View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by HKU Scholars Hub

The HKU Scholars Hub The University of Hong Kong 香港大學學術庫

Title	Nebulous hotspot and algorithm variability in computation lithography
Author(s)	Lam, EY; Wong, AK
Citation	Journal Of Micro/ Nanolithography, Mems, And Moems, 2010, v. 9 n. 3
Issued Date	2010
URL	http://hdl.handle.net/10722/135094
Rights	Journal of Micro/Nanolithography, MEMS, and MOEMS. Copyright © SPIE - International Society for Optical Engineering.

Nebulous hotspot and algorithm variability in computation lithography

Edmund Y. Lam
The University of Hong Kong
Department of Electrical and Electronic Engineering
Imaging Systems Laboratory
Pokfulam Road, Hong Kong
E-mail: elam@eee.hku.hk

Alfred K. Wong
P.O. Box 1342
Boston, Massachusetts 02113

Abstract. Computation lithography relies on algorithms. However, these algorithms exhibit variability that can be as much as 5% (one standard deviation) of the critical dimension for the 65-nm technology. Using hotspot analysis and fixing as an example, we argue that such variability can be addressed on the algorithm level via controlling and eliminating its root causes, and on the application level by setting specifications that are commensurate with both the limitations of the algorithms and the goals of the application. © *2010 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.3459937]

Subject terms: algorithm variability; computational lithography; lithography process check; hotspot.

Paper 09114PR received Jun. 11, 2009; revised manuscript received May 9, 2010; accepted for publication May 21, 2010; published online Jul. 14, 2010.

1 Introduction

1.1 *Lithography Process Check and Design Rule Check*

Lithography hotspots found through lithography process check (LPC)[1,2] and lithography design rule violations[3] are cousins, sharing some common characteristics while differing in other aspects. Both relate to the robustness of physical designs; the existence of hotspots or violations indicates a propensity to yield loss. Despite yield being an inherently continuous quantity, for design convenience and expediency, hotspots and design rule violations have well-defined pass-fail boundaries. A sharp demarcation is (artificially) placed between violations and nonviolations, and between hotspots and non-hotspots.

Given a layout with a design rule error, it is usually straightforward to understand the cause of the violation. Such cannot be said for hotspots. Except for the simplest of cases where guidelines and rules-of-thumb are applicable, the subtleties of optical imaging and pattern transfer preclude the use of intuition and elementary logic. In fact, the complexity of low-k_1 lithography is the original motivation for model-based verification.

The nonintuitive nature of LPC should not hamper its utility. Numerical computation is increasingly complementing instinctive reasoning in sub-100-nm integrated circuit creation. Model accuracy is perhaps a cause of greater concern. State-of-the-art lithography models still fall short in critical dimension (CD) prediction of 2-D patterns and of non-nominal exposure conditions.[4–6] Resist collapse is also difficult to simulate properly.[2] Nevertheless, design rule check (DRC) is not immune to inaccuracies; design rules and their parameters are also defined with uncertainty and incertitude.

What truly distinguishes LPC from DRC is the former's imprecision. This is, at present, one of the two greatest hindrances to wholesale adoption of LPC in the design flow. (The other obstacle is the general lack of evidence for or conviction in the necessity of LPC.) Imprecision in the present context means the fluctuation of simulation results caused by LPC algorithm imperfections or mismatches. Two identical layout configurations may have inconsistent hotspot classification, with one indicating a hotspot while the other does not. Sources of variability may include process model asymmetry, image interpolation, target layer deviation (due, for example, to hierarchical operations), optical proximity correction (OPC) iteration, and contour computation. Cutting-edge DRC algorithms do not exhibit such impreciseness.

1.2 *Analogy Between Algorithmic and Manufacturing Variability*

The nebulous hotspot is a manifestation of the more general issue of variability in computational lithography algorithms. It is instructive to refer to Fig. 1(a), which shows the measured across-wafer critical dimensions of nominally identical isolated lines. The color shade at each point represents the electrical linewidth of the isolated line at that wafer location. Through a series of analyses, the original distribution [Fig. 1(a)] was decomposed into intrafield [Fig. 1(b)] and extrafield [Fig. 1(c)] components, which were then further partitioned into photomask errors, scanning errors of exposure systems, aberrations, and development nonuniformity.[7] As a result, the root causes of linewidth fluctuation can be quantified separately and remedied individually, thereby reducing the overall variability.

Such analysis and quantification also necessitate and enable the setting of proper design margins to ensure high-yielding and robust circuits. The design margin, or indirectly, the achievable circuit performance, is intimately tied to the amount of linewidth variation. Higher degrees of variability require larger margins, resulting in lower performance, and vice versa. Circuits must be designed with margins commensurate with the level of variability. Forcing circuit performance without proper regard for process fluctuation is a path to a miserable end.

The relationship between computational lithography algorithms, such as image calculation, and applications, such

