Fast Lithography Image Simulation By Exploiting Symmetries in Lithography Systems

Peng Yu, Weifeng Qiu and David Z. Pan

***Abstract*—Lithography simulation has been widely used in many applications, such as optical proximity correction, in semiconductor industry. It is important to reduce the runtime of such simulations. Dedicated hardware and parallel compuatation have been used to reduce the runtime. For full chip simulation, the simulation method, Optimal Coherent Approximations (OCA's), is widely used. But it has not been improved since its first inception. In this paper, we improve it by considering the symmetric properties of lithography systems. The new method could speed up the runtime by $2\times$ without loss of accuracy. We demonstrate the speedup is applicable to vectorial imaging model as well. In case the symmetric properties do not hold strictly, the new method can be generalized such that it could still be faster than the old method.**

I. Introduction

Lithography image simulation as a step in lithography simulation is widely used in applications such as Optical Proximity Correction (OPC) [1]. The industry has pushed hard to reduce the simulation runtime by using parallel computation and dedicated hardware. Mentor Graphics has used multiprocessing and multithreading on Linux workstation clusters [2]. Specific hardware-accelerated computational lithography platform has also been used in the industry [3] IBM has developed software for IC design and DFM software with the IBM's BlueGene supercomputer [4]. These efforts are very important but require huge software and hardware investments.

Optimal Coherent Approximations (OCA's) [5], [6] is used for the full-chip image simulation. In this work, we improve this method by exploiting the symmetric properties that are commonly found in lithography systems. The new method could give a speedsup of $2\times$ without any loss in accuracy. Such improvement can be easily integrated with the other speedup techniques mentioned above.

The main contributions of this paper are as follows:

- We derive that the well known Hopkins equation can be reduced such that the imaginary part of the transmission cross coefficient is not necessary.
- We derive an improved image simulation formula, which could give a $2\times$ speedup without any accuracy loss by using the symmetric properties in common lithography systems.
- It works for both scalar and vectorial imaging model.
- The new method still improves runtime even if lithography systems are not perfectly symmetric.

Peng Yu and David Z. Pan are with the Dept. of Electrical and Computer Engineering, the University of Texas at Austin.

Weifeng Qiu is with the Institute for Computational Engineering and Sciences, the University of Texas at Austin.

The rest of this paper is organized as follows. In Section II, we review the scalar imaging model and the symmetric properties commonly found in lithography systems. In Section III, we derive the properties of the transmission cross coefficient (TCC) and the reduced Hopkins equation. In Section IV, we show the new simulation formula. The detailed derivations can be found in the Appendix. In Section V, we show that the new formula works for both vectorial models and non-perfectly symmetric lithography systems. Section VI shows the experiment results. Section VII concludes this paper.

II. Lithography Simulation Review and Symmetries in Lithography System

We review the scalar lithography imaging model. The vectorial model will be discussed in Section V. We then point out some symmetric properties commonly found in lithography systems, which be used for the derivation of the new formula.

The latent image intensity in the photoresist is given by the Hopkins equation [7],

$$\mathcal{I}(\boldsymbol{k}) = \iint \mathcal{T}(\boldsymbol{k}+\boldsymbol{k}', \boldsymbol{k}')\mathcal{F}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{F}^*(\boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}'. \quad (1)$$

$\mathcal{F}(\boldsymbol{k})$ is the mask transmission function $F(\boldsymbol{r})$ in the frequency domain, where $\boldsymbol{k}$ denotes a point in the frequency domain and $\boldsymbol{r}$ denotes a point in the spatial domain. $\mathcal{I}(\boldsymbol{k})$ is the chemical latent image in the Fourier domain. $\mathcal{T}(\boldsymbol{k}, \boldsymbol{k}')$ is the *transmission cross coefficient* (TCC) (including diffusion, see below), given by

$$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') = \mathcal{G}(\boldsymbol{k}' - \boldsymbol{k}'') \times \iint \mathcal{J}(\boldsymbol{k})\mathcal{K}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{K}^*(\boldsymbol{k}+\boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k}. \quad (2)$$

The meaning of the symbols are described below:

- $\mathcal{G}(\boldsymbol{k})$ is the diffusion kernel, written as
$$\mathcal{G}(\boldsymbol{k}) = e^{-2\pi^2 d^2 k^2}, \quad (3)$$
which corresponds the diffusion of the latent image during the post-exposure-bake (PEB), where d is the diffusion length, and $k = |\boldsymbol{k}|$.
- $\mathcal{J}(\boldsymbol{k})$ is the illumination function. We illustrate some commonly used ones in Figure 1.
- $\mathcal{K}(\boldsymbol{k})$ is the projection system transfer function. Assuming a circular pupil, it can be written as
$$\mathcal{K}(\boldsymbol{k}) = \begin{cases} e^{i\frac{2\pi}{\lambda}z\sqrt{1-k^2\sin^2\theta_{\text{obj}}}+i2\pi\Phi(\boldsymbol{k})} & k<1 \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

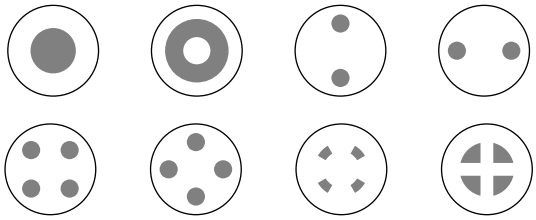


Figure 1. Commonly used illumination schemes. The radii of the outer circles are 1. $\mathcal{J}$ is a constant over the gray regions.

where z denotes the focus error, λ is the wavelength and θ_{obj} is the semi-aperture angle at the image plane [8] and $\Phi(\boldsymbol{k})$ is the aberration term.

- The superscript * denotes the complex conjugation operation.

