Estimating cross-section semiconductor structure by comparing top-down SEM images*

Jeffery R. Price†, Philip R. Bingham, Kenneth W. Tobin, Jr., and Thomas P. Karnowski

Oak Ridge National Laboratory, Oak Ridge, TN, USA

ABSTRACT

Scanning electron microscope (SEM) images for semiconductor line-width measurements are generally acquired in a top-down configuration. As semiconductor dimensions continue to shrink, it has become increasingly important to characterize the cross-section, or sidewall, profiles. Cross-section imaging, however, requires the physical cleaving of the device, which is destructive and time-consuming. The goal of this work is to examine historical top-down and cross-section image pairs to determine if the cross-section profiles might be estimated by analyzing the corresponding top-down images. We present an empirical pattern recognition approach aimed at solving this problem. We compute feature vectors from sub-images of the top-down SEM images. Principal component analysis (PCA) and linear discriminant analysis (LDA) are used to reduce the dimensionality of the feature vectors, where class labels are assigned by clustering the cross-sections according to shape. Features are extracted from query top-downs and compared to the database. The estimated cross-section of the query is computed as a weighted combination of cross-sections corresponding to the nearest top-down neighbors. We report results obtained using 100nm, 180nm, and 250nm dense and isolated line data obtained by three different SEM tools.

Keywords: CD-SEM metrology, semiconductor inspection, lithography, image retrieval, linear discriminant analysis (LDA)

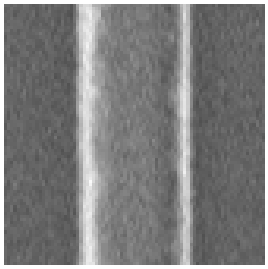
1. INTRODUCTION

Line-width measurements in the lithographic process are made almost exclusively from scanning electron microscope (SEM) images acquired by critical dimension SEM (CD-SEM) tools. These images are usually acquired in a top-down configuration, i.e., "looking down" onto the top of the semiconductor surface. As line-widths continue to shrink, however, it has become increasingly important to characterize the sidewall shape (e.g., the cross-section profile) of the line features rather than just their width. For example, two pairs of top-down and corresponding cross-section images are shown in Fig. 1. Although traditional CD-SEM tools may return the same line-width (i.e., critical dimension or CD) for the lines shown in (a) and (b), it is evident from their corresponding cross-sections in (c) and (d) that the sidewall structure of the two lines is quite different. In fact, the line feature of (b), whose cross-section is shown in (d), is wider at the top than elsewhere along its height; this characteristic, referred to as "undercut," decreases the stability of the structure and leads to higher defect rates. To acquire sidewall images like those of Fig. 1 (b) and (d), however, the semiconductor must be physically cleaved, a process which is destructive and time consuming and, therefore, hampers both throughput and sampling. Considering top-down images can be acquired much more efficiently – and with existing tools – we investigate in this paper sidewall shape estimation using only top-down images and a historical database of corresponding top-down and cross-section data.

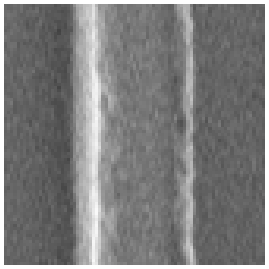
Towards this goal of cross-section estimation, we propose a system motivated by earlier image retrieval efforts for semiconductor defect yield analysis.[1] In the proposed system, a top-down image is submitted as a query. A set of features (i.e., a feature vector) is then extracted from this query image. Initially, a large number of features (1165) is extracted. The dimensionality of this feature vector is subsequently reduced to a more tractable level (20) through principal component analysis (PCA) and linear discriminant analysis (LDA), where the corresponding transformation matrices are computed using the historical data as the training set. The (reduced dimensionality) top-down feature vector of the query is then compared against those in the historical database. The query cross-section is estimated by computing a weighted average of the historical cross-sections that correspond to the nearest neighbor top-downs from the database. Details of each of

*Prepared by Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

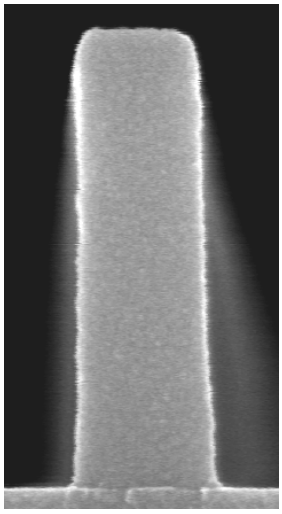
†Correspondence: J. Price, pricejr@ornl.gov, (865) 574-5743.



