

## A.2. プログラム制御方式による図形読取装置

大泉充郎, 高橋 理, 小川靖彦, 宮崎正俊 (東北大学)

### § 1. 序

計算機へ図形を入力するための方法や装置はいろいろ考案されている。東北大学計算センターでは、図形処理の研究を行なうために必要な図形読取装置の実験を行なってきたが<sup>(1)</sup> この程、試作装置 (READAC 1000S) が完成したので報告する。

この装置には2つの目的がある。1つは比較的簡単な曲線(データ曲線)を読取ってこれをそのまま計算処理用のデータとして使用することであり、他は図形処理の研究に必要なあらゆる図形を入力することである。このために本装置は固定記憶にプログラムを内蔵しており、単独で論理操作を行ないながら自動的に図形を追跡して読取ることができる外、小型計算機 (NEAC-2230) とオンラインで接続して計算機からの命令で動作できるようになっている。

### § 2. 装置の構成

本装置は処理装置本体、リーダ、センサー、紙テープさん孔機によって構成されており、更に小型計算機及びX-Yプロッタへのインターフェースをもっている(図1)。

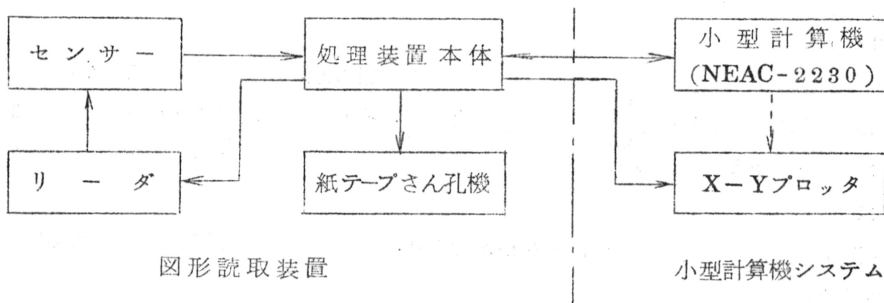
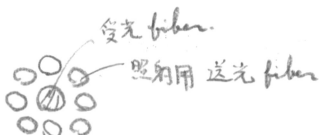


図1 図形読取装置システム

#### 2-1 リーダ

X-Yプロッタ (0.1mm Step, 2.5ms/Step) のプロット・ペンの代りに読取ヘッドを取付けたもので、これを処理装置の命令で駆動する。読取ヘッドは送光ファイバー及び受光ファイバーでセンサーに接続されている。読取面には1本の受光ファイバー(直径約0.2mm)とその周囲に8本の送光ファイバーが配置されている。送光ファイバーからの光で記録用紙を照射すると、その反射光は受光ファイバーを通してセンサーへ導かれる。



紙面の距離 0.1mm.  
◎ ファイバー束の中心からの距離

## A-22

### 2-2 センサー

照射用光源、光電変換器、A-D変換器などで構成されている。受光ファイバーからの光は、あらかじめセットされているレベルに従って0又は1の信号(白と黒)に変換される。一方、濃淡図形の場合には白と黒の間が8段階に量子化される。このときの量子化特性は直線、対数及び逆対数特性を選択することができる。

### 2-3 処理装置本体

本体は処理部と操作部に分けられる。処理部は1K語の固定コア・メモリ(1語32ビット、メモリ・サイクル2 $\mu$ s)をもち、読取ヘッドの駆動命令や信号の読取命令などを含む17種の命令を備えている。操作部ではプログラムの動作に必要なパラメータを設定する。

読取装置で得られた結果は線分の方向又は図形の濃淡を示す1~8の数値であり、これは紙テープさん孔機に出力される。また、X-Yプロッタを接続した場合は、モニタとしてこれを同時に動作させることができる。小型計算機とのオンライン動作に切換えたときは、計算機の入力命令でこの値をレジスタへ読込むことができる。

## §3. 機能

この装置は、プログラム制御により曲線図形、ひろがりのある図形の輪郭、図形の濃淡を読取ることができる。読取りは読取ヘッドを走査し、その位置と読取信号の組合せを基本情報として処理することによって行なわれるもので、このときの走査と処理の方法は読取る図形の種類によって異なる。

### 3-1 曲線図形

曲線の中心追跡の場合は正方形走査を行なって曲線の方法を逐次判定していく。曲線に4分岐以内の分岐点(交差点も含む)がある場合でも、それが8個以内であればすべての曲線を追跡することができる。分岐処理を行なわない単一曲線であれば、スパイク状の分岐の数に制限はない。

