

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3556076号
(P3556076)

(45) 発行日 平成16年8月18日(2004.8.18)

(24) 登録日 平成16年5月21日(2004.5.21)

(51) Int.Cl.⁷

H04N 1/04

F I

H04N 1/04 106A

請求項の数 1 (全 32 頁)

(21) 出願番号	特願平9-185414	(73) 特許権者	000006747
(22) 出願日	平成9年7月10日(1997.7.10)		株式会社リコー
(65) 公開番号	特開平10-107967		東京都大田区中馬込1丁目3番6号
(43) 公開日	平成10年4月24日(1998.4.24)	(74) 代理人	100078134
審査請求日	平成14年4月16日(2002.4.16)		弁理士 武 顕次郎
(31) 優先権主張番号	特願平8-185012	(72) 発明者	和田 真一郎
(32) 優先日	平成8年7月15日(1996.7.15)		東京都大田区中馬込1丁目3番6号 株式
(33) 優先権主張国	日本国(JP)		会社リコー内
(31) 優先権主張番号	特願平8-190998	(72) 発明者	野口 浩一
(32) 優先日	平成8年7月19日(1996.7.19)		東京都大田区中馬込1丁目3番6号 株式
(33) 優先権主張国	日本国(JP)		会社リコー内
		審査官	木方 庸輔
		最終頁に続く	

(54) 【発明の名称】 画像の位置誤差測定装置

(57) 【特許請求の範囲】

【請求項1】

有効画像領域外に設置され、主走査方向に対して一定の傾きの複数の線から構成された斜線パターンと、

該斜線パターンを有効画像領域内に載置された原稿とともに読み取る主走査方向に延伸した読み取り手段と、

原稿の変倍率を設定する設定手段と、

副走査方向の原稿走査速度を前記変倍率に応じて変更する副走査手段と、

前記読み取り手段によって得られた前記斜線パターンの画像データにウインドウを設定してウインドウ内の画像データの重心を計算する計算手段と、

該計算手段によって得られた重心の値によりウインドウの移動を制御する制御手段と、

前記重心の値と前記変倍率とに基づき、各ラインの画像の位置誤差のデータを得る演算手段と、

を備えていることを特徴とする画像の位置誤差測定装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、読取装置で読み取ったビットマップ形式の画像データの位置誤差を測定する装置に関する。

【0002】

10

20

【従来の技術】

この種の画像読取装置として例えば日本機械学会第71期通常総会講演会講演論文集（ⅠⅤ）で発表された「高精細画像入力装置の開発」（従来例1）が知られている。ここでは、副走査方向に並べて配置された等ピッチラインのテストチャートを読み取った画像、すなわち、副走査方向のライン間隔で離散化された画像データに対して補間演算を行い、演算された結果から、等ピッチラインの黒線、白線の中心位置を求め、テストチャートの基準ピッチとの差を読み取ることで、装置の振動などに起因する画像データの読み取り位置誤差を検出するようになっている。

【0003】

他の従来例として特開平6-297758号公報「走査線ピッチ計測方法」（従来例2）がある。この公知例は、等ピッチパターンのデータを書き込んだハードコピーのパターンを読み取ってハードコピー装置の書き込みの走査線のピッチムラを計測するようになっている。

10

【0004】

光学的リニアスケールとして、例えばオーム社から出版された「サーボセンサの基礎と応用」（大島康次郎、秋山勇治共著）〔昭和63年2月20日発行〕（従来例3）が知られている。この刊行物に記載された技術を図32ないし図34を参照して説明する。

【0005】

ここではリニアスケールの一例としてポジションスケールを例に挙げている。例に挙げられたリニアスケールは、図32に示すように全く等しいピッチの明暗の格子をもった2枚1組のメインスケール81とインデックススケール82とからなるガラススケール83と、そのスケール83を照明するLEDからなる光源84と、スケール83を透過した光を検知するフォトダイオード85から構成される。通常はインデックススケール82が固定され、メインスケール81が移動するが、その移動に連れてフォトダイオード85の出力が変化する。

20

【0006】

図33（a）に示すように2枚のガラスの透過部が一致したとき、出力は最大となり、透過部とクロム蒸着された不透明部86が重なったときには、出力は理想状態では0となる。したがって、その出力波形は理想的には図33（b）に示したような光量変化となるが、実際には明暗の格子ピッチが8μmと小さいため、光の回折の影響やクロム蒸着面での反射の影響があり、図33（c）に示すような出力波形のように近似正弦波の形で出力される。この出力波形の山の間隔がスケールのピッチに相当するので、山の数を数えることにより移動量を知ることができる。これがポジションスケールの基本原理であるが、実際には図32のフォトダイオードA、B、/A、/B（なお、「/」は反転を示す。）の4個を用いて各種の処理が行われている。

30

【0007】

A、B、/A、/Bのおのおのに対応するインデックススケール82の格子は、0°、90°、180°、270°の位相関係になっている。これをAと/A、Bと/Bを組み合わせて差動方式で検出し、スケール82の汚れや光量変化に対して強くなるように設定し、信頼性を高めている。このようにして得られた信号をおのおの改めてA、Bとし、さらに電氣的に反転された信号をそれぞれ/A、/Bとする。そして、これらの信号を用いてさらに細かい寸法まで読み取るための処理が実行されている。

40

【0008】

スケール83の移動方向は図34に示すようにA信号とB信号のどちらの信号の位相が進んでいるかを知ることによって判定できる。スケール83のピッチよりも細かく読む手法としては、A信号だけ用いると基準レベルを下からよぎるときと上からよぎるときの両方をとらえて4μm単位で読める。さらにB信号を用いると2μmまで読める。これ以上細かく読むためには、AとBの信号を用いて45°位相差の信号、Bと/Aの信号から135°位相差の信号を作る必要がある。

【0009】

50

ところで、上記従来例 1 では、等ピッチラインのパターンのエッジと読み取りのサンプリングのタイミングとの位置関係との相違により同じ形状のパターンを読み取って得られるデータがそれぞれ異なってしまうモアレという現象がある。読み取ったデータはこのモアレによって必ずしもパターンのエッジの位置と対応しないので、位置誤差の測定精度を劣化させる。モアレの影響は、等ピッチラインパターンを精細にして読み取り装置の分解能に近づけると非常に顕著になり、条件によっては位置誤差の測定ができなくなるほどになる。したがってこの方式では、読み取り装置の分解能に近い、あるいはそれ以下の位置誤差を高精度で測定することはできない。

【0010】

また、等ピッチラインのパターンを使うので、モアレの影響を無視しても、高い周波数成分の位置誤差を測定するためにパターンのピッチを精細にすると、結像光学系の MTF (Modulation Transfer Ratio) の限界によって得られた画像の濃淡の信号の差が小さくなり、測定精度が劣化せざるを得ない。

【0011】

さらに、パターンの精細化では、測定の周波数帯域を高い方向に広げ、精度を上げることができないのでサンプリングしたデータを補間する処理を行っている。より良い補間を行うには、より多くの周辺データを使ったり、複雑な演算処理が必要になり、処理時間が長くなる。さらに補間はあくまでも補間であり、真のデータとのずれが生じることは避けられず、測定精度を劣化させる要因になる。また、光電変換装置の中の特定の 1 つの受光素子が副走査方向に走査することによって得られる画像データを使用しているので、受光素子そのものが持つノイズが測定そのものの精度に影響を与えて、精度を劣化させる。

【0012】

従来例 2 では、計測時には光電変換装置でパターンを読み込んだデータを使うので、計測に当たってはハードコピーの読み取り時の走査ムラはないという条件で読み取ってハードコピーのピッチムラを計測している。その他、特に説明しないが、前述の従来例 1 と同様のモアレの問題を有する。

【0013】

従来例 3 では、上述のようなりニアスケールにおいては、光源 (LED) 84 の発する光をコリメートレンズ 87 で平行光線にしてメインスケール 81 とインデックススケール 82 の重なりを通過してくる光を受光素子で検出するようにしているので、微細かつ高精度のメインスケール、インデックススケール、および精密なコリメートが必要になる。その結果、当然コストも高くなってしまう。

【0014】

そこで、本出願人は、特願平 7 - 256481 号や特願平 7 - 311015 号において、これらの問題点を解決するための発明をすでに提案している。

【0015】

【発明が解決しようとする課題】

しかし、これらの先願に係る発明では、読み取り装置が等倍で読み取った場合を対象としており、変倍の場合についてまで配慮されてはいなかった。

【0016】

ここで、変倍時の現象について考える。通常、読み取り装置の変倍は、副走査方向はキャリッジの走査速度を変化させて変倍の画像を生成するのに対して、主走査方向の変倍は画像処理によって行っている。したがって、主走査方向の変倍処理の後段に位置誤差測定装置を置いた場合に、斜線パターンの読み取り画像の角度は、斜線パターンが 45 度の場合には主走査方向変倍後の画像も 45 度となるが、主走査方向変倍後の画像は、変倍の値に応じて斜線の主走査方向の線幅が変化する。すなわち、倍率が大きければ線幅も大きくなり、倍率が小さければ線幅も小さくなる。そのため、ウインドウの大きさを変倍率に応じて変化させる必要もでてくる。このようにウインドウの大きさを変倍率に応じて変化させると重心の測定時間が異なってきてしまい、位置誤差の測定に誤差が生じる場合も起こり得る。

10

20

30

40

50

【 0 0 1 7 】

本発明は、このような点に鑑みてなされたもので、その目的は、倍率が変化しても正確に位置誤差を測定することができる画素の位置誤差測定装置を提供することにある。

【 0 0 1 9 】

【課題を解決するための手段】

前記第 1 の目的は、有効画像領域外に設置され、主走査方向に対して一定の傾きの複数の線から構成された斜線パターンと、該斜線パターンを有効画像領域内に載置された原稿とともに読み取る主走査方向に延伸した読み取り手段と、原稿の変倍率を設定する設定手段と、副走査方向の原稿走査速度を前記変倍率に応じて変更する副走査手段と、前記読み取り手段によって得られた前記斜線パターンの画像データにウインドウを設定してウインドウ内の画像データの重心を計算する計算手段と、該計算手段によって得られた重心の値によりウインドウの移動を制御する制御手段と、前記重心の値と前記変倍率とに基づき、各ラインの画像の位置誤差のデータを得る演算手段とを備えた画像の位置誤差測定装置によって達成される。

10

【 0 0 2 2 】

【発明の実施の形態】

以下、図面を参照し、本発明の実施の形態について説明する。なお、以下の説明において、同等と見なせる各部に同一の参照符号を付し重複する説明は適宜省略する。

【 0 0 2 3 】

1. 第 1 の実施形態

20

1. 1. 測定原理

図 1 は、本出願の測定原理を典型的な場合を前提にして説明するための図である。図の主走査と書いた矢印 1 0 1 は線順次で画像を読み取る装置が同時に読み取る 1 ラインの画像の画素の並びと、この並列のデータを直線のデータに変換したときの時間軸上の順序を示す。図の副走査と書いた矢印 1 0 2 は主走査の 1 列が読み取る範囲を順次移動させながら読み取って行く方向を示している。移動する手段としては、原稿の画像を光電変換素子に投影するミラー、照明ランプなどを機械的に移動させるもの、原稿を移動させるもの、光電変換素子とその結像光学系を一体にして移動させるものなどがある。ここではこの主走査方向と副走査方向に平行な線で囲まれたそれぞれの 4 角形を画素ということにする。画素によって構成される平面は、原稿の画像を電気信号に変換されたデータが原稿の画像の写像がそのまま並んでいるというイメージでとらえることができ、ビットマップということもある。読み取り装置からリアルタイムで出力されるときには、主走査、副走査の方向が時間的な順序を示すが、出力されたデータをメモリに取り込んだ状態では、それぞれの画素を任意にアクセスすることも可能であり、主走査、副走査、時間の順序にとらわれない扱いも可能になる。

30

【 0 0 2 4 】

図 1 は主走査と副走査の画素サイズが等しい場合で、副走査方向の走査速度が変動するときと、一定速度で 45° の斜線を読み取るときに光電変換装置に投影されるが、像を全く劣化のないかたちでビットマップに対応させて示したものである。すなわち、a は副走査方向の読み取りのタイミングを制御するクロックに対応する所定の一定速度で走査したときで、ビットマップにも 45° の像ができる。b は速度が変動するときの像で、速度に応じて傾きが異なってくる。