1932-5150/2010/$25.00 © 2010 SPIE

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

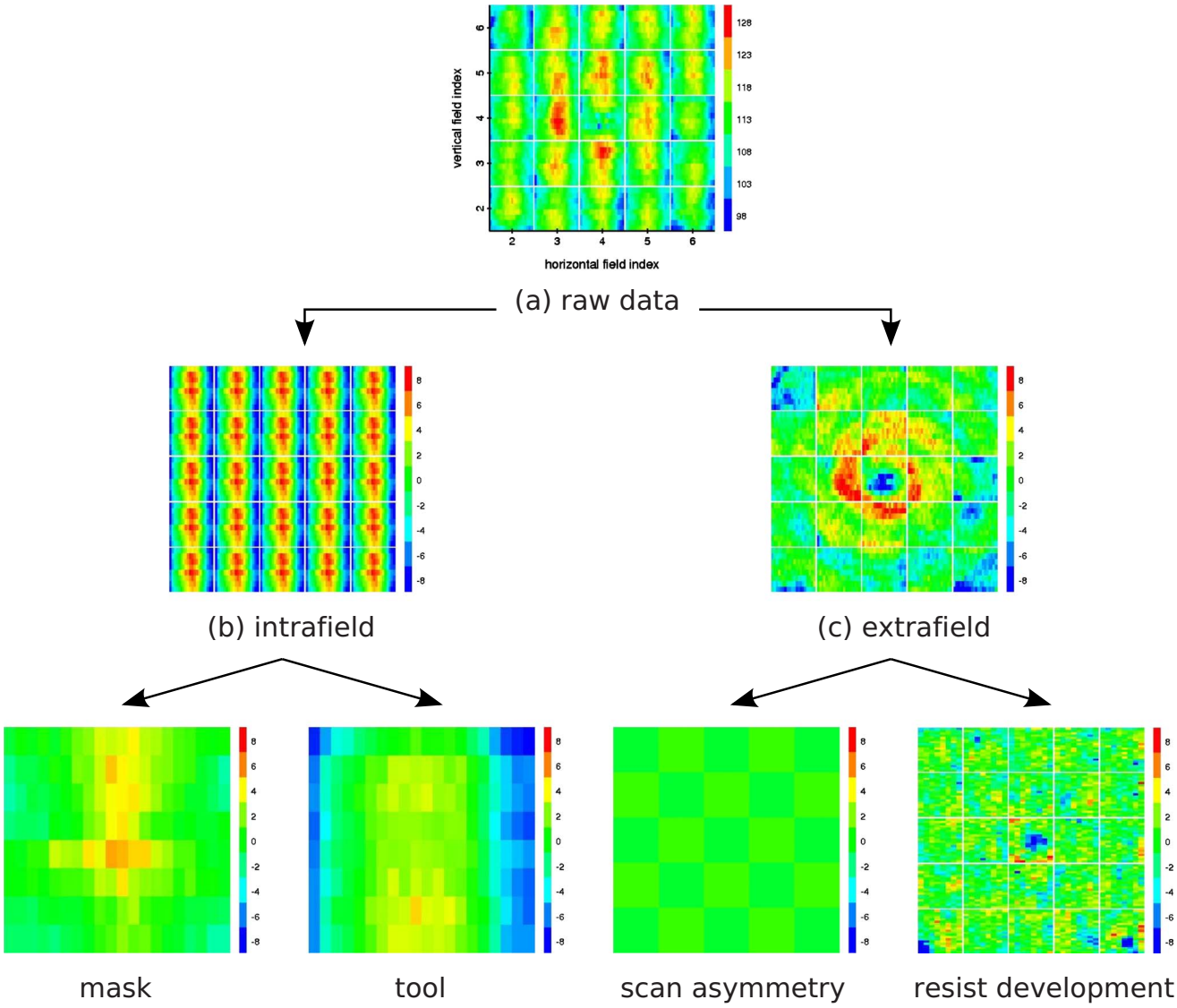

Fig. 1 Measured across-wafer critical dimension variation.[8]

as lithography design-for-manufacturability (DFM), is analogous to that between fabrication and circuit design. Algorithms (semiconductor processing) are the foundation on which applications (circuits) are built, and applications (circuit design) should be cognizant of algorithmic (manufacturing) variability. On the one hand, we need to analyze and decompose sources of variation in order to quantify, monitor, and minimize such fluctuation. On the other hand, given a particular level of variability, we can devise reasonable specifications that are at once realistic, practical, and useful, without imposing unrealistic requirements or delusive expectations.

1.3 *Lithography Physical DFM*

Consider lithography physical DFM as a vehicle for investigation and exposition. This process, illustrated in Fig. 2, includes hotspot analysis (LPC)—identifying regions of a layout that have relatively poor lithography latitude—and hotspot fixing—adjusting the layout to improve the process robustness of the identified regions. For our present pur-

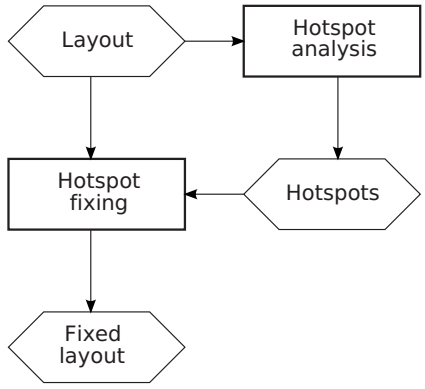

Fig. 2 Lithography physical DFM comprising hotspot analysis and hotspot fixing.

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

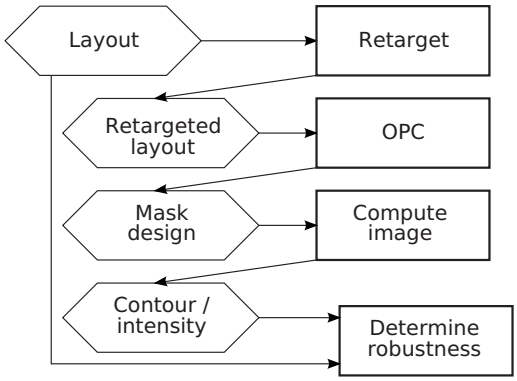

Fig. 3 Typical LPC flow.

poses, we are concerned with algorithmic variability during hotspot analysis, and given such fluctuation, sensible measures we can adopt for hotspot fixing.

2 LPC Variability

It is worthwhile, prior to examining sources of LPC variability, to delineate the concept of *constancy*. LPC behavior that falls outside the bounds of constancy is variability.

We expect that, within the confines of lithography modeling, identical features with identical layout environments would have identical lithography characteristics. For practical purposes, "identical surroundings" means that the neighborhood layouts are identical at least up to a distance on the order of the optical interaction range[8]

$$r_{\text{optical_interaction}} \sim \frac{1}{\sigma} \times \frac{\lambda_0}{\text{NA}},$$

where λ_0, NA, and σ denote, respectively, the wavelength in vacuum, the numerical aperture, and the partial coherence factor of the exposure system. For a 193-nm system with a NA of 1.2, two features should have the same hotspot categorization if their environments within

$$\mathbf{O}(10) \times \frac{1}{0.8} \times \frac{193}{1.2}\ \text{nm} \simeq 2{,}000\ \text{nm}$$

are identical.