Based on Eq. (3), Figure 1 and Eq. (4), it is obvious that $\mathcal{G}(\boldsymbol{k})$, $\mathcal{J}(\boldsymbol{k})$ and $\mathcal{K}(\boldsymbol{k})$ have the following two properties.

Property 1.

$$\mathcal{G}(\boldsymbol{k}) \in \mathbb{R},\ \mathcal{J}(\boldsymbol{k}) \in \mathbb{R} \text{ and } \mathcal{K}(\boldsymbol{k}) \in \mathbb{C}, \tag{5}$$

where $\mathbb{R}$ and $\mathbb{C}$ denote the set of all real numbers and all complex numbers, respectively.

Property 2. *It is reasonable to assume that the diffusion is rotational invariant. Then we have*

$$\mathcal{G}(\boldsymbol{k}) = \mathcal{G}(-\boldsymbol{k}). \tag{6}$$

Since symmetrical illumination schemes are commonly used (not necessary limited to the ones in Figure 1), we have

$$\mathcal{J}(\boldsymbol{k}) = \mathcal{J}(-\boldsymbol{k}). \tag{7}$$

Assuming no odd order aberrations, we have

$$\mathcal{K}(\boldsymbol{k}) = \mathcal{K}(-\boldsymbol{k}). \tag{8}$$

Remark. *These properties are true in a general sense and are not necessarily specific only to the forms in Eq.* (3)*, Figure 1 and Eq.* (4)*.*

The mask transmission function $F(\boldsymbol{r})$ is real for commonly used masks, such as binary mask (BIM) or phase shift mask (PSM) with the phases of 0° and 180°. By the definition of the Fourier Transform, we can easily prove the following property of $F(\boldsymbol{r})$'s inverse Fourier Transform $\mathcal{F}(\boldsymbol{k})$.

Property 3.

$$\mathcal{F}(\boldsymbol{k}) = \mathcal{F}^*(-\boldsymbol{k}). \tag{9}$$

III. The Reduced Hopkins Equation

In this section, we prove a few lemmas on TCC and derive a reduced Hopkins Equation.

With Property 1, we have the following lemma.

Lemma 1. *$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'')$ is Hermitian, that is,*

$$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') = \mathcal{T}^*(\boldsymbol{k}'', \boldsymbol{k}'). \tag{10}$$

Proof:

$$\begin{aligned}
&\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') \\
&= \mathcal{G}(\boldsymbol{k}' - \boldsymbol{k}'') \iint \mathcal{J}(\boldsymbol{k})\mathcal{K}(\boldsymbol{k} + \boldsymbol{k}')\mathcal{K}^*(\boldsymbol{k} + \boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k} \\
&= \Big(\mathcal{G}(\boldsymbol{k}' - \boldsymbol{k}'') \iint \mathcal{J}(\boldsymbol{k})\mathcal{K}^*(\boldsymbol{k} + \boldsymbol{k}')\mathcal{K}(\boldsymbol{k} + \boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k}\Big)^* \\
&= \mathcal{T}^*(\boldsymbol{k}'', \boldsymbol{k}')
\end{aligned} \tag{11}$$

■

With Property 2, we have the following lemma.

Lemma 2. *$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'')$ is symmetric under the reflection operation about the origin in the frequency domain,*

$$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') = \mathcal{T}(-\boldsymbol{k}', -\boldsymbol{k}''). \tag{12}$$

Proof:

$$\begin{aligned}
&\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') \\
&= \mathcal{G}(\boldsymbol{k}' - \boldsymbol{k}'') \iint \mathcal{J}(\boldsymbol{k})\mathcal{K}(\boldsymbol{k} + \boldsymbol{k}')\mathcal{K}^*(\boldsymbol{k} + \boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k} \\
&= \mathcal{G}\big((-\boldsymbol{k}') - (-\boldsymbol{k}'')\big) \iint \mathcal{J}(-\boldsymbol{k})\mathcal{K}(-\boldsymbol{k} - \boldsymbol{k}')\mathcal{K}^*(-\boldsymbol{k} - \boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k} \\
&= \mathcal{G}\big((-\boldsymbol{k}') - (-\boldsymbol{k}'')\big) \iint \mathcal{J}(\boldsymbol{k})\mathcal{K}(\boldsymbol{k} - \boldsymbol{k}')\mathcal{K}^*(\boldsymbol{k} - \boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k} \\
&= \mathcal{T}(-\boldsymbol{k}', -\boldsymbol{k}'')
\end{aligned} \tag{13}$$

■

Remark. *The proofs of Lemma 1 and 2 do not use the particular function forms of $\mathcal{G}$, $\mathcal{J}$ and $\mathcal{K}$ but only Property 1 and 2. These conclusions are generally true for common lithography systems.*

With Lemma 1 and 2, we immediately have the following corollary.

Corollary 1.

$$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') = \mathcal{T}^*(-\boldsymbol{k}'', -\boldsymbol{k}'). \tag{14}$$

Using Property 3, we have the following lemma.

