

Segmentation of chip microimages by contour tracking

Chochia P.A.

Institute for Information Transmission Problems of the Russian Academy of Sciences

(Kharkevich Institute), Moscow, Russia

chochia@iitp.ru

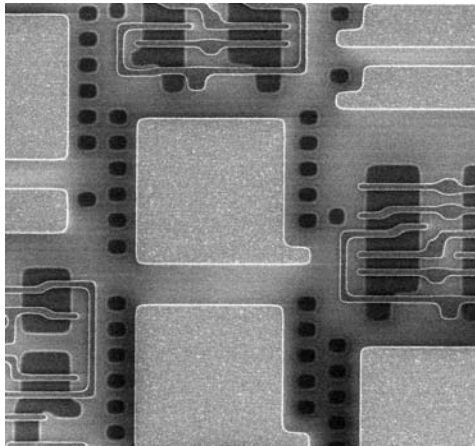
Abstract

The segmentation of chip microimages, obtained by scanning electron microscope is considered. In addition to high noise level these images are characterized by the presence of two rather different types of borders between the patterns, where the width of the first-type borders varies in enough large range. The algorithm for transformation and segmentation of these images is proposed.

Keywords: Image Segmentation, Image Processing, Contour Detection, Contour Tracking.

1. INTRODUCTION

The developing of microelectronic devices needs the wafer layer studying. One of the ways is to analyze the wafer surface picture, captured by Scanning Electron Microscope (SEM). The task we consider is the detection of all patterns in the scanned data, i.e. the separation of the image to the set of segments. The image, obtained by SEM, is interpreted as two-dimensional monochrome signal; the example is represented in the picture. The approach to segmentation is based on detecting and tracking contour lines at the borders of patterns; the peculiar property of source images is the presence of two border types: light contrast ridges of different width (the first type), and brightness edges (the second one).



To detect the border between patterns we follow contour-tracking approach [1, 2]. Contrary to filter-based algorithms [3, 4] it has some advantages for our task, because it allows to apply the analyze window of various size and shape, and to obtain continuous and connected lines of minimum thickness.

Two types of pattern borders need different approaches to their detection and tracking. Basically on two-component image model [5], appropriate image conversion transforms the second border type to the first one and allows using only one contour tracking

algorithm. Then this half-stuff is subjected to filtration and correction, and the final contour map is obtained.

2. SOURCE IMAGE PREPROCESSING

Under the preprocessing source image is transformed to the set of values, reflecting the contour existence probability in image pixel, that is near zero values in inner pattern areas and near ones at the ridges between them; we will name it *zero-aligned image*.

2.1 Aligned image with first-type contours

Zero-aligned image may be obtained by unsharp masking method [3, 5] as the difference between an image and its smoothed copy: $M_s(x) = x - S_s(x)$. Here x is the pixel of the source image X , $S_s(x)$ is smoothing operator response (mean value of X over $v \times v$ pixels fragment, centered at x point), and $M_s(x)$ is the result of filtration. The ridges width and contrast have large variety. To reduce some failures and to improve the result, the operator $S_c(x)$ is applied to the image with previously removed light ridges. This can be done with the following clipping procedure: $x_c = x$, if $M_s(x) \leq t$, and $x_c = M_s(x)$, if $M_s(x) > t$. To reduce the image noise level, the median filtering [3] procedure $x_m = \text{med}\{x \in \mu\}$ over 3×3 pixels vicinity μ was used. The image background alignment is carried out by subtracting smoothed clipped and filtered image X_m from the source image X : $A(x) = x - S_s(\text{med}(x_m))$. If the difference is less than zero, $A(x)$ becomes equal to 0.

2.2 Aligned image with second-type contours

The gradient filter is applied to clipped and filtered image X_m to obtain zero-aligned image with second-type contours. This filter response equals to pixel values span over 3×3 vicinity, i.e. to the difference $x_g = \max\{x_m \in \mu\} - \min\{x_m \in \mu\}$. The fluency of the first-type contours is suppressed by using the clipped image x_m .

2.3 Directional map

The directional map contains the information about contour trend in every image point. These data are used later for contour tracking. As known, the pixel values variance has its minimum in the direction along the contour, and maximum — in the perpendicular one. Therefore contour direction at a point can be defined as the direction of minimum pixel variation. In the algorithm we chose 4 directions for analysis: 0° , 45° , 90° и 135° .

3. CONTOUR TRACKING ALGORITHM

Contour tracking is carried out over zero-aligned image together with the directional map. After finding the contour threshold and detecting the set of initial contour tracking points, the contour lines tracking is performed. It includes: finding the (next) initial contour tracking point and interpret it as the current tracking point, choosing the contour direction at this point, finding the coordinates of the next point, defining ridge profile parameters, drawing the line segment to the next tracking point, calculating

the line profile parameters, and jumping to the next tracking point.

3.1 Contour threshold

Contour tracking may be started from any contour point, therefore it is not necessary to detect all or the majority of contour points; one point is enough. Initial contour tracking points set is found by thresholding previously prepared zero-aligned image. This threshold is found by analysis of the pixel values distribution.