Such expectation is not always met in practice. In a typical LPC flow, shown in Fig. 3, the original layout first undergoes retargeting that may include global and selective sizing. OPC is then applied on the retargeted layout, producing a photomask design;[9,10] images of the photomask are computed using a model of the lithography process. From the images are derived contours, critical dimensions, and process latitude information, based on which hotspot decisions are made. Any of these steps may be a source of variability.

2.1 *Simulator-to-Itself Variability (Auto-Inconsistency)*

It may seem peculiar that the same LPC software applied on the same layout can give different results; random generators aside, software should be deterministic. Indeed it is, provided the layout is in all respect *identical*. However, that is rarely the case. Consider the situation in which the same physical design (such as a standard cell) is instantiated twice, once in the vertical position and once in the lateral position, rotated by 90 degrees. Even assuming horizontal-vertical symmetry in the illumination, the simulation results of these two instantiations may still be different if the lithography model exhibits asymmetries.

Careful model building may eliminate such model-induced fluctuation. Nevertheless, variability may remain if the image is computed using the so-called dense approach.[11] This method can be implemented by computing image intensities at a set of grid points. The intervening values are determined by interpolation from the intensities at these grid points. Imagine a scenario with a square image grid of spacing δ and two instantiations of the same physical design having a relative translation of $(\Delta x, \Delta y)$. (A square grid was chosen for exposition convenience here. In general, the grid can be nonuniform and nonrectangular.) If Δx or Δy is not an integral multiple of δ, the images of the two instantiations could be slightly different due to interpolation. Such discrepancy leads to variability in both LPC and OPC.

Measures for improving computation throughput may also contribute to variability. Consider the situation shown in Fig. 4(a), where a long wire terminates at a region densely populated with shapes, and our hotspot interest lies within this dense region. During hierarchical operation or parallel processing, the long wire may be truncated to limit the area of computation. Suppose two instances of the layout are truncated at different locations, indicated by the labels "case 1" and "case 2" in the figure. We further suppose the following rudimentary set of retargeting rules:

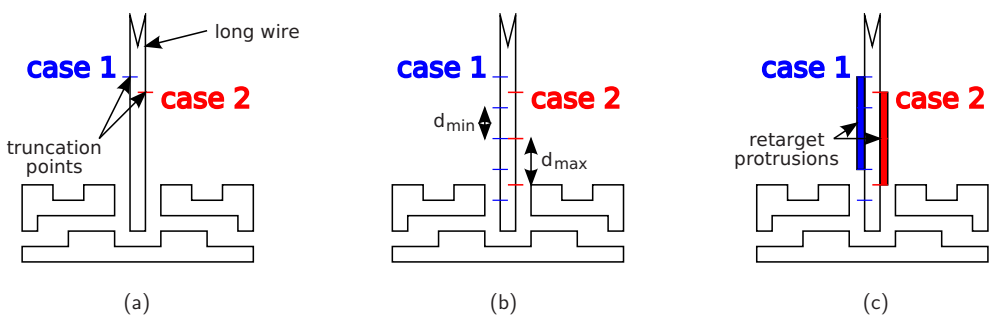

Fig. 4 Measures for improving computation throughput that may contribute to variability. (a) Layout with long wire, (b) segmentation differences, and (c) retargeting differences.

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

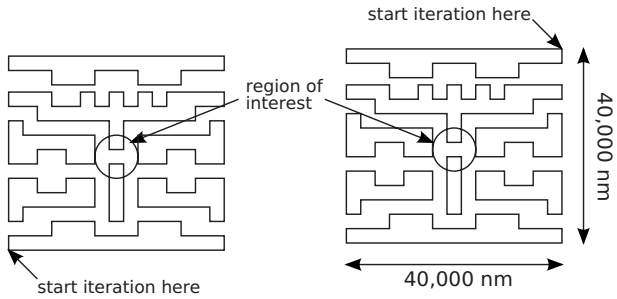

Fig. 5 Diagram showing how OPC result is dependent on the order in which edges are moved. A critical dimension discrepancy as large as 15% is possible.

1. Edges longer than $2 \times d_{\min}$ are segmented uniformly;
2. Each segmented edge length lies between $d_{\min}$ and $d_{\max}$; and
3. The local bias that each segmented edge receives is a function of spacing to its nearest neighbor.

The edges of the truncated long wire would then be segmented differently between case 1 and case 2, as shown in Fig. 4(b). If the truncated original edge of case 1 is segmented into five parts, as illustrated by the left edge of the long wire in Fig. 4(b), each segment is of length $d_{\min}$. At the same time, the shorter truncated original edge of case 2 would have an altogether distinct segmentation of, say, three parts of length $d_{\max}$ each [depicted by the right edge of the long wire in Fig. 4(b)]. Such variance in segmentation, which is a result of minor differences in how a shape is truncated far away, directly translates into differing retargeted layouts, as shown in Fig. 4(c)—which, in turn, causes variability in the lithography image, and hence in the OPC and LPC in the region of interest.

OPC iteration is another source of variability. Some OPC algorithms adjust edges sequentially such that designed masks exhibit a dependency on the order in which the edges are moved. It is possible to contrive a situation, as shown in Fig. 5, in which starting the OPC iteration from opposite corners of a 40 μm $\times$ 40 μm 65-nm layout results in a simulated critical dimension discrepancy of more than 15% at the center.

2.2 *Simulator-to-Simulator Variability (Across-Simulator Mismatch)*

During the hotspot analysis stage in physical DFM, it is sometimes impractical, for turnaround or accessibility reasons, to apply the same algorithm as the one used for signoff verification. The use of different algorithms leads to many more sources of inconsistency. Possible culprits include all aspects of hotspot computation, from model fitting to image calculation, from retargeting to OPC, from contouring to deriving process robustness. Even when two algorithms meet the same set of specifications, implementation differences may still result in variability.

As an example, consider image calculation with a given set of coherently decomposed kernels.[12] Assuming the kernel set is discretized, i.e., each kernel is specified as a collection of values at discrete locations, any algorithm would need to interpolate these supplied values during image calculation. (This interpolation is distinct from the interpolation for dense image computation discussed in Sec. 2.1.) For accuracy considerations, such a feat is best performed using the sinc sampling function.[13] However, for throughput concerns, a relatively local scheme such as cubic spline may be expedient. Algorithms that make different accuracy-throughput tradeoffs give discrepant images, even from the same specified kernel set.

