# 文　献　紹　介

## 72-23　2 値図形の連結成分の収縮について

S. Levialdi: On Shrinking Binary Picture Patterns. 〔C'ACM, Vol. 15, No. 1, Jan. 1972, pp. 7～10〕 key: counting binary patterns, shirinking, parallel processing, multiple connected pictures.

"0" と "1" で表わされた2値図形中に含まれている最大連結成分の個数を数える方法について考察している.

画像は画面中にとられた正方格子点における "0" と "1" の2値のパターンで与えられている. 画素間の連結の定義は, 1つの画素に関してその周囲8個の画素を連結しているとしている. 収縮の手順の概略は, 1つの連結成分 ("1" からなる1かたまりのパターン) に対しその連結成分の外郭周辺部の "1" を順次 "0" に変えてゆき, その連結成分を孤立の "1" 1つにまで落しその数を数えるものである. この方法の最終的な目的はこの操作を LSI を用いてハードウェア的に並列に行なうことである.

実際の収縮の手順では $n$ 時刻での各格子点の値から次の $n+1$ 時刻での各格子点の値を求めるのに, 1つの格子点に着目したとき検査の "窓" としてその近傍の格子点画素を対象としている. 第 $i$ 行 $j$ 列の格子点画素の $n$ 時刻での値を $a_{ij}$ とし, $n+1$ 時刻でのその値を $a_{ij}^*$ としたとき, $a_{ij}^*$ の値はたとえば $n$ 時刻でのその画素の値 $a_{ij}$, その左の画素の値 $a_{i,j-1}$, 左下の画素の値 $a_{i+1,j-1}$, および下の画素の値 $a_{i,j+1}$ を用い下記の式で求める.

$$a_{ij}^*=[h(a_{i,j-1}+a_{ij}+a_{i+1,j-1}-1)+h(a_{ij}+a_{i+1,j-1}-1)].$$

ただし　$h(t)=0$　$t\leq 0$,
　　$h(t)=1$　$t>0$.

上式を繰り返し用いると1つの連結成分中の "1" は右上へ収縮され, 最終的には孤立した画素1つとなる. 他の収縮方法として上式以外に右上, 左上, 左下の方向へ連結成分の "1" を収縮する方式も示されている.

この方式の特徴は並列的処理に適することおよび1つの格子点画素の次の時刻での値を求めるのにその周囲の 2×2 の格子点画素の値を用いるだけであり, 従来の方式よりその対象とする格子点画素が少なくてよいことである.　(村上伸一)

## 72-24　Hough の変換を用いて画像中の直線と曲線の検出をする方法

R. O. Duda, P. E. Hart: Use of the Hough Transformation to Detect Lines and Curves in Pictures. 〔C'ACM, Vol. 15, No. 1, Jan. 1972, pp. 11～15〕 key: picture processing, pattern recognition, line detection, curve detection, colinear points, point line transformation, Hough transformation.

2次元平面上に描かれた2値画像から直線成分および曲線成分を検出する方法について述べている.

対象とする画像は写真等を濃淡で量子化しその濃度差に関する微分操作により輪郭線抽出処理が施された線図形である. 画面上の各曲線はそれを構成している画素格子点の $xy$ 座標値の系列で表わされている. この与えられた線図形の描かれている $xy$ 平面を画像平面と呼ぶ. 直線成分の検出はこの画像平面上の線図形を構成している各画素格子点を極座標平面へ変換し (その変換される極座標平面をパラメター空間と呼ぶ), その時同一の変換パラメータを持つ画素格子点の個数を数えることによって行なう. すなわち画像局面上の1画素格子点 $(x_i, y_i)$ を極座標平面 $(\rho, \theta)$ のつぎのような1つの曲線に変換する.

$$\rho=x_i\cos\theta+y_i\sin\theta \quad (0\leq\theta\leq\pi). \qquad (1)$$

このときつぎのことが言える.

「画像平面上で同一直線上にある格子点列 $(x_i, y_i)$ $(i=1, 2, \ldots)$ を (1) 式によりパラメータ空間へ変換すると, そのおのおのの格子点に対応して得られるパラメータ空間での曲線は一点で交わる.」

そこでパラメータ空間を格子状に区切り, 画像平面内の線要素を構成する各画素格子点のおのおのに対し, (1) 式によって変換された曲線をこのパラメータ空間内の格子点の系列で示す. そしてパラメータ空間上の各格子点に加算器を置き, その各格子を通る上記の曲線の数を数えることにすれば, その数は対応する画像平面内の画素格子点のうち同一直線を形づくっているものの数を示すことになる. そこでこの数に適当

