Automatic Analysis of CMP Dishing in Via Arrays from AFM Images

Jan Langer[1,2], Andreas Zienert[1,2], Doreen Hensel[1],
Lutz Hofmann[1], Knut Gottfried[1,3], and Tom Rothe[1]

[1] Fraunhofer Institut für Elektronische Nanosysteme (ENAS), Chemnitz, Germany
[2] Technische Universität Chemnitz, Chemnitz, Germany
[3] ErzM-Technologies UG, Chemnitz, Germany

INTRODUCTION

Hybrid wafer bonding is one of the most advanced 3D integration technologies. It uses metal-filled vias with a well-defined, very narrow permitted range of metal recess (dishing) provided by chemical mechanical planarization (CMP) [1]. The via dishing and via roll-off factors must be monitored precisely, which can be achieved by atomic force microscopy (AFM). Hence, the automated evaluation of AFM data is key for in-line process control but also one of the major challenges.

In this work, we propose an image-processing pipeline that detects a grid of vias in the AFM image based on expected ranges for via radius, grid pitch in vertical and horizontal direction, and grid rotation angle.

Once the positions of all vias in the grid have been obtained, we can run several further analysis steps, such as determining the average dishing level inside the via perimeter or roll-off factor from the substrate to copper. We also compute a statistical summary across all vias of the grid, across different measurements on the same wafer, and across multiple wafers in a lot. The results are used to automatically generate customer reports.

MANUAL DISHING ANALYSIS

As shown in Fig. 1, we define the via dishing as the difference between the average height of all the pixels inside the via circle and a baseline height, which is the average height outside the via circle. Accordingly, the via roll-off factor is defined as the slope between the via edge and the maximum height between two adjacent vias.

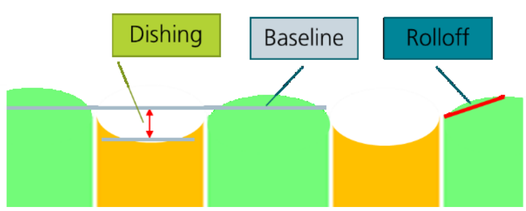

Fig. 1: Illustration of the via dishing and via roll-off factor.

In the previous manual metrology flow, a commercial software tool reads the AFM images, which are stored in a proprietary text format defined by the equipment manufacturer. The software then performs a line-by-line levelling of the images, as described in the next section. Afterward, for each row of vias an analyst draws a single line in the image, trying to cross approximately the middle of each via. Only the pixels of that line will be used for the latter analysis, making it rather fragile. In the next step, the analyst places via boundaries. However, distinguishing the edges between the copper via and the surrounding substrate is especially difficult in the case of shallow dishing. Finally, the tool computes the average height of the pixels inside via boundaries and as a baseline the height of those pixels outside the via. The difference between the two is the dishing parameter. Similarly, the roll-off factor is computed using the same 1D line segment.

This semi-manual process is error-prone and extremely time-consuming. To apply it to the large amounts of images that arise during mass production we need to automate the process and increase robustness for noisy images or images with hard to distinguish via boundaries.

OBJECTIVE

A major goal of automating the via analysis was moving away from a 1D line-based analysis towards a 2D analysis. In order to achieve that we needed to find the via circles inside an image. We quickly found, that identifying single via circles in the images with classical circle detection algorithms such as Hough transformation [2,3] lacked the robustness to detect all vias in the grid. For example too smooth via boundaries, irregular void patterns or circle-like structures in the substrate worsen the performance of the algorithm and result in undetected or misplaced vias. Here, an algorithm that utilizes the regularity of the grid to find the whole grid at once will still recognize hard-to-identify single vias in an expected grid location s and ignore off-grid circle pattern in the substrate.

In the following, we present such an algorithm that efficiently finds the grid that best fits on the image. Based on the optimal grid we then perform a 2D via analysis and compute the dishing for all vias as well as the roll-off factor for all via gaps. Executing in real-time it allows for interactive via analysis.

We will explain the algorithm on an example image, using pixels as vertical and horizontal dimension and an arbitrary height unit. For demonstration purpose, the example image is of outstanding quality, while in reality, the images are more heterogeneous, containing noise, artefacts, and manufacturing imperfections.

ALGORITHM

Image pre-processing

The original AFM images as shown in Fig. 2 a) typically have substantial vertical and horizontal drifts, for example due to surface irregularities of the wafer. This is illustrated by the line profile in Fig. 3 a) for the blue line marked in the 2D image. Hence, we need to perform a so-called line-levelling equivalent to the one of the commercial tool.

