Shot overlap model-based fracturing for edge-based OPC layouts

Shangliang Jiang and Avideh Zakhor
Signetron Inc. and University of California at Berkeley, CA, 94720, USA
Email: {sjiang,avz}@eecs.berkeley.edu

ABSTRACT
In this paper, we develop a novel fracturing algorithm with shot overlap that is tailored towards rectilinear masks, such as those generated via edge based OPC software. Our proposed fracturing algorithm generates both the location and dosage of shots given the mask layout and mask making parameters. In the first step we heuristically cover the mask polygon with overlapping shots. Next, we incorporate the forward scattering and resist model in a least squares problem to compute the best dosage for all shots. Finally, we update the locations of the shot edges by computing the edge placement error between our simulated contour and the desired contour. One unique feature of our algorithm is that it can readily trade off between edge placement error and shot count by adjusting two input parameters. Compared to a commercially available non-overlapping shot software package, for a 400 µm × 400 µm micron SRAM unit with about 1 million polygons, our algorithm results in a 23% reduction in shot count, while increasing the weighted average EPE from 0.7 to 1 nanometers.

Keywords: Model-based fracturing, mask data preparation, Variable Shaped Beam mask writing

1. INTRODUCTION
Resolution enhancement techniques such as Optical Proximity Correction (OPC) have enabled the semiconductor manufacturing industry to continuously shrink the critical dimension (CD) of integrated circuits. Current photomasks contain increasingly sophisticated OPC patterns resulting from complex model-based OPC. While this trend results in higher wafer fidelity, the masks are more challenging to write as they are more complex. Specifically, masks are first decomposed by a process called fracturing that transforms mask polygons into smaller basic shapes, rectangles or trapezoids. Currently, geometric rules are used to decompose the masks resulting in a fast fracture time but also a prohibitively large number of shots resulting in long write times. This increased write time is a major concern for the semiconductor industry.

In this work, we propose an alternative fracturing approach to minimize mask write time, whereby the shot location, size, and dosage are determined via a model-based approach. We model the resist with a fixed nonlinear threshold function and the electron beam proximity effect by convolution with a two dimensional (2-D) Gaussian scattering filter. By incorporating the mask model it is possible to overlap shots thereby achieving a lower shot count as compared with commercially available non-overlap software.

In this paper, we develop a novel fracturing algorithm with shot overlap that is tailored towards rectilinear masks, such as those generated via edge based OPC software. Our proposed fracturing algorithm generates both the location and dosage of shots given the mask layout and mask making parameters. In the first step we heuristically cover the mask polygon with overlapping shots. Next, we incorporate the forward scattering and resist model in a least squares problem to compute the best dosage for all shots. Finally, we update the locations of the shot edges by computing the edge placement error between our simulated contour and the desired contour. One unique feature of our algorithm is that it can readily trade off between edge placement error and shot count by adjusting two input parameters. Compared to a commercially available non-overlapping shot software package, for a 400 µm × 400 µm micron SRAM unit with about 1 million polygons, our algorithm results in a 23% reduction in shot count, while increasing the weighted average EPE from 0.7 to 1 nanometers.

Keywords: Model-based fracturing, mask data preparation, Variable Shaped Beam mask writing

1. INTRODUCTION
Resolution enhancement techniques such as Optical Proximity Correction (OPC) have enabled the semiconductor manufacturing industry to continuously shrink the critical dimension (CD) of integrated circuits. Current photomasks contain increasingly sophisticated OPC patterns resulting from complex model-based OPC. While this trend results in higher wafer fidelity, the masks are more challenging to write as they are more complex. Specifically, masks are first decomposed by a process called fracturing that transforms mask polygons into smaller basic shapes, rectangles or trapezoids. Currently, geometric rules are used to decompose the masks resulting in a fast fracture time but also a prohibitively large number of shots resulting in long write times. This increased write time is a major concern for the semiconductor industry.

In this work, we propose an alternative fracturing approach to minimize mask write time, whereby the shot location, size, and dosage are determined via a model-based approach. We model the resist with a fixed nonlinear threshold function and the electron beam proximity effect by convolution with a two dimensional (2-D) Gaussian scattering filter. By incorporating the mask model it is possible to overlap shots thereby achieving a lower shot count as compared with commercially available non-overlap software.

