

Linux を用いた画像による回転角検出

Detection of Rotational Angle Using the Image on Linux

○鈴木 俊行, 三浦 武, 秋山 宜万, 谷口 敏幸

○Toshiyuki Suzuki, Takeshi Miura,

Yoshikazu Akiyama, Toshiyuki Taniguchi

秋田大学

Akita University

キーワード: 制御システム(control system), ディスクレス機(diskless computer),

リナックス(Linux), 回転角度検出(rotor angle detection)

連絡先:〒010-8502 秋田県秋田市手形学園町 1-1 秋田大学工学資源学部 電気電子工学科
三浦 武, TEL: (018)889-2329, FAX: (018)837-0406, E-mail: miura@ipc.akita-u.ac.jp

1. はじめに

CCD カメラを回転 (回転角度, 回転速度) 検出センサとして用いることによって, 比較的容易に回転検出を行うことが可能である.

CCD カメラを回転検出センサとして用いる方法と比較して, 従来から用いられているセンサである回転角度に比例したパルスを発生させる光学式ロータリーエンコーダのようなセンサを用いると, モータの回転子シャフトに直接取り付けが必要があるためモータにとって負荷となったり, 取り付けに物理的な制約を受けることがある. また光学式ロータリーエンコーダは原理上小型化が非常に困難であり高価になってしまう. 一方 CCD カメラによる画像データを用いてモータの回転検出を非接触で行うと, センサを取り付けることによってかかる負荷を小さくできる.

また, セキュリティーに優れマルチユーザ, マルチタスクといった機能を持つフリー OS, Linux の機能拡張であり, 実時間軸をもつ

RT-Linux を用いることにより, リアルタイム制御/計測系の構築が可能となる. この Linux は優れたネットワーク機能を持つ OS であり, X-Window System や NFS(Network File System)が使用できる.

X-Window System はマウスやキーボードなどの入力デバイスおよび画面描写を行う出力デバイスを管理する X サーバと, 実際の作業や計算を行う X クライアントから構成されており, この X サーバと X クライアントが同一マシンで動作している必要がないため, X サーバはネットワーク接続された別のマシンにある X クライアントをコントロールすることが可能となる. つまり, あるマシンで動作しているアプリケーションプログラムのウィンドウを別のマシンで表示することが可能となる¹⁾.

NFS は TCP/IP ネットワークにおけるネットワークファイルシステムのひとつであり, あるコンピュータ上のファイルシステムを,

別のコンピュータから利用できるようにするシステムである。この NFS を用いることによりハードディスクを持たないディスクレスコンピュータで OS、アプリケーションが使用できるようになる²⁾。

これらのネットワーク機能を有する Linux を用いて角度検出を行う遠隔制御系をイーサネット上で構築することが出来れば、制御専用のケーブルを用いるより安価に遠隔制御系の構築ができる。

上述のような事実を踏まえ、本研究ではディスクレス機で CCD カメラおよびモータの制御を行い、遠隔地にあるコンピュータ上で検出結果をモニタするシステムを構築し、パターンを使用した非接触での回転角検出結果の比較・検討を行った。

2. システム

制御システムとして PC を使用する時、同じ OS のマシンをまとめて管理・運用できれば、システム管理、システムのバージョンアップなどにかかる手間が大幅に改善され、管理コストが低くなることが期待される。

また、ハードディスクを持たない PC が他の PC にインストールされているプログラムを実行できるように設定できれば、ハードディスクそのものを使用しないためシステムの構築が安価になる。

これらの要望は、ある 1 台の PC をサーバ機として設定し、それ以外の PC をディスクレス・クライアント機と設定することによって実現される。クライアント機は起動時にサーバから IP アドレスの取得、カーネルの読み込み、サーバ機上のファイルの読み込みなどを行う。

上述の内容を踏まえ、本研究では回転角検出を行うシステムを CCD カメラとモータの制御を行うクライアント機と、クライアント機への OS のアップロード、データのグラフィック表示を行うサーバ機の 2 台の PC から構成した(図 1)。クライアント機はハードディ

