Fast-Converging Iterative Gradient Decent Methods for High Pattern Fidelity Inverse Mask Design

Jue-Chin Yu and Peichen Yu
Department of Photonics and Institute of Electro-Optical Engineering,
National Chiao Tung University, 1001 Da-hsueh Rd., Hsinchu 30050, Taiwan
E-mail address: yup@faculty.nctu.edu.tw

Abstract

Convergence speed and local minimum issue have been the major issues for inverse lithography. In this paper, we propose an inverse algorithm that employs an iterative gradient-descent method to improve convergence and reduce the Edge Placement Error (EPE). The algorithm employs a constrained gradient-based optimization to attain the fast converging speed, while a cross-weighting technique is introduced to overcome the local minimum trapping.

Keywords: optical proximity correction, inverse lithography, image gradient

1. Introduction

Semiconductor fabrication is the cornerstone of the current IC (Integrated Circuit) industry. With recent advances in microlithography now pushing towards nano-scale features, the problem of printing circuit layouts on wafers has become more intricate and convoluted. Optical Proximity Correction (OPC) is a resolution enhancement technique that modifies mask layout designs in order to minimize their distortion when transferred to silicon. A good OPC implementation may prove sufficient for a given process technology, precluding the need for a more expensive alternative like Double Patterning, Alternating Phase-Shift Masks (AltPSM), Immersion Lithography and so on. Clearly, OPC has obvious advantages in terms of efficiency and manufacturing cost.

Segment-based OPC has been the general industry approach and has proven successful through many CMOS generations. Because it only modifies existing edges in the layout, segment-based OPC has the advantage of being easy to implement, particularly in iterative algorithms. However, as the Critical Dimension (CD) becomes ever smaller, this type of edge-only compensation is not flexible enough to exploit the full range of possible mask corrections. Therefore, inverse mask design, also called Inverse Lithography Technology (ILT), has been suggested as an alternative due to its more relaxed constraints and full-mask approach. However, inverse calculation is faced with several problems, including bad convergence and the existence of local minima. To overcome these issues, many approaches have been proposed, such as pixel-flipping, gradient strategies and so on. Still, these approaches need to be further developed and refined to become the next-generation OPC.

In this paper, we propose an inverse algorithm that employs an iterative gradient-based method to improve convergence and reduce the Edge Placement Error (EPE). The algorithm achieves fast convergence by defining a digitized gradient vector and a cross-weighting matrix to determine the corresponding weighting factors. The digitized gradient vector of the cost function depicts an optimized step direction for the iteration, while the cross-weighting matrix is a tensor expression that takes the correlations of EPEs of different edges into account for the weighting factors.

2. Methodology

The Köhler's illumination model [1, 2, 3] is widely used in optical lithography. Figure 1 shows the configuration of an exposure system. The condenser lens $\mathbf{L}_c$ collimate the radiative light from the illumination source which is a quasi-monochromatic light source with a central wavelength $\overline{\lambda}$ and imaged on the pupil plane by a lens $\mathbf{L}_1$. Moreover, 193 nm ArF excimer laser is current exposure illumination source. With the nature of the quasi-monochromatism, the partially coherent image formation is applied to evaluate the image intensity on the wafer. The aerial image intensity of position (x, y) on image plane can be expressed as [1, 2]

$$I(x,y) = \int\int_{-\infty}^{\infty}\int\int J(x_o^{'} - x_o^{''}, y_o^{'} - y_o^{''}) m(x_o^{'}, y_o^{'}) m^{*}(x_o^{''}, y_o^{''}) H(x - x_o^{'}, y - y_o^{'}) H^{*}(x - x_o^{''}, y - y_o^{''}) dx_o^{'} dy_o^{'} dx_o^{''} dy_o^{''} \quad (1)$$

Optical Microlithography XXIII, edited by Mircea V. Dusa, Will Conley, Proc. of SPIE Vol. 7640,
76402L · © 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.846568

where $J(x_o^{'}-x_o^{''},y_o^{'}-y_o^{''})$ is the mutual intensity which indicates the interference by two object points, $(x_o^{'},y_o^{'})$ and $(x_o^{''},y_o^{''})$, $m(x_o,y_o)$ is the mask function, $H(x_o,y_o)$ is the impulse response of the optical image system. The asterisk * denotes the complex conjugate.

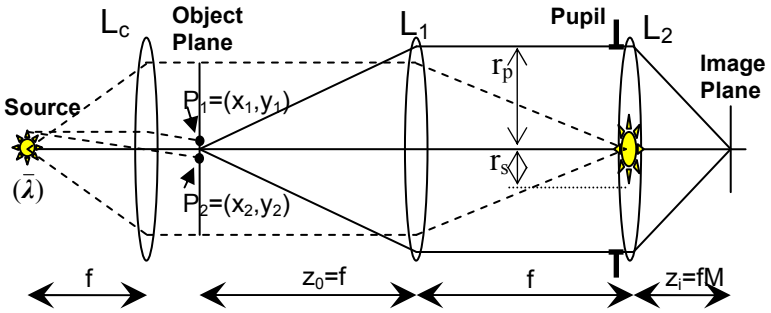


Figure 1. Configuration sketch of an optical lithography imaging system.