追跡できる線の太さは0.2mm~6mmであり、特に単一曲線の場合は2mm以内の断線は断線情報を出し追跡を継続する。

### 3-2 ひろがりのある図形

ひろがりのある図形に対しても正方形走査により、その境界を追跡することができる。

### 3-3 濃淡図形

濃淡図形に対しては全面走査を行なう。全面走査の走査線の長さは25.6mm $\times$ n(nは1~8の整数)及び270mmの9段階で、走査線の数には制限はない。走査線の間隔および走査線上の読取点の間隔はともに0.1mm $\times$ 2<sup>n</sup>(nは0~6の整数)で、それぞれ独立に設定できる。

## §4. 曲線追跡のプログラム

曲線追跡の基本は正方形走査であり、1回の走査で得られる各読取点の情報から曲線の進行

方向、分岐点などを判定し、走査の基点を逐次移動させながら追跡を行なう。

#### 4-1 正方形走査

正方形走査における読取点と読取信号の一例を図2に示す。<sup>(2)</sup> 図の白丸は読取点が線上にないこと(信号が0)を、また黒丸は線上にあること(信号が1)を表わす。信号の読取点の数は $2^n$ 個( $n$ は3~8の整数)、読取点間の距離は $0.1\text{mm} \times 2^n$ ( $n$ は0~6の整数)を任意に設定できる。

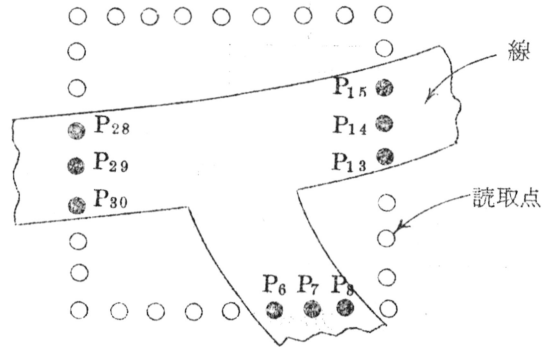


図2 正方形走査

正方形に走査していく途中で読取信号が0から1になり、次に0に変れば(信号1の点群があることになる)線を横切った

と判断し、1の中間点を線の中心点とする(図では $P_7$ ,  $P_{14}$ ,  $P_{29}$ )。横切った線(点群)の数が2個なら単なる曲線, 3個又は4個なら分岐, 鋭角があるものと判定する。走査の基点が進行してきた過去方向を追跡禁止方向とし、この正反対方向に最も近方向を追跡方向(基点の移動方向)とする。この方向は正方形の中心からみて8方向に正規化される。

#### 4-2 分岐点と鋭角処理

分岐点は点群が3個以上になったことによって判定できるが、最初に分岐点に到着してしばらくの間、点群が3個以上の走査が続く。これを1つの分岐点と判定するには、最初に点群が3個以上になったときから2個になるまでを分岐の処理中であることにしておけばよい。

分岐点では追跡禁止方向、既に追跡した方向、その点の座標(分岐点座標)を記憶しておき、分岐点に戻ったときにこれらをもとにして次に追跡すべき方向を選択する。また、追跡時には2次元のベクトル(1~8方向)を累積しておき、ベクトル和が相対的に零になったことを検出することにより、曲線のループを判定することができる。

曲線に鋭角が存在するときには点群が4個になる場合がある。鋭角では追跡禁止方向と追跡方向の点群を結ぶ線分の同一側に他の2つの点群が存在する。一方、点群が4個の分岐点では他の点群が両側に分れる。この位相関係によって鋭角と分岐点の区別を正確に行なうことができる。一方、鋭角に接近したときには、点群が2個から3個、4個へと移ることがある。この場合3個のときには一応分岐点と判定しておき、次に4個になった時点で位相関係により鋭角であることを判定して、分岐点の判定を取消せばよい。

