

Segmentation d'images de semiconducteur pour la détection de défauts

P. BOURGEAT^{1,2}, F. MERIAUDEAU², K.W. TOBIN¹, P. GORRIA²

¹Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6010, USA

bourgeatpt@ornl.gov

²Laboratoire Le2i, 12 rue de la fonderie, 71200 Le Creusot, France

Résumé – Ce papier présente un algorithme de segmentation adapté aux images de semiconducteur générées par un outil optique d'inspection. Le but principal de ce travail est la segmentation d'images pour une sélection automatique du niveau de seuillage pour l'inspection « die-to-die » de wafer électronique. Les différentes structures produisent des niveaux de bruit différent lors de la détection de défaut. Il est donc nécessaire de générer automatiquement un masque de ces différentes régions afin de déterminer un niveau de seuillage local, et optimiser le rapport signal sur bruit. Une segmentation supervisée basée sur la transformée en ondelettes est utilisée afin de créer le masque d'une puce entière. Pendant l'inspection, le masque est appliqué sur l'image de différence, et le seuil est automatiquement déterminé, basé sur le niveau de bruit à l'intérieur de chaque région, et un coefficient spécifique au type de structure. La segmentation permet d'augmenter le nombre de défauts trouvés, et de diminuer le nombre de fausses détections.

Abstract – This paper introduces a segmentation algorithm suitable for semiconductor wafer images generated by optical inspection tools. The primary application of this work is content-based region segmentation for automatic threshold selection during recipe generation in die-to-die wafer inspection. Structures associated with different functional areas lead to different levels of noise in the difference image during the defect detection process. The ability to automatically create a mask to separate the different structures and materials is necessary to determine local thresholds for each area and thus to improve the signal-to-noise ratio. A supervised segmentation based on the discrete wavelet transform is used to segment a whole die to create a mask. During the inspection, the mask is applied on the difference image, and the threshold is automatically set as a function of the noise within the region and the thresholding coefficient specific to that region. The use of the segmented region in content-based threshold defect detection improves the number of defects detected, and reduces the number of false detections.

1. Introduction

Dans le monde du semiconducteur où les dimensions deviennent de plus en plus petites, la détection de défauts est un réel challenge [1], [2]. Ce papier présente une méthode de segmentation d'image qui convient à ce type de texture dans le but d'améliorer la détection de défauts [3], [4].

Lors de l'inspection d'un wafer à l'aide d'un microscope optique, la méthode la plus généralement utilisée est appelée « die-to-die inspection ». Elle repose sur l'hypothèse que deux puces voisines sur un même wafer sont identiques, à l'exception des défauts produits lors de la fabrication. Deux images provenant de la même région sur deux puces voisines sont soustraites pour créer une image de différence. A partir du niveau de bruit présent sur l'image de différence, un seuil est défini de manière à optimiser le rapport signal sur bruit. L'image de différence est ensuite seuillée pour localiser les défauts. Dans le cas où différentes structures sont présentes sur la même image, le niveau de bruit créé par chacune de ces structures sera différent dans l'image de différence. Il est donc nécessaire d'appliquer un seuil différent pour chaque structure générant un niveau de bruit différent. La segmentation des différentes structures du wafer est utilisée pour déterminer ces régions, et créer des masques dans lesquels le niveau de bruit permet de définir un seuil optimal pour chaque type de structure.

Les structures présentes sur un wafer créent des textures artificielles, généralement régulières, orientées dans les directions privilégiées horizontale et verticale. Ces propriétés conduisent naturellement au choix de la décomposition en ondelette [5] par l'algorithme à trous [6], [7]. Ce choix repose sur la grande aptitude des ondelettes à séparer localement les textures de fréquence donnée, dans les directions horizontales et verticales. L'utilisation de l'algorithme à trous permet d'ajouter aux ondelettes l'invariance en translation indispensable en segmentation. Le classifieur utilisé est celui des polytopes de contraintes [8] pour ses performances en terme de rapidité de traitement, et sa facilité d'implémentation sur une structure FPGA.