スクを持たないディスクレス機となっている。この 2 台の PC は 100BASE-TX のイーサネット上で通信を行っている。表 1 に、それぞれの PC の環境を示す。

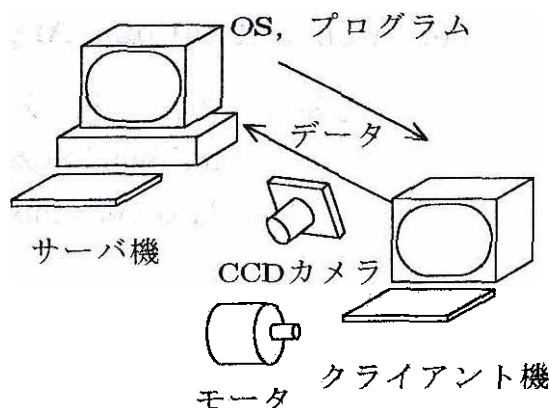


図 1 システム

サーバ機は、NFS サーバとして動作するために以下のようなアプリケーション、サービスプログラムを使用する。

- bootp(Bootstrap protocol)
- rarp(Reversed Address Resolution Protocol)

クライアントに IP アドレスを割り当てる。

- tftp(Trivial File Transfer Protocol)
クライアント機にネットワーク経由でカーネルを読み込ませる。
- NFS(Network File System)
クライアントにファイル資源を提供する。

NFS クライアント機には、特別なプログラム等は必要なくカーネルが持つネットワークファイルシステムのサポートを有効にする事で NFS が使用できる。

なお、本研究で利用したクライアント機の起動には、etherboot により作成した起動用フロッピーディスクを用いている。

表 1 2 台の PC の環境

環境	サーバ機	クライアント機
マシン名	labo2	labo1
IP アドレス	192.168.1.12	192.168.1.11
LAN カード	Logitec SAN-PCTX	同左
MAC アドレス	00:40:05:A9:CF:19	00:40:05:40:4A:40
ゲートウェイサーバ	192.168.1.1	同左
DNS サーバ	192.168.1.1	同左
Linux カーネル	rtlinux-3.1pre3	同左
ディストリビューション	Redhat6.2J	同左

3. 画像処理

本研究では、CCD カメラにより撮影された画像の認識を簡単化するために 2 値化、膨張・収縮、輪郭化等の画像処理を加えた。以下でそれぞれの手法について述べる。

1) しきい値処理

濃淡画像の各点の濃度値を、適当な基準値に基づいて 0 と 1 の 2 値に分けることによって濃淡画像を 2 値画像に変換する処理を、しきい値処理とよぶ³⁾。しきい値処理によって得られる 2 値画像は、しきい値の決定法によって左右される。多くの場合経験的に求められるが、以下にいくつかの代表的な方法を挙げる。

- a) p-タイル法: 画像中の対象物の占める面積比率が前もってわかっているとき、濃度値のヒストグラムの p パーセント点をしきい値とする。
- b) モード法: 濃度値の確率密度関数（または、確立関数）の谷（極小点）に対応する濃度値をしきい値とする。実際のヒストグラムでは、細かな凹凸によって多数の極大、極小点が存在するため、適当なしきい値を定めるために補助的な規則を用いる必要がある。
- c) 判別分析法: 入力画像の全体の画像の、

濃度値の平均値、分散を、それぞれ μ_F , σ_F^2 , 2 値化のためのしきい値を T とし、入力画像の中で T 以下の濃度値の画素全体にわたる平均値、分散を μ_1 , σ_1^2 , T より大きい濃度値の画素に対する平均値、分散を μ_2 , σ_2^2 , とすると次の関係が成り立つ。

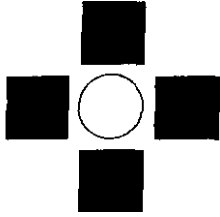
$$\begin{aligned}\sigma_F^2 &= \sigma_B^2 + \sigma_W^2 \\ \sigma_B^2 &= [n_1(\mu_1 + \mu_F)^2 + n_2(\mu_2 + \mu_F)^2] / N \\ \sigma_W^2 &= (n_1 \sigma_1^2 + n_2 \sigma_2^2) / N \quad (N = n_1 + n_2)\end{aligned}\quad (1)$$