Lemma 3. *If $\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') = -\mathcal{T}(-\boldsymbol{k}'', -\boldsymbol{k}')$, we have $\mathcal{I}(\boldsymbol{k}) = 0$.*

Proof: We prove the lemma by proving

$$\mathcal{I}(\boldsymbol{k}) = -\mathcal{I}(\boldsymbol{k}). \tag{15}$$

$$\begin{aligned}
\mathcal{I}(\boldsymbol{k}) &= \iint \mathcal{T}(\boldsymbol{k} + \boldsymbol{k}', \boldsymbol{k}')\mathcal{F}(\boldsymbol{k} + \boldsymbol{k}')\mathcal{F}^*(\boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}' \\
&= -\iint \mathcal{T}(-\boldsymbol{k}', -\boldsymbol{k} - \boldsymbol{k}')\mathcal{F}(-\boldsymbol{k}')\mathcal{F}^*(-\boldsymbol{k} - \boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}' \\
&= -\iint \mathcal{T}(\boldsymbol{k} + \boldsymbol{k}', \boldsymbol{k}')\mathcal{F}(\boldsymbol{k} + \boldsymbol{k}')\mathcal{F}^*(\boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}' \\
&= -\mathcal{I}(\boldsymbol{k}).
\end{aligned} \tag{16}$$

We replace $-\boldsymbol{k} - \boldsymbol{k}'$ by $\boldsymbol{k}'$ to get the third integral. ■

$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'')$, as a complex function, can be separated into a real part ($\mathcal{T}_{\text{real}}(\boldsymbol{k}', \boldsymbol{k}'') \in \mathbb{R}$) and an imaginary part ($\mathcal{T}_{\text{imag}}(\boldsymbol{k}', \boldsymbol{k}'') \in \mathbb{R}$),

$$\mathcal{T}(\boldsymbol{k}', \boldsymbol{k}'') = \mathcal{T}_{\text{real}}(\boldsymbol{k}', \boldsymbol{k}'') + i\mathcal{T}_{\text{imag}}(\boldsymbol{k}', \boldsymbol{k}''). \tag{17}$$

$\mathcal{I}(\boldsymbol{k})$ can be separated accordingly as

$$\mathcal{I}(\boldsymbol{k}) = \mathcal{I}_{\text{real}}(\boldsymbol{k}) + i\mathcal{I}_{\text{imag}}(\boldsymbol{k}), \tag{18}$$

where

$$\mathcal{I}_{\text{real}}(\boldsymbol{k}) = \iint \mathcal{T}_{\text{real}}(\boldsymbol{k}+\boldsymbol{k}', \boldsymbol{k}')\mathcal{F}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{F}^*(\boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}' \tag{19}$$

and

$$\mathcal{I}_{\text{imag}}(\boldsymbol{k}) = \iint \mathcal{T}_{\text{imag}}(\boldsymbol{k}+\boldsymbol{k}', \boldsymbol{k}')\mathcal{F}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{F}^*(\boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}'. \tag{20}$$

By Corollary 1, we have $\mathcal{T}_{\text{real}}(\boldsymbol{k}', \boldsymbol{k}'')$ and $\mathcal{T}_{\text{imag}}(\boldsymbol{k}', \boldsymbol{k}'')$ are symmetric and antisymmetric, respectively,

$$\mathcal{T}_{\text{real}}(\boldsymbol{k}', \boldsymbol{k}'') = \mathcal{T}_{\text{real}}(-\boldsymbol{k}'', -\boldsymbol{k}') \tag{21}$$

and

$$\mathcal{T}_{\text{imag}}(\boldsymbol{k}', \boldsymbol{k}'') = -\mathcal{T}_{\text{imag}}(-\boldsymbol{k}'', -\boldsymbol{k}'). \tag{22}$$

Using Lemma 3, we have $\mathcal{I}_{\text{imag}}(\boldsymbol{k}) = 0$. Therefore, we have the following theorem.

Theorem 1. *The image can be computed by the* Reduced Hopkins Equation*,*

$$\mathcal{I}(\boldsymbol{k}) = \iint \mathcal{T}_{real}(\boldsymbol{k}+\boldsymbol{k}', \boldsymbol{k}')\mathcal{F}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{F}^*(\boldsymbol{k}')\mathrm{d}^2\boldsymbol{k}'. \tag{23}$$

Using the above theorem, we can derive the speedup simulation formula in the next section.

IV. Improved Optimal Coherent Approximations

Optimal Coherent Approximations (OCA's) has been derived in [5], which shows that the image can be computed by

$$I(\boldsymbol{r}) = \sum_{n=0}^{\infty} \sigma_n \left| Q_n \ast\ast F \right|^2, \tag{24}$$

where F is the mask transmission function, $\ast\ast$ is the convolution operator and Q_n's (called kernels) are complex functions. The real numbers σ_n's are ordered such that

$$|\sigma_0| \geqslant |\sigma_1| \geqslant \cdots \geqslant |\sigma_n| \geqslant \cdots. \tag{25}$$

An image can be approximated by using only the first a few terms,

$$\widetilde{I}_p(\boldsymbol{r}) = \sum_{n=0}^{p-1} \sigma_n \left| Q_n \ast\ast F \right|^2. \tag{26}$$

The error can be estimated as

$$\sup_{\boldsymbol{r}} |I(\boldsymbol{r}) - \widetilde{I}_p(\boldsymbol{r})| \leqslant \sigma_p \|F\|_2^2, \tag{27}$$

where $\|\cdot\|_2$ denotes the L^2-norm.

However, Theorem 1 was not used in [5]. When we use Theorem 1, we can prove in Appendix that the operator $|\cdot|$ is not needed. That is, instead of (24), we have

$$I(\boldsymbol{r}) = \sum_{n=0}^{\infty} \sigma'_n (Q'_n \ast\ast F)^2, \tag{28}$$

where F and Q'_n's are all real, and

$$|\sigma'_0| \geqslant |\sigma'_1| \geqslant \cdots \geqslant |\sigma'_n| \geqslant \cdots. \tag{29}$$

Note that σ_n and σ'_n, and Q_n and Q'_n may not be the same. As an approximation, we also take the first a few terms for the image simulation

$$\widetilde{I}'_{p'}(\boldsymbol{r}) = \sum_{n=0}^{p'-1} \sigma'_n (Q'_n \ast\ast F)^2, \tag{30}$$

Similarly, the error can be estimated as

$$\sup_{\boldsymbol{r}} |I(\boldsymbol{r}) - \widetilde{I}'_{p'}(\boldsymbol{r})| \leqslant \sigma'_{p'} \|F\|_2^2. \tag{31}$$

Based on Eq. (27) and Eq. (31), we can choose the numbers of terms that are needed (p and p') for a given error requirement ϵ.