2. Correction d'image et extraction des paramètres

2.1 Prétraitement des images

Avant l'extraction des paramètres, il est nécessaire de corriger les non uniformités des images causées d'une part par le système optique, et d'autre part dues à des variations présentes physiquement à la surface du semiconducteur.

L'éclairage des images par « bright field » crée une illumination non uniforme de type gaussien, où l'image est beaucoup plus lumineuse en son centre qu'à la périphérie. Les variations d'épaisseur des couches de matériaux sur le wafer, ainsi que les variations de focus créent des modifications locales du contraste. Ces variations d'illumination et de contraste sont corrigées à l'aide des statistiques locales au voisinage de chaque pixel. L'amplitude de chaque pixel est divisée par la somme de la valeur moyenne et de l'écart type pris dans un voisinage 5*5. La figure 1 montre une image non traitée où l'on peut voir clairement la variation d'illumination, et la figure 2 montre l'image corrigée.

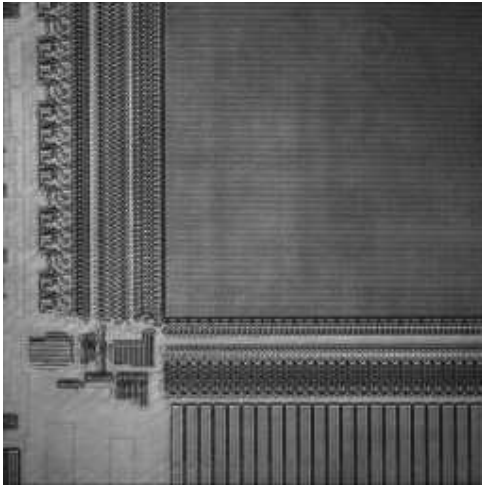


FIG. 1 : image brute

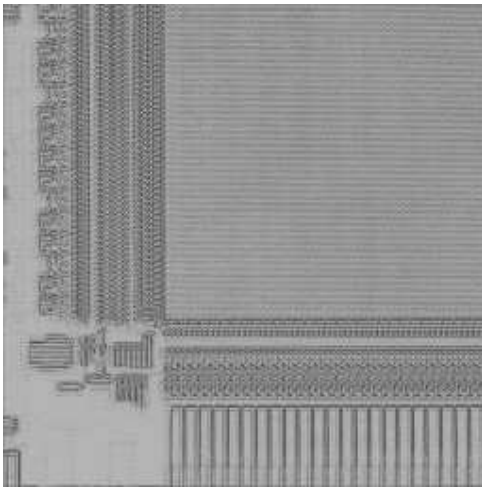


FIG. 2 : image corrigée

2.2 La transformée en ondelettes

L'image corrigée est ensuite décomposée en ondelettes sur trois niveaux, en utilisant les filtres de Coifman [9]. Seuls les détails horizontaux et verticaux sont conservés ; les détails diagonaux n'étant pas suffisamment discriminants, et contiennent l'essentiel du bruit. L'utilisation de l'algorithme à

trous crée une redondance d'information qui permet de maintenir l'invariance en translation à tous les niveaux de décomposition. De plus, elle permet une bonne localisation spatiale des fréquences les plus basses. Pour chaque niveau de décomposition, la décimation de l'image est remplacée par un sur-échantillonnage des filtres d'ondelettes qui est réalisé pratiquement par l'insertion de zéros entre chaque coefficient du filtre, comme présenté sur la figure 3. Ceci permet de conserver un temps de calcul constant à chaque niveau, car la convolution est effectuée sur le même nombre de coefficients significatifs pour chaque niveau.

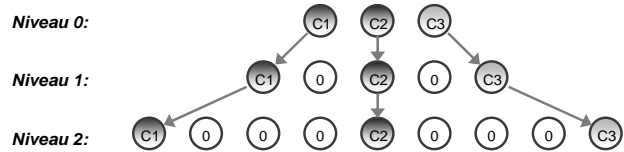


FIG. 3 : Algorithme à trous

Les coefficients d'ondelette ne sont pas utilisables directement dans un classifieur car leur réponse n'est pas uniforme pour une fréquence donnée. L'information est extraite à l'aide d'une estimation locale de l'écart type, dans une fenêtre glissante de 17 par 17 pixels. Cette phase d'extraction des paramètres nous permet d'associer à chaque pixel un vecteur à 6 dimension qui est ensuite traité par le classifieur.