σ_B^2 / σ_W^2 を最大にする T を求めてその値をしきい値とする。

本研究で使用するパターンを撮影したとき、前もって対象の占める面積比率を知ることが難しく、また撮影された画像の濃度ヒストグラムに多数の極小値が存在した。このため、2 値化の際のしきい値決定法として判別分析法を用いた。

2) 膨張・収縮

図形の膨張（収縮）処理は、図形を 1 画素分太める（細める）2 値画像特有の処理である。いろいろな図形処理の基礎となる重要な処理であり、広く使われている画像処理の手法である⁴⁾。図 2 に画素の連結性の定義を示す。この連結性を用いて膨張・収縮処理が行われる。



(a) 4近傍

(b) 8近傍

図2 画素の連結性

入力画像を f ，出力画像を g とすると，膨張，収縮は式 (2) のように定義できる。

膨張：

$$f(i,j) \text{ あるいはその4近傍 (8近傍) のいずれかが1} \\ \text{その他の場合}$$

$$f(i,j) \text{, あるいはその4近傍 (8近傍) のいずれかが0} \\ \text{その他の場合}$$

(2)

膨張と収縮を組み合わせることによって，図形の細かい凹凸や溝の検出，あるいは除去が可能となりより単純な図形に変換することが可能となる。図3に8近傍による膨張と収縮の組み合わせの例を示す。■は1，●は0を表す。

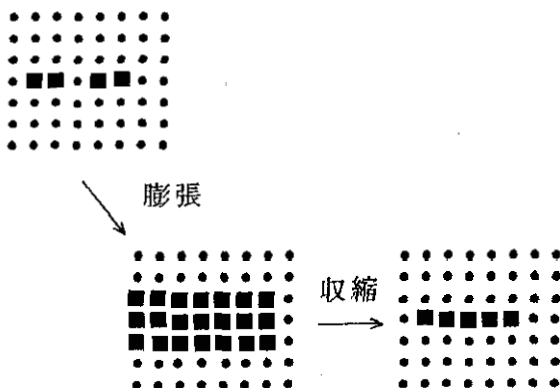


図3 膨張と収縮

3) 輪郭抽出

図形の形状特徴を抽出する上で，輪郭は重要な特徴を担っている。この輪郭を2値画像から抽出する。輪郭化された画像は単純な形

であり，図形の形状判断が容易である。

輪郭抽出は8近傍のすべてが1の場合のみ中心点を0とし，その他の場合すべてを1とすることによりおこなわれる (図4)。式 (2) と同様に輪郭抽出は式 (3) のように定義できる。

$$g(i,j) = \begin{cases} 0 & f(i,j) \text{ の8近傍のすべてが1} \\ 1 & \text{その他の場合} \end{cases} \quad (3)$$

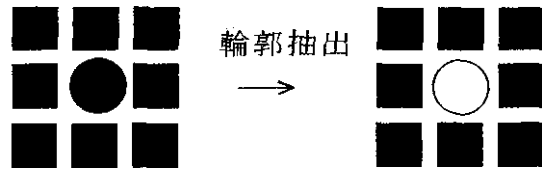


図4 輪郭抽出

4. 回転角検出方法

回転角検出は図5のパターンをモータシャフトに取り付け，このパターンがモータの回転に伴って円周方向に移動する様子を CCD カメラで撮影して行った。また，取り付けられたパターンの認識を簡易化するために，撮影された画像に2値化，膨張・収縮，輪郭化処理を加えた。2値化の際のしきい値には判別分析法を用いている。回転角検出にはモータ始動前の画像(初期画像)と，回転角検出時の画像を用いる。初期値画像から円運動の半径 r を求め，回転角検出時画像から長さ x を求めて(4)式から回転角度を求める(図6)。

