GTC 2018: 
Physics-Based AI for Semiconductor Inspection Using a GPU Based Optical Neural Network

Jing Zhang, Principle Software Engineer
KLA-Tencor, AI group
March 29 2018
KLA-Tencor Background

Process Control, Portfolio, Challenges
KLA-Tencor Overview

Global Leader in Process Control since 1976

~6,400 global employees

~23,000 tools installed worldwide

>40 years

$3.5B FY17 revenue

$2.1B R&D investment over last 4 fiscal years

17 countries
Our Focus: *Process Control = Foundation of Semiconductor Quality*
“We’re blind without you guys…”

**Inspection**
Find Critical Defects

**Metrology**
Measure Critical Parameters

You can’t fix what you can’t find

You can’t control what you can’t measure
KLA-Tencor’s Inspection Portfolio

3900 Series
Broadband Plasma Wafer

2930 Series
Broadband Plasma Wafer

Puma™ 9980
Laser Scanning Wafer

8920
High Sampling Wafer

CIRCL™
All-Surface Wafer

Surfscan® SPSXP
DUV Unpatterned Wafer

eDR7280™
Wafer SEM Review

eS805™
e-beam Wafer

CIRCL-AP™
Wafer-Level Packaging

ICOS® T3 & T7
Component

Klarity® Defect & ACE
Defect Data Management

RDC
Reticle Data Analysis

Teron™ 640e
Reticle (Mask Shop)

FlashScan™ 200 Series
Reticle Blank

Teron™ SL655
Reticle (IC Fab)

Comprehensive wafer, reticle and component inspection with advanced data analysis supports defect discovery, process optimization and production monitoring.
KLA-Tencor Typical Inspection Paradigm

Find Defects

Identify/Verify Defects
Detecting defects without Fully Resolving in Optical Images

Optical Image
(Fast/Low Resolution)

SEM Image
(Slow/High Resolution)
Moore’s Law leads to the Challenges for Inspection

The Actual Moore’s Law
(About transistor size.)

Approximated Optical Resolution:

<table>
<thead>
<tr>
<th>Wavelength</th>
<th>Red</th>
<th>Green</th>
<th>Violet</th>
<th>UV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>700 nm</td>
<td>500 nm</td>
<td>400 nm</td>
<td>200 nm</td>
</tr>
<tr>
<td>Min Pitch</td>
<td>390 nm</td>
<td>280 nm</td>
<td>220 nm</td>
<td>110 nm</td>
</tr>
</tbody>
</table>

[Computed based on: X. Ma, G. R. Arce, Computational Lithography.]
Significant Technical Challenges

- Multi-Patterning
- 3D Transistors
- 3D Memory
- Advanced Lithography

- Advanced Packaging
- New Processes
- Novel Materials

- Scaling Challenges
- Eroding Process Margins
- Increased Variability

- Increased Variability
## Motivation

<table>
<thead>
<tr>
<th>Physical Modeling</th>
<th>Machine Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Knowledge</strong>: explicitly abstracted into mathematical formula.</td>
<td><strong>Knowledge</strong>: implicitly represented as learned parameters or dataset.</td>
</tr>
<tr>
<td><strong>Data</strong>: Limited</td>
<td><strong>Data</strong>: Plentiful</td>
</tr>
<tr>
<td>PROGRAM = ALGORITHMS + data</td>
<td>MODEL = DATA + network</td>
</tr>
<tr>
<td><em>Use when the physical theory is well-understood and the solution is efficient.</em></td>
<td><em>Use when the first principle theory is missing but data is plentiful.</em></td>
</tr>
</tbody>
</table>
Motivation

<table>
<thead>
<tr>
<th>Physical Modeling</th>
<th>Machine Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Knowledge</strong>: explicitly abstracted into mathematical formula.</td>
<td><strong>Knowledge</strong>: implicitly represented as learned parameters or dataset.</td>
</tr>
<tr>
<td><strong>Data</strong>: Limited</td>
<td><strong>Data</strong>: Plentiful</td>
</tr>
<tr>
<td>PROGRAM = ALGORITHMS + data</td>
<td>MODEL = DATA + network</td>
</tr>
<tr>
<td>Use when the physical theory is <strong>well-understood</strong> and the solution is <strong>efficient</strong>.</td>
<td>Use when the first principle theory is <strong>missing</strong> but data is <strong>plentiful</strong>.</td>
</tr>
<tr>
<td><strong>It is often too complicated to fully understand every physical process for semiconductor manufacturing.</strong></td>
<td>Acquisition of meaningful data can be very time-consuming.</td>
</tr>
</tbody>
</table>

Can we combine these two to address our challenges?
Key Technique Questions

• Is it possible to combine a physical model with Machine Learning?

• How does the combined model meet robustness requirements?

• How does the combined model deal with limited data?

• How to train the combined model in unsupervised manner when we don’t have the ground truth?