However, such four-fold integration like eq. (1) is a time-consuming calculation. To enhance the calculation efficiency, a numerical method named singular value decomposition (SVD) [4] are early proposed to decompose the eq.(1) to the summation of eigenvalues and eigenvector multiplication [5, 6, 7] that is

$$I(x,y)=\sum_{q=1}^{Q}\lambda_q\left|h_q(x,y)\otimes m(x,y)\right|^2, \tag{2}$$

where λ_q is the eigenvalue of the eigenvector h_q. Only Q dominant eigenvalues and their eigenvectors are used to synthesize the aerial image. The Eq.(2) shows the aerial image formation as a sum of coherent system [6]. Furthermore, such light intensity expression in Eq.(2) can be represented by the field form where

$$\begin{cases} E_q(x,y)=h_q(x,y)\otimes m(x,y), & (3)\\ I(x,y)=\sum_{q=1}^{Q}\lambda_q\left|E(x,y)\right|^2 & (4)\end{cases}$$

that is consistent with the time-average electromagnetic wave intensity. Finally a cost function is defined as following to evaluate the image performance in our optimization that is

$$F(m(x,y))=\left\|I_t(x,y)-I(x,y)\right\|^2 \tag{5}$$

where I_t is the target image which is configured by the drawn patterns. $\|\cdot\|$ denotes the *Euclidean* length. Moreover, the mask is constrained in the range of $0\leqq m(x,y)\leqq 1$.

Our approach is first based on the Frank and Wolfe method [8, 9] which transforms a nonlinear problem into an approximate linear optimization. In our inverse problem, we would like to find the minimum of a cost function F,

$$F\left(m(x,y)\right)\rightarrow \min., \tag{6}$$

where $m(x,y)$ is the mask function that spans in the x and y directions. Our approach can be expressed as an iterative calculation,

$$\nabla F^k\left(m^k(x,y)\right)\cdot m'(x,y)\rightarrow \min., \tag{7}$$

where k is the iteration number and

$$\left|m'(x,y)\right|\leq 1. \tag{8}$$

To calculate the iteration step, we establish the solution of $m'(x,y)$ as

$$m'(x,y) = \hat{m}^k(x,y). \tag{9}$$

Moreover, to satisfy Eq.(2) and Eq.(3) $\hat{m}^k(x,y)$ is given the following expression,

$$\hat{m}^k(x,y) = sign\left(\nabla F^{k-1}\left(m^{k-1}(x,y)\right)\right), \tag{10}$$

where

$$sign(x) = \begin{cases} 1, & when\ x > 0 \\ 0, & when\ x = 0 \\ -1, & when\ x < 0 \end{cases}, \tag{11}$$

and the step direction is given by

$$\Delta m^k(x,y) = \hat{m}^k(x,y) - m^{k-1}(x,y). \tag{12}$$

To properly scale the step length through the iterations, we define a parameter $\kappa \in [0,1]$ that specifies the ratio between the configurations $m^{k-1}(x,y)$ and $m^k(x,y)$. This parameter can be used as

$$m^k(x,y) = m^{k-1}(x,y) + \kappa \Delta m^k(x,y). \tag{13}$$

This shows a linear exploration with origin in $m^{k-1}(x,y)$ that moves in the direction of $\Delta m^k(x,y)$ by the amount of $\kappa \left|\Delta m^k(x,y)\right|$. Therefore, the minimization problem becomes

$$F^k\left(m^{k-1}(x,y) + \kappa \Delta m^k(x,y)\right) \to \min, \tag{14}$$

therefore a line search approach is employed to find the optimal κ every iteration.

Moreover, the optimization is accelerated by appropriate weighting factors for the gradient cost function, which are calculated using a cross-weighting matrix that takes into account the correlations among different edges, such matrix can be expressed as follows

$$\begin{bmatrix} w_{Inner} \\ w_{Outer} \\ w_{Side} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} EPE_{Inner} \\ EPE_{Outer} \\ EPE_{Side} \end{bmatrix}, \tag{15}$$

where w_{Inner}, w_{Outer} and w_{Side} are the weighting factors for the gradient vectors of inner, outer, and side edges, respectively, and C_{ij} are coefficients that take into account the effects of EPEs of different edges on the weighting factors. Figure 2 shows the exemplary definitions of the edges for two closely-placed vias and one via near a vertical bar. Such approach classifies the edges that have their own detecting points into three kinds of groups. Therefore the correction of the eight edge segments in figure 2 (a) originally described by an 8×8 matrix can be reduced to a 3×3 matrix. As the same reason a 14×14 matrix in figure 2 (b) can also be reduced to a 3×3 matrix. So the computing time is reduced especially in large size template. Furthermore other demands can be added by extending the dimension of the cross-weighting matrix.

