TR－A－0048

67

視覚・認知機構研究における並列処理計算機 NCUBEの利用について（プログラムリスト付）

三宅　誠　乾　敏郎　川人　光男　曽根原　登
Sei Miyake Toshio Inui Mitsuo Kawato Noboru Sonehara

本郷　節之　池田　孝利　中根　一成
Sadayuki Hongo Takatoshi Ikeda Kazunari Nakane

1989. 4. 6

ＡＴＲ視聴覚機構研究所

© ㈱ＡＴＲ視聴覚機構研究所

目次

１．はじめに…… 1

２．視覚・認知機構研究と並列処理計算機…… 2

2.1 ニューラルネット…… 2

2.2 視覚・認知機構の計算論的研究…… 3

2.3 ニューラルネットの具体化研究－ニューロコンピューター…… 4

2.4 並列処理計算機の利用－画像処理とニューラルネット－…… 5

３．並列処理計算機ＮＣＵＢＥ…… 8

3.1 導入の経緯…… 8

3.2 システムの概略……11

４．光線追跡法による画像生成……15

５．Sobel-Filtering による輪郭抽出……18

６．ヒトの網膜における情報処理とそのモデル……23

７．エネルギー最小化による画像復元とエネルギー学習……34

８．ネオコグニトロンによる手書き数字認識……47

９．３層逆伝播学習ネットワークによる画像データ圧縮……53

１０．前注意過程における図形の特徴点抽出モデルと画像復元……63

１１．エネルギー学習をする局所並列確率アルゴリズムを用いた輪郭抽出‥74

１２．おわりに……88

付録１．ＮＣＵＢＥシステム操作マニュアル
付録２．プログラム例

１．はじめに

ＡＴＲ視聴覚機構研究所では１９８７年４月１５日にハイパーキューブ型並列処理計算機ＮＣＵＢＥを導入、以後視聴覚機構の計算論的研究に使用しほぼ２年が過ぎようとしている。本稿では国内で初めて研究ツールとして購入したＮＣＵＢＥのハードウェアとソフトウェア、及びＡＴＲでの利用実績についてまとめた。

本稿が今後の「並列処理」、「ニューロコンピューティング」、「視聴覚機構」、「ビジョン」、「画像処理・理解」などの研究に新しいアプローチを提供する一助となれば幸いである。

個々の利用実績の記述については、４、５、８章を三宅が、６、１０章を乾が、７章を川人が、９章を曽根原が、１１章を本郷が、主として担当した。

なお、本稿の各利用例におけるプログラムの大部分は㈱ＣＳＫ（担当：前川雅則氏、大本明氏、米元雅裕氏）の協力を得て作成した。ここに謝意を表する。

また、いつも一緒に研究を進めている、松下博、藤井秀夫、岡本俊明、入江文平、片山正純の研究員諸氏に感謝する。

2．視覚・認知機構研究と並列処理計算機

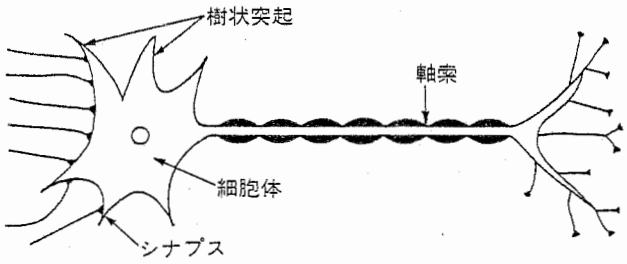
人間の視聴覚情報処理や認知・行動の仕組みを解明しつつ、そこから得られる知見を工学的に応用しようというのが本研究所の主要テーマである。

このような研究においては、実際にニューラルネットの構築原理を導出し、設計図を描くと同時にその応用可能性を示すことが求められる。

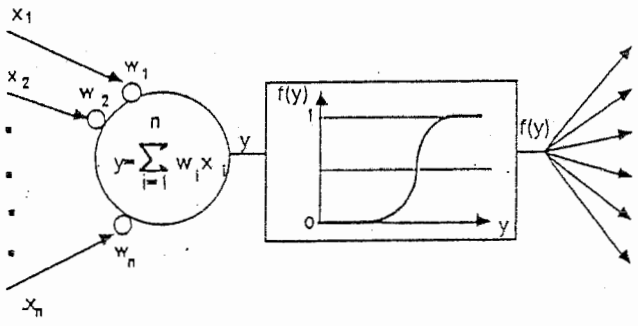
以下、本研究所における研究の内容と並列処理計算機との関わりについて述べる。

2.1 ニューラルネット

人間の視覚情報処理機構ことに脳の構造と機能については、おもに生理学者達によって着実に研究が続けられその基本構造が徐々に明らかにされつつある。神経細胞の数は１００億個以上と言われそれらが複雑に結合している。それぞれの細胞（図１）は多数の入力端末とひとつの出力端末を持ち、端末の種類によって信号を送る相手の細胞の出力を増大させたり（興奮性）減少させたり（抑制性）する。細胞は、通常パルス出力を出すのであるがパルス密度によってアナログ変調がかかっており、単純化した信号処理ユニットとしてモデル化すると図１ｂ に示すような非線形処理を行なっているものと推定される。



(a)神経細胞



(b)神経細胞モデル

図１．神経細胞と神経細胞モデル

すなわちごくおおまかに言って、脳は非常に多くのアナログプロセッサが相互に結合した計算機と見なせるが、従来の計算機とは異なり「自己組織能力」(1) を持つ。つまり、環境に適応して自己の構造を変えていく柔軟性を有する。我々が成長とか学習と呼んでいる機能はこの自己組織能力に負うところが大きい。

視覚情報処理という視点で考えれば、眼球を介して網膜上に投影された画像は並列的に配置された視細胞でとらえられ電気信号に変換されて脳に送られる。脳では入ってきた情報をさらに分析・統合し、学習、認識、連想といった高度の機能を実現している。例えば、我々は、風景やテレビジョン画像の中から自分の知っている物体や人の顔を、かたち（パターン）として瞬時に認識する。たとえそれらが多少変形したり、ノイズで汚されていても良く似た形のものであれば同じものと解釈出来る。またパターンの全体を、全て同じ重みで受容するのではなく刺激に積極的に働きかけ、興味のある部分や必要とする部分に注目して能動的に情報を処理する。

記憶については、入って来た情報をただ順番に何の脈絡もなくため込むのではなく、良く似たものや関連のあるものをまとめながら、時にはパターンそのものとして時には言葉で、想起しやすい方法で蓄積する。すなわち、従来の計算機原理にはないパターン認識、効率的な記憶方法、あいまいさを許容する柔軟性、能動的情報処理など、種々の機能を有しているのである。

このような優れた生体機能がどのようなメカニズムで実現されているかを探る研究は、すでにニューラルネット（神経回路モデル）の構築という形で進められ着実に成果をあげつつある。ニューラルネット研究は、神経細胞というプロセッサを多数搭載しそれらが相互に結合したシステムがどのように働けば高度の機能を発揮し得るか、という並列階層処理向きの計算原理を探る研究であると言ってもよい。

2.2 視覚・認知機構の計算論的研究

従来のニューラルネット研究においては生理学的知見を参考にして、計算手順やそのモデル化が試みられ多くの成果が上がっている。しかし画像や音声・言語といった実在の対象に関する処理については「何が計算の目標か？」、「その方略は？」といった計算論的視点が不足していたと言っても過言ではない。

人間の視覚情報処理の解明を課題にした計算論的研究の枠組みを最初に与えたのは、１９８０年３５歳の若さで世を去った David Marr である。１９８２年に刊行された "Vision"(2) で彼は、網膜像の強度変化（濃淡値）から実世界の３次元構造を推測することが視覚情報処理系の目標であるととらえている。従って、視覚の並列・階層的情報処理機構はこの「３次元構造の推測」という目標を達成すべく合理的に設計されているはずである。そこで階層処理の格段では何がどのような目的で計算されているかを明らかにすることが重要である。これは視覚・認知の計算理論とも呼ぶべきもので系の入出力関係の記述と言ってもよい。

ここで重要な点は計算が何らかの暗黙の仮定（拘束条件）のもとでなされているということである。網膜像から３次元構造を推測すること自身が、３次元物体から２次元濃淡画像への写像の逆問題に相当するために、拘束条件なしに計算すれば解が一意に定まらないことが多い。その反対にコンピュータグラフィクスにおける画

像生成の問題は、３次元構造から２次元濃淡画像への写像に関する順方向の問題であり、拘束条件なしに解が得られる。

そこで、この拘束条件についての知識を適切に導入することが求められる。例えば、立体視における左右の網膜像の対応決定過程においては、対象物の奥行きが連続的に変化しているという「連続性」が仮定される。運動の奥行き効果（kinetic depth effect）のように動きから対象物の構造を復元する過程では、その対象物が剛体（形がかわらない）という「剛体性」の仮定が必要とされる。後述するマルコフ確率場の考えによる画像復元問題については、各画素の持つ濃淡値がその近傍の画素の濃淡値にのみ依存して定まるという「局所性」を仮定する。

計算理論が明らかにされれば、計算が具体的にどのような手順で実行されているのか、入出力はどのような表現形式が採用されているのかを明らかにせねばならない。例えば、立体視の対応決定過程においては、入力はゼロ交差であり出力は両眼視差であると Marr は考えている。

最後に、得られた計算手順がどのような神経回路で実現(implementation)されているかが問題になる。従来のニューラルネット研究においても生理学的知見を基礎にしたモデルが提出されている。例えば、後述するネオコグニトロンについては、視覚一次野に関する回路構造的知識をヒントにした「階層的位置ずれ吸収」の機構が備わっている。このような構造についての知識も、計算理論の観点から言えばある種の拘束条件を与えていると解釈できる。つまり、上述したような入力情報そのものに由来する拘束条件と同時に、観察者である人間が進化の過程で獲得して構造化された視覚系そのものからも拘束条件についての知識が得られるのである。

従って、「計算理論」、「計算手順」、「回路構成」の３段階の過程を順を追ってクリアしていくことによってのみ視覚系が理解できるという訳ではなくそれぞれのレベルでの知識を統合したモデルを作るという新しい研究の枠組みを設定することができる。本研究所においても、このような枠組みの中で、「知識（拘束条件）を得るための実験や計算理論構築」、「システムとして具体化するための並列処理を機軸とするモデルとその実証」の両側面から研究を進めている。

2.3 ニューラルネットの具体化研究ーニューロコンピューター

ニューラルネットの特に応用的側面からの研究開発の最も大きな原動力は「並列処理メカニズム」の魅力である。これまでの工学的情報処理方式がノイマン型計算機のアーキテクチャの影響を受け逐次処理に偏っていたため、まだ未開拓の並列処理計算原理がクローズアップされていると言ってもよい。すでにニューラルネットのような並列処理向き計算機構を既存の技術でハード化し、ニューロコンピュータを開発しようという動きも活発化している。

デバイス技術においては、多数のプロセッサを組み合わせてチップ化することが夢物語ではなくなりつつあり、その技術を生かす計算方式についてはニューラルネットに期待出来るとする認識が高まりつつある。例えば、ＡＴ＆Ｔでは、物性物理出身の研究者や生理学研究者が共同し、神経細胞を摸した多数のプロセッサを結合してチップ化する研究を始動させた[3]。カリフォルニア工科大学[4]やＭＩＴ[5]

でも同種のＬＳＩ化研究が始められている。

Hecht-Nielsen によれば[6]、ニューラルネットとして新しい機能を実現できる基本原理は着実に蓄積されつつあるが、いずれも実用化レベルのシステムにはなっていない。実用化に当たっては、システムとして組み上げる高密度並列ハードウェア開発が不可欠であり、３つのアプローチが考えられるという。示唆される点が多くその概要を紹介する。

①光演算器の利用　②ＶＬＳＩ化　③仮想ハード化

①は、レンズの組み合わせやホログラフィによって相関等の演算を行なわせるという方式である[7]。このアプローチは、アナログ性とリアルタイム性を原理的に確保出来る。しかし、細胞や結合という単位で構築されるニューラルネット計算原理を生かせるかどうかは今後の課題である。

②は、ＡＴ＆Ｔ[3]等で試みようとしているアプローチで、近い将来の実現性が高い[8]。しかし現状ではモデルの多くが開発途上にあり、逐一チップ化する労力が問題となる。

③は、とりあえずニューラルネットがリアルタイムに近い形で動くデジタルコンピュータを用意することが得策、とするアプローチである。計算機を並列処理型に置き換え、より実用レベルに近づいた実験で応用可能性を見いだそうという方略である。

Hecht-Nielsen は、③のような方向で利用出来るアクセラレータとしてマークⅢ、マークⅣを相次いで発表した後、ニューロコンピュータ社という会社を作り、１ボードアクセラレータとそのソフトを製品化している。この他シナプティクス、ニューラルテックといったベンチャーも誕生している[9]。国内においてもこの種の開発が盛んになってきている[10]。本稿で紹介する並列処理計算機ＮＣＵＢＥを利用した研究もこの仮想ハード的アプローチの枠組みにあると言えよう。

2.4 並列処理計算機の利用－画像処理とニューラルネット－

並列処理向き計算原理を持つのは、ニューラルネット、画像処理、流体力学、計算物理、地質学など数多い[11]。それらの分野は、実世界を対象にした非常に解きにくい、あるいは解けても時間のかかる問題を抱え計算機による視覚化(visualization)によって実体に挑もうとする特徴を持っている。

本節では、人工知能研究に新風を吹き込む可能性の強い画像処理やニューラルネットを中心に並列処理計算機との関係について述べることにする。

2.4.1.画像処理

画像についてはこれまでもパイプライン処理やベクトル化といった並列化が試みられているが、データそのものの並列性を生かした処理が行なわれているとは言い難い。逆に、瞬間的にいっせいに入ってくるデータの並列性と膨大さという性質が大きなネックになっていた。すなわち画像を共用メモリーに行列としてストアし、

１画素ごとに取り出してＣＰＵが演算し、その結果をまた別の行列に置くといった方式を取らざるを得ない。ソフトウェアでは、データを２重のＤＯループで処理することになりその計算時間は膨大である。また、データの並列性をあえていったん放棄し、逐次処理的に扱うという翻訳の手間を求められていた。

メモリー容量も問題となる。テレビジョン画像でも数十万画素を一度に扱うことになり、１画素を１ワードでデータ表現しようとすると、数百キロワードを１画像行列で占有してしまう。このようなアーキテクチャは、メモリー価格がプロセッサのそれに比して非常に安かったという理由で生まれたものであるが、高速性を要求される新たな応用分野が拓けてこない。それでも、コンピュータグラフィクスなど画像生成については方式研究が進み、種々の並列処理向き原理が考案されている。その一つ光線追跡法は、１画素ごとに視点に到達する光を物体との交点における光学的性質（反射、屈折、透過）に従って逆方向に探索し、現実感のある画像を生成する方法である。画素ごとに独立した操作の繰り返しであるため、並列処理が得策とされる[12]。光線追跡法についてはすでにコネクションマシン、ＮＣＵＢＥで画像生成が行なわれている[13],[14]。

2.4.2.ニューラルネット

一昨年のスノーバード会議の後、ビジネスウイーク誌は初めてニューラルネット特集を組み、コネクションマシン、ｉＰＳＣ、ＮＣＵＢＥ、Ｔシリーズ等の並列計算機を新しい研究の道具として紹介した[15]。

従来の計算機は、応用を意図した規模の大きいシミュレーションを行なうには処理速度などに限界があり下記のような要請に対する解決策が求められていた。

(1)音声や画像などの実データを扱えるモデルを構築してその能力を評価すること

(2)計算が時間的に高コストとなる大規模モデルを実行する手だてを講ずること

このような要請の下に、並列計算機が新たな研究の道具として利用され始めた。ニューラルネットの特徴は『相互に結合した多くの処理ユニットが同時並列的に働いて目的とする結果を得る』ことにあり、この点でニューラルネットと並列計算機は非常に重要な共通の性格を有すると考えられるのである。

2.5 文献

(1) 福島邦彦:神経回路と自己組織化,共立出版(1979)

(2) D.Marr:Vision,Freeman（乾,安藤訳:ビジョン-視覚の計算理論と脳内表現,産業図書(1987)）

(3) L.D.Jackel,H.P.Graf,R.E.Howard: Electronic Neural Network Chips, Applied Optics, 26, 23, pp.5077-5080(1987)

(4) M.A.Sivilotti,M.R.Emerling,C.A.Mead:VLSI Architectures for Implementation of Neural Networks, AIP Conf.Proc.151, pp.408-413(1986)

(5) J.P.Sage,K.Thompson,R.S.Withers: An Artificial Neural Network Integrated Circuit Based on MNOS/CCD Principles, AIP Conf.Proc. 151, pp.381-385(1986)

(6) R.Hecht-Nielsen: Performance Limits of Optical, Electro-Optical and Electronic Neurocomputers,TRW Rancho Carmel AI Center Report,pp.1-45 (1986)
(7) N.H.Farhat:Optoelectronic Analogs of Self-programming Neural Nets: Architecture & Methodologies for Implementing Fast Stochastic Learning by Simulated Annealing,Applied Optics, 26, No.23, pp.5093-5103(1987)
(8) 土屋泰一他:鳴動するニューロチップ,日経マイクロデバイス,1988.7月号, pp.43-89(1988)
(9) E.Rosenfeld:INTELLIGENCE,April,pp.1-3(1987)
(10) 日経エレクトロニクス, 1988.6.27号,No.450,p.92(1988)
(11) G.C.Fox,S.W.Otto:Algorithms for Concurrent Processors,Physics Today, 37,5,pp.50-59(1984)
(12) 安田孝美,横井茂樹,鳥脇純一郎,鶴岡信治,三宅康二：透明物体表示のための改良光線追跡法,情報処理学会論文誌, 25, 6, pp.953-959(1984)
(13) D.Hillis:The Connection Machine, Scientific American, 256, 7, pp.108-115(1987)
(14) 三宅誠,安田孝美,鳥脇純一郎：並列処理計算機による画像の高速生成,昭和62年電気関係学会関西支部連合大会論文集, S64(1987)
(15) O.Port:Computers that Come Awfully Close to Thinking, Business Week, June 2,pp.66-71(1986)

３．並列処理計算機 ＮＣＵＢＥ

１９８６年６月のビジネスウィーク誌がニューラルネット研究ツールとしてハイパーキューブ型並列処理計算機を紹介して以来、それらの処理性能、応用などについての議論が高まった。

ＡＴＲにおいても、並列処理向きモデルを効率的にシミュレーションし、時にはリアルタイム処理や会話的処理によってその工学的応用を示せる計算機の導入を計画しつつあった。以下、機種選定を含む導入および整備の経緯について述べる。

3.1 導入の経緯

選定に入った１９８６年秋には表１に示すような商用機が登場しつつあった。本表には、機種、メーカー、最大搭載プロセッサ数だけを示しているが、その他の仕様については日経エレクトロニクス誌（1986.11.3 号）の記事に詳しい[1]。

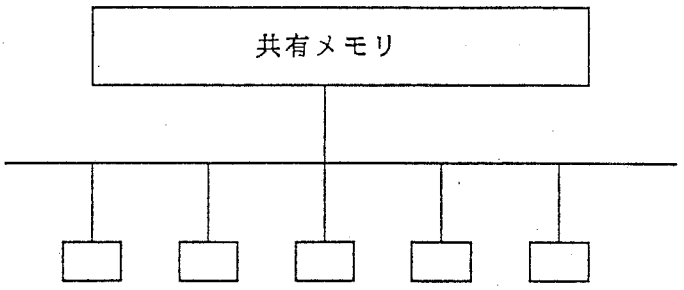
	型名	開発元	ﾌﾟﾛｾｯｻ数
密結合	FX/8	Alliant	8
	Balance	Sequent	30
	Multimax	Encore	20
	FLEX/32	Flexible	40
疎結合	Connection Machine	Thinking Machine	65536
	NCUBE	NCUBE	1024
	T-Series	Floating Point Systems	16348
	iPSC	Intel	128
	Butterfly	BBN	256

表１．商用並列処理計算機(1987.1月)

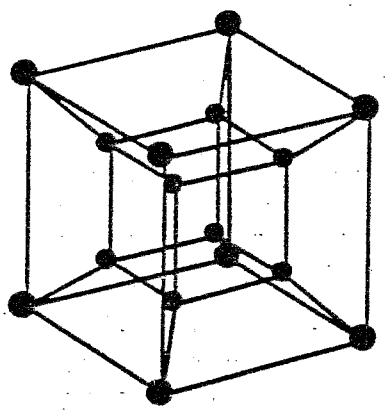
密結合型とは、複数のプロセッサで一つのメモリを共有するタイプ（図１a） で、ＯＳは共有メモリ上に置き各ローカルプロセッサで共用する。共通バスのアクセス頻度が非常に高くなることから、各ローカルプロセッサにはキャッシュメモリを付

加する。

これに対する疎結合型とは、複数のプロセッサを複数のバスによって互いに接続したタイプ（図１b） であり、その接続形態にはメッシュ、バタフライ、ハイパーキューブなど様々な結合方式がある。密結合型との本質的差異は、メモリを各ローカルプロセッサに分散配置する点である。各ローカルプロセッサにはローカルメモリやＩ／Ｏ機能が付加され、ＯＳも全てのローカルメモリに置かれる。



(a)密結合型



(b)疎結合型

図１．並列処理計算機のプロセッサ結合方式

密結合型については１９８５年初頭から商用化が始まり、稼働実績も上がりつつあった。例えばアライアント社のＦＸ／８については、ＡＴＲ自動翻訳電話研究所で音声認識等の研究に使用すべく１９８６年秋にはすでに導入を決定していた。

一方疎結合型については、わが国における使用実績もほとんどゼロに等しく研究開発レベルとの認識が先行していた。多数のプロセッサ間通信のロスをどう軽減するのか、軽減できなくともバランスをどう取るのかといった問題についての不安があったのである。

しかし、共有バスの競合によって数十台のプロセッサで理論的限界に到達してしまう密結合型に比べて数百、数千、数万といったプロセッサを非同期で独立に動かすことに多大の期待が寄せられていた。ことにニューラルネット型計算モデルは通

常数千、数万のエレメントが相互作用する構造になっており、データレベルの並列性を利用できる疎結合型の先端的アーキテクチャは大きな魅力であった。従って機種の選定においては、「将来性、先端性の疎結合」か「実績、安定性の密結合」かという選択を迫られた。

息の長い基礎研究としての広い枠組みで「新しいタイプの並列計算原理を新しいタイプのアーキテクチャで実証する」という課題に挑戦しようとするならば、疎結合型について海外も含めたユーザーの利用状況を確かめておくことが先決であると考えた。ヒアリング等の調査の結果、研究使用実績が蓄積されつつある疎結合型はThinking Machine Corp. の Connection Machine と NCUBE Corp.の NCUBE の２機種に絞られた。いずれも結合方式（トポロジー）はハイパーキューブである。

ここで簡単にハイパーキューブ結合方式について述べておく。ｎ次元のハイパーキューブでは、ノードが 2^n個あるようなｎ次元立方体が構成され、その各頂点にプロセッサが配置される。ｎが０、１、２、３、４のそれぞれの場合についてそのハイパーキューブ構造を図２に示す。

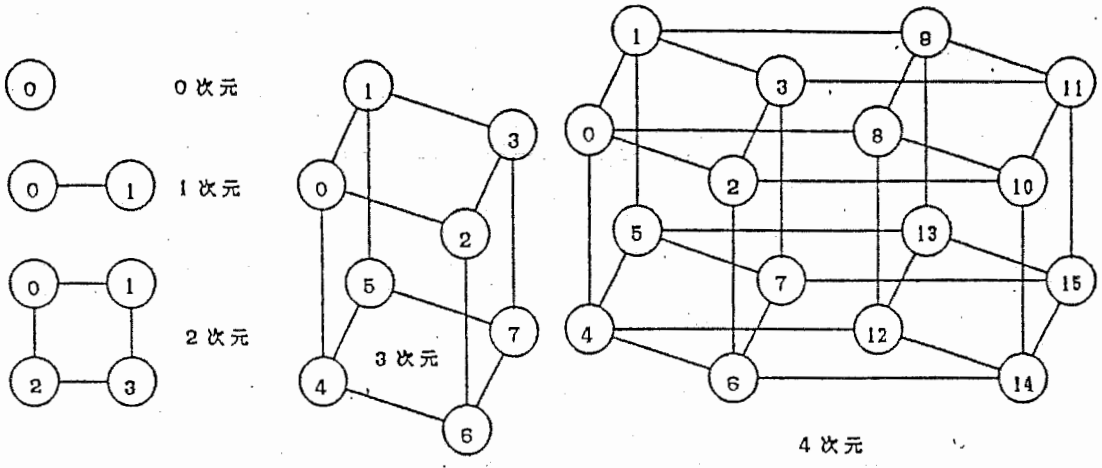


図２．ハイパーキューブ結合方式

図２から推察されるようにハイパーキューブ構造は、各ノードプロセッサから他の全てのプロセッサと等距離で通信できるような理想的構造からは程遠い。つまり、手が出ている隣の頂点（３次元ハイパーキューブの場合は隣の３個）とは短距離通信が可能であるが、それ以外の頂点のプロセッサとは別の頂点を経由して通信せねばならず通信負荷が大きい。これがハイパーキューブのネックと言われている。しかし、相手のプロセッサによって通信距離の差が大きくなってしまうといっても、その差はｎ次元座標上での差にとどまり、例えばメッシュの左下と右上間の距離と隣接点の距離の差ほど大きくはならない。

さて Connection Machine は疎結合型の中でも最も粒度の高い(fine grained)計算機であり、 65536個という「数」と引き換えに各々１ビットの簡単な計算処理を

行なうプロセッサで構成される。従ってノードプロセッサの負荷を小さくすることによってはじめてその能力を最大限に発揮できる。もしこのアーキテクチャに非常に良くマッチする計算原理を開拓できれば画像や音声の実時間処理も夢ではない。実際 MIT AI ラボ等では両眼で取り込んだ画像情報を手がかりにした実時間運動制御システムを構成している[2]。 以上のようなマシンの性格から次ぎの２点に留意して使用することが求められる。

(1) １ビット、65536個 を生かすに適した問題の選択と並列計算原理の構築

(2)各プロセッサの負荷と通信時間のバランスを取る工夫

もちろんこの２点は疎結合型に共通の問題であるが NCUBEのように数百個の32ビットプロセッサで構成される粒度の低い(coarse grained)マシンでは、かなり複雑な処理を各プロセッサに担当させることができると同時にプロセッサ負荷に比べて通信時間が小さく、適合する問題を選択する範囲が広がるとの感触を得た。NCUBEの具体的ユーザーが存在する分野としては、石油探査、流体、気象などにおけるグラフィックシミュレーションなどが挙げられ、それらの成果に関する報告を入手した[3]。

また NCUBE の原型 Cosmic Cubeを開発したカリフォルニア工科大学の G.C.Fox 教授および NCUBE Corp.の S.R.Colley 社長と面談し、 NCUBE型の低粒度マシンがConnection Machine型の高粒度マシンをターゲットとしている訳ではなく、低粒度マシンとしての長所を生かした設計指針を持っていることを確かめた。さらに実時間性を狙う次世代ヴァージョンは４倍から８倍のプロセッサ数を見込むとの考え方が提示された。

上述の事前評価に基づき、主として下記２点の理由から NCUBEを選択した。

(1)国内初の疎結合型の導入にあたって先端性と安定性のバランスがとれている

(2)Connection Machineについては、Thinking Machine社が日本への輸出を考えていない時点で導入スケジュールの青写真が描きにくかった

3.2 システムの概略*

NCUBE/10 は、公称 2.5MIPS,0.4MFLOPS の能力を持つ32ビットプロセッサを最大1024個実装できる完全分散処理型の計算機である。 NCUBEは密結合型計算機と異なり、大規模なパイプライン／ベクタ処理による高速化も、プロセッサ間での同期を取りながらの並列処理も行なわない。 NCUBEの各ノードは、ユーザーのプログラムからのみ他のノードやホストプロセッサとの通信を行なう。従って通信のプロトコルもユーザーにまかされた形態になっている。図３にシステム外観を、図４にシステムの仕様概略を示す[4]。

3.2.1 ホストプロセッサ

NCUBE/10のホストプロセッサはインテル社製 80286 で、 ノードプロセッサとユーザーとのインターフェースの役割を担っている。ホストプロセッサは、まず必要