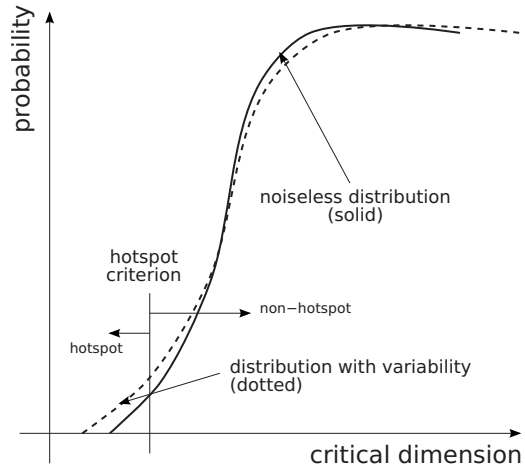

Fig. 6 Example probability distribution of simulated critical dimension without noise (solid line) and with algorithm variability (dotted line).

3 Simulator Matching

3.1 *Formulation*

The existence of variability in computational lithography algorithms means that the results from any application are not be perfectly accurate or absolutely precise. Consider an example hotspot analysis procedure that comprises the following three steps:[14]

1. identify candidate hotspots from the layout,
2. compute the critical dimension of each candidate, and
3. classify each candidate as a hotspot or non-hotspot according to the calculated critical dimension.

By applying this three-step procedure, the critical dimensions of all candidates of a layout would form a distribution. Let $\mathcal{P}_{\text{noiseless}}(x)$ denote the true distribution, which may resemble the solid curve in Fig. 6. Note that the x axis, marked "critical dimension," can be taken to refer to the geometric dimension of the hotspot, but more generally it can also represent the hotspot severity, score, or index. In any case, we have no way of knowing the precise curve unless we have a noiseless algorithm at our disposal. What we observe instead is $\mathcal{P}_{\text{variability}}(x)$, a tainted version of $\mathcal{P}_{\text{noiseless}}(x)$ that may look like the dotted curve in Fig. 6. If we assume the algorithm variability is random and that such randomness can be represented by the noise distribution function $f_{\text{noise}}(x)$, we can then establish the following expression:

$$\mathcal{P}_{\text{variability}}(x) = \mathcal{P}_{\text{noiseless}}(x) * f_{\text{noise}}(x), \tag{1}$$

where the symbol $*$ denotes convolution.

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

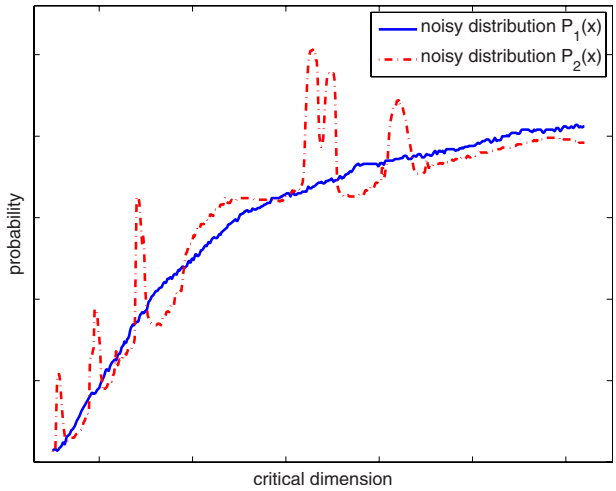

Fig. 7 Actual critical dimension distribution of a layout as determined by two simulators.[15]

Table 1 Eight cases of hotspot determination from two analyses with variability. A ✓ means a particular hotspot classification agrees with the truth; × means the reverse.

Label	Truth	Simulator 1	Simulator 2
(A)	Hotspot	✓	✓
(B)	Hotspot	✓	×
(C)	Hotspot	×	✓
(D)	Hotspot	×	×
(E)	Non-hotspot	✓	✓
(F)	Non-hotspot	✓	×
(G)	Non-hotspot	×	✓
(H)	Non-hotspot	×	×

The use of a random model for algorithm variability requires further justification here. It may be argued that since software is deterministic, all apparent variabilities must be due to immature or incorrect implementations that should be fixed. While it is true that the physical design is fixed (deterministic) and the LPC software is fixed (hence also deterministic), the interaction between the two (e.g., location and orientation of an instantiation) is not. Using the examples we gave in Sec. 2, a second instantiation of a layout being a rotation of the first one may affect where we start the OPC iteration, and a displacement that is not at integer multiples of the grid distance may cause different truncations in hierarchical operations. Therefore, from the LPC software's perspective, the data it works on contains random variability, and it is reasonable that its output does, too. In general, probability and statistics are ubiquitous tools for analyzing software for the same reason: The input data vary, but we need a way to characterize the performance of the software. Consequently, the observed critical dimension distribution exhibits effectively random variability with the true distribution.

Of particular interest is the situation in which two analyses are performed that result two noisy distributions $\mathcal{P}_1(x)$ and $\mathcal{P}_2(x)$. These two distributions are related to $\mathcal{P}_{\text{noiseless}}(x)$ via an expression similar to Eq. (1) with noise functions $f_1(x)$ and $f_2(x)$, respectively. If the two analyses are performed using different algorithms, $f_1(x)$ and $f_2(x)$ are most likely different functions. As an example using real data, Fig. 7 shows $\mathcal{P}_1(x)$ and $\mathcal{P}_2(x)$ resulting from two simulators on the same layout.[15]

Consequently, the two sets of hotspots from the two analyses are unlikely to be identical. Note that $f_1(x) = f_2(x)$ implies $\mathcal{P}_1(x) = \mathcal{P}_2(x)$, but even this case does not mean that the set of hotspots (with critical dimensions less than the threshold T) in $\mathcal{P}_1(x)$ and $\mathcal{P}_2(x)$ are the same. For each candidate hotspot, taking into account the "truth" (we put "truth" in quotes here because "truth" is something of which we have no knowledge, but this does not prevent an examination of the correlation between the two noisy distributions, as derived below) as well as reported results from the two analyses, there are altogether eight possibilities, as shown in Table 1. They are labeled from (A) to (H) for ease of reference in subsequent discussions. For cases (B), (C), and (D), depending on whether the first or second algorithm is used, a true hotspot will go undetected and hence unremedied. On the other hand, extraneous fixing may occur for cases (F), (G), and (H). Even if we perform LPC with two algorithms to safeguard against variability, in case (H), both analyses decide incorrectly that the same candidate is a hotspot. Consequently, we may expend a resource unnecessarily to rectify the extraneous hotspot. On the other hand, in case (D), both analyses decide incorrectly that the same candidate is not a hotspot. We may be blissfully unaware until we receive our wafers!

