

Figure 4-5: 2 training input images before rotation or mirroring

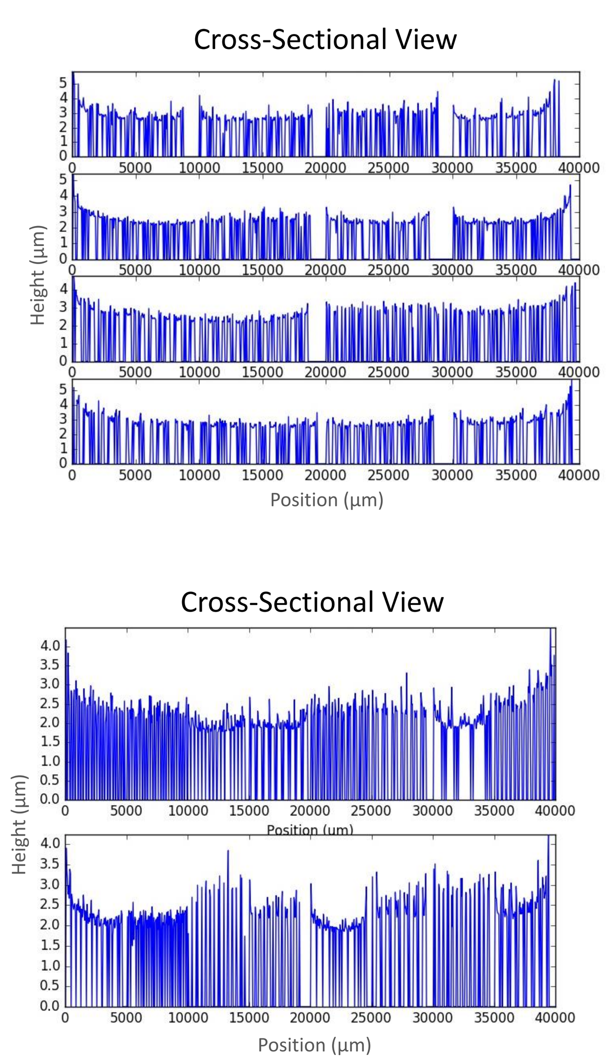

Figure 6-7: 2 training comparison images before rotation or mirroring

Simple Convolution NN

- Simple convolutional neural network model with a single filter and bias to the output (fig. 8)
- The model was able to get the average height of the interconnects but was unable to predict much of the corner variations (fig. 9-11).
- With multiple images, this model ignored the large corner variations in height and just retained the average height.
- Both with a single image and multiple image training, the model did poorly on new layouts.
- Other tests were done with non-linearity unit pass-through added after the bias to take into account the corner variations in actual results. Only resulted in marginal difference.

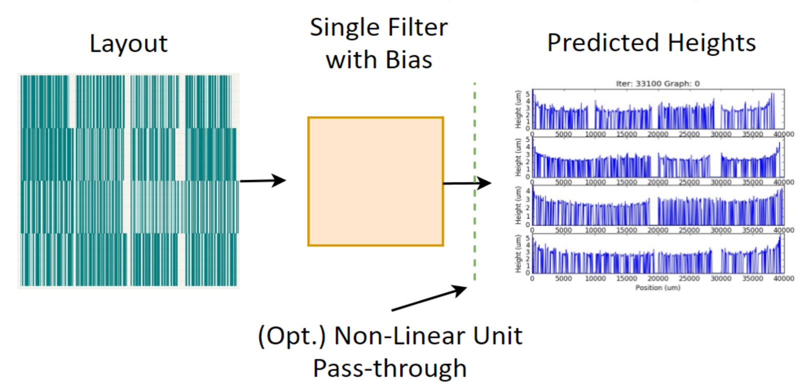

Figure 8: Simple Convolution NN Model

Model	Layouts	Iteration Count	Training RMSE (µm)
Filter and Bias	Single Layout	10,000	~0.392
Filter and Bias	24 Layouts	10,000	~0.501
Filter and Bias with Non-Linear Unit	Single Layout	10,000	~0.388

