

5T-4

ビデオヘッド用トラック幅測定
アルゴリズムの開発

高島 和男、金子 祥宏、伊藤 努
ソニー・マグネ・プロダクツ(株)

1. はじめに

現在、ビデオヘッドの面検査は、顕微鏡を見ながら目視により行われているため、測定精度および作業効率の面で様々な問題がある。そこで我々は、面検査自動化のための1stステップとして、検査の1項目であるトラック幅を、画像処理を用いて測定するための最適なアルゴリズムを開発したので、ここに報告する。

2. トラック幅測定

Fig.1 に示した様に左右のコアが重なり合った部分の幅 ($|ad|$: トラック幅)、左右コアのずれ幅 ($|ac|$: $X_1 \cdot |bd|$: X_2)を測定することをトラック幅測定と呼んでいる。

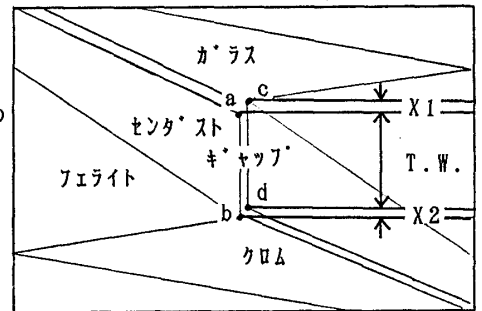


Fig.1 ヘッド構造図

3. システム構成

アルゴリズムの開発に使用したシステムの構成を次に示す。

- 顕微鏡 対物レンズ × 60
- コンピュータ データ処理用 MC68000
- データ収集用 PC-9801
- 画像メモリ 512×512×8(ビット)

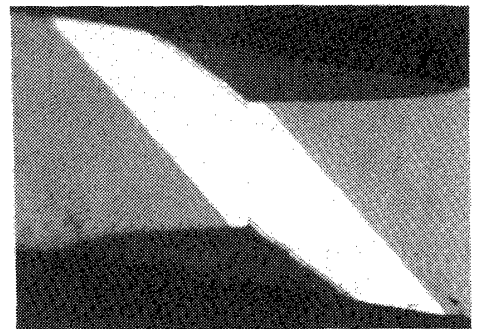


Fig.2 原画像

4. トラック幅測定のアプローチ

トラック幅は、Fig.1 に示した4点 (a・b・c・d) の垂直方向に関する画像座標を求め、その座標値から1画素当りの大きさを考慮し、実際のトラック幅を計算する。次に4点検出のためのアルゴリズムを述べる。今回開発したアルゴリズムは、次の5つのステップからなる。

4-1 ギャップ位置の検出

取り込んだ画像 (Fig.2) の輝度データを各列ごとに加算すると、Fig.3 に示したグラフになり、さらにこのグラフの差分をとると、Fig.4 に示したグラフになる。これより、最大値と最小値の座標を求め、それらの中間座標をギャップとし、ギャップ位置を検出する。

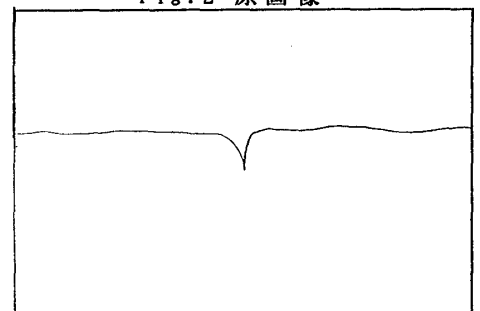


Fig.3 加算グラフ

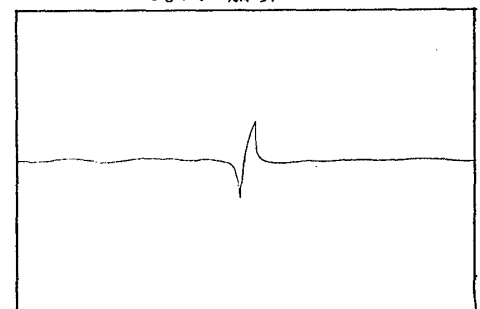


Fig.4 差分グラフ

4-2 左・右コアのエッジ検出

4-1 で求めたギャップの座標から、走査線方向に ± 10 画素の範囲にウィンドウを設定する。最初に、ウィンドウのデータに $(1, 0, -1)$ の微分フィルタをかけ垂直方向に加算し、加算値の最大値の座標を左コアのエッジとして検出する。同様に $(-1, 0, 1)$ の微分フィルタをかけ右コアのエッジを検出する。