---

* 本節(3.2) の記述の主要部分は当所視覚研究室城和貴研究員によるものである

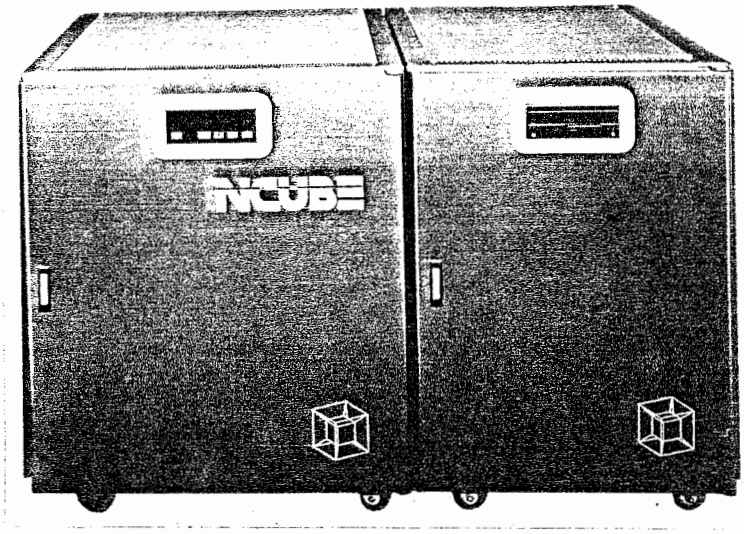


図３．システムの外観

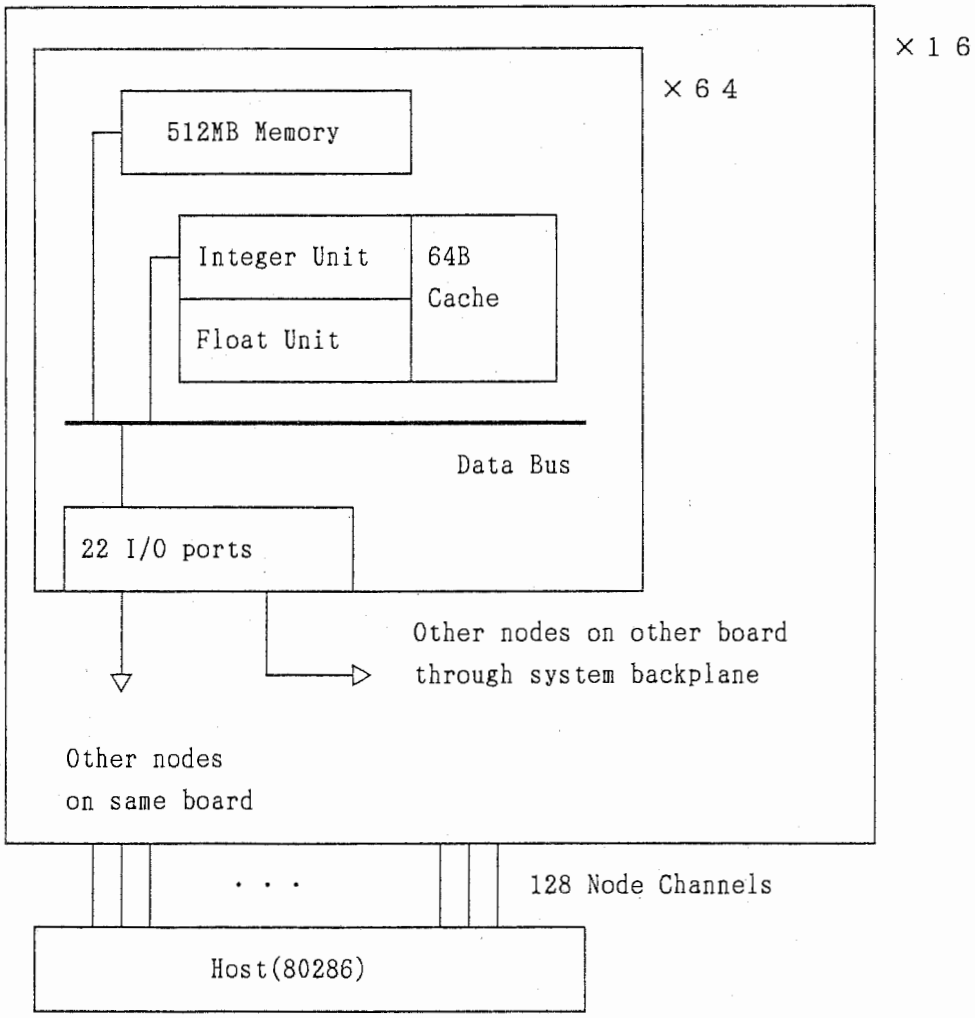


図４．NCUBE/10 のシステム仕様概略

な数のノードプロセッサをアロケートし（ただしアロケートするノード数は $2^n \leqq 1024$に限定される）、ノードプロセッサ用プログラムをロードし、必要に応じてデータを転送する。各ノードプロセッサはホスト経由でディスクやユーザーからのデータを受け取る以外は全く独立に機能する。なお、ホストプロセッサは Axis と呼ばれる Unix に似たＯＳで管理されている。また、ノードとの通信は Vortex と呼ばれる Axis の拡張システムで管理される。

3.2.2 ノードプロセッサ

NCUBE/10のノードプロセッサは、８MHz の３２ビットマイクロプロセッサに高速のＩ／Ｏチャネルを付加した形態になっている。従って、各ノードプロセッサは通常の数値計算、データ転送、制御命令の他にノード間通信を行なうことができる。

各ノードプロセッサは 512KBのローカル物理メモリを持ち、物理メモリシステムで動作する。また、通信用の Vertex と呼ばれる 32KB のシステム以外はＯＳを持たず、ユーザープログラムがホストからダウンロードされると直ちにそのプログラムを実行する。

レジスタについては、汎用レジスタが１６個、プロセッサレジスタが１２個、Ｉ／Ｏレジスタを２２個持ち、すべて３２ビット長である。汎用レジスタは整数計算、小数計算の他にアドレスレジスタとしても使用される。倍精度計算を行なう時は２個の汎用レジスタを一組にして使う。プロセッサレジスタは、スタックポインタ、プロセッサＩＤ、フォルトレジスタをはじめとする各種プロセッサ状態の貯蔵やＩ／Ｏ割り込みに使用される（ただしここで言うＩ／Ｏとは、プロセッサ間通信を指す）。Ｉ／Ｏレジスタは、他のノードプロセッサとの通信を行なう際のバッファアドレスとメッセージ長を示すのに用いられる。

3.2.3 その他

NCUBE は前述のように大規模なパイプラインベクトル化は行なっていないが、各ノードプロセッサにおいて以下のような高速化の工夫がなされている。

(1) 小数計算時に次のオペランドのプリフェッチを行なう疑似パイプライン

(2) ３２バイトのインストラクションキャッシュ

ノード間通信は２２個のＤＭＡ・Ｉ／Ｏチャネルによって行なわれる。チャネル０から９までは他ノードからの入力に、３２から４１は出力に、３１と６３はホストおよびＩ／Ｏポート（例えばグラフィックスのフレームバッファ）との入出力に使用される。チャネル０から３０と４２から６２は将来の拡張に備えてリザーブされている。各チャネルは送受信のためのメッセージアドレスとメッセージ長を示すレジスタを２個ずつ持っている。各チャネルのＤＭＡ転送速度は 180MB/Sで、プロセッサレジスタの状態に依ってメッセージの送受信、インタラプト、ペンディングウエイト等を、ノードプロセッサと独立して行なう。各ノードはブロードキャスト命令により１１の出力チャネルから同時に同じメッセージをＤＭＡ転送できる。

なおＡＴＲにおいては昭和６２年４月の時点で２５６個のノードプロセッサを実装し、その後昭和６３年２月に新たに２５６個を追加現在５１２個で稼働中である。

3.3 文献

(1) 稲葉則夫:期待と不安を乗せて飛び立った並列処理コンピュータ,日経エレクトロニクス,407,pp.119-129(1986)

(2) M.Drumheller,T.Poggio: On Parallel Stereo, Proc.IEEE Conf. on Robotics(1986)

D.Hillis:The Connection Machine, MIT Press(1985)

(3) In; Caltech:Concurrent Supercomputing Initiative at Caltech Information Package, C³P-428(1987)

(4) NCUBE Corp.: NCUBE Handbook 1.0(1987)

４．光線追跡法による画像生成[1]

4.1 はじめに

ニューロコンピュータ[2]、 流体力学、理論物理、画像処理など並列処理向き問題を抱える様々な分野において、並列処理計算理論の構築とそれらを高速実行できる新しい計算機に対する需要が高まっている。

最近では、互いに通信可能な多数のプロセッサを同時に働かせて高速化する計算機の実用化も徐々に始まっている[3]。 本章では、並列独立に分割処理させるのに適した光線追跡法[4] をハイパーキューブ（超立方体）型並列処理計算機で処理した結果について報告する。

4.2 並列処理計算機

パイプライン処理やベクトル処理も並列処理の１種であるがそれとは別に、多数のプロセッサを搭載した並列処理計算機が商品化され始めた[3]。 今回使用したシステムは、NCUBE と呼ばれるハイパーキューブ型並列処理計算機であり、ホストとは別に、32ビットのプロセッサ 256個を搭載している（プログラム作成当時）[5]。

4.3 光線追跡法による透明物体表示

4.3.1 光線追跡法

スクリーン上の各画素ごとに視点に到達する光を各物体との交点における光学的性質（反射、屈折）に従って逆方向に探索し、現実感のある表示画像を得る方法である。独立した操作の繰り返しであるため、並列処理が得策とされる[3]。

4.3.2 並列処理

NCUBE システム本体は最大1024個のノードプロセッサを搭載可能であるが、今回は、図１に示すように 256個のプロセッサを持つシステムで分割処理しながら計算結果をグラフィクス装置に表示した。

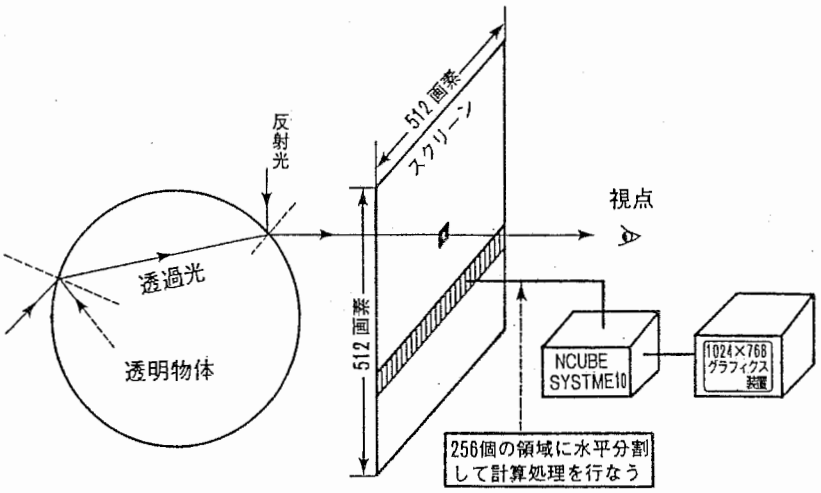


図１． 光線追跡法の並列処理

4.3.3 実験結果

図２のような９個の透明球の生成時間を表１に示す。表示も含めて汎用大型コンピュータを上回るスピードが得られた。また、汎用大型コンピュータでは、いったんディスクに結果を蓄えるため、表示プロセスが計算処理と切り離されてしまう。本システムでは、処理を終えたノードプロセッサは直ちにその結果を表示するので、表示を含めた一連の処理時間を大幅に短縮することが可能となる。

図２．画像生成結果

計算機名	N社マイクロコンピュータ	F社汎用大型コンピュータ	NCUBE
処理時間	132分	50.5秒	27.0秒

但し，マイコンと大型汎用コンピュータの場合は計算時間のみ。NCUBEの場合は計算時間と表示処理の時間を加算した値。

表１．処理速度の比較

4.4 おわりに

本稿では、画像生成を並列処理計算機で高速化し、実時間処理やアニメーションへの利用可能性を示した。今後このような計算機の応用範囲をさらに拡大するには、並列計算向き問題を解くための計算理論の構築と蓄積が重要課題である。

4.5 文献

(1)○三宅誠,安田孝美,鳥脇純一郎:並列処理計算機による画像の高速生成,電気関係学会関西支部連合大会講演論文集, 2A-1-5, pp.137-140(1987)

○三宅誠:ニューラルコンピュテーション,信学会誌, Vol.70, No.12, pp.1255-1262(1987)

○S.Miyake, T.Inui:Neural Computation Research for Vision, Cybernetics and Systems'88, Kluwer Academic Publishers, pp.1007-1014(1988)

(2)三宅誠:日経ﾏｲｸﾛﾃﾞﾊﾞｲｽ,1987年3月号,p.98

(3)稲葉則夫:日経ｴﾚｸﾄﾛﾆｸｽ,No.413,p.159(1987)

(4)安田孝美,横井茂樹,鳥脇純一郎,鶴岡信治,三宅康二:透明物体表示のための改良光線追跡法,情報処理学会論文誌,Vol.26,No.4,pp.591-599(1985)

(5)NCUBE社:NCUBE HANDBOOK Ver.1.1(1986)

5．Sobel Filtering による輪郭抽出[1]*

5.1 はじめに

ＭＩＭＤ構成並列処理コンピュータＮＣＵＢＥ[2] の画像処理への適用性を検討した。本章では画像認識の前処理として輪郭抽出をとり上げ、従来からよく用いられる Sobelの１次微分についてＮＣＵＢＥの能力を検証する。

5.2 輪郭抽出処理

輪郭やエッジの抽出は神経生理学的にみても認識過程における重要な要素であることが示唆されており[3],[4]、 原画像の豊富な情報を含むことが多い。従来から工学的にも種々の方法が提案されているが[5]、 ここでは、並列処理に向き比較的よい結果を示すと言われる Sobelの１次微分オペレータを採用した。また輪郭処理の前処理としてスムージングによる雑音除去を施した。

5.3 ＮＣＵＢＥと処理方法

ＮＣＵＢＥは１個のホストとハイパーキューブ型に接続された複数の（当時２５６個）３２ビットマイクロプロセッサから成る並列処理計算機である（図１）。

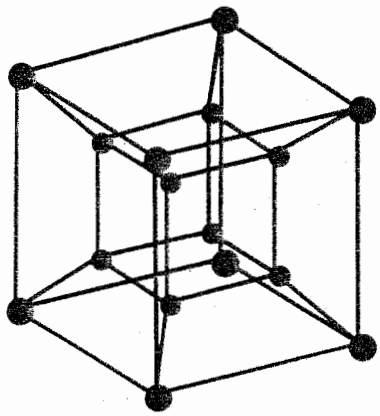


図１．16(=2^4)個のノードプロセッサをハイパーキューブ型につないだ例

使用した画像の画素数は 256×256とし、各ノードに１ライン(256画素）分の計算処理を割り当てた。

ホスト側のプログラムは、各々のノードに計算法（独立したプログラム）と必要な画像データ（３ライン分）を送る。各ノードプロセッサは独立に定められた処理を遂行し、終了しだい結果を返す。ノード側ではプログラムがロードされると直ちに処理を開始する。今回の実験では、初めに雑音除去とその結果の受渡し（ノード間通信）を行なう。次いで Sobelの１次微分（並列処理）を施し、処理結果をホストに返す（図２）。

---

*本実験は高橋裕子（NTT HI研）が S.62.12.1-S.63.1.14 に実習生として担当した.

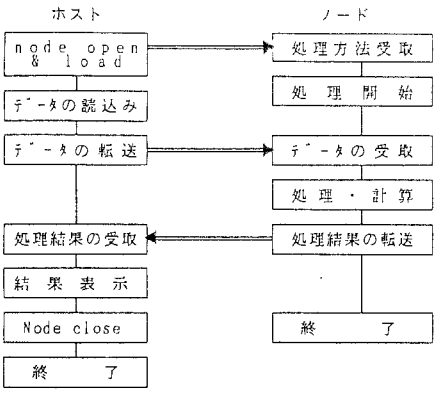


図２．処理の手順

以下実際のプログラム作成について概説する。

ホスト側では、パラメータの読み込み、画像データの読み込み、ノードプロセッサのプログラムロード、各ノードに必要なデータ転送、さらにノードからのデータ受け取り及びその結果の表示を行なう。ノードプロセッサはロードされたプログラムと与えられたデータを使って独立にその処理を行なう。今回はノードの処理として Sobelの１次微分を行なった。

256×256，8bitの画像データに対し 256の各ノードに１ライン(256画素)分 の処理を割り当てた。それぞれのノードには３ライン分のデータを送った。３×３近傍のデータのみが必要な Sobelの１次微分の場合、各ノードは完全な並列処理を行なうことができる。従って、このプログラムではノード・ノード間通信は行なわない。

Sobel の１次微分を施すにはまず求めたい画素の３×３近傍の値に、図３に示す Sobel 演算子 d_x，d_y を掛けてそれぞれ加算し、d_1，d_2 とする。絶対値の和 $|d_1| + |d_2|$ または $\sqrt{|d_1|^2 + |d_2|^2}$ を求める値とする。今回のプログラムにおいてはこの２種類の方式のうち絶対値の和を採用した。

$$d_x = \begin{Bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{Bmatrix} \qquad d_y = \begin{Bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{Bmatrix}$$

図３．Sobel演算子

プログラムのフローを図４に示す。ノードプログラムは無限ループになっているが、ノードにデータが送られるまでその時点で待ち状態となる。ノードプログラム終了(node close)が発せられた時（または全ての処理が終了した場合）、ノードのプログラムは終了する。

上述の輪郭処理の前処理として雑音除去を追加してあるので概説する。このプログラムにおいてもホストによるデータ転送は３ライン分の画像データである。従って各ノードは、隣接ラインを計算しているノードから雑音除去の結果を受け取る（ノード・ノード間通信）。ここで用いた雑音除去とは、各画素の３×３近傍（９個の値）の中心値をとる方法である。また、後処理として Sobelの１次微分の結果にさらに雑音除去を施すプログラムも作成した。

処理速度の比較を目的として VAX8550上に上述のプログラムに相当するプログラムを作成した。VAX8550 は公称 6MIPSなので処理時間の比較によってＮＣＵＢＥの処理能力をある程度推定できる。

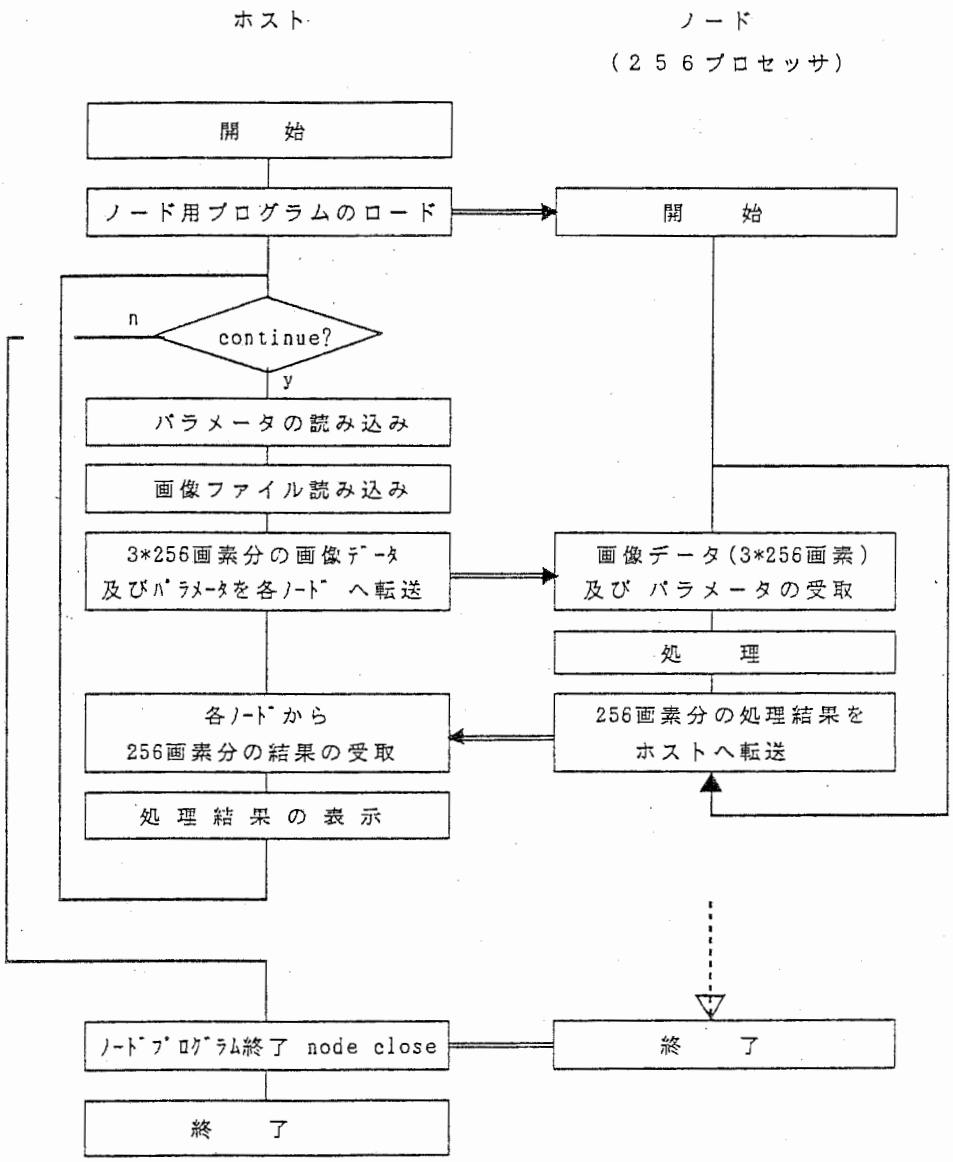


図４．輪郭処理プログラムのフローチャート

5.4 結果

図５に原画像と処理結果を示す。また、表１に処理速度と MIPS値概算を示す。それぞれのノードは公称 2MIPS, 0.5MFLOPS の性能を持つ。しかしながら単純計算による値 512MIPSは得られなかった。これは、ホストによるプログラムロードやホスト・ノード通信が他のノードを介して行なわれることや、公称値ほど実際の能力が発揮できていないなどの理由によるものと考えられる。通信および処理に要する時間は個別に計測し難いため、各ノードの処理の繰り返し回数を増加させながら所用時間を実測した（表１）。その結果、システムの能力は通信に要する時間の減少につれて増大し、約168MIPS に近づくと予想される。 Sobelの１次微分のみの処理の場合には通信ロスが無いので速度は向上し、約250MIPSに近づく。

図５．原画像(256×256,8bit)および Sobelの１次微分の結果例

回　　数	1	10	30	50	100	500	回
load+ﾃﾞｰﾀ転送	2	3	3	3	3	3	秒
処理＋結果受取	1	2	6	10	21	102	秒
合　　計	3	5	9	13	24	105	秒
MIPS値概算(*)	9	66	114	132	138	163	MIPS

(*注．MIPS値はVAX8550(公称6MIPS)で
同等の処理を行った時の処理時間から逆算)．

表１．ノードにおける処理の繰り返し回数による処理時間の変化とMIPS値概算

5.5 おわりに

画像処理の計算原理によってはＮＣＵＢＥの性能を有効に活用できることを確認した。本検討が画像認識・理解などの複雑な方式を実証する際の基礎資料となれば幸いである。ところで、今回のプログラミングでは、ＮＣＵＢＥのハイパーキュー

ブ構造利用した並列化を行っていない。今後ハイパーキューブ構造に合わせた並列化手法が開拓されることが望まれる。

5.6 文献

(1) 高橋裕子,三宅誠,梅田三千雄:並列計算機（ＮＣＵＢＥ）の画像処理性能の評価,昭和63年度電子情報通信学会春季全国大会講演論文集,D-420(1988)
(2) NCUBE Corp.:NCUBE Handbook, 1.0(1986)
(3) D.Marr:Vision,W.H.Freeman and Company(1982) 乾,安藤訳:ビジョン－視覚の計算理論と脳内表現－,産業図書(1987)
(4) 福島邦彦:視覚の生理とバイオニクス,電子通信学会(1976)
(5) 田村秀行:コンピュータ画像処理入門,星雲社(1985)

6．ヒトの網膜における情報処理とそのモデル

ヒトの網膜における情報処理様式を完全に理解するためには、個々のニューロンの時空間特性を調べるだけでなく、ニューロンの受容野密度をも調べる必要がある。なぜなら、網膜を一つの情報処理システムと見たとき、ニューロンの時空間特性は画像フィルタ特性に対応し、ニューロンの受容野密度は情報の標本化頻度に対応すると考えられるからである。本研究の目的は、心理物理学的実験データに基づくヒトの網膜モデルを提案し、それによってヒトの網膜のＸ型、Ｙ型、Ｗ型ニューロンの受容野密度を推定することにある。

6.1 心理物理学的知見

被験者にスポット光を提示し、検出閾（光覚閾）を測定すると、スポット光の面積の増加とともに閾値が低下することが知られている。通常、検査光の面積と検出閾の関係は両対数グラフで示される。このようなグラフを閾値－面積曲線(area-threshold curve)と呼ぶ。検査光が小さいときの閾値は面積の増加とともに（両対数グラフでは）－１の傾きで低下する。これは閾値と刺激光の面積が逆比例していることを意味する。すなわち（閾値）×（検査光の面積）＝一定、という関係が成立しているのである。これは何らかの機構によって検査光のエネルギーが完全に加算され閾値が決定されていることを意味しており、空間加重が完全であると言う(complete spatial summation)。完全な空間加重が成立する範囲は単に空間加重領域（spatial area）とも言われる。 検査光をさらに大きくすれば傾きは－１より小さくなる。しかし傾きは０ではなく部分的に空間加重が成立している(partial summation)。さらに検査光の面積を大きくすると閾値は検査光面積によらず一定になる(non-summation)。Inui ら[1],[2] は、空間加重領域の大きさが、網膜偏心度とともにどのように変化するかをしらべた。背景輝度は 10asb であり、 検査光は 200ms 間パルス状に提示された。 その結果、空間加重領域の直径 d(min) は、網膜偏心度 E(deg) と直線関係にあり、

$$d = 0.76E + 4.2$$

であった。すなわち、中心窩では直径約 4′、 網膜偏心度が 10° の位置では直径 12′の範囲で空間加重が成立した。 Okamoto ら[3] は、検査光の輝度の対数が時間とともに線形に増加するようにして同様の実験を行なった。この提示条件では時間的に急激な変化が見られない。その結果、空間加重領域の直径 d(min) は、やはり網膜偏心度 E(deg) と直線関係にあったが、その勾配は小さく

$$d = 0.2E + 2.2$$

であった。前述のパルス状に提示した条件、およびゆっくりと輝度を上げる条件で得られた空間加重領域の直径はそれぞれ、サルの網膜神経節細胞のＹ型、Ｘ型の受

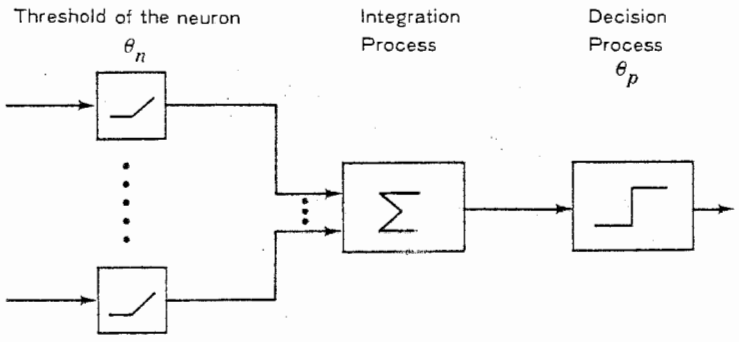
容野中心部の直径にほぼ等しい大きさであった。乾ら[2]は、抑制効果が働く範囲、すなわち抑制野(inhibitory field)の大きさと網膜偏心度の関係をしらべた。その結果、抑制野の直径は網膜偏心度とともに増大することが明らかにされた。すなわち、中心窩では直径 12′、網膜偏心度 9° の位置では直径 40′であった。この結果を、前述のＹ型の空間加重領域の大きさと比較すると、網膜偏心度によらず、抑制野の直径は空間加重領域の直径の約３倍であることが明らかにされた。以上の事実から、検査光が提示されたごく近傍では光エネルギーが加算され、その周辺部では逆に検査光に対し抑制効果が生ずることがわかる。

6.2 問題の設定

まず初めに、上記の閾値－面積曲線において不完全加重領域をどう捉えればよいのかが問題である。閾値に関してはいくつかのモデルが考えられるが、不完全加重領域のデータをうまく説明できるものはこれまでなかった。したがって各々の網膜部位における閾値－面積曲線をシミュレートできる網膜モデルを考案することが第１の目的である。次に、網膜モデルによって、個々の受容野特性のみならず、受容野密度をも推定し、網膜情報処理の不均一性を定量化することが第２の目的である。

6.3 モデル“ＴＡＭＩＴ”の構成

ＴＡＭＩＴ (Total Activity Model for Increment Threshold)は大きく２つの処理部分から構成される。第一の処理は網膜神経節細胞（ＲＧＣ）の情報処理に対応するもので、第二の処理はすべてのＲＧＣの出力の加算を行なう部分である。ＴＡＭＩＴの構成を図１に示す。ＲＧＣは網膜上に受容野をもつ閾素子であり、これらの出力の総和（全活動度）が心理的閾値を越えると光が検出されると仮定する。したがって、ＴＡＭＩＴには２つの閾パラメータが存在する。以後、ＲＧＣの閾値をθ_n、心理的閾値をθ_pと書くことにする。ＴＡＭＩＴには表１に示すように８個のパラメータが存在する。以下でこれらのパラメータについて説明する。



TAMIT

図１．ＴＡＭＩＴの構成

Maximum center sensitivity	A
Maximum surround sensitivity	AQ
Radius of receptive field center	σ_e
Radius of receptive field surround	σ_i
Threshold of one neuron	θ_n
Psychological threshold	θ_p
Receptive field density	D_r
Retinal eccentricity	E