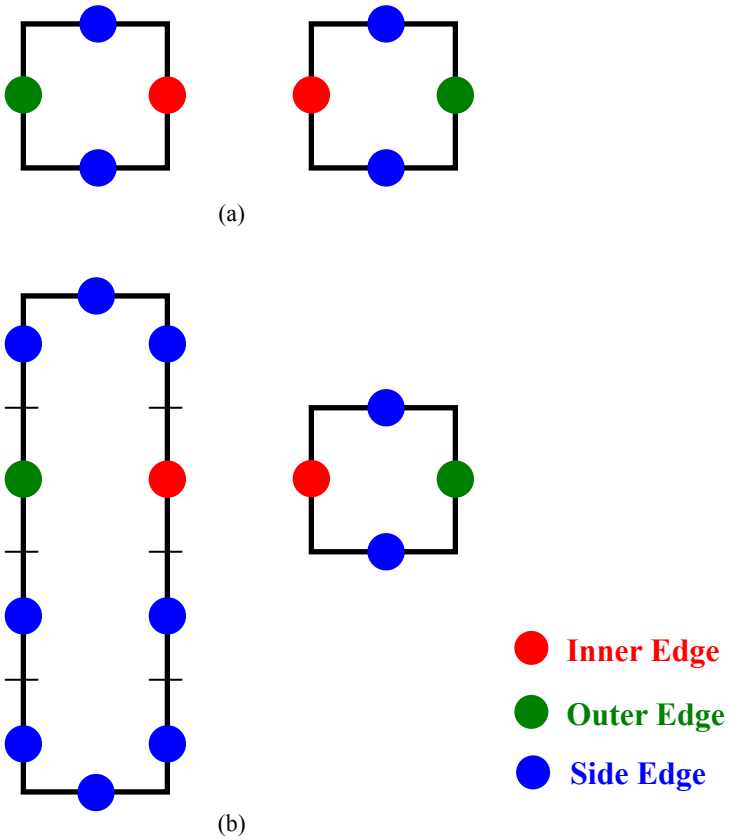


Figure 2. Two exemplary definitions of the edges for (a) two closely-placed vias and (b) one via near a vertical bar.

4. Results and Discussion

In this section we demonstrate our algorithm under λ=193nm, NA = 0.7 and partial coherent illumination modeled with eight kernels. For comparison, the drawn mask of two 90 nm×90 nm vias with 100 nm separation placed on a 1.28 μm×1.28 μm template composed of 10 nm×10 nm pixels is first calculated under different weighting strategies. Three weighting approaches are first applied. One is non weighting optimization where the eq. (15) becomes

$$\begin{bmatrix} w_{Inner}^{k} \\ w_{Outer}^{k} \\ w_{Side}^{k} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \overline{EPE}_{Inner}^{k} \\ \overline{EPE}_{Outer}^{k} \\ \overline{EPE}_{Side}^{k} \end{bmatrix}, \tag{16}$$

another is constant weighting where the eq. (15) can be expressed as

$$\begin{bmatrix} w_{Inner}^{k} \\ w_{Outer}^{k} \\ w_{Side}^{k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 25 \\ 25 \end{bmatrix}, \tag{17}$$

and the other is dynamic allocating the weighting value by the EPE of every iteration where the eq. (15) can be adapted

to

$$\begin{bmatrix} w_{Inner}^{k} \\ w_{Outer}^{k} \\ w_{Side}^{k} \end{bmatrix} = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 20 \end{bmatrix} \begin{bmatrix} \overline{EPE}_{Inner}^{k} \\ \overline{EPE}_{Outer}^{k} \\ \overline{EPE}_{Side}^{k} \end{bmatrix}, \tag{18}$$

where k is the iteration number and $\overline{(\cdots)}$ means the average operation in eq. (16), (17) and (18). Finally because the correlations between different kinds of the edges are included in our cross-weighting approach, the non-zero elements do not only locate along the diagonal entries of the matrix. So in our simulation the eq. (15) has the formation as

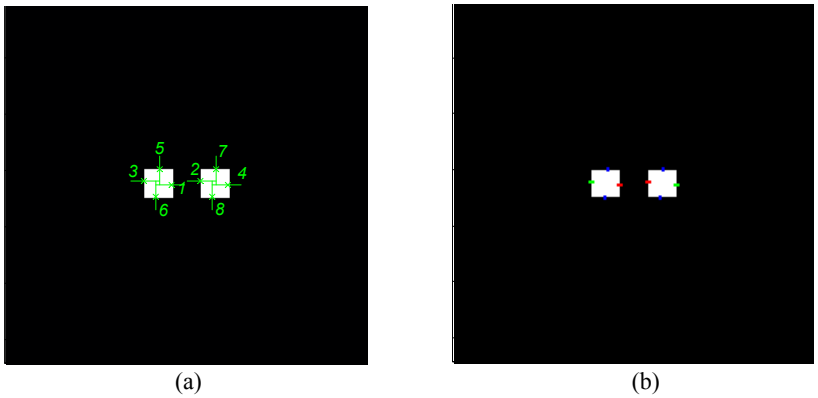
$$\begin{bmatrix} w_{Inner}^{k} \\ w_{Outer}^{k} \\ w_{Side}^{k} \end{bmatrix} = \begin{bmatrix} 20 & -3 & -2 \\ -3 & 20 & -2 \\ -1.5 & -1.5 & 23 \end{bmatrix} \begin{bmatrix} \overline{EPE}_{Inner}^{k} \\ \overline{EPE}_{Outer}^{k} \\ \overline{EPE}_{Side}^{k} \end{bmatrix} \tag{19}$$

where k is the iteration number. $\overline{(\cdots)}$ means the average operation.