40

【 0 0 2 5 】

つまり、A - B は副走査方向の走査速度が 0 のときで、副走査方向の読み取りのタイミングを制御するクロックにより副走査方向のビットマップのアドレスが進んでも原稿を読み取っている位置が変わらないため、副走査方向に平行な線になってしまう。

【 0 0 2 6 】

B - C は副走査方向の走査速度が所定の速度の $1/2$ のときで、ビットマップのアドレスが進んでも、その半分しか進まない位置の画像を読んでいることになり、画像の副走査方向の線との角度は $\tan \theta = 0.5$ から、約 26.57° である。

50

【0027】

C - Dは所定の速度で走査しているときで、傾きは45°である。同様にD - 以降は走査速度が1.5倍の場合で、その角度は約56.31°である。つまり、走査速度によって像の傾きが異なること、言い換えれば斜線の主走査方向への移動量が、副走査方向の移動速度に対応することを測定原理として副走査方向の移動速度のムラ、ミラー、レンズ、光電変換装置の振動などに起因するビットマップ画像の画素の位置誤差を計測する。

【0028】

以上、正方形の画素を持ち、45°の線を使用した場合で説明したが、画素が正方形でなく、例えば、主走査の分解能400dpi、副走査の分解能600dpiといった読み取り装置の画像データに適用することもでき、45°以外の斜線を用いても同様に、斜線の画像の主走査方向への移動量が副走査方向の読み取り方向の速度に依存するという関係は成立するので、画素の位置誤差を計測することができる。

10

【0029】

1.2. システム構成

図2は、本実施形態に係る位置誤差測定装置のシステム構成の一例を示すブロック図で、画像読取装置への付加機能として組み込み、リアルタイムでその位置誤差を測定するものである。このシステムは光電変換部1、A/D変換部2、シェーディング補正部3、斜線判別部4、位置誤差測定部5、制御部6、画像処理部7および操作部8から基本的に構成されている。

【0030】

光電変換装置1は、例えばラインCCDで、画像が電気信号に変換される。電気信号に変換された画像はA/D変換部(器)2でデジタルの多値の画像データに変換される。変換されたデータは、照明の不均一さ、レンズの周辺光量の低下、光電変換装置の画素間の感度の違いなどをシェーディング補正部3によってシェーディング補正される。シェーディング補正された画像データは、斜線判別部4に入力され、画像データの斜線パターン部分を判別し、その判別結果を制御部6に出力する。また、画像データは位置誤差測定部5に入力され、測定結果の誤差信号のデータ10を出力する。同時に画像データをビデオ信号11として画像処理部7に出力する。画像処理部7では、主走査方向変倍、MTF補正、γ補正などが行われる。それぞれの機能ブロックは、制御部6によってタイミングの制御、動作条件の設定などがなされ、相互に関連して動作する。また、操作部8からは制御部6に対してオペレータから変倍率の設定を含む種々の操作入力が行われる。なお、制御部6にはビデオ制御信号12が入出力される。

20

30

【0031】

光電変換装置として等倍センサを用いている読取装置ではレンズの特性による周辺光量の低下という問題がないので、シェーディング補正を省く場合があるが、そのような形式の読取装置にも、本願を適用することができる。

【0032】

1.3. 位置誤差測定処理

図3は図1と同様のビットマップに斜線の画像データK1, K2, K3があるときの位置誤差の測定を行うときの処理を説明するためのものである。W1は画像データK2の位置を求めるための演算を行う10×3のウィンドウである。ウィンドウ内のデータの位置を求めるため、主走査方向における重心を演算する。この演算では、順次ウィンドウの位置をW2, W3...と移動させながら重心を求める。重心の主走査方向の位置は45°の線の場合、画素の位置がなんらかの誤差要因で移動することがなければ、ウィンドウを図のように移動させた場合、主走査方向に1画素分ずつ移動するはずである。画素の移動量が1画素分と異なる場合は、何らかの原因で画素の位置が変動したことになり、位置誤差を求めることができる。位置誤差の主要な要因が副走査方向の走査速度のムラによることが分かっている場合には、位置誤差のデータが速度ムラにデータを変換することは容易である。

40

【0033】

50

重心を求めるのに周辺の画素のデータを含む多数の画素データを使っているので、CCD固有のノイズを始めとしてさまざまなノイズが画像データに含まれるが、重心を求める過程でノイズの影響が軽減され、S/Nの高い測定が可能になっている。通常、ウインドウの画素の数が多いほどS/Nは高くなる。

【0034】

ウインドウの形状は主走査方向の重心を求めることから、主走査側に大きいことが望ましい。副走査方向は1としても測定可能である。こうすることによって測定に要する時間を短くすることができる。同様にW2、W3とウインドウを移動させる。そして、あらかじめ同一ラインの測定回数をn回と設定しておき、 $i = n$ となったとき、すなわち、ウインドウWnに達したとき、その次のウインドウとして画像データK3のウインドウWn+1に移動させる。その方法としては、斜線の主走査方向の間隔に相当する画素分より正数画素少ない値mだけ主走査方向にウインドウの座標をシフトした後、カウンタ値iをクリアして斜線判別部4の処理に戻る。ここで、Wn+1の重心の値mom2とWnの重心の値mom1との差、
mom2 - mom1

が重心のズレmomとなる。以下、同様にWn+1、Wn+2、Wn+3・・・と移動させて位置誤差を測定する。

【0035】

このように複数の斜線を使って位置誤差を測定するようにすれば、読取装置の読取範囲が縦長であっても、副走査の全域にわたっての測定の可能になる。さらに主走査方向の狭い幅の中だけで測定を行うようにすれば、主走査方向における中央部とか、手前とか、奥側とかに分けて位置を誤差を測定することも可能になる。

【0036】

1.4. 斜線パターン判別処理

次に、斜線パターン判別処理について説明する。図4は図1と同様にビットマップに斜線が有る場合を示し、図5はその場合の8ビット(0~255)の読み取り値を示している。なお、0=白、255=黒であり、主走査方向の座標をXn、副走査方向の座標をYmとしている。また、図6は主走査方向3画素×副走査方向3画素の斜線パターン検知用ウインドウを示し、図6(a)~(e)はそれぞれ主走査方向に1画素ずつシフトしたウインドウを示している。

【0037】

ここで、図6(a)に示すウインドウ(X2~X4、Y1~3)内の中心画素を挟む対角方向、すなわち中心画素を含む左上斜め方向の3つの画素値の和Paと右下斜め方向の3つの画素値の和Qaを計算すると、

$$\begin{aligned} Pa &= (X2, Y1) + (X3, Y1) + (X2, Y2) \\ &= 3 + 1 + 1 = 5 \end{aligned}$$

$$\begin{aligned} Qa &= (X4, Y2) + (X3, Y3) + (X4, Y3) \\ &= 3 + 4 + 8 = 15 \end{aligned}$$

となる。

【0038】

同様に、図6(b)~(e)について求めると、

10

20

30

40

$$Pb = (X3, Y1) + (X4, Y1) + (X3, Y2) \\ = 1 + 4 + 2 = 7$$

$$Qb = (X5, Y2) + (X4, Y3) + (X5, Y3) \\ = 13 + 8 + 201 = 222$$

$$Pc = (X4, Y1) + (X5, Y1) + (X4, Y2) \\ = 4 + 2 + 3 = 9$$

$$Qc = (X6, Y2) + (X5, Y3) + (X6, Y3) \\ = 216 + 201 + 250 = 667$$

$$Pd = (X5, Y1) + (X6, Y1) + (X5, Y2) \\ = 2 + 18 + 13 = 33$$

$$Qd = (X7, Y2) + (X6, Y3) + (X7, Y3) \\ = 248 + 250 + 252 = 750$$

$$Pe = (X6, Y1) + (X7, Y1) + (X6, Y2) \\ = 18 + 220 + 216 = 454$$

$$Qe = (X8, Y2) + (X7, Y3) + (X8, Y3) \\ = 250 + 252 + 249 = 751$$

となる。

【0039】

次に、中心画素と右下斜め方向の3画素（中心画素を含む）の差Rを求めると、

$$Ra = 15 - 5 = 10$$

$$Rb = 222 - 7 = 215$$

$$Rc = 667 - 9 = 658$$

$$Rd = 750 - 33 = 717$$

$$Re = 751 - 454 = 297$$

となる。

【0040】

この差Rの値が大きい場合に3×3画素のウィンドウ内に斜線パターンが有ることを示す。したがって、例えばRの値が500以上の場合に斜線パターンが有ると判断すれば図6(c), (d)に示すウィンドウ内に斜線パターンが有ると判断することができる。

【0041】

次に、図7を参照して他の斜線パターン判別処理を説明する。図7(a)～(e)はそれぞれ図6(a)～(e)に示すウィンドウ内の各値を閾値=128で2値化した場合を示し、同様に各ウィンドウ内の中心画素の左上斜め方向の3つの画素値の和Pa～Peと右下斜め方向の3つの画素値の和Qa～Qeを計算すると、

10

20

30

40

$$\begin{aligned} P a &= (X 2, Y 1) + (X 3, Y 1) + (X 2, Y 2) \\ &= 0 + 0 + 0 = 0 \end{aligned}$$

$$\begin{aligned} Q a &= (X 4, Y 2) + (X 3, Y 3) + (X 4, Y 3) \\ &= 0 + 0 + 0 = 0 \end{aligned}$$

$$\begin{aligned} P b &= (X 3, Y 1) + (X 4, Y 1) + (X 3, Y 2) \\ &= 0 + 0 + 0 = 0 \end{aligned}$$

10

$$\begin{aligned} Q b &= (X 5, Y 2) + (X 4, Y 3) + (X 5, Y 3) \\ &= 0 + 0 + 1 = 1 \end{aligned}$$

$$\begin{aligned} P c &= (X 4, Y 1) + (X 5, Y 1) + (X 4, Y 2) \\ &= 0 + 0 + 0 = 0 \end{aligned}$$

$$\begin{aligned} Q c &= (X 6, Y 2) + (X 5, Y 3) + (X 6, Y 3) \\ &= 1 + 1 + 1 = 3 \end{aligned}$$

20

$$\begin{aligned} P d &= (X 5, Y 1) + (X 6, Y 1) + (X 5, Y 2) \\ &= 0 + 0 + 0 = 0 \end{aligned}$$

$$\begin{aligned} Q d &= (X 7, Y 2) + (X 6, Y 3) + (X 7, Y 3) \\ &= 1 + 1 + 1 = 3 \end{aligned}$$

$$\begin{aligned} P e &= (X 6, Y 1) + (X 7, Y 1) + (X 6, Y 2) \\ &= 0 + 1 + 1 = 2 \end{aligned}$$

30

$$\begin{aligned} Q e &= (X 8, Y 2) + (X 7, Y 3) + (X 8, Y 3) \\ &= 1 + 1 + 1 = 3 \end{aligned}$$

となる。

【 0 0 4 2 】

次に、中心画素と右下斜め方向の3画素（中心画素を含む）の差 $R a \sim R e$ を求めると、

$$R a = 0 - 0 = 0$$

$$R b = 1 - 0 = 1$$

40

$$R c = 3 - 0 = 3$$

$$R d = 3 - 0 = 3$$

$$R e = 3 - 2 = 1$$

となる。

【 0 0 4 3 】

したがって、この場合にも同様にこの差 R の値が大きい場合に 3×3 画素のウィンドウ内に斜線パターンが有ることを示し、例えば $R a \sim R e$ の値が2以上の場合に斜線パターンが有ると判断すれば図7(c), (d)に示すウィンドウ内に斜線パターンが有ると判断することができる。また、このように画素値を2値化することにより、加算演算を簡単にすることができる。

50

【 0 0 4 4 】

図 8 (a) ~ (d) は斜線パターン検出用のマッチングパターンを示し、図中の白領域は「 0」、黒領域は「 1」を表している。先ず、画像データを図 7 に示すように 2 値化し、その 2 値化データと図 8 (a) ~ (d) に示すマッチングパターンを比較し、合致した場合に斜線パターンがあると判断する。この例では、図 7 (c) と図 8 (b)、及び図 7 (d) と図 8 (a) が合致しており、このウインドウ内に斜線パターンがあると判断される。

【 0 0 4 5 】

なお、上記実施例では、ウインドウの大きさを 3×3 としたが、もちろんウインドウサイズが異なる場合にも同様な判断方法により斜線パターンを検知することができる。但し、一般にウインドウサイズが大きい程、判別精度は上がるが、その分処理時間が長くなり、また回路規模も大きくなる。