表１．モデルのパラメータ

(1) ＲＧＣのモデル

各ＲＧＣの受容野はＤＯＧ (Difference of Gaussian)関数で記述することにし、それぞれのガウス関数の標準偏差をσ_e、σ_i、ガウス関数のピーク値を A，AQ とする。したがって、ＲＧＣの受容野の感度分布 G(r) は

$$G(r)=A\ (\exp(-r^2/\sigma_e^2)-Q\cdot\exp(-r^2/\sigma_i^2)) \qquad (1)$$

である。ここに　r は受容野の中心からの距離である。

既に述べたように受容野中心部の直径は網膜偏心度Ｅと直線関係にある[1),2)]。ここでは被験者 H.M. に関するデータを採用することにする。被験者 H.M. の場合は、

$$2\sigma_e=0.55E+4.15 \qquad (2)$$

ここで Eの単位は度であり、σ_eの単位は分である。また、受容野周辺部の直径は、網膜偏心度によらず受容野中心部の直径のほぼ３倍なので[2)]、

$$\sigma_i=3\sigma_e \qquad (3)$$

とする。また、A の値は Linsenmeier ら[4)]の生理学的データから

$$A=2.13/\sigma_e^{1.23} \qquad (4)$$

とした。

(2) ＲＧＣの受容野密度

中心窩付近では神経節細胞の細胞体はその受光部からずれているので、神経節細胞の細胞体密度（個／deg^2）は、錐体の密度のように網膜部位の情報処理能力と必

ずしも対応しない。Drasdo[5] は、細胞体のデータから、網膜情報処理能力を示す神経節細胞の受容野密度 D_r（１平方度deg^2 あたりの受容野の数）を推測する式を計算した。それによると $\sqrt{D_r}$ と偏心度 E(deg) の関係は

$$\frac{1}{\sqrt{D_r}} = 0.0055(1+0.59E) \qquad (5)$$

で近似できる。(5)式を積分すると1.5×10^6 となりほぼ全神経節細胞数と一致する。この受容野密度には、Ｘ型、Ｙ型、Ｗ型のすべての細胞が含まれている。

(3) Ｙ型ＲＧＣの受容野密度

我々がパルス刺激を用いて測定した受容野(2)式 は、いくつかの理由でＹ型ＲＧＣの受容野であると考えられる。そこでＴＡＭＩＴではＲＧＣ全体の受容野密度 D_r ではなく、Ｙ細胞の受容野密度 D_Y を考慮せねばならない。この点に関しPerryら[6] は、ＨＲＰ注入法によって、サルのＹ型ＲＧＣの細胞密度をしらべた結果、9.4° 以上の網膜偏心度においては 偏心度によらずＲＧＣ全体の約10％であることを見い出している。網膜周辺部では細胞密度は受容野密度に一致している。それゆえ周辺部のＹ型ＲＧＣの受容野密度 D_Y は上記のDrasdoの式から求めた D_r の10％であるとする。

6.4 並列処理計算機ＮＣＵＢＥにおけるシミュレーション

計算機シミュレーションでは、各細胞の受容野が同心円上に分布しているものと仮定して行った。同心円の間隔Δｒは各位置における受容野密度の逆数に等しいとした。すなわち、

$$\Delta r(E) = 1/\sqrt{D_r(E)} \qquad (6)$$

ここに、E は網膜偏心度である。また、各受容野はほぼΔr で同心円上に等間隔に位置すると仮定した。座標 (x_0, y_0) にある細胞の出力 $O(x_0, y_0)$は、

$$O(x_0, y_0) = \varphi\left\{ \int_{-\infty}^{\infty}\int S(x,y)\cdot G(x-x_0, y-y_0)\, dxdy - \theta_n \right\} \qquad (7)$$

で表わすことができる。ここに S(x,y) は刺激パターンの輝度、G(x,y) は式（１）で表わされる受容野の感度分布である($r = \sqrt{x^2 + y^2}$)。

また、φは次式で与えられる非線形関数である。

$$\varphi(x) = \begin{cases} x & (x \geqq 0) \\ 0 & (x < 0) \end{cases} \qquad (8)$$

次にすべてのＹ型ＲＧＣの出力の総和 GO を計算した。すなわち、

$$GO = \sum_{x,y=-\infty}^{\infty} O(x,y) \tag{9}$$

ここにO(x,y)は各細胞の出力値である。

最後に全活動度 GO を心理的閾値θ_pと比較し、GO≧θ_pのときに光が検出されると仮定した。本シミュレーションでは、スポット光に対する閾輝度の決定を行なったので、閾輝度での S(x,y) は直径 d、高さΔIの円柱関数になる。したがって、シミュレーションでは

$$GO = \theta_p \tag{10}$$

となるΔIを計算することになる。すなわち

$$\theta_p = \sum_{x_0,y_0=-\infty}^{\infty} \varphi \left\{ \Delta I \cdot \iint_{(x,y)\in S} G(x-x_0,y-y_0)dxdy - \theta_n \right\} \tag{11}$$

なるΔIを求めることが目的である。特に$\theta_n=0$ のときは、

$$\theta_p = \sum_{x_0,y_0=-\infty}^{\infty} \Delta I \cdot \varphi \left\{ \iint_{(x,y)\in S} G(x-x_0,y-y_0)\ dxdy \right\} \tag{12}$$

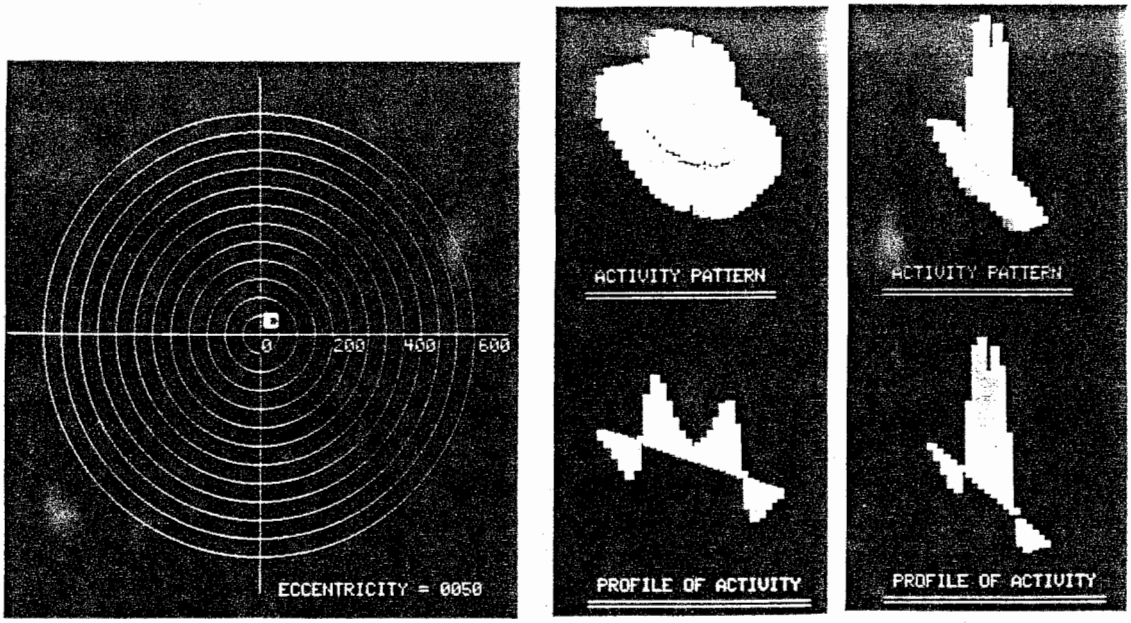
よって

$$\Delta I = \theta_p \Big/ \left[\sum_{x_0,y_0=-\infty}^{\infty} \varphi \left\{ \iint_{(x,y)\in S} G(x-x_0,y-y_0)\ dxdy \right\} \right] \tag{13}$$

となる。

今回構成した網膜モデルでは、データの並列性や計算の複雑さからたった一つのスポット刺激に対するＲＧＣ群の反応を求めるのに、従来の計算機処理で数時間を費やすことになる。従って、網膜上のいろいろな位置に提示されたスポット光についてその反応特性を調べるような会話的シミュレーションが実行しにくかった。

本研究では、ハイパーキューブ型の並列処理計算機ＮＣＵＢＥを用い、一つのスポット光に対するＲＧＣ群の反応を 256個のプロセッサに振り分けて処理した。実際には、網膜中心窩からの偏心度によって描かれる同心円上のＲＧＣ群の反応を同一のプロセッサで処理するという方法を採った。すなわち、一つのスポット光で引き起こされるＲＧＣ群全ての反応は、偏心度をパラメータにして振り分けた複数のプロセッサで並列的に処理された。図２にシミュレーションの結果の一部を示す。図２はそれぞれ偏心度 50′，600′にスポット光が提示されたときのＲＧＣ群の反応である。



(a)偏心度 50′　　(b)偏心度 600′

上図の左の同心円は視野を示しカーソルでスポット光の提示位置を指定する（上図左には偏心度 50′ を指定した時のディスプレイ画面を示す）．右上のパターンはスポット光に対するすべてのＹ型ＲＧＣの出力を、右下のパターンは断面を示す．

図２．ＮＣＵＢＥによるシミュレーションの結果

6.5 モデルのパラメータの推定

表１に示したようにＴＡＭＩＴは８個のパラメータから成る。σ_eとσ_iは (2)式と (3)式で与えられる。A は (4)式で与えられている。心理的閾値は同一被験者の場合、網膜偏心度 E によらず一定である。また、Q，θ_n も E によらず一定であると仮定した。Drは既に述べたように(5)式 から計算できるが、まず E=10(deg)においては(5)式 から得られる D_r の10％が Ｙであるとした。すなわち、

$$D_Y = 0.1 \cdot D_r \quad (E=10) \qquad (14)$$

まず、中心窩すなわちE=0(deg)において、付近のＲＧＣの受容野密度が一定であると仮定しその密度を変えて閾値－面積曲線の変化を調べた。結果を図３に示す。

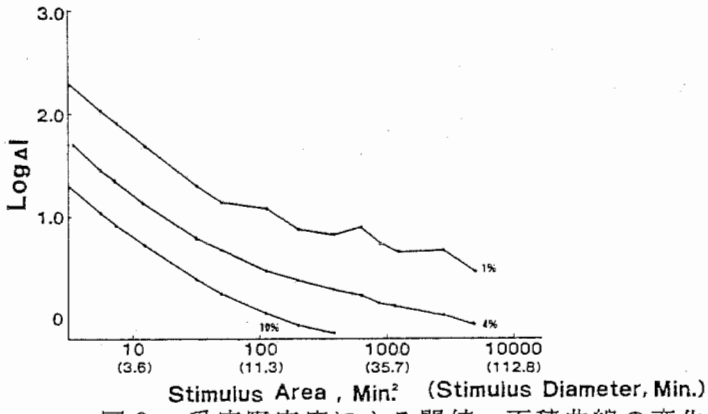


図３．受容野密度による閾値－面積曲線の変化

ＲＧＣの受容野密度は、閾値－面積曲線を上下に移動させる効果をもつ。次に、$D_Y=0.04\cdot D_r$として、すなわち、Ｙ型受容野が全体の４％であると仮定して、ＲＧＣの閾値の効果を調べた（図４）。

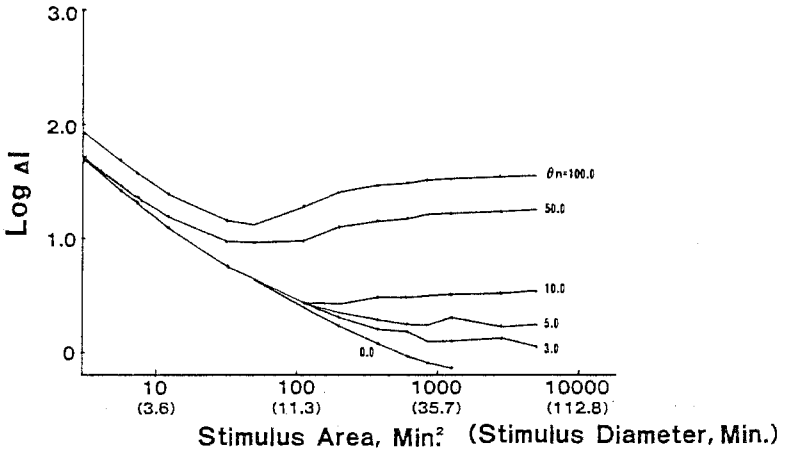


図４．閾値－面積曲線に対するＲＧＣの閾値の効果

θ_n が小さいときは、刺激光の直径が12′以上で閾値が上昇する。また抑制の強さ Q を大きくすると、刺激光の直径が 4′以上で、閾値が上昇する（図５）。

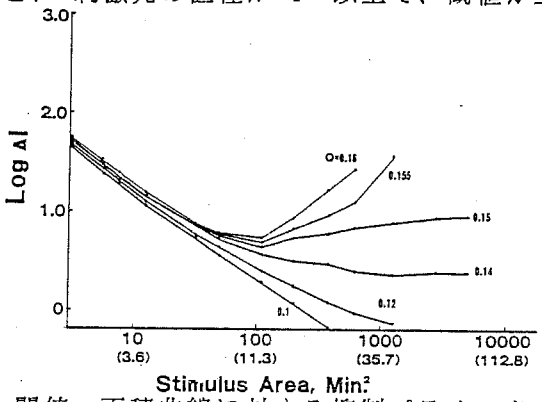


図５．閾値－面積曲線に対する抑制パラメータの効果

被験者 H.M. の閾値－面積曲線から以下のようなパラメータの値が推定された。Ｙ型ＲＧＣ受容野に割合を表２に示す。

Eccentricity	0°	1°	2°	4°	5°	10°
Percentage of Y-cells	4	8	9	10	14	10

表２．推定されたパラメータ値

Y型受容野の割合は、網膜偏心度が 3° 以上であれば、約10％で一定であるが、それ以下の偏心度ではその割合は減少し中心窩では 4％であった。その他のパラメータ値は、θ_n=2.0，θ_p=480，Q=0.12 であった。 これらのパラメータ値によってデータにあてはめた結果が図６である。推定された密度分布から網膜偏心度が 3° 以内であれば全体の 8.8％が、1° 以内であれば 全体の 8.0％がＹ型受容野であることが予想される。

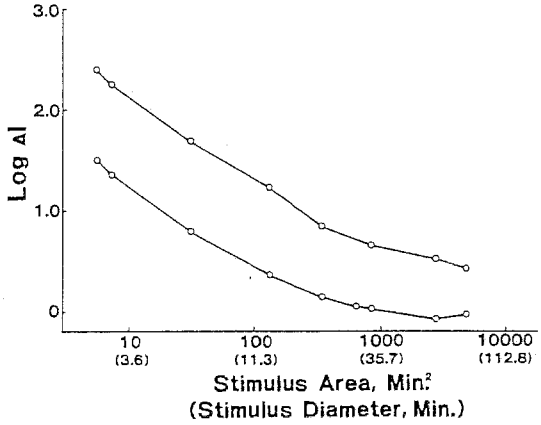


図６．ＴＡＭＩＴによる閾値－面積曲線の予測値

6.6 Ｘ型ＲＧＣの受容野密度と視力

最小分離閾（ＭＡＲ）は、受容器である錐体間距離によっては説明できない。また順応水準や視標の輝度によってもＭＡＲや視力（ＭＡＲの逆数）が変化するが、これも錐体のレベルでは説明できないことが知られている。視力のさまざまな側面は、ＲＧＣの水準で説明可能であろう。本節では上記のモデルから推定されるＸ型ＲＧＣの密度と視力の関係をしらべモデルの妥当性を示す。前節では、偏心度10°の位置でのＹ型チャンネルの受容野数が全体の10％であると仮定して偏心度10° 以下でのＹ型チャンネルの受容野の割合を推定した。この結果に基づき、どの偏心度においてもPerry ら[6] のデータが示唆するようにＹ型とＷ型が同数であるとして、推定されたＸ型ＲＧＣの受容野の密度を表３に示す。

Estimated receptive field density

Eccentricity	0°	1°	2°	3°	4°	5°	10°
Y-cell	0.37	0.29	0.17	0.12	0.08	0.08	0.019
X-cell	8.44	3.05	1.59	0.96	0.65	0.43	0.15
total	9.18	3.63	1.93	1.20	0.81	0.59	0.19

(min^{-2})

表３．推定された受容野密度

X型ＲＧＣの受容野密度 D_X は、信号の標本化密度なのでShannon の標本化定理を用いて処理できる遮断周波数 f_c が計算できる。すなわちX型ＲＧＣが六方格子状に配列されているとすれば、ＲＧＣ間の距離$\Delta\phi$は、

$$\Delta\phi = (2/(\sqrt{3}D_X))^2 \qquad (15)$$

で与えられる。このとき、遮断周波数 f_c は

$$f_c = (\sqrt{3}\cdot\Delta\phi)^{-1} \qquad (16)$$

である。しかし、X型ＲＧＣには、ＯＮ－中心型（正の刺激に反応）とＯＦＦ－中心型（負の刺激に反応）があるので、上記の D_X をさらに 1/2倍した条件（ＯＮ－ＯＦＦ Separate ）でも f_c を算出した。次に遮断周波数 f_c から視力を次式によって算出し、干渉縞視力(interference fringe acuity)のデータ[7] と比較した（図７）。

$$V.C. = 2f_c \qquad (17)$$

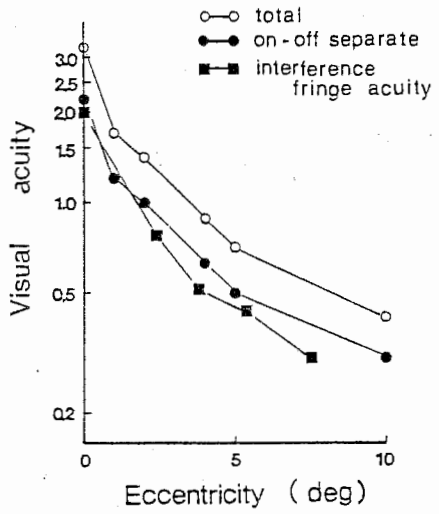


図７．干渉縞視力とモデルの予測値

その結果、ＯＮ－ＯＦＦ separate 条件で算出した視力 V.C. と干渉縞視力がよく一致することが明らかにされた。ＯＮ中心型ＲＧＣとＯＦＦ中心型ＲＧＣとの対で正負の信号が検出されるのでＯＮ－ＯＦＦ separate 条件が信号検出の意味で合理的である。図８は輝度を変化させて測定した視力[8] とモデルから予測される視力（黒三角）を示している。以上の予測値は視力の最高の値であり、輝度変化による視力低下は主として受容野の帯域通過特性が変化することによるものと考えられる。

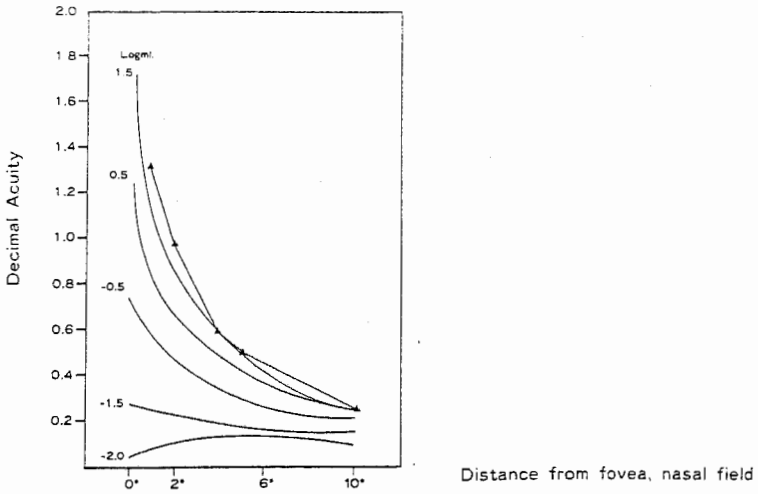


図８．視力の実測値（曲線）とＴＡＭＩＴと予測値（黒三角）

6.7 考察

本研究では、網膜モデルＴＡＭＩＴを通して、心理物理学的データと単一神経細胞の出力との関係が詳細に検討された。すなわち、心理物理学的に得られた閾値は、Ｙ型ＲＧＣの全活動度によって決定されることが示唆された。

つぎに推定された抑制の強さ Q の値は 0.12であった。関数 $\exp(-r^2/\sigma^2)$の体積はσ^2 に比例する。一方、受容野周辺部の半径σ_i は受容野中心部の半径σ_e の３倍であった。それゆえ、Q の値が 0.11 のとき、２つのガウス関数は同一の体積となり入力刺激に対する直流成分は遮断される。また(1)式 を Fourier変換すると

$$g(\omega) = \frac{\sqrt{\pi}A}{2} \{ \sigma_e \cdot \exp(-\sigma_e^2 \omega^2) - Q \cdot \sigma_i \cdot \exp(-\sigma_i^2 \omega^2) \}$$

となり

$$g(0) = \frac{\sqrt{\pi}A}{2} (\sigma_e - Q \cdot \sigma_i)$$

なのでやはり直流成分が遮断されていることがわかり、帯域通過チャンネルになっていることが理解される。網膜の情報処理能力を理解するためには、ＲＧＣ受容野特性と受容野密度の両方を明らかにする必要がある。受容野密度は信号の標本化頻度(sampling frequency)に対応する。人間のＲＧＣの受容野密度は、既に述べたようにDrasdo[5] により理論的に予測されている。しかし、Ｘ、Ｙ，Ｗのそれぞれについての密度分布は明かではない。最近、比較的周辺部においてＹ型細胞がＲＧＣ細胞の10％を占めていることが明かにされた(Perryら[6] ；Fukudaら、準備中)が、より中心部での密度分布はいまだ不明である。さらに、中心部では細胞体の位置と受容野の位置が異なるため、受容野密度を解剖学的に知ることは不可能である。我々は心理物理学的実験データに網膜モデルＴＡＭＩＴを適用することによって、Ｙ型ＲＧＣの受容野密度を推定した。その結果、網膜偏心度が3°以内の中心部ではＹ

型の割合は低下し、平均すると8.8%であった。Fukuda(準備中) は、中心部に受容野をもつＲＧＣの軸索の太さからおよそ 7%がＹ型であると推定している。またモデルから推定されるＸ型ＲＧＣの受容野密度から推定される中心外視力がこれまでに報告されているデータときわめてよく一致することが明らかにされた。これは我々が提案した網膜モデルＴＡＭＩＴの妥当性を実証しているものと考えられる。

6.8 文献

(1) Inui,T.,Mimura,O.& Kani,K.:Retinal sensitivity and spatial summation in the foveal and parafoveal region, Journal of the Optical Society of America,71,151-154(1981)

(2) 乾敏郎、三村治、可児一孝:点刺激による空間加重領域ならびに抑制領域の検討：網膜偏心度との関係,日本基礎心理学会誌,第１巻,pp.77-84(1982)

(3) Okamoto,Y.,Mimura,O.,Kani,K.,Inui,T.:Characteristics of two systems of human vision using fundus perimetry. In: E.L.Gleve(Ed.) Documenta Ophthalmologica Proceeding Series, Dr.W.Junk Publishers(1986)

(4) Linsenmeier,R.A.,Frishman,L.J.,Jakiela,H.G.,Enroth-Cugell,C.:Receptive field properties of X and Y cells in the cat retina derived from contrast sensitivity measurement,Vision Research,22,pp.1173-1183(1982)

(5) Drasdo,N.:The neural representation of visual space. Nature,266,pp. 554-556(1977)

(6) Perry,V.H.,Oehler,R.,Cowey,A.: Retina ganglion cells that project to dorsal lateral geniculate nucleus in the macaque monkey, Neuroscience, 12,pp.1101-1123(1984)

(7) Campbell,E.W.,Green,D.G:Optical and retinal factors affecting visual resolution, J. Physiol.,181,pp.576-593(1965)

(8) Slon,L.L.:The photopic acuity-luminance function with special reference to parafoveal vision, Vision Res., 8, pp.901-911(1968)

7．エネルギー最小化による画像復元とエネルギー学習

7.1 はじめに

最近、神経回路網モデルを現実的な情報処理問題に応用する試みが注目されている。このきっかけとなったのは、Hopfield & Tank[1]の巡回セールスマン問題の研究であった。この論文で示されたように、複雑な非線形拘束条件の下である評価関数（エネルギー）を最小（大）化する最適化問題は、問題のサイズが大きくなったとき、神経回路モデルの並列構造に問題の巨大なサイズをうまくマッピングできれば、神経回路が得意とする情報処理課題の一つとなる。画像処理や視覚情報処理の分野でも、いくつかの問題が大規模な最適化問題とみなせることが指摘され、視覚情報の局所並列性を生かした、並列アルゴリズムや、神経回路モデルが提案されてきた。しかしこれらの具体的問題において、評価関数（エネルギー）やそれに対応する神経回路モデルの構造とシナプス結合荷重は、天下り的に実験者によって与えられてきた。これは脳のモデルとしても不自然であるし、工学的な応用においても大きな障害となる。

宇野、川人、鈴木[2]は、ヒト上肢多関節運動における軌道生成の問題が、ある評価関数を最小化する問題であることを示した。さらに我々[3],[4]は、この評価関数を運動の例から学習によって獲得し、しかも回路の並列演算によってこの評価関数を最小化できる４層神経回路網モデルを提案した。

本章では、画像復元や初期視覚の問題について、軌道生成の神経回路モデルと同様に最小化すべきエネルギーを画像の例から学習によって獲得し、さらに回路の並列演算でそのエネルギーを最小化する神経回路モデルを提案する。

7.2 初期視覚・画像復元と神経回路モデル

7.2.1 初期視覚

ヒトの視覚情報処理の第一段階は初期視覚と呼ばれており、その目的は、網膜像（２次元画像）から実世界（３次元物体）の様子を推測することである。すなわち初期視覚とは、２次元の原画像の強度を決定している４つの要因、(1)可視表面の幾何学的構造、(2)可視表面の反射率、(3)光景に対する照明、(4)観察点を、２次元画像情報をもとに分離表現することである[5]。Poggioら[6]は、この過程が、三次元物体から二次元の濃淡画像への写像である光学のちょうど逆になっているので、初期視覚を逆光学と呼び、それが数学的には不良設定問題であることを示している。従って画像データ以外にも何らかの拘束条件がなければ、(i)解が存在しない、(ii)解が一意に定まらない、(iii)解がデータに対して連続とならない、のうちいずれかが生じてしまう。

Poggio & Koch[7]は適当な拘束条件を与えて、結果として得られる最適化問題を標準正則化理論を用いて解くとともに、並列神経回路モデルも提案した。さて標準正則化理論では評価関数は二次形式として与えられる。従って標準正則化理論によって提供される作用素は線形であり、並列神経回路モデルも線形のものに限られる。

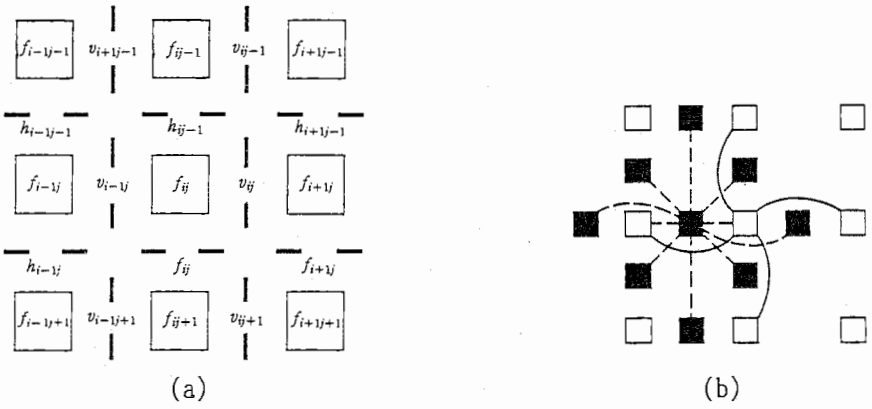
このような理論でも、心理物理学のいくつかの現象を再現したり、コンピュータビジョンのために有効なアルゴリズムを与えることはできるが、画像に不連続性を含む本質的に非線形の問題を取り扱うことはできない。

7.2.2 マルコフ確率場とライン過程

Geman & Geman[8]は、1984年に少なくとも４つの点で画期的な進歩をもたらした論文を発表した。彼らは、初期視覚の問題ではなく、ノイズに汚されたデータから原画像を推定し復元する画像復元の問題を取り扱ったが、その基本的なアイデアが初期視覚の問題にも適用できることは、この論文の後を追う MIT や Brown 大学グループの数々の論文[6],[9],[10],[11] からも明かであろう。彼らは第１に画像のマルコフ確率場モデルが Gibbs 分布のエネルギーを定義することと等価であることを示した。第２に、ライン過程と呼ばれる画像の不連続性を表現する隠れた（仮想の）確率変数を導入した。第３に画像復元の問題を、最大事後確率を与える原画像を推定することととらえて、対応するエネルギー最小化が局所並列演算（確率緩和）によって実行可能であることを示した。第４にこの確率緩和に温度のゆっくりとした低下（シミュレートされた焼きなまし:Simulated Annealing）を導入することにより、エネルギー最小化が厳密に達成できることを示した。

7.2.3 神経回路モデルによる奥行き面の再構成

Kochら[9]は、Geman & Geman[8] のライン過程とマルコフ確率場の概念を応用し、Hopfield型の神経回路で、不連続性を本質的に含む初期視覚の問題を解いた。この神経回路は、両眼視によって雑音を伴う奥行きデータが格子状の計測点上にまばらに与えられているとき、滑らかな曲面を再構成する。物体の端に相当する部分では、奥行きデータが不連続になる。この不連続を誤って曲面化してしまわないように、ライン過程と呼ばれる隠れた確率変数を格子点の間に配置する。図１(a) の正方形が各画素でのデータ、その間にある線がライン過程を示す。



(a)格子状に配列した奥行きデータと不連続を表すためのライン過程
(b)奥行き（白四角）とライン過程（黒四角）を表すニューロン間の相互作用

図１．初期視覚の問題を解決する神経回路モデルの構造

Geman & Geman は「焼きなまし(Simulated Annealing)」 による確率緩和でエネルギー最小化を達成したが、 Koch らは神経回路の並列演算でエネルギーの最小化を達成した。これは計算時間の点で確率緩和よりはるかに優れている。

