

増分符号相関を用いたサブピクセル単位での位置照合 Subpixelimage alihnment by Increment Sign Correlation

北大大学院 ○竹田健祐 金子俊一 田中孝之

要旨

半導体などの極小の世界では目に見えないほどの小さな欠陥が存在することから、画像照合の際の位置合わせにはサブピクセル単位での精度が要求される。しかし半導体では膜厚の変化などにより明度変化が起きてしまうため、よりロバストな照合法が求められる。そこで本研究では明度変化等にロバストな特長を持つ増分符号相関を用い、それをサブピクセルの範囲に対応させた方法を紹介する。

1. 目的と課題

本発表では、半導体ウェハ上に生じる欠陥を画像を用いて抽出する方法の第1段階である位置照合の手法を紹介する。欠陥抽出は隣り合うウェハの画像を同位置で比較して差を判定する。しかし同位置で撮影した画像でも、走行ステージの問題などにより、微小な位置のずれが起きる。そこでまず精度の高い位置合わせの手段が必要となる。半導体上の欠陥の中には撮影した画像のピクセル幅より小さなものも存在する事から、ピクセル単位での位置照合から更に細かなサブピクセル単位での位置照合が求められる。

しかし半導体では膜厚の差などからによる明度変化や、明度反転等が起こる事から、ロバストな画像照合の技術が求められることとなる。

そこで本研究では、明度変化や遮蔽等にロバストな特長を持つ増分符号相関を用いる事とした。増分符号相関とは、画素の輝度値をそのまま用いるのではなく、隣り合う画素との輝度値の差を2値化する事によって、照合判定を行う手法である。通常増分符号相関ではピクセル単位での位置合わせしか行う事が出来ない。そこで、増分符号相関の特長を生かしつつ、サブピクセル単位での位置合わせを行える方法を考案した。

2. 増分符号相関

本研究で用いている増分符号相関について簡単に説明する。増分符号相関とは、見本画像のビット列 $G = \{g_n\}_{n=1,2,\dots,N+1}$ に対応する長さ N のビット列 $B = \{b_n\}_{n=1,2,\dots,N+1}$ を作る。

$$b_n = \begin{cases} 1 & \text{if } (g_{n+1} \geq g_n) \\ 0 & \text{if } (g_{n+1} < g_n) \end{cases} \quad (1)$$

同様に、照合すべき対象画像を $B = \{b'_n\}$ に変換して、 b_n と b'_n の一致率を測る。この式は次式で定義される。

$$r_{isc} = \frac{1}{N} \sum_{n=1}^N \{b_n b'_n + (1 - b_n)(1 - b'_n)\} \quad (2)$$

この手法は輝度値情報を2値化するため、明度変化に強く、雑音の影響も受けにくく更に遮蔽にも強いといった特長を持つ。図1に半導体の実際の画像、図2にその画像の片方からテンプレートを取り出し、もう一方に対して増分符号相関を用いた r_{isc} の位置毎の分布を示した。画像から解るように、2つの画像には明度の差があるに

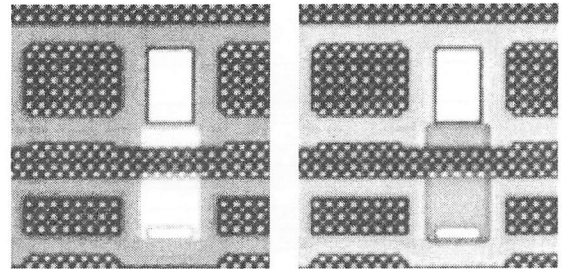


図1: 明度変化を起こした半導体の画像
※(株)日立ハイテクノロジーのご協力にて取得した画像

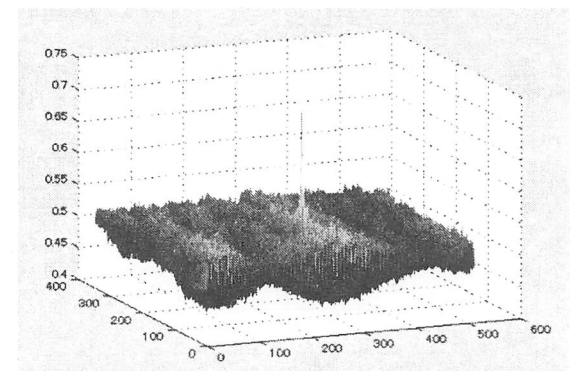


図2: ISC 値の分布

もかわらず、一致率を示す r_{isc} がある位置だけ急峻に増加している。この事から、増分符号相関の誤認識の低さが解る。

3. サブピクセル単位での位置合わせ

3.1 双線形補間法の適用

通常増分符号相関は、画像のピクセル単位での位置合わせまでは行える。しかし前述したように実際の生産環境では、サブピクセル単位での位置合わせが求められる。増分符号相関はデジタルで表された画像データに基づくため、画素間のデータは持ち合わせていない。

そこでサブピクセル単位での位置合わせの方法として、画素間を何らかの形で補間して、任意のサブピクセルの幅だけずらした画像を作成し、その画像とテンプレートとの増分符号相関を再度計算してやる事によって、サブピクセル単位での位置合わせを行う手法を考案した。