In this paper, we develop a novel fracturing algorithm with shot overlap that is tailored towards rectilinear masks, such as those generated via edge based OPC software. Our proposed fracturing algorithm generates both the location and dosage of shots given the mask layout and mask making parameters. In the first step we heuristically cover the mask polygon with overlapping shots. Next, we incorporate the forward scattering and resist model in a least squares problem to compute the best dosage for all shots. Finally, we update the locations of the shot edges by computing the edge placement error between our simulated contour and the desired contour. One unique feature of our algorithm is that it can readily trade off between edge placement error and shot count by adjusting two input parameters. Compared to a commercially available non-overlapping shot software package, for a 400 µm × 400 µm micron SRAM unit with about 1 million polygons, our algorithm results in a 23% reduction in shot count, while increasing the weighted average EPE from 0.7 to 1 nanometers.

Keywords: Model-based fracturing, mask data preparation, Variable Shaped Beam mask writing

1. INTRODUCTION
Resolution enhancement techniques such as Optical Proximity Correction (OPC) have enabled the semiconductor manufacturing industry to continuously shrink the critical dimension (CD) of integrated circuits. Current photomasks contain increasingly sophisticated OPC patterns resulting from complex model-based OPC. While this trend results in higher wafer fidelity, the masks are more challenging to write as they are more complex. Specifically, masks are first decomposed by a process called fracturing that transforms mask polygons into smaller basic shapes, rectangles or trapezoids. Currently, geometric rules are used to decompose the masks resulting in a fast fracture time but also a prohibitively large number of shots resulting in long write times. This increased write time is a major concern for the semiconductor industry.

In this work, we propose an alternative fracturing approach to minimize mask write time, whereby the shot location, size, and dosage are determined via a model-based approach. We model the resist with a fixed nonlinear threshold function and the electron beam proximity effect by convolution with a two dimensional (2-D) Gaussian scattering filter. By incorporating the mask model it is possible to overlap shots thereby achieving a lower shot count as compared with commercially available non-overlap software.

In Section 2, we review past results on fracturing. In Section 3 we provide a brief overview of our algorithm. Section 4 contains experimental results for our proposed algorithm. We conclude with possible directions for future work in Section 5.

2. BASICS OF FRACTURING
In this section we review prior results on fracturing. Section 2.1 reviews past results on mask manufacturability and Section 2.2 reviews recent results on model-based fracturing, in which the mask fabrication process is modeled.
2.1 Mask manufacturing quality

Bloiecker et al. have evaluated many metrics to quantify fracturing quality; they proposed and demonstrated that shoreline or external sliver length is a suitable metric for evaluating fracture quality as it closely correlates with manufacturability while being fast to evaluate. Features with width below a threshold $\delta$, as determined by the Variable Shape Beam (VSB) mask-writing tool, are called slivers. Slivers whose length is along the boundary of the layout polygon are called external slivers. As the shot size becomes smaller, the electron current density becomes steeper, adversely affecting the shot placement. Thus slivers result in large size variability, negatively affecting CD control. Many existing fracturing approaches focus on reducing sliver length. In addition, Spence et al. have demonstrated that the number of shots is directly correlated with the write-time.

While there are algorithms which simultaneously minimize sliver number and length along with number of shots, they ignore the underlying physics behind the mask writing process; rather, they solve a simpler geometric partitioning problem by introducing rules to capture the manufacturing parameters. A parallel can be drawn to early work in OPC that was built upon a series of rules for corrections rather than incorporating exact models. Our goal in this paper is to demonstrate a fracturing algorithm that uses models to create masks with lower shot count without degrading the mask and wafer image quality.

2.2 Model based fracturing

Model-based fracturing incorporates the electron beam model into the fracturing process. In the absence of models, fracturing is limited to non-overlapping shots. However, even a simple model allows one to consider overlapping shots which may lead to a reduction in shot count while improving mask and wafer quality. The lower shot count is an important goal in limiting mask cost, write-time, and improving fidelity.

Recently, D2S has introduced the concept of model-based mask data preparation by simulating the electron beam (e-beam) mask writing process. Their approach operates on the mask that has been optimized by pixel-based OPC. They place overlapping shots in such a way that the simulated mask image approximates the desired target mask. In contrast to conventional rule-based fracturing, model-based fracturing places shots based upon the models for both the mask writer and the photoresist. We have also recently developed an alternative model-based fracturing technique for pixel-based OPC masks achieving a substantial improvement in shot count, upwards of 50%, while maintaining mask and wafer fidelity.