図1に示す神経回路モデルを簡単に説明しよう。格子点上での奥行きデータの推定値を f，観測により得られた奥行きデータを d，格子点の間に配置するライン過程で水平に並んだ奥行きデータの不連続を示すものを v，垂直に並んだデータの不連続を示すものを h とする。f は実数、h,v は、0 と 1 の間の数とし、1 のとき、完全な不連続を、0 のときには完全な連続を表している。変数の定義:(1)式、エネルギーの各項の定義:(2)-(5)式、及び変数の時間変化則:(6)式を以下に示す。

$$h_{ij}=g(m_{ij})=\frac{1}{1+\exp(-2\lambda m_{ij})} \tag{1}$$

$$E_I+E_D=\sum_{ij}(f_{ij+1}-f_{ij})^2(1-h_{ij})+C_D\sum_{ij}(f_{ij}-d_{ij})^2 \tag{2}$$

$$E_L=C_U\sum_{ij}h_{ij}(1-h_{ij})+C_P\sum_{ij}h_{ij}h_{ij+1}+C_C\sum_{ij}h_{ij}$$

$$+C_L\sum_{ij}h_{ij}[(1-h_{i+1j}-v_{ij}-v_{ij+1})^2+(1-h_{i-1j}-v_{i-1j}-v_{i-1j+1})^2] \tag{3}$$

$$E_G=C_G\sum_{ij}\int_0^{h_{ij}}g_{ij-1}(h_{ij})dh_{ij},\quad g(U_i)=\frac{1}{1+\exp(-2\lambda U_i)} \tag{4}$$

$$E=E_I+E_D+E_L+E_G \tag{5}$$

$$\left.\begin{aligned}\frac{df_{ij}}{dt}&=-\frac{\partial E}{\partial f_{ij}}\\ \frac{dm_{ij}}{dt}&=-\frac{\partial E}{\partial h_{ij}}\end{aligned}\right\} \tag{6}$$

ここで，i,j は格子点の位置を表わす。簡単のために水平なライン過程に関する式のみを示す。m_{ij} は、g というシグモイド関数を通して、h_{ij} を決める。これはニューロンの膜電位に対応する。

エネルギーの各項の意味は次の通りである。E_I は奥行きデータの連続性、E_D は計測されたデータの信頼性、E_L は不連続（ライン過程）の生じ方、 E_G は m が発散しないことをそれぞれ要求する。これらのエネルギー関数の各項の係数 C_D,C_U, C_P,C_C,C_L,C_G は、それぞれの項の重みを決定する係数である。次節で説明するＮＣ

ＵＢＥ上の実験では、Koch らの用いたパラメータとほぼ同じ値を用いた。これらの係数を決定するのはなかなか困難である。

この神経回路モデルのシミュレーションには、Ｎ個の格子点に対して、３Ｎ個のニューロンの状態変化をを表す複雑な非線形常微分方程式を解く必要がある。図１(b) に奥行きを表わす格子点（白ヌキ四角）と、ライン過程を表わす格子点（黒四角）がどのように相互作用するかを示してある。これからわかるように１つのライン過程に対応する１個のニューロンは、周囲の１０個のニューロンと相互作用する必要があり、格子状に配列された画像処理専用の並列計算機では結合が足りなくなることがわかる。

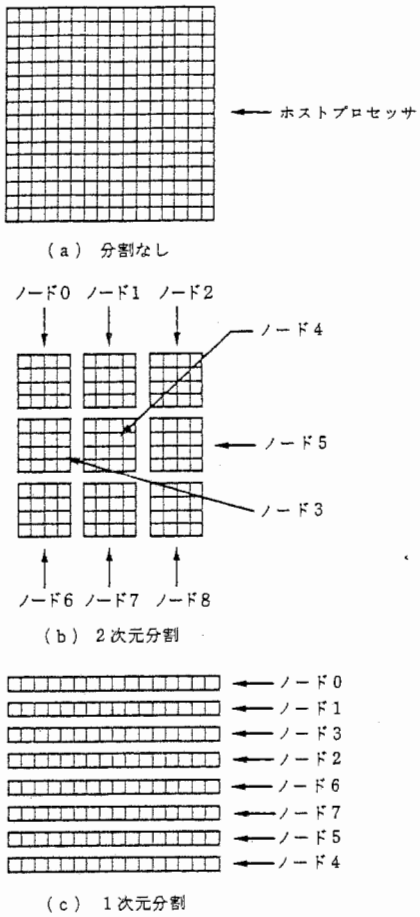
7.3 ＮＣＵＢＥ上での神経回路モデルによる画像復元

本節では 256個のノードを持つハイパーキューブ構造の強並列計算機ＮＣＵＢＥ／１０[12],[13] に前節の神経回路モデルをインプリメントし、画像復元の問題を解く。ＮＣＵＢＥ上のプログラミングの詳細は池田ら[14]を参照されたい。

まず、２次元に配列している１画素（m,h,v に対応する３個のニューロン）の処理を、おのおの１ノードに処理させるべくプログラムを行なったが（図２ｂ）、この方法では、ＮＣＵＢＥの能力がうまく引き出せなかった（表１参照）。原因は、二次元格子状に配列している画素の処理を、神経回路モデルのニューロンの昇順そのままにノードに割り付けたことにある。それぞれのノードは、受け持ちの画素と隣接する８つの画素を受け持つ８つのノードと密に情報交換する必要があるが、この通信が、ハイパーキューブ上で直接隣接しない２ノード間で頻繁に行なわれるため、ノードは、処理の大半の時間を他のノード間の通信の中継処理や、通信に必要なルートの空き時間待ちに費やしてしまい、計算速度が上がらないことがわかった。

分割方法	データ	実行時間	試行回数	相対速度
分割なし	32 * 32	12 秒	100	1.0
2 次元	32 * 32	19 秒	100	0.63
1 次元	256 * 256	91 秒	600	506

表１．ＮＣＵＢＥ上の神経回路モデルによる画像の復元実験のパフォーマンス比較



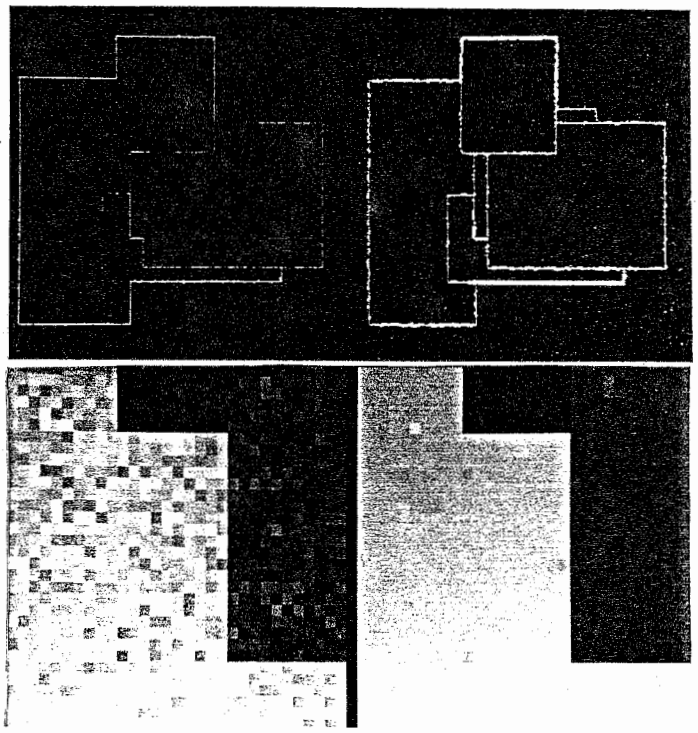
(a)すべての画素に対応するニューロンをホストで処理
(b)近傍の画素に対応するニューロンをまとめて１つのノードで処理（２次元分割）
(c)横１列の画素に対応するニューロンをまとめ１つのノードで処理（１次元分割）
図２．神経回路モデルのニューロンのＮＣＵＢＥのノードへの割り付けの様子

そこで、ハイパーキューブのリンクにうまく適合するようにモデルを移植するために問題の並列化から再度検討し直し、２次元に配列した画素の横方向の列を一まとめにしてノードに割り当てた（図２(c)）。このように分割すれば、隣接している画素の左右２つの画素は、同一のノードに含まれ、左上、真上、右上の画素は、上隣のノードに、左下、真下、右下の画素は下隣のノードに含まれる。そのため、密に通信するのは、上下のノードのみであることがわかる。つまり、ノードを１次元にリンクすれば、特定のノード間で行なわれる通信が他の通信の終了を待つことはない。このリンクは言わば、ハイパーキューブに沿った一筆書きであり、それはまた双方向にデータが流れるパイプラインを形成している。このリンクの構造は幾何学的で、ハイパーキューブの次元が大きくなると複雑化する。今回は、計算機プログラムでリンクを決定しそれを利用した。ノード割り付けの様子を図２下に示す。

今回採用した１次元鎖の拡張として、その両端が互いに接続している１次環、１

次鎖を平行に配置して鎖のなかの相対するノードをさらに鎖状に接続してできるメッシュ状のもの（２次鎖）、１次の場合と同様にして得られる２次環、ひいては、これらをｎ次にまで拡張した形態も、ハイパーキューブ構造の内部にノードを無駄にすることなくリンクできる。特に２次鎖構造は現在研究しているモデルのシナプス結合荷重の学習や、ライン過程を介して異種視覚情報を統合する場合に適する。

図３に示すように６つの濃淡値を取る幾何学的な人工画像(256*256) を考える。これに標準偏差 約 20 のホワイトな正規乱数を加える。画像の最大濃淡差は 150 であった。わずか 50ステップ程度（8秒の実行時間）で輪郭線を含めて原画像を再現できた（図３上、下）。



上段：ライン過程の状態変化．左上：(6)式に従う１回の更新の後の状態
右上：２００回の更新の後の状態
下段：濃淡値の状態変化．左下：ノイズに汚されたデータ
右下：(6)式に従う２００回の更新の後の状態
ただし全画像の中心部分のみ示した．

図３．雑音をともなう画像データからの不連続を考慮にいれた復元実験

計算時間については表１に示すようにホスト単独の 約 500 倍の計算能力が発揮された。ホストの能力が 0.5MIPS と低めに見積ってもこれは 250MIPS に相当し、ほぼＮＣＵＢＥの最大能力が引き出せるように、モデルをハイパーキューブ上で処理することができた。神経回路モデルと、並列計算機でのシミュレーションとの両面から満足できる結果がえられたといえよう。

7.4 エネルギー最小化とエネルギー学習を同時に行なう神経回路モデル

前節の実験で、原画像に加える正規乱数の分散を増やしていくと、本来画像の輪郭のない場所にライン過程が立つ（h_{ij}, v_{ij} = 1）ようになる（図3上のノイズのように見える過剰なライン過程に注意）。計測データの信頼性が下がれば当然(2)式の C_D を減らさねばならないし、不連続の少ない画像データに対しては (3)式の E_L を増やさねばならない。しかし これを実験者がいちいち決めていたのでは何のための画像復元かということになってしまう。 (2)-(4)式に現われるパラメータあるいはエネルギーの形は、画像のモデルあるいは物理世界の拘束条件として導入されたものなのだから、ある決められた画像のセットに対して一意に決まるはずである。エネルギーを学習によって神経回路モデル内に獲得できないだろうか。

f がニューロンの状態、d が画像復元の計測データを、C がエネルギーを特徴づけるパラメータを表わすベクトルとする。また $f^{\wedge}$ は画像の真の状態を示すとする。(1)-(6)式で与えられる神経回路モデルは、エネルギーを E(f,d,C)と表わせば、ほぼ E の最急降下方法に状態変化を起こし、画像の推定値 f^{*} として、

$$\frac{\partial E(f^{*},d,C)}{\partial f} = 0 \qquad (7)$$

を満たすものを与える。今画像の推定値と真の値の二乗誤差 $(f^{*} - f^{\wedge})^2$ を最小にするように、エネルギーを定義するパラメータ C を変化させることを考える。

$$\frac{\partial C}{\partial t} = - \frac{\partial (f^{*}-f^{\wedge})^2}{\partial C} \qquad (8)$$

まず陰関数の定理を用いて(7)式から $\partial f^{*} / \partial C$ を計算する。 f^{*}で E が極小であることと、E の主要項が E_D, E_G であるという仮定から、$(\partial^2 E(f^{*},d,C) / \partial f^2))^{-1}$ を正定数で置き換える。さらに神経回路の状態 f が、パラメータ C の学習時間のスケールではほとんど安定平衡状態 f^{*} の近くにいるという準静近似の仮定をすれば、結局パラメータ C の変化則として次式が得られる。

$$\frac{\partial C}{\partial t} = \varepsilon \frac{\partial^2 E}{\partial f \, \delta C} (f-f^{\wedge}) \qquad (9)$$

各ニューロンの状態遷移が次の方程式に従うことに留意すれば、

$$\frac{df}{dt} = -a(f) \frac{\partial E}{\partial f} , \quad 0 < a(f) \ll \infty \qquad (10)$$

結局、

$$\frac{\partial C}{\partial t} = -\varepsilon \frac{\partial}{\partial C} \left\{ \frac{df}{dt} \right\} (f-\hat{f}) \tag{11}$$

を得る。さらに(6)式を計算すればわかるが 一般に(11)式の df/dt は次のようにパラメータ C をシナプス荷重としたシナプス後電位の線形和となることがわかる。

$$\frac{df_{ij}}{dt} = \sum_{k} C_{ij}{}^{k} g_k(f_{lm}, h_{lm}, d_{ij}) \tag{12}$$

$$\frac{dm_{ij}}{dt} = \sum_{k} C'_{ij}{}^{k} g_k(f_{lm}, h_{lm}, d_{ij}) \tag{13}$$

g_k, r_k は非線形関数、添字 lm は エネルギー（あるいは神経回路の相互作用）が局所的であることから格子点 ij の近傍の値のみを取る。従って最終的に、エネルギーを定めるパラメータつまりシナプス結合荷重の学習則は、次式で与えられる。

$$\frac{dC_{ij}{}^{k}}{dt} = \varepsilon g_k(f_{lm}, h_{lm}, d_{ij})(\hat{f}_{ij}-f_{ij}) \tag{14}$$

$$\frac{dC'_{ij}{}^{k}}{dt} = \varepsilon r_k(f_{lm}, h_{lm}, m_{ij})(\hat{h}_{ij}-h_{ij}) \tag{15}$$

(12),(13)式のニューロンの状態の時間変化に比べて、(14),(15)式で与えられるシナプス結合荷重の変化が十分遅くなければいけないことをもう一度注意しておく。

さて(12)-(15)式 をもう一度検討すると、これはよく知られた異シナプス可塑性学習則となっている。つまり図４に示すように、１個のニューロンは、通常のシナプス入力 $g_1,\cdots,g_N$ をシナプス荷重 $C_1,\cdots,C_N$ で重み付けて、その状態を(12),(13)式に従って変化させている。

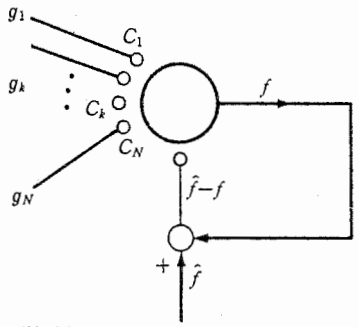
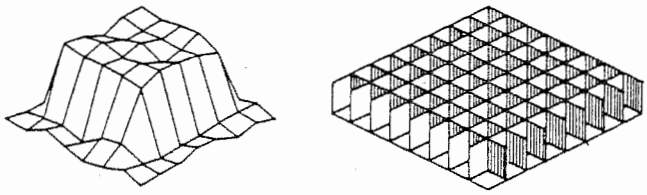


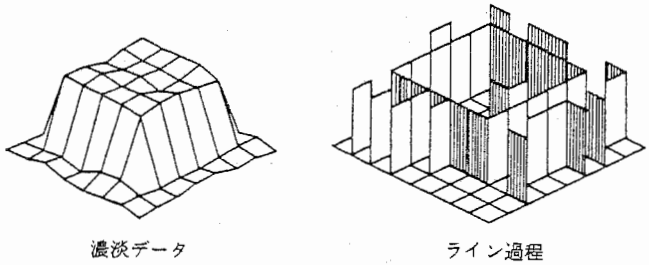
図４．エネルギー学習のためのニューロンの異シナプス可塑性
（ニューロンの状態が時間と共に連続的に変化する点が普通の学習則と異なる）

このニューロンは誤差信号 f^-f を第２の種類のシナプス入力として受け取っており、２種類のシナプス入力が同時に活動したときのみ、第１種のシナプス荷重だけが変化する。(14),(15)式 では画像の各点で、エネルギーを定義するパラメータ値が異なっていても良いようになっている。画像が等方的であれば、 (14),(15)式を ij について平均すればよい。結局、エネルギーの最小化を行いながら ((12),(13)) 、そのエネルギーをより画像のモデルとして適当なように学習によって変更する ((14),(15)) ことが一つの神経回路で達成できるのである。

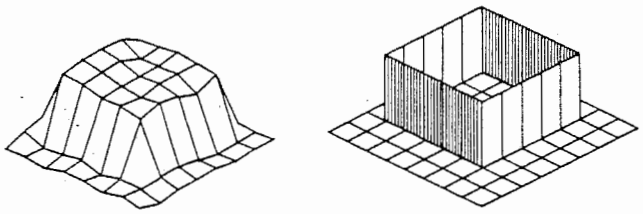
我々は、この学習則を、８×８の格子点上に与えられた簡単な幾何学図形（５×５の正方形：図５(a)）である２次元画像（最大濃淡差 200 ）のエネルギーを特徴づけるパラメータの学習に応用した。まず標準偏差 12 のホワイトな正規乱数を各格子点に加える（図５(a)左）。 神経回路の濃淡データ f の初期値としては、ノイズに汚されたデータをそのまま用いる。一方 ライン過程の初期値はすべて 0.5 とする（図５(a)右）。 エネルギーの形としては，式(2)から(5) のように選び、エネルギーを特徴付けるパラメータは、ＮＣＵＢＥ上のシュミレーションと同じとした。図５(b) に示すように、エネルギーの学習前には原画像で不連続のない場所にまでライン過程が立ってしまい、エネルギーのパラメータの値が不適当であることが分かる。これに対して約８０個の同じ統計的性質をもつ画像のパターンを用いて学習を行った後では、図５(c) に示すように、本来不連続のある場所にのみライン過程が立ち濃淡データも滑らかに再構成されていることが分かる。



(a)初期値



(b)学習前の画像復元



(c)学習後の画像復元

(a)濃淡データの初期値（左、生データ）とライン過程の初期値（右）
(b)学習前の神経回路モデルによる画像復元の結果
(c)エネルギーのパラメータを学習した後の神経回路モデルによる画像復元の結果

図５．簡単な２次元画像に対するエネルギー学習の実験

図６(a) に、標準偏差 10 のノイズを加えたときの３０個のパターンを用いた学習中のライン過程の誤差、濃淡値の誤差、エネルギーを特徴づける６個のパラメータの変化を示す。図６(b) は、ノイズの標準偏差が小さいとき（$\sigma = 8$）と大きいとき（$\sigma = 10$）とを比較すると、真の画像に加えるノイズの標準偏差が大きいときには、C_D は小さな値に収束することを示していて、エネルギー学習がうまく進んでいることが分かる。

ここで提案した学習則はエネルギーを特徴づけるパラメータの変化則として導出したが、図４を見れば、これはエネルギーの形そのものの学習に拡張できることがわかる。つまり、図７に示すように、入力層と出力層の二層からなる神経回路モデルがエネルギーの形そのものを学習し、さらに回路のダイナミクスにしたがって、獲得したエネルギーを最小化するモデルになっている。

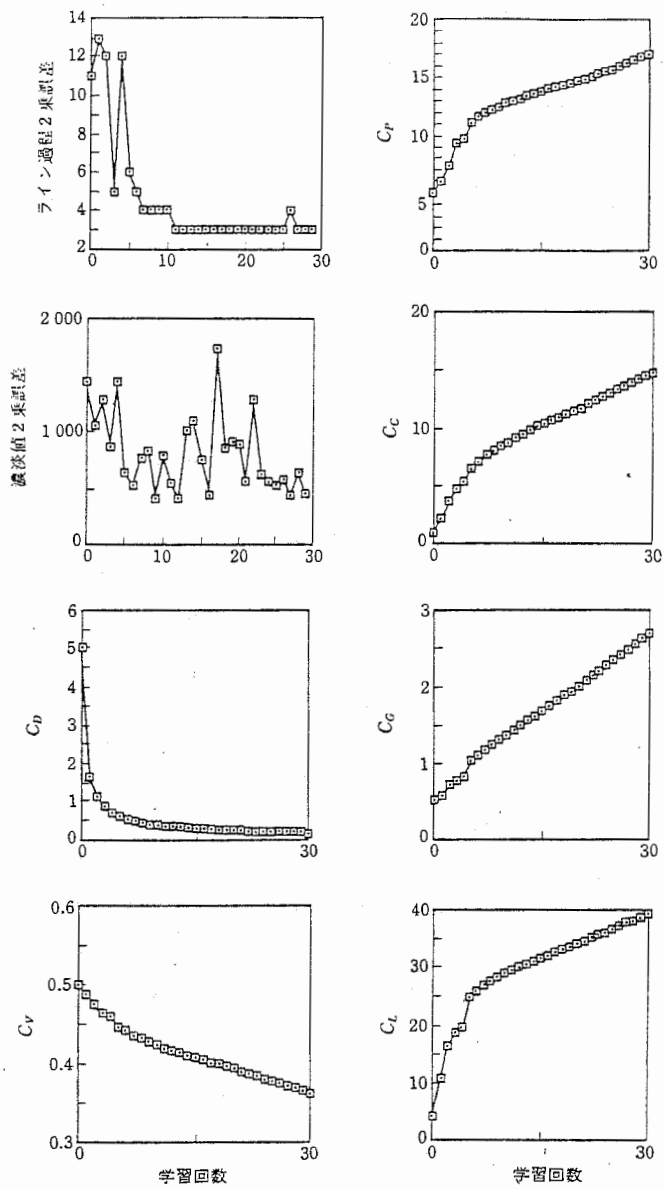


図6(a) エネルギーの学習中における、神経回路モデルが推定したライン過程自乗誤差、濃淡値の自乗誤差、エネルギーを特徴付ける6つのパラメータの学習回数に伴う時間変化

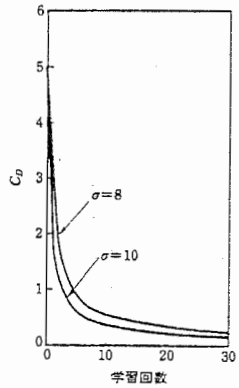
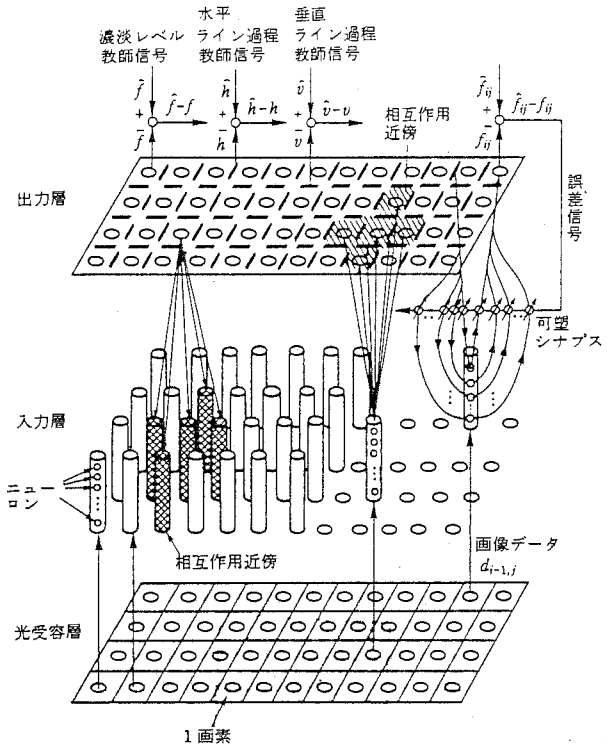


図6(b) 画像に加えるノイズの標準偏差によるパラメータ C_D の収束先の違い



２層とも、２次元画像の格子点に対応して、ニューロンがどの画素に対応するかが決まっている。入力層から出力層への結合も、出力層から入力層への結合も、画像モデル（エネルギー）の局所性から、その範囲は相互作用近傍に限られている。この神経回路モデルは、その状態をフィードバック結合によって連続的に変化させながら、シナプスの結合荷重を学習によって変えていく.

図７．エネルギーの形そのものを学習する２層神経回路モデル

7.5 おわりに

本研究では、初期視覚・画像復元問題で近来大きな進歩の見られた局所並列アルゴリズム、ライン過程を含むマルコフ確率場、神経回路モデルによるエネルギー最小化について紹介した。神経回路モデルを強並列計算機ＮＣＵＢＥ上にインプリメントし、画像の不連続性を含めた画像復元に成功した。またエネルギーの学習と最小化を同時に行える神経回路モデルを提案し、簡単な例についてパラメータ学習がうまく行えることを確認した。

初期視覚における線形アルゴリズム（フィルター）つまり二次形式のエネルギーを表現する行列を画像データから推定する試みが MIT[10] と ブラウン大学[11]で行なわれている。本論文で提案した神経回路は非線形のアルゴリズム、一般的なエネルギーを学習できるという点でこれらの研究より進んでいることを最後に強調しておく。

7.6 文献

(1) J.J.Hopfield, D.W.Tank:"Neural" Computation of Decisions in Optimization Problems, Biol. Cybern., 52, pp.141-152 (1985)

(2) Y.Uno, M.Kawato, R.Suzuki:Formation and Control of Optimal Trajectotory in Human Multi-joint Arm Movement -- Minimum Torque-Change Model, Biol. Cybern., in press(1989)

(3) M.Kawato, Y.Uno, M.Isobe, R.Suzuki:A Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics,IEEE Control Systems Magazine, 8, pp.8-16(1988)

(4) 前田,川人,宇野,鈴木:ヒト上肢多関節運動軌道を学習,生成する多層神経回路モデル,信学技報,MBE-87-133, pp.233-240(1988)

(5) D.Marr:Vision, Freeman(1982) 乾,安藤訳:ビジョンー視覚の計算理論と脳内表現ー,産業図書(1987)

(6) T.Poggio, V.Torre, C.Koch:Computational Vision and Regularization Theory, Nature, 317, pp.314-319 (1985)

(7) T.Poggio, C.Koch:Ill-posed Problems in Early Vision: from Computational Theory to Analogue Networks", Proc. R. Soc. London, Ser.B 226, pp.303-323(1985)

(8) S.Geman and D.Geman:Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images, IEEE Trans. Pattern Anal. Machine Intell., PAMI-6, pp.721-741(1984)

(9) C.Koch, J.Marroquin, A,Yuille:Analog "Neural" Networks in Early Vision, Proc. Natl. Acad. Sci. USA, 83, pp.4263-4267(1986)

(10) A.Hurlbert, T.Poggio:Synthesizing a Color Algorithm from Examples, Science, 239, pp.482-485 (1988)

(11) D.Kersten, A.O'Toole, E.Sereno, D.Knil, J.Anderson:Associative Learning of Scene Parameters from Images, Applied Optics, 26, pp.4999-5006(1987)

(12) 三宅:ニューラルコンピュテーション,信学会誌,70,pp.1255-1262(1987)

(13) NCUBE:NCUBE HANDBOOK v1.1 (1986)

(14) 池田,三宅,川人,乾,淀川,鈴木:初期視覚の計算理論と神経回路による解法,第２回生体･生理工学シンポジウム論文集,pp.137-140(1987)

(15) 川人,池田,三宅:神経回路の学習と視覚情報処理,テレビジョン学会誌,Vol.42,No.9,pp.918-924(1988)

8．ネオコグニトロンによる手書き数字認識[1],[2],[3]＊

8.1 はじめに

最近になって並列処理計算機という新しいタイプのマシンが計算機工学の分野で注目を浴びつつある。並列処理計算機は情報が並列的に処理されるような問題については非常に効果的に働く。ニューラルネットはそのような性質を持つ問題の一つと考えられよう。本章においては、並列処理計算機ＮＣＵＢＥに最も複雑なニューラルネットと言われるネオコグニトロン[4] をインプリメントした結果について述べる。ＮＣＵＢＥはハイパーキューブ状に結合された５１２個のノードプロセッサとフロントエンドのホストプロセッサ１個から成る。ノードプロセッサは 2MIPS, 0.5MFLOPS の計算能力を有する。 ＮＣＵＢＥは、コネクションマシン[5]のような「高粒度マシン」に対し、「低粒度マシン」の範中に位置づけられる。

8.2 ネオコグニトロンの性質

8.2.1 並列処理構造

図１にネオコグニトロンの構造を示す。ネオコグニトロンは入力層 U_0 と４段の処理ステージから成る。各段は２つの層（Ｓ細胞層とＣ細胞層）が縦続接続されて構成される。図１において、これら各層は細い線で描かれた長方形で示されている。各層は細胞面と呼ばれるいくつかのサブグループに分割されている。これら細胞面は太線で描かれた長方形によって図示されている。