Given a simulated critical dimension x, let us define our hotspot criterion as

$$x \begin{cases} < T & \text{classified as hotspot,} \\ \geqslant T & \text{classified as non-hotspot,} \end{cases}$$

where T is the hotspot threshold. We can then derive the occurrence probability of the eight cases tabulated in Table 1 (denoted as $\mathcal{P}_A$ to $\mathcal{P}_H$, respectively) as follows:

$$\mathcal{P}_A = \int_{-\infty}^{T} \mathcal{P}_{\text{noiseless}}(x) g_1(T-x) g_2(T-x) dx,$$

$$\mathcal{P}_B = \int_{-\infty}^{T} \mathcal{P}_{\text{noiseless}}(x) g_1(T-x)[1-g_2(T-x)] dx,$$

$$\mathcal{P}_C = \int_{-\infty}^{T} \mathcal{P}_{\text{noiseless}}(x)[1-g_1(T-x)] g_2(T-x) dx,$$

$$\mathcal{P}_D = \int_{-\infty}^{T} \mathcal{P}_{\text{noiseless}}(x)[1-g_1(T-x)][1-g_2(T-x)] dx,$$

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

$$\mathcal{P}_E = \int_T^\infty \mathcal{P}_{\text{noiseless}}(x)[1 - g_1(T-x)][1 - g_2(T-x)]dx,$$

$$\mathcal{P}_F = \int_T^\infty \mathcal{P}_{\text{noiseless}}(x)[1 - g_1(T-x)]g_2(T-x)dx,$$

$$\mathcal{P}_G = \int_T^\infty \mathcal{P}_{\text{noiseless}}(x)g_1(T-x)[1 - g_2(T-x)]dx,$$

$$\mathcal{P}_H = \int_T^\infty \mathcal{P}_{\text{noiseless}}(x)g_1(T-x)g_2(T-x)dx,$$

where

$$g_{\{1,2\}}(x) = \int_{-\infty}^x f_{\{1,2\}}(x')dx'$$

is the cumulative noise distribution function, and

$$\mathcal{P}_A + \mathcal{P}_B + \mathcal{P}_C + \mathcal{P}_D + \mathcal{P}_E + \mathcal{P}_F + \mathcal{P}_G + \mathcal{P}_H = 1.$$

The sum $(\mathcal{P}_A+\mathcal{P}_H)$ accounts for the fraction of hotspots common to both analyses, $(\mathcal{P}_B+\mathcal{P}_G)$ is proportional to the number of hotspots found in the first analysis but not in the second, and $(\mathcal{P}_C+\mathcal{P}_F)$ denotes the probability of a candidate determined to be a hotspot in analysis 2 but not in analysis 1.

If we call the first analysis *reference* and the second *trial*, we can define the hotspot matching rate as

$$\mathcal{P}_{\text{matching}} = \frac{\text{number of common hotspots}}{\text{number of hotspots in reference}} = \frac{\mathcal{P}_A + \mathcal{P}_H}{\mathcal{P}_A + \mathcal{P}_B + \mathcal{P}_G + \mathcal{P}_H},$$

and the extra hotspot rate as

$$\mathcal{P}_{\text{extra}} = \frac{\text{number of trial hotspots not in reference}}{\text{number of hotspots in reference}} = \frac{\mathcal{P}_C + \mathcal{P}_F}{\mathcal{P}_A + \mathcal{P}_B + \mathcal{P}_G + \mathcal{P}_H}.$$

The missing hotspot probability is then

$$\mathcal{P}_{\text{missing}} = 1 - \mathcal{P}_{\text{matching}}.$$

3.2 *Ramifications*

Consider a few examples from the 65-nm technology that illustrate the consequence of algorithm variability in hotspot analysis. Assume a unitless hotspot threshold $T=100$, and that the hotspot distribution $\mathcal{P}_{\text{noiseless}}(x)$ around the threshold is a linearly increasing function of the critical dimension x, such as that shown in Fig. 6, with a slope of 1/7,200 and an x intercept of 80. This leads to a maximum critical dimension at $x=200$, since the total probabilities must sum to unity. Further suppose that variability is characterized by a Gaussian distribution, i.e., the noise functions $f_1(x)$ and $f_2(x)$ can be described by the standard deviations σ_1 and σ_2, respectively. The matching and extra rates of five case scenarios are evaluated below using numerical integration.

Case 1: $\sigma_1=\sigma_2=15$

A variation that amounts to 15% of the hotspot threshold results in a substantial mismatch between the results from the two analyses. The matching rate is 43.9%, while the extra rate is 56.1%. Note that in this case,

$$\mathcal{P}_{\text{missing}} = (1 - \mathcal{P}_{\text{matching}}) = \mathcal{P}_{\text{extra}}$$

because of the symmetry between the two analyses. The missing rate $(\mathcal{P}_B+\mathcal{P}_F)$ and the extra probability $(\mathcal{P}_C+\mathcal{P}_G)$ are equal because the two analyses have the same variability function.

Case 2: $\sigma_1=\sigma_2=5$

Compared with case 1, a smaller variability leads to better matching (73.4%) and less extra (26.6%). Nevertheless, at a variability level of 5% of the hotspot threshold, as many as one out of four hotspots are still mismatched. Note that we again have $\mathcal{P}_{\text{missing}}=\mathcal{P}_{\text{extra}}$ due to the symmetry of the two analyses.