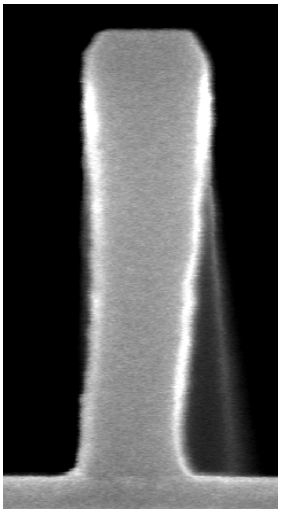
(a) Top-down image of a line feature.



(b) Top-down image of another line feature.



(c) Cross-section corresponding to (a).



(d) Cross-section corresponding to (b).

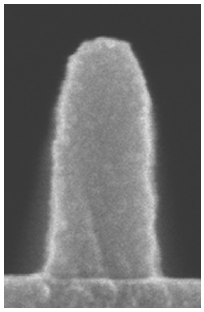
Figure 1. Two top-down and cross-section (sidewall) image pairs from 250nm dense (1:1) data. Traditional CD-SEM metrology may report the same single number (line-width) for both lines, even though it is clear from (c) and (d) that the cross-sections are quite different.

these steps are provided in the remainder of the paper. In Section 2, we describe how cross-section shapes are represented in the historical database. We then describe how features are extracted from the top-down images in Section 3. In Section 4, we present the techniques used for dimensionality reduction of the top-down feature vectors. Experimental results are presented in Section 5 and some closing comments are made in Section 6.

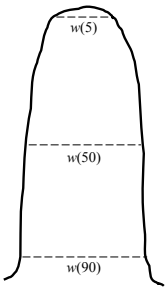
2. CROSS-SECTION REPRESENTATION

As a first step in the cross-section estimation problem, we must construct a quantifiable representation of cross-section shape. One goal for the finished system is the capability to estimate cross-sections across various design rules – i.e., different line-widths, pitches, and aspect ratios (ratio of line height to width). For this reason, we seek a cross-section representation that is invariant to these parameters (although future work may also include the estimation of aspect ratio). We accomplish this goal by defining cross-section shape by its normalized width at 100 equally spaced points along its height from top-to-bottom, where these widths (at a few locations) are illustrated by the horizontal dashed lines in Fig. 2(b) and (e), where the cross-section profiles (the outlines of the line structure) are extracted through a semi-automated graphical user interface.[2] Letting the widths (in nm) be represented by the 101-point vector $\mathbf{w}$, where $\mathbf{w}_n = w(n)$ for $n = 0, \ldots, 100$ ($n = 0$ is the top) and letting the design rule (i.e., target) line-width (in nm) be represented by L, the normalized cross-section representation (as a vector) is given by

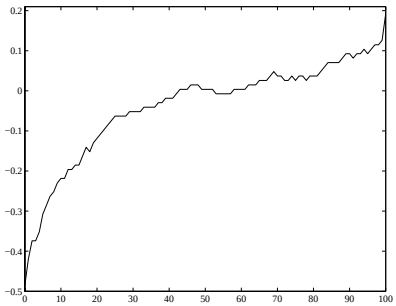
$$\mathbf{c} = \frac{1}{L}(\mathbf{w} - \bar{\mathbf{w}}) \tag{1}$$



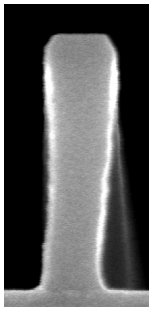
(a) Line cross-section (180nm, 1:1).



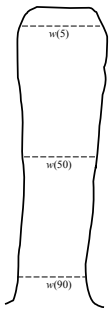
(b) Profile extracted from (a) with widths indicated.



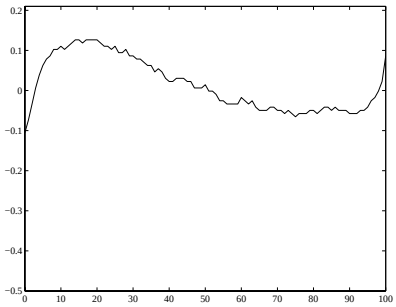
(c) Width curve corresponding to (b).



(d) Line cross-section (250nm, 1:1).



(e) Profile extracted from (d) with widths indicated.



(f) Width curve corresponding to (e).

