

Supplementary Information for “Automatic Beam Optimization Method for Scanning Electron Microscopy based on Electron Beam Kernel Estimation”

Yunje Cho¹, Junghee Cho¹, Jonghyeok Park¹, Jeonghyun Wang¹, Seunggyo Jeong¹, Jubok Lee¹, Yun Hwang¹, Jiwoong Kim¹, Jeongwoo Yu¹, Heesu Jeong¹, Hyenok Park², Subong Shon¹, Taeyong Jo¹, Myungjun Lee¹, and Kwangrak Kim^{1,*}

¹ Metrology and Inspection Equipment R&D Team, Mechatronics Research, Samsung Electronics Co., Ltd., 1-1 Samsungjeonja-ro, hwaseong-si, Gyeonggi-do 18848, Republic of Korea

² Foundry Metrology & Inspection Technology Team, Global Manufacturing & Infra Technology, Samsung Electronics Co., Ltd., 1-1 Samsungjeonja-ro, hwaseong-si, Gyeonggi-do 18848, Republic of Korea

*Corresponding author: Kwangrak Kim; E-mail: kwangrak.kim@samsung.com; Tel.: +82-10-5154-4805.

Algorithm I | Beam Kernel Estimation Process

for i in range (number of sweep images):

$$f_i = \text{standard_normalization}(f_i)$$

$$u = \text{standard_normalization}(u)$$

for n in range(number of iterations):

update u :

$$\frac{\partial F_u}{\partial u} = \frac{1}{n} \sum_{i=0}^n [\overline{k_{\Sigma_i}} \otimes (k_{\Sigma_i} \odot u(x + v_i) - f_i)] - \lambda \nabla \cdot \frac{\nabla u}{|\nabla u|}$$

$$u \leftarrow u - \epsilon_u \frac{\partial F_u}{\partial u}$$

update kernels:

for i in range (number of sweep images):

$$\frac{\partial F_{\Sigma_i}}{\partial k_{\Sigma_i}} = (k_{\Sigma_i} \odot u(x + v_i) - f_i) \otimes \overline{u(x + v_i)}$$

$$\frac{\partial k_{\Sigma_i}}{\partial \Sigma_i} = \frac{1}{K} \int_{\mathbf{x}} -\frac{1}{2} k_{\Sigma_i}(\mathbf{x}) (\Sigma_i^{-1} - \Sigma_i^{-1} \mathbf{x} \mathbf{x}^T \Sigma_i^{-1}) d\mathbf{x}$$

$$\frac{\partial F_{\Sigma_i}}{\partial \Sigma_i} \leftarrow \text{clipping} \left(\frac{\partial F_{\Sigma_i}}{\partial k_{\Sigma_i}} \cdot \frac{\partial k_{\Sigma_i}}{\partial \Sigma_i} \right)$$

$$\Sigma_i \leftarrow \text{clipping} \left(\Sigma_i - \epsilon_{\Sigma} \frac{\partial F_{\Sigma_i}}{\partial \Sigma_i} \right)$$

$$k_{\Sigma_i} = \text{get_kernel_image_from_Gaussian_covariance}(\Sigma_i)$$

update v :

for i in range (number of sweep images):

$$v_{i1} \leftarrow -\frac{\sum_{\text{pixels}} [(k_{\Sigma_i} * u_x) \cdot (k_{\Sigma_i} * (v_{i2} u_y + u) - f_i)]}{\|k_{\Sigma_i} * u_x\|_2^2}$$

$$v_{i2} \leftarrow -\frac{\sum_{\text{pixels}} [(k_{\Sigma_i} * u_y) \cdot (k_{\Sigma_i} * (v_{i1} u_x + u) - f_i)]}{\|k_{\Sigma_i} * u_y\|_2^2}$$

$$\epsilon_u \leftarrow \eta_{\text{decay}} \cdot \epsilon_u$$

$$\epsilon_{\Sigma} \leftarrow \eta_{\text{decay}} \cdot \epsilon_{\Sigma}$$

Supplementary Algorithm 1 | Algorithm 1 illustrates the beam kernel estimation process, demonstrating the operations required to single update the parameter set of SEM. Please refer to the manuscript for the meanings of each character and operational symbol. The function named *clipping* refers to the process of multiplying a specific scalar with a matrix, preventing the elements of the matrix from exceeding a certain limit in both covariance and its gradient. This method is employed to clip the matrix, mitigating overshooting and enhancing the convergence speed. The function named *get_kernel_image_from_Gaussian_covariance* extracts an image of a specified size of Gaussian kernel using covariance matrix.