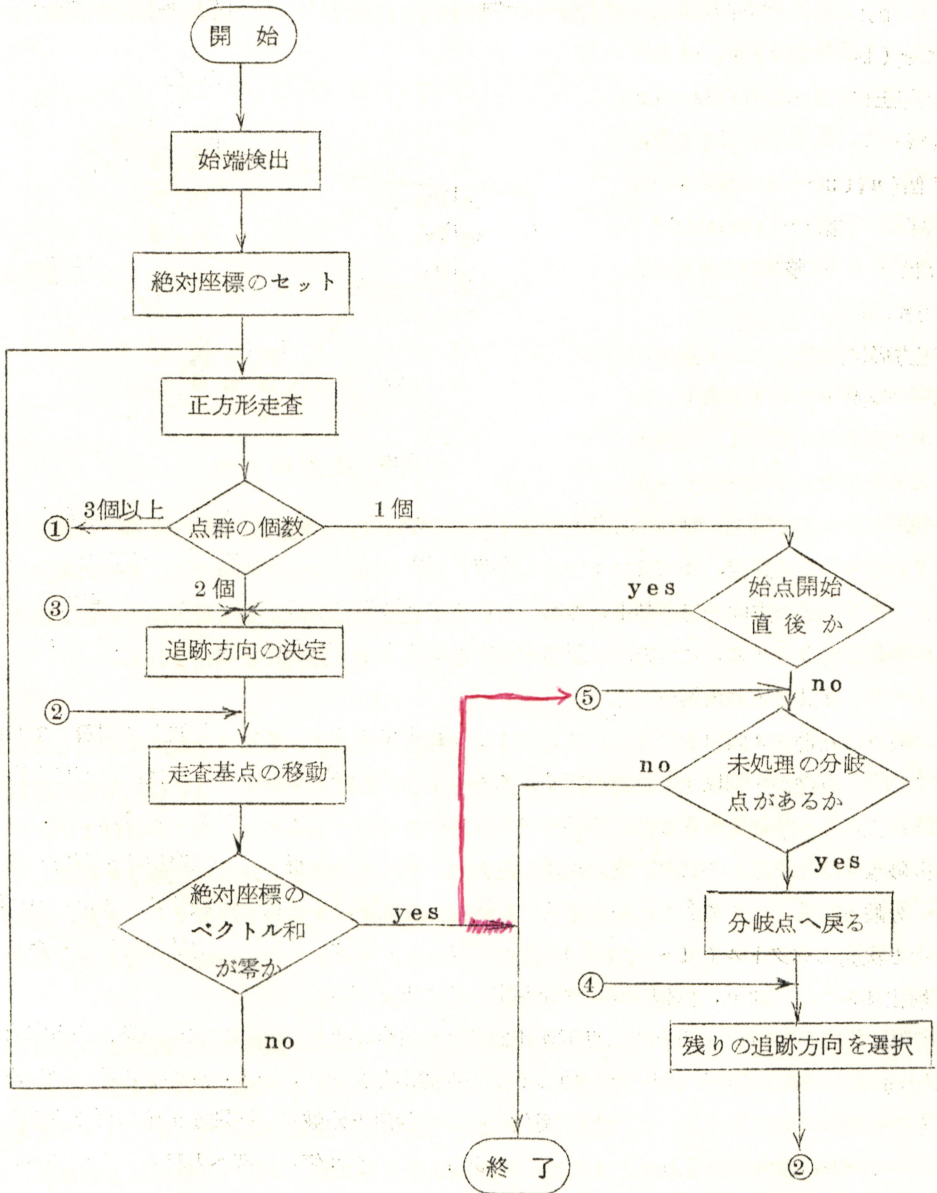


図3 曲線追跡のフローチャート(その1)