【 0 0 4 6 】

1 . 5 . ウインドウのデータと重心の計算

1 . 5 . 1 重心の測定処理

次に、重心の測定処理を説明する。図 9 に示す処理は原稿の走査開始と同時にスタートし、先ず、主走査方向、副走査方向の各座標値 X 、 Y がイニシャライズ ($X = 0$, $Y = 0$) される〔ステップ 9 0 1〕。この座標値 X 、 Y は斜線判別用の例えば 3×3 のウインドウ内のある画素位置例えば中心画素の座標となる。次に、1 本の斜線に対する測定回数を示す変数 i がイニシャライズ ($i = 0$) される〔ステップ 9 0 2〕。

【 0 0 4 7 】

次に斜線判別部 4 により斜線判別用の 3×3 のウインドウ内に斜線パターンが存在するかどうか判断され〔ステップ 9 0 3〕、存在しなければ、その 3×3 のウインドウを主走査方向に 1 画素分シフト ($X = X + 1$) する〔ステップ 9 0 4〕。なお、このシフト量はウインドウの大きさ、斜線の太さに応じて決められ、1 画素以上でもよい。ステップ 9 0 3 において斜線パターンが存在する場合には、重心測定用の例えば 10×3 のウインドウ W_1 を設定し、そのウインドウ W_1 内の重心を求める〔ステップ 9 0 5〕。このとき、ウインドウ W_1 の大きさ、斜線の太さに応じて、斜線と判別された画素の位置から主走査方向に整数画素分だけシフトし、斜線の部分がウインドウ W_1 の中心付近になるようにウインドウ W_1 を設定してもよい。

【 0 0 4 8 】

重心の測定を終了すると、重心のズレを計算し〔ステップ 9 0 6〕、次いで主走査方向に - 1 画素分、副走査方向に + 1 画素分シフトしたウインドウ W_2 を設定し、また、測定回数用のカウント値 i を 1 つインクリメントする〔ステップ 9 0 7〕。なお、この実施例では、ウインドウ W を 1 画素ずつ移動させているが、画素の位置誤差を起こす原因となる振動などの周波数帯域が低い場合には、2 画素以上ずつ移動させてもよく、この方法により測定に要する時間を短縮することができる。

【 0 0 4 9 】

次いで、予め設定された同一ラインの測定回数 n に対して $i = n$ とならない場合にはステップ 9 0 8 からステップ 9 0 5 に戻り、他方、 $i = n$ となった場合すなわちウインドウ W_n に達した場合には次の斜線のウインドウ W_{n+1} に移動させる〔ステップ 9 0 8 → 9 0 9〕。その方法としては、斜線の主走査方向の間隔に相当する画素分より整数画素 m だけ、ウインドウ座標を主走査方向にシフトした後、測定カウント値 i をクリアし〔ステップ 9 0 2〕、斜線判別処理〔ステップ 9 0 3〕に戻る。以下同様に、1 本の斜線に対してウインドウ W_{n+1} 、 W_{n+2} 、 W_{n+3} ~ のように移動させて位置誤差を測定する。

【 0 0 5 0 】

このように複数の斜線を用いて位置誤差を測定することにより、読み取り装置の読み取り範囲が縦長であっても、副走査領域の全域に渡って位置誤差を測定することができる。更に、主走査方向の狭い幅だけ測定するので、主走査方向の中央部、手前、奥側のように分

10

20

30

40

50

けて測定することもできる。また、高い分解能で位置誤差を測定する場合にも、斜線のパターンを細くする必要は全くなく、システムのMTFの制約を受けずに幅が広いパターンを用いることができる。

【0051】

更に、幅が広いパターンを用いた場合、幅に応じてウインドウも大きくなるので結果として測定精度を向上させることができる。したがって、斜線の幅は処理速度、リアルタイム処理を行う場合にはバッファのサイズ、回路規模の経済性などとのバランスを考慮して設定すればよい。また、幅が広いパターンを用いてその片側のエッジを検出することにより位置誤差を測定することができる。更に、例えば副走査方向の読み取りタイミングに関係なく白黒パターンを副走査方向に配列するとモアレの発生が問題となるが、本実施例では副走査方向の読み取りタイミングと斜線の関係は常に同じであるのでモアレの発生が問題とならず、その結果、高精度で位置誤差を測定することができる。

10

【0052】

1.5.2 ウインドウのデータと重心の計算

次に、ウインドウのデータと重心の計算について詳細に説明する。図10はウインドウデータと斜線パターンの各画素の読み取り値の関係を示し、読み取り値は8ビットであって10進(0~255)で示されている。主走査方向の重心を求めるには、副走査方向の各列(3ライン分)の和を求め、図に示すようにこれを左側からX0、X1、・・・X9とするとそれぞれ18、50、202、427、590、562、345、150、37、14を求める。そして、各画素の主走査方向の中心座標を左から順に0~9とし、主走査方向の重心位置をRmとすると、重心位置Rmの回りのモーメントは0になるので、 $Z_0(Rm - 0) + Z_1(Rm - 1) + \dots + Z_9(Rm - 9) = 0$ が成り立ち、数値を代入して計算するとRm = 4.362が得られる。

20

【0053】

重心を求める理由は、補間などの前処理を必要とせず、演算を簡素化、高速化することができるからである。また、画像位置を求める場合、各列毎のデータの和の並びから補間により所定の分解能のデータ列を得て、そのデータからピーク値が存在する位置を求める方法を用いることができる。

【0054】

1.5.3 複数本の斜線からなるチャートの重心の計算

30

次に、複数本の斜線からなるチャートの重心を計算する場合について説明する。図3に示すように複数本から成る斜線の重心を計算する場合、同一線上の線では問題とならないが、違う線にウインドウが移動したときには移動前と移動後では斜線の主走査方向の間隔が丁度、整数画素数でない限り重心の値が異なるので、補正しなければならない。一例として図3に示す斜線K2のウインドウWnの重心の値Rnが4.65となり、次の斜線K3に移動した場合のウインドウWn+1の重心の値Rn+1が4.38、ウインドウWn+2の重心の値Rn+2が4.40、ウインドウWn+3の重心の値Rn+3が4.41となった場合、ウインドウが移動したラインにおける重心の差ΔRを計算する。すなわち、

$$\Delta R = R_n - R_{n+1} = 4.65 - 4.38 = 0.27$$

40

となる。

【0055】

この値ΔRを斜線K3の重心の値に加算し、この加算結果を重心の値として位置誤差を求める。この場合、ウインドウWn+2の重心の値Rn+2、ウインドウWn+3の重心の値Rn+3は、

$$R_{n+2} = R_{n+2} + \Delta R = 4.40 + 0.27 = 4.67$$

$$R_{n+3} = R_{n+3} + \Delta R = 4.41 + 0.27 = 4.68$$

となる。したがって、このように複数本の斜線から成るチャートを使用しても、連続して高精度で位置誤差を測定することができる。但し、斜線K2のウインドウWnから斜線K3のウインドウWn+1に移動する場合、斜線K2、K3は主走査方向に同

50

時に存在しなければならない。

【 0 0 5 6 】

図 1 1 は斜線の配置関係を示し、長さ L_1 の複数の斜線が主走査方向に対して角度 θ で配置され、主走査方向の斜線の始点と終点の位置が同一の場合、主走査方向の斜線間隔を L_2 とすると、

$$L_2 < L_1 \times \cos \theta \quad \cdots (1)$$

の関係が成り立つように斜線を配置すれば、斜線は主走査方向には重なるので、ウインドウを主走査方向に移動して次の斜線の重心を連続して測定することができる。ここで、斜線の長さ L_1 と斜線の始点、終点の主走査方向の位置は式 (1) の大小関係が大きいほど精度を必要としなくなる。

10

【 0 0 5 7 】

1. 6. 位置誤差の測定手順

図 1 2 は位置誤差を測定するときの処理手順を示すフローチャートである。走査開始と同時に処理はスタートする。まず、ステップ 1 2 0 1 でイニシャライズ処理が行われ、主走査方向、副走査方向の座標値 x, y の初期座標値がセットされる ($X = X_START$, $Y = Y_START$)。この座標値は、ウインドウのある個所、例えば中心画素の座標となる。次に、斜線 1 ラインの測定回数のカウント値を格納する変数 r と読み取り画像全体の測定回数をカウントする変数 i がイニシャライズされる ($i, r = 0$)。さらに、ウインドウ内の重心値から各ラインの重心値を求める際に使用するシフト量を示す変数 s がクリアされる ($s = 0$)。

20

【 0 0 5 8 】

ステップ 1 2 0 2 では、斜線判別部 4 によってウインドウ内に斜線パターンがあるかどうかを判別する。もし、ウインドウ内に斜線パターンがなければステップ 1 2 1 1 で斜線判別用のウインドウを主走査方向に 1 画素シフトする。このとき、シフトする量はウインドウの大きさ、斜線の線の太さに応じて決められ、1 画素以上でもよい。斜線判別用ウインドウ内に斜線パターンがあれば、ステップ 1 2 0 3 で重心測定用のウインドウを設定し、ステップ 1 2 0 4 でその設定された重心測定用のウインドウ内の重心値 $p o s$ を求める。このとき、斜線がウインドウの中央にあるかどうかを確認するため、ステップ 1 2 0 5 で重心の値と図 1 3 ~ 図 1 5 に示すウインドウの幅 (X_W) の $1/2$ とを比較する。その差が 0 に近いほど斜線パターンがウインドウの中央付近にあることを示す。その差が + 1 以上のときは、ステップ 1 2 1 2, 1 2 1 3 でウインドウの座標を + 1 移動し、シフト量カウント変数 s に 1 加算する。その差が - 1 以下のときはステップ 1 2 1 2, 1 2 1 4 でウインドウの座標を主走査方向に - 1 画素移動し、シフト量カウント変数 s から 1 減算する。

30

【 0 0 5 9 】

以上の手順を繰り返して、上記差が 1 以下になったとき、次の手順〔ステップ 1 2 0 6 〕に進む。なお、この実施形態ではカウント値の差を「1」としているが、これに限定されるものではなく、ウインドウの大きさによって適宜設定される。

【 0 0 6 0 】

ステップ 1 2 0 6 で i ラインの重心 P が計算される。ここで、 $m o m$ は図 3 のように複数の斜線パターンを使用する場合の斜線間の重心のズレを補正する計数である。 i ラインの重心 P の値は、重心の値から重心のズレ補正值 $m o m$ を減算し、シフト量 s を加算して算出する。したがって、 i ライン目の重心 P とは、ライン L に対する重心を表している。

40

【 0 0 6 1 】

ステップ 1 2 0 7 では i ライン目の位置誤差 $M i$ を演算する。この $M i$ は i ラインの重心 $P i$ と $(i - 1)$ ラインの重心 $P i - 1$ の差分に変倍率を掛けたものと副走査方向に移動量 1 を $(i - 1)$ ラインの位置誤差に加算して求める。ステップ 1 2 0 8 では副走査方向 + 1 画素シフトしたウインドウを設定する。また、測定回数のカウント値 i, r をインクリメントし、シフト量 s に t を加算する。この t の値は、変倍率に応じて変わり、

$$t = 100 / \text{変倍率}(\%) \quad \cdots (2)$$

50

となる。すなわち、等倍（１００％）時は $t = 1$ 、変倍率２００％時は $t = 0.5$ 、変倍率５０％時は $t = 2$ となる。この t は変倍率の逆数であり、これは、直線 L の傾きを意味し、 y が副走査方向に１移動したときに x がどのくらい移動するかを示している。そこで、 t の値と倍率の関係が上記のようになる。

【００６２】

このステップ１２０７における M は実際に読み取られた位置を示す。図１６（ａ）は、倍率１００％で読んだときのビットマップパターンにおいて速度変動がないときには斜線は $S3$ （点線）のようになる状態を示している。この間に速度が倍に変動すると、４５度の斜線 $S3$ は斜線 $S4$ で示されるように読まれる。このとき、点 $T5$ に着目すると、この点 $T5$ は本来 $T7$ の位置で読むべき画像である。したがって、本来の読取位置は点 $T0$ から $L6$ の距離で点 $T5$ を読み取っていることとなる。一方、ステップ１２０７において、 $P_{i-1} - P_i$ は点 $T5$ の位置で斜線 $S3$ からの主走査方向の距離 $L4$ を示す。（ $P_{i-1} - P_i$ ）×変倍率は、斜線 $S3$ において主走査方向の距離を副走査方向の距離に換算する部分で、倍率１００％のときは１対１のため、 $L4$ は $L5$ の距離となり、 $L5 + 1$ は $L6$ （＝２）となる。さらに、ステップ１２０７においては、基準点（読取開始位置）からの距離を示しており、この場合には、点 $T0$ の基準点からの距離を示している。したがって、 M_{i-1} を加算することによって点 $T8$ の基準点からの距離が求められる。