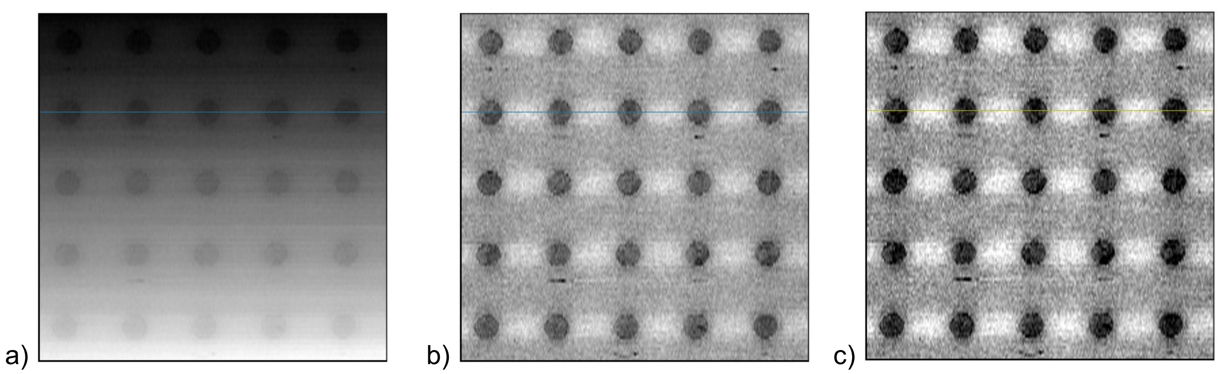

Fig. 2: Example AFM image with a 5x5 via grid. Original image a), levelled image b), and equalized image c).

Therefore, each profile line is fitted separately with a first-order polynomial (orange line in Fig. 3 a)) from which it is then subtracted. Fig. 2 b) and Fig. 3 b) show the results respectively, with new color mappings adapted to the dynamic range of the image. Only for the visualization of the results, we clip the lowest and highest percentile of the pixels to equalize the image as seen in Fig. 2 c) and remove outliers.

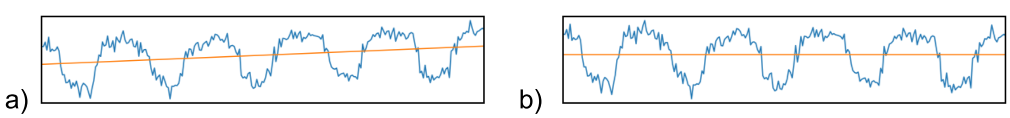

Fig. 3: Marked line profile before a) and after b) line levelling.

Parameter space definition

The parameter space for a regular grid of vias has eight parameters listed in Table 1. For now, we disregard trapezoidal grids, as it would increase parameter space even further. Furthermore, the parameter range is narrowed through previous knowledge of the manufacturing process limitations. The grid center point and the angle are caused by the AFM measurement due to a slightly rotated or misaligned wafer. This leaves a 6-dimensional parameter space to be searched, which typically has between 1 and 30 million data points.

Table 1: Overview of the grid parameters.

Parameter	Description	Example selection
(n_x, n_y)	Number of vias in the grid	Single fixed values (5,5)
r	Via radius	Target radius +/-2 pixels
(p_x, p_y)	Grid pitch	Target pitch +/-2 pixels with 0.2 pixel resolution
(x, y)	Grid center point	All pixels where all the vias of the grid are inside the image
α	Grid angle	Between -1° and +1° with 0.5° resolution

Edge detection

In order to identify a circle based on the edge between the copper via and the substrate, we need to perform an edge detection algorithm. We can use a simple Sobel operator in the horizontal and vertical directions and then compute the magnitude of both values. In Fig. 4, we show the result of the edge detection algorithm. We clearly see the actual vias, but some additional artefacts that can throw off a Hough circle detection.