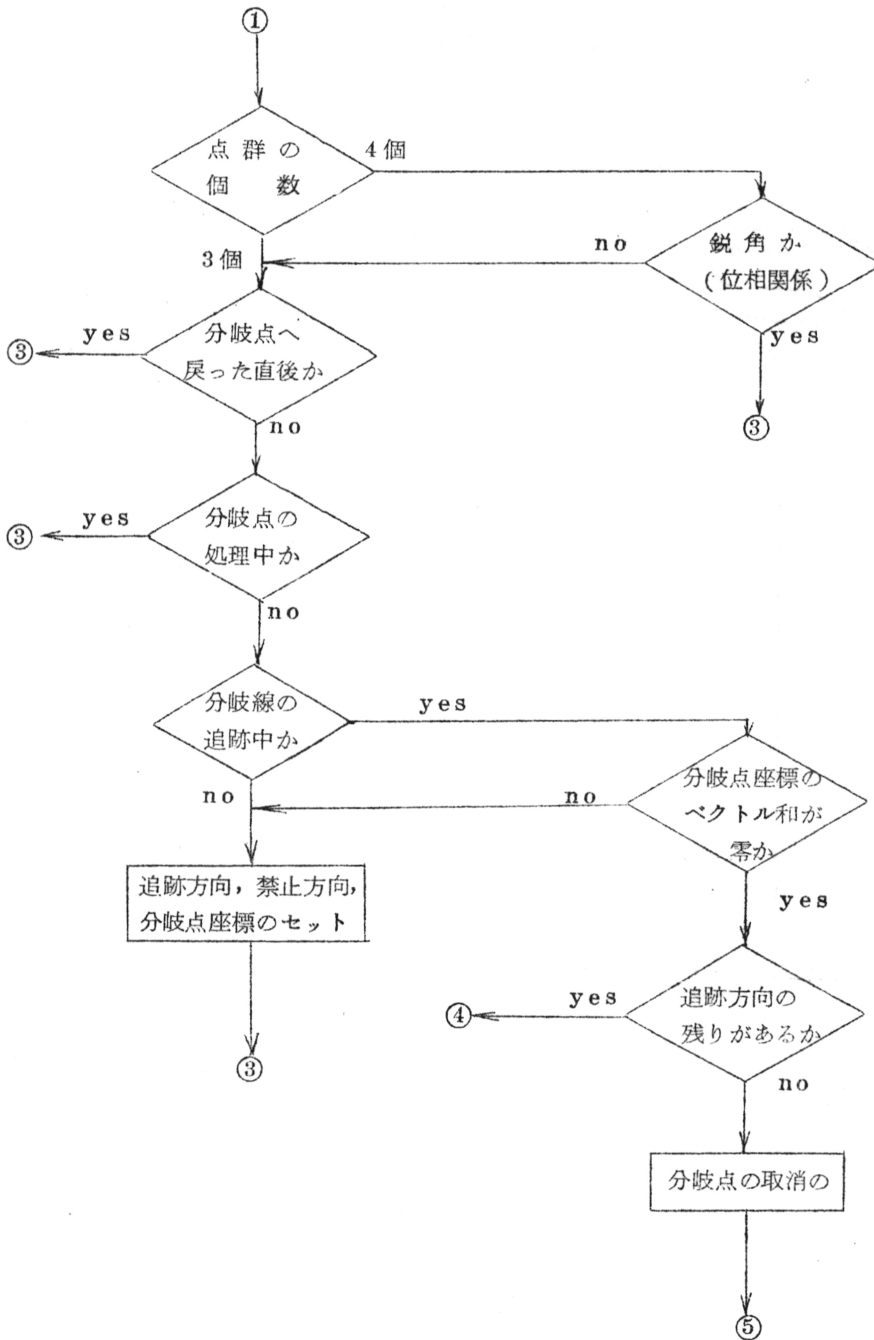


図3 曲線追跡のフローチャート

§ 5. 使用例 (左側が原図形, 右側に追跡結果を示す)

→ 入刀, 右 = X-Y plotter 1 の出力

5-1 データ曲線

5-1-1 実線

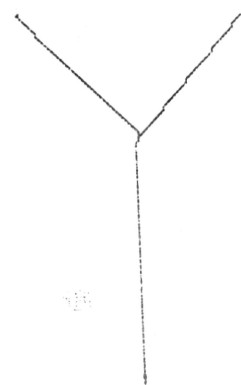
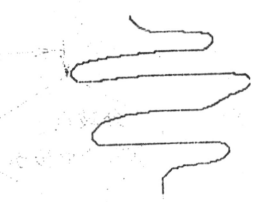
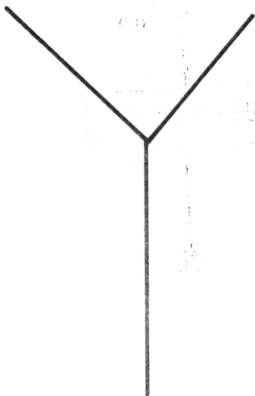


5-1-2 断線

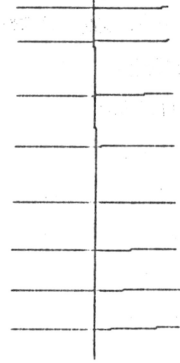
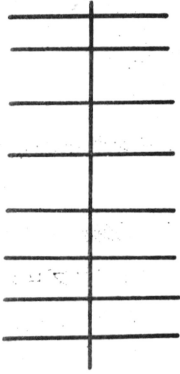