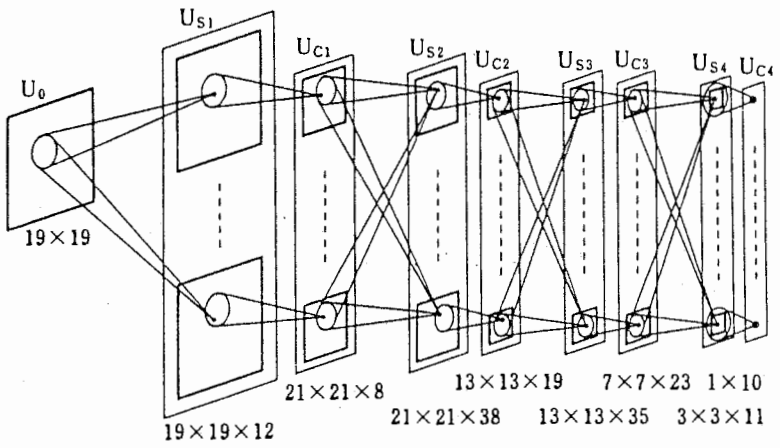


図１．ネオコグニトロンの構造

＊ ＮＨＫとの共同研究であり本章記述主要部分はＮＨＫ伊藤崇之氏の助力を得た．

S細胞の細胞面を「S面」、C細胞の細胞面を「C面」と呼ぶ。一つのS面は前段の多数のC面からの入力を受ける。一方、一つのC面は同一ステージ（段）の一つのS面からのみ入力を受ける。このS面とC面の対を「SCモジュール」と呼ぶ。

福島らはすでにネオコグニトロン[6]による手書き数字認識システムを構成した。各層における総細胞数を図1の各層の下に示した。

さて並列処理計算機のプログラムにあたっては、解くべき課題をどのようにして複数のサブ問題に分割できるかを知ることが重要である。問題の分割法の一つはデータを多くのサブデータに切り分けるやり方である。この方法は各サブデータが独立に同じオペレーションで処理されるような場合にのみ有効である。もう一つの方法は処理そのものを独立に実行できる多くのサブプロセスに分割する手法である。ニューラルネットにおいては、処理を分割するのにいろいろなやり方が考えられる。以下に述べる方法はこのような分割法のいくつかの候補である。どの方法が最良であるかは採用されるハードウェアの性質に依存する。今回のインプリメントにおいて、われわれは細胞面による分割を選択した。すなわち、一つのSCモジュールがNCUBEの一つのノード（SCノード）に割り当てられる。

(1) 細胞面による分割

巨視的観点から言えば、各細胞面は前層からのデータを受け取り、後続の層にデータを送る。この過程は細胞面毎に並列に実行される。

(2) 細胞による分割

一つの細胞面において、各細胞は独立に情報を処理する。このような分割は数万個のプロセッサを持つような「高粒度マシン」に適していると言えよう。

(3) シナプスによる分割

微視的観点からは一つの細胞の各シナプスが他の細胞からの一つの入力を受け取っておりそれぞれ独立に処理されていることがわかる。このような分割にはニューロチップが最も適したハードであろう。

つまりニューラルネットのような問題については多くのサブプロセスのデータが組み合わされそれが後続のプロセスに伝達される。通常この種のオペレーションは図2のような入れ子構造を成す。

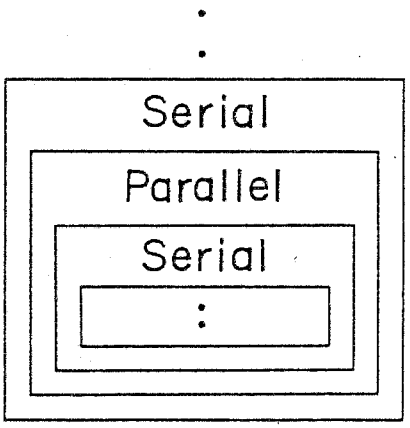


図2．処理の入れ子構造

8.2.2 ネオコグニトロンのその他の性質

ここで並列処理に関わるネオコグニトロンのその他の性質について議論する。

(1) 全てのＳ面は同一処理を実行する。Ｓ面の全細胞は同一の機能を有し、異なる点はその入力結合だけである。同じことがＣ面についても成り立つ。

(2) 第１ステージの細胞面の多くは各入力パターンについてゼロでない出力を出す。一方、第２ステージ以降の奥のステージではゼロでない出力を出す細胞面がわずかになってしまう。この違いは、各ステージにおいて特徴を検出するための受容野のサイズの違いに起因する。すなわち、初段のＳ面は特定方位の線分を検出するが、そのような線分特徴はほとんど全ての入力パターンが共通して持っている特徴である。一方、もっと奥のステージで観測される大局的特徴は各カテゴリーについて特異的なものであり、反応する細胞の数は少なくなる。

(3) 第１と第２のステージ間の結合は all-to-all の形態で形成される。すなわち、第２ステージのＳ面は第１ステージのほとんど全てのＣ面からデータを受け取っている。このような結合形態については、奥の層ではそれほど密集した形態にならない。つまり、一つのＳ面は前段の少数のＣ面からだけデータを受け取る。

8.3 インプリメント

8.3.1 ノードの機能

前述したように一つのＳＣモジュールは一つのノード（ＳＣノード）に割り当てられる。この方法は以下のような利点を有する。

(1) データ転送の単純化：ＳＣモジュールの処理は転送すべきデータ量は大きいが、ノード間のデータ交換は単純である。

(2) 拡張性：ＳＣノードの数は学習パターンや認識すべきパターンの変更に応じて増やすことができる。ＮＣＵＢＥに余ったノードがある限り新しいＳＣノードを容易に追加できる。

なお、ネオコグニトロンはＶ面と呼ばれる別のタイプの細胞面を持つ。Ｖ面は各ステージに一つずつ存在する。Ｖ面はもう一つ別のノードに割り当てられている。

8.3.2 データ転送

一つのＳＣノードは限られた数のリンクを通じて他の多くのノードからデータを受け取る。これはデータ転送における衝突を引き起こす。これを避けるために、われわれはブロードキャスティング[7] とギャザリングという手法を用いた。

図３に３次元ハイパーキューブにおけるブロードキャスティングの手続きを示した。ｎサイクルのデータ転送で、ルートノードのデータは、ｎ次元ハイパーキューブ内の他の全てのノードに伝えられる。データを集めるギャザリングは、図３のデータの流れを逆転することによって実現する。中間のノードは受け取ったデータを

自分自身のデータと組み合わせ、それを次のノードへ転送する。このようなアルゴリズムは、全てのノードが同期して働くという拘束条件があるにもかかわらず、データの衝突を防ぐ効果がある。転送時間が大きくならないために、もしデータがゼロの要素のみで構成されている場合にはＳＣノードからのデータ転送は省かれる。

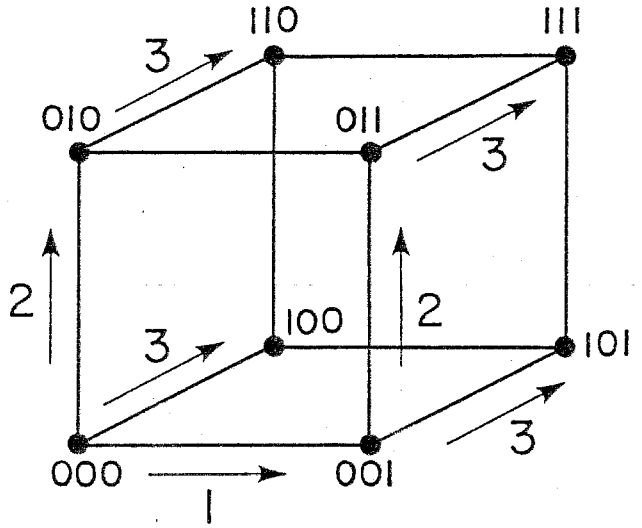
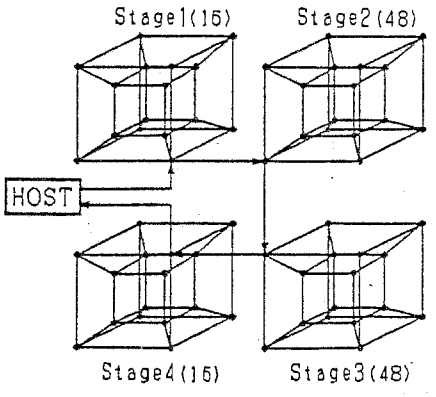


図３．３次元ハイパーキューブにおけるブロードキャスティング

8.3.3 結果

図４にノードの割り当ての概略を示した。各ステージの機能はＳＣモジュールの数を満足するだけのノード数を持つサブキューブによって実現される。Ｖノードはサブキューブのルートノードに割り当てられる。ルートノードはホストや他のＶノードとのデータの送受信を行なう。図１のシステムを実現するのに１２８個のノードを要した。したがって、５１２ノードのＮＣＵＢＥに４個のネオコグニトロンが同時に実行できる形でインプリメントした。



()内の数字は各サブキューブにおけるノードの数を示す

図４．ＮＣＵＢＥにおけるネオコグニトロンの構成

表１はパターン“８”を認識するに要する計算時間の一例である。通常“８”は最も長い計算時間を必要とする。同一パターンについて Micro VAXⅡでの計算時間との速度比較を表１に示した。図５に認識結果のＣＲＴ表示例を示す。

	NCUBE	microVAX II
time(sec)	0.95/4chars.	17.3
speed ratio	72.8	1.0

表１．ネオコグニトロンの実行時間

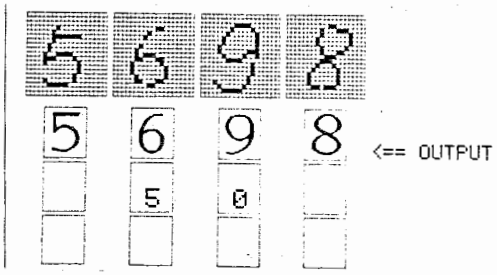


図５．認識結果のＣＲＴ表示の一例

8.4 考察

8.4.1 データ転送

ブロードキャスティングやギャザリングは多くの結合を持つネットワークのシミュレーションに有効であるが、各ステージで全データがＶノードに集中することが一つのボトルネックになっている。しかしながら、以下の評価によってこのことが無視し得ることがわかる。隣接するノード間のデータ転送時間を測定し、線形減衰評価値を算出した。すなわち、

（転送時間）＝1.1×（データカウント）＋298.1(μsec)

第１ステージからの出力データが最も大きいので第１ステージから第２ステージへのデータ転送時間を調べた。これがほぼ 4.2msecであり表１の計算時間に比べれば無視し得る範囲にある。

8.4.2 負荷配分

異なるＳＣモジュールは前段の異なるＳＣモジュール集合からデータを受け取る。前段のＳＣモジュールがゼロでない出力を出すかどうかは入力パターンに依存する。従って、一つのＳＣモジュールが受け取るデータ量はダイナミックに変動する。これはＳＣモジュール内に負荷の不均衡を生む。特に第３、第４といった奥のステージでの不均衡は大きい。一つの解決策としては次のような方法が考えられる。一つのステージにおけるＳＣモジュールはその前段のモジュールの類似性に従ってクラス分けできる。異なるクラスに属するＳＣモジュールは、それぞれの前段モジュールについて重複しない。ノードのクラスタがこれらのＳＣモジュールを共同して処理するなら、各ノードの負荷は入力パターンに依らずに平均化する。もう一つの解決策は、flood fill algorithm[8] のようなダイナミックなノード割り当てである。この方法においては、「管理者ノード」が「労働者ノード」に逐一データパケットを配分する。

8.5 おわりに

以上、ネオコグニトロンをＮＣＵＢＥにインプリメントした結果について述べた。現在以下のような課題を検討中である。

(1) 同時に処理する文字数を増やす
(2) テレビカメラから直接入力した文字を読み取る
(3) 認識対象をアルファベットに拡張する

8.6 文献

(1) 三宅誠,伊藤崇之,福島邦彦:神経回路モデルの並列処理－ＮＣＵＢＥによるネオコグニトロンの高速実行－,昭和63秋季信学全大講演論文集,p.D-1-9(1988)
(2) 伊藤崇之,福島邦彦,三宅誠:並列計算機によるネオコグニトロン型数字認識システムの実現,昭和63秋季信学全大講演論文集,p.D-1-10(1988)
(3) T.Ito,K.Fukushima,S.Miyake:Pattern Recognition and Parallel Processing:NEOCOGNITRON on NCUBE, 2nd International Symposium on Bioelectronic and Molecular Electronic Devices, R&D Association for Future Electron Devices, Dec.12-14, Fujiyoshida(1988)
(4) K.Fukushima:Neocognitron: A self-organizing multilayered neural network, Biol. Cybern., 36, No.4, pp.193-202(1980)
(5) W.D.Hillis:The Connection Machine,MIT Press(1985)
(6) K.Fukushima: Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition, Neural Networks, 1, No.1, pp.119-130(1988)
(7) H.J.Siegel, R.J.McMillen:The Multistage Cube: A Versatile Interconnection Network, IEEE Computer, 14, No.12, pp.65-76(1981)
(8) Parallel Fortran User Guide, 3L Ltd.(1988)

９．３層逆伝播学習ネットワークによる画像データ圧縮

9.1 はじめに

近年コンピュータ処理速度の向上を背景に、変換規則を自動的にサンプルデータから抽出する学習能力と、複雑な処理を高速に実現する同時並列処理性、の２つの特徴に着目したニューラルネットの工学的応用研究が進展しつつある。

ニューラルネット応用の一つに視聴覚パターン処理がある。これらの応用では、より小さな次元の特徴空間へ変換する機能（情報圧縮性）と、学習データから規則性を抽出して未知の類似データに対して学習データと同様の性能を得る機能（汎化性）が重要である[1]。

一方、画像圧縮技術は、再生画像の品質を維持しつつ信号の持つ冗長度を抑圧してデータを圧縮することを基本とする。これまで主に、輝度、色相等の信号レベルや画素の２次元空間分布に関する統計的冗長度を抑圧する符号化が検討されている。さらに人間の視覚・心理特性に基づく視覚的冗長度[2] や、送受信で共有する対象の構造モデル等の知識的冗長度[3] を抑圧する方式も研究されてきている[4]。

本章では、再生画像の２乗平均誤差を最小にするという拘束条件のみを与えたニューラルネットを用い画像のデータ圧縮を行い、ニューラルネットの情報圧縮と汎化能力について評価する。また、学習によって自動的に獲得した圧縮・再生規則の性質や、ネットワーク内でのデータ表現方法を検討し、新たな画像の符号化方式を探る。この種の研究には、逆伝搬学習則(Back-propagation)を用い単一ネットを画像に適用した検討[5] や、局所並列的に適用した検討[6] 等の報告がある[7],[8]。

今回は、３層の非巡回形ニューラルネットと逆伝搬学習則を用い、静止画像を対象としたデータ圧縮について報告する。しかしながら現在、ニューラルネットを構成するパラメータ、圧縮特性、記憶容量、学習の収束性、実現する写像関数の精度、汎化特性等をシステマティックに決定できるような設計原理は確立していない。このため、計算機シミュレーションによる実験的検討を行った。主な内容は、まず、画像データをニューラルネットで高速処理するプログラムの、超立体アーキテクチャを用いた汎用並列計算機（ＮＣＵＢＥ）上での実現方法、得られた性能について述べる。次に、これを用いた画像圧縮実験結果として、学習画像数と再生した学習画像のＳＮ比特性、未学習画像に対するＳＮ比特性、中間層出力の量子化に対するＳＮ比特性、重みの初期値に対するＳＮ比特性について述べる。

9.2 ニューラルネットによる画像圧縮モデル

9.2.1 システム・モデル

人工的ニューラルネットを用いて、画像信号の圧縮・再生を行う符号化方式を、ここではニューロ・コーデック(Neuro-CODEC) と呼ぶこととする。そのシステム・モデルの概略を図１に示す。ニューラルネットには、入力画像をより低次元のデータに変換して圧縮機能を実現させることが要求されるので、入出力層のユニット数に比して中間層のユニット数は小さくする。

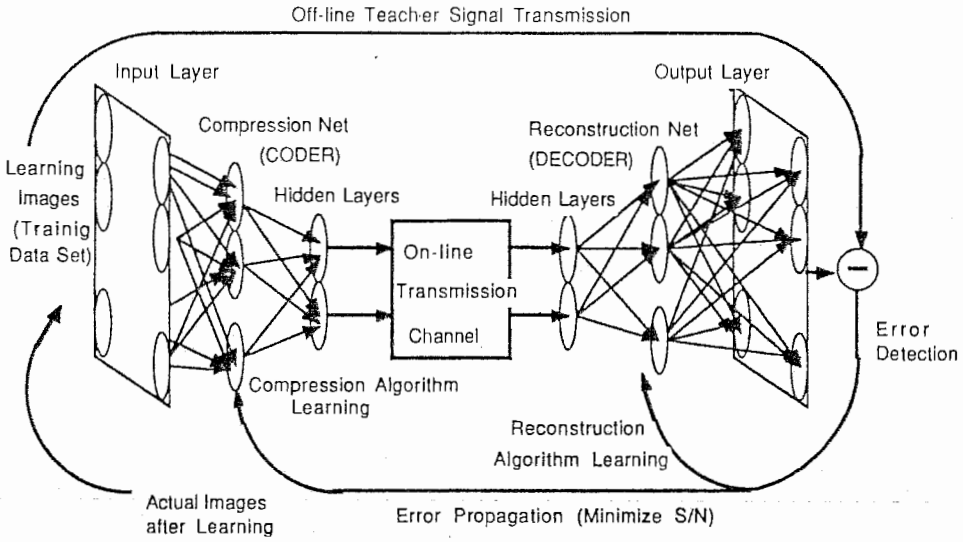


図１．Neuro-CODEC のシステム・モデル概要

学習過程では、学習すべき画像データを入力層に与え、出力層には教師信号として入力層と同一の信号を与える。ニューラルネットは、圧縮ネット(Compression Net)と再生ネット(Reconstruction Net)からなる。圧縮ネットは、サンプルデータから画像信号の持つ冗長度を抑圧する変換規則を学習する。再生ネットは、圧縮されたデータから原画像を再生する変換規則を学習する。

なお、通信過程については、ニューラルネットを中間層を中心にして、圧縮ネットと再生ネットに分離して回線の送受信符号器(CODER)、復号器(DECODER)として通信路を構成することになる。

9.2.2 ニューラルネットのモデル

実験に用いたニューラルネットの構成は、図２に示す３層の非巡回型ネットワークであり、学習則は誤差逆伝搬法を用いた[9]。

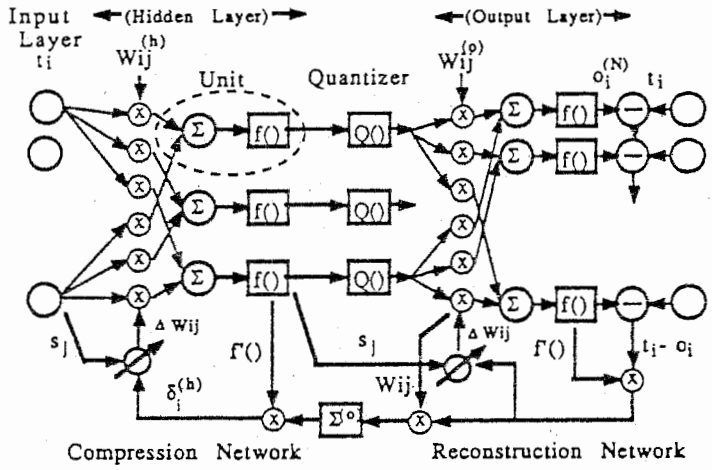


図２．ニューラルネットの構成

(1)ネットワーク構成

第n層 i番目のユニットの入力 $net_i^{(n)}$ は、重み係数 $w_{ij}^{(n)}$ と (n-1)層の j番目のユニット出力 $s_j^{(n-1)}$ を用いて、次式で示す線形結合演算により得られる。

$$net_i^{(n)}=\Sigma_i w_{ij}^{(n)}\ s_j^{(n-1)},\quad n\geqq 2 \tag{1}$$

ここで、$net_i^{(n)}$ のバイアス値は、$s_J w_{iJ}$ ($s_J=1$) として(1)式に組み込む。第n層の活性化関数としては次式のシグモイド関数を用いた。

$$o_i^{(n)}=f(net_i^{(n)})=1/[(1+\exp(-net_i^{(n)}))] \tag{2}$$

再生ネット出力 $o_i(N)$ と教師 t_i との誤差は次式で示す２乗誤差関数を採用した。

$$ERROR=\ (1/2)\Sigma_i (t_i-o_i^{(N)})^2 \tag{3}$$

(2)学習則

第(n)層での i番目のユニット入力値 net_i に対する誤差 ERROR の変化分 $\delta_i^{(n)}$ は次式となる。

$$\delta_i^{(n)}=d\ ERROR/d\ net_i^{(n)}=[s_j^{(n)}(1-s_j^{(n)})]\Sigma_i \delta_i^{(n+1)} w_{ij}^{(n+1)} \tag{4}$$

学習時に於ける重みの変化分は以下によって修正する。

$$\Delta w_{ij}^{(n)}=\alpha\ \delta_i^{(n)} * s_j^{(n-1)} \tag{5}$$

次に α , ε を重み修正利得係数、p をパターン数とすると重みの修正は以下となる。

$$\Delta w_{ij}^{(n),p}=\alpha\ \Delta w_{ij}^{(n),p}+(1-\alpha)\ \delta_i^{(n)} * s_j^{(n-1)} \tag{6}$$

$$w_{ij}^{(n)}=w_{ij}^{(n)}+\varepsilon\ \Delta w_{ij}^{(n),p} \tag{7}$$

ここでは、ε ,p = 1 とし１学習毎に $\Delta w_{ij}^{(n)}$, $w_{ij}^{(n)}$ を変更し、$\Delta w_{ij}^{(n)}$ は全学習での累積値を用いた。

(3)画像品質評価尺度

画像品質の評価尺度は、次式で示す規格化２乗平均誤差を用いた。

$$SNR=10\log_{10}[E\langle t_i^2\rangle/E\langle(t_i-o_i)^2\rangle] \tag{8}$$

ここで、E〈 〉は、画面全体での平均を示す。

9.2.3 並列計算機によるインプリメント

ニューラルネットは、並列・多層構造と最急降下法によるコネクション重みの適応処理を基本とするため演算量が膨大である。特に、画像を対象とする場合、ユニット数及びコネクション数の増加と長大データの入出力時間が大きくなり、長時間の計算を要する。このためシミュレーションには、超立体アーキテクチャの汎用並列計算機（ＮＣＵＢＥ）を用いた[10]。この種の並列計算機は、本来ノードプロセッサ間の通信が少ないことを前提に高性能が得られるように設計されている。

そこで、画像が離れた画素間での相関が少ない性質を用い、図３に示す局所並列形(Local-Parallel NN) を採用した。これは、画像をブロック分割し、ブロックごとに独立な小ニューラルネットを適用し、同時並列処理する。このため、小ニューラルネットは、顔画像であれば、目、鼻、髪、背景等の各部位の特徴に対応する圧縮・再生アルゴリズムを学習することになる。また、小ニューラルネット間での通信をなくすため、ブロックが重複した部分は複数の小ニューラルネットで処理する形式とした。

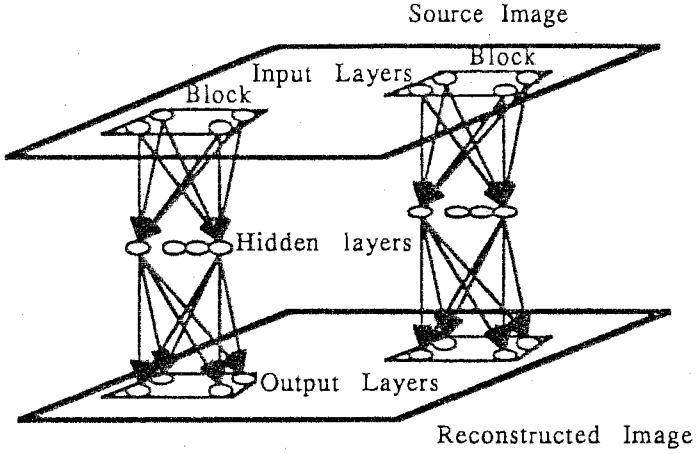


図３．画像圧縮のためのニューラルネット型情報処理

次に、このニューラルネットを実現する時、画像を蓄積するホストプロセッサとノードプロセッサ間でのデータ転送時間をなくし、ノードプロセッサの実効処理効率を向上させるため以下のノードプロセッサ機能配分を行った。図４に示すようにノードプロセッサを、①フォワード演算及びバックプロパゲイション演算を行うカーネル処理と、②プロセス制御、同期及び学習データの蓄積を行うフロント・エンド処理、とに分割するアーキテクチャとした。この方式においては、ニューラルネット処理時にノードプロセッサの局所メモリ（最大512kB）に画像データを常駐させ、学習用の画像ブロックデータをカーネル処理ノードに分配転送する。この結果、ノードプロセッサの稼働率約８０％を得た。

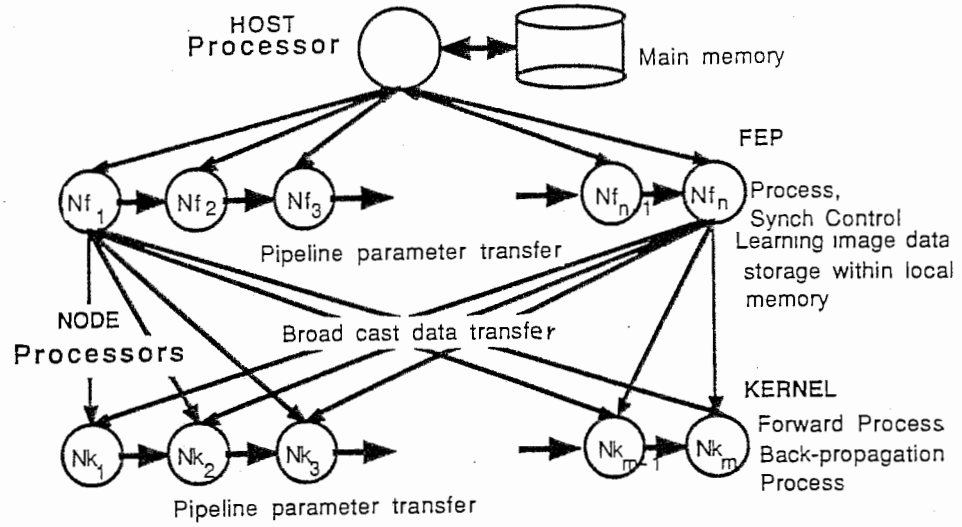


図４．ノードプロセッサへの機能割り当て

本方式の処理能力を図５に示す。図５は、３層の小ニューラルネット（ニューロン数 64x3、 コネクション数 64x64x2=8192 ）の処理を単一のノードプロセッサで実行する場合の処理時間を示している。信号の表現精度は、入出力画像が 128画素x128画素x8bits、ニューラルネット内部演算:4bytes 実数である。

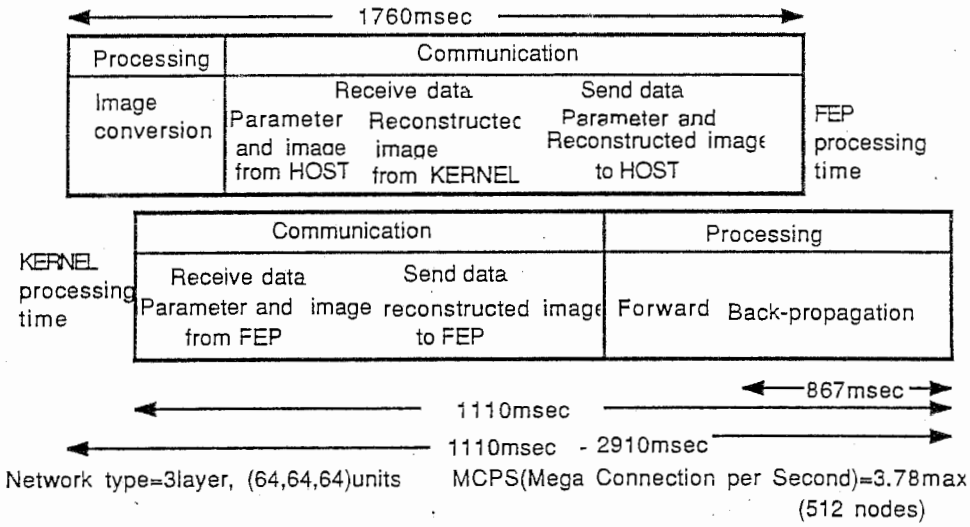


図５．一つのノードプロセッサにおける処理速度

処理能力については、５１２個のノードプロセッサを用いて、これと同数の小ニューラルネットにブロック分割した単一画像を同時並列で学習させる場合、 3.78 MCPS(Mega Connection Per Second) の性能を得ている[11]。

一方、学習毎に画像を更新する場合、フロントエンド・プロセッサからカーネル・プロセッサに画像ブロックデータが毎回転送される。この時の処理能力は、５１２個で 1.44 MCPSである。この結果、１枚の画像を約１秒で学習可能となり、ホス

トメモリから画像を毎回転送した場合に比べ約２０倍高速となった。

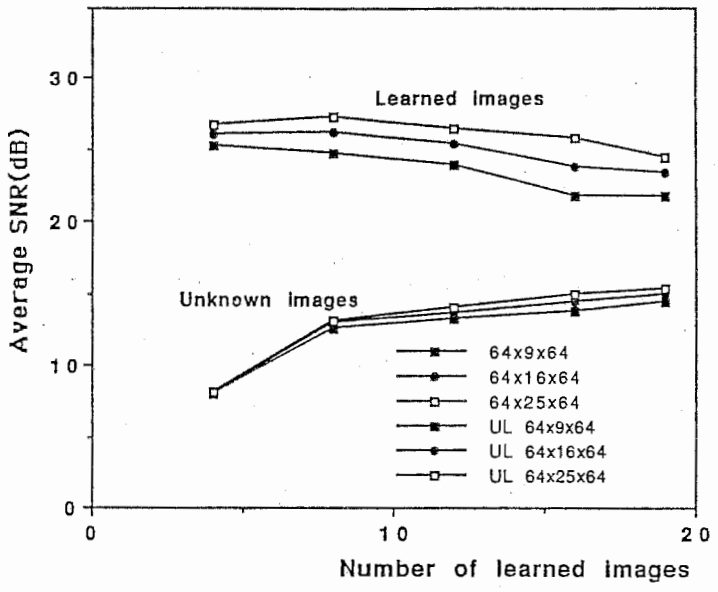
9.3 実験と考察

２節で述べた局所並列形ニューラルネットについてシミュレーション実験を行った。適用したニューラルネットは、３層の非巡回型ニューラルネットである。入力層と出力層のニューロン数(Ni)は、64であり、中間層を Ni より少ないニューロン数(Nh) で構成した。