Continuously before running the simulation, the detecting points for evaluating the EPE of every edge should be first decided. Subsequently the edges' types which are belonging to inner, outer or side are also defined. Figure 3 (a) shows the settings of the detecting points and (b) displays the classification of inner, outer and side edges that are labeled by red, green and blue spots respectively.

In figure 4 (a) and (b) respectively show the optimized gray level mask and its aerial image under non-weighting IL correction. Obviously, there is no threshold contours. (c) and (d) show the constant weighting results. (e) and (f) show the results under dynamic weightings allocated by every iteration's EPEs. (g) and (h) show our proposed cross-weighting results where the weightings are governed by eq. (19). Moreover, the contours at a threshold value of 0.5 in figure 4 (b), (d), (f) and (h) are displayed with a deep green line.

Figure 5 shows the EPEs as recorded in every iteration. The results under constant weighting are shown in figure 5 (a). And figure 5 (b) shows dynamic weightings which are allocated by the EPE when every iteration. Then the results calculated with our proposed cross-weighting approaches are shown in figure 5 (c). On all graphs, the fluctuations in the beginning are caused by searching the direction of the local minimum. Starting around iteration ten and for about twenty iterations, the EPEs converge stably and keep their trend toward zero. Then, just after the thirtieth iteration there is a clear change in the weight settings that further pushes the EPEs closer to zero. The final EPEs are approximately 8.5 nm in figure (a), 8.2 nm in figure (b) and 1.5 nm in figure (c).



(a) (b)

Figure 3. Two closely placed vias which are 90 nm×90 nm in area are aligned in a 1.28 μm×1.28 μm template, where (a) shows the detecting point setting of the two vias. (b) displays the classification of the edges where the red spots denote the inner edges, the greens denote the outer edges and the blues denote the side edges.