Figure 2. Representation of the cross-section profiles. Figures (a) and (d) show cross-section images, (b) and (e) show extracted profiles, and (c) and (f) show the normalized width curves of those profiles sampled uniformly over 101 points from top to bottom.

where $\bar{\mathbf{w}}$ is the approximate width at the cross-section vertical midpoint as given by

$$\bar{\mathbf{w}} = \frac{1}{21} \sum_{n=40}^{60} w(n) \tag{2}$$

Plots of this cross-section representation (normalized width curve) are shown in Fig. 2 (c) and (f).

3. TOP-DOWN FEATURE EXTRACTION

Processing of the top-down images – for both historical storage (training set construction) and querying – involves two steps: (1) preprocessing and (2) feature extraction. Preprocessing, described in Section 3.1 below, comprises the steps of rotational correction, line localization, sub-image extraction, grayscale normalization, and masking. Feature extraction, described in Section 3.2, is the process of computing and storing potentially descriptive features from the extracted, normalized sub-images. We note that certain system parameters must be set by the user in an initialization or recipe generation phase. These parameters include the process design rule line-width (in nm), the image resolution (in nm/pixel), and the design rule pitch (i.e., the ratio of line spacing to line-width), and the line orientation (i.e., horizontal or vertical).

3.1. Preprocessing

Without loss of generality, we assume henceforth that all lines are vertically oriented. We begin with a top-down image such as that shown in Fig. 3. The first preprocessing step is to correct for any small angular deviation (within a few degrees) from vertical. Although there are many different ways to accomplish this, we currently employ the following

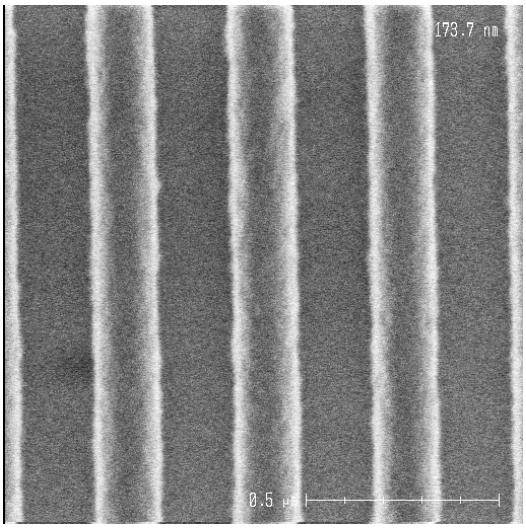


Figure 3. An example top-down image with three complete lines. This particular image is from a 180nm (1:1 pitch) design rule, is 512×512 pixels in size, and has a resolution of 2.635 nm/pixel.

simple approach. We first rotate the image over a small, fixed set of angles (seven angles spaced equally over -3° to $+3^\circ$ in the current implementation) and compute the average column variance at each angle. These variances are then fit to a quadratic and the minimizer of this quadratic is taken as the angle of rotation. This approach, although computationally inefficient, is very easily implemented and has proven robust in our experiments.

Once minor rotational variation has been corrected, we turn our attention next to automatically locating the lines within the image. A normalized profile of the image is calculated by summing along each column and scaling the resulting curve to lie between 0.0 and 1.0; an example is shown in Fig 4. From this profile, we first locate peaks by labeling all curve regions above 0.5 as peak regions, dilating those regions to fill in any holes, and then labelling as peaks the maximum values in these regions. Peaks near the image edges are discarded. In Fig. 4, the six peaks detected in this manner are indicated with circles. Given the peaks, we next determine which of the between-peak regions, or gaps, represent lines and which represent the space between two lines by applying heuristic rules based on the between-peak average value, the between-peak spacing, and the gradients to the left and right of each peak.

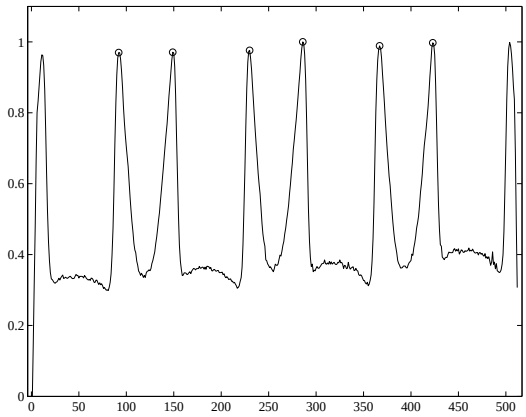
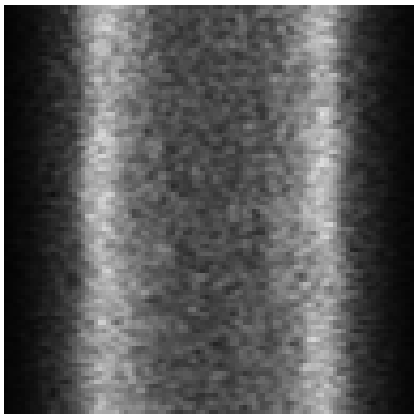