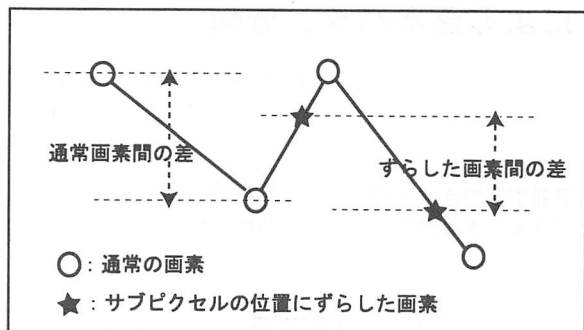


図 3: サブピクセルでの増分符号の取り方

図 3 が考案した増分符号の取り方のイメージである。画素の間を線形で補間し、全ての画素を同じ幅だけ補間線に基づいてずらし、その画像を用いて ISC 値を求める。1次元で考えるとこれで済む話であるが、画像の情報は2次元であるため、補間の方法は双線形補間を用いている。この方法を次式に示す。なお δ_x, δ_y は任意のサブピクセル単位のずれの幅である。

$$f' \quad (x \pm \delta_x, y) = f(x, y) + [\delta_x \{f(x \pm 1, y) - f(x, y)\}] \quad (3)$$

$$f'' \quad (x \pm \delta_x, y \pm 1) = f(x, y \pm 1) + [\delta_x \{f(x \pm 1, y \pm 1) - f(x, y \pm 1)\}] \quad (4)$$

$$f''' \quad (x \pm \delta_x, y \pm \delta_y) = f(x \pm \delta_x, y) + [\delta_y \{f(x \pm \delta_x, y \pm 1) - f(x \pm \delta_x, y)\}] \quad (5)$$

ここで f' はある画素を x 方向に δ_x だけずらしたものであり、 f'' はその画素の隣接画素を同じく δ_x だけずらしたものであり、 f''' はずらしたその2つの画素を用いて y 方向に δ_y だけずらした画素の輝度値の値となる。

これを式 (1) を用いてテンプレートと同じ画像の大きさだけ b'_n を再度求めてやり、 b_n との一致度を式 (2) を用いて求める。この方式を隣接画素との間で全探索してやることで最も一致率の高いずれ幅 δ_x, δ_y を出力する。

3.2 提案手法の有効性の検証

前節でも記述したように、画素間の正確な情報は解らないため提案したサブピクセル単位での位置合わせの有効性は、通常の画像では計る事ができない。そこで画素間の輝度値の変化の流れがあらかじめ解るような画像を作り、更にそこから任意のサブピクセル単位の幅だけずらした画像を作り、この2種類の画像を使って、0.01 単位でずらしていき、手法の有効性を検証した。

$$f(x, y) = 255(\sin(x)\sin(y) + 1)/2 \quad (6)$$

$$g(x, y) = 255(\sin(x + \delta_x)\sin(y + \delta_y) + 1)/2 \quad (7)$$

ここで $f(x, y)$ が \sin 関数に添った画像であり、 $g(x, y)$ は f を任意の幅 δ_x, δ_y だけずらした画像である。

照合の結果としては、 $g(x, y)$ のずれから最大で ± 0.04 の誤差が出た。この原因としては、画像の輝度値の流れは \sin 関数であるが補間方法は線形である事や、線形で補完する時は少数点以下も考えているが、画像データにするとときには整数のデータに変換しているため等が考え

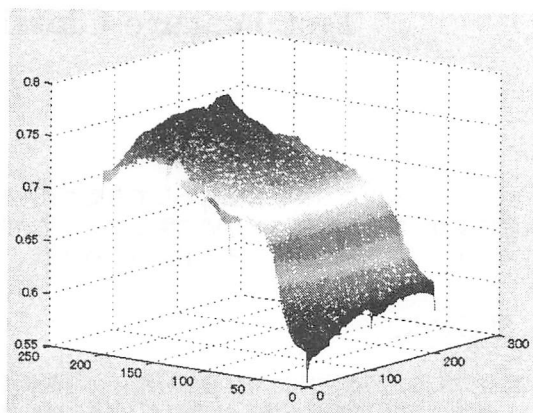


図 4: サブピクセルでの ISC 値の分布

られる。しかしこのぐらいの誤差であればこの手法の有効性は確かめられる。

また、この状態でも明度変化に対応できるかどうかを検証するために、式 (7) で作った画像を全体、部分的に10~40の明度変化を起させた画像を作り、再度効果を検証してみたが精度は変わらなかった。

ここで実際の画像である図1を用いてこの手法を実践し、ISC 値の分布を測った。図4がそのグラフである。ピークがいくつもあるわけではなく、ある局所でのみピークが存在している事が解る。

4. 考察と課題

今回の発表では増分符号相関を用いたサブピクセル単位での位置合わせの方法を考案し、その手法の有効性を確認した。

現段階では、サブピクセル単位での位置照合に拡張した手法でも増分符号相関が明度変化にロバストであり、この手法の方向性が有効であることが解った。しかし確認に使用した画像は実際の画像ではなく作成した画像であるため、どの程度まで実画像でも有効であるかが解らない。現に実際の画像に対しての結果を見てもある1点だけでピークが突出しているのではなく、高い値を出す範囲が広い。そういった意味でも実際の画像を用いた、手法の有効性の確認も必要であると感じた。

また現段階では全探索をしているためサブピクセル単位での位置検索においては 128×128 の画像で100秒強の計算コストがかかっているため高速化は必須の課題である。

参考文献

[1] 村瀬一朗, 金子俊一, 五十嵐悟, "増分符号相関によるロバスト画像照合", 電子情報通信学会論文誌, D-II, Vol.J83-D-II, No.5, pp.1323-1331, 2000年5月

[2] Takashi Hiroi, Shunji Maeda, Hitoshi Kubota, Kenji Watanabe, Yasuo Nakagawa, "Precise Visual Inspection for LSI Wafer Patterns Using Subpixel Image Alignment", IEEE Computer Society, 10662, Los Vaqueros Circle, P.O.Box 3014, Los Alamitos, CA 90720-1264.