な閾値を設けることにより画像平面内の直線を検出することができる．

例としては 120×120 の正方格子からなる画像平面内に描かれた立方体の後の検出を $\theta$ 方向に，9，$\rho$ 方向に 160 に区切ったパラメータ空間に変換して行なっている．最後にはこの方法の円弧の検出に応用する仕方についても触れている．

この方法の特徴は画素格子点の数を $n$ としたとき，画素格子点のパラメータ空間への変換の局数は $n$ に比例するオーダでよく，画素格子点のすべての組合せに対する直線性の判定を行なう手数 $(n-1)\,n/2$ より有利であることである．　（村上伸一）

## 72-25　レイベルド・グラフを用いた図形の記述法

Theodasios Pavlidis: Representation of Figures by Labeled Graphs. 〔Pattern Recognition, Vol. 4, pp. 5～17, 1972.〕 key: pattern recognition, generative grammar.

図形を凸な多角形に分解して，グラフで記述する方法が述べられている．図形が与えられると，まず境界を直線で近似して多角形とする．次にその多角形を primary subset と呼ばれる凸な基本小多角形の組合せに分解した後，primary subset のつながり具合をグラフで記述する．線図形に限らず一般図形の取扱いができること．前記グラフは一種のウエッブ文法で生成されることなどが興味深い．

まず primary subset であるが，これは対象とする多角形により一意に決まる．その決め方は多角形の各辺を $s_1, \ldots, s_k, \ldots, s_n$ とし，$s_k$ を含む直線によって決まる2つの半平面のうち多角形の存在する側を $h_k$ とする．$Q_0=\bigcap_{i=1}^{n} h_i$, $Q_j=\bigcap_{\substack{i=1\\ i\neq j}}^{n} h_i$, $Q_{jk}=\bigcap_{\substack{i=1\\ i\neq j,k}}^{n} h_i \ldots$ のようにいくつかの半平面によって囲まれる図形の組を順に考える．たとえば，$Q_0$, $Q_i$, $Q_{ij}$, $Q_{ijk}$, $Q_{ijkl}, \ldots$ のようにである．これを $Q$-sequence と呼べば，$Q$-sequence の各要素は，1. 空である場合，2. 元の多角形の一部分となっている場合，3. 元の多角形を一部分として含む場合の3つの場合がある．各要素のうちで2の場合であり，かつそれより前の要素がすべて1の場合，その要素を元の多角形の nucleus とする．同様に2の場合であり，かつそれより後の要素がすべて3の場合，その要素を元の多角形の primary subset とする．

nucleus と primary subset を node とし，nucleus がどの primary subset を結びつけている力を branch で表わすと，元の図形を記述するグラフを得ることができる．このようにして得られたグラフは5つの書き換え規則を持つ一種のウエッブ文法により生成することができるので，この文法を用いてグラフを parse することにより，図形の構造を分析することができる．なお，primary subset の和3元の多角形となることは定理であり，その証明は次の論文に与えられている．

T. Pavlidis: Analysis of set patterns 〔Pattern Recognition, Vol. 2, pp. 11～32, 1970〕（坂井邦夫）

## 72-26　ARPA コンピュータ・ネットワークのための機能本体のプロトコール

Function-oriented protocols for the ARPA Computer Network, S. D. Crocker, J. F. Heafner, R. M. Metcalf and J. B. Poster 〔SJCC, 1972, pp. 271-279〕 key: computer network, protocol, user-level protocol, process communication, file transfer, remote batch.

プロトコールとは，2つのプロセスが連結されたとき，その間でメッセージを交換するために必要な，形式上の全ての協定とタイミングの協定をいう．

ARPANET のプロトコールは，ユーザ・レベル，HOST-HOST, HOST-IMP, IMP-IMP の4層の各プロトコールからなる．各コンピュータサイトのコンピュータ HOST は通信制御用の衛星コンピュータ IMP をもっているのである．

どのサイトの HOST にあるプロセスも，見かけ上のネットワークにより，互いに接続したり，バイト列を転送したりすることが可能なようにプロトコールは設計された．端末ユーザは，自分のサイトの local HOST の OS にあるプログラムにより，相手のサイトの remote HOST のプログラムを起動して，remote HOST における端末ユーザに化けることができる．

7ビット ASCII コードに2つの制御用文字を追加した．その1つは remote に入力スキャンを強制するもので，バッファ溢れのとき入力を捨てていてもその中から第2の制御用文字である 'long space (200 ms)' を remote HOST に検出させて「割り込み」を果すことができる．エコー（端末の印字は全て HOST が行う）方式を採用しているシステムでは，パスワードのエコーにあたってノイズを発生したりする．