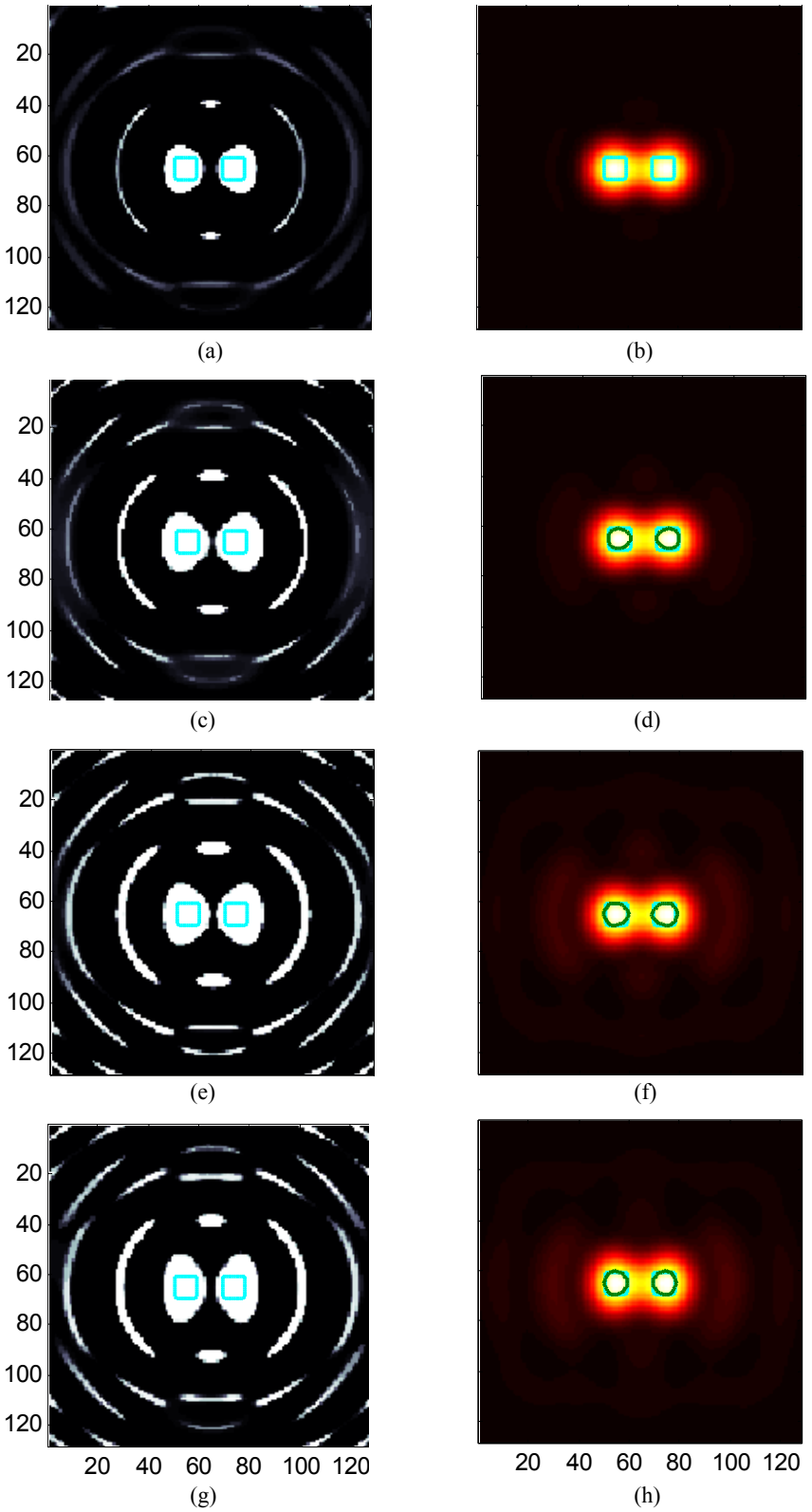


Figure 4. The corrected gray-level mask patterns of the two closely-placed vias under different weighting approaches. Part (a), (c), (e) and (f) show the optimized gray level mask corresponding to the varying transmittance from 0 to 1. Part (b), (d), (f) and (h) show the aerial image with contours at a threshold equal to 0.5 in deep green. The cyan lines in both plots show the drawn patterns.

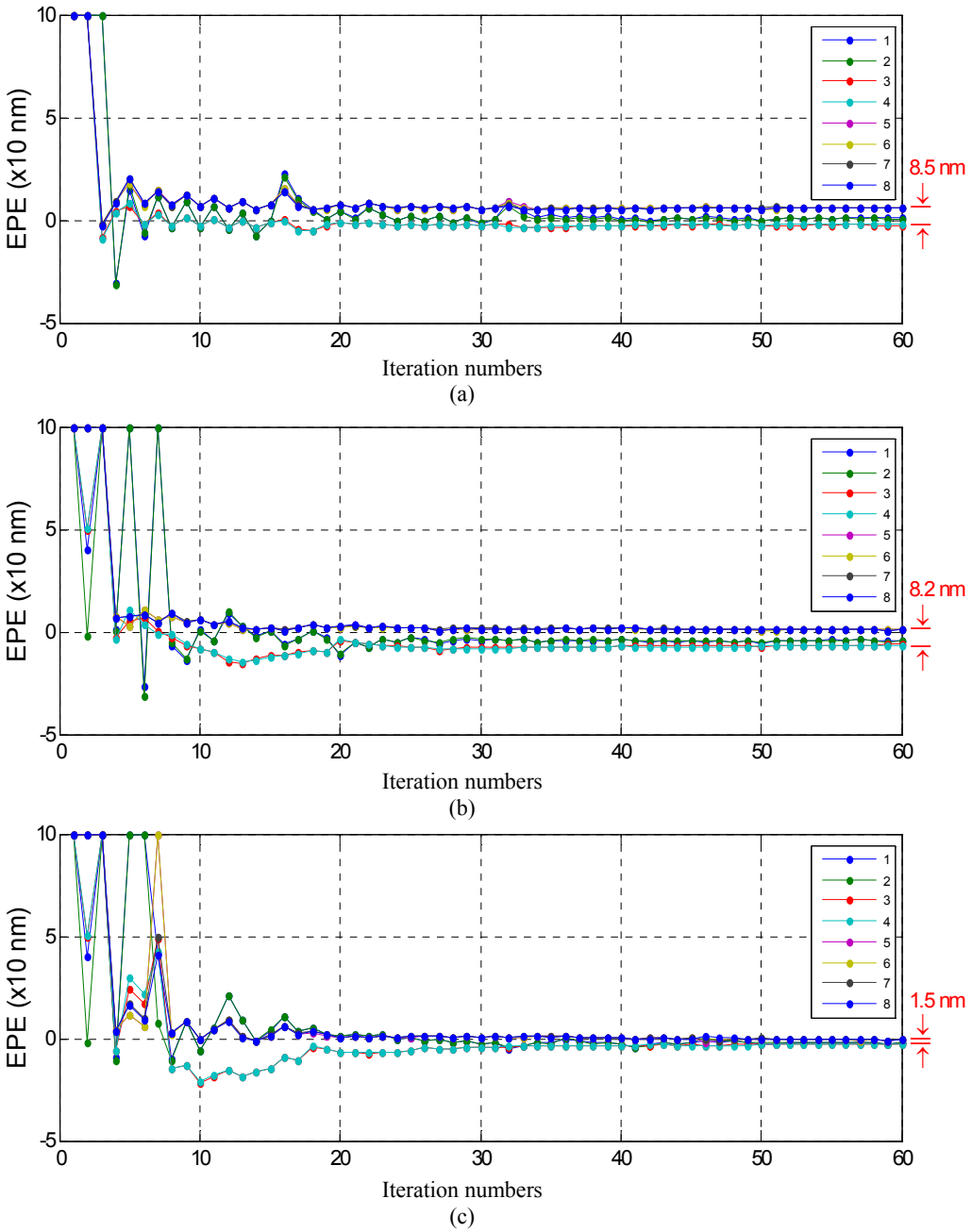


Figure 5. Edge Placement Error (EPE) vs. Iteration Number. (a) Constant weighting. (b) The weighting is re-assigned in every iteration according to the measured EPE of individual anchor points. (c) Cross-weighting technique. There are eight anchor points on the drawn edges. The horizontal axis shows the iteration numbers and the vertical axis is the EPE in number of pixels with a unit of 10 nm.