A number of challenges need to be addressed in designing model-based fracturing algorithms. First, the electron transfer is not exact and is instead modeled by a low-pass filter, typically Gaussian or sum of Gaussians. This may imply a deconvolution step which is an inherently ill-conditioned problem. Second, the energy of the electrons is transmitted and absorbed by a chemical resist which acts as a thresholding operator in terms of what appears on the resulting mask. Thresholding is a nonlinear operator and makes the problem formulation more complex as many possible energy profiles may result in the same mask. Finally, minimizing shot count is similar to minimizing the number of overlapping rectangles needed to cover a rectilinear polygon. However, this has been shown to be an NP-hard problem. In the next section, we describe the way our proposed algorithm addresses these issues.

3. PROPOSED ALGORITHM

We quantify the shot placement with two error metrics. The first is the number of shots, which relates to minimizing mask fabrication time, as stated in Section 2.1. The second metric is the edge placement error (EPE) which is defined as the difference in length between the Simulated Resist threshold Energy (SRE) and the target mask at edges. Figs. 1a and 1b show an example of a target mask and its SRE respectively, and Fig. 1c shows the resulting difference between the two. To compute the EPE, each edge is isolated. In addition, as shown in Figs. 1d through 1g, the corners are ignored during the error computation, to account for the low-pass effect of the mask writing process. Thus, the difference is ignored within the black boxes and the remaining error is summed.

There are 3 major steps to our algorithm. First, we cover an input polygon with as few rectangles as possible. During this step we also enforce maximum rectangle size. This is similar to covering a rectilinear polygon with overlapping rectangles which is known to be NP-complete, in general. As such, we resort to a greedy algorithm
Figure 1: EPE computation: (a) target; (b) SRE; (c) difference; (d-g) EPE computation: error is only computed in the green region and not in the black boxes.

that grows rectangles. Specifically, we first select a convex corner of the input polygon, shown in Fig. 2a. From there, the rectangle is expanded until it reaches the boundary of the polygons, as shown in Figs. 2b and 2c. An additional feature of our algorithm is the ability to create approximate covers. The algorithm starts the same by selecting a convex corner, as shown in Fig. 3a. However, rather than expanding to only within the input
polygon, the shot is allowed to exceed the polygon, as shown in Fig. 3b. This feature, an approximate covering of the input polygon, allows our algorithm to tradeoff shot count with mask fidelity. Another way to make this tradeoff is to change the minimum shot size allowed. Raising the minimum shot size results in fewer shots which again decreases shot count at the cost of mask fidelity.

![Figure 2: The rectangle covering: (a) initial selection; (b) middle selection; (c) final selection.](image)

![Figure 3: The approximate rectangle covering: (a) initial selection; (b) final selection.](image)

The second step is to compute the shot dosage. In particular, we solve a convex optimization problem which forces the simulated resist energy to be close to the resist threshold along the boundary of the polygon and to be greater than the resist threshold within the interior of the polygon. These goals reflect the thresholding property of the resist. The SRE is computed by first convolving the dosage of each shot with the forward scattering filter, namely, a 2-D Gaussian, with $\sigma_f$ as the forward scattering parameter:

$$g_f(x) = \frac{1}{\pi \sigma_f} \exp\left(\frac{x^2}{\sigma_f^2}\right)$$  \hspace{1cm} (1)

An example of the forward scattering is shown in Fig. 4. Fig. 4a shows the original shot and Fig. 4b shows the resist energy after convolution with the forward scattering model of Equation 1. The convolved energy of all shots are then summed. To set up the optimization problem we sample multiple points on the boundary and interior of the polygon to form a optimization problem as shown in Fig. 5. The variables are the shots while the objective function and constraints are derived from the sampled points. The optimization problem can be written as:
\[ \begin{align*} 
\text{min :} & \quad \| A\vec{x} - \vec{a} \|_2 \\
\text{s.t. :} & \quad B\vec{x} \geq \vec{b} 
\end{align*} \] (2)

where \( \vec{x} \) corresponds to the final shot dosages and \( \vec{a} \) corresponds to the desired resist energy for samples on the edges of the polygon as shown in Fig. 5b. \( A \) models the forward scattering of each shot \( \vec{x} \) onto the various \( \vec{a} \) edge locations. Fig. 5c shows the samples \( \vec{b} \) of the interior of the input polygon, and the matrix \( B \) models the forward scattering of the shots \( \vec{x} \) on each interior point. Both matrices \( A, B \) are generated by computing the effect of forward scattering for each shot at the corresponding sampled point. This optimization step generates continuous dosage values even though current e-beam mask writers have a finite number of dosage values. As such, we quantize the continuous valued solutions to be one of the valid dosage levels.