$$\theta = \arccos\left(\frac{x}{r}\right) \times \frac{180}{\pi} \quad (4)$$

図7にフローチャートを示す。

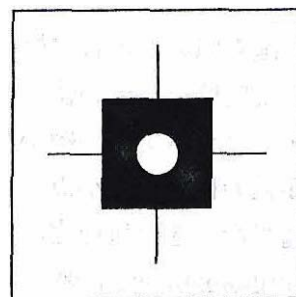
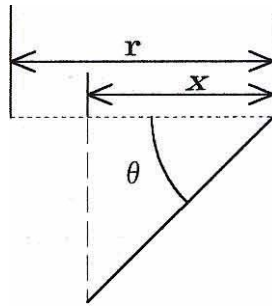


図5 90度用パターン



- ・ 初期画像のパターン位置
- ・ 角度検出時のパターン位置

図 6 角度検出方法

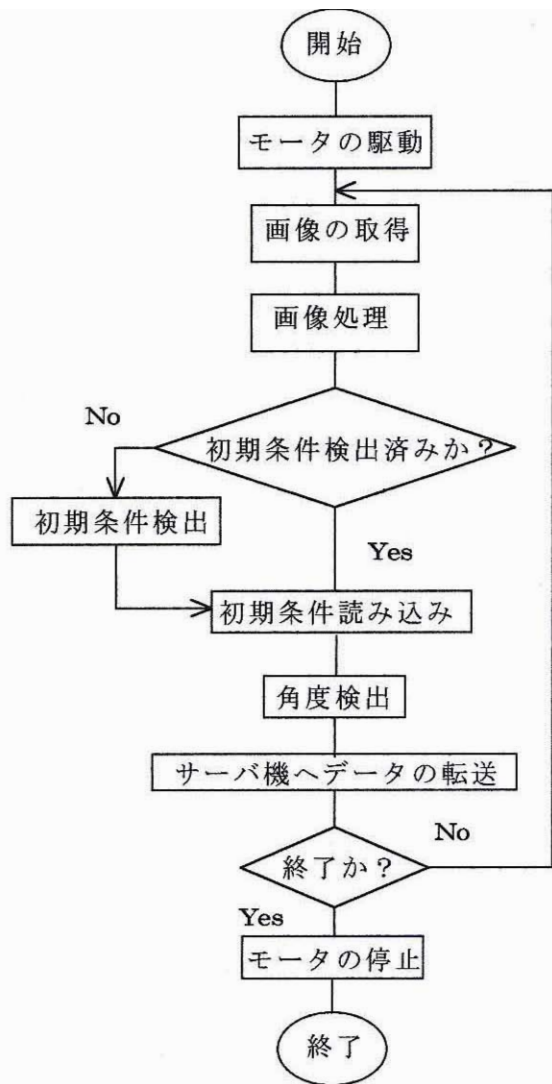


図 7 角度検出のフローチャート

上述の方法は 1 フレームで 90 度の範囲を撮影すれば良いため、パターン全体を撮影す

るよりも角度検出に使用できる画素数が増えるという利点がある。しかし、この検出方法では $\cos$ 関数を使用しているため 0 度付近で角度分解能が低下する欠点がある。そこで、今回の実験では角度検出を 30 度から 60 度の範囲で行う図 8 のパターンを用いた。この時の初期画像と検出時画像を図 9、10 に示す。