5-2 一般線図形

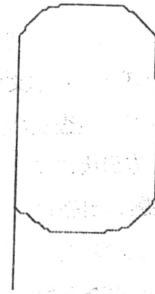
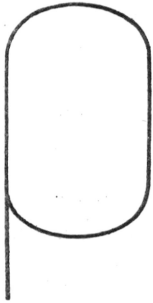
5-2-1 分岐



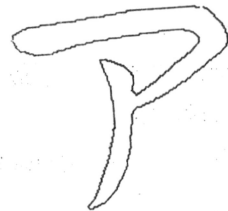
5-2-2 交叉



5-2-3 ループ

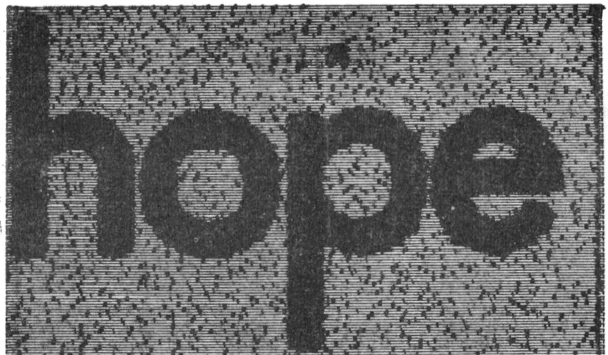


5-3 輪郭追跡



5-4 全面走査

hope



## § 6. むすび

以上、図形読取装置の概要及び簡単な読取例について述べたが、本装置の主な使用方法として次の三通りが可能である。

- (1) 読取装置の内臓プログラムの制御により、オフラインでデータの作成。(紙テープ出力。追跡速度はテープパンチ速度におさえられる。)
- (2) (1)と同様に内臓プログラムにより図形を読取り、オンラインで計算機に直接データを出力する。
- (3) 内臓プログラムを完全に切離し、計算機からのオンライン制御によって図形を処理する。このことから明らかなように、(1)、(2)の読取動作は完全にマイクロプログラムによって決定され、それ以外の動作は一切不可能である。我々は、基本的な走査法(正方形走査、全面走査)及び処理法(分岐、交叉、ループ、断線等)などを、固定記憶装置を用いて内蔵させたが、分岐、交叉などの初歩的な判断、或いは走査、追跡等のロジックにも問題が山積しており、今後も検討を加えて行きたい。従って、上記(3)の機能ももたせた訳であるがこの場合には本装置の目的からは外れるが、リーダーにペンをさしかえるだけで、オンライン、X-Yプロッターとして使用することも可能である。

又、リーダーは、本文中にも述べてあるように、デジタルX-Yプロッターを流用しているため、図形の走査及び追跡速度は、本質的にプロッターの速度におさえられてしまう。

例えば正方形状に32ポイントの走査を行なう場合、最小限80msec必要である。

我々の実験の出発点が、デジタルX-Yプロッターの利用であり、これを活用する以上、処理速度を上げるためには走査のパターンを検討し、可能な限り無駄なステップを省くこと、例えば簡単な走査と複雑な走査の組合せ或いは曲線の追跡中の過去のデータからその凹凸を判断し、滑らかな部分はステップを粗く、変化のはげしい部分は細かく追跡する様な柔軟性をもったシステムを考慮することも必要であろう。

欲を云えば、現システムのワンポイントずつの検出ではなく複眼のように複数個の読取部を設け、同時に多くの点を読取って論理判断を加え、機械的動作は複眼の移動即ち図形の追跡だけに限れば処理速度はかなり向上するものと思われる。

最後に、この研究に協力いただいた岩崎通信機株式会社の諸氏に深謝する。

## 文 献

- (1) 高橋、小川、大泉：Incremental X-Y Plotterによる図形読取について  
(第9回プログラミング—シンポジウム報告集)
- (2) 高橋、宮崎、小川、大泉：計算機へのパターン入力について  
(電子通信学会電子計算機研究会資料、1968.6)



本 PDF ファイルは 1970 年発行の「第 11 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

[https://www.ipsj.or.jp/topics/Past\\_reports.html](https://www.ipsj.or.jp/topics/Past_reports.html)

### 過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者 (論文を執筆された故人の相続人) を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 ([tsuji@math.s.chiba-u.ac.jp](mailto:tsuji@math.s.chiba-u.ac.jp)) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日 ~ 2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>