Figure 9: Model Results

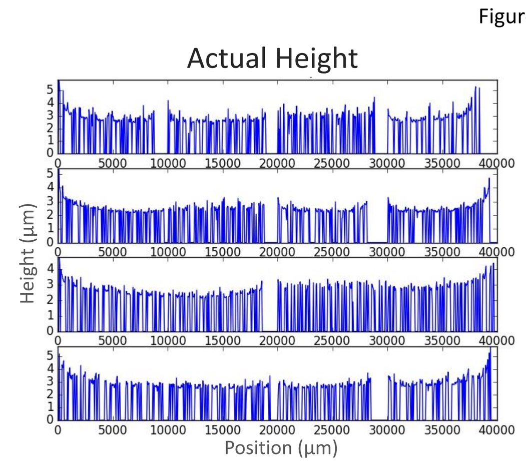

Figure 10: Single Image Actual Height

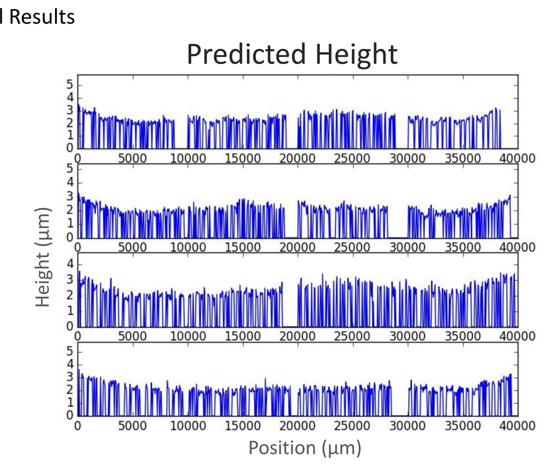

Figure 11: Single Image Test Fit

Cascaded NN with Non-Linear Units

- Cascaded model structure of decreasing size filters. After each layer there was softplus non-linear unit (fig. 12).
- Is a commonly used neural network structure used for image recognition.
- Able to account for the corner variations and general shape of the input layout but was unable to get the feature scale variations present in the output (fig. 13-15).
- Had "flat" property in training fit due to the non-linearity units present in the model.
- Test layout resulted in larger errors due to **overfitting**, where the model is fitting the noise rather than the underlying relationship.

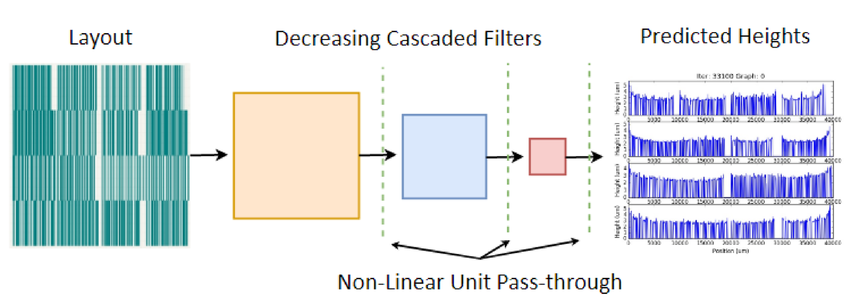

Figure 12: Single Image RMSE

Layout	Iteration Count	Training RMSE (µm)	Testing RMSE (µm)
Single Layout	245,000	~0.088	~0.905