In summary because the feature size and the configuration are both far beyond the diffraction limit which is governed by *Rayleigh* criterion where R = 0.61λ/NA ~ 170 nm, the IL correction has no threshold contour without weighting treatment. Moreover duo to the sub-wavelength feature size and configurations, the sever diffraction effects cause complex correlations between different location. The weighting methods only incorporating with single edge are not sufficient to obtain the promising EPEs. Therefore, the optimization incorporating the interference between different locations should be considered. Comparing to figure 5 (a) and (b), our proposed cross-weighting technique that had taken such phenomena into account present the better performance as shown in figure 5 (c).

Similarly, figure 6 (a) and (b) show the optimized gray level mask and aerial image of an arbitrary SRAM contact array. The two middle vias array are composed of 190 nm×190 nm vias and the two side vias array are composed of 120 nm×120 nm vias. The template composed of 10 nm×10 nm pixels has size of 2.56 μm×2.56 μm. Moreover the final EPEs are still aggressive as shown in figure 7 where the EPEs at the every iteration are plotted and finally converged to 1.6 nm by our IL calculation incorporating with cross-weighting technique.

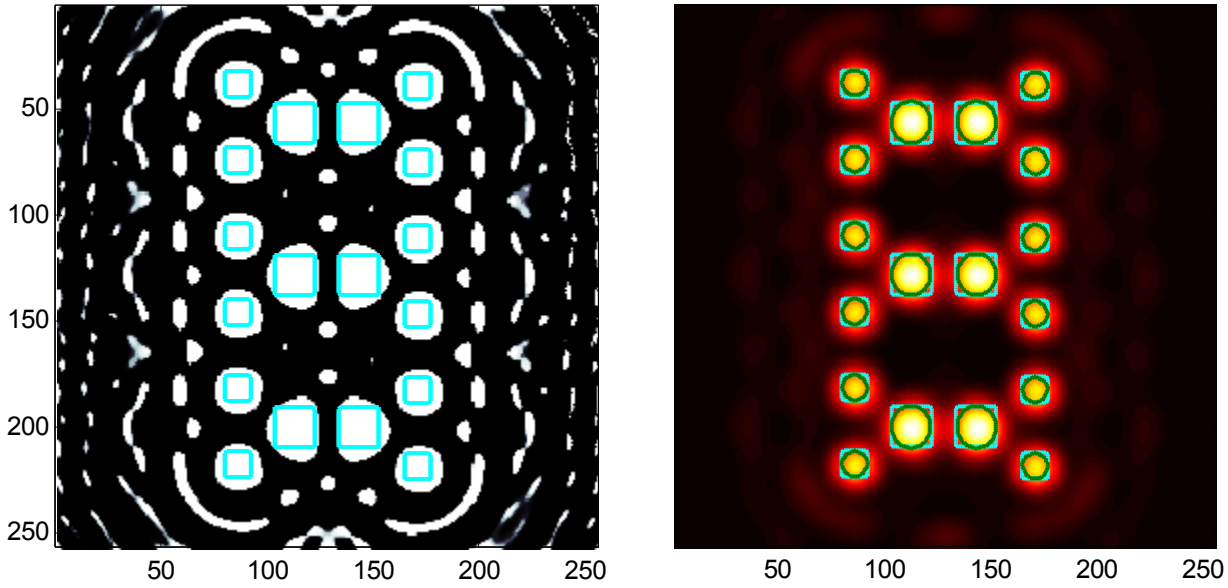


Figure 6. The corrected gray-level mask patterns of the two closely-placed vias under different weighting approaches.