3.2 Contour direction and the next tracking point

The first step in finding sequent contour tracking point is the choice of the contour direction. Mean pixel values are calculated inside 3 sectors of 45°: in the sector that extends the current contour direction, and in two adjacent. The direction with the maximum of calculated values is chosen as new contour direction.

The width of the real ridges may vary in large range, and possible direction set is also enough coarse. Therefore the next tracking point coordinates need clarification realized by finding the central point of the ridge. This is done by finding the point, maximizing the correlation between ridge profile and rectangular aperture. Simultaneously with the line tracking the parameters of profile asymmetry A_{aver} and convexity C_{aver} are calculated.

3.3 Contour line leg drawing

The contour line leg is drawn from the previous point (i_n, j_n) to the next one (i_{n+1}, j_{n+1}) , simultaneously extra initial points are cleared in some "corridor" near the line. The corridor width d is chosen $d = 2u' + 4$, where u' is the half-width of the ridge. The line drawing process finishes when current line leg touches another contour line or the image border. In such a case the algorithm starts new line tracking process beginning from the next initial point. Otherwise tracking point coordinates (i_n, j_n) are replaced by new ones (i_{n+1}, j_{n+1}) , and the algorithm repeats tracking procedure.

4. UNITED CONTOUR TRACKING

For united tracking of the first- and the second-type borders, the intermediate contour map X_1 with first-type contours is integrated with aligned image X_2 containing second-type contours. Simultaneously the "corridor" of the same width $d = 2u' + 4$ near the tracked lines of X_1 is cleared for removing the parasitic lines in the image X_2 . It may be written as: $x = x_1$, if $x_1 > 0$; $x = 0$, if x is in u' -vicinity of the contour; and $x = x_2$ otherwise. Then the tracking algorithm, discussed above, is applied to new united image X .

5. CONTOUR MAP PROCESSING AND FILTERING

5.1 Open lines prolongation and closing

Open line prolongation up to its closing is performed by the same contour tracking algorithm, discussed in Section 3. The algorithm is applied to new synthesized image X , being combined as the weighted sum of pixels of the first- and second-type aligned images (z_1 and z_2) with the map of already found contours (m) by the following formula: $x = k_1z_1 + k_2z_2 + k_3m$. Here k_1, k_2, k_3 are the weight coefficients, and x is the synthesized image pixel value.

5.2 Segment numbering

Segment numbering is carried out by attributing some index (segment number) to every pixel of obtained contour map image. Simultaneously the numbering and verification of the resulting

segments (that is the inner areas, separated by contour lines) is executed, and residuary open lines, if any, are removed.

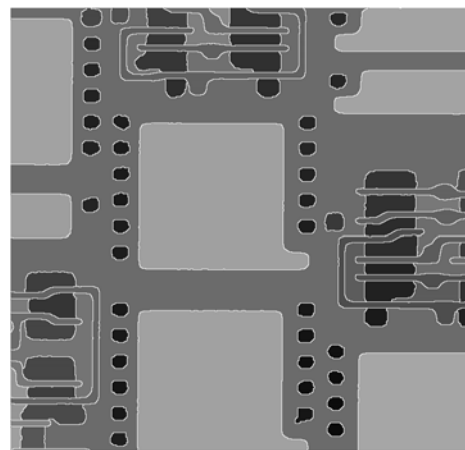
5.3 False lines and segments removing

Two types of false contour lines are possible. First of them (open line) has at least one non-connected end. Second one is correct from the connection point of view, but topologically it not constitutes the border and entirely places inside one segment. The algorithm verifies the segment numbers at both sides of every contour line. If the numbers are equivalent, the line is interpreted as false one and should be removed. This is realized simply by replacing its pixels by the value of neighbor segment pixel.

Due to the presence of noise, some false segments may appear as formally correct contour lines. For their detection and removing the contour line profile parameters of asymmetry A_{aver} and convexity C_{aver} , discussed in the subsection 3.2, are used. These values are compared with some thresholds T_A and T_C . If $C_{aver} < T_C$ or $A_{aver} < T_A$, the line is considered as false one and removed.

5.4 Final outcome

The segmentation result is represented in the picture below. The segments are displayed as constant value areas, filled by mean area brightness, and separated by thin white contour lines. The overall accuracy of detecting the segments is about 98%.



6. REFERENCES

- [1] Roberts J.M. *Attentive Visual Tracking and Trajectory Estimation for Dynamic Scene Segmentation*. Ph.D thesis., Univ. of Southampton, 1994.
- [2] Smith P., Drummond T., Cipolla R. *Edge tracking for motion segmentation and depth ordering* // Proc. 10th British Machine Vision Conf., V. 2, Nottingham, 1999, pp. 369–378.
- [3] Gonzalez, R.C., Woods R. E. *Digital Image Processing*. 2nd ed., Prentice Hall, New Jersey, 2002.
- [4] Фурман Я.А., Кривецкий А.В., Передреву А.К., и др. *Введение в контурный анализ и его приложения к обработке изображений и сигналов*. Физматлит, 2003.
- [5] Чочия П.А. *Двухмасштабная модель изображения* // Кодиров. и обработка изображений. М.: Наука, 1988, с. 69-87.