The final step is to refine the shot edge placement. We use the shot placement, size, and quantized dosage from previous steps to generate the SRE, which is the wafer energy after the electron beam shots have passed through the resist. This simulation includes both the low-pass filtering due to the scattering and the thresholding of the resist. We compute the difference between SRE and target mask to obtain the EPE for each edge in the polygon and then we compute the contribution of each shot to the EPE of the edge. We use this to derive the amount of edge perturbation needed for each shot edge. This last step of edge refinement may be repeated in order to further improve upon EPE.

Our algorithm can tradeoff shot count with mask fidelity by changing the minimum shot size and by approximating the greedy polygon cover in the first step. These features may lead to fewer rectangles which corresponds to fewer shots in the final fracturing but may also increase the EPE.

4. EXPERIMENTAL RESULTS

We have tested out our proposed algorithm on five separate mask layouts which are portions of an SRAM chip. The characteristics including size in microns and number of polygons of the layout clips are shown in Table 1. The size of the polygon is given at the mask scale and we assume a magnification of 4x.

For the fracturing we assumed a max shot size of 350nm, \( \sigma_f = 30\text{nm} \), and a constant threshold resist model, in which the resist is an ideal threshold function. We use the same process model to evaluate the shot fracturing resulting from both the commercial software and our proposed algorithm to create the simulated masks. We compare both EPE and shot count between the two systems as shown in Fig. 6.

After generating the EPE for each edge, we compute the the weighted average absolute EPE using Equation 3 as follows:
Figure 5: Elements in dosage computation: (a) shots; (b) edges and their samples; (c) interior points.

<table>
<thead>
<tr>
<th>Layout Names</th>
<th>Mask Size in microns</th>
<th>Number of polygons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout 1</td>
<td>$400\mu m \times 400\mu m$</td>
<td>914k</td>
</tr>
<tr>
<td>Layout 2</td>
<td>$400\mu m \times 400\mu m$</td>
<td>850k</td>
</tr>
<tr>
<td>Layout 3</td>
<td>$20\mu m \times 16\mu m$</td>
<td>401</td>
</tr>
<tr>
<td>Layout 4</td>
<td>$21\mu m \times 29\mu m$</td>
<td>763</td>
</tr>
<tr>
<td>Layout 5</td>
<td>$44\mu m \times 43\mu m$</td>
<td>2.4k</td>
</tr>
</tbody>
</table>

Table 1: Layout characteristics.

Weighted Average EPE = \[
\frac{\sum_{edges} |EPE(edge)| \ast \text{length}(edge)}{\sum_{edges} \text{length}(edge)}
\]  

(3)

Where $EPE(edge)$ is defined in Figs. 1d-1g and excludes the corners. For each layout clip we first run our algorithm across multiple input parameters, allowing us to create a rate distortion curve trading off shot count and weighted average EPE. These plots are shown in Fig. 7 for each layout in Table 1. For each plot, the point enclosed in a circle (triangle) corresponds to the highest (lowest) fidelity. As seen, our algorithm is capable of achieving a large range in shot count by adjusting input parameters. The ‘*’ point in Fig. 7 shows the characteristic of the solution obtained via a non-overlapping commercial package. This point results in the lowest error but highest shot count.
Fig. 8a shows an example of a target mask while Figs. 8b, 8c, and 8d show the non-overlapping commercial, highest fidelity, and lowest fidelity fracturing containing 6, 4, and 2 shots respectively. These correspond to the asterisk, circle, and triangle points in Fig. 7 respectively. Furthermore, Figs. 8e, 8f, and 8g show the commercial, highest fidelity, and lowest fidelity SRE and Figs. 8h, 8i, and 8j show the commercial, highest, and lowest fidelity fracturing error, respectively. Table 2 compares the shot count and the weighted average EPE for the commercial system, highest, and lowest fidelity. The commercial system has the most shots, i.e. 6, but the lowest weighted average EPE, i.e. 0.30nm. The highest fidelity shot overlap solution has only 4 shots with a minor increase in weighted average EPE, to 0.39nm. The lowest fidelity shot overlap solution only has 2 shots but 2.22nm weighted average EPE. The non-overlap commercial system has difficulty with the minor jogs in the polygon and introduces a sliver in the fracturing. Our highest fidelity overlap solution uses two overlapping shots to cover the two jogs. Since the commercial and the highest fidelity cover the target polygon perfectly, the EPE is low, as can be seen from the figures and table. Our lowest fidelity overlap solution instead ignores the jogs in forming the cover which decreases the shot count dramatically but greatly increases the EPE due to the jogs.