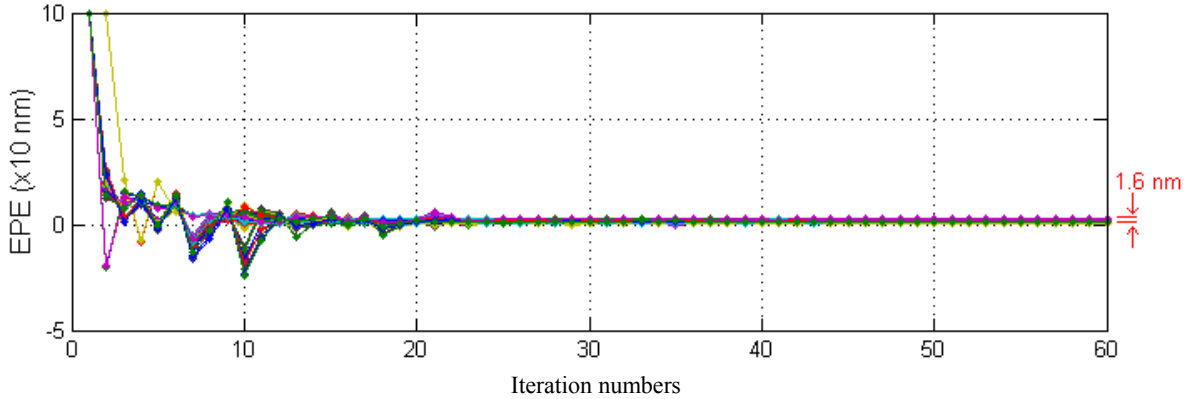


Figure 7. Edge Placement Error (EPE) vs. Iteration Number.

5. Conclusion

We successfully demonstrate the inverse lithography approach by using the constrained gradient approach iteratively. By incorporating with the proposed cross-weighting technique, the optimization calculations hardly suffer from local minimum trapping. The drawn patterns with sub-wavelength feature size and configuration receive the sizing corrections that are similar with segment-based OPC. Moreover, the surrounding assist features are simultaneously generated by inverse lithography calculation and with irregular geometrics which is not achievable in conventional OPC. Furthermore, the fast convergence results are obtained where there are less than thirty iterations for converging in all above simulation.

Reference

[1] M. Born and E. Wolf, *Principles of Optics*, Pergamon, 7th ed., Cambridge University, (1999).

[2] J. W. Goodman, *Statistical Optics*, John Wiley and Sons, (1985).

[3] Alfred Kwok-kit Wong, *Optical Imaging in Projection Microlithography*, SPIE, Washington, (2005).

[4] Steven J. Leon, *Linear Algebra with applications*, 6th ed., Prentice-Hall, (2002).

[5] B. E. A. Saleh and M. Rabbani, "Simulation of partially coherent imagery in the space and frequency domains and by modal expansion," Applied Optics **21**, 15066-15079 (1982).

[6] N. B. Cobb, *Fast optical and process proximity correction algorithms for integrated circuit manufacturing*, University of California at Berkeley, Berkely, California, (1998).

[7] Edmund Y. Lama and Alfred K. K. Wong, "Computation lithography: virtual reality and virtual virtuality," Optics Express **17**, 12259-12268 (2009).

[8] Yuri Granik, "Fast pixel-based mask optimization for inverse lithography," J. Microlith., Microfab., Microsyst. **5**, 043002 (2006).

[9] M. Minoux, *Mathematical programming in Theory and Algorithms*, Wiley, New York, (1986).

Appendix

To calculate the gradient of the cost function F, we first apply the operator ∇, which is defined as

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial m_{(x_1,y_1)}} & \cdots & \frac{\partial}{\partial m_{(x_1,y_n)}} & \cdots & \frac{\partial}{\partial m_{(x_1,y_N)}} \\ \vdots & \ddots & & \ddots & \vdots \\ \frac{\partial}{\partial m_{(x_m,y_1)}} & \cdots & \frac{\partial}{\partial m_{(x_m,y_n)}} & \cdots & \frac{\partial}{\partial m_{(x_m,y_N)}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial m_{(x_M,y_1)}} & \cdots & \frac{\partial}{\partial m_{(x_M,y_n)}} & \cdots & \frac{\partial}{\partial m_{(x_M,y_N)}} \end{bmatrix}, \tag{A1}$$

where $m_{(x_m,y_n)}$ is the (m,n) pixel value and (x_m, y_n) the coordinate of this pixel. $m_{(x_m,y_n)}$ covers the region $\left[x_m - \frac{\delta x}{2}, x_m + \frac{\delta x}{2}\right] \cap \left[y_n - \frac{\delta y}{2}, y_n + \frac{\delta y}{2}\right]$ where the δx and δy are the differentials of the pixel in the x and y directions respectively. We may assume the value of $m_{(x_m,y_n)}$ is independent to the other pixels. This means

$$\frac{\partial m(x,y)}{\partial m_{(x_m,y_n)}} = \delta_d(x - x_m, y - y_n) = \begin{cases} \frac{1}{\delta x \delta y}, & x = x_m \ and \ y = y_n \\ 0, & x \neq x_m \ or \ y \neq y_n \end{cases}, \tag{A2}$$

where $\delta_d(x - x_{m,} y - y_n)$ is a discrete delta function. So, if $\delta x \to 0$ and $\delta y \to 0$, $m(x,y)$ is a continuous function and $\nabla m(x,y)$ becomes