Figure 13: Single Image RMSE

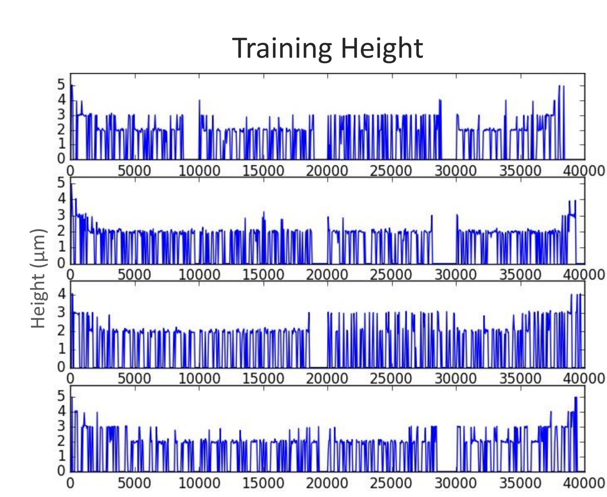

Figure 14: Single Image Training Fit

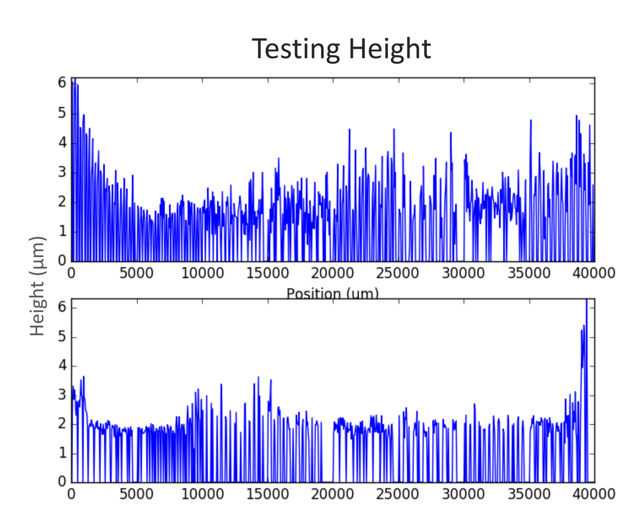

Figure 15: Single Image Test Fit

Modified Model Based on Previous Work

- Previous statistical and physically based NN architecture (fig. 16).
  - **Explicit feature maps:** Density and Pitch filters.
  - **Non-linear function (NLF)**: Ratio of Polynomials.

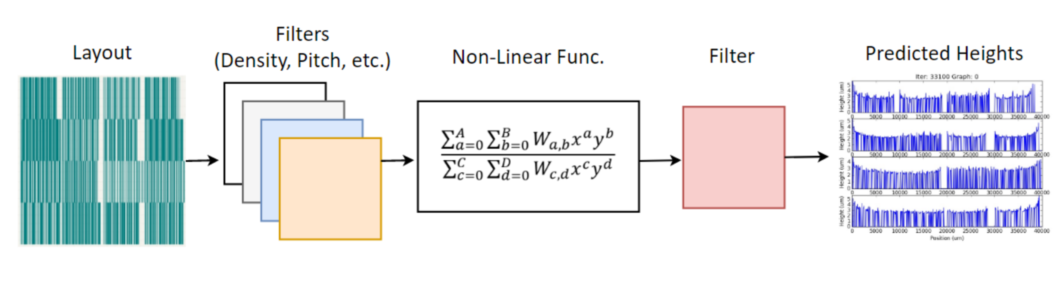

Figure 16: Prev. Highly Structured Model

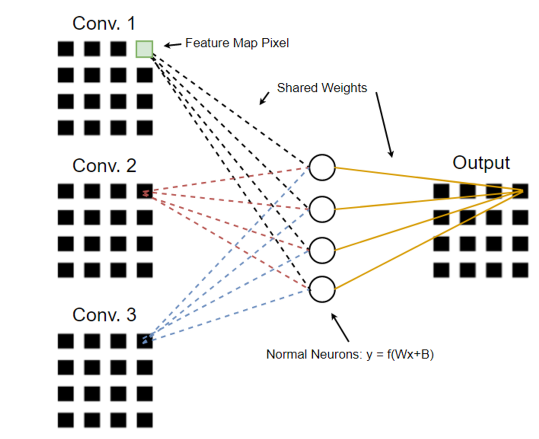

Figure 17: NN Non-Linear function estimator

- Current work changes the previous model by estimating the feature maps and estimating the NLF using a NN estimator (fig. 17).
- New model fitted the macro heights but failed to estimate the inner corner variations (fig. 18-20).
- Additional neurons added to the estimated function resulted in a better training fit but more training time.
- Similar to previous models, the fit of the test layouts was large in error due to overfitting.