10

【００６３】

図１６（ｂ）は、倍率１００％で読んだときのビットマップパターンにおいて速度が１／２に変動したときを示しており、斜線 $S2$ で示されるように４５度の斜線は読まれる。このとき点 $T1$ に着目すると、この点 $T1$ は本来 $T2$ の位置で読むべき画像であり、本来の読取位置は点 $T0$ から $L3$ の距離で点 $T1$ を読み取っていることとなる。そのため、ステップ１２０７における $P_{i-1} - P_i$ は、点 $T1$ の位置で斜線 $S1$ からの主走査方向の距離 $L1$ を示す。また、変倍率１００％であるため、 $L1 = L2$ となり、 $L3$ は $1 - L2$ （＝０．５）と求められる。

20

【００６４】

図１６（ｃ）は、倍率２００％で読んだときのビットマップパターンにおいて速度変動がないときは斜線 $S5$ （点線）のようになる状態を示している。この間に速度が倍に変動すると、斜線 $T6$ で示されるように４５度の斜線として読まれる。このとき点 $T9$ に着目すると、この点 $T9$ は、本来 $T11$ の位置で読むべき画像である。したがって、本来の読取位置は、点 $T0$ から $L9$ の距離で点 $T11$ を読み取っていることとなる。ステップ１２０７において、 $P_{i-1} - P$ は点 $T9$ の位置で斜線 $S5$ からの主走査方向の距離 $L7$ を示す。（ $P_{i-1} - P_i$ ）×変倍率は、斜線 $S5$ において主走査方向の距離を副走査方向の距離に換算する部分で、倍率２００％のときは１対２のため、 $L7$ は２倍の $L8$ の距離に換算され、 $L8 + 1$ が $L9$ （＝２）となる。このようにして、主走査方向の重心 P の移動が読取位置に換算される。

30

【００６５】

なお、この実施形態では、ウインドウを副走査方向に１画素ずつ移動させているが、画素の位置誤差を起こす原因となる振動などの周波数帯域が低い場合には、ウインドウを２画素以上ずつ移動させてもよい。そして、ステップ１２０９では、次の斜線に移るかどうか判断し、ステップ１２１０で最終ラインまで測定したかどうか判断する。もし、最終ラインまで測定していなければ、ステップ１２０４あるいはステップ１２０２に戻って、以降の処理を繰り返す。

40

【００６６】

１．７．主走査方向の斜線の画像の移動量を副走査方向の画素の位置誤差の関係

この実施形態では、副走査方向の画素の位置誤差を測定するために、斜線を読み取った画像の主走査方向へ画像の位置の移動を見ている。正方形の画素で４５度の斜線を使って測定する場合には、これまでの説明で明らかなように主走査方向の移動量のウインドウ間における偏差がそのまま副走査方向の位置誤差になる。画素が正方形でない場合、斜線の角度が４５度でない場合には、換算をして副走査方向の位置誤差を得る必要がある。その方

50

向について図 1 3、図 1 4、図 1 5 を参照して説明する。

【 0 0 6 7 】

図 1 3 は等倍 (1 0 0 %) 時の、図 1 4 は変倍率 2 0 0 % 時の、図 1 5 は変倍率 5 0 % 時の斜線のビットマップパターンをそれぞれ示す。なお、これらの図は中央付近で速度変動が発生した場合を示し、図 1 3 および図 1 5 は速度が遅くなった場合、図 1 4 は速度が速くなった場合の例である。

【 0 0 6 8 】

まず、図 1 3 における n ラインの制御について説明する。

ウインドウの重心 $p o s = 2$ 、重心のズレの補正量 $m o m = 0$ 、シフト量 $s = 1$ とすると、 n ラインの重心の値は、

$$3 (= 2 + 0 + 1)$$

となる〔ステップ 1 2 0 6〕。次に位置誤差の演算において、 $n - 1$ ラインの重心を $P n - 1 = 3$ 、位置誤差を $M n - 1 = 5$ とすると、 n ラインの重心は

$$M n = (3 - 3) \times 1 + 5 + 1 = 6$$

となる。

【 0 0 6 9 】

次いでステップ 1 2 0 8 では、シフト量 s に t 加算される。ここで、(2) 式により、等倍のため

$$t = 1$$

となる。したがって、

$$s = 2$$

となる。また、 y を 1 加算する。すなわち、ウインドウの y 方向に 1 ライン移動してステップ 1 2 0 4 にもどり、ウインドウの重心 $p o s$ を求める。ステップ 1 2 0 5 では、ウインドウ内の中心に斜線画像があるかどうかを判断する。ここで、 $p o s = 1$ であったとすると、

$$| 1 - 6 / 2 | = 2$$

となり、ステップ 1 2 1 4 に進む。次いで、ウインドウを x 方向に - 1 移動し、 s を 1 減算して $s = 1$ としてステップ 1 2 0 4 に戻り、重心を求めると、

$$P n + 1 = 2 - 0 + 1 = 3$$

となる。次に、誤差位置は、

$$M n + 1 = (3 - 3) \times 1 + 6 + 1 = 7$$

となる。そして、ステップ 1 0 8 で $s = 2$ となり、ウインドウを y 方向に 1 移動する。ステップ 1 2 0 4 では、ウインドウの重心 $p o s$ を求める。ここで $p o s = 1.2$ であったとすると、

$$| 1.2 - 6 / 2 | = 1.8$$

となり、ステップ 1 2 1 4 に進む。ステップ 1 2 1 4 では、ウインドウを x 方向に - 1 移動し、 s を 1 減算し、 $s = 1$ とする。そして、ステップ 1 2 0 4 に戻って重心を求める。

【 0 0 7 0 】

このときの重心 $p o s$ が 2.2 になったとすると、ステップ 1 2 0 6 で $n + 2$ ラインの重心を求めると、

$$P n + 2 = 2.2 - 0 + 1 = 3.2$$

となる。一方、位置誤差は、

$$M n + 2 = (3 - 3.2) \times 1 + 7 + 1 = 7.8$$

となる。ステップ 1 2 0 8 で $s = 2$ となり、ウインドウを y 方向に 1 移動する。以下、同様に測定していくと、図 1 7 の図表のようになる。これらのデータから分かるように重心 P は 4 5 度の斜線 L からの距離を表すことになる。

【 0 0 7 1 】

なお、この直線 L は、位置誤差 M が無い場合にウインドウの左端が移動する軌跡を示している。すなわち、位置誤差が無いときに 4 5 度の斜線を読み込んだときのビットマップ上の直線が L である。そして、ステップ 1 2 0 6、1 2 0 8 (後述のステップ 1 2 1 3, 1

10

20

30

40

50

2 1 4) で使用されているシフト量 s は、直線 L からウインドウの左端までの距離を表している。

【0072】

一方、変倍率 200% の場合には、図 14 に示すように、 n ラインのウインドウの重心 $p o s = 2.8$ 、重心のずれの補正值 $m o m = 0$ 、シフト量 $s = 0$ とすると、 n ラインの重心の値は、

$$2.8 (= 2.8 + 0 + 0)$$

となる〔ステップ 1206〕。位置補正の演算において、 $n - 1$ ラインの重心を $P_{n-1} = 2.8$ 、位置誤差を $M_{n-1} = 5$ とすると、 n ラインの重心は、

$$M_n = (2.8 - 2.8) \times 2 + 5 + 1 = 6$$

10

となる。

【0073】

ステップ 1208 では、シフト量に 8 加算される。ここでは、(2) 式から変倍率 200% のため $t = 0.5$ となる。したがって、 $s = 0.5$ となる。また、 y に 1 を加算する。すなわち、ウインドウの y 方向に 1 ライン移動してステップ 1204 に戻ってウインドウの重心 $p o s$ を求める。ステップ 1205 でウインドウ内の中心に斜線画像があるかどうか判断する。ここで $p o s = 2.3$ であったとすると、

$$|2.3 - 6/2| = 0.7$$

となり、ステップ 1206 に進んで $n + 1$ ラインの重心を求める。すると、重心は、

$$P_{n+1} = 2.3 - 0 + 0.5 = 2.8$$

20

となり、位置誤差は、

$$M_{n+1} = (2.8 - 2.8) \times 2 + 6 + 1 = 7$$

となる。すると、ステップ 1208 で

$$s = 0.5 + 0.5 = 1$$

となってウインドウを y 方向に 1 移動する。

【0074】

そして、ステップ 1204 でウインドウの重心 $p o s$ を求める。ここで、 $p o s = 1.5$ であったとすると、

$$|1.5 - 6/2| = 1.5$$

となり、ステップ 1214 に進む。そして、ステップ 1214 でウインドウを x 方向に -1 移動し、 s を 1 減算して $s = 0$ とする。その後、ステップ 1204 に戻って重心を求める。このときの重心 $p o s$ が 2.5 になったとすると、ステップ 1206 に進み、 $n + 2$ ラインの重心を求める。重心は、

$$P_{n+2} = 2.5 - 0 + 0 = 2.5$$

30

となり、位置誤差は、

$$M_{n+1} = (2.8 - 2.5) \times 2 + 7 + 1 = 8.6$$

となる。ステップ 1208 で $s = 0.5$ となり、ウインドウを y 方向に 1 移動する。以下、同様に測定していくと、図 18 に示す図表のようになる。

【0075】

変倍率 50% の場合には、図 15 に示すようになる。すなわち、 n ラインのウインドウの重心 $p o s = 2.4$ 、重心のズレの補正量 $m o m = 0$ 、シフト量 $s = 0$ とすると、 n ラインの重心の値は、

$$2.4 (= 2.4 + 0 + 0)$$

となる〔ステップ 1206〕。次に、位置誤差の演算において、 $n - 1$ ラインの重心を $P_{n-1} = 2.4$ 、位置誤差を $M_{n-1} = 5$ とすると、 n ラインの重心は、

$$M_n = (2.4 - 2.4) \times 5 + 1 = 6$$

となる。ステップ 1208 では、シフト量 s に t 加算される。ここでは、(2) 式によって変倍率 50% のため、 $t = 2$ となる。したがって、シフト量 s は、 $s = 2$ となる。また、 y を 1 加算する。つまり、ウインドウの y 方向に 1 ライン移動してステップ 1204 に戻り、ウインドウの重心 $p o s$ を求める。

50

【 0 0 7 6 】

ステップ 1 2 0 5 では、ウインドウ内の中心に斜線画像があるかどうか判断する。ここで重心 $p o s = 0 . 4$ であったとすると、

$$| 0 . 4 - 6 / 2 | = 2 . 6$$

となり、ステップ 1 2 1 4 に進む。ウインドウを x 方向に - 1 移動し、シフト量 s を 1 減算し、 $s = 1$ とする。また、ステップ 1 2 0 4 に進み、ウインドウの重心 $p o s$ を求める。このとき $p o s = 1 . 4$ であったとすると、

$$| 1 . 5 - 6 / 2 | = 1 . 6$$

となり、ステップ 1 2 1 4 に進む。ウインドウを x 方向に - 1 移動し、シフト量 s を 1 減算し、 $s = 0$ とする。また、ステップ 1 2 0 4 に進み、ウインドウの重心 $p o s$ を求める

10

【 0 0 7 7 】

このとき $p o s = 2 . 4$ となり、ステップ 1 2 0 6 に進む。このときウインドウの重心とウインドウの中心との差 $| p o s - X_W / 2 |$ の大きさによってステップ 1 2 1 3、ステップ 1 2 1 4 の x 、 s に対する加減算量を制御すれば、ステップ 1 2 0 4、1 2 0 5、1 2 1 2、1 2 1 3 および 1 2 1 4 のサイクルを減らすことができ、演算時間を短くすることができる。例えば、

$$| p o s - X_W / 2 | = 2$$

のとき、ステップ 1 2 1 3、1 2 1 4 の x 、 s に対する加減算量を 2 とすれば、これまでの説明のものでは、ステップ 1 2 0 4、1 2 0 5、1 2 1 2、1 2 1 3 および 1 2 1 4 の