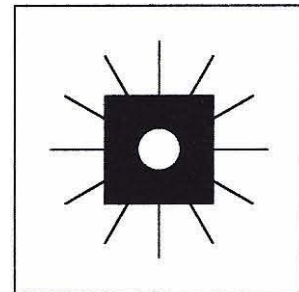
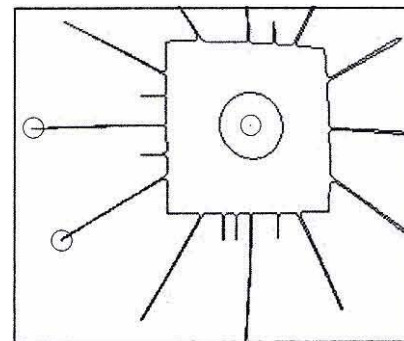
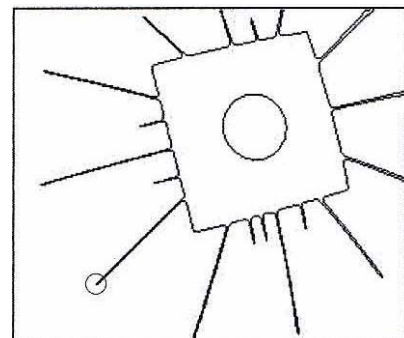


図 8 30～60 度用パターン



○ 検出点

図 9 初期画像



○ 検出点

図 10 角度検出時の画像

5. 実験装置

実験装置の構成を図 11 に示す。今回の実験では回転角検出対象として、オリエンタルモーター社製の 2 相 HB 形ステッピングモータ PK244-02B を用いた。モータの各パラメータは、定格電圧 6.0V、定格電流 0.8A、基本ステップ角 1.8 度である。ビデオキャプチャボードは I-O DATA 社製の GV-VCP/PCI(チップ BT848)を用いた。表 2 にビデオキャプチャボードの仕様を、表 3 に実験で用いた CCD カメラの仕様を示す。

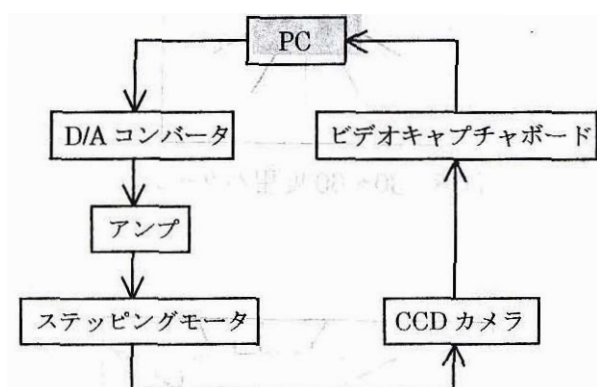


図 11 実験装置

表 2 GV-VCP/PCI の仕様

信号処理方式		YUV4 : 2 : 2
割り込み(IRQ)		PCI システムによる 自動設定
メモリマッピング		PCI システムによる 自動設定
NTSC 入力仕様		NTSC コンポジット 1Vp-p 75Ω S ビデオ 1Vp-p 75Ω
コネクタ		ビデオ入力 RCA ピン×2, S ビデオ×1
動作環境	電源	+5V ±5%
	温度	0℃~40℃
	湿度	20%~80%
	消費電流	0.25A (最大)

表 3 CCD カメラの仕様

信号方式	NTSC
撮像素子	1/3 インチ CCD
有効画素数	512×492
水平周波数	15.732 [kHz]
水平解像度	380 本
垂直解像度	380 本
最低被写体照度	0.2 [Lux]
S/N 比	46 [dB]以上(AGC-OFF)
映像出力	1.0[V _{p-p}], 75[Ω]
電源電圧	DC12 [V]±1[v]

6. 実験結果

実験は、円運動半径 $r=20[\text{mm}]$ の 30~60 度用パターン(図 8)を用い、カメラ、パターン間の距離 30[mm]の状態で行った。検出対象のステッピングモータは基本ステップ角を 10 分割し、0.18[deg]のステップ間隔でマイクロステップ駆動させ、サンプリング間隔 500[ms]で 0.0[deg]から 90.0[deg]までの間、回転角度検出を行った。この時の初期画像を図 12 に、検出結果を図 13 に、ステップ間隔×ステップ数によって算出した目標値との誤差を図 14 に示す。また、サーバ機に表示させたグラフを図 14 に示す。