コマンド SCRIPT により鍵盤入力を local HOST

にファイルし，コマンド SEND FILE により，これを remote HOST のエディタ・システムの入力とすることができる．現在製作中のファイル転送プロトコールによれば，互換性のないファイルシステム間のファイル転送ができる．

ARPANET の重要な目的の1つは，各サイトの specialized facilities を他のサイトに供給することである．このため remote HOST へカードデックを転送して，バッチ処理をしてもらい，出力を local HOST に戻してもらうというサービスがある．将来はこのサービスを自動化して，端末ユーザが使用中でないときにも，自動的に local HOST は remote HOST の出力を受取ってプリントしておくことができるようにする．

コンピュータネットワークの困難な問題は，互換性のないデータ間の転送にある．これを解決するには，標準的な表現による方式と，仲介的表現を経る方式がある．後者は作業が進んでいる仲介的表現をユーザが指定しなければならないという欠点があり，前者については Virtual Terminal の採用，Virtual File の有望視は見られるもののグラフィクス，データ管理を扱う場面についての作業はあまり進展していない．

（川合英俊）

## 72-27 SYMBOL ハードウェアのデバッギング法

M. A. CALHOUN: SYMBOL hardware debugging facilities〔SJCC, 1972, pp. 359～368〕key: debugging, test, high-level language machine.

SYMBOL は，昨年の SJCC で紹介されたが，ALGOL に似た汎用の手続き向き言語を直接ハードウェアで実行するマシンである．オペレーティング・システムのかなりの部分もまたハードウェア化されている．Iowa 州立大学に実在する．

本論文は，このようなシステムの検査をいかにして行いつつ構成して来たが，そして保守のためにどのような工夫がなされているかを詳説したものである．

検査の第一段階は，システムを構成する要素（この場合は CTμL IC）の入荷検査から始まった．約18000個の IC をていねいに静的動的にテストしたがプロジェクトの終りごろに入荷した約100個についてだけ手を抜いたところ，この少数のデバイスは他より多くのトラブルを生じたとのことである．

一段階は，ボード検査で，IC が平均200個位のっている2層プリント板である．（これは古くさく見えるが，プロジェクトの経違から見て，1965年以前に決定された設計仕様で，1968年ごろ実装されたと推察される．）この段階で設計ミスと組立工程のミスとの両方が発見されるが，計算機によるボード・テストは，exhaustive なテストを企てる限り，時間がかかってコスト的に見合わないので，もっと研究せねばならないと警告している．

それより大規模になって来た段階のテストはより複雑であるので，SYSTEM TESTER というデバッギング用の小システムを作った．これは，SYMBOL のメイン・ブス（111本）にそれぞれアクセスできるように特製のインターフェース装置を介して，IBM 1130 を付け，それのプログラムで SYMBOL を制御したり追跡したりするものである．それをプログラムするため，DEBUG というデバッグ用言語を作り，1130 にインタープリンタを備えた．DEBUG のおのおのの文はカードで 1130 に与えられ，SYMBOL のバグが発見されると 2230 のプリンタにエラー・メッセージや付帯情報がダンプされる仕組である．何故セルフ・ダイアグノステイック設備を SYMBOL に早期に備えなかったかというと，SYMBOL においては，IO 制御のプログラムもまたハードウェアで直接実行するのであるが，この部分はプロジェクトの中では後まわしとされていたのである．

SYSTEM TESTER は，現在は，つまり保守のためにはもう使用せず，この機能はコンパクトにシステムの一部のユニットになっている．MAINTENANCE UNIT といい，SUPER BUG という言語を持ち，レパートリはせまいが DEBUG と基本的には同じ機能で，DEBUG によるライブラリを書き直したところのライブラリを持つとのことだ．お見事．

（真子ユリ子）

## 72-28 TSS を開発・使用して得られた経験

Experience gained in the development and use of TSS, R. E. Schwemm〔SJCC, 1972, pp. 559-569〕key: TSS, soft-waredevelopment, interactive systems, system performance debagging aids, TSS development management.