20

【 0 0 7 8 】

次いで、 $n + 1$ ラインの重心を求めると、

$$P_{n+1} = 2 . 4 - 0 + 0 = 2 . 4$$

となり、位置誤差は、

$$M_{n+2} = (2 . 4 - 3) \times 0 . 5 + 7 + 1 = 7 . 7$$

となる。これによりステップ 1 2 0 8 ではシフト量 $s = 2$ となり、ウインドウを y 方向に 1 移動する。

【 0 0 7 9 】

そして、ステップ 1 2 0 4 でウインドウの重心 $p o s$ を求める。ここで、前記と同様なサイクルを経過し、重心 ($p o s$) が 3 になったとすると、ステップ 1 2 0 6 に進んで、 $n + 2$ ラインの重心を求める。すると、重心は、

30

$$P_{n+2} = 3 - 0 + 0 = 3$$

となり、位置誤差は、

$$M_{n+2} = (2 . 4 - 3) \times 0 . 5 + 7 + 1 = 7 . 7$$

となる。これにより、ステップ 1 2 0 8 でシフト量 s は、 $s = 2$ となり、ウインドウを y 方向に 1 移動する。以下、同様に測定していくと、図 1 9 に示す図表のようになる。

【 0 0 8 0 】

このような制御によって、変倍に対応した位置誤差測定を行うことができる。

【 0 0 8 1 】

40

2 . 第 2 の実施形態

引き続き、第 2 の実施形態について説明する。以下の各実施形態において、前述の第 1 の実施形態と同等と見なせる各部には同一の参照符号を付し、重複する説明は適宜省略する。

【 0 0 8 2 】

2 . 1 . 測定原理

図 2 0 は、第 2 の実施形態の測定原理を説明するためのビットマップを示す図であるが。この実施形態では、第 1 の実施形態と主走査方向が逆方向に設定されていることを除けば図 1 と同等であるので、ここでの詳細な説明は省略する。

【 0 0 8 3 】

50

2.2. システム構成

図21は、第2の実施形態に係る位置誤差測定装置のシステム構成の一例を示すブロック図で、画像読取装置へ付加機能として組み込み、リアルタイムでその位置誤差を測定するものである。このシステムは光電変換部1、A/D変換部2、シェーディング補正部3、位置誤差測定部5、制御部6、画像変倍部9および操作部8から基本的に構成されている。なお、図21(a)は位置誤差測定を行った後に原稿の画像データを変倍する例で、図21(b)は斜線の画像データを変倍(等倍)し、その後、位置誤差測定を行う例である。

【0084】

図21(a)において、光電変換装置1は、例えばラインCCDで、画像が電気信号に変換される。電気信号に変換された画像はA/D変換部(器)2でデジタルの多値の画像データに変換される。変換されたデータは、照明の不均一さ、レンズの周辺光量の低下、光電変換装置の画素間の感度の違いなどをシェーディング補正部3によってシェーディング補正する。シェーディング補正部3でシェーディング補正が行われた画像データは、位置誤差測定部5で副走査方向の読取ラインごとにライン間の位置誤差が測定され、測定結果を画像変倍部9に出力する。画像変倍部9では、原稿の画像データ(ビデオ信号)とともに位置誤差信号のデータ10を受け取り、原稿の画像データを操作部8から入力された変倍率にしたがって変倍し、原稿の画像データ(ビデオ信号)11として出力する。それぞれの機能ブロックは、制御部6によってタイミングの制御、動作条件の設定などがなされ、相互に関連して動作する。符号12は制御部6に対して送受信されるビデオ制御信号である。

【0085】

一方、図21(b)においては、シェーディング補正された斜線の画像データは、まず画像変倍部9で変倍処理され、位置誤差測定部5で変倍された斜線の画像データに基づいて誤差測定処理を実行する。

【0086】

光電変換装置として等倍センサを用いている読取装置ではレンズの特性による周辺光量の低下という問題がないので、シェーディング補正を省く場合があるが、そのような形式の読取装置にも、本願を適用することができる。

【0087】

2.3. 位置誤差測定処理

図22は図20と同様のビットマップに斜線の画像データaがあるときの位置誤差の測定を行うときの処理を説明するためのものである。W1は画像データの位置を求めるための演算を行う 11×3 のウィンドウである。ウィンドウ内のデータの位置を求めるため、主走査方向における重心を演算する。この演算では、順次ウィンドウの位置をW2, W3...と移動させながら重心を求める。重心の主走査方向の位置は45°の線の場合、画素の位置がなんらかの誤差要因で移動することがなければ、ウィンドウを図のように移動させた場合、主走査方向に1画素分ずつ移動するはずである。画素の移動量が1画素分と異なる場合は、何らかの原因で画素の位置が変動したことになり、位置誤差を求めることができる。位置誤差の主要な要因が副走査方向の走査速度のムラによることが分かっている場合には、位置誤差のデータが速度ムラにデータを変換することは容易である。

【0088】

重心を求めるのに周辺の画素のデータを含む多数の画素データを使っているので、CCD固有のノイズを始めとしてさまざまなノイズが画像データに含まれるが、重心を求める過程でノイズの影響が軽減され、S/Nの高い測定が可能になっている。通常、ウィンドウの画素の数が多ほどS/Nは高くなる。

【0089】

ウィンドウの形状は主走査方向の重心を求めることから、主走査側に大きいことが望ましい。副走査方向は1としても測定可能である。

【0090】

10

20

30

40

50

図 2 3 は斜線の数が増え、複数の斜線 a_1 , a_2 , a_3 を使用して位置誤差を測定する場合のウィンドウの移動とそれに伴う処理を説明するものである。図 3 の例と同様にウィンドウを順次移動させ、あらかじめ設定しておいた W_n に達したとき、その次のウィンドウとして W_{n+1} に移動させる。移動する前後の斜線のパターン a_1 と a_2 の間隔は測定用チャートを作成する段階で決めておき、その間隔の値を主走査方向の重心の移動を計算するときに補正し、 W_{n+1} 、 W_{n+2} 、 W_{n+3} … と移動させる。パターン間の間隔を画素サイズの整数倍に設定しておく、ウィンドウをジャンプさせたときの補正が簡単であり、測定に先立って測定装置にこの補正量を入力するときにも便利である。

【 0 0 9 1 】

10

この例ではウィンドウを 1 画素ずつ移動させているが、画素の位置誤差を起こす原因となる振動などの周波数帯域が低い場合は、ウィンドウを 2 画素以上ずつ移動させても良い。こうすることによって測定に要する時間短くすることができる。

【 0 0 9 2 】

また、複数の斜線を使って位置誤差を測定するようにすれば、読み取り装置の読み取り範囲が縦長であっても副走査方向の全域にわたっての測定が可能になる。さらに、主走査方向の狭い幅のなかだけで測定するようにすれば、主走査方向における中央部とか、手前とか、奥側とかに分けて位置誤差を測定することも可能になる。

【 0 0 9 3 】

これらの図からも明らかなように、高い分解能で位置誤差を測定する場合でも、それに
20
応じて斜線のパターンを細くする必要は全くなく、システムの M T F の制約の影響を受けない幅の広いパターンを使うことができるという特徴がある。幅の広いパターンを使えば、それに応じてウィンドウも大きくなり、結果として測定の精度を上げることができる。なお、処理速度、リアルタイム処理を行う場合は、バッファのサイズ、回路規模の経済性などとのバランスでパターンの幅を設定すればよい。

【 0 0 9 4 】

なお、他の例として、幅の広い線のパターンを用い、どちらか片側のエッジのデータによっても同様に位置誤差を測定することが可能である。

【 0 0 9 5 】

また、副走査の読み取りタイミングと斜線との関係は常に同じであるから、前述の公知例
30
のように副走査方向に並べられた等間隔の白黒のパターンでは避けることのできないモアレの問題を回避することができ、高精度な位置誤差の測定を可能にしている。

【 0 0 9 6 】

2 . 4 . ウィンドウのデータと重心の計算

図 2 4 はウィンドウのデータと、斜線のパターンの関係を示すものである。ウィンドウの各画素には斜線のパターンを読み取って得られる画像データの値が記入されている。画像データの値は 8 ビットのデジタルデータで、10 進法で表すと 0 ~ 2 5 5 の値を取ることができる。図の値は画像のデータを 10 進法で表記した値である。

【 0 0 9 7 】

主走査方向の重心を計算するには、各列ごとにデータの和を求める。これを右側から h_0 40
、 h_1 、 … h_{10} とすると、それぞれ 1 4 、 3 7 、 1 5 0 、 3 4 5 、 5 6 2 、 5 9 0 、 4 2 7 、 2 0 2 、 5 0 、 1 8 、 1 3 である。各画素の主走査方向の中心の座標を右から順に 0 ~ 1 0 とし、重心の主走査方向の位置を m とすると、 m の周りのモーメントは 0 となるので、

$$h_0 (m - 0) + h_1 (m - 1) + \dots + h_{10} (m - 10) = 0$$

が成り立ち、数値を入れて計算すると、

$$m = 4 . 6 6 7$$

が得られる。

【 0 0 9 8 】

重心を求めるのは、補間などの前処理を必要とせず、演算の簡素化、高速化に有用である 50

。画像の位置を求めるのは、各列ごとのデータの和の並びから、補間により所定の分解能のデータ列を得て、そのデータからピーク値の存在する位置を求める方法を使うこともできる。

【0099】

2.5. 斜線の幅

重心を計算するに当たり、斜線の幅はデータをきちんと読み取れるものであれば問題ないが、画素が正方形で、斜線の角度が45°であり、画像の走査速度を所定の目標速度からのわずかなズレをより高精度で測定する場合、斜線の主走査方向の幅を画素の整数倍にしておくと、ウインドウを斜め方向に移動しても、斜線と画素の関係は斜線の両側で同じになり、画像データの誤差要因もバランスし、画像の位置を計算する精度を高めることができる。

10

【0100】

2.6. 主走査方向の斜線の画像の移動量と副走査方向の画素の位置誤差の関係

この実施形態では、副走査方向の画素の位置誤差を測定するために、斜線を読み取った画像の主走査方向へ画像の位置の移動を見ている。正方形の画素で45°の斜線を使って測定する場合には、これまでの説明で明らかなように、主走査方向の移動量のウインドウ間における偏差がそのまま、副走査方向の位置誤差になる。画素が正方形でない場合、斜線の角度が45°でない場合には、換算をして副走査方向の位置誤差を得る必要がある。

【0101】

2.7. 測定の処理手順

20

図25は、測定の処理手順を示すフローチャートである。この処理手順では、まず、計算するウインドウの位置を示すW・P・(ウインドウポインタ)をセットし〔ステップ2501〕、次に、W・P・で指示されるウインドウのデータを取り込み〔ステップ2502〕、取り込んだデータの総和Vを計算する〔ステップ2503〕。そして、データの総和Vがあらかじめ設定したaとbとの間の値を持っているかどうかをチェックする〔ステップ2504〕。このチェックでaとbとの間に入っていれば、重心の計算を行い〔ステップ2505〕、さらに、重心のずれを計算した〔ステップ2506〕後、次のW・P・をセットする〔ステップ2507〕。その後、ステップ2502に戻ってデータフェッチ以降の処理を繰り返す。

【0102】

30

一方、ステップ2504で、データの総和Vがaとbとの間に入っていなければ、ループから抜け出し、処理を終了する。

【0103】

なお、ステップ2504で処理の総和をチェックするのは、スタートのときにW・P・を誤ってセットしたため、ウインドウ内に斜線のデータがないような場合に、正しい測定がされていないのに測定結果が出力されるのを防止するという理由からである。また、測定に使う斜線の長さを短くしておけば、斜線が途切れた位置で打ち切ることができ、必要以上の測定を無駄を省くことができる。

【0104】

2.8. 装置の概略構成

40

図26は、この実施形態に係る画像読取装置の概略構成を説明するための断面図である。同図において、筐体28の上面に、読み取るための原稿を載せるコンタクトガラス21が設けられ、当該コンタクトガラス21は筐体28に支えられた状態になっている。コンタクトガラス21の上面に画像を下にして置かれた原稿は、照明光源22によって照明され、原稿の反射光は第1ミラー23、第2ミラー24、第3ミラー25および結像レンズ26によって光電変換装置27上の光電変換素子の受光面に投影され、原稿の画像は電気信号に変換される。電気信号に変換されたデータは所定の処理をした後、出力される。

【0105】

照明光源22と第1ミラー23は、図示しない第1キャリッジに取り付けられており、同じく図示しない駆動装置によって原稿を線順次に読み取るため、原稿面との距離を一定に

50

保った状態で移動する。第2ミラー24と第3ミラー25は、図示しない第2キャリッジに取り付けられ、第1キャリッジの1/2の速度で第1キャリッジと同様に移動する。このような構成で原稿を走査することによってコンタクトガラス21上の所定の範囲の画像を線順次で読み取る。