$$\begin{bmatrix} \delta(x - \frac{M}{2}\delta x, y - \frac{N}{2}\delta y) & \cdots & \delta(x - \frac{M}{2}\delta x, y) & \cdots & \delta(x - \frac{M}{2}\delta x, y + \frac{N}{2}\delta y) \\ \vdots & \ddots & & \ddots & \vdots \\ \delta(x, y - \frac{N}{2}\delta y) & \cdots & \delta(x,y) & \cdots & \delta(x, y - \frac{N}{2}\delta y) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \delta(x + \frac{M}{2}\delta x, y - \frac{N}{2}\delta y) & \cdots & \delta(x - \frac{M}{2}\delta x, y) & \cdots & \delta(x + \frac{M}{2}\delta x, y + \frac{N}{2}\delta y) \end{bmatrix}, \tag{A3}$$

where M, N are the total number of pixels in the x and y directions respectively and are both infinite. The delta function $\delta(x,y)$ has the property

$$\delta(x,y)=\begin{cases}\infty, & x=y=0\\ 0, & x\neq 0 \ \ or \ \ y\neq 0\end{cases}, \tag{A4}$$

$$\int_{0^-}^{0^+}\int \delta(x,y)dxdy \ = 1. \tag{A5}$$

According to the above description, if $m(x,y)$ is a mask with infinitesimal pixels, we may express $\nabla m(x',y')$ as

$$\nabla m(x',y')=\delta(x-x',y-y')\Big|_{x'\in\left[-\frac{M}{2},\frac{M}{2}\right],\ y'\in\left[-\frac{N}{2},\frac{N}{2}\right];\ M\to\infty,N\to\infty}. \tag{A6}$$

The cost function F may be expressed by the Manhattan norm of the difference between the real image $I(m,x,y)$ and ideal image $I_0(x,y)$ as

$$F(m)=\left\|I(m,x,y)-I_0(x,y)\right\|^2=\iint\left(I(m,x,y)-I_0(x,y)\right)^2dxdy, \tag{A7}$$

then we derive F by using the ∇ operator, where

$$\begin{aligned}\nabla F(m)&=\nabla\left\|I(m,x,y)-I_0(x,y)\right\|^2\\ &=\nabla\iint\left(I(m,x,y)-I_0(x,y)\right)^2dxdy\\ &=\iint\nabla\left(I(m,x,y)-I_0(x,y)\right)^2dxdy\\ &=\iint 2\left(I(m,x,y)-I_0(x,y)\right)\nabla I(m,x,y)dxdy.\end{aligned} \tag{A8}$$

We may substitute $I(m,x,y)-I_0(x,y)$ by ΔI and define that the phasor formation of the electric field $E(x,y)$ is equal to $h(x,y)\otimes m(x,y)$ where $h(x,y)$ is the optical system response function and $\otimes$ denotes the convolution operation. Therefore the intensity $I(m,x,y)$ can be expressed as $I(m,x,y)=E(x,y)E(x,y)^*=\left(h(x,y)\otimes m(x,y)\right)\cdot\left(h(x,y)\otimes m(x,y)\right)^*$. Then, the equation can be adapted to

$$\nabla F(m)=\iint 2\Delta I(x,y)\nabla\left(\sum_{k=1}^{K}\lambda_k\left(h(x,y)\otimes m(x,y)\right)\left(h(x,y)\otimes m(x,y)\right)^*\right)dxdy$$

$$=\sum_{k=1}^{K}\lambda_k\iint 2\Delta I(x,y)\left(E(x,y)\left(h(x,y)\otimes\nabla m(x,y)\right)^*+\left(h(x,y)\otimes\nabla m(x,y)\right)E^*(x,y)\right)dxdy,$$

$$= 4\sum_{k=1}^{K} \lambda_k \iint \Delta I(x,y)\,\mathrm{Re}\left[E^*(x,y)\left(h(x,y)\otimes \nabla m(x,y)\right)\right]\,dxdy$$

$$= 4\sum_{k=1}^{K} \lambda_k \iint \mathrm{Re}\left[\Delta I(x,y)E^*(x,y)\left(\iint h(x-x',y-y')\nabla m(x',y')dx'dy'\right)\right]\,dxdy$$

$$= 4\sum_{k=1}^{K} \lambda_k \iint\iint \mathrm{Re}\left[\Delta I(x,y)E^*(x,y)h(x-x',y-y')\nabla m(x',y')\right]\,dxdydx'dy'$$

$$= 4\sum_{k=1}^{K} \lambda_k \iint \mathrm{Re}\left[\iint \left(\Delta I(x,y)E^*(x,y)\right)h(x-x',y-y')dxdy\right]\nabla m(x',y')\;\;dx'dy'$$

$$= 4\sum_{k=1}^{K} \lambda_k \iint \mathrm{Re}\left[\left(\Delta I(x',y')\cdot E^*(x',y')\right)\otimes h(-x',-y')\right]\nabla m(x',y')\;\;dx'dy'$$

$$= 4\sum_{k=1}^{K} \lambda_k \iint \mathrm{Re}\left[\left(\Delta I\cdot E^*\right)\otimes h(-x',-y')\right]\delta(x''-x',y''-y')\;\;dx'dy'$$

$$= 4\sum_{k=1}^{K} \lambda_k\;\mathrm{Re}\left[\left(\Delta I\cdot E^*\right)\otimes h(-x'',-y'')\right] \qquad \text{(A9)}$$