モデル 67 による TSS/360 に市場性をもたせるのは，技術的にも経済的にも非常に困難であった．1967年10月に完成した version I は，「安定な hard core をまず作り，測定に基いて修正拡張する」という原則

にのっとり，一年かかって version II に補強された．レジデントスーパバイザは1965年の設計のままでよかったが，ユーザの関連する部分は全面的に作り直された．ユーザが開発に参加できるようにしたため，6529個中2442個の誤りはユーザが発見した．エディタの応答時間4秒，データセット作成の応答時間7.5秒のうちに実行できる最大タスク数を性能の物指しとしと改良を続けた．動的な環境を再現可能にするため Measurement Driver を 360/40 に作り，カーネギーメロン大の 360/67 用 SLIN と互換性をもたせた．

内部性能の記録のために，ハードウェアモニタ SPAR，ソフトウェアモニタ SIPE の他に，各モジュール通過時間とモジュール内 SVC 頻度を知るためのトレーサ ITM を設けた．ユーザの開発した XDEMON では，特権ユーザに優先権やスケジューリングパラメタの変更を許している．

TSS レジデントスーパバイザは BOS/360/40 のジョブとしてマクロアセンブラで書き，これをモデル40に作った67シミュレータにのせた．これをモデル67でもデバッグ可能にするため，このうち virtual support subsystem の部分はインシャライズのときにユーザ個有のものも使えるようにし，Resident support subsystem の部分については，CP-67/CMS というOS の中に設けられたモデル67の virtual machine に，ユーザ個有のものをのせることとした．

大規模な相互作用システムの開発管理については，誤り訂正グループのほかに設計変更審議グループ New Scope Review Board を設けた．後者は，8項目の書式からなる沢山のプロポーザルの中から最も性能を改善するものを採用して，その実現に資源を割当てた．この件数は568件にのぼる．よい設計とは，続出する要求を受入れて変更することが可能なものをいうのである．1971年10月には version 8.1 ができている．

当面の課題は，信頼性の向上にあり，このためには稼動中にも測定できる新しいベンチマークを作らなければならない．工業にとって，コンピュータの可用性よりも効果のある優秀な人間の可用性を向上させる唯一の方策は，ユーザにインタラクティブ・システムを使用させることである．コンピュータ工業の隘路は，ハードウェアの設計製作にあるのでなく，ソフトウェアのそれらにある．　　(川合英俊)

# ニュース

## コンピュータ装置の予測

米国の著名な調査会社カンタム・サイエンス社 (Q-SC) はこのほど「コンピュータ装置予測」と題する報告書を出した.

この報告書で QSC は「ユーザーの EDP 支出は, 今後6年間は年間 15% のペースで増大し, 特にハードウェアよりソフトウェアへの支出が急速にふえる」と述べている.

また, 改良されたオペレーティング・システム (OS) とか, より高速の周辺機器によって, コンピュータ・ユーザーはごくわずかのコスト, あるいはまったく追加コストなしに彼らのシステム機能を高めることが可能になったことなども報告されている.

以下に, 同報告書の主なものをあげる.

- EDP 市場全般からみると, ハードウェアの重要性はユーザーにとって徐々に減少している. 一方サービスの重要性は増大の一途をたどり, 結局コンピュータ・サービス産業は年間24%の伸び率を示すことになろう.
- ごくわずかのコストあるいはまったく追加コストなしにシステムの機能を高めることについては IBM 3330ディスク・システムといった新型高速周辺装置がその好例である. しかしこれらの新型機はメーカーの純粋な収入を減少させる可能性もある.
- ハードウェアとソフトウェアの改良によって, コンピュータ機能は年間41%という率で増加すると予想される. たとえば IBM 360/50 のコンピュータ能力は65年—70年にかけて3倍に増強された. 少なくともここの半分はソフトウェアの改良によるものである. 残り半分はコンフィギュレーションとか周辺装置の改善によるものである.
- 1972年には, 典型的 EDP ユーザーの支出の47%はハードウェアに向けられ, 残りはソフトウェアとオペレーション・コストに半分ずつ割りふられるだろう. 1977年までには, この比率がハードウェア42% ソフトウェア 29%, オペレーション 29%というぐあいに変化することになろう. 同じく1977年の時点では, CPU が典型的ユーザの EDP 支出に占める割り合いはわずか5%程度だろう.

## 情報処理関係学科の新設

わが国の大学における情報処理関係学科の設置は, 最近とみに増加しつつあり, 本年度は, 国立大学では茨城大定工学情報工学科, 滋賀大学経済学部管理科学科および神戸大学工学部システム工学科が発足した. 各学科とも, 講座4, 担当の教授および助教授各1名, 定員40名となっている.

また私立大学では, 専修大学経営学部情報管理学科 (定員80名), 玉川大学工学部情報通信工学科 (定員50名) のほか長崎造船大学管理工学科の新設が決まっている.

(なお, 上記大学のほかに, 本年度新設または今後の新設予定がありましたら, 本学会までお知らせください. ——文献ニュース小委員会)