It is obvious that the TCC $\mathcal{T}$ is real when there are no aberrations ($z = 0$ and $\Phi(\boldsymbol{k}) = 0$). In this case, we have

$$Q_n = Q'_n \quad \text{and} \quad \sigma_n = \sigma'_n \tag{32}$$

for any n. Therefore, the same numbers of terms ($p = p'$) are needed for the same error requirement ϵ. Since the convolution of a complex function (Q) with a real function (F) is $2\times$ slower than the convolution of two real functions (Q' and F), we get a $2\times$ speedup. We can easily see that the approximated images are always the same

$$\widetilde{I}_p(\boldsymbol{r}) = \widetilde{I}'_{p'}(\boldsymbol{r}).$$

Therefore, there is no loss in the accuracy from (26) to (30). We have the following corollary.

Corollary 2. *When there are no aberrations,* (28) *gives a* $2\times$ *speedup without loss of accuracy.*

Otherwise the speedup is

$$s = \frac{2p}{p'}. \tag{33}$$

In Section VI, we show experimentally the speedup for some other cases.

V. Extensions to Vectorial Imaging and Non-Perfect Symmetries

The above improvement was shown for scalar image modeling with perfect symmetries. We will show below it works for vectorial image modeling [8] and non-perfect symmetric lithography systems.

A. Vectorial Imaging

According to [8], the TCC in the scalar model becomes a TCC matrix in the vectorial model, which can be written as

$$\mathcal{T}_{ij}(\boldsymbol{k}',\boldsymbol{k}'') = \iint \mathcal{J}(\boldsymbol{k})\mathcal{K}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{K}^*(\boldsymbol{k}+\boldsymbol{k}'') \sum_{k=\{\mathrm{x,y,z}\}} \mathcal{M}_{ki}(\boldsymbol{k}+\boldsymbol{k}')\mathcal{M}^*_{kj}(\boldsymbol{k}+\boldsymbol{k}'')\mathrm{d}^2\boldsymbol{k},$$

where

$$\boldsymbol{\mathcal{M}}_0(\boldsymbol{k}) = \boldsymbol{\mathcal{M}}_0(f,g) = \begin{pmatrix} \mathcal{M}_{0\mathrm{xx}} & \mathcal{M}_{0\mathrm{yx}} \\ \mathcal{M}_{0\mathrm{xy}} & \mathcal{M}_{0\mathrm{yy}} \\ \mathcal{M}_{0\mathrm{xz}} & \mathcal{M}_{0\mathrm{yz}} \end{pmatrix} = \begin{pmatrix} \frac{\beta^2+\alpha^2\gamma}{1-\gamma^2} & -\frac{\alpha\beta}{1+\gamma} \\ -\frac{\alpha\beta}{1+\gamma} & \frac{\alpha^2+\beta^2\gamma}{1-\gamma^2} \\ -\alpha & -\beta \end{pmatrix}$$

and

$$\alpha = f\sin\theta_{\mathrm{obj}},$$
$$\beta = g\sin\theta_{\mathrm{obj}},$$
$$\gamma = \sqrt{1-(f^2+g^2)\sin^2\theta_{\mathrm{obj}}}.$$

For unpolarized illumination, an equivalent TCC can be written as [9], [10]

$$\mathcal{T} = \mathcal{T}_{xx} + \mathcal{T}_{yy}. \tag{34}$$

Similar to Lemma 1 and 2, it is easy to check that $\mathcal{T}$ in Eq. (34) is Hermitian

$$\mathcal{T}(\boldsymbol{k}',\boldsymbol{k}'') = \mathcal{T}^*(\boldsymbol{k}'',\boldsymbol{k}')$$

and symmetric under the reflection operation about the origin

$$\mathcal{T}(\boldsymbol{k}',\boldsymbol{k}'') = \mathcal{T}(-\boldsymbol{k}',-\boldsymbol{k}'').$$

Therefore, Eq. (30) is valid for vectorial imaging as well.

B. Non-Perfect Symmetries

Practically, lithography systems may not be perfectly symmetric due to some errors (Property 1 and 2 may not hold perfectly). But we can separate $\mathcal{T}(\boldsymbol{k}',\boldsymbol{k}'')$ into two parts as

$$\mathcal{T}(\boldsymbol{k}',\boldsymbol{k}'') = \mathcal{T}_{\mathrm{sym}}(\boldsymbol{k}',\boldsymbol{k}'') + \mathcal{T}_{\mathrm{anti}}(\boldsymbol{k}',\boldsymbol{k}''), \tag{35}$$

where

$$\mathcal{T}_{\mathrm{sym}}(\boldsymbol{k}',\boldsymbol{k}'') = \frac{\mathcal{T}(\boldsymbol{k}',\boldsymbol{k}'') + \mathcal{T}^*(-\boldsymbol{k}'',-\boldsymbol{k}')}{2} \tag{36}$$

and

$$\mathcal{T}_{\mathrm{anti}}(\boldsymbol{k}',\boldsymbol{k}'') = \frac{\mathcal{T}(\boldsymbol{k}',\boldsymbol{k}'') - \mathcal{T}^*(-\boldsymbol{k}'',-\boldsymbol{k}')}{2}. \tag{37}$$

It is easy to check that $\mathcal{T}_{\mathrm{sym}}$ is symmetric

$$\mathcal{T}_{\mathrm{sym}}(\boldsymbol{k}',\boldsymbol{k}'') = \mathcal{T}^*_{\mathrm{sym}}(-\boldsymbol{k}'',-\boldsymbol{k}') \tag{38}$$

and $\mathcal{T}_{\mathrm{anti}}$ is antisymmetric

$$\mathcal{T}_{\mathrm{anti}}(\boldsymbol{k}',\boldsymbol{k}'') = -\mathcal{T}^*_{\mathrm{anti}}(-\boldsymbol{k}'',-\boldsymbol{k}'). \tag{39}$$

Similar to the deduction of Theorem 1, we only need the real part of $\mathcal{T}_{\mathrm{sym}}(\boldsymbol{k}',\boldsymbol{k}'')$ and the imaginary part of $\mathcal{T}_{\mathrm{anti}}(\boldsymbol{k}',\boldsymbol{k}'')$ for the computation of $\mathcal{I}(\boldsymbol{k})$.