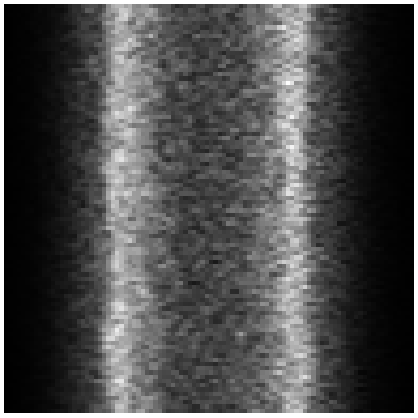
Figure 4. Normalized horizontal profile from the rotationally corrected version of the image in Fig. 3. Detected peaks are indicated with circles.

Once the lines have been located in this manner, we extract one or more sub-images along each line. The size of each of these sub-images is set to be three times the target line-width on a side. For example, the sub-image size for the top-down

image of Fig. 3 would be $(180\text{nm} \times 2.635^{-1}\ \text{pix/nm} \times 3) \approx 69$ pixels on a side. We note that no sub-images are extracted from any line whose width is greater than 2.5 times the target line-width. Such excessively wide (relative to the design rule) lines cannot be processed in our system; this, however, is not a limitation since such lines are far outside the normal operating range of the process (i.e., the width alone, regardless of cross-section shape, would indicate a process problem). Extracted sub-images are also permitted to overlap vertically by some arbitrary amount. In the current implementation, sub-images are allowed to overlap one another by 0.3 times the sub-image size. Each extracted sub-image is subsequently scaled to be zero-mean and unit-variance to account for gross grayscale (contrast) variations. Finally, a masking operation is applied to zero sub-image areas far from the line edges. Two examples of fully preprocessed sub-images (which have been rescaled for grayscale display) are shown in Fig. 5.



(a) Example sub-image.



(b) Another example sub-image.

Figure 5. Examples of preprocessed sub-images, both from a design rule of 100nm (5:1 pitch) with different lithographic focus and exposure settings. Each image is 300nm square.

3.2. Feature Extraction

A set of features (i.e., a feature vector) is computed for each sub-image obtained as described above. In order to compare structures of different physical dimensions, we must compute features that are invariant to the image scale. The first 13 features are taken from the well-known invariant image moments. The next 128 features are taken from a normalized (both in value and scale) profile curve. This curve is computed by summing along each column of the sub-image, scaling the result to lie between 0.0 and 1.0, and then interpolating to 128 points. These 128 points are appended to the feature vector. Finally, the sub-image is resized to dimensions 32×32 pixels; the resulting image is raster scanned and appended as the final 1024 points of the feature vector, yielding a 1165-dimensional feature vector.

4. DIMENSIONALITY REDUCTION

The aim of dimensionality reduction is to map feature vectors in a high-dimensional space to some lower-dimensional subspace, usually because of computational difficulties related to the large dimensionality of the original space. After the feature extraction process described above, our feature vectors have $p = 1165$ dimensions. Although it is possible to compare top-down feature vectors in this 1165-dimensional space, it is computationally demanding and unnecessary since there are redundant features as well as features that are not helpful with respect to the cross-section estimation problem. Motivated by techniques that have been successfully applied to template-based face recognition,[3, 4] we adopt a principal component analysis (PCA) plus linear discriminant analysis (LDA) approach for dimensionality reduction.