<table>
<thead>
<tr>
<th>Fracturing Method</th>
<th>Shot Count</th>
<th>Weighted Average EPE (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial</td>
<td>6</td>
<td>0.30</td>
</tr>
<tr>
<td>Highest Fidelity</td>
<td>4</td>
<td>0.39</td>
</tr>
<tr>
<td>Lowest Fidelity</td>
<td>2</td>
<td>2.22</td>
</tr>
</tbody>
</table>

Table 2: Highest vs lowest fidelity.

Table 3 compiles the shot count and weighted average EPE for all layouts using the non-overlapping commercial system and our proposed solution for our proposed highest and lowest fidelity shot overlap solutions. As seen, our proposed algorithm results in a significant improvement in the shot count as compared to the commercial system. For the two largest layouts, Layout 1 and 2, there is a 20-23% reduction in shot count with 30-40% increase in weighted average EPE for the highest fidelity. For the 3 smaller layouts, we also achieve similar performance at highest fidelity, namely, 14-25% shot count reduction and 30-45% increase in weighted average EPE at the highest fidelity. However, at highest fidelity, the weighted average EPE value is still around 1nm for all layouts for our algorithm which is quite acceptable in practice. For all 5 layouts, the lowest fidelity fracturing shows a much greater improvement in shot count of 43-53% but with an increase in weighted average EPE.

Fig. 9 compares the resulting fracturings for a polygon, shown in Fig. 9a, from Layout 4 that has the highest error. The fracturings for the non-overlap commercial system, highest, and lowest fidelity are shown in Figs. 9b, 9c, and 9d respectively. From the error profile shown in Figs. 9e, 9f, and 9g, we see that the commercial non-overlap approach has the lowest EPE. At highest fidelity, our proposed solution is rather comparable as there is only some degradation at a sharp jog. For the lowest fidelity fracturing, there is further degradation at
Figure 7: Rate distortion curve of shots and weighted average EPE for: (a) Layout 1; (b) Layout 2; (c) Layout 3; (d) Layout 4; (e) Layout 5.
all jogs. The shot count and weighted average EPE are compared quantitatively for all three modes in Table 4. The commercial system has the greatest shot count but lowest error while the highest fidelity solution has 12.5% improvement in shot count with only 0.13nm increase in error. This shot reduction and percentage error change is consistent with that seen for the complete Layout 4.

<table>
<thead>
<tr>
<th>Layout</th>
<th>Commercial</th>
<th>Proposed</th>
<th>Percentage Reduction</th>
<th>Weighted Average EPE (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shot Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Commercial</td>
<td>Proposed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layout 1</td>
<td>3.78M</td>
<td>2.9M</td>
<td>23.3%</td>
<td>0.7</td>
</tr>
<tr>
<td>Layout 2</td>
<td>4.7M</td>
<td>3.76M</td>
<td>20%</td>
<td>0.78</td>
</tr>
<tr>
<td>Layout 3</td>
<td>4.0k</td>
<td>3k</td>
<td>25%</td>
<td>0.19</td>
</tr>
<tr>
<td>Layout 4</td>
<td>4.76k</td>
<td>4.1k</td>
<td>13.87%</td>
<td>0.14</td>
</tr>
<tr>
<td>Layout 5</td>
<td>26.5k</td>
<td>22.7k</td>
<td>14.34%</td>
<td>0.22</td>
</tr>
</tbody>
</table>

Table 3: Comparison of shots and weighted average EPE for all layouts.

Fracturing Method

<table>
<thead>
<tr>
<th>Fracturing Method</th>
<th>Shot Count</th>
<th>Weighted Average EPE (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial</td>
<td>32</td>
<td>0.34</td>
</tr>
<tr>
<td>Highest Fidelity</td>
<td>28</td>
<td>0.47</td>
</tr>
<tr>
<td>Lowest Fidelity</td>
<td>23</td>
<td>1.18</td>
</tr>
</tbody>
</table>

Table 4: Comparing fracturings.