(1)汎化特性

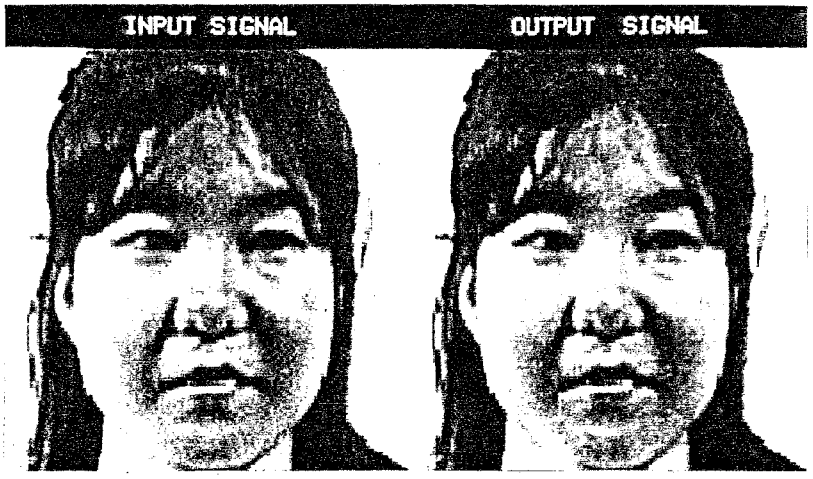
ニューラルネットが獲得した変換規則は、データ圧縮性とともに特定の学習パターンだけでなく類似の特性をもつ未学習パターンに対しても適用できる必要がある。

そこで、図６に示すように、学習データに対するＳＮ比特性と中間層のニューロン数の関係を求め、未学習画像に対するＳＮ比特性の傾向を求めた。学習回数は、各学習画像に対して 500回 であり （全体の学習回数は学習画像数 5-19 枚に対し 2500-9500回）、 学習後の再生画像のＳＮ比は、学習画像全体での平均値である。また、図７に学習画像と学習後の再生画像例を示す。学習画像数の増加に伴い、再生画像のＳＮ比は減少するが、平均ＳＮ比は、中間層のニューロン数 (32,42,52) が多いほどＳＮ比は高い。これは、ニューラルネットがパターン情報を重みベクトルの形で圧縮して記憶する能力に対応するものと考えられる。また、ＳＮ比の増加は、中間層のニューロン数の２乗のオーダの増加に対して約１～２dBである。

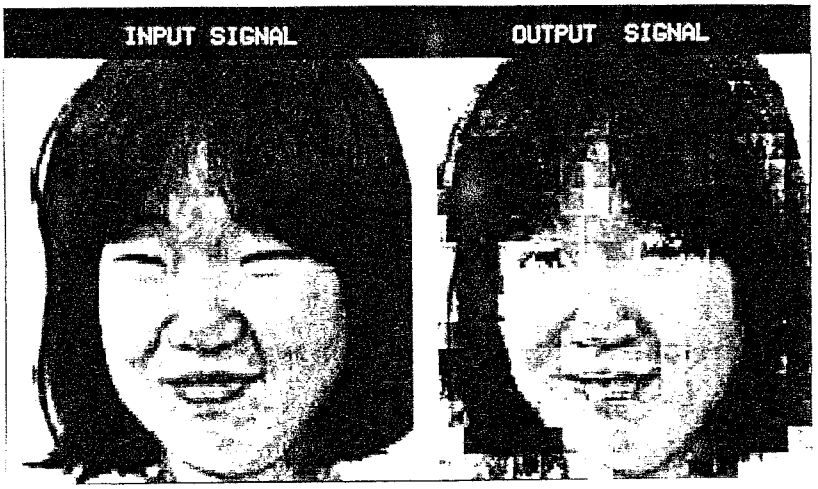


学習回数:500回／１画像　　量子化ビット: 8 bits

図６．ネットワークの汎化能力



学習画像とその再生例　SNR=23.74db



未学習画像とその再生例　SNR=15.55db

学習回数:500/1image，量子化ﾋﾞｯﾄ:8bits，回路:64×16×64(3層)，画像数:19

図７．再生画像の例

一方、未学習画像に対する平均ＳＮ比は学習画像数の増加に応じて向上する。図６に示した未学習画像に対するＳＮ比は、未学習画像３画面についての平均値であり、図７の下段に未学習画像とその再生画像例を示す。図７から分かるように再生画像にはブロック歪が現れ、ＳＮ比劣化要因となっている。未学習画像の平均ＳＮ比は、中間層のニューロン数 16 の時、学習画像数が 4 で 8.14dB、19 で 14.92 dBであった。学習画像１９枚を用い、 学習回数を 9500 から 38000 まで行った場合の再生画像の特性を図８に示す。学習画像に対しては上記の結果と異なり、ＳＮ比が 約 3.7dB 向上するが、未学習画像に対するＳＮ比は若干劣化する。これは、学習画像の特性にチューニングされ過ぎる過学習効果のためであると考えられる。

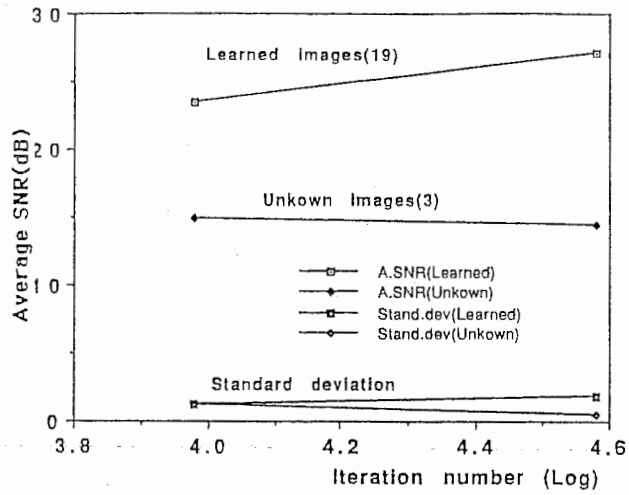


図８．学習回数に対する画像品質　初期値:ﾗﾝﾀﾞﾑ，回路:64×16×64，α=0.9

汎化性能を上げるには、直交変換等で次元を減らす必要がある。また、ブロック歪軽減のためには、画像ブロックをオーバラップさせる等の処理が必要である。

(2)量子化特性

データ圧縮では、入力層と出力層のニューロン数（Ni）と中間層のニューロン数(Nh）の比とともに、中間層での量子化精度が問題となる。 そこで、図２に示すように中間層の出力全てに対して、一様量子化器(シグモイド関数の出力値 0.0-1.0 を 2-n ステップで等間隔量子化、n は量子化ビット数）を学習時に挿入した。 量子化精度に対する再生画像特性を図９に示す。

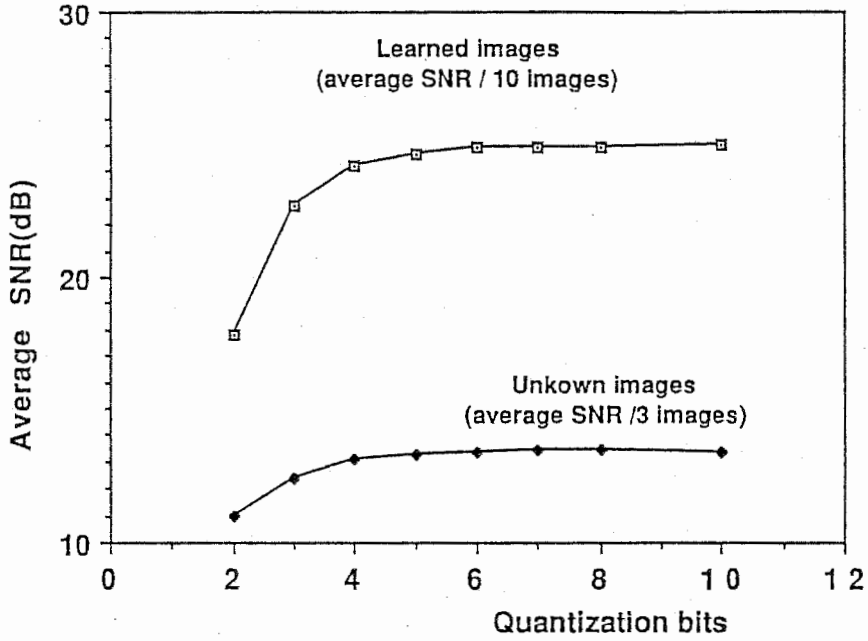


図９．量子化精度に対する画像品質　学習回数:400回/image，回路:64×16×64

学習時と学習後に中間層出力を n ビット量子化する場合を比較すると、学習時に挿入した場合の方が良い結果を得る。n=3 とすると、学習時量子化では 約2dB 劣化であるのに対し、学習後では 約 6dB 劣化であった。これは、学習により２乗平均誤差を最小にするよう量子化入力信号分布を変化させているためと考えられる。

本稿では、シグモイド関数が圧伸特性を持つことから、中間層の出力全てに対して同一ビットの線形量子化を用いたが、非線形量子化及び各ニューロンの出力に対して最小量子化ビット数となるよう学習させる検討を行う。

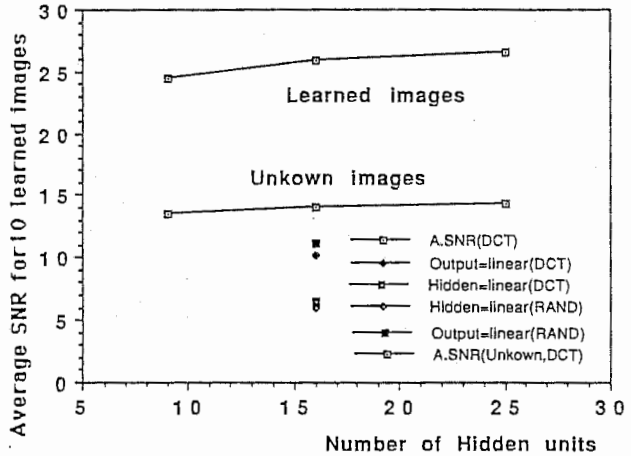
(3)重み初期値

従来、ニューラルネットのコネクションの重み初期値は、一様乱数が用いられていた。しかし、図２の回路を見ると、中間層の積和演算では直交変換を並列演算し、出力層で逆変換しているものとも見なせる。ただし、出力値に非線形のシグモイド関数が挿入されており、学習によって重み係数を最急降下法で適応制御しているので、初期値を適当に選べば非線形の多層適応型直交変換符号化と考えても良い。あるいは、初期値として画素間の自己相関係数を ρ_{nm} を選べば、並列演算タイプの多層適応型線形予測符号化とも見なせる。そこで、離散コサイン変換核（ＤＣＴ）を重みの初期値とした場合について検討した。重みの初期値として、

$$w_{k,l}=\cos\{(2k+1)m\pi/2M\}\cos\{(2l+1)n\pi/2M\}$$

を採用する。ここで、k,lは入出力層ニューロン、m,nは中間層のニューロンを示す。

図１０に、ＤＣＴ変換核を重み初期値として学習した場合の特性を示す。低次の係数のみを用いる直交変換符号化では、汎化特性は殆ど問題にならない。しかし、図６と比較すると、ニューラルネットが収束状態での学習、未学習画像に対するＳＮ比は、一様乱数の場合とほぼ同一であった。中間層でのニューロンの発火状態を図１１に示す。同図に示すように、初期状態からは学習によって変形するが、8x8 ブロックの左下が直流分、右上が周波数面での高次スペクトル成分に対応する。このため、初期値をＤＣＴ変換核とすることでスペクトル成分での学習が可能となる。



学習回数:400回/image，回路:64×16×64

図１０．初期値としてＤＣＴ変換核を採用した場合の再生画像品質

（中間層出力）

学習回数:400回/image，回路:64×16×64，学習画像数:10

図１１．初期値としてＤＣＴ変換核を採用した場合の学習済み画像の再生例

9.4 まとめ

① バックプロパーゲイション学習則を用いた高速な画像処理用ニューラルネットを並列計算機ＮＣＵＢＥ上に実現し、約１秒で画像を学習できる性能を得た。

② 圧縮特性として、学習画像数、中間層出力の量子化、重み初期値、それぞれに対するＳＮ比特性をシミュレーションにより求めた。

9.5 文献

(1) 上坂:ﾆｭｰﾗﾙﾈｯﾄの基礎論,TAIS,No.2,pp.2-30 (1988.06)

(2) M.Kunt,A.Ikonomopoulos,M.Kocher:Second-Generation Image-Coding Techniques,Proc.IEEE,Vol.73,No.4,April(1985)

(3) 原島博:知的画像符号化と知的通信,TV誌,Vol.41,No.7,pp.519-525(1988)

(4) 田崎,山田:動画像符号化の体系化とその応用,信学誌,Vol.71,No.7,pp.663-668(1988)

(5) G.W.Cottrell,P.Munro,D.Zipser:Image compression by Back Propagation: An Example of Extensional Programming, ICS Report 8702, February(1987)

(6) 曽根原,川人,三宅,中根:ﾆｭｰﾗﾙ･ﾈｯﾄによる画像ﾃﾞｰﾀ圧縮(Neuro-CODEC)の検討,信学技報,Vol.88,No.195(1988)

(7) J.G.Daugman,D.M.Kammen:Image Statistics, Gases, & Visual Neural Primitives, ICNN June, IV-163(1987)

(8) L.D.Jackel,R.E.Howard,et al:Building a Hieralchy with Neural network :an Example Image Vector Quntization,Vol.26,No.23,APPLIED OPTICS(1987)

(9) D.E.Rumelhart,G.E.Hinton,R.J.Williams:Learning representations by back-propagation errors, NATURE, Vol.323, 9 October (1986)

(10) 三宅他:神経計算論と並列信号処理,ﾃﾞｨｼﾞﾀﾙ信号処理ｼﾝﾎﾟ,pp.1-10(1987,12)

(11) G.Blelloch,C.R.Rosenberg:Network Learning on the Connection Machine", IJCAI,Knowledge Acquisition,pp.323-324(1987)

１０．前注意過程における図形の特徴点抽出モデルと画像復元

パターン認識の能力は対象の表現方法に大きく依存している。人間は高度なパターン表現機能を持っており、人間のパターン表現方法、記憶情報のアルゴリズム、パターンの学習アルゴリズム等を知ることは、工学的にもきわめて有意義である。これまで認知心理学を初めとする心理学の領域で人間のパターン認識のさまざまな特性がしらべられてきた。しかしながら、ほとんどの心理学的研究は情報処理過程の時間特性や容量をしらべたものであり、情報の内部表現すなわち脳内表現について検討したものはきわめて少ない。本研究はこれまで明らかにされてきた心理学的ならびに生理学的知見を踏まえ、パターンの脳内表現とりわけ２次元曲線（図形）の記憶表現について考察を行なう。

10.1 本論文の構成

本稿では、まず曲線の表現に関する計算論的考察を行なった後、曲線の「特徴点」を定義し、曲線の特徴点を並列に検出するアルゴリズムを提案する。次に曲線の構造記述という立場からアルゴリズムを改良し、パターンの大局的記述に関してシミュレーションを行なう。最後に得られた特徴点から元の曲線の復元を試みる。

10.2 計算論的考察

数学では、曲線の方程式を与えられたとき、その大域的形状を推測することを曲線の追跡と呼ぶ。この場合、大域的形状とは

対称性
特異点の位置、その付近の形状
曲線の凹凸、変曲点

などが挙げられる。人間のパターン認識ではもちろん曲線の方程式が与えられるわけではないが以上のような特徴は重要であり、何らかの方法で検出しているにちがいない。

２次元曲線の構造記述はパターン認識においてきわめて重要な問題である。人間の視覚研究においても、まず網膜像の強度変化が検出されその局所的な幾何学的構造が表現されると考えられている。この表現は原始スケッチ（primal sketch）と呼ばれている[1]。 原始スケッチの重要な機能は、画像の強度変化の検出と局所的な幾何学的構造の表現にある。原始スケッチは、素原始スケッチ（raw primal sketch）がと完全原始スケッチ（full primal sketch）に分けられている。素原始スケッチでは、縁edge、棒bar、端点termination、小塊blobといった表現素を用いて画像を表現する。

一方これまで曲線を特徴づける方法としてさまざまな方位を持つ線検出器によって曲線の各位置における接線方向を計算する試みがなされてきた[2]。 また、複数の長さの線検出器から曲線の端点を求める方法も考えられてきた[3],[4]。 ２次元

曲線の表現に関する研究はこれまで数多くなされてきたが、まず曲線の特徴点を検出し、それらの関係を記述することが基本的な方法だと思われる[5],[6]。

そこでパターン認識に重要な役割をはたしていると思われる曲線の「特徴点」を定義し、それらを並列に検出するアルゴリズムを提案する。２次元曲線の重要な特徴として曲率の大きい部分（high curvature point）、曲線の交点（cross point）、曲線の端点（termination）、間隙（gap）などが考えられる。これらは、曲線を特徴づける重要な位置であり特徴点（臨界点）と呼ばれている。曲率の大きい部分は心理学的にも工学的にも重要であることが認められている。例えばAttneave[7]は心理学的実験から、曲率の大きい位置は情報量が高いことを示唆した。またHoffmanとRichards[8]は図形の部分分割法として、曲率の大きい凹の部分で分割するのが自然であることを見い出している。一方心理学的研究から、人間が特定の位置に注意を払う前に（前注意過程において）並列に処理できる特徴があることが示唆されている。Julesz は並列処理可能な特徴をテクストン(texton)と名づけた[9]。テクストンの構成要素は線分の有無・長さ・幅・方位、線分の端点、線分の交点、小塊の有無・長さ・幅・方位などがある。興味深いのはこれらの構成要素の多くが上記の特徴点の特徴と一致している点である。以上のことから曲線における特徴点は曲線を特徴づける上で重要であるばかりでなく、前注意過程において並列に処理されるらしい（図１）。さらに、これらの特徴点付近の特徴だけでもとの図形情報をかなり精度よく復元できることが予想される。

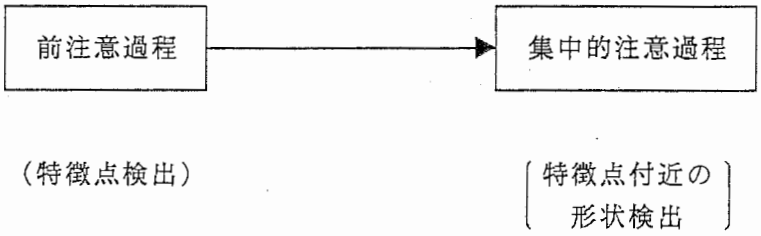


図１．形状の階層的処理

10.3 曲線の臨界点検出に関する並列アルゴリズム

ここでの基本的な仮定は曲線の詳細な特徴を線検出器などを用いずに方向選択性の低い微小なチャンネルによって検出されたエッジ情報によって知覚されるということである。すなわち以下に示すような並列アルゴリズムによって臨界点をまず検出し、次に注意過程を通して臨界点付近の詳細な形状をこの微小なチャンネルからのエッジ情報に基づいて知覚すると考えるのである。

(1) 計算論的ならびに生理学的知見

雑音が含まれる画像の中からできるだけ正確に線分を検出するための画像フィルタが研究されてきた。Canny[10]は、変分法を用いて最適フィルタの研究を行った。

Canny フィルタは、基本的には２次元ガウス関数の一階微分の形をしている。

一方、高等動物の大脳視覚領には、方向選択性の低いきわめて小さな受容野を持つ細胞が高密度で存在していると同時に方向選択性の高い複数の大きさの受容野を持つ細胞も存在することが生理学的研究から明らかにされている。さらに人間の視覚領には複数の大きさの線検出器が存在することが知られている。

(2) 並列アルゴリズム

我々はここでまず曲線データを方向選択性の低い微小なチャンネルと方向選択性の高い線検出器（以後単に線検出器と呼ぶ）によって、同時並列的に処理すると仮定する。次に同じ位置の情報を処理する線検出器から方向選択性の低い微小なチャンネルに対して抑制性の出力がかかると仮定する。このように仮定すると曲線の直線性の高い部分では線検出器の出力は高くなり方向選択性の低い微小なチャンネルの出力は抑えられる。一方直線性の低い部分では線検出器の出力が低く、微小なチャンネルの出力にはあまり影響を与えないであろう。その結果、直線性の低い部分、すなわち臨界点付近においてのみ微小なチャンネルの出力が得られることが予想される。図２は処理の流れ図を示している。まずステップ１ではエッジが抽出される。この段階での出力は２次元曲線の詳細な形状情報を与えている。次にステップ２ではこの出力に方向性を持つ線検出器から抑制がかかり臨界点がマークされる。

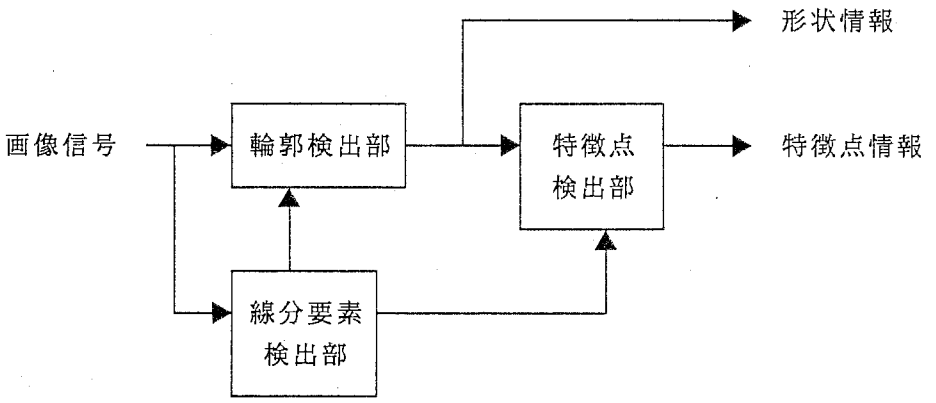
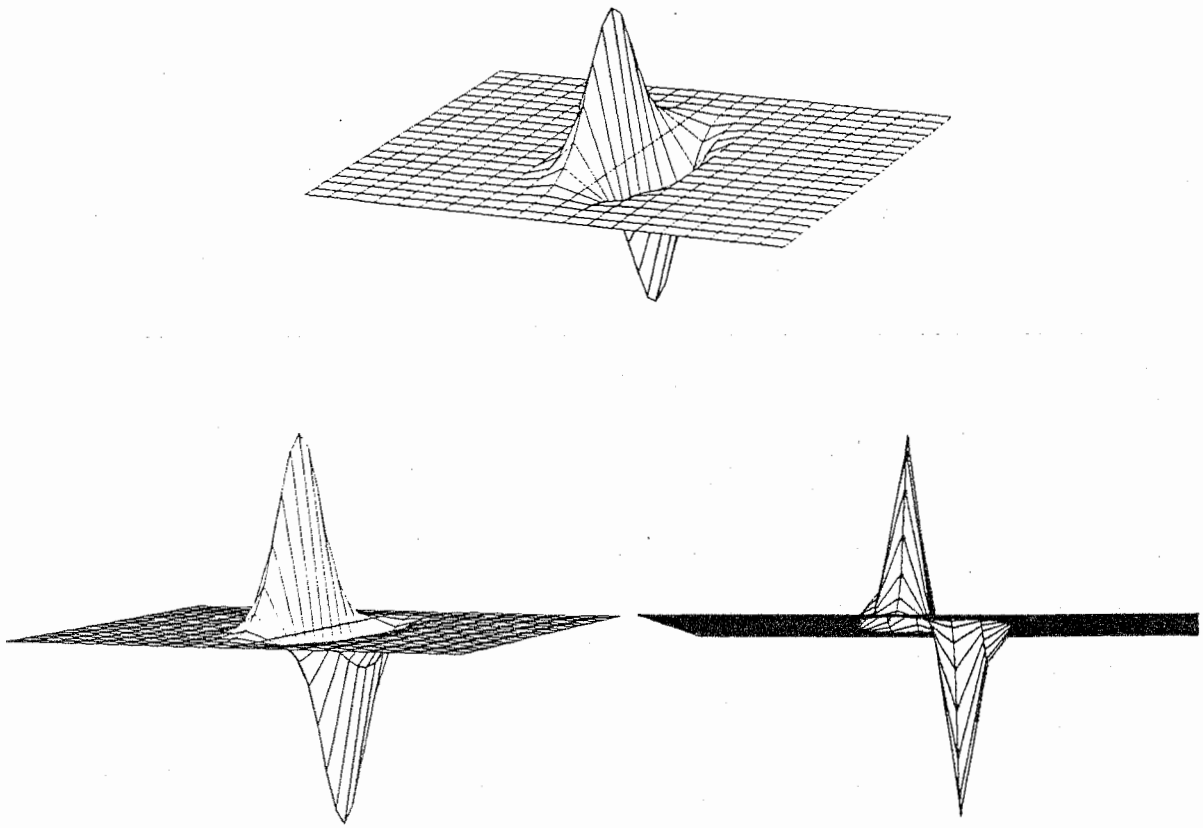


図２．処理の流れ図

次に各チャンネルのフィルタ特性について説明する。ここでは方向選択性の低い微小チャンネルと線検出器が同一の関数で決まるフィルタ特性を持つと仮定した。ここでは、軸に垂直な方向の断面がガウス関数の一階微分の形をし、軸に平行な断面がガウス関数になっているフィルタを用いた（図３）。すなわち、フィルタの空間特性f(x,y)は、

$$f(x,y) = x\exp(-x^2/2\sigma_x^2 - y^2/2\sigma_y^2)/(2\pi\sigma_x^3\sigma_y)$$



図は $\sigma_y=3\cdot\sigma_x$ の場合を示している。σ_y の方位が線検出器の方位である.