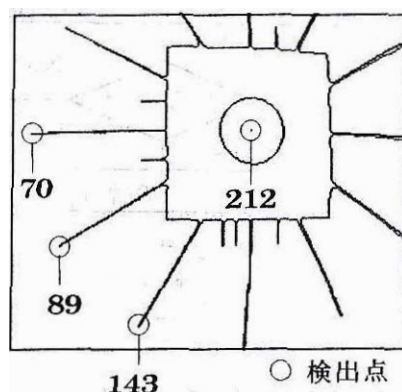


図 12 カメラ-パターン間 30[mm]の時の初期画像

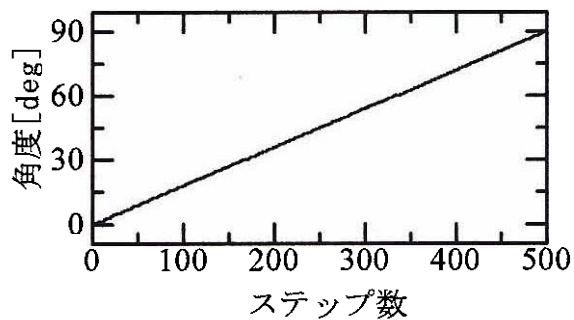


図 13 角度検出結果

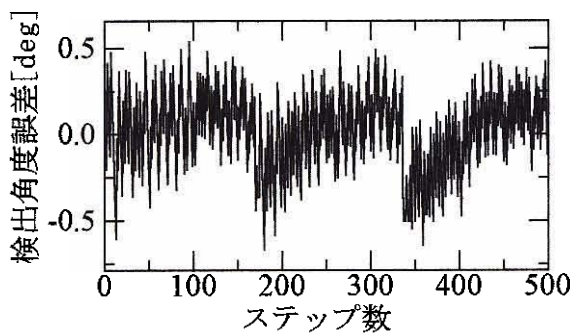


図 14 検出角度誤差

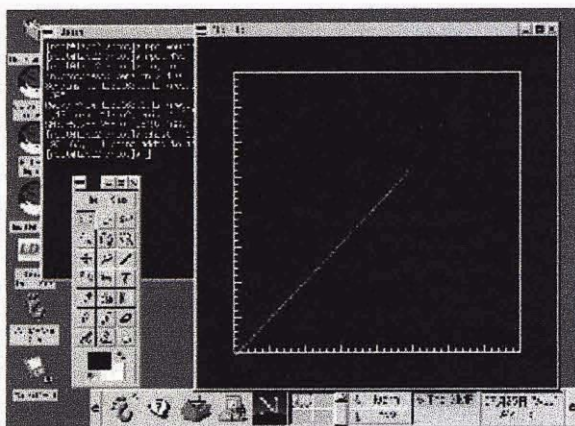


図 15 サーバ機上のグラフ

この時、目標値との誤差は最大 $0.67[\text{deg}]$ 、平均 $0.20[\text{deg}]$ となった。また、このパターンを使用したとき、 30 度を $29.98[\text{deg}]$ 、 60 度を $59.99[\text{deg}]$ とほぼ正確に認識するため、検出範囲 30 度から 60 度の間の角度の変化量を角度分解能とすることを考えた。しかし、本研究で用いたパターンは角度検出に $\cos$ 関数を使用しているため、角度の変化量は一定ではない。よって、変化量の平均を角度分解能とした。この時、実験結果の角度分解能は $0.58[\text{deg}]$ となった。

7. おわりに

Linux を用いたディスクレス機上で画像による回転角検出行い、平均誤差 $0.20 [\text{deg}]$ が得られた。また、この結果からディスクレス機を用いた計測・制御システムの有効性が確認された。

参考文献

- 1) スタークラスター：Linux ハンドブック，242/243，ナツメ社(1999)
- 2) 秀和システム出版編集部：Red Hat Linux で作るネットワークサーバ構築ガイド改訂版 6.2 対応，513/523，秀和システム(2000)
- 3) 鳥脇純一郎：画像理解のためのデジタル画像処理 [I]，158/188，昭晃堂(1988)
- 4) 高木幹雄，下田陽久：画像解析ハンドブック，575/580，東京大学出版会(1991)