A New Lithography Hotspot Detection Framework Based on AdaBoost Classifier and Simplified Feature Extraction

Tetsuaki Matsunawa¹, Jhih-Rong Gao², Bei Yu² and David Z. Pan²

¹Toshiba Corporation
²The University of Texas at Austin
Outline

• Background
• Simplified Feature Extraction
• AdaBoost Classifier
• Experimental results
• Conclusion
Hotspot detection

- Issue: Lithography simulation is time consuming
- Goal: High accurate hotspot detection in short runtime

Input design → OPC → Lithography simulation → Extract hotspot → Refine design

Simulation-contour → Hotspot → Expand

Ratio of lithography simulation time (normalized by 40nm node)

40nm → 28nm → 20nm

Required computational time reduction!

Simulation-based hotspot detection is the most widely used technique
Two major simulation-less approaches

**Pattern Matching**

- Extract layout features
- Hotspot detection model
- Pattern matching
- Library
- Detected hotspot
- Undetected hotspot
- Cannot detect hotspots not in the library

**Machine Learning**

- Extract layout features
- Hotspot detection model
- Classification
- Non-Hotspot
- Hotspot
- Hard to trade-off accuracy and false alarms
Issues of conventional methods

- Lithography Simulation-based
  - Time consuming

- Pattern Matching-based
  - Cannot detect unknown hotspot

- Machine Learning-based
  - Trade off relation between accuracy and false-alarm
Machine learning based hotspot detection

Training data

Feature extraction:
Abstracted vector data

\[ x_1 = (0, 1, 0, 1.5, \ldots) \]
\[ x_2 = (2, 0.5, 0.1, -1, \ldots) \]
\[ x_3 = (1, -1, 0, 0.3, \ldots) \]

Labels

\[ y_1 = -1 \]
\[ y_2 = -1 \]
\[ y_3 = 1 \]

Testing data

Feature extraction

\[ y = f(x) \]

Model training

Feature extraction:
Predicted labels

\[ \hat{y}_1 = -1 \]
\[ \hat{y}_2 = -1 \]
\[ \hat{y}_3 = 1 \]

\[ \hat{y}_1 = 1 \]
\[ \hat{y}_2 = 1 \]
\[ \hat{y}_3 = 1 \]
New hotspot detection framework

State-of-the-art approach

Training Layout

Feature Extraction

Pre-classification

1D patterns

Simple 2D patterns

Complicated 2D patterns

Model Training

Detector 1

Detector 2

Detector 3

Our framework

Training Layout

Simplified Feature Extraction

Model Training with Efficient Algorithm
Outline

• Background

• **Simplified Feature Extraction**

• AdaBoost Classifier

• Experimental results

• Conclusion
What is a “Simplified Feature”

• Optimized layout feature to make model training easier

1. Feature extraction
2. Feature space analysis
3. Optimal feature selection

Optimized layout feature and parameters
Layout features

Fragmentation-based

Higher order local auto-correlation

Density-based

0 order

1st order

2nd order

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{ccccccc}
a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\
a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\
a_{31} & a_{32} & a_{33} & a_{23} & a_{35} \\
a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\
a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\
\end{array}
\]

\[
W_S
\]

\[
W_n
\]
Feature space analysis

- **Principal Component Analysis (PCA)**
  - Dimensionality reduction based on orthogonal transformation
- **Mahalanobis Distance**
  - Normalized Euclidean distance

Compare different features

Dimensionally reduced feature space
Feature space analysis for generalization capability

Training

Decision boundary

Testing

Feature space index

\[ H = \left| 1 - \frac{1}{\alpha + \exp(D_m)} \right| \]

Mahalanobis distance between Hotspots and Non-Hotspots
Comparison of different features

- Fragmentation-based
- Higher order local auto-correlation
- Density-based
Outline

• Background
• Simplified Feature Extraction
• AdaBoost Classifier
• Experimental results
• Conclusion
Machine learning algorithms

LR
(Logistic Regression)

\[ y(x) = \sigma(w^T x) \]

\[ \sigma(a) = \frac{1}{1 + \exp(-a)} \]

SVM
(Support Vector Machine)

Input feature space

High-dimensional space

Kernel trick
How to learn hotspot features

Why AdaBoost algorithm?

Hotspot detection is extremely complicated multiclass-classification problem
  ➢ Hotspot has many defect mode

Conventional method is hard to generalize all variations of hotspot features
  ➢ Accuracy limitation because of too many factors of hotspot

AdaBoost can simultaneously learn many factors of hotspot
  ➢ Utilize boosting algorithm in conjunction with several classifiers
AdaBoost Classifier

Input layout (features)

Calibrate final classifier

\[ Y_M(x) = \text{sign} \left( \sum_{m=1}^{M} a_m y_m(x) \right) \]

x: feature vector
\( a_m \): weight
M: # of base classifier

Set model

Learn layout features in each classifier

\( y_1(x) \)
Classifier for “Narrow” mode

\( y_2(x) \)
Classifier for “Open” mode

\( y_3(x) \)
Classifier for “Not printing” mode

Generalize final classifier in conjunction with base classifiers corresponding to many defect modes

Hotspot
Narrow
Open
Not printing

Non-Hotspot

TOSHIBA
Leading Innovation

© 2015 Toshiba Corporation
Outline

• Background
• Simplified Feature Extraction
• AdaBoost Classifier
• Experimental results
• Conclusion
Detection model training

ICCAD 2012 Benchmark Problem3*


<table>
<thead>
<tr>
<th>Data</th>
<th>Size (KB)</th>
<th># of HS</th>
<th># of NHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train/Predict</td>
<td>Train/Predict</td>
<td>Train/Predict</td>
<td>Train/Predict</td>
</tr>
<tr>
<td>Benchmark1</td>
<td>918 / 1112</td>
<td>99 / 226</td>
<td>340 / 319</td>
</tr>
<tr>
<td>Benchmark2</td>
<td>31655 / 28043</td>
<td>174 / 498</td>
<td>5285 / 4145</td>
</tr>
<tr>
<td>Benchmark3</td>
<td>29933 / 33333</td>
<td>909 / 1808</td>
<td>4643 / 3541</td>
</tr>
<tr>
<td>Benchmark4</td>
<td>12320 / 10072</td>
<td>95 / 177</td>
<td>4451 / 3386</td>
</tr>
<tr>
<td>Benchmark5</td>
<td>5726 / 4893</td>
<td>26 / 41</td>
<td>2716 / 2111</td>
</tr>
</tbody>
</table>
Comparison experiments

- Comparison with related detection methods:
  - [1] Topological classification and critical feature extraction, Yen Ting et. al., (DAC’13)

- Comparison of different algorithms:
  - Logistic Regression and Support Vector Machine
Comparison with state-of-the-art methods

Detection Accuracy

(#correctly detected hotspots/#total hotspots)

False Alarm

(#of falsely detected patterns as hotspots)

Average 95% accuracy with almost 0 false alarm
Comparison of different algorithms

Detection Accuracy
(#correctly detected hotspots/#total hotspots)

False Alarm
(#of falsely detected patterns as hotspots)

Average 95% accuracy with almost 0 false alarm
Conclusion

• Toshiba and UTDA developed a new hotspot detection framework.

• Our method utilizes AdaBoost classifier and simplified feature extraction method.

• Experimental results show that our method can achieve over 95% accuracy with almost 0 false-alarm.