図３．フィルタの形状

これらのチャンネルのフィルタ特性は、ガウス関数の一階微分の形をしているので、σ_xの値を変えることによって、画像に対してさまざまな度合で平滑化した後、微分操作を施すことにより強度変化の大きい部分を検出することができる。ここで方向選択性の低い微小なチャンネルの場合、

$\sigma_x=\sigma_y$

とし、線検出器の場合、

$\sigma_y=n\,\sigma_x$　　（nは整数）

とした。図３にフィルタの空間特性f(x,y)の一例を示す。nを変化させると線検出器のフィルタの長軸方向の長さが変わり直線性の判定基準が変化し、検出される曲率の範囲が変化する。すなわちnを大きくすることによって検出される曲率の範囲が広くなる。

次に線検出器の閾値は各フィルタの最大出力の１／２とした。このように閾値を設定すると直線に近い部分だけが出力されその結果方向選択性の低いチャンネルの出力は曲線の臨界点付近だけが残ることが予想される。

具体的なアルゴリズムは以下の通りである。画像データが与えられると二種類のチャンネルによって同時に処理が始まる。微小チャンネルのマスクサイズは7×7（フィルタの形状を7×7で表現）、サンプリングは3×3おきに行なった。$\sigma x = \sigma y = 2.0$で方位は10°おきに36種類のフィルタを用意した。次に、閾値を2.5として出力が閾を越えた点をエッジとして表示した。このようにして抽出されたエッジ部分について

$\sigma x = 2.0$、 $\sigma y = 6.0$
$\sigma x = 2.0$、 $\sigma y = 7.0$
$\sigma x = 2.0$、 $\sigma y = 8.0$

の３種類の線検出器によって再びコンボリューションを行なった。

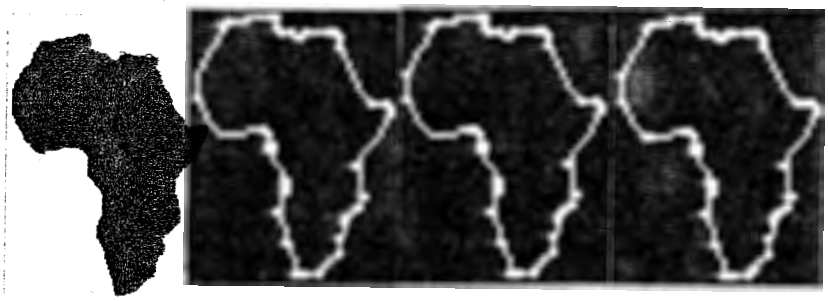
線検出器の最大出力は２値(0,-1)パターンの場合、$\sqrt{2}\pi\sigma_y$となる。したがって上記３種類のフィルタに対して、最大出力の1/2すなわち、7.56、8.82、10.08を閾値とした。ここで閾値を越えなかった部分で方向性の低いフィルタの出力が得られている部分を特徴点としてマークした。

10.4 並列計算機ＮＣＵＢＥによるシミュレーション（曲線の臨界点検出）

ＮＣＵＢＥは５１２個のプロセッサがハイパーキュブ上に接続されている並列計算機であり、これを用いてシミュレーションを行った。ディスプレーは256×256画素からなりディジタイザを用いて図形入力を行った。図形部分の画像データは－１であり背景は０である。256×256画素に対して3×3のサンプリング間隔で処理を行なった。検出器の方位は36種類である。上述のようにノード（プロセッサ）数は512個なので、画像の1行（256画素）を6個のノードに割り付けて処理を実行した（各ノードは6方位を処理する）。計算時間は約2秒であった。入力図形として北海道とアフリカ大陸を用いた結果を図４および図５に示す。。各σ_yに対して抽出されたエッジ（灰色）と検出された臨界点（白丸）が示されている。つぎに線図形を入力したときの結果を図６に示す。曲率の高い点以外にも、端点、交点等の特徴点がうまく抽出されている。またσyを大きくすることによって曲率の小さい部分まで検出されていることがわかる。このことは、量子化ノイズがあり鮮明ではないものの図７からも明らかである。

左から原画像、σ_y=6.0の結果、σ_y=7.0の結果，σ_y=8.0 の結果．それぞれエッジの数２１９に対して、６４、８８、１００個の特徴点が抽出された．

図４．北海道を入力図形とした場合の特徴点



左から原画像、σ_y=6.0の結果、σ_y=7.0の結果，σ_y=8.0 の結果．それぞれエッジの数１５４に対して、４０、４４、５７個の特徴点が抽出された．

図５．アフリカ大陸を入力図形とした場合の特徴点

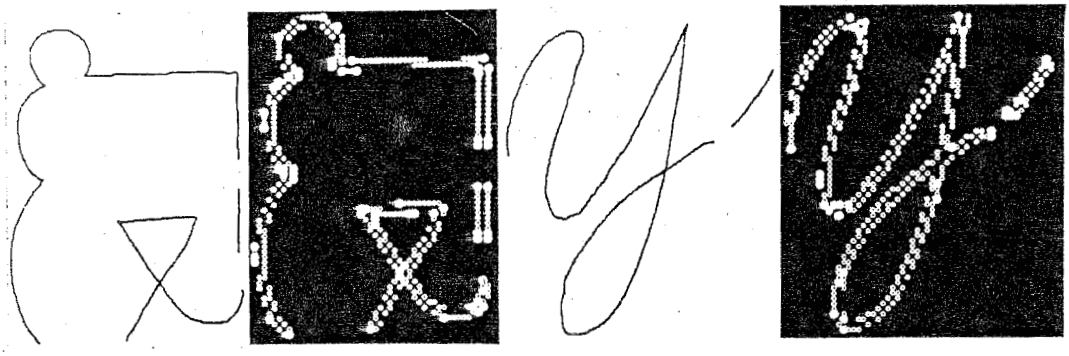


図６．線図形を入力とした場合の特徴点

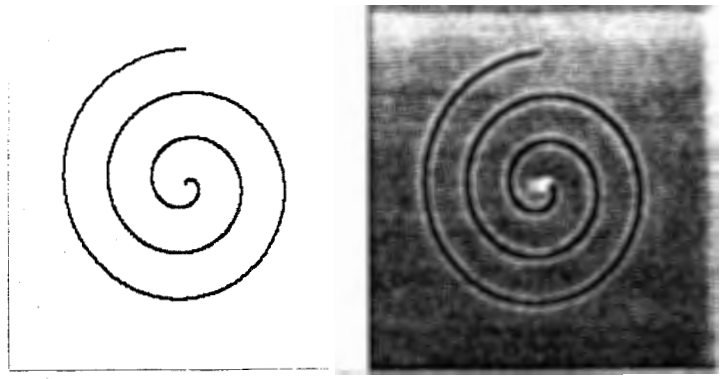


図７．螺旋に対する特徴点

10.5 図形の構造化と特徴点検出

上記のようなアルゴリズムを用いれば、図形の特徴点を検出することができる。しかし、このアルゴリズムをそのまま用いれば、ノイズのような細かな凹凸をも検出してしまう。我々は、まず、図形の大局的特徴となる点を検出する必要がある。そこで、図形の輪郭線を平滑化すると同時に、特徴点が検出できるようなアルゴリズムを考えた。これは、$\nabla^2 G$(Marr-Hildrethoperator)[11]で画像を平滑化しゼロ交差をもとめた後に、単純型細胞様のフィルターを用いて上記のアルゴリズムと同様に特徴点を検出するものである。いま

$$G(r) = \frac{1}{2\pi\sigma^2} \exp(-r^2/2\sigma^2)$$

とすれば、

$$\nabla^2 G = (r^2/2\sigma^2 - 1)(\exp(-r^2/2\sigma^2)) / \pi\sigma^4$$

となる（図８）。

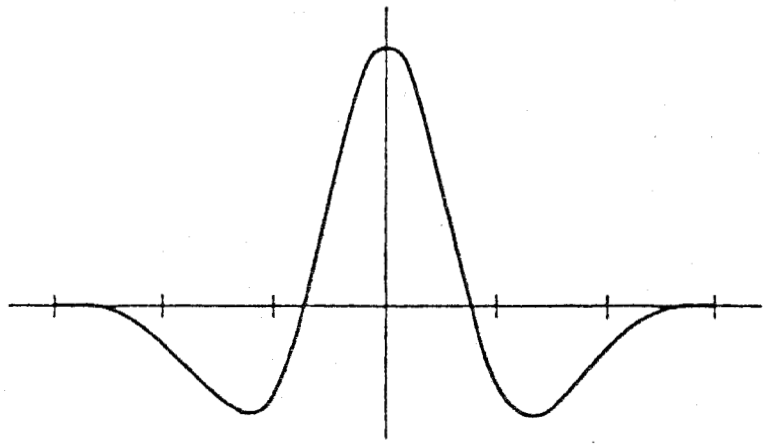


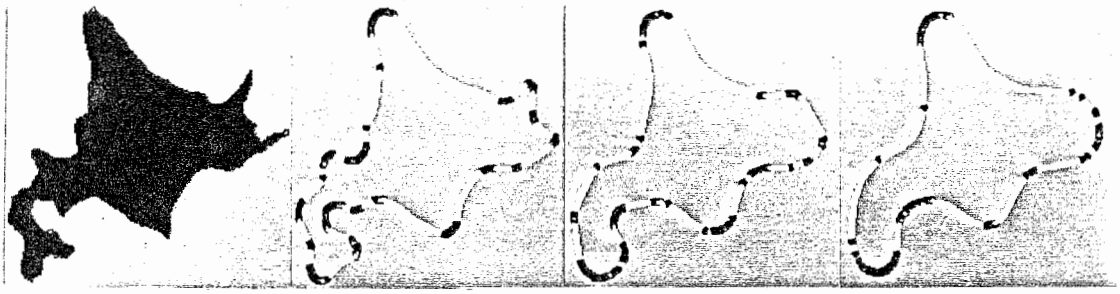
図８．Marr-Hildreth オペレータの１次元断面

ここで画像をI(x,y)とすれば、$\nabla 2G*I = 0$　なるI(x,y)がもとめるゼロ交差である。

σの大きさを変化させることにより様々な平滑化ができ、解像度を変化させて特徴点を検出することが可能である。ここでは、まず画像のゼロ交差を求めた後、その出力に対して以下のような線検出器を作用させ前述の特徴点検出アルゴリズムと同様の方法で特徴点を検出した。具体的にはσは5，10，15，20画素の4種類である。ゼロ交差を求めた後、3×31のマスクをかけ、その中に入るゼロ交差の数を計算する。この計算は各ゼロ交差点を中心に36方位で行ない、この最大値が25以下のゼロ交差点を特徴点としてマークした。

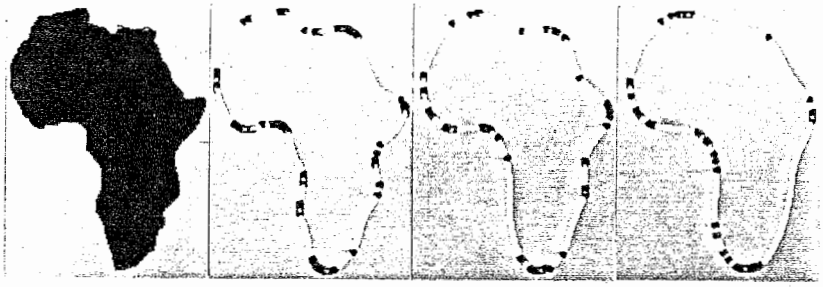
10.6 並列計算機ＮＣＵＢＥによるシミュレーション（特徴点検出）

256×256画素に対して1×1のサンプリング間隔で処理を行った。既に述べたようにノード数は 512個なので、画像の１行分を２分割し、２個のノードに割りつけて処理を実行した。したがって各ノードは 127画素を 36方位で処理を行なった。 計算時間は約２秒であった。北海道とアフリカ大陸に対する結果を図９および図１０に示す。



左から、原画像、σ=10.0 の結果、σ=15.0 の結果、σ=20.0 の結果．それぞれエッジの数は、７６６、６７８、６４２であり、特徴点の数は、１８２、１２９、１１０個であった．

図９．北海道を入力図形とした場合の特徴点



左から原画像、σ=10.0の結果、σ=15.0の結果，σ=20.0 の結果．それぞれエッジの数は、５５６、５３４、５２６であり、特徴点の数は、１０７、６０、５２個であった．

図１０．アフリカ大陸を入力図形とした場合の特徴点

本アルゴリズムにおいても端点等の特徴点が抽出される（図１１）。次節ではこれらの特徴点に基づき原画像の復元を試みる。

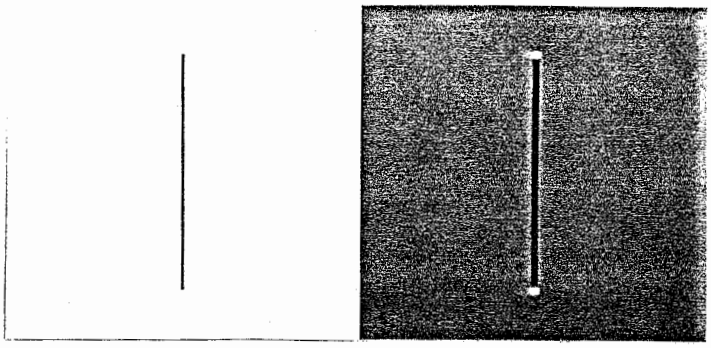


図１１．線図形を入力とした場合の特徴点

10.7 補間と画像復元

このような特徴がいったん検出されると、これをもとに図形を復元できるる可能性がある。スプライン補間C^1級曲線では曲率がある程度不連続になると「不連続」が知覚される。しかし、C^2級曲線ではすべてなめらかに知覚されるといわれる。

一方、数学で「最もなめらかな曲線」とは、f(x)がk回微分可能であるとき

$$\int_b^a [f(x)^{(k)}]^2 dx$$

を最小にする曲線のことを言い、これは、2k-1次のスプライン関数になっている。そこでC^2級で最もなめらかな曲線であり、minimum curvature property をもつ３次のスプライン補間をここでは考える。３次スプラインとは各節点(knot)間を異なる３次の多項式で補間するものである。

(1) 点線の補完

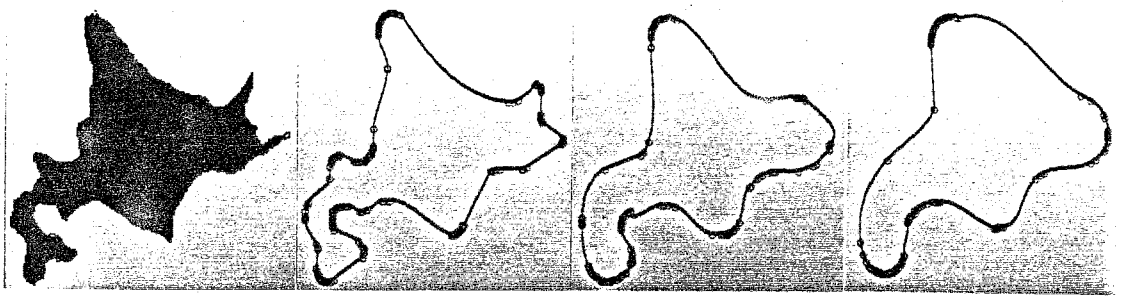
点が適当な条件下で配置されると曲線（直線ではなく）が知覚される。この問題を最初に論じたのは Koffka(1935)であろう。Koffka の例では、円周上に点が等間隔に配列されている。この場合、８点以下では８角形に見えてしまう。すなわち、直線補完がなされる。９点以上になると円が見えてくる。この場合３点がなす角度は１４０°になる。SmitsとVos[12]は同様の実験を行い３点のなす角度が１４０°以上のときに曲線が知覚されるという結果を得ている。

(2) 補間アルゴリズム

ここでは、上記の知見を踏まえて特徴点から以下のようにして補間を行い原画像を復元する。隣合う２つの特徴点のなす角が１４０°以下のときは折れ線補間を、それ以外のときはスプライン補間を行う。

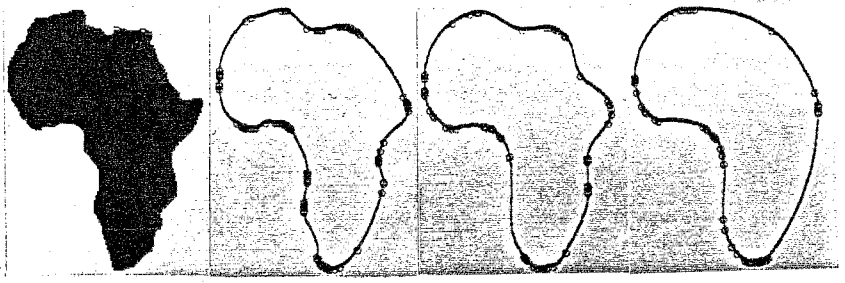
10.8 並列計算機ＮＣＵＢＥによるシミュレーション（特徴点からの画像復元）

上記のアルゴリズムに基づいて、図９および図１０で得られた特徴点から原画像（図９、１０の白線）の復元を試みた。結果を図１２および１３に示す。図から明らかなようにかなり正確に復元された。



左から σ=10.0の結果，σ=15.0の結果，σ=20.0 の結果．それぞれ図１０に対応する．

図１２．北海道の特徴点からの復元



左から原画，σ=10.0の結果，σ=15.0の結果，σ=20.0 の結果．それぞれ図１１に対応．

図１３．アフリカ大陸の特徴点からの復元

10.9 考察

２次元曲線の重要な特徴として、曲率の曲大点（curvature extrema）、曲率の交差点（cross point）、曲線の端点（termination）、間隙（gap）等が挙げられる。心理学的研究から、我々はどこかに注意を向ける前に（すなわち前注意過程において）このような特徴をすばやく並列に処理していることが知られている。今回このような現象、すなわち異なる種類の特徴を持つ位置（臨界点）を単一のアルゴリズムでシミュレートすることができた。本アルゴリズムから曲線の細部の情報は、高密度で存在する方向選択性の低い微小なチャンネルによって処理され、方向選択性の高いチャンネルは線条性の判定に用いられており、方向選択性の高いチャンネ

ルから方向選択性の低い微小なチャンネルへの抑制性の結合の存在が予想される。これらの３つの仮定を脳の神経系と対応させるなら、線検出器は４層あるいは６層のニューロンに、ゼロ交差は４ｃβ層のニューロンに対応しているのかもしれない。最近６層から４層への抑制性のフィードバックがあることが報告されている(3)。

通常パターン認識の分野で使われている手法はあらかじめ特徴空間を与えておき、学習によって特徴空間の各軸の重みを変えていくという方式をとっている。一方細線化などの手法を用いて、パターンの骨格を出し骨格の交点や角、端点等の関係を抽出するという方法がある。しかしこの方法においては、このような特徴点を抽出することがむずかしいため、主として前者の方法がとられてきた。ここで示した並列アルゴリズムは、このような難点を克服するものであると考えられ、従来のパターン認識の前処理の部分に採用することによって、パターン認識能力を向上させることが期待できる。

10.10 文献

(1) Marr,D: Vision, Freeman(1987)

(2) Zucker,S.W.:Early processes for orientation selection and grouping. Computer Vision Graphics Image Process(1986)

(3) Dobbins,A.,Zucker,S.W.,and Cynader,M.S.:Endstopped neurons in the visual cortex as a substrate for calculating curvature, Nature,329, pp.438-441(1987)

(4) Hubel,D.H.,Livingstone,M.S.:Segregation of form,color,and stereopsis in primate area 18, The Journal of Neuroscience,7,pp.3378-3415 (1987)

(5) Asada.H and Brady.M:The curvature primal sketch,IEEE Trans.P.A.M.I., 8,pp.2-14(1986)

(6) Freeman,H.:Shape description via the use of critical points,Pattern Recognition,10,pp.159-166(1981)

(7) Attneave.F: Some informational aspects of visual perceptrio,Psychol. Rev.,61,pp.183-193(1954)

(8) Hoffman,D.D., and Richards,W.A.:Representing smooth plane curves for recognition:Implications for figure-ground reversal,in Proc.Nat.Conf. Artif. Intell.,Pittsburgh,PA,pp.5-8(1982)

(9) Julez,B.: Texton gradients: The texton theory revisited. Biological Cybernetics,54,pp.245-251(1986)

(10) Canny,J.: A computational approach to edge detection. IEEE Trans., P.A.M.I.,8,pp.679-698(1986)

(11) Marr,D.,Hildreth,E.:Theory of edge detection, Proc.R.Soc.Lond.,B207, pp.187-217(1980)

(12) Smits,J.T.S.and Vos,P.G.: The perception of continuous cutves in dot stimuli, Perception 16,pp.121-131(1987)

１１．ｴﾈﾙｷﾞｰ学習をする局所並列確率ｱﾙｺﾞﾘｽﾞﾑを用いた輪郭線抽出[11],[12]

11.1 はじめに

輪郭線の抽出は、図形認識の基礎となる最も大切なものの一つである。画像をまず線図形に直し、それを用いて構造記述することが便利でかつ優れているために、輪郭線抽出がしばしば行われている[1],[2],[3]。 一般にビデオカメラを用いて入力された「顔」等の自然画像の場合には、コントラストの変化が大きく、また反射や影等のノイズもあり、正確な輪郭線を抽出することは困難である。そこで、対象画像の形状に関する知識を用いることが必要になってくる。

コンピュータに形状の知識をもたせる際、対象画像の形状を完全な輪郭線として与えたのでは、対象画像の数だけ形状の知識が必要となるため効率が悪く、さらに知識を持たない対象の輪郭線は抽出できないことになる。そこで線画のごく近傍だけに着目して、端点や交点などの非常に基本的な近傍系の表現素を組合せることによって形状を表現し、表現素の出現確率として形状の知識を表現することが記憶効率，および知識表現の柔軟性の点で優れていると思われる。

Geman & Geman らによって、画像の形状の知識を確率場のパラメータとして表現し、このパラメータとノイズの乗った不完全な画像から原画像を復元する試みが行われている[4]。 彼らは、ラインプロセスと呼ばれる画像の不連続性を表現する隠れた確率変数を用いて形状の知識を確率により表現し、シミュレーテッドアニーリング等の手法を用いて原画像の復元を行っている。また、D.Geman は、ラインプロセスが画像の濃淡値の不連続，すなわち輪郭線を表すことに着目して、おもちゃや家等の自然画像の輪郭線抽出を行っている[5]。 ラインプロセスは、近傍系における連結の形式を、その出現確率を反映する局所的エネルギーとして持つことができるため、輪郭線の途切れを禁止するといった局所的な知識を持たせることができ、連結性に優れた輪郭線抽出を行わせることができる。さらに、Kawatoらはそれまでの手法では予め設定しなければならなかった局所的パラメータを学習により自動的に獲得する手法[6]を提案している。

本稿では、まず、確率緩和やシミュレーテッドアニーリングを用いて近傍系の表現素の出現確率として与えられた確率パラメータから輪郭線抽出を行う手法について述べる。次に、対象とする濃淡画像と輪郭線とを繰り返し学習させることで確率パラメータを自動的に調節する手法について述べる。さらに、これらの手法を用いて、人間の顔画像の輪郭線を抽出する方法および実行結果について述べる。最後に、従来のＭＲＦモデルを拡張した階層的ＭＲＦモデルを提案する。

11.2 輪郭抽出のマルコフ確率場モデル

２次元のＭＲＦを考える。まず S を画素およびラインプロセスからなる画像の格子点（サイト）の集合とする。 また S の各要素 s に付随している確率変数を x_s で、さらに S上の確率場を $X=\{x_s, s \in S\}$ で表す。x_s の値域は、0 から L-1 までの離散 L値であるとし、この値域を $\Lambda=\{0,1,\cdots,L-1\}$ で表す。それぞれの x_s が特定の値をとったものを状態配置 ω と表す。この ω がひとつの画像に対応する。S

上の確率場 X がマルコフ的であるとき、全ての x_s に対する同時確率分布，すなわち画像ωが生起する確率 Π(ω) は、次式で表される。

$$\Pi(\omega)=\frac{\exp(-U(\omega)/T)}{Z} \tag{1}$$

サイトを表す番号の系列 $\{n_t, t\geqq 1\}$ が任意のサイトを無限に多く含むと仮定すると、任意の $X(0)=\eta$ について、確率変数 X の値の分布関数 P(X) と同時確率分布関数 Π(X) との間に、

$$\lim_{t\to\infty} P\{X(t)=\alpha \mid X(0)=\eta\}=\Pi(\alpha), \qquad \forall \alpha \in \omega \tag{2}$$

が成り立つ。さらに、 S の近傍系が作るクリークを C、 クリークが持つ局所的エネルギーを V_c で表し、サイト s の確率変数が x でありかつ他のサイトがωに一致する様な状態配置を ω^x で表すと、Π(ω)=P(X=ω)がギブス的であるならば、

$$P(X_s=x_s \mid X_r=x_r, r\neq s) = Z_s^{-1}\exp\{T^{-1}\cdot \sum_{C:s\in C} V_C(\omega)\} \tag{3}$$

$$Z_s = \sum_{x\in\Lambda} \exp\{T^{-1}\cdot \sum_{C:s\in C} V_C(\omega^x)\} \tag{4}$$

が成り立つ。s に関するクリークの中の任意のサイトは s の近傍でなければならないから、サイトs が x_s をとる確率は s の近傍の確率変数のみに着目した局所演算によって求めることができる。

したがって、各々のサイトにおける x_s の各々の値の出現確率 P(ω)を次式で定めて十分に状態変化を繰り返せば、局所並列演算によって同時確率分布Π(ω)を実現することができる。

$$P(\omega) = \frac{\exp\{-U(\omega)/T\}}{Z} \tag{5}$$

ここで U はＭＲＦのポテンシャルエネルギーであり、 S の近傍系が作るクリーク全体の集合Ｃの要素 C に含まれる ω のみに依存する局所的エネルギー V_C の和として表される。

$$U(\omega)=\sum_{C\in\mathcal{C}} V_C(\omega) \tag{6}$$

また、Z は分配関数と呼ばれる正規化のための定数であり、(7)式で表される。

$$Z = \sum_{\omega} \exp\{-U(\omega)/T\} \qquad (7)$$

また、T は温度と呼ばれるＭＲＦの動作環境パラメータであり、T の値によって各 ω の $P(\omega)$ の比をコントロールすることができる。この性質により 次の条件式を満足するスケジュールに従って T を減少させれば、状態の確率分布が $U(\omega)$ の最小値を与える状態については一様でそれ以外の状態については確率 0 とするような $\Pi(\omega)$ を実現することができる。

$$a)\quad T(t) \rightarrow 0 \qquad t \rightarrow \infty \qquad (8a)$$

$$b)\quad T(t) > \frac{N\delta}{\log(t)} \qquad \text{for} \quad t > t_0 \qquad (8b)$$

ここで N はサイトの数、δ は $U(\omega)$ の最大値と最小値との差である。

ポテンシャルエネルギーの定義式からわかるように、あるサイズのクリーク C が取り得る可能な全ての状態 C_i $(i=1,2,\cdots,M)$ について局所的エネルギー V_{Ci} の値を決めてやれば、それはマルコフ確率場モデルを、ひいては画像がこうあるべきだという事前知識を決定することになる。その上で、シミュレーテッドアニーリングを施しながら局所並列繰り返し演算を行えば $U(\omega)$ の最小値を厳密に求めることができる[4]。

画像の濃淡値を入力画像に固定したまま、ラインプロセスのみを変化させて輪郭線を抽出することを考える。いま、図１の様な近傍系を考えてみよう。

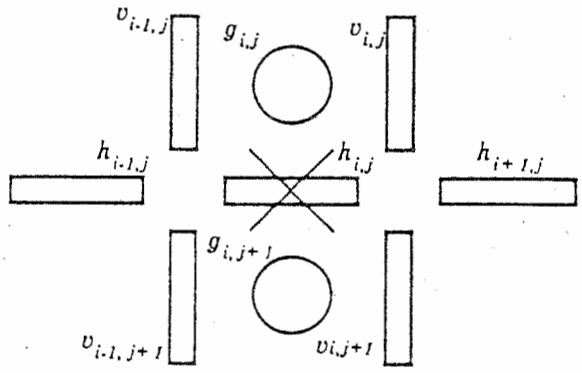


図１．近傍系

｜，－，○はそれぞれ縦，横のラインプロセスと画像の濃淡値を表す。×印をつけたサイトを中心に考える。 サイト s_{ij} が取り得る状態は、そこにエッジが存在

する（$l_{ij}=1$），または存在しない（$l_{ij}=0$）のいずれかである。そこで、画素の濃度勾配のポテンシャルを $U_g(\omega)$，ラインプロセスのポテンシャルを $U_l(\omega)$，サイト全体のポテンシャルを $U(\omega)$ とするとそれぞれ次式のようになる。

$$U(\omega) = U_g(\omega)+U_l(\omega) \tag{9}$$

$$U_g(\omega) = V_0 \cdot \sum_{i,j} \{(g_{i[j+1]}-g_{ij})^2\cdot(1-h_{ij})+(g_{[i+1]j}-g_{ij})^2\cdot(1-V_{ij})\} \tag{10}$$

$$U_l(\omega) = \sum_{i,j} \sum_{C\in \mathcal{C}} V_C(\omega) \tag{11}$$

このポテンシャルに従って十分に状態変化を繰り返すことによって、画像の境界として輪郭線を求めることができる[5]。

11.3 エネルギー学習

画像のマルコフ確率場モデルを特徴付ける局所的エネルギーを真の画像の状態を教師信号として同じ統計的性質を持つ画像データに対して推定・学習するアルゴリズム[6]が Kawato らによって提案されている。ここでは目標とする輪郭線画像を教師信号として濃淡画像から輪郭線画像を推定するのに必要となる局所的エネルギー V_{Ci} の値を学習するアルゴリズムについて説明する。

いま、対象としている輪郭線画像で、ある状態配置 ω が生起する確率を$P(\omega)$とする。一方、局所的エネルギー $V_{Ci}(i=1,2,\cdots,M)$ が決められたマルコフ確率場モデルが、同じ状態 ω をとる定常確率を $P'(\omega,V_C)$ とする。このとき、カルバック の divergence $G(V_C)$ は、次式のように定義される。

$$G(V_C)=\sum_{\omega} P(\omega)\log\{P(\omega) / P'(\omega,V_C)\} \tag{12}$$

$G(V_C)$ は、$P(\omega)= P'(\omega,V_C)$ のとき 0 で、それ以外では正である。マルコフ確率場モデルが輪郭線画像のなるべくよいモデルになるよう、G の最急降下方向に局所的エネルギーを変化させるとすると、Boltzmann-machine の学習則の導出と同様にして次の学習則が得られる。

$$\Delta V_{Ci} = -\varepsilon\, \partial G / \partial V_{Ci}$$

$$= -\eta\{ \sum_{C\in \mathcal{C}} I_i(C) - \sum_{C\in \mathcal{C}} I_i(C)' \} \tag{13}$$

ここで I_i はクリーク C の状態が C_i のときだけ 1 をとり、それ以外では 0 となる C_i の定義関数である。右辺第 1 項は、学習の目標となる輪郭線画像（教師画像）において、クリーク C が C_i をとる個数を画像全体で数え、それをサイトの

数で平均したものである。一方右辺第２項は、 ある局所的エネルギー V_{Ci} を固定した時のマルコフ確率場モデルで、クリーク C が C_i をとる個数の平均値である。これは、マルコフ確率場モデルで画像を生成して計算する。

11.4 顔画像輪郭抽出の実験結果

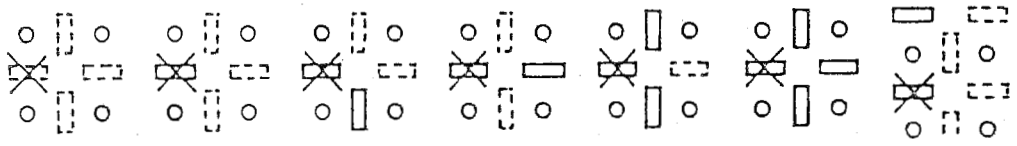
11.4.1 実験条件とパラメータ

縦１２８画素／横１２８画素、モノクロ２５６階調に表現された人間の顔写真の輪郭線を抽出した。原画像を図２（ａ）に、またＶＣｉの学習を行う際に学習の目標とする輪郭線画像を同図（ｂ）に示す。

(a)原画像　　(b)教師画像

図２．原画像と学習目標

ここでは 11.2 節で述べた１近傍のクリークに加えて平行二重線のコントロールを目的に "parallel" を加えて、図３に示す合計７つのタイプの近傍系を用いた。エネルギー学習をしない場合には、ラインプロセスのポテンシャルを求める際の係数 V_{Ci} は、図示の値を用いた。 また、エネルギー学習をする場合には、この値を初期値として学習を行わせた。また濃度勾配のポテンシャルを求める際の係数 $V_{\emptyset}$ は、297.0 とした。