Case 3: $\sigma_1=\sigma_2=5$; *increased trial threshold*

If it is desirable to increase the matching rate given a fixed level of variability, we can increase the hotspot threshold of the trial analysis. (Note that in practice, it is very difficult to change the threshold of the reference because the value of T is often calibrated and determined by actual patterning results.) Using the same variabilities as those in case 2, we increase the hotspot threshold of the trial analysis to 105, i.e., $T_1=100$ and $T_2=105$, and modify the calculations of $\mathcal{P}_A$ to $\mathcal{P}_H$ accordingly. With these modifications, it is possible to increase the matching rate to close to 90% at a significantly higher extra percentage of over 60%. If we desire a matching rate above 99%, we need to increase T_2 to 114. Consequently, the extra rate is as large as 178.8%!

Case 4: $\sigma_1=5$; $\sigma_2=15$

For the case in which variability of one analysis—say, the reference—is less than that of the second (the trial), the missing and extra rates are no longer the same. The results for $\sigma_1=5$ and $\sigma_2=15$ are tabulated in Table 2. The missing rate is less than the extra rate because the extra hotspots found by the first analysis $(\mathcal{P}_B+\mathcal{P}_G)$ are fewer than their counterparts in the second, which is a consequence of our assumption that distribution function $\mathcal{P}_{\text{noiseless}}(x)$ increases around the threshold. In other words, under the assumption, which is often true in practice, of an increasing distribution function $\mathcal{P}_{\text{noiseless}}(x)$ around the hotspot threshold, the matching and extra rates can be used to determine the relative noise levels of algorithms. If $\mathcal{P}_{\text{missing}}<\mathcal{P}_{\text{extra}}$, we can infer that the reference is closer to the truth. The trial has less variability if the inequality is reversed.

Case 5: $\sigma_1=0$; $\sigma_2=5$

Finally, consider the hypothetical case where the reference is identical to the "truth" and has no variability in the

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

Table 2 Numerical calculation of various cases of algorithm variability.

	Case 1	Case 2	Case 3a	Case 3b	Case 4	Case 5
$\mathcal{P}_A$	0.0128	0.0201	0.0228	0.0236	0.0160	0.0236
$\mathcal{P}_B$	0.0062	0.0035	0.0008	0.0000	0.0076	0.0053
$\mathcal{P}_C$	0.0062	0.0035	0.0049	0.0053	0.0030	0.0000
$\mathcal{P}_D$	0.0038	0.0018	0.0005	0.0000	0.0023	0.0000
$\mathcal{P}_E$	0.9299	0.9612	0.9517	0.9184	0.9441	0.9654
$\mathcal{P}_F$	0.0176	0.0042	0.0137	0.0470	0.0213	0.0057
$\mathcal{P}_G$	0.0176	0.0042	0.0023	0.0002	0.0034	0.0000
$\mathcal{P}_H$	0.0059	0.0014	0.0034	0.0055	0.0023	0.0000
matching rate	43.9%	73.4%	89.4%	99.2%	62.3%	81.5%
extra rate	56.1%	26.6%	63.5%	178.8%	82.9%	19.6%

hotspot analysis, i.e., $\sigma_1=0$. This is a way to characterize the "absolute" matching rate of a simulator. As shown in Table 2, where $\sigma_2=5$ and $T=100$, the matching rate is only 81.5% while the extra rate is 19.6%. This further underscores the challenge of demanding a high matching rate and a low extra rate simultaneously in the face of variability.

3.3 *Experimental Results*

Table 2 was computed using the analytic formulas given in Sec. 3.1. These numbers were verified by calculating the missing and extra rates using emulated data. In each trial, we generated thousands of CDs to examine where their values followed a linear distribution from $x=80$ to $x=200$, as discussed above. In passing these CDs to be examined by the simulators, we added a zero-mean Gaussian noise with standard deviations σ_1 and σ_2, respectively, to the CD values, to reflect the variability in the two simulators. Hotspot detection was simulated by thresholding the CD values with T. This process was repeated many times, and the missing and extra rates were recorded for each one.

Figure 8 shows a record of 500 trials for the case $\sigma_1=\sigma_2=5$ and $T=100$, which corresponds to case 2 in Sec. 3.2. The blue curve on top denotes the matching rate for each trial; its average, marked by the red line, is at 73.5%. The blue curve below denotes the extra rate for each trial and its average, marked by another red line, is at 26.7%. These two numbers are very close to those computed in Table 2.

The simulation also allowed us to explore relaxing the assumption of Gaussianity on the variability distribution. Instead of using a zero-mean Gaussian distribution, we can incorporate various distribution shapes by using a generalized Gaussian, i.e.,

$$f_{\text{noise}}(x)=\frac{\nu}{2\beta\Gamma(1/\nu)}\exp\left\{-\left(\frac{|x|}{\beta}\right)^{\nu}\right\}, \tag{2}$$

where $\nu>0$ controls the shape of the distribution, and β controls the spread.[16,17] The generalized Gaussian distribution includes several special cases: when $\nu=2$, it is the standard Gaussian distribution; when $\nu=1$, it becomes a Laplacian distribution; and when $\nu\to\infty$, it converges to the uniform distribution. Together they represent both fat-tail (leptokurtic) and thin-tail (platykurtic) distributions. We selected the three special cases, namely, Laplacian, Gaussian, and uniform distributions, for the distribution of simulator variability, then computed several cases that are tabulated in Table 3. While the Laplacian distributions generally gave a higher matching rate and a lower extra rate compared to

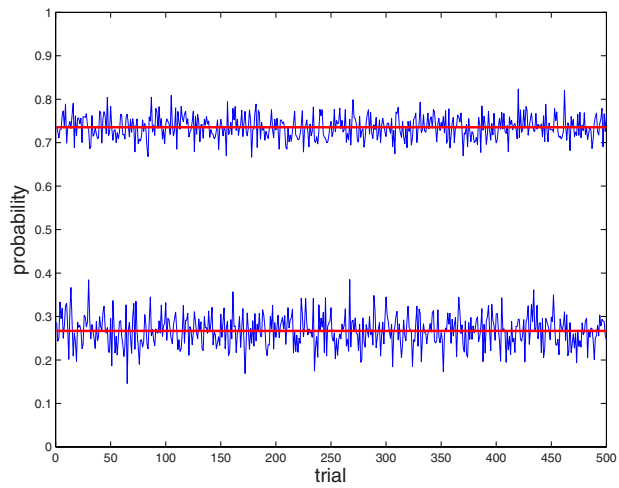

Fig. 8 Computation of the matching rate (curve on top) and the extra rate (curve on bottom) using emulated data.