Model Type	Iteration Count	Training RMSE (µm)	Testing RMSE (µm)
2 Filters, 10 Neurons	250,000	~0.33	~0.791
2 Filters, 20 Neurons	320,000	~0.275	~1.013
2 Filters, 40 Neurons	225,000	~0.388	~0.784

Figure 18: Model Results for NN Non-Linear Function Estimator

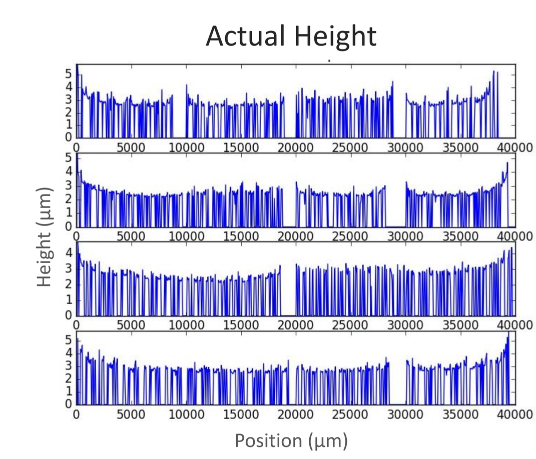

Figure 19: Single Image Actual Height

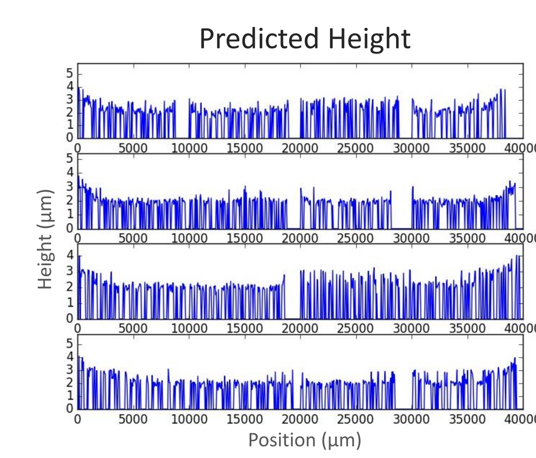

Figure 20: 40 Neuron NLF Model Training Fit

Conclusion

- Explored the use of neural networks to predict ECP variations using multiple model structures.
- Initial filter layers are only able to take into account the macro variations present in the model. Thus adding more depth to the model, results in a better training fit. Additionally, inputting non-linear units at each stage results in capturing a wider range of phenomenon.
- The non-linear function estimator model based on previous statistical and physical based work was much more effective in reducing the "flat" property found in previous models.
- The best performing model structure was the modified model work when there were a large number of neurons in the NLF estimator.

Future Work

- Only a select number of NN architectures were explored in the course of this research. Thus, future work can focus on determining whether different NN models can provide similar or improved results.
- Much of the current work was not run until the model converged since runtimes were too long. Thus, further improvements can be made on model initialization and training speed. Work was done in TensorFlow which was found to be non-ideal for this project; an alternative to explore would be to make the neural networks in a different machine learning tool or from scratch.
- Current work seems to show that more physical intuition behind the model results in a better prediction. Thus, adding physical constraints, or choosing better initializations could improve results.
- Explore in more depth the mixed model. Are the filters learned similar to the filters found previously? Are there more features that can be used to get a better fit?

Acknowledgements

Thank you to Professor Boning, and Chris Lang for the opportunity to learn more about this field and work in their lab. Additionally, thank you to MSRP for the opportunity to do research over this summer.