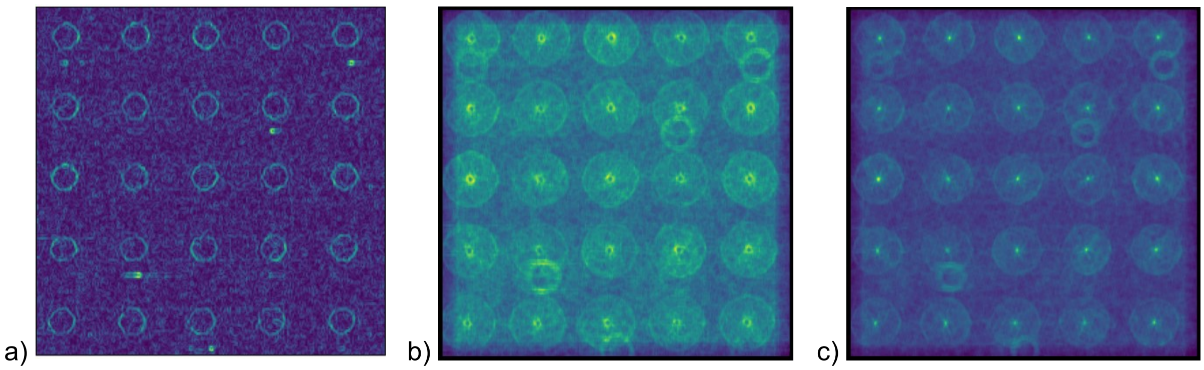

Fig. 4: Sobel edge detection on the example image a). Hough transform for r=9 b) and r=11 c).

Hough circle transformation

The next step is performing a Hough circle transform based on the Sobel edge detection. In some implementations, the Hough transform is based on a Canny edge detection, which binarizes the Sobel image by “walking” along the high ridges of keeping the gradual slopes. However, in this context this works not reliable, because we actual want the gradual edge values to detect also the less obvious via circles.

The Hough circle transform iterates over every possible radius in our parameter space, and over every pixel in the image and computes a value that indicates how probable a circle with that radius in the location actually is. The value is simply the sum of all pixels along the perimeter of the circle. The algorithm scales linearly with the number of pixels in the images, the number of different radii to evaluate, and the radius itself, because the perimeter length is proportional to the radius. In our example image, the algorithm ran for 1.5s on 256x256 pixels and five radii between 9 and 13 pixels.

We define the Hough circle transform result as a 3-dimensional tensor

$$H(r, x, y) \in \mathbb{N}^{|r| \times 256 x 256}.$$

For each radius, the result can be visualized as an image of the same size as the input. We show two of those images in Fig. 4 b) and c). The optimal radius, in this case, is 11, where the corresponding transform result shows clear and focused peaks in the center of each via. In contrast, the image for radius 9 has less focussed peaks with lower values.

Brute force grid search

While the next step is not of ultimate algorithmic beauty, it works well for limited parameter spaces. We simply iterate over all possible combinations of the parameters shown in Table 1, generate all center points of the corresponding grids, and sum the values from the Hough transform. Finally, the best fitting grid is the one with the maximum value in W.

$$W(r, p_x, p_y, \alpha, x, y) = \sum_i^{n_x} \sum_j^{n_y} H(r,\ \ x + i \cdot p_x \cdot \sin(\alpha) + j \cdot p_y \cdot \cos(\alpha),\ \ y + i \cdot p_x \cdot \cos(\alpha) - j \cdot p_y \cdot \sin(\alpha))$$

For the example image, the parameter space has ~3.2 million data points and runs in 0.95 seconds on a standard office laptop. Fig. 5 shows the results.

a) b)

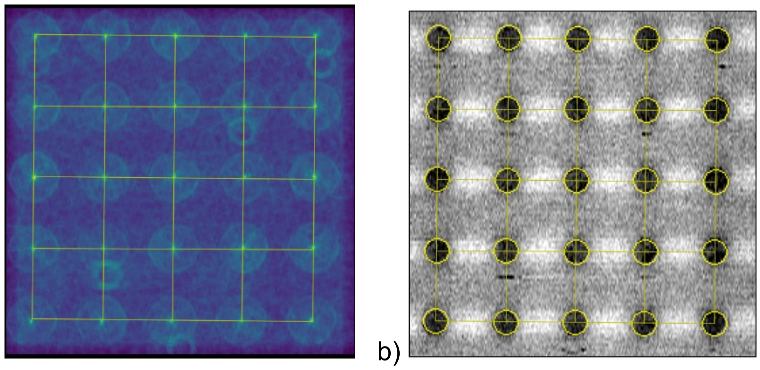

Fig. 5: Optimal grid of vias annotated on top of the r=11 Hough transform a) and the equalized example image b).