Principal component analysis[5] or PCA seeks the linear transformation that maximizes the scatter of all the sample vectors in the reduced dimensionality space. This transformation is given by the eigenvectors of the ensemble covariance matrix corresponding to the largest eigenvalues. The total scatter (i.e., covariance) matrix in the original space is given by

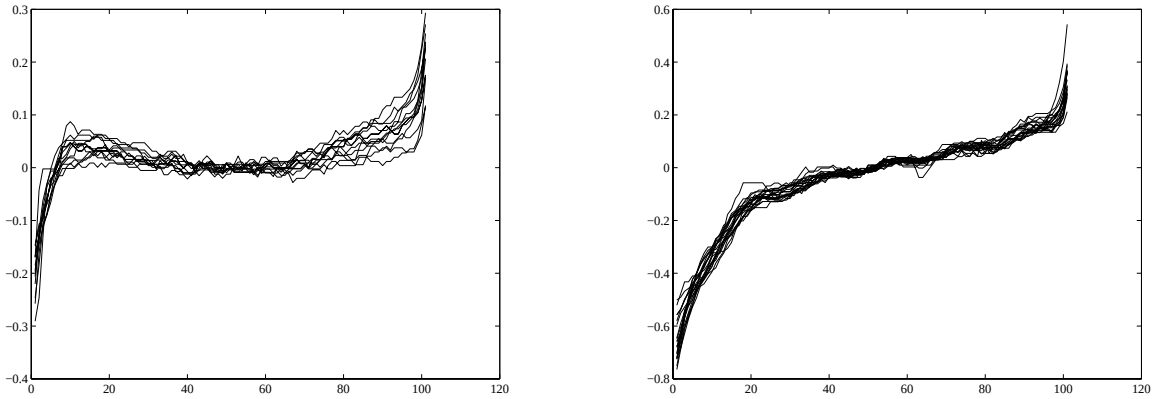
$$\mathrm{S}_t = \sum_{k=1}^{N} \big(\mathbf{x}_k - \boldsymbol{\mu}\big)\big(\mathbf{x}_k - \boldsymbol{\mu}\big)^T, \tag{3}$$

where $\mathbf{x}_k \in \mathbb{R}^p$ represents an original 1165-point feature vector, and $\boldsymbol{\mu}$ represents the mean feature vector. In the reduced space $\mathbb{R}^n$, the total scatter matrix is simply $\mathrm{P}^T\mathrm{S}_t\mathrm{P}$. PCA seeks the linear transformation P to maximize the determinant of the total scatter matrix in the reduced space:

$$\mathrm{P} = \arg\max_{\hat{\mathrm{P}}} |\hat{\mathrm{P}}^T \mathrm{S}_t \hat{\mathrm{P}}|. \tag{4}$$

The well-known solution to this problem is the matrix $\mathrm{P} \in \mathbb{R}^{p\times n}$ whose columns are the n eigenvectors of S_t with the n largest eigenvalues. In our current implementation, we use $n = 150$, hence the resulting transformation matrix P has dimensions 1165×150 and the vectors in the 150-dimensional space are given by $\mathbf{y} = \mathrm{P}^T\mathbf{x}$.

While the PCA projection is optimal in the sense of representation, and aids in removing redundant and/or noisy features, it does not take into account discriminative capability. Note that discrimination in our system refers to the ability to differentiate between top-downs associated with different cross-section shapes. To discriminate between different cross-section shapes, however, we must first define groupings of similar cross-sections. We accomplish this by applying the well-known k-means clustering algorithm[6] to the 101-point normalized cross-section representations defined by Eq. (1). In the current implementation, we employ $C = 20$ clusters. In Fig. 6 we plot the width curves for two example clusters. The cluster numbers (1-20) are then used as class labels for the top-down images.



(a) Cross-section width curves of an example cluster.

(b) Cross-section width curves of another example cluster.

Figure 6. Clustering of the normalized cross-section width curves.

With the above process in mind, we would like for the reduced-dimensionality feature vectors to preserve as much of the original separation between classes (defined by the cross-section clusters) as possible. Linear discriminant analysis

(LDA) seeks to accomplish this goal by seeking a projection matrix – now taking $\mathbb{R}^n$ to $\mathbb{R}^m$ – that minimizes the within-class scatter while simultaneously maximizing the between-class scatter. The underlying principle of LDA is to project high-dimensional feature vectors in $\mathbb{R}^n$ onto a lower-dimensional subspace $\mathbb{R}^m$, where $m < n$, while preserving as much discriminative information as possible. One formal expression for the corresponding optimization criterion (there are several equivalents[5]) is given by solving

$$\mathrm{A} = \arg\max_{\hat{\mathrm{A}}} \frac{\mathrm{tr}\big(\hat{\mathrm{A}}^T \mathrm{S}_b \hat{\mathrm{A}}\big)}{\mathrm{tr}\big(\hat{\mathrm{A}}^T \mathrm{S}_w \hat{\mathrm{A}}\big)} \tag{5}$$