4-3 4点検出のためのウィンドウ設定

ギャップから左右それぞれのコア方向に10画素の範囲で分走査線方向に輝度データを加算すると、Fig.5のグラフのようになる。左右それぞれのグラフから、最大値の9割を許容値とするラインと、上下最初に交わる点をそれぞれ $a' \cdot b' \cdot c' \cdot d'$ 点とする。Fig.6示す様に、 $a' \cdot b' \cdot c' \cdot d'$ 点を中心に上下 ± 40 画素、幅10画素のウィンドウを設定する。

4-4 ラベリング

4-3で設定した4つのウィンドウ内のデータを2値化する。Fig.7でもわかるように、ただ2値化した状態では、クロムの部分が残ってしまう場合が多い。そこでラベリングを行い、一定面積より小さなものを取り除く処理を行う(Fig.8参照)。この状態で、2値化による欠けを考慮し、4-2で求めた左右それぞれのエッジから1画素コア側のライン上を走査し、4つのウィンドウ内の最初のセンダストの位置を検出する。それを $A' \cdot B' \cdot C' \cdot D'$ 点とする。

4-5 真の4点検出

再びグレー画像に戻り、4つのウィンドウ内に 3×3 の垂直微分フィルタをかける(Fig.9参照)。4-4で検出した $A' \cdot B' \cdot C' \cdot D'$ 点の位置から各エッジ上で ± 5 画素の範囲内で、微分画像上の最大値の位置を検出する。これらの点を真の $A \cdot B \cdot C \cdot D$ 点とする。これらの4点をもとに、トラック幅を計算する。

5. まとめ

本研究で開発したアルゴリズムを用いることにより、ビデオヘッドのトラック幅を高精度(± 1 画素 = 約 $\pm 0.21[\mu\text{m}]$)かつ再現性良く($\pm 0.3[\mu\text{m}]$ 以内)測定できるようになり、信頼性が向上した。また、ウィンドウを設定し必要な領域だけを処理しているので、高速に処理できる(約1秒)。固定値を使用していないので、ある程度の光源変化は、測定に支障をきたさない。今後形状異常などの、より人間の行う検査に近いアルゴリズムの開発を行いたいと考えている。

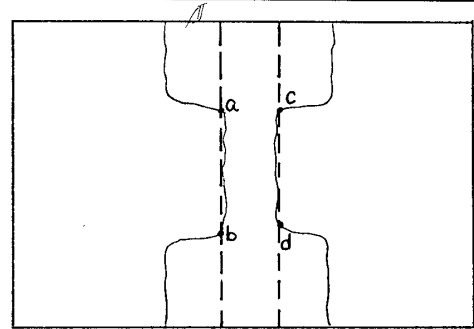


Fig. 5 横加算グラフ

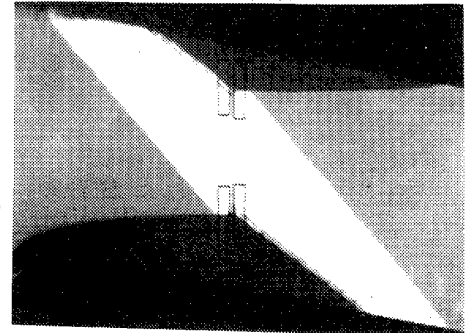


Fig. 6 ウィンドウ

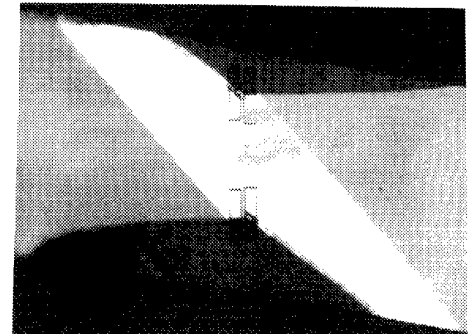


Fig. 7 2値化

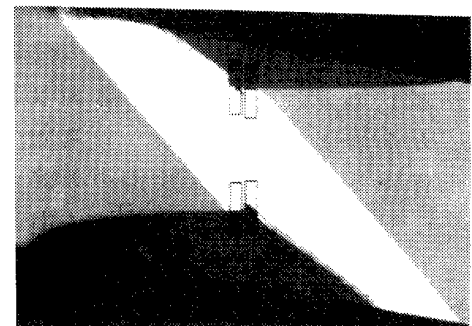


Fig. 8 ラベリング

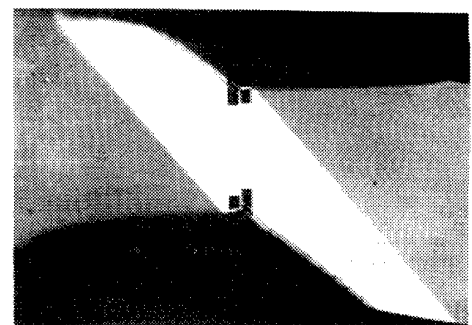


Fig. 9 微分フィルタ