Dishing and roll-off analysis

Given via locations from the previous step, we define a circle with an inner radius $r_i = f_i \cdot r$ smaller than the via radius (typically f_i is 80%) to avoid taking into account also the voids and other via boundary effects. We average the height of all pixels inside that circle. Similarly, we define an outer radius $r_o = f_o \cdot r$ (typically f_o is 120%) and define a rectangular box between the outer radii of two adjacent vias as the baseline box. The vertical extent of the box corresponds to the diameter of the detected via. The average height of all pixels inside that box is our baseline value. The difference between the two average heights is the dishing value. In Fig. 6, we illustrate that. The yellow circle represents the detected via radius, the red area the inner circle used for dishing computation, and the blue box between the dashed outer circles the baseline box.

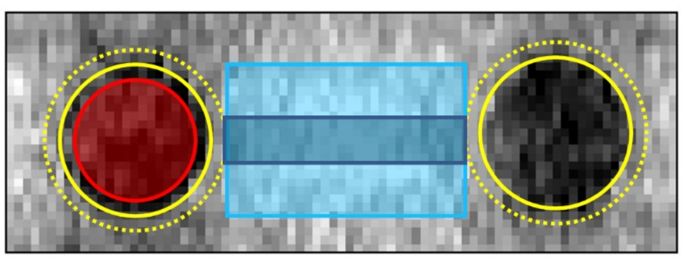

Fig. 6: Dishing and roll-off factor analysis in the 2D image. The red circle inside the via and the light blue baseline window are used for computing the dishing value and the dark blue strip is used for the roll-off factor.

In order to compute the roll-off factor we define another rectangular box vertically centered in the baseline box with a fixed height (5 pixels in our example). The pixels in this roll-off box are averaged vertically, such that only a single line is obtained. The dark blue box in Fig. 6 illustrates the roll-off box. For the same example, Fig. 7 shows the plot for the five lines in the roll-off box in grey and the averaged line in dark blue. Over that averaged line, a polynomial of second order is fitted as shown in red. Now the roll-off factor can be calculated as the slope between the via edge and the maximum height (dashed lines).

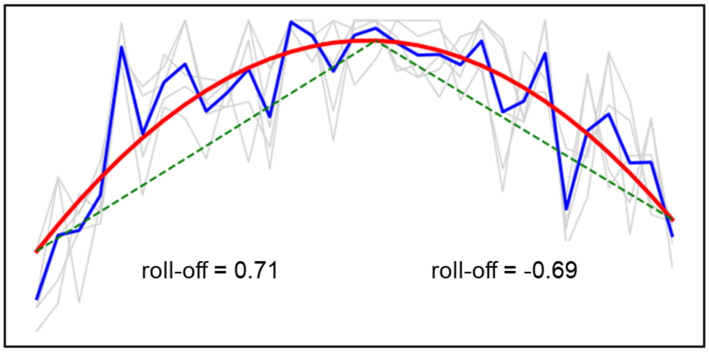

Fig. 7: Illustration of the roll-off factor computation.

IMPLEMENTATION

The complete code is implemented in Python. We read the AFM images using the pySPM library, which supports various proprietary formats such as the one of our AFM equipment. The algorithmic uses the NumPy library, exploiting the full potential its broadcasting rules to achieve a performance that is similar to an optimized C implementation. The report generation includes Microsoft Excel, CSV and graphical reports.

In order to provide the analysis tool to our metrology analysts, we developed a web-based form using Streamlit. It allows uploading one or more images at once, setting the target parameters such as the expected radius and grid pitch, and generating the corresponding report automatically, with the push of a button.

CONCLUSION

We implemented an efficient algorithm that automatically analyzes the dishing and roll-off factor in AFM images after CMP. The algorithm is part of our in-house AFM analysis software suite, assisting the process engineers on a daily basis and saving hours in comparison to manual analysis with conventional software tools. We plan to extend our AFM analysis suite with additional algorithms for various use cases.

BIBLIOGRAPHY

[1] Wenbiao, R. (2009). Effects of pattern characteristics on copper CMP. Journal of Semiconductors, 30(4), pp. 046001-1 - 046001-5.
[2] Hough, P. V. C. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654.
[3] Illingworth, J., Kittler, J. (1987). The Adaptive Hough Transform. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9, pp. 690-698.

Corresponding Author:
Jan Langer
Tel: +49 371 45001 641
E-mail: jan.langer@enas.fraunhofer.de
Fraunhofer Institute for Electronic Nano Systems (ENAS)
Technologie-Campus 3
09126 Chemnitz, Germany