Assume p terms are needed to decompose $\mathcal{T}$, q_1 terms are needed to decompose $\mathcal{T}_{\mathrm{sym,real}}$ and q_2 terms are needed to decompose $\mathcal{T}_{\mathrm{anti,imag}}$. Therefore, the runtime speedup is

$$s = \frac{2p}{q_1+q_2}. \tag{40}$$

If lithography systems are close to symmetric, we have that $\mathcal{T}_{\mathrm{sym}}$ is close to $\mathcal{T}$ and $\mathcal{T}_{\mathrm{anti}}$ is small. Therefore, p is close to q_1 and q_2 is much smaller than p. In this case, the speedup is close to 2.

VI. Experimental Results

In this section, we numerically validate our previous statements. The implementations were in C++ [11], and simulations were on a 2.8 GHz Pentinum-4 Linux machine.

We used the conventional partially coherent illumination with $\sigma = 0.7$, the numerical aperture $\mathrm{NA} = 0.8$, the wavelength $\lambda = 193\,\mathrm{nm}$ and the defocus $z = 100\,\mathrm{nm}$ unless otherwise noted.

A. Validation of TCC's symmetrical property

For the properties of TCC, we only numerically validate the symmetric property (Lemma 2), because the Hermitian property (Lemma 1) is well known.

We denote $\mathcal{T}(\boldsymbol{k},\boldsymbol{k}')$ as $\mathcal{T}(f,g,f',g')$, where $(f,g)=\boldsymbol{k}$ and $(f',g')=\boldsymbol{k}'$ to simplify the discussions below. It is easy to check that

$$\mathcal{T}(f,g,f',g') = 0, \quad \text{for} \quad (f,g,f',g') \notin \mathbb{B}, \tag{41}$$

where $\mathbb{B}$ is a 4-dimensional box

$$\mathbb{B} = (-1-\sigma, 1+\sigma)^4. \tag{42}$$

We numerically simulate the TCC on the all points

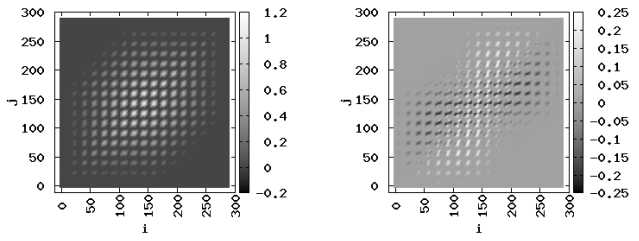
$$(i_1,j_1,i_2,j_2)\Delta \in \mathbb{B}, \tag{43}$$

where the grid size in the frequency domain $\Delta = 0.2$, and the numbers i_1, j_1, i_2 and j_2 are integers in the interval $[-N,N]$, where $N = \lfloor \frac{1+\sigma}{\Delta} \rfloor = 8$.

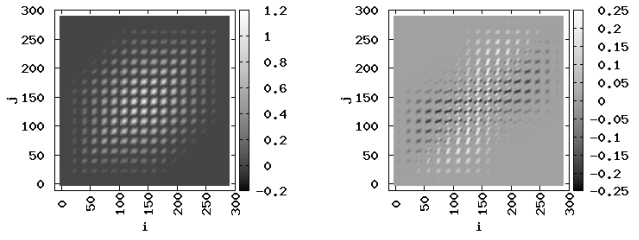
$\mathcal{T}(f,g,f',g')$ is a 4-dimensional function, which need to be reindexed to draw 2-dimensionally. We denote $\mathcal{T}(i_1\Delta, j_1\Delta, i_2\Delta, j_2\Delta)$ as $\mathcal{T}_{ij}$, where

$$\begin{cases} i = (i_1+N) + (2N+1)(i_2+N), \\ j = (j_1+N) + (2N+1)(j_2+N) \end{cases} \tag{44}$$

to help visualize the TCC [1]. The indexes i and j are in $[0,(2N+1)^2-1] = [0,288]$. We also denote $\mathcal{T}(-i_1\Delta,-j_1\Delta,-i_2\Delta,-j_2\Delta)$ as $\widetilde{\mathcal{T}}_{ij}$ We use $\Re$ and $\Im$ to denote the real part and the imaginary part. Figure 2 and 3 show $\Re(T_{ij})$, $\Im(T_{ij})$, $\Re(\widetilde{T}_{ij})$ and $\Im(\widetilde{T}_{ij})$ for both the scalar model and the vectorial model. From these two figures, it is clear that $T_{ij} = \widetilde{T}_{ij}$ for both models. Therefore, Lemma 2 is validated.