where $\mathrm{A} \in \mathbb{R}^{n \times m}$, $\mathrm{tr}(\cdot)$ is the trace operator, $\mathrm{S}_w \in \mathbb{R}^{n \times n}$ is the *within-class* scatter matrix, and $\mathrm{S}_b \in \mathbb{R}^{n \times n}$ is the *between-class* scatter matrix. The within-class scatter matrix is given by

$$\mathrm{S}_w = \sum_{i=1}^{C} \sum_{j=1}^{N_i} (\mathbf{y}_j^{(i)} - \mu_i)(\mathbf{y}_j^{(i)} - \mu_i)^T \tag{6}$$

where C is the total number of classes (cross-section clusters), N_i is the number of samples in class C_i, $\mathbf{y}_j^{(i)} \in \mathbb{R}^n$ is the j^{th} vector of C_i, and $\mu_i \in \mathbb{R}^n$ is the mean of C_i. The between-class scatter matrix is given by

$$\mathrm{S}_b = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T \tag{7}$$

where $\mu \in \mathbb{R}^n$ is the ensemble mean. Note that $\mathrm{rank}(\mathrm{S}_b) \leq C - 1$ since it is the sum of C rank-one or rank-zero (if $\mu_i = \mu$) matrices, of which only $C - 1$ are linearly independent. The intuitive interpretation of Eq. (5) is that LDA attempts to simultaneously minimize the within-class scatter and maximize the between-class scatter. The well-known solution to Eq. (5) is given by the m generalized eigenvectors of S_b and S_w corresponding to the m largest eigenvalues, i.e., the columns of Ψ in Eq. (8) below that correspond to the largest values of the diagonal generalized eigenvalue matrix Θ:

$$\mathrm{S}_b \Psi = \mathrm{S}_w \Psi \Theta. \tag{8}$$

We note also that finding the generalized eigenvalue solution is equivalent to simultaneously diagonalizing S_w and S_b.[5] The simultaneous diagonalization process is accomplished (assuming S_w is non-singular) by whitening S_w, diagonalizing the resulting S_b, and then taking the largest eigenvalue eigenvectors of S_b. Intuitively, this process can be described as whitening the denominator of Eq. (5) and then maximizing the numerator over a reduced dimensionality. Since we use $C = 20$ cross-section clusters (implying $\mathrm{rank}\,\mathrm{A} \leq 19$) in our current implementation, the resulting transformation matrix A has dimensions 150×19 and the vectors in the 19-dimensional space are given by $\mathbf{z} = \mathrm{A}^T \mathrm{P}^T \mathbf{x}$. Hence, the final vector space where comparisons are performed has dimensionality of $m = 19$.

5. EXPERIMENTAL RESULTS

In this section, we report results obtained using the proposed system on real semiconductor data where different cross-section shapes were produced by varying the focus and exposure (producing a so-called focus/exposure or F/E matrix) of the lithographic tool. The available data set tested comprised five design rules, described as follows, with top-down images captured by one or more of three different CD-SEM tools:

- 100nm dense (2:1 pitch) lines, 47 cross-sections with 126 top-downs;
- 100nm isolated (5:1 pitch) lines, 94 cross-sections with 269 top-downs;
- 180nm dense (1:1 pitch) lines, 70 cross-sections with 201 top-downs;
- 180nm isolated (5:1 pitch), 88 cross-sections with 263 top-downs; and
- 250nm dense (1:1 pitch) lines, 113 cross-sections with 113 top-downs.

Hence, the complete set of available data comprised 412 cross-sections and 972 top-downs (complete top-down images, not sub-images). A few cross-sections and top-downs had to be discarded due to broken or non-existent lines or the lack of a corresponding top-down or cross-section image, respectively. The final test set contained 407 cross-sections and 958 top-downs. From these 958 top-down images, 6629 sub-images were extracted according the process of Section 3.

Hold-one-out type tests were performed by removing a single cross-section and all corresponding top-downs from the training data when computing the transformation matrix for dimensionality reduction as described in Section 4. Each of these hold-out top-downs was then submitted as a query. The corresponding cross-section was estimated via weighted averaging (described below) and compared to the true cross-section. This process was repeated for each of the 407 available cross-sections, corresponding to 958 different top-down queries.