【0106】

図27は、図26に示した画像読取装置の平面図で、コンタクトガラス21、筐体28、シェーディング補正の基準データを光電変換部に与えるための基準濃度板29、および読み取った画像データの画素の位置誤差を測定するために設けられた測定用パターン30の配置の状態を示している。ここで、基準濃度板29および測定用パターン30が鎖線で示してあるのは、光電変換装置で読み取れるように読取装置の外面には出ていないことを示すためである。特に測定用パターン30は画像データとともに光電変換装置で読み込むので原稿と同様に光電変換素子の受光面に結像する必要があり、コンタクトガラス21の原稿が置かれる面に設けられている。

10

【0107】

図28は図27において2点鎖線の円C1で囲んだ部分の詳細を示す図である。基準濃度板29は測定用パターン30を読み取る光電変換素子の画素に対してもシェーディング補正が行えるようにするため、測定用パターン30が配置されている領域まで延ばしてある。

【0108】

図29は測定用のパターン30の一部を拡大した平面図であり、このパターン30は黒の斜線Lと背景の白で構成している。このパターン30については、1.1および2.3で説明したものである。なお、ここでは、測定用のパターン30を画像外の図27に示す位置に設置し、画像と同時に読み取って測定する。

20

【0109】

2.9. 位置誤差に基づく読取データの配置
図30は位置誤差の補正をどのように行っているかを説明するための図である。縦軸は図21におけるシェーディング補正部3におけるシェーディング補正後の画像データの値を示し、データは8ビットのデジタル値であるから、10進法で表記すれば0~255の値を持つ座標軸である。横軸はライン順次で読み取るラインの位置を示す。正の正数を割り当てた位置は、前述のシステムの制御部5が水晶振動子による発振周波数を分周して作ったライン読み込みのタイミング信号に対応する各ラインの位置である。水晶振動子の発振周波数の安定度は非常に高いので、整数を割り当てた位置はシステムの画像ラインの本来あるべき位置を示していることになる。この間隔はまた、本システムの読み取りの分解能、400dpiのドット間の距離にも対応する。

30

【0110】

装置に備えなければならないメモリの量を最小で済ますため、この例では画素の位置の補正をリアルタイムで行う。リアルタイムで行うためには処理に伴う演算を簡略化する必要があり、簡略にすれば処理系の回路規模も小さくなり、経済的な効果もあるので、処理の分解能を1/16ドットとしている。そのため、横軸の整数の間は16に分割したメモリが設けられている。

40

【0111】

横軸の0においては、システムの制御部5が決める位置と実際に読み取った画像の位置は一致しているものとして図示している。画素の位置誤差が生じる原因はいろいろあるが、中でも大きな原因となるのはキャリッジの速度が変動することである。図では、第1キャリッジの速度が所定の値よりも1/16、言い換えれば6%速い状態が続いた場合を示している。横軸の1の位置に本来対応する位置の画像を読み取るはずであるが、キャリッジが速いため、実際には1/16ドット分先の画像を読むことになる。この位置を細線bで示し、そのときのデータを小さな○で示している。このとき位置誤差の測定は、1ライン前の位置を基準とした次のラインの位置ずれを順次ラインごとに行うので、1/16の誤差が測定結果として画像データとともに出力され、補正部はそれを受け取る。位置誤差測

50

定における重心の演算の精度は $1/16$ よりも高いが、その結果を $1/16$ の分解能になるように丸めを行っている。

【0112】

キャリッジの速度はそのまま速い状態を継続しているので、1ライン前に読み取ったデータとの関係で測定して得られる位置誤差は同じく $1/16$ である。しかし、システムのクロックによって決まる本来画素があるべき位置2とは1ライン前のラインの位置がすでに $1/16$ ずれていたのに、さらに $1/16$ ずれることになり、結果として $2/16$ ずれた位置cの画像データを読んでいることになる。同様に、次の読み取りでは $3/16$ ずれた位置dの画像データを読み取り、順次e, f, gの位置の画像を読み取って、それぞれの小さな○で示すデータが得られる。すなわちラインごとに測定する位置誤差の累積によって読み取った画像データの位置は決まり、 $1/16$ の分解能を持つ横軸に読み取ったデータが割り付けられる。なお、ここでは、原稿を走査する速度が所定の速度よりも速い場合を示しているが、遅ければ逆の方向にずれることになる。

10

【0113】

2. 10. 読み取りデータの補正

図30のa~gに対応している小さな○で示す位置誤差を含んだ読み取りデータから本来画像があるべき位置0~7に対応する大きな○で示すデータを補間法で求めることによってデータの補正を行う。例えば、横軸の座標2に対応するデータを求めるには、2より前のデータ2個(a, bに対応する読み取りデータ)と後のデータ2個(c, dに対応する読み取りデータ)から3次補間法(Cubic Convolution)を使っている。順次補正を行う整数の座標の前のデータ2個と後のデータ2個を使って補正データを求める。補間法はこれに限らず、このほかのさまざまな補間法を使用することもできる。また、補間に使用するデータの数も必要に応じて増減することもできる。

20

【0114】

このような補間法を使用することによって画素の位置誤差が効果的に補正され、良好な画像データが得られるが、補間のための演算に要する処理量が大いという問題があった。副走査ラインの位置誤差を補正する処理は各ラインを構成する主走査方向に5000個も並んだそれぞれのデータについて補正演算することになるので、リアルタイム処理を行うためには、比較的規模が大きく、しかも、高速で演算ができる処理回路とバッファメモリを必要としていた。

30

【0115】

これに対し、この実施形態においては、副走査方向の画素の位置誤差の補正を、読み取った副走査のラインの中から、本来あるべき位置に最も近いラインのデータをそのまま本来あるべき位置のデータとして使用することによって行う。したがって、読み取ったラインから最も近いラインを選択するだけでよいので、ライン単位で処理が完了し、先願のように副走査のラインを構成する主走査の画素毎に演算する必要はなくなり、これによって補間に伴う処理は大幅に減少することになる。この処理を図30を参照して説明する。

【0116】

この処理では、0のラインのデータとしてaのラインを使用し、1のラインのデータとしてbのラインのデータを使用する。つまり、a, b, c, d, e, f...の読み取った副走査方向のラインの系列から本来あるべき位置に最も近い位置にあるラインのデータを選択して本来あるべき位置のデータとするものである。

40

【0117】

通常の読み取り装置の原稿を走査する速度の変動は、定格速度の1%内外であり、速度変動があるといっても読み取る副走査のラインの間隔はほぼ所定の分解能に近いので、一番近くにあるラインのデータをそのまま本来あるべき位置のデータとしても本来あるべき位置が、読み取ったラインのデータの丁度中央にあるという最悪条件でも画像データの位置誤差は分解能の $1/2$ にしかならない。したがって、画素の位置誤差は $1/2$ ドット以下の精度を持っていることを保障できる。画像読取装置の多くの用途では、この程度の位置精度を持っていれば十分である。従来のフラットベット型の画像読取装置では、原稿を走

50

査するキャリッジの速度変動をなくすことは非常に困難であり、比較的大きな画素の位置誤差は避けられなかったが、本願では、斜線の繰り返しのパターンを読み取った画像データから画素の位置誤差を測定して画素の位置の補正を行うことによって非常に簡単な構成で画素の位置誤差の小さい画像読み取り装置を実現することができた。

【0118】

2.11. 変倍読取時の動作

変倍を説明するときの準備として、まず、等倍読み取りについて説明する。この実施形態における読取装置は、ラインCCDを読取素子として、等倍のとき、A3サイズ of 原稿の短手を主走査方向にして400dpiの分解能で読み取る機能を有する。ラインCCDは10μmピッチで5000個の受光素子が並んでいるものである。400dpiの分解

10

能であるから、原稿面の1ドットに相当する長さは

$$25.4 / 400 = 0.0635 \text{ mm}$$

すなわち、63.5μmである。これが光学系でCCDに結像するときには、CCD1ドットの大きさ、つまり、10μm縮小されて投影される。すなわち、原稿の画像は6.35分の1に縮小されてCCDの受光面に投影されることによって当倍の画像が形成される。副走査方向はキャリッジの移動によって走査され、その走査速度は副走査のラインクロックが1クロック進む時間に63.5μm移動する速度である。この速度で原稿を走査しながらラインクロックごとにデータを読み取る。このような読み取り方法をラインクロックごとに1列ずつ実行することから線順次読み取りという。

【0119】

20

そこで、倍率を変えて読み取るときには、すなわち、変倍時には、光学系の倍率は等倍のときと同じまま、原稿を走査する速度を倍率に対応させて変更して読み取る。例えば、2倍で読み取るとき、副走査のラインクロックの速度はそのままにして原稿を走査するキャリッジの速度を等倍のときの1/2にして読み取る。キャリッジの速度の変更はキャリッジを駆動するモータの回転数を変更することによって行う。副走査のラインクロックごとに得られるデータを等倍のときと同じ63.5μm離れたデータとして取り扱うことによって副走査方向に2倍に拡大された画像データが得られる。しかし、このままでは、光学系の倍率は変更されていないので、主走査方向に関しては等倍のままである。主走査方向の2倍の画像を得るために、等倍で得た画像データから変倍したときに得られるはずの位置の画像データを補間法によって求める。補間法としては、前述の3次関数コンボリューションが使用される。ここでは、わかり易くするため、2倍の例で説明したが、他の倍率で読み取るときにも同様である。

30

【0120】

以上、変倍読み取りの概要を説明したが、以下、画素の位置誤差を得るために原稿を読み取る領域外に設けた斜線の繰り返しにより構成されたパターンの画像データと主走査方向の変倍処理を行う画像データの範囲と原点の位置関係を図31を参照して説明する。

【0121】

正方形の並びで示されている横長の矩形は、ラインセンサで読み込まれる主走査の1ラインの画像データの並びを示すとともに、ラインセンサの受光部の配置を示している。主走査と書かれた矢印は主走査の方向を示し、ラインセンサで光電変換された電気信号がCCDに内蔵されたシフトレジスタに同時に転送された後、主走査のクロックに同期して画像データが出力される順序を示している。矢印の方向は時間が経過する方向を示しており、図の右端のデータから順に出力される。図のA～Bは読み取り装置の主走査方向の原稿の読み取り領域に対応し、図27において上方の辺がCCDに投影される位置がAであり、下辺が投影される位置がBである。

40

【0122】

読み取り領域の上辺に沿って配置されている斜線の繰り返して構成された画素の位置誤差を測定するためのパターンの画像は、画像データの並びの右端からAまでの範囲に形成される。この関係を保ったままキャリッジを副走査方向に移動させることによって線順次にラインごとの読み取りを繰り返すことによって画像データをj得ている。装置が原稿の読み

50

取りを開始するときに主走査方向のA位置に画像が形成されるコンタクトガラス21の左手奥の角に原稿を突き当てて載置するように設定しており、この角の位置を原点としている。すでに説明したように光学系の倍率は変更しないので、変倍読み取りを行うときでも、主走査方向の画像の位置は変わらない。そこで、変倍読み取りを行うときでも、斜線のパターンの画像の主走査方向の位置は変わらないが、副走査方向の移動速度が変わるため、得られる斜線の画像データの傾きは、速度によって変化する。これについては後述する。変倍で読み取りを行うときにも斜線を読み取ったデータの主走査方向における範囲は変わらないので、特別の前処理を行うことなく、画素の位置誤差を求めることができる。しかし、原稿を読み取った画像の部分に対しては、主走査方向の倍率変換処理を行う必要がある。原点を原稿先端のAに対応するコンタクトガラス21の右奥としているので、読み取ったAB間のデータをA点を基準とする所定倍率の画像データに変換する。変換には前述の3次関数コンボリューション法を使用する。

10

【0123】

このように変倍で読み取るときにも、画素の位置誤差を求めるために斜線のデータに対しては変倍に伴う主走査方向の処理を行わず、原稿を読み取ったデータに対してだけ変倍処理を行うようにしている。こうすることによって、変倍時にも斜線を読み取ったデータの大きさや画素の位置が変わらないので位置誤差を求める処理に際して重心を計算するウィンドウのサイズを変更したり、主走査方向の座標を変える必要がなく、等倍時と同様に画素の位置誤差を求めることができ、これによって処理系を簡素化することができる。