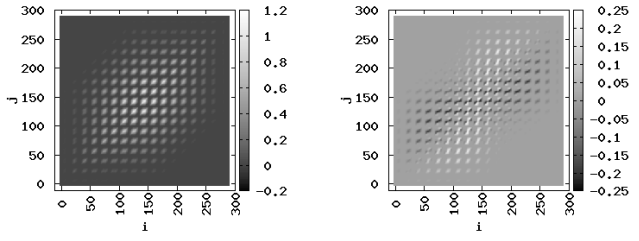


(a) $\Re(\mathcal{T}_{ij})$ (b) $\Im(\mathcal{T}_{ij})$

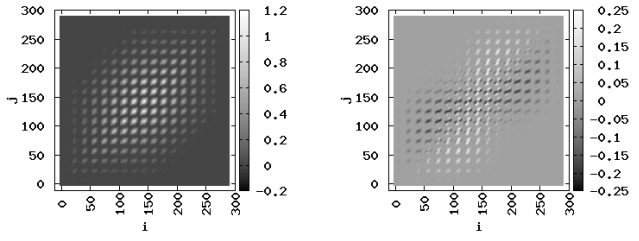


(c) $\Re(\widetilde{\mathcal{T}}_{ij})$ (d) $\Im(\widetilde{\mathcal{T}}_{ij})$

Figure 2. Visualization of $\mathcal{T}(\boldsymbol{k}, \boldsymbol{k}')$ and $\mathcal{T}(-\boldsymbol{k}, -\boldsymbol{k}')$ of the *scalar* model ($z = 100\,\mathrm{nm}$). Subfigure (a) and (c) are the same, and Subfigure (b) and (d) are the same. Therefore, $\mathcal{T}(\boldsymbol{k}, \boldsymbol{k}') = \mathcal{T}(-\boldsymbol{k}, -\boldsymbol{k}')$.

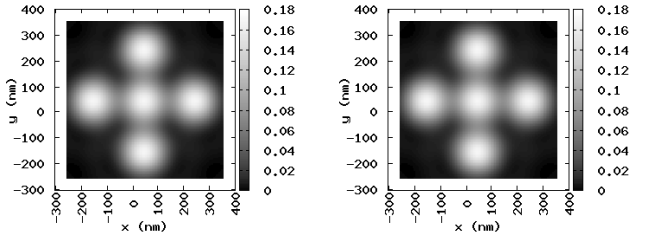


(a) $\Re(\mathcal{T}_{ij})$ (b) $\Im(\mathcal{T}_{ij})$



(c) $\Re(\widetilde{\mathcal{T}}_{ij})$ (d) $\Im(\widetilde{\mathcal{T}}_{ij})$

Figure 3. Visualization of $\mathcal{T}(\boldsymbol{k}, \boldsymbol{k}')$ and $\mathcal{T}(-\boldsymbol{k}, -\boldsymbol{k}')$ of the *vectorial* model ($z = 100\,\mathrm{nm}$). Subfigure (a) and (c) are the same, and Subfigure (b) and (d) are the same. Therefore, $\mathcal{T}(\boldsymbol{k}, \boldsymbol{k}') = \mathcal{T}(-\boldsymbol{k}, -\boldsymbol{k}')$.



(a) Using Hopkins Equation (b) Using Reduced Hopkins Equation

Figure 4. The simulated image for a five-via pattern. Each via is of size $100\,\mathrm{nm}$. The distance between the center via and any other via is $100\,\mathrm{nm}$.

B. Validation of the Reduced Hopkins Equation

Figure 4 shows the simulated image using the scalar model for a five-via pattern using Hopkins Equation and Reduced Hopkins Equation. The maximum image difference between these two images is $1.387\,78 \times 10^{-16}$, which is numerically zero. Therefore, we verified the Reduced Hopkins Equation for the scalar model. The Reduced Hopkins Equation for the vectorial model can also be verified.

C. Runtime Speedup

Figure 5 shows the numbers of terms p and p', and their ratio as a function of the error requirement ϵ for $z = 100\,\mathrm{nm}$ and $z = 200\,\mathrm{nm}$, respectively (the scalar model). The experiments show that the runtime speedup is 2 for $z = 0\,\mathrm{nm}$. When $z = 100\,\mathrm{nm}$ and $z = 200\,\mathrm{nm}$, The speedup can be bigger than 2 for some ϵ. In the worst, the speedup is approximately 1.2.

D. Non-Perfect Symmetries

When there are odd aberrations, Eq. (8) does not hold. Let us consider a small x-coma aberration ($z = 0\,\mathrm{nm}$). The x-coma aberration term $\Phi(\boldsymbol{k})$ is [8]

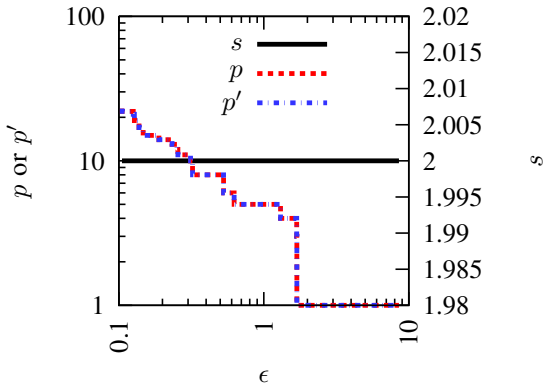
$$\Phi(\boldsymbol{k}) = \Phi(f, g) = c2\sqrt{2}(3k^2 - 2)f$$

where c is a coefficient (we take it as a small number, 0.01) Figure 6 shows the numbers of terms p and p', and their ratio as a function of the error requirement ϵ (the scalar model).

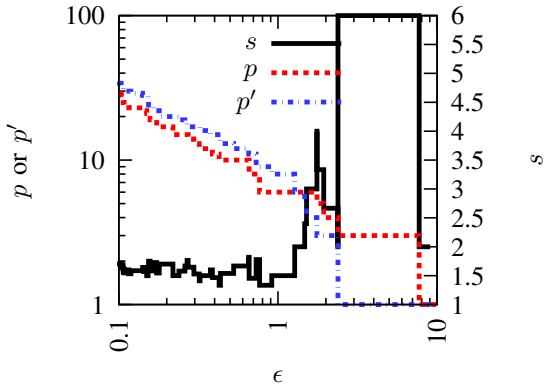
The speedup s varies as the error requirement ϵ changes. But in the worst case, the speedup is approximately 1.2. The speedup can be bigger than 2.