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

Table 3 Numerical simulation of several variability distributions.

	Laplacian	Gaussian	Uniform
$\sigma_1=\sigma_2=15$:			
matching rate	45.0%	44.0%	44.0%
extra rate	55.1%	56.5%	56.1%
$\sigma_1=\sigma_2=5$:			
matching rate	74.9%	73.5%	72.6%
extra rate	24.8%	26.7%	27.3%
$\sigma_1=5$, $\sigma_2=15$:			
matching rate	65.3%	62.2%	29.8%
extra rate	77.6%	83.5%	87.3%
$\sigma_1=0$, $\sigma_2=5$:			
matching rate	85.2%	83.1%	81.4%
extra rate	21.0%	22.9%	24.7%

the Gaussian distribution, and a still lower matching rate and a higher extra rate were obtained for the uniform distribution, the results of the three distributions are similar. These results suggest that our analysis above is not very sensitive to the shape of the simulator variability distributions.

3.4 *Further Considerations*

In the preceding analyses, we assumed a linearly increasing hotspot distribution $\mathcal{P}_{\text{noiseless}}(x)$ at a particular slope and a threshold at $T=100$. It is natural to ask whether changing the slope or the threshold would affect the matching and extra rates. If so, the next question to ask is if there is an *optimal* distribution (not necessarily a linear function) that simultaneously maximizes the former and minimizes the latter.

The answer to the first question is affirmative. Suppose $\mathcal{P}_{\text{noiseless}}(x)$ has an x intercept of 90 and a maximum critical dimension at $x=150$. The resulting slope is then $1/1{,}800$—a four-fold increase from the earlier simulation. Using the equations for $\mathcal{P}_A$ to $\mathcal{P}_H$ in Sec. 3.1, for $\sigma_1=\sigma_2=5$ (similar to case 2 above), we find that the matching rate drops substantially to 54.6%, while the extra rate increases to 45.4%. Simulations of the other cases for other slopes revealed that an increase in slope led to a decrease in the matching rate and an increase in the extra rate.

We also investigated the cases where the slope was again $1/7{,}200$, and $\sigma_1=\sigma_2=5$, but the threshold was a variable. Again, the benchmark was case 2 in Table 2. When $T=95$, the matching rate was 66.1%; when $T=105$, the matching rate was 78.3%. (The extra rate is 1 minus the matching rate, as discussed earlier.) Therefore, we can argue that an increase in threshold leads to a better matching rate and a lower extra rate.

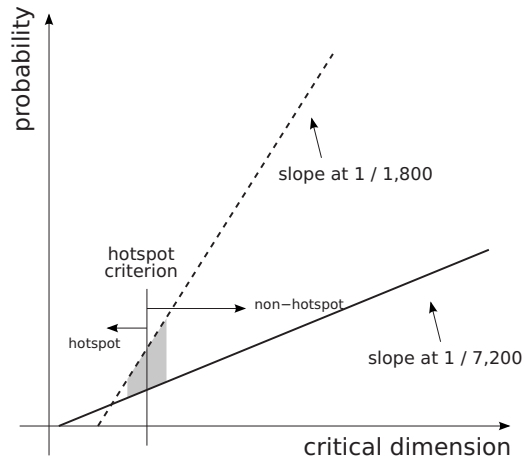

Fig. 9 Example showing the effect of increasing the slope of the distribution.

Closer examinations revealed that, there is a direct explanation for the above results. What contributed to the mismatch in the simulator results was mainly the proportion of critical dimensions around the threshold. The case for increasing the slope is shown in Fig. 9. The shaded area indicates the increase in the number of critical dimensions near the threshold, i.e., the likely candidates causing the mismatch. Note that in calculating the matching and extra rates, the number of hotspots in the reference must be normalized. In this case, this value did not change much as we increased the slope, so the extra critical dimensions contributed to the decrease in the matching rate (and correspondingly, the increase in the extra rate).

Likewise, we analyzed the effect of increasing the threshold, depicted in Fig. 10. The area shaded in light gray indicates the neighborhood of the original hotspot criterion, while the area shaded in dark gray is the neighborhood of the new criterion. Certainly the latter is larger. However, we had to normalize it with the proportion of hotspots to cal-

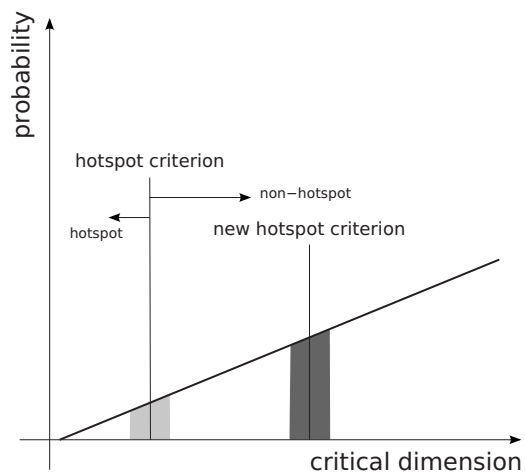

Fig. 10 Example showing the effect of increasing the hotspot criterion of the distribution.

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms

culate the matching and extra rates. Note that while the shaded area increases linearly with the hotspot criterion, the proportion of hotspots (the area bounded by the threshold) rises quadratically. Hence, the proportion of critical dimensions around the threshold actually decreased, leading to a decrease in the mismatch and therefore a better matching rate.

We also pinpointed the optimal distribution that maximized the matching rate and minimized the extra rate. This distribution should have as small an area in the neighborhood of the hotspot criterion as possible. In other words, to have two simulators delivering identical results, we want critical dimensions that are either clearly a hotspot or clearly not a hotspot! In hindsight this may sound trivial, but it helps to put our analysis into perspective since this scenario is clearly too idealistic in practice.