Fig. 10 compares the distribution of the EPE at highest fidelity and the fracture resulting from the commercial non-overlap system for the two large Layouts 1 and 2. From the distributions in Figs. 10a and 10c we see that the commercial system achieves an EPE=0nm more frequently than our proposed method. However, our cumulative distribution functions in Figs. 10b and 10d show that the high fidelity solution is comparable to the commercial solution for achieving an absolute EPE within 2nm.

Fig. 11 shows the distribution of the EPE at lowest fidelity and the fracture resulting from the commercial non-overlap system, for Layouts 1 and 2. From the distributions in Figs. 11a and 11c we see that the commercial system achieves an EPE=0nm significantly more frequently than our proposed method and has a tighter EPE range. However, our cumulative distribution functions in Figs. 11b and 11d show that lowest fidelity solution is comparable to the commercial non-overlap solution for achieving an absolute EPE within 5nm while resulting in more than 50% reduction in shot count.

We repeat these experiments for the smaller Layouts 3 through 5. Fig. 12 compares the distribution of the EPE for the smaller layouts at both highest and lowest fidelity. From the distributions at highest fidelity we see that our proposed algorithm is comparable in achieving EPE=0nm. However, at lowest fidelity our algorithm results in a lower percentage of sites with EPE=0nm. The cumulative distribution functions are shown in Fig. 13. For all three layouts we note that at highest fidelity our proposed algorithm results in better EPE performance for EPE=2nm. Again, at lowest fidelity for Layout 4 our proposed algorithm matches the commercial non-overlap approach at EPE=2nm while for Layouts 3 and 5 our proposed algorithm matches at EPE=5nm.
5. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed a novel model-based fracturing algorithm. By incorporating the mask model into fracturing we are able to demonstrate a 14-55% improvement in shot count. This is achieved with a combination of a greedy polygon cover algorithm to estimate initial shot placement and least squares optimization to generate shot dosages. Finally, we draw upon edge-based OPC\textsuperscript{19} to more accurately place shot edges. The resulting fracturings not only achieve a 55% improvement in shot count but also minimally impact mask fidelity. In future work we plan to test the limits of our algorithm on smaller technology nodes. We intend to explore the influence of mask writer parameters on our results and to further incorporate mask constraints into our algorithm. Finally, we plan to extend this work to non-rectangular shots such as trapezoids and triangles.

6. ACKNOWLEDGMENTS

We wish to express our gratitude to Vivek Singh of Intel for his part in initiating this line of research. We wish to acknowledge the contributions of the students, faculty, and sponsors of the Berkeley Wireless Research Center, in particular Brian Richards. Yi Zou and Gek Soon Chua from GlobalFoundries provided valuable input to help guide our development. This work could not have been done without the support from IMPACT. We wish to thank Yuri Granik and Emile Sahouria from Mentor Graphics for their critical help in working with the Calibre\textsuperscript{©} software package. Finally, we express our thanks to Richard Lozes, Peter Buck, Jim DeWitt, and Ines Stolberg for many useful technical inputs on mask writers.
REFERENCES


Figure 8: Example of fracturings: (a) target; (b) commercial non-overlap shots; (c) high fidelity shots; (d) low fidelity shots; (e) commercial SRE; (f) high fidelity SRE; (g) low fidelity SRE; (h) commercial error; (i) high fidelity error; (j) low fidelity error.
Figure 9: (a) Target polygon; shot placement for polygon with (b) commercial system, (c) proposed algorithm highest fidelity, (d) proposed algorithm lowest fidelity.
Figure 9: Edge placement error for (e) commercial system, (f) proposed algorithm highest fidelity, (g) proposed algorithm lowest fidelity.
Figure 10: (a) EPE distribution, (b) EPE cumulative distribution for Layout 1 at highest fidelity; (c) EPE distribution, (d) EPE cumulative distribution for Layout 2 at highest fidelity.
Figure 11: (a) EPE distribution, (b) EPE cumulative distribution for Layout 1 at lowest fidelity; (c) EPE distribution, (d) EPE cumulative distribution for Layout 2 at lowest fidelity.
Figure 12: Distribution of EPE for: (a) highest fidelity, (b) lowest fidelity for Layout 3; (c) highest fidelity, (d) lowest fidelity for Layout 4; (e) highest fidelity, (f) lowest fidelity for Layout 5.
Figure 13: Cumulative distribution of EPE for: (a) highest fidelity, (b) lowest fidelity for Layout 3; (c) highest fidelity, (d) lowest fidelity for Layout 4; (e) highest fidelity, (f) lowest fidelity for Layout 5.