【0124】

20

2.12.位置誤差の測定手順および主走査方向の斜線の画像の移動量と副走査方向の画素の位置誤差の関係

位置誤差の測定手順、および主走査方向の斜線の画像の移動量と副走査方向の画素の位置誤差の関係は、第1の実施形態において図12のフローチャート、図13ないし図15のビットマップとウィンドウとの関係を示す説明図、さらには、図16の説明図、図17ないし図19の図表を参照したものと同様なので、ここでの説明は省略する。

【0125】

このような制御によって、この第2の実施形態では、変倍に対応した位置誤差測定を行うことができる。

【0126】

30

3.第3の実施形態

引き続き、第3の実施形態について説明する。この実施形態は、変倍読み取りを行うとき、光学系の結像倍率を所定の値に変更するとともに、副走査方向の走査速度も倍率に合わせて変更して読み取る形式の読み取り装置に画素の位置誤差を計測し、そのデータに基づいて画素の位置誤差を補正する機能を搭載するための改良を施したものである。

【0127】

光学系の倍率の変更は第2の実施形態の図31におけるA点を基準に、すなわち、倍率を変更しても原稿台のA点对应する位置がCCDの受光部に投影される位置が移動しないように行う。この光学系の倍率の変更は図26に示した結像レンズ26の位置を光電変換装置27に対して移動させる駆動装置(図示せず)により行われる。副走査方向に関しては第1および第2の実施形態とまったく同一である。この実施形態では、主走査、副走査とも変倍されて読み取られるので、読み取った画像データに対して追加の処理を必要としない。しかし、読み取りの倍率を変えたときに、画素の位置誤差を求めるための斜線の繰り返しパターンを読み取った画像も読み取り倍率に対応して光学系の倍率を変更しているので、主走査方向の倍率が変ってしまう。この倍率が変ってしまった斜線を読み取った画像データを等倍のときと同様に処理できるようにするため、等倍のデータに変換する。

40

【0128】

等倍に変換する際、光学的にはAを基準として倍率が変わっているので同じA点を基準に等倍に変換する。この変換は、第1の実施形態で等倍のまま読み取って変倍のデータを得ているのと倍率を変換するという観点に立てば、全く同じである。A点を基準に等倍にする

50

ので、等倍に戻した斜線の画像の位置と等倍で斜線を読み取ったときの画像の位置は同じになる。したがって、等倍のときと全く同じようにして画素の位置誤差を測定することができる。このように変倍のとき、斜線のデータを等倍に変換することによって、読み取り倍率に応じてウインドウのサイズや位置を変える必要がなくなり、画素の位置誤差の測定にかかわる回路を簡素化することができる。もちろん原稿を読み取った画像は光学的に変倍されているので、変倍に伴う画像処理の必要がないことはいうまでもない。

【0129】

なお、特に説明しない各部は第2の実施形態と同等に構成され、同様に動作するので、重複する説明は省略する。

【0130】

このように処理することによって、第2の実施形態の変倍方式をとっても、第3の実施形態の変倍方式をとっても、同じ画素の位置誤差を測定する処理系を使用することができる。

【0131】

【発明の効果】

これまでの説明で明らかなように、本発明によれば、有効画像領域外に設置され、主走査方向に対して一定の傾きの複数の線から構成された斜線パターンと、該斜線パターンを有効画像領域内に載置された原稿とともに読み取る主走査方向に延伸した読み取り手段と、原稿の変倍率を設定する設定手段と、副走査方向の原稿走査速度を前記変倍率に応じて変更する副走査手段と、前記読み取り手段によって得られた前記斜線パターンの画像データにウインドウを設定してウインドウ内の画像データの重心を計算する計算手段と、該計算手段によって得られた重心の値によりウインドウの移動を制御する制御手段と、前記重心の値と前記変倍率とに基づき、各ラインの画像の位置誤差のデータを得る演算手段とを備えているので、倍率が変化しても画像の位置誤差を正確に測定することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態における測定原理を示す説明図である。

【図2】第1の実施形態におけるシステム構成を示すブロック図である。

【図3】第1の実施形態におけるビットマップで複数の斜線を使って位置誤差を測定する場合のウインドウの移動とそれに伴う処理を示す説明図である。

【図4】斜線パターンを拡大して示す説明図である。

【図5】図4の斜線パターンの読取値を示す説明図である。

【図6】斜線判定用のウインドウを示す説明図である。

【図7】他の斜線判定用のウインドウを示す説明図である。

【図8】斜線判定用のマッチングパターンを示す説明図である。

【図9】図2の位置誤差測定部の重心の測定処理の処理手順を示すフローチャートである。

【図10】重心測定用ウインドウにおける読取値と重心の測定方法を示す説明図である。

【図11】斜線の長さ及び角度を示す説明図である。

【図12】図2の位置誤差測定部の重心の位置誤差の測定処理の処理手順を示すフローチャートである。

【図13】位置ズレを生じた斜線の変倍率100%時のビットマップパターンを示す説明図である。

【図14】位置ズレを生じた斜線の変倍率200%時のビットマップパターンを示す説明図である。

【図15】位置ズレを生じた斜線の変倍率50%時のビットマップパターンを示す説明図である。

【図16】位置誤差の演算における読取位置と位置誤差との関係を示す説明図である。

【図17】等倍時のライン数とステップ1204ないし1208の各処理によって求められる各値との関係を表形式で示す図である。

【図18】変倍率200%時のライン数とステップ1204ないし1208の各処理によ

10

20

30

40

50

って求められる各値との関係を表形式で示す図である。

【図 19】変倍率 50% 時のライン数とステップ 1204 ないし 1208 の各処理によって求められる各値との関係を表形式で示す図である。

【図 20】本発明の第 2 の実施形態における測定原理を示す説明図である。

【図 21】第 2 の実施形態におけるシステム構成を示すブロック図である。

【図 22】第 2 の実施形態におけるビットマップに斜線の画像データがあるときの位置誤差測定を行うときの処理を示す説明図である。

【図 23】第 2 の実施形態におけるビットマップで複数の斜線を使って位置誤差を測定する場合のウインドウの移動とそれに伴う処理を示す説明図である。

【図 24】第 2 の実施形態におけるウインドウのデータと斜線のパターンの関係を示す図である。 10

【図 25】第 2 の実施形態における測定の処理手順を示すフローチャートである。

【図 26】第 2 の実施形態における画像読取装置の概略構成を示す断面図である。

【図 27】第 2 の実施形態における画像読取装置の平面図である。

【図 28】図 27 において 2 点鎖線の円で囲んだ部分の拡大図である。

【図 29】図 27 における測定用パターンの部分を拡大した拡大図である。

【図 30】第 2 の実施形態において行っている位置誤差の補正の方法を示す説明図である。

。 【図 31】第 2 の実施形態におけるパターンの画像データと主走査の変倍処理を行う画像データの範囲と原点の位置関係を示す説明図である。 20

【図 32】従来例に係るリニアスケールの概略構成を示す斜視図である。

【図 33】従来例に係るリニアスケールの検出原理を示す説明図である。

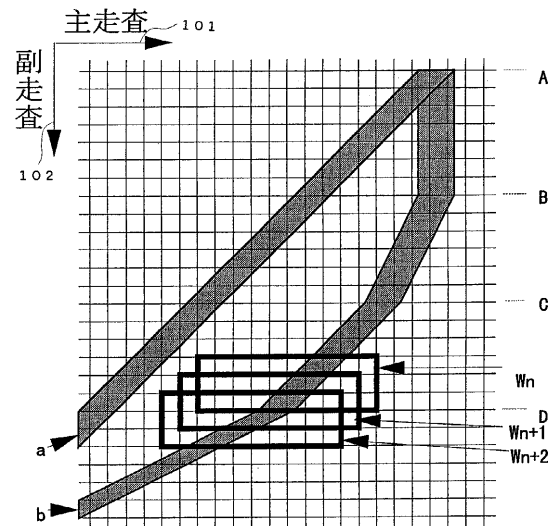
【図 34】従来例に係るリニアスケールの移動方向の判別方法を示す図である。

【符号の説明】

- 1 光電変換部
- 2 A / D 変換部
- 3 シェーディング補正部
- 4 斜線判別部
- 5 位置誤差測定部
- 6 制御部 30
- 7 画像処理部
- 8 操作部
- 9 画像変倍部
- 10 誤差信号
- 11 原稿の画像データ (ビデオ信号)
- 12 ビデオ制御信号
- a, b, L, K1, K2, K3 斜線
- W1, W2, W3, Wn, Wn+1, Wn+2, Wn+3 ウインドウ

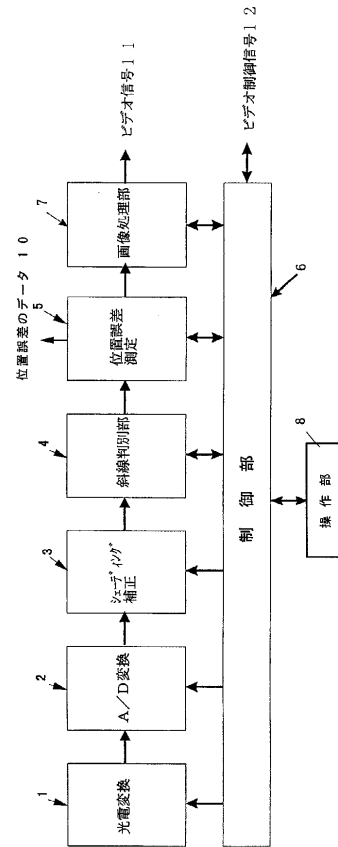
【図 1】

【図 1】



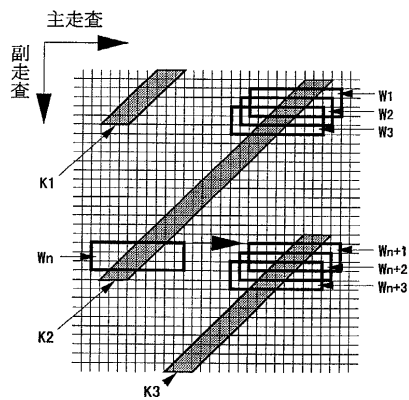
【図 2】

【図 2】



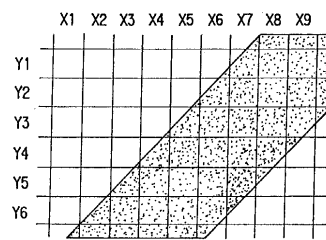
【図 3】

【図 3】



【図 4】

【図 4】



【図 5】

【図 5】

	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y1	2	3	1	4	2	18	220	250	248
Y2	2	1	2	3	13	216	248	250	252
Y3	3	2	4	8	201	250	252	252	251
Y4	2	3	4	183	251	254	252	249	32
Y5	3	5	157	250	251	252	251	50	4
Y6	2	130	249	250	251	250	62	5	3

【図 6】

【図 6】

X6	X7	X8
18	220	250
2	18	220
13	216	248
201	250	252
Y1	Y2	Y3

(a)

X3	X4	X5
1	4	2
2	3	13
4	8	201
Y1	Y2	Y3

(b)

X4	X5	X6
4	2	18
3	13	216
8	201	255
Y1	Y2	Y3

(c)

X5	X6	X7
2	18	220
13	216	248
201	250	252
Y1	Y2	Y3

(d)

X6	X7	X8
18	220	250
2	18	220
13	216	248
201	250	252
Y1	Y2	Y3

(e)

【図 7】

【図 7】

X6	X7	X8
0	1	1
0	1	1
0	1	1
0	1	1
Y1	Y2	Y3

(a)

X3	X4	X5
0	0	0
0	0	0
0	0	1
0	0	1
Y1	Y2	Y3

(b)

X4	X5	X6
0	0	0
0	0	1
0	0	1
0	1	1
Y1	Y2	Y3

(c)

X5	X6	X7
0	1	1
0	1	1
0	1	1
0	1	1
Y1	Y2	Y3

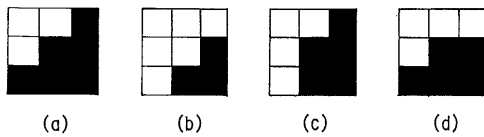
(d)

X6	X7	X8
0	1	1
0	1	1
0	1	1
0	1	1
Y1	Y2	Y3

(e)