Weighted averaging was employed to estimate the query cross-section, where the weighting is determined by the distances between the query and the K-nearest neighbor, historical top-downs (where various values of K were tested). The distance between a full query top-down and a full historical top-down is defined by the closest pair of sub-image feature vectors. In other words, let Q represent the full top-down query image with $q = 1, \ldots, S_Q$ sub-images and let H with $h = 1, \ldots, S_H$ sub-images be a top-down in the historical (training) database. The distance between Q and H, $D(Q, H)$, is then defined to be

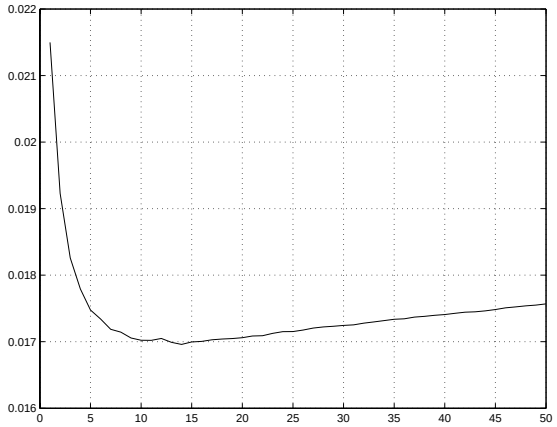
$$D(Q, H) = \min_{\substack{q=1,\ldots,S_Q \\ h=1,\ldots,S_H}} d(\mathbf{z}_q, \mathbf{z}_h) \tag{9}$$

where $\mathbf{z}_q$ and $\mathbf{z}_h$ represent the sub-image feature vectors, computing according to Sections 3 and 4, for sub-image q of full top-down Q and sub-image h of full top-down H, respectively. For the reported experiments, Euclidean distance was used for the distance measure $d(\cdot)$. For a given query image Q, $D(Q, H)$ was computed for every top-down H in the training set and sorted in ascending order. The cross-section curves corresponding to the closest K historical top-downs were used to estimate the query cross-section shape, $\hat{\mathbf{c}}(Q)$ (as a vector), as follows:

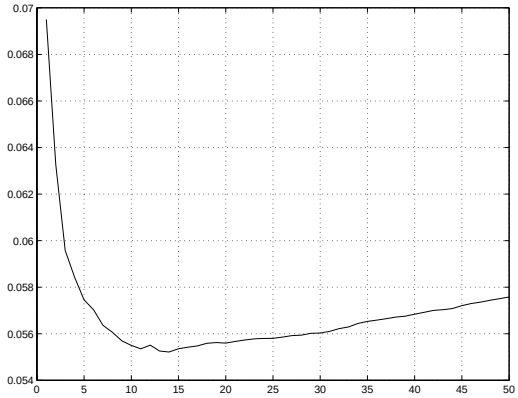
$$\hat{\mathbf{c}}(Q) = \sum_{i=1}^{K} \alpha_i \mathbf{c}(H_i) \tag{10}$$

where $\mathbf{c}(H_i)$ is the cross-section of nearest-neighbor i and the weighting factors are given by

$$\alpha_i = \Big(\sum_{j=1}^{K} \frac{1}{D(Q, H_j)} \Big)^{-1} \frac{1}{D(Q, H_i)} \tag{11}$$



(a) Average root-mean-square error (as a fraction of design rule line width) versus number of nearest neighbors used in Eq. (10).



(b) Average maximum absolute error (as a fraction of design rule line-width) versus number of nearest neighbors used in Eq. (10).

Figure 7. Average root-mean-square (a) and average maximum absolute (b) error over all hold-out data.

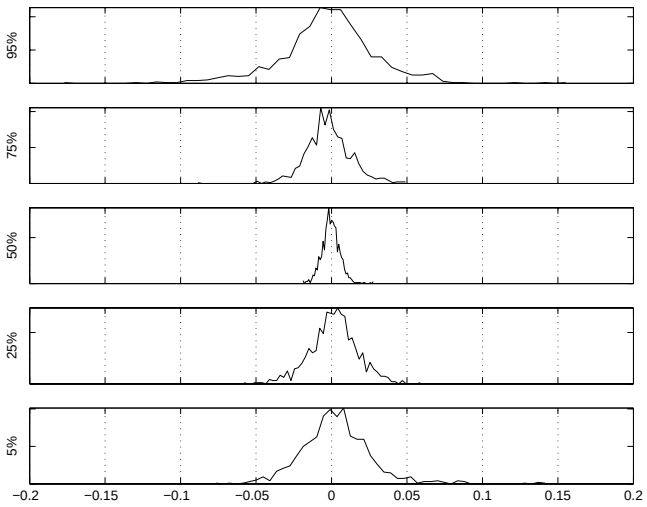


Figure 8. Cross-section error distributions (as a fraction of design rule line-width) at various points along the cross-section height (top of line structure is 100%, bottom is 0%).

so that $\sum_i \alpha_i = 1$. The number of nearest neighbors used in the tests was allowed to take on values $K = 1, \ldots, 50$.