• Is computation cost of the combined model economic?
Physics-based Deep Learning to Solve Inverse Problem
Inverse Problem: Deconvolution as an example

Ignoring geometry transformation/distortion and sampling errors, the **blur process** is simply modeled by,

\[
\begin{bmatrix}
0.01 & 0.02 & 0.03 & 0.02 & 0.01 \\
0.02 & 0.06 & 0.08 & 0.06 & 0.02 \\
0.03 & 0.08 & 0.11 & 0.08 & 0.03 \\
0.02 & 0.06 & 0.08 & 0.06 & 0.02 \\
0.01 & 0.02 & 0.03 & 0.02 & 0.01 \\
\end{bmatrix}
\]

\( \times \)

\[
\begin{bmatrix}
0.01 & 0.02 & 0.03 & 0.02 & 0.01 \\
0.02 & 0.06 & 0.08 & 0.06 & 0.02 \\
0.03 & 0.08 & 0.11 & 0.08 & 0.03 \\
0.02 & 0.06 & 0.08 & 0.06 & 0.02 \\
0.01 & 0.02 & 0.03 & 0.02 & 0.01 \\
\end{bmatrix}
\]

\( \Rightarrow \)

\[
\begin{bmatrix}
0.01 & 0.02 & 0.03 & 0.02 & 0.01 \\
0.02 & 0.06 & 0.08 & 0.06 & 0.02 \\
0.03 & 0.08 & 0.11 & 0.08 & 0.03 \\
0.02 & 0.06 & 0.08 & 0.06 & 0.02 \\
0.01 & 0.02 & 0.03 & 0.02 & 0.01 \\
\end{bmatrix}
\]

**Deconvolution** is to restore the “real” image from the possibly **band-limited, noisy and distorted** observation, provided some prior information.
Conventional Methods to Solve Deconvolution

• Bayesian Inference Methods, e.g., MAP:

\[
\min_{\{x\}} \left( y \otimes k - x - x \cdot \log \frac{y \otimes k}{x} \right) + \lambda \cdot \Psi(\nabla x)
\]

- Poisson Distribution
- Prior/Regularization

• Optimization Methods, e.g., Alternating Direction Method of Multipliers (ADMM):

\[
\min_{\{x\}} \frac{1}{2} \left\| \frac{y \otimes k - y}{2} + \lambda \frac{z}{1} \right\| \text{ f(x) g(z)}
\]

\[s.t. \quad \nabla x = z\]

\[
\min_{\{x, z, \beta\}} L_\rho(x, z, \beta) = f(x) + g(z) + \frac{\rho}{2} \left\| x - z \right\|_2^2 - \beta^T (x - z)
\]

Deconvolution – Beyond Resolution Limit

- Original:
- Deconvolution:
Can we mapping the iterative algorithm to Neural Network?


ADMM Network = Layer mapping + Loop unrolling

\[
x^{(n+1)} = \mathbb{P}^{-1} \left\{ \mathbb{F} \left\{ k^T \otimes y + \frac{x^{(n)}}{2\mu} - \frac{\rho}{2} \left( \sum_i W_i^T \otimes (W_i \otimes x^{(n)} - (z_i^{(n)} - u_i^{(n)})) \right) \right\} \right\}
\]

\[
z_i^{(n+1)} = S_{\lambda/\rho} \left( W_i \otimes x^{(n+1)} + u_i^{(n)} \right)
\]

\[
u_i^{(n+1)} = u_i^{(n)} + (W_i \otimes x^{(n+1)} - z_i^{(n+1)})
\]
ADMM Network = Layer mapping + Loop unrolling

\[
x^{(n+1)} = \mathbb{F}^{-1} \left\{ \frac{\mathbb{F} \left\{ k^T \otimes y + \frac{x^{(n)}}{2\mu} - \frac{\rho}{2} \left( \sum_i W_i^T \otimes \left( W_i \otimes x^{(n)} - (z_i^{(n)} - u_i^{(n)}) \right) \right) \right\}}{\mathbb{F} \left\{ k^T \otimes k + \frac{1}{2\mu} \right\}} \right\}
\]

\[
z_i^{(n+1)} = S_{\lambda/\rho} \left( W_i \otimes x^{(n+1)} + u_i^{(n)} \right)
\]

\[
u_i^{(n+1)} = u_i^{(n)} + \left( W_i \otimes x^{(n+1)} - z_i^{(n+1)} \right)
\]
ADMM Network Performance

GUID=22608

Original:

ADMM Network:

<table>
<thead>
<tr>
<th></th>
<th>Ref 1</th>
<th>Def 1</th>
<th>Ref 2</th>
<th>Def 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difference 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>15.9 db</td>
<td>9.7 db</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difference 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>16.6 db</td>
<td>15.4 db</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Practical and Theoretical Limitation

- If we don’t have an explicit solution from first principle, it is very difficult to do “direct mapping”.
  - We can not solve it, if we don’t fully understand the physical process...
  - Sometimes, an approximated solution can be complex also.
Optical Neural Network
Optical Neural Network

- Physical process can be modeled from first principle theories, e.g., Fourier Optics, Maxwell Equation, etc.
Optical Neural Network

• Physical process can be modeled from first principle theories, e.g., Fourier Optics, Maxwell Equation, etc.