4 In Face of Variability

Variability should be addressed on both the algorithm and the application levels. For the former, we must improve algorithmic precision via examination and control of the root causes of variability. This is essentially an effort to shape the noise function $f_{\text{noise}}(x)$ in Sec. 3.1 closer to a delta function, i.e.,

$$f_{\text{noise}}(x) \to \delta(x).$$

At the same time, for each application we must set specifications that are at once commensurate with the limitations of the underlying algorithms and the goals of the particular application. For example, in a physical DFM application in which the hotspot analysis algorithm is distinct from the signoff verification engine, it is impractical to expect to fix all hotspots since, according to Sec. 3.2, unless we are willing to tolerate a large number of erroneous hotspots, 100% hotspot matching is highly unlikely.

5 Remarks

Computation lithography owes its current prevalence to algorithm innovations[18] and improvement in computer processing performance over the last few decades. In this virtual world, "realities" are tools that perform useful service for integrated circuit development. They are characterized by

1. specifications that make engineering sense, and
2. performance that meets the specification targets.[19]

"Virtuality" includes applications that fall short of one or both of the above two criteria. Thus, whether physical DFM—in particular, the hotspot analysis discussed in this paper—can be a reality depends on proper specifications in line with the algorithm variabilities, and efforts that reduce variability can help position this goal deeper in the realm of reality.

Acknowledgments

The authors express their sincere gratitude to Dr. Burn Lin and Dr. Ru-Gun Liu for their editorship of the present manuscript, and to Dr. C.-M. Lai for helpful comments on the simulation. The work described in this paper and its earlier conference version[20] was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (project number 713408), and by the University Research Committee of the University of Hong Kong (project number 10400898).

References

1. S. D. Shang, Y. Granik, N. B. Cobb, W. Maurer, Y. Cui, L. W. Liebmann, J. M. Oberschmidt, R. N. Singh, and B. R. Vampatella, "Failure prediction across process window for robust OPC," *Proc. SPIE* **5040**, 431–440 (2003).
2. J. Kim and M. Fan, "Hotspot detection on post-OPC layout using full-chip simulation-based verification tool: a case study with aerial image simulation," *Proc. SPIE* **5256**, 919–925 (2003).
3. L. Scheffer, L. Lavagno, and G. Martin, *Electronic Design Automation for Integrated Circuits Handbook*, CRC Press, Boca Raton, FL (2006).
4. K. N. Taravade, E. H. Croffie, and A. Jost, "Two-dimensional image-based model calibration for OPC applications," *Proc. SPIE* **5377**, 1522–1527 (2004).
5. S. Mansfield, G. Han, and L. Liebmann, "Through-process modeling for design-for-manufacturability applications," *J. Micro/Nanolith. MEMS MOEMS* **6**, 031007 (2007).
6. Y. Zhang, M. Feng, and H.-Y. Liu, "A focus exposure matrix model for full chip lithography manufacturability check and optical proximity correction," *Proc. SPIE* **6283**, 62830W (2006).
7. A. K. Wong, A. F. Molless, T. A. Brunner, E. Coker, R. H. Fair, G. L. Mack, and S. M. Mansfield, "Linewidth variation characterization by spatial decomposition," *J. Microlithogr., Microfabr., Microsyst.* **1**, 106–116 (2002).
8. A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography*, pp. 55–58, SPIE Press, Bellingham, WA (2001).
9. Y. Shen, N. Wong, and E. Y. Lam, "Level-set-based inverse lithography for photomask synthesis," *Opt. Express* **17**, 23690–23701 (2009).
10. N. Jia and E. Y. Lam, "Machine learning for inverse lithography: Using stochastic gradient descent for robust photomask synthesis," *J. Opt.* **12**, 045601 (2010).
11. N. B. Cobb and Y. Granik, "Dense OPC for 65 nm and below," *Proc. SPIE* **5992**, 599259 (2005).
12. Y. C. Pati, A. A. Ghazanfarian, and R. F. Pease, "Exploiting structure in fast aerial image computation for integrated circuit patterns," *IEEE Trans. Semicond. Manuf.* **10**, 62–74 (1997).
13. H. Gamo, "Matrix treatment of partial coherence," in *Progress in Optics*, Vol. **3**, E. Wolf, Ed., pp. 187–332, North-Holland, Amsterdam (1964).
14. T. Kong, H. Leung, V. Raghavan, A. Wong, and S. Xu, "Model-assisted routing for improved lithography robustness," *Proc. SPIE* **6521**, 65210D (2007).
15. Private communications with Dr. Ru-Gun Liu and Dr. Chih-Ming Lai, both of Taiwan Semiconductor Manufacturing Company (2010).
16. E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.* **9**, 1661–1666 (2000).
17. E. Y. Lam, "Analysis of the DCT coefficient distributions for document coding," *IEEE Signal Process. Lett.* **11**, 97–100 (2004).
18. A. Neureuther, "If it moves, simulate it!," *Proc. SPIE* **6924**, 692402 (2008).
19. E. Y. Lam and A. K. Wong, "Computation lithography: Virtual reality and virtual virtuality," *Opt. Express* **17**, 12259–12268 (2009).
20. A. K. Wong and E. Y. Lam, "The nebulous hotspot and algorithm variability," *Proc. SPIE* **7275**, 727509 (2009).

Edmund Y. Lam received the BS (with distinction), MS, and PhD degrees in electrical engineering from Stanford University. He was a senior engineer at KLA-Tencor Corporation in San Jose, California, before joining the Electrical and Electronic Engineering Department at the University of Hong Kong, where he is currently an associate professor. He is also the founding director of its Imaging Systems Laboratory. His research interests include computational optics and imaging, particularly their applications in the semiconductor manufacturing process. He has been a program committee member of the SPIE Lithography Asia conference since its inception, and is a senior member of SPIE.

Biography and photograph of the second author is not available.

Downloaded from SPIE Digital Library on 03 Feb 2012 to 147.8.21.59. Terms of Use: http://spiedl.org/terms