VII. Conclusions

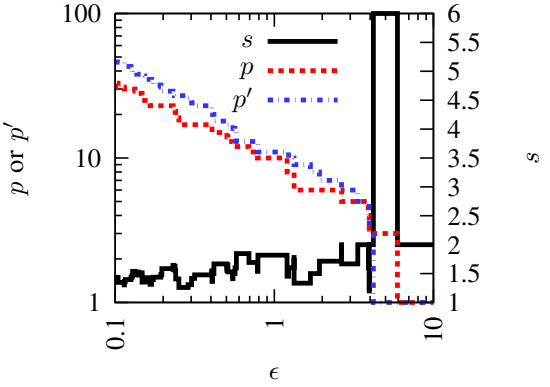
In this paper, we derive a new method for the lithography simulation, which speeds up the widely used method (OCA's) using the symmetric properties of the lithography imaging system. It can give $2\times$ speedup if there are no aberrations. It works for both the scalar and the vectorial model. The new method still gives speedup when lithography imaging systems are not perfectly symmetric.



(a) $z = 0\,\mathrm{nm}$. The curves for p and p' are the same.



(b) $z = 100\,\mathrm{nm}$



(c) $z = 200\,\mathrm{nm}$

Figure 5. Numbers of terms (p and p') and the runtime speed (using Eq. (33)) vs. the error requirement (ϵ).

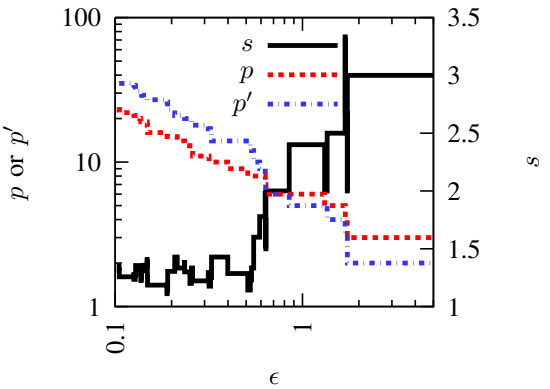


Figure 6. Improvement for x-coma with $c = 0.01$.

APPENDIX

In this section, we will prove that the eigenfunctions a real Hermitian operator under certain condition can be made either symmetric or antisymmetric. This result will be applied to our lithography image simulation problem at the end of this appendix.

Define the operator A based on a real function $A(\boldsymbol{k}, \boldsymbol{k}')$ as

$$A\phi(\boldsymbol{k}) = \int A(\boldsymbol{k}, \boldsymbol{k}')\phi(\boldsymbol{k}')\mathrm{d}\boldsymbol{k}',$$

where $A(\boldsymbol{k}, \boldsymbol{k}') = A(\boldsymbol{k}', \boldsymbol{k})$. Define the parity operator P as

$$P\phi(\boldsymbol{k}) = \phi(-\boldsymbol{k}). \tag{45}$$

Theorem 2. *The parity operator P has only eigenvalues 1 and -1. If ψ is the eigenfunction associated with the eigenvalue 1, then $\psi(\boldsymbol{k}) = \psi(-\boldsymbol{k})$. If ψ is the eigenfunction associated with the eigenvalue -1, then $\psi(\boldsymbol{k}) = -\psi(-\boldsymbol{k})$.*

Proof: Assume ψ is an eigenfunction of P such that $P\psi = \lambda\psi$.

We put

$$\psi_1(\boldsymbol{k}) = \frac{\psi(\boldsymbol{k}) + \psi(-\boldsymbol{k})}{2},$$
$$\psi_2(\boldsymbol{k}) = \frac{\psi(\boldsymbol{k}) - \psi(-\boldsymbol{k})}{2}.$$

Then $P\psi = P\psi_1 + P\psi_2 = \psi_1 - \psi_2$. Since $P\psi = \lambda\psi$, we have $\psi_1 - \psi_2 = \lambda\psi_1 + \lambda\psi_2$. Then we have $(\lambda - 1)\psi_1 = (\lambda + 1)\psi_2$.

Since ψ_1 is an even function and ψ_2 is a odd function, we have $\lambda = 1$ or $\lambda = -1$. So we can say that the parity operator P has only eigenvalues 1 and -1.

Obviously, if $\lambda = 1$, then $\psi_2 = 0$. So we have $\psi(\boldsymbol{k}) = \psi(-\boldsymbol{k})$. If $\lambda = -1$, then $\psi_1 = 0$. So we have $\psi(\boldsymbol{k}) = -\psi(-\boldsymbol{k})$. ■

Theorem 3. *If $A(\boldsymbol{k}, \boldsymbol{k}')$ is real and $A(\boldsymbol{k}, \boldsymbol{k}') = A(-\boldsymbol{k}', -\boldsymbol{k})$, $A(\boldsymbol{k}, \boldsymbol{k}')$ can be expanded in terms of orthonormal real functions $\psi_i(\boldsymbol{k})$ as*

$$A(\boldsymbol{k}, \boldsymbol{k}') = \sum_{i=1}^{\infty} \sigma_i \psi_i(\boldsymbol{k})\psi_i(\boldsymbol{k}'),$$

where $\psi_i(\boldsymbol{k})$ satisfies

$$\psi_i(\boldsymbol{k}) = \psi_i(-\boldsymbol{k}) \text{ or } \psi_i(\boldsymbol{k}) = -\psi_i(-\boldsymbol{k}).$$

Proof: Assume $\{\sigma_i\}$ are eigenvalues and $\{\phi_i\}$ are normalized eigenfucntions.