3. Classification

Les polytopes de contraintes permettent d'effectuer une segmentation supervisée, basée sur le découpage de l'espace des paramètres en hypercubes à l'aide des points d'apprentissages. Ces hypercubes sont ensuite fusionnés afin d'en réduire le nombre et augmenter la rapidité du traitement.

3.1 Apprentissage

Un hypercube est créé autour de chaque point d'apprentissage de manière à n'inclure que des points de la même classe, et à n'intersecter aucun hypercube de classe différente. Les frontières sont définies en utilisant la distance du Max. Considérant deux vecteurs $X(x_1, \dots, x_N)$, et $Y(y_1, \dots, y_N)$, la distance $l(X, Y)$ entre les vecteurs X et Y est définie par :

$$l(X, Y) = \|X - Y\|_{\infty} = \text{Max}_{1 \leq p \leq N} |x_p - y_p| \quad (1)$$

Dans chaque direction, la distance la plus courte avec un point de classe différente permet de définir le plus proche voisin. Afin de définir les frontières, cette distance est ensuite multipliée par un coefficient R , avec :

$$0 < R < \frac{1}{2} \quad (2)$$

La valeur de R permet de régler l'écart entre les hypercubes. Avec R proche de 0, l'écart sera maximal, et la fusion sera facilitée. Avec R proche de 0.5, l'écart est minimal, et la fusion devient plus difficile. Nous prenons

généralement une valeur de 0.3. Les relations (1) et (2) sont suffisantes pour garantir qu'il n'y a qu'une classe par hypercube, et que les hypercubes ne se chevauchent pas. Si aucun voisin n'est trouvé dans une direction, la distance utilisée dans la direction opposée est utilisée, ou dans le cas où elle n'est pas définie non plus, la plus petite des dimensions toute direction confondue est utilisée. Dès lors, les hypercubes sont complètement définis, et ne saturent pas l'espace des paramètres. Une fois tous les hypercubes complètement définis, une fusion est effectuée entre les hypercubes de la même classe. La fusion est effectuée si et seulement si le nouvel hypercube obtenu ne chevauche aucun hypercube de classe différente. Cette étape réduit grandement le nombre d'hypercube à gérer.

3.2 Segmentation

Pour la segmentation, les coordonnées de chaque nouveau vecteur sont comparées avec les frontières des hypercubes afin de déterminer celui auquel il appartient. Le nouveau point est associé à la classe de l'hypercube auquel il appartient. Dans le cas où le point n'appartient à aucun hypercube, nous prenons la distance Euclidienne la plus courte entre le nouveau vecteur et le vecteur moyen de chaque classe pour le classifier. La figure 4 présente un exemple de segmentation.



FIG. 4 : résultat de la segmentation

La segmentation est testée en segmentant la même zone de 11 images par 11 images sur 20 puces différentes. Chaque image fait 432 par 432 pixels et est segmentée pixel par pixel. Les résultats sont donnés dans le tableau 1. Avec un taux de mauvaise classification d'environ 2.76%, les résultats sont très prometteurs. Nous étudions ensuite l'intérêt de la segmentation sur la détection de défauts.

Les zones de mémoire tel que nous les étudions, sont composées d'une grille de logique, avec à l'intérieur des zones de DRAM (respectivement segmentées en bleu et en rouge sur la figure 4). La zone de DRAM génère très peu de bruit dans l'image de différence, alors que la zone de logique génère énormément de bruit dans l'image de différence. Nous

n'avons pas de statistiques sur la position des défauts dans la zone de logique, donc nous choisissons d'ignorer ces zones, et de chercher les défauts uniquement dans la zone de DRAM, tout en gardant à l'esprit que le nombre de défauts est dans les mêmes proportions pour les deux zones. Le tableau reporte le nombre de défauts réels trouvés dans la DRAM, contre le nombre total de défauts trouvés (défauts réels, et fausses détections) pour différentes valeurs de seuillage.