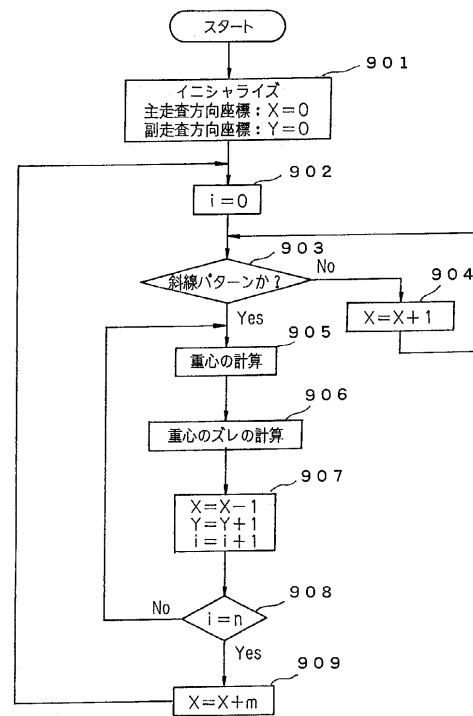
【図 8】

【図 8】



【図 9】

【図 9】



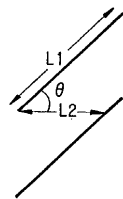
【図10】

【図10】

	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y0	4	4	10	40	150	241	202	120	22	6
Y1	4	6	42	148	240	200	122	20	10	4
Y2	10	40	150	239	200	121	21	10	5	4
	18	50	202	427	590	562	345	150	37	14

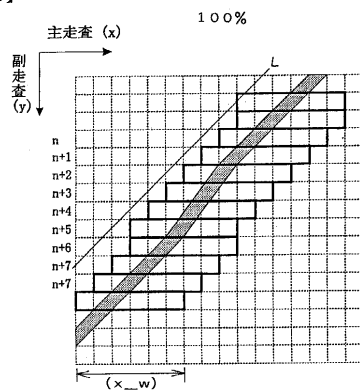
【図11】

【図11】



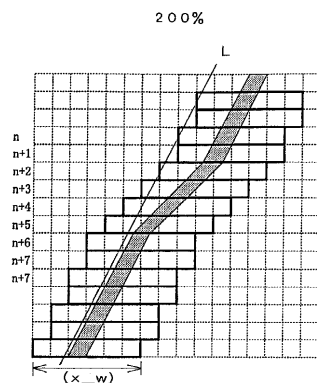
【図13】

【図13】



【図14】

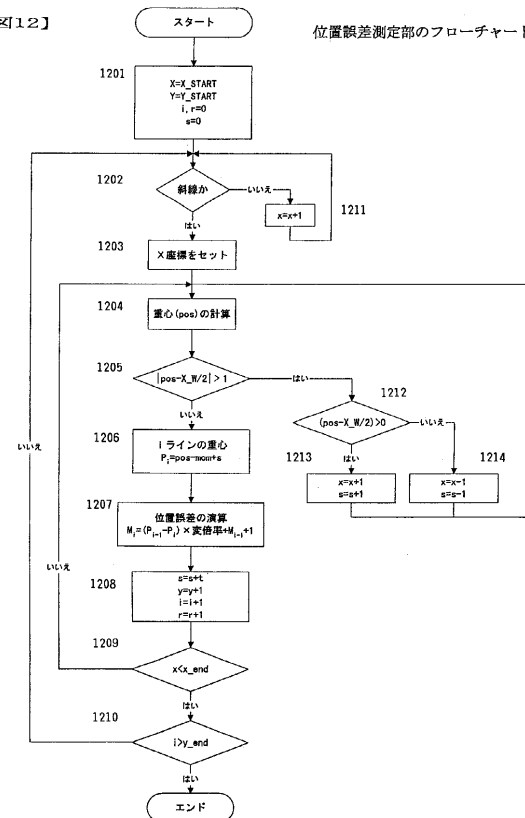
【図14】



【図12】

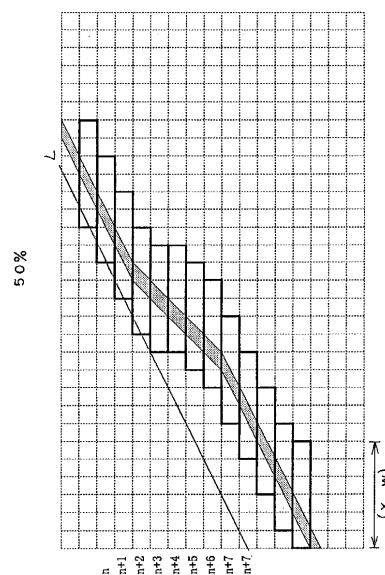
【図12】

位置誤差測定部のフローチャート

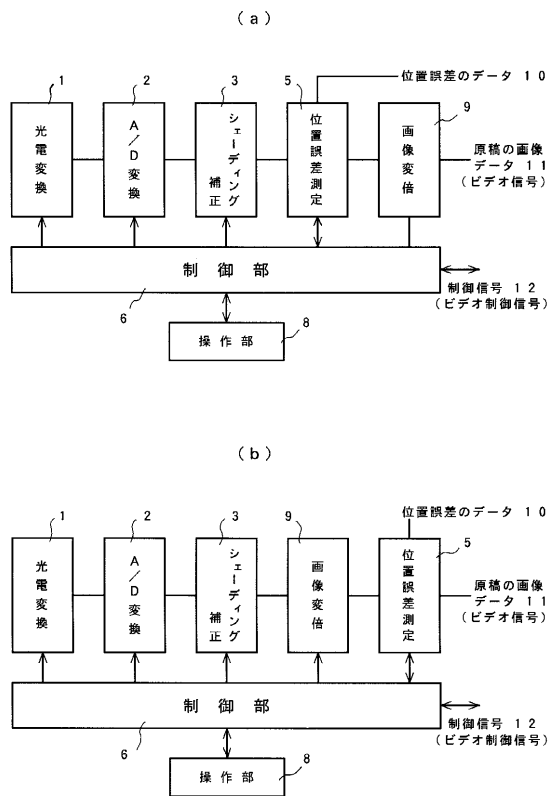


【図15】

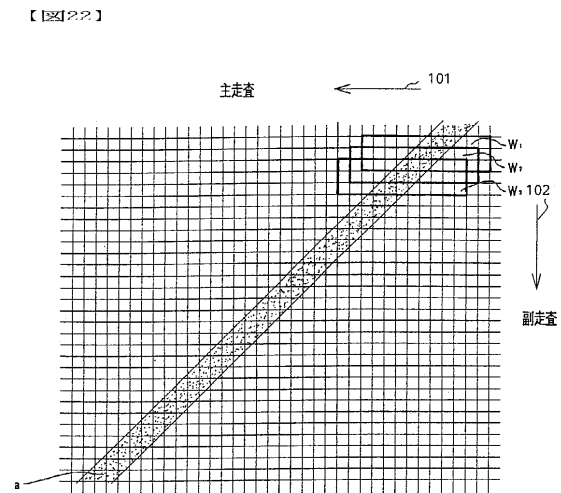
【図15】



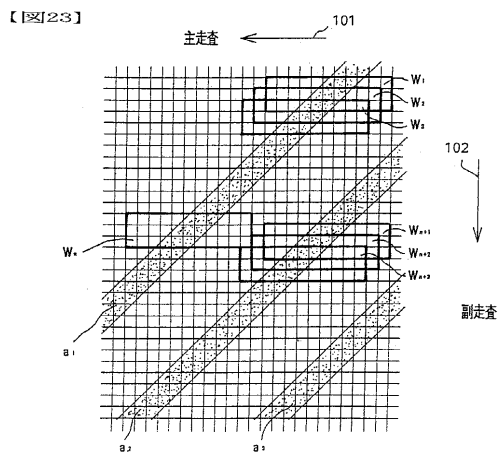
【図 2 1】
【図 2 1】



【図 2 2】

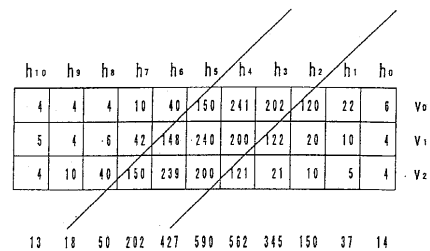


【図 2 3】



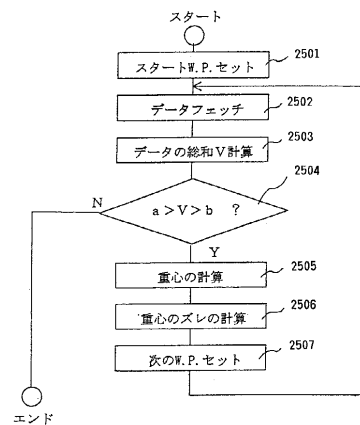
【図 2 4】

【図 24】



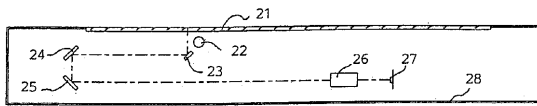
【図 2 5】

【図 25】



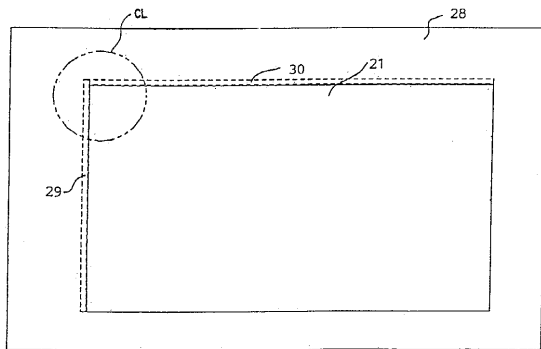
【図 26】

【図26】



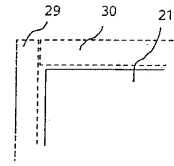
【図 27】

【図27】



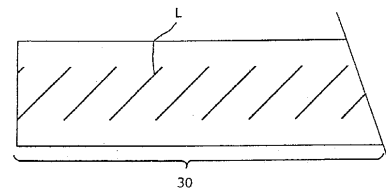
【図 28】

【図28】



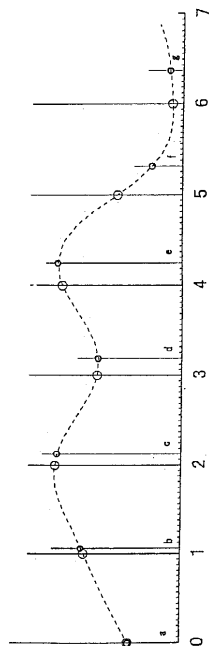
【図 29】

【図29】



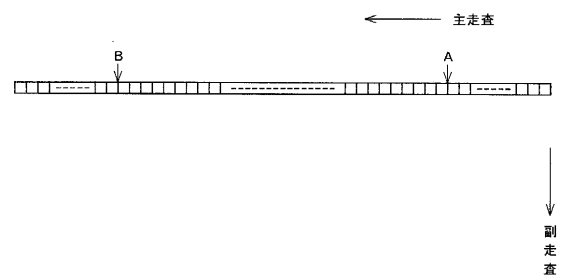
【図 30】

【図30】



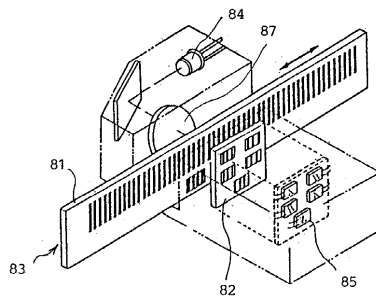
【図 31】

【図31】



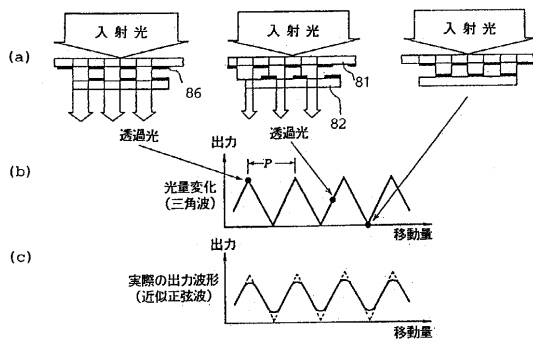
【図 3 2】

【図32】



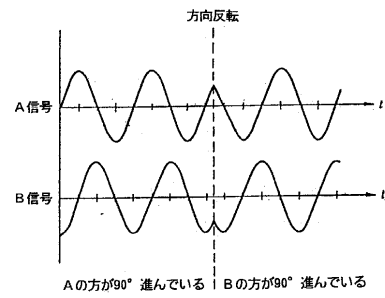
【図 3 3】

【図33】



【図 3 4】

【図34】



フロントページの続き

- (56)参考文献 特開平06-205169(JP,A)
特開平07-322016(JP,A)
特開昭60-256267(JP,A)

- (58)調査した分野(Int.Cl.⁷, DB名)
H04N 1/04 106