Since A is real and symmetric, σ_i and ϕ_i are real with $\int \phi_i(\boldsymbol{k})\phi_j(\boldsymbol{k})\mathrm{d}\boldsymbol{k} = \delta_{ij}$ for any $i, j \in \mathbb{N}$.

It is easy to see that

$$\int A(\boldsymbol{k}, \boldsymbol{k}')\phi_i(-\boldsymbol{k}')\mathrm{d}\boldsymbol{k}' = \sigma_i\phi_i(-\boldsymbol{k}).$$

This implies that

$$\sigma_i(P\phi_i) = A(P\phi_i).$$

for any $i \in \mathbb{N}$.

So $P\phi_i$ is still an eigenfunction of A associated with engenvalue σ_i.

Since A is a compact operator from L_2 to L_2, then $\forall\lambda \in \mathbb{R}$, if $\lambda \neq 0$, there are at most finitely many $i \in \mathbb{N}$ such that $\sigma_i = \lambda$.

Without losing generality, we assume $\sigma_1 = \cdots = \sigma_n \neq \sigma_{n+1}$.

Put $V = \mathrm{span}\{\phi_1, \cdots, \phi_n\}$, then $P(V) \subset V$. So there is an orthonormal basis $\{\psi_1, \cdots, \psi_n\}$ of V such that $P\psi_i = \psi_i$ or $P\psi_i = -\psi_i$ for $1 \leq i \leq n$.

It is easy to see that

$$\begin{aligned}&\phi_1(\boldsymbol{k})\phi_1(\boldsymbol{k}') + \cdots + \phi_n(\boldsymbol{k})\phi_n(\boldsymbol{k}')\\ =&\psi_1(\boldsymbol{k})\psi_1(\boldsymbol{k}') + \cdots + \psi_n(\boldsymbol{k})\psi_n(\boldsymbol{k}').\end{aligned}$$

Then we can conclude that

$$A(\boldsymbol{k}, \boldsymbol{k}') = \sum_{i=1}^{\infty} \sigma_i \psi_i(\boldsymbol{k})\psi_i(\boldsymbol{k}')$$

with $\psi_i(\boldsymbol{k}) = \psi_i(-\boldsymbol{k})$ or $\psi_i(\boldsymbol{k}) = -\psi_i(-\boldsymbol{k})$ for $i \in \mathbb{N}$ and $\psi_i(\boldsymbol{k})$ is real. ■

Since $\mathcal{T}_{\text{real}}$ satisfies the conditions of Theorem 3, therefore we have

$$\mathcal{T}_{\text{real}}(\boldsymbol{k}', \boldsymbol{k}'') = \sum_n \sigma_n \mathcal{Q}_n(\boldsymbol{k}')\mathcal{Q}_n(\boldsymbol{k}''), \tag{46}$$

where each σ_n is a real number, $\mathcal{Q}_n(\boldsymbol{k})$ is real, and $\mathcal{Q}_n(\boldsymbol{k})$ satisfies

$$\mathcal{Q}_n(\boldsymbol{k}) = \mathcal{Q}_n(-\boldsymbol{k}) \text{ or } \mathcal{Q}_n(\boldsymbol{k}) = -\mathcal{Q}_n(-\boldsymbol{k}) \tag{47}$$

Therefore, Eq. (24) can be derived [5], where Q_n is the inverse Fourier transform of $\mathcal{Q}_n$.

Using Eq. (47), it is easy to see the inverse Fourier transform Q_n of $\mathcal{Q}_n$ is either real or pure imaginary. So the magnitude operator $|\cdot|$ in Eq. (26) is not necessary, because even it is pure imaginary, we can make it real by multiplying the imaginary unit i, $iQ_n \rightarrow Q_n$. Therefore, taking only p' terms as an approximation, we can prove Eq. (30).

REFERENCES

[1] N. B. Cobb, "Fast Optical and Process Proximity Correction Algorithms for Integrated Circuit Manufacturing," Ph.D. dissertation, University of California at Berkeley, 1998.

[2] N. Cobb and Y. Granik, "New concepts in OPC," in *Proc. SPIE 5377*, 2004, pp. 680–690.

[3] J. Ye, Y.-W. Lu, Y. Cao, L. Chen, and X. Chen, "System and method for lithography simulation," Patent US 7,117,478 B2, Jan. 18, 2005.

[4] G. A. Gomba, "Collaborative Innovation: IBM's Immersion Lithography Strategy for 65 nm and 45 nm Half-pitch Nodes & Beyond," in *Proc. SPIE 6521*, 2007.

[5] Y. C. Pati and T. Kailath, "Phase-shifting masks for microlithography: automated design and mask requirements," *Journal of the Optical Society of America A*, vol. 11, pp. 2438–2452, Sep. 1994.

[6] Y. Pati, A. Ghazanfarian, and R. Pease, "Exploiting structure in fast aerial image computation for integrated circuit patterns," *IEEE Trans. on Semiconductor Manufacturing*, vol. 10, no. 1, pp. 62–74, Feb. 1997.

[7] M. Born and E. Wolf, *Principles of Optics : Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7th ed.

[8] A. K.-K. Wong, *Optical Imaging in Projection Microlithography*. SPIE Publications, Mar. 2005.

[9] K. Adam, Y. Granik, A. Torres, and N. B. Cobb, "Improved modeling performance with an adapted vectorial formulation of the Hopkins imaging equation," in *Proc. of SPIE 5040*, Jun. 2003, pp. 78–91.

[10] K. Adam and W. Maurer, "Polarization effects in immersion lithography," *Journal of Micro/Nanolithography, MEMS and MOEMS*, vol. 4, no. 3, p. 031106, 2005.

[11] P. Yu and D. Z. Pan, "ELIAS: An Extensible Lithography Aerial Image Simulator with Improved Numerical Algorithms," 2008, in preparation.