TAB. 1 : taux de mauvaise classification pour chaque puce

Puce	Taux de mauvaise classification	Puce	Taux de mauvaise classification
1	2.30%	11	2.24%
2	1.99%	12	2.22%
3	2.57%	13	2.61%
4	3.07%	14	3.69%
5	2.76%	15	2.84%
6	2.37%	16	2.28%
7	2.53%	17	2.49%
8	2.55%	18	3.51%
9	3.61%	19	3.36%
10	3.41%	20	2.88%

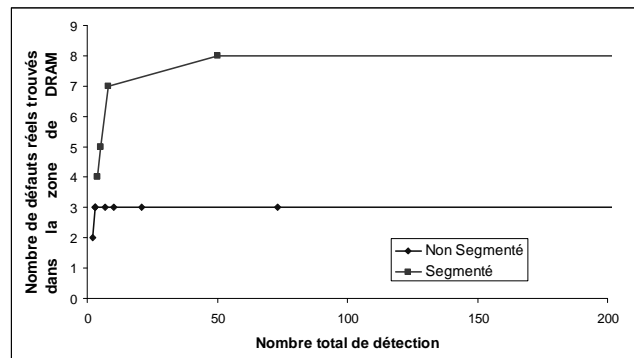


FIG. 5 : détection de défauts

Pour les valeurs de seuillage les plus hautes, la segmentation permet de trouver 7 défauts dans la DRAM sans générer de fausses détections, alors que sans segmentation, on ne peut pas trouver plus de 3 défauts dans la DRAM, et cela même avec un seuillage très bas

4. Conclusion

Nous avons démontré l'intérêt d'utiliser une segmentation sur les images de semiconducteur afin d'optimiser le rapport signal sur bruit dans chaque zone de l'image de différence. Nous avons développé un enchaînement d'algorithmes optimisé pour les images de wafer électronique, tout en répondant à des contraintes de rapidités, en utilisant des outils permettant une analyse en temps réel. La segmentation permet de générer rapidement un masque qui peut être directement appliqué sur les images de différence afin de déterminer le

seuillage optimal, et ainsi augmenter le nombre défauts trouvés, tout en diminuant le nombre de fausses détections.

Références

- [1] K.W. Tobin, *Inspection in Semiconductor Manufacturing*, Webster's Encyclopedia of Electrical and Electronic Engineering, vol. 10, pp. 242-262, Wiley & Sons, NY, NY, 1999.
- [2] P. Bourgeat, F. Meriaudeau, P. Gorria, K.W. Tobin, *Content-based segmentation of patterned wafer for automatic threshold determination*, Proc. SPIE, Vol. 5011, Santa Clara, January 2003.
- [3] P. Bourgeat, F. Meriaudeau, K.W. Tobin, P. Gorria, *Patterned wafer segmentation*, accepté pour publication au QCAV, Gatlinburg, Mai 2003.
- [4] *The National Technology Roadmap for Semiconductors: Technology Requirement*, Semiconductor Industry Association, 2001.
- [5] M. Unser, *Texture Classification and Segmentation Using Wavelet Frames*, IEEE Transactions on Image Processing, Vol. 4, n°11, pp 1549-1560, November 1995.
- [6] M. J. Shensa, *The Discrete Wavelet Transform: Wedding the A Trous and Mallat Algorithms*, IEEE Transactions on Signal Processing, vol. 40, pp 2464-2482, October 1992.
- [7] P. Dutilleul, *An Implementation of the algorithmes à trous to compute the wavelet transform*, in Wavelets: Time Frequency Methods and Phase Space. Berlin: Springer IPTI, pp 298-304, 1989.
- [8] J. Miteran, P. Gorria, M. Robert, *Classification géométrique par polytopes de contraintes. Performances et intégration*, Traitement du Signal, vol. 11, n°5, pp 393-408, 1994.
- [9] J. Tian, *The Mathematical Theory and Applications of Biorthogonal Coifman Wavelet Systems*, Ph.D. Thesis, Rice University, February 1996.