• The inverse process is approximated via a parametric model, e.g., Neural Network.
Optical Neural Network

• The learning of such inverse model can be performed via minimizing the loss function:

\[
\arg \min_{\theta} \frac{1}{2} \| F(G(y; \theta)) - y \|_2^2
\]

Parametric Model as the Inverse Method

- Parametric Inverse Model $G(y; \theta)$
- Convolution Neural Network
- Domain Knowledge

• Blur Image $y$
  - CNN
  - Deconvolved $x$
  - Optical Blur Model
  - Residue $y' - y$

$y \xrightarrow{\text{CONV}} CONV \xrightarrow{\text{CONV}} CONV \xrightarrow{\text{POOL}} ....... \xrightarrow{\text{CONV}} CONV \xrightarrow{\text{CONV}} CONV \xrightarrow{\text{CONV}} x$
Differentiable Optical Model as the Forward Method

Blurred Image $y$ → CNN → Deconvolved $x$ → Optical Blur Model → Residue $y' - y$

Deconvolved $x$ → FFT($x$) → FFT($x$)•OTF → FFT$^{-1}$(FFT($x$)•OTF) → $y'$

- Point Spread Function (PSF)
- Wavelength
- Aperture
- Pixel Size
- ....
[Synthetic image is generated from incoherent simulation based on known PSF and shot noise to added to simulate the sensor noise.]
## ONN – Real Experiment

### Original:

<table>
<thead>
<tr>
<th>GUID=22608</th>
<th>Ref 1</th>
<th>Def 1</th>
<th>Ref 2</th>
<th>Def 2</th>
<th>Difference 1</th>
<th>SNR = 15.9 db</th>
<th>Difference 2</th>
<th>SNR = 9.7 db</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR = 16.6 db</td>
<td>SNR = 14.9 db</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ADMM Network:

<table>
<thead>
<tr>
<th>GUID=22608</th>
<th>Ref 1</th>
<th>Def 1</th>
<th>Ref 2</th>
<th>Def 2</th>
<th>Difference 1</th>
<th>SNR = 16.2 db</th>
<th>Difference 2</th>
<th>SNR = 15.4 db</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR = 16.2 db</td>
<td>SNR = 14.9 db</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ONN:

<table>
<thead>
<tr>
<th>GUID=22608</th>
<th>Ref 1</th>
<th>Def 1</th>
<th>Ref 2</th>
<th>Def 2</th>
<th>Difference 1</th>
<th>SNR = 16.2 db</th>
<th>Difference 2</th>
<th>SNR = 14.9 db</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR = 16.2 db</td>
<td>SNR = 14.9 db</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Performance

<table>
<thead>
<tr>
<th></th>
<th>PSNR 1 [db]</th>
<th>PSNR 2 [db]</th>
<th>PSNR Difference</th>
<th>Throughput [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>14.1 db</td>
<td>14.0 db</td>
<td>2.8 db</td>
<td>-</td>
</tr>
<tr>
<td><strong>ADMM Network</strong></td>
<td>15.6 db (+1.5 db)</td>
<td>15.6 db (+1.6 db)</td>
<td>1.7 db</td>
<td>0.77 ms</td>
</tr>
<tr>
<td><strong>Optical Neural Network</strong></td>
<td>15.4 db (+1.3 db)</td>
<td>15.4 db (+1.4 db)</td>
<td>1.7 db</td>
<td>0.78 ms</td>
</tr>
</tbody>
</table>

- For both ADMM and ONN, PSNR is improved by +1.4 db on average, and up to +6 db; equivalently the peak/std is improved by +20%.
- PSNR difference is reduced by 1.1 db, which suggests the noise discrepancy is improved.

Benchmark is performed as follow: 32x32 image patch as input, 128x128 image patch as output; Pascal Titan X, Tensorflow, CUDNN 5.1.5 are used on experiment machine. ADMM network is unrolled to have fixed 15 iterations.
Summary

• Is it possible to combine a physical modeling with Machine Learning?
  ➢ Yes, e.g., Optical Neural Network.

• How does the combined model meet robustness requirements?
  ➢ Yes, with enough constraints.

• How does the combined model deal with limited data?
  ➢ Yes, the forward physical model helps.

• How to train the combined model in unsupervised manner when we don’t have the ground truth?
  ➢ Yes, ONN is unsupervised.

• Is computation cost of the combined model economic?
  ➢ Better.
Summary

• Is it possible to combine a physical modeling with Machine Learning?
  ➢ Yes, e.g., Optical Neural Network.
• How does the combined model meet robustness requirements?
  ➢ Yes, with enough constraints.
• How does the combined model deal with limited data?
  ➢ Yes, the forward physical model helps.
• How to train the combined model in unsupervised manner when we don’t have the ground truth?
  ➢ Yes, ONN is unsupervised.
• Is computation cost of the combined model economic?
  ➢ Better.

Thanks to Nvidia for enabling Deep Learning!!!
Thank you!
Backup Slides