We computed the root-mean-square and maximum absolute errors (normalized by the design rule line-width) for the estimated cross-sections of every top-down hold-out. The average of these errors across all 958 top-down queries is plotted against the number of nearest neighbors in Fig. 7, where the vertical axis represents the error divided by the design rule line-width (i.e., an error of 0.05 for a 100nm design rule implies a 5nm error). In Fig. 8, we show the error distributions (using all 958 hold-outs) at various positions along the vertical extent of the cross-section curves using $K = 11$ nearest neighbors.

6. CONCLUSIONS AND FUTURE WORK

We present a system for estimating semiconductor cross-section structure from top-down CD-SEM imagery. Results are reported on a variety of data sets comprising variations in line-width, line pitch, and CD-SEM tools for top-down image acquisition. The results indicate that cross-section structure can be estimated quite accurately from only top-down CD-SEM images and historical information. Experiments with 100nm, 180nm, and 250nm dense and isolated lines indicated an average root mean square error of about 1.7% (1.7nm for 100nm lines or 4.25nm for 250nm lines) and an average maximum absolute error of about 5.5% (5.5nm for 100nm lines or 13.75nm for 250nm lines). This cross-section estimation can be performed with existing inspection equipment (requiring only additional software) and does not necessitate physical cleaving of semiconductor devices after the initial historical data collection.

In ongoing work, we intend to explore some modifications of the proposed system with the goal of extending capabilities and/or improving performance. One such modification will involve removing the implicit dependence of the top-down features upon the actual, printed line-width. This dependence does not exist in the cross-section representation due to the normalization from Eq. (1); it is exhibited in the top-downs, though, since the sub-image features are effectively only normalized with respect to the design rule rather than the actual, printed line. We additionally intend to investigate the possibility of estimating of the actual height of the sidewall, which would provide the true aspect ratio. Furthermore, recalling that our experiments here used data across multiple processes (design rules) and top-downs from different CD-SEM tools, we would like to conduct further experiments to determine if accuracy can be improved using data from a single design rule and/or top-downs from a single CD-SEM tool. Finally, in an upcoming paper,[7] we will investigate the use of Gabor filter-based top-down features,[8] exracted along the line edges, and enhanced LDA algorithms[9,10] for dimensionality reduction.

ACKNOWLEDGMENTS

Funding for this effort was provided by International SEMATECH (ISMT). We would like to thank the member companies of ISMT for their support and guidance. We would also like to specifically thank Michael Bishop, Marylyn Bennett, and Hal Bogardus of ISMT.

REFERENCES

1. K. Tobin, T. Karnowski, L. Arrowood, R. Ferrell, J. Goddard, and F. Lakhani, "Content-based image retrieval for semiconductor process characterization," *EURASIP Journal on Applied Signal Processing* **2002**, p. 704, 2002.
2. P. R. Bingham, J. R. Price, K. W. Tobin, T. P. Karnowski, M. Bennett, and H. Bogardus, "Sidewall structure estimation from CD-SEM for lithographic process control," in *Process and Materials Characterization and Diagnostics in IC Manufacturing II*, SPIE, 2003. To appear.
3. J. R. Price and T. F. Gee, "Towards robust face recognition from video," in *30th Applied Imagery and Pattern Recognition Workshop (AIPR)*, pp. 94–100, October 2001.
4. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, pp. 711–720, July 1997.
5. K. Fukunaga, *Statistical Pattern Recognition*, Morgan Kaufmann, 1990.
6. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, second ed., 2001.
7. J. R. Price, P. R. Bingham, K. W. Tobin, and T. P. Karnowski, "Semiconductor sidewall estimation using top-down image retrieval," in *6th International Conference on Quality Control by Artificial Vision*, 2003. To appear.
8. S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on gabor filters," *IEEE Transactions on Image Processing* **11**, pp. 1160–1167, October 2002.
9. H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data – with application to face recognition," *Pattern Recognition* **34**, pp. 2067–2070, October 2000.
10. M. Loog, R. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**, pp. 762–766, July 2001.