(no line) (ending)(bending)(continu-(branching) (crossing) (parallel)
ation)

$V_{\xi 1}=0$　$V_{\xi 2}=30$　$V_{\xi 3}=10$　$V_{\xi 4}=10$　$V_{\xi 5}=20$　$V_{\xi 6}=30$　$V_{\xi 7}=20$

○ … pixel　　× … site

図３．実験に用いた近傍系

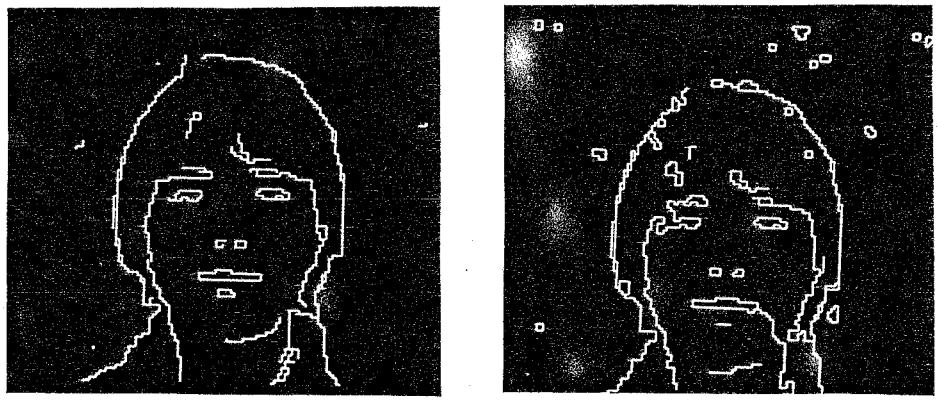
11.4.2 アニーリングスケジュール

金属結晶を作る際にアニーリングスケジュールが速すぎると結晶の中に傷が生ずるように、シミュレーテッドアニーリングにおいても、急激に T を下げると、マルコフ確率場の中に結晶の傷に似た雑音が生ずる。 11.2 節で説明したアニーリングスケジュールを用いれば最適な同時確率分布が得られるが、多大な時間を必要とする。そこで次式のように定数 α を用いて α 倍の加速を行い、試行錯誤により傷ができない範囲の α と初期温度 T_0 を決定した。なお、アニーリングスケジュールを決める際には、エネルギー学習は行っていない。

$$T(k) = \frac{T_0}{\ln(\alpha k+2)} \qquad (k : \text{sweep number}) \tag{14}$$

(1) 初期温度 T_0

まず α=1 とし、様々な初期温度 T_0 について輪郭線の推定を行わせてみた。図4（a）は、T_0=50 として約 4000回 状態変化を行わせたものであり、（b）は、T_0=10 として約 8000回 状態変化を行わせたものである。（a）には見られない雑音が（b）に現れてきている。これは、T_0=10 では初期温度が低すぎたためである。従ってここでは、初期温度 T_0 の値は 50 が適当であると考えられる。



(a) T_0=50　　(b) T_0=10

図4．初期温度 T_0 を変えた場合の輪郭画像

(2) 加速係数 α

T_0=50 とし、様々な加速係数 α について輪郭線の推定を行わせた。図5（a）は α=10,（b）は α=100,（c）は α=1000 として、それぞれ 1000回づつ状態変化を行わせたものである。（a），（b）には見られない雑音が（c）に現れてきている。これは、α=1000 では加速が急すぎたためである。従ってここでは、加速係数 α の値は 100 が適当であると考えられる。

(a) α=10　　(b) α=100　　(c) α=1000

図５．加速係数αを変えた場合の輪郭画像

11.4.3 学習

第11.2節で説明した学習則に従えば、状態変化を十分に行った後に、マルコフ確率場モデルの中でクリーク C が C_i をとる個数の総和をサイトの数について平均したものと、教師画像中でクリーク C が C_i をとる個数の総和をサイトの数について平均したものとの差に、微小な係数ηを乗じた値を ΔV_{Ci}として学習を行う。いま、η/(サイト数) を改めてηとすれば、C_i をサイト数で平均する必要はないので個数をそのまま用いることにする。

学習が進むにつれて、教師画像中の C_i の数とマルコフ確率場モデル中の C_i の数との差

$$\Delta N_i(c) = \sum_{C \in \mathcal{C}} I_i(C) - \sum_{C \in \mathcal{C}} I_i(C)' \qquad (15)$$

すなわちクリーク数の誤差は減少する。しかし教師画像中に数が少ない C_i においては、教師画像中の C_i の数に対するクリーク数の誤差の比率、すなわち

$$\frac{\Delta N_i(C)}{\sum_{C \in \mathcal{C}} I_i(C)} \qquad (16)$$

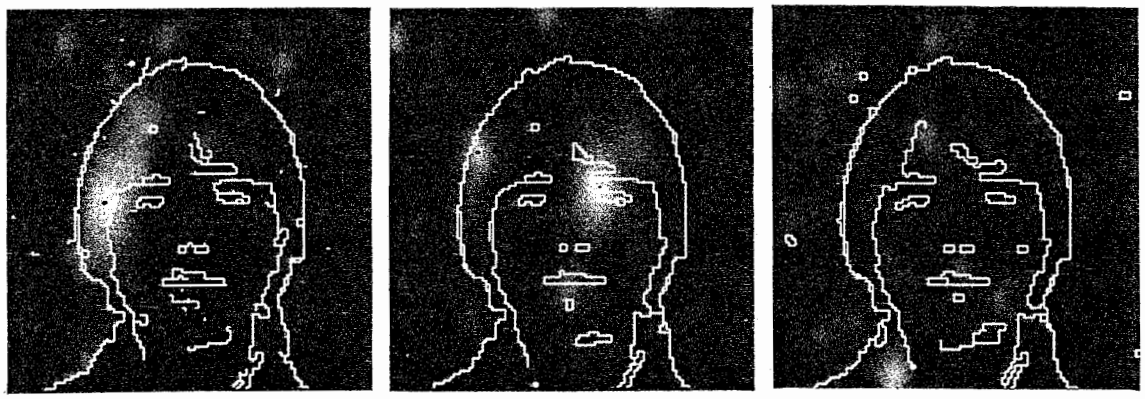
は、なかなか減少せず、エネルギー学習に非常に時間がかかる。そこで、教師画像とマルコフ確率場モデルとの C_i の数の差を次式のように教師画像中の C_i の個数で規格化したものをΔV_{Ci}として学習を行った。

$$\Delta V_{Ci} = -\eta \left\{ \sum_{C \in \mathcal{C}} I_i(C) - \sum_{C \in \mathcal{C}} I_i(C)' \right\} / \sum_{C \in \mathcal{C}} I_i(C) \qquad (17)$$

なお、教師画像中の C_i の個数が 0、すなわち $\sum_{C \in \mathcal{C}} I_i(C)=0$ の場合には ΔV_{Ci}

の値が定まらないので、規格化を行わずに ΔV_{ci} とした。また、"No lines" は最小のエネルギーのままであることは明白なので、V_{c1} は学習を行わせなかった。また以下では、η の値を試行錯誤により選定し、0.1 とした。

図６（ａ）は学習前，（ｂ）は１回学習後，（ｃ）は２回学習後のエネルギーパラメータにより各々 10000回 の状態変化を行った結果得られた輪郭線画像である。（ａ）では端点部分 "ending" が非常に多くみられるが、（ｂ）ではこれが著しく減少していることがわかる。さらに、（ｃ）では前髪と顎の輪郭線が延びて連結しつつあることがわかる。しかし、"ending" や "parallel" 等を禁止する必要から、学習によってこれらにかかわる V_{ci} が著しく増大したために $U(\omega)/T$ が増大し、結果的に T_0 が下がったことと等価となり、生成された輪郭線画像中に雑音が現れてきている。



(a) 学習前　　(b) １回学習　　(c) ２回学習

図６．学習による輪郭画像の変化

この問題点を解決するには、学習による V_{ci} の増大を見込んで初期温度 T_0 の値を十分に高く設定することが必要となる。とくに、教師画像中に存在しない C_i (個数が 0 の C_i)がある場合は、V_{ci}が極めて大きくなるため、T_0 を非常に高く設定することが必要になる。ここで用いた教師画像の場合には その中に"ending" が存在していないため、学習により V_{c2}が極めて大きくなることが予想される。したがって T_0 を非常に高くしてゆっくりとアニーリングすることが必要であり、多大な時間が必要となる。

11.4.4 Constrained Optimization

前節の問題点を克服する手段として、D.Geman 等が提案している Constrained Optimization という手法[7]がある。これはアニーリングによって温度 T を下げて行く際に、アニーリングスケジュールと一定の関係を保ちながら特定の V_{ci} の値を強制的に増加させて行く手法である。すなわち 禁止したい C_i に対応する V_{ci} のみにλなる係数を乗じ、このλを次の条件式を満足させるように増加させるというものである。

$$T^{-1} \cdot \lambda \leqq \text{const} \cdot \ln(k) \qquad (k : \text{sweep number}) \tag{18}$$

したがって、アニーリングスケジュールを

$$T(k) = \frac{T_0}{\ln\{\alpha(k-1)+e\}} \tag{19}$$

とすれば、λは

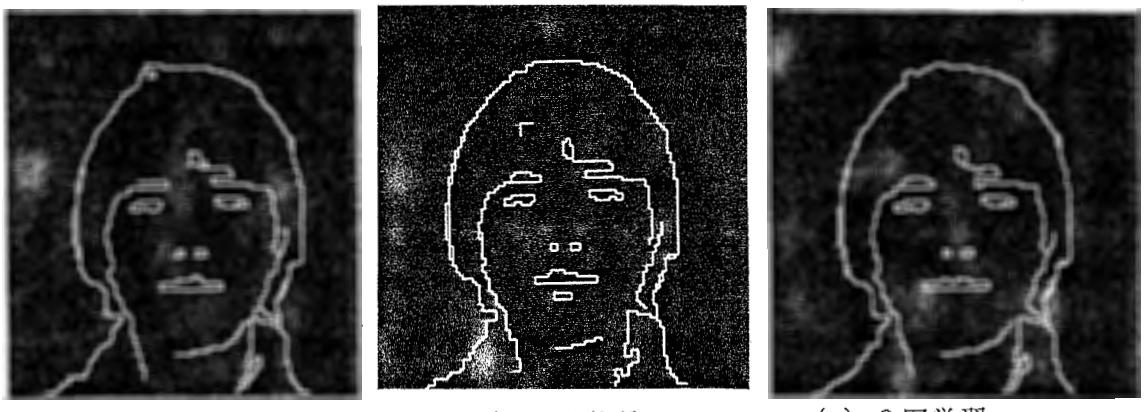
$$\lambda(k) \leqq \frac{\text{const} \cdot T_0 \cdot \ln(k)}{\ln\{\alpha(k-1)+e\}} \tag{20}$$

となり、この条件式を満足する範囲でλを増加させれば、C_i を禁止することができる。

図７に、Constrained Optimization を導入して抽出した輪郭線を示す。 const =1/30, T_0=50, α=100 とし、λを次式のように定めて輪郭線の抽出を行った。

$$\lambda(k) = \frac{\text{const} \cdot T_0 \cdot \ln(k)}{\ln\{\alpha(k-1)+e\}} \tag{21}$$

（ａ）は学習前，（ｂ）は１回学習後，（ｃ）は２回学習後のエネルギーパラメータにより各々 1000回 の状態変化を行った結果得られた輪郭線画像である。



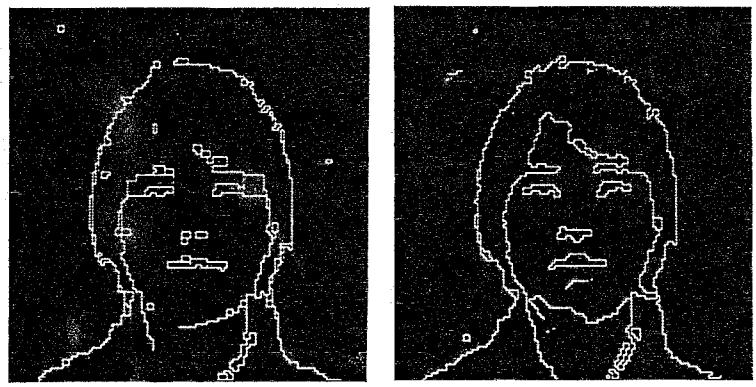
(a) 学習前　　(b) １回学習　　(c) ２回学習

図７．Constrained Optimization を用いた輪郭画像

11.4.5 並列計算機ＮＣＵＢＥによる高速化

ＮＣＵＢＥは、１０２４個までのプロセッサをハイパーキューブ状に接続できる並列計算機である。シミュレーテッドアニーリングを施しながら繰り返し計算を行い、さらに学習を行うには多大な時間を必要とするが、ＭＲＦモデルは局所並列演算が可能であり、ＮＣＵＢＥを用いることにより劇的に計算時間を短縮することができる。

図８に 11.4.4 節と同じアルゴリズム(const=2/30)によって抽出した輪郭線を示す。画素数が１２８×１２８であることから、１２８個のプロセッサを用いることとし、各々に横１ラインの処理を割り当てた。各ラインへのノードの割付は、１次元鎖[8] の方法を用いた。ＶＡＸ８６５０においては 1000回 の状態変化を行うのに約7600秒を要するが、ＮＣＵＢＥでは 約710秒であった。したがって、ＮＣＵＢＥを用いることにより実行時間を 約1/10以下に短縮することができた。



(a) 学習前　　(b) ３０回学習

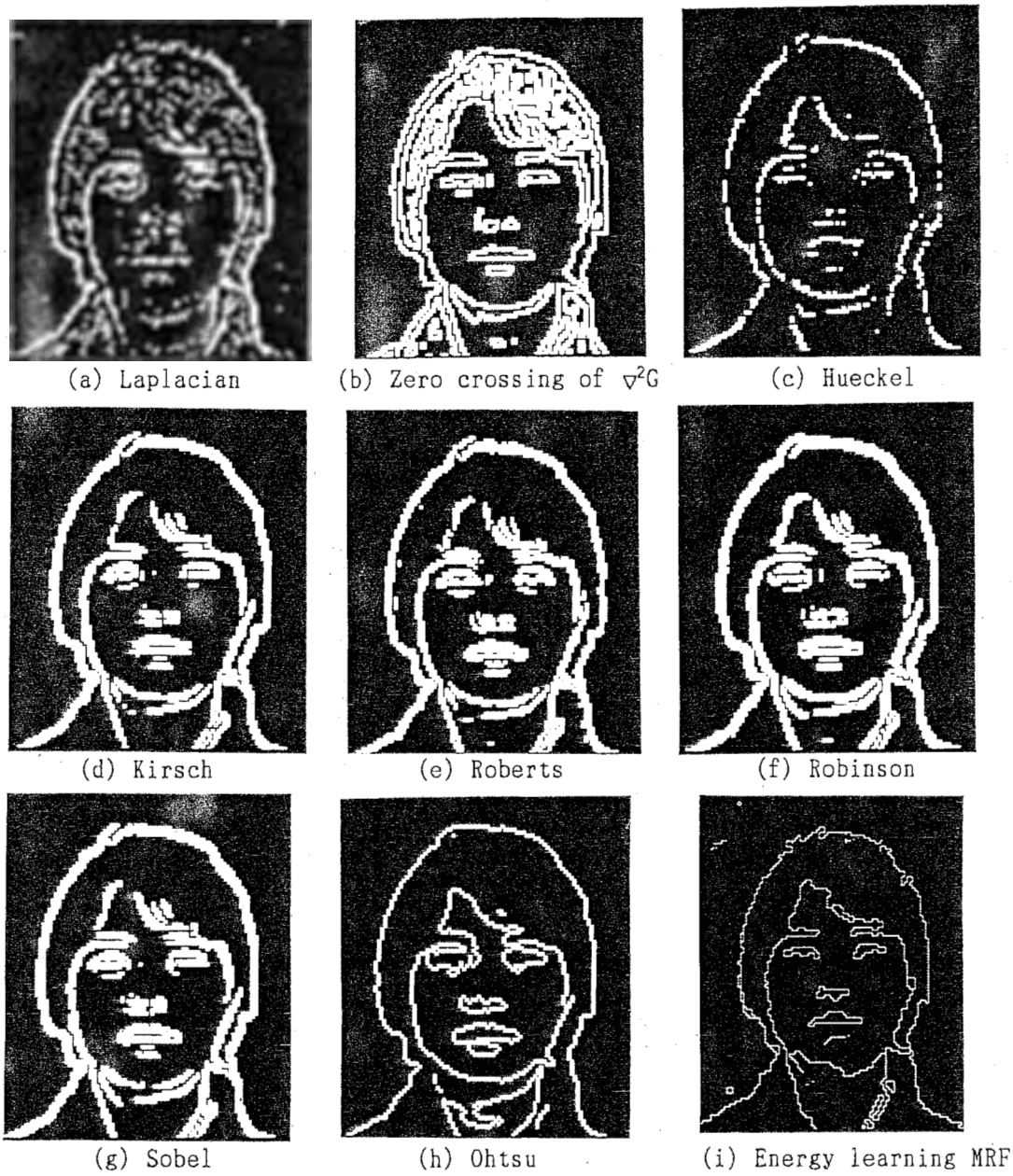
図８．ＮＣＵＢＥを用いて抽出した輪郭画像

11.4.6 従来手法との比較

本手法で得られた輪郭線画像がどの程度の性能であるかを知る目的で、いくつかの輪郭線抽出法との比較を行ってみた。図９に各々の方法で抽出した輪郭線画像を示す。（ｂ）のゼロ交差および（ｈ）の Ohtsuの方法以外は、閾値が必要なので、試行錯誤により輪郭線の連結性が保たれる範囲でできるだけ雑音が少なくなる点に設定した。なお、（ｂ）以外は、通商産業省電子技術総合研究所によって開発されたグラフィックライブラリであるＳＰＩＤＥＲを用いた。

（ａ）は３×３の Laplacianオペレータを用いて求めた輪郭線であるが、非常に連結性が悪い。（ｂ）は ６×６の $\nabla^2 G$ フィルタをかけた後にゼロ交差を求めたもの[9]であるが、雑音が多い。（ｃ）は Hueckelオペレータを用いて求めた輪郭線であるが、連結性が良くない。（ｄ）は Kirschオペレータ，（ｅ）は Robertsオペレータ，（ｆ）は Robinsonオペレータ，（ｇ）は Sobelオペレータを用いて求

めた輪郭線であるが、閾値を調節しても雑音が消える前に輪郭線が切れてきている。（ｈ）は Ohtsuの方法により濃淡画像を２値化した後に、境界を追跡して輪郭線を求めたものであるが、雑音が少なく，かつ輪郭線の連結性も良好であり、比較的良好な輪郭線画像が得られている。（ｉ）は 11.4.4 項において得られた輪郭線画像である。



(a) Laplacian　(b) Zero crossing of $\nabla^2 G$　(c) Hueckel

(d) Kirsch　(e) Roberts　(f) Robinson

(g) Sobel　(h) Ohtsu　(i) Energy learning MRF

図９．従来手法により求めた輪郭画像

11.5 考察

11.4.5 項 で行った比較からもわかるように、エネルギー学習をするＭＲＦモデルによって良好な輪郭線が得られた。従来手法が単に濃度変化のみで輪郭線を求め

ているのに対して、ＭＲＦモデルはさらに輪郭線同士の相互作用までも操作することができ、平行二重線（Parallel）や端点（ending）等を必要に応じて禁止したりできるという優れた特徴を持っている。この性質を用いて端点や平行二重線を制限した輪郭線検出を行うことができた。

しかしＭＲＦモデルには、直接的には対象画像の全体的な構造は無く、局所的状態変化の繰り返しの結果としての間接的な全体しか存在しない。したがって、得られる輪郭線画像は、ある時点で各クリークにとって最も都合の良い状態の集合でしかなく、ひとつの構造としての"輪郭線"には必ずしもなっていない。いわばこのような局所的な拘束条件だけを持ち全体的な拘束条件（構造）を持たないところにひとつの限界があると思われる。そこで我々は、画素数，すなわち"画像を表現する粗さ"が異なる複数のレベルを設け、その各々に対してＭＲＦモデルを適用し、さらに各レベル間に相互作用を有する"階層的ＭＲＦモデル"を提案する。

図１０に階層的ＭＲＦモデルの概念図を示す。（ａ）は階層間の対応を表す概念図である。粗いレベルにおいては１つの画素が、細かいレベルにおいて同じ位置にある複数の画素に対応する。図の例では、レベル n+1 の１画素がレベル n の４画素に対応している。一方、（ｂ）は階層間のクリークの対応の例である。先に粗いレベル(level n+1) で輪郭線を求めておき、それより１段細かいレベル(level n) で輪郭線を求める際に、level n+1 でのクリークの要素と対応するクリーク要素のエネルギーが低くなるように条件づけて計算を行う。すなわち、0 から N-1 までのＮ個のレベルからなる階層的ＭＲＦモデルにおいて、level n におけるクリークを $C_n = \{c_n, n=1,2,\cdots,N\}$、状態配置 ω に対してクリークの要素 c_n ごとに与えられる局所的エネルギーを $V_n^*(\omega, c_n)$、１または微小な値 ε をとる係数を A とすると、実際にエネルギー計算に用いる局所的エネルギー $V_n(\omega, c_n)$ は次式のように表すことができる。

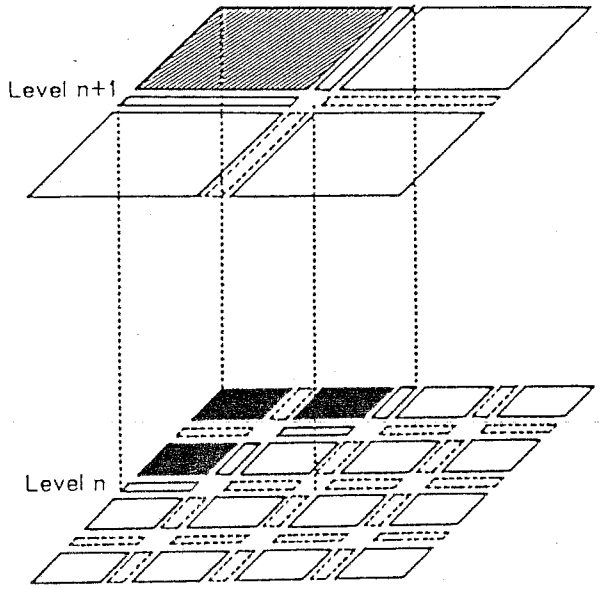
$$V_n(\omega, c_n) = V_n^*(\omega, c_n) \cdot A \qquad (22)$$

$$A = \varepsilon \ll 1 \quad \text{if} \quad c_n \in C_n$$
$$A = 1 \quad \text{otherwise}$$

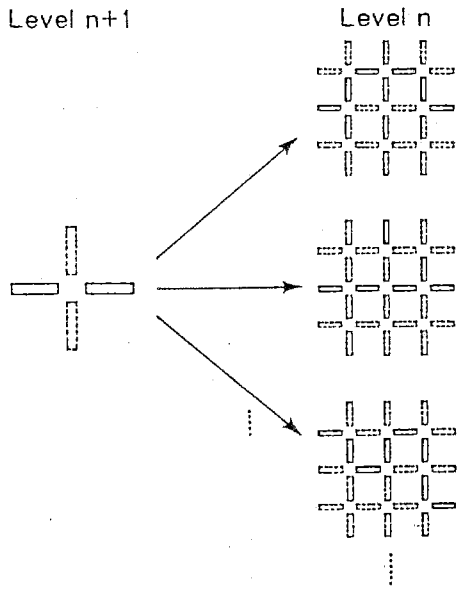
この手法を用いれば、状態変化の局所性を保ちつつ、粗いレベルで求めた全体的構造が細かいレベルの輪郭線画像においても反映されることになる。

さらに、階層的ＭＲＦモデルにおいて粗いレベルの輪郭線を求める際に T.Inui 等が提案している"特徴点検出アルゴリズム"[10] を用いることによって、より正確な輪郭線を求めることができると考えられる。このアルゴリズムは、複数のサイズのエッジオペレータを組み合わせて輪郭線で曲率の大きな部分、分岐点、端点、交点等を同時に抽出するものである。このアルゴリズムによって特徴点を検出し、特徴点の間をＭＲＦモデルを用いて補間して行くのである。具体的には、特徴点が検出されたサイトは特徴点の形状に対応するラインプロセスの配置のポテンシャルを下げておく。その上で "conntinuation" のエネルギーを低くし 逆に "ending" のエネルギーを高くしてやれば、特徴点間をできるだけ直線的に補間することがで

きるものと考えられる。



(a) 階層間の対応



(b) クリークの対応の例

図１０．階層的ＭＲＦモデルの概念図

11.6 おわりに

本章では、エネルギー学習をするＭＲＦモデルを用いて自然画像の輪郭線を抽出する試みについて報告した。実験の結果、シミュレーテッドアニーリング，エネルギー学習， Constrained optimization 等の手法を用いて良好な輪郭線画像が得ら

れた。さらに画像の全体的構造を反映する手法として "階層的ＭＲＦモデル" を提案した。階層的ＭＲＦモデルの実証が今後の課題である。

11.7 文献

(1) L.S.Davis: "A survey of edge detection techniques", Computer Vision, Graphics, and Image processing, Vol.4, pp.248-270(1975)

(2) R.M.Haralick: "Edge and region analysis for digital image data", Computer Vision, Graphics, and Image processing, Vol.12,pp.60-73(1980)

(3) P.K.Sahoo and et. al.: "A survey of thresholding Techniques", Computer Vision, Graphics, and Image processing, Vol.41,pp.233-160,(1988)

(4) S.Geman and D.Geman: "Stochastic relaxation, Gibbs distributions and the basian restoration of images", IEEE Trans., Vol. PAMI-6,pp.721-741 (1984)

(5) D.Geman: "Stochastic model for boundary detection", Image and vision computing, Vol.5, No.2, pp.61-65(1987)

(6) M.Kawato and et. al.: "Information processing of image and neural network models", J. of Japanese society for artificial intelligence, Vol.4, No.2(1989)

(7) D.Geman and et.al.:"Boundary detection by constrained optimization", preprint(1988)

(8) T.Ikeda and et. al.: "Energy learning in neural network model which reconstracts image from noizy data", ITEJ Technical Report, Vol.12, No.14, pp.31-36(1988)

(9) D.Marr: "Vision", Freeman(1982)

(10) T.Inui and S.Miyake: "Structural shape description and a model of human visual memory",IEC Technical Report,Vol.MBE88-107,pp.91-98(1989)

(11) 本郷,川人,乾,三宅:輪郭線抽出をする局所並列確率アルゴリズムのエネルギー学習,電子情報通信学会春季全国大会(1989.3.29 発表予定)

(12) 本郷,川人,乾,三宅:エネルギー学習をする局所並列確率アルゴリズムを用いた輪郭線抽出,電子情報通信学会ＭＥとバイオサイバネティクス研究会 (1989.3.15 発表予定)

１２．おわりに

ＡＴＲにおける視覚・認知機構研究における並列処理計算機ＮＣＵＢＥの利用について具体的事例を中心にまとめた。並列処理計算機、ことに疎結合型については従来の計算機利用とは異なる発想が要求される。例えば、導入の際に種々の応用例やユーティリティが組み込まれているわけではなく、データや問題の並列性を有効に利用しようとするユーザー群によって徐々にそれら応用例が蓄積されて初めて汎用性を身につけていく。この意味でユーザー主導によってのみ存在価値が高まる新しい性格の商品であると言えよう。本稿によって並列的計算原理の有効性とそれを実際に動かす道具に関する認識が高まることを期待する。

視覚・認知機構研究においては従来の計算機でモデルをシミュレーションすることも十分可能であり、手中の道具立てを念頭においた研究に限界があることも事実である。しかしながら、画像や音声パターンといった実世界の超並列データを高速で扱える道具の供給は、単なる高速化という「研究の量的変化」だけでなく会話的、実時間的シミュレーションによる「研究の質的変化」にも大きく寄与する。基礎研究で培われた並列計算原理の実証が可能となり、マシンアーキテクチャにとらわれないさらに自由な発想に基づく原理やモデルの構築が促進されるからである。

ＮＣＵＢＥの利用は上述の問題意識を持たせてくれた最初の１ステップである。今後、生理学や心理学の実験、計算理論、計算機工学、デバイスなど様々の分野が協調し合う学際研究が有効に機能し、新たな成果が生み出されることを期待する。

国内１号機のＮＣＵＢＥを利用した研究の最初の１歩を踏み出すにあたり大変多くの方々の助力と支援がありました。ここに以下の方々に深甚なる謝意を表します。

ＮＣＵＢＥ導入を支援し研究の機会を与えていただいたＡＴＲ視聴覚機構研究所葉原耕平会長、淀川英司社長、梅田三千雄前視覚研究室長（現ＮＴＴ）に感謝いたします。ＮＣＵＢＥ選定にあたり助言をいただいた早稲田大学村岡洋一教授、事前評価実験を助けていただいたＮＨＫ放送技術研究所伊藤崇之研究員に感謝します。第４章の研究では、名古屋大学鳥脇純一郎教授、安田孝美助手に御助力をいただきました。第５章の研究は、ＮＴＴヒューマンインターフェース研究所高橋裕子研究員を中心に実施されたものであり同所塩昭夫主任研究員にも多大の助言を賜りました。第６章の研究には滋賀医科大学可児一孝教授の御助力がありました。第７章について研究の機会を与えていただき御助言をいただいた東京大学鈴木良次教授に深謝します。第８章はＮＨＫとの共同研究によるものであり、尽力いただいたＮＨＫ放送技術研究所日下秀夫前部長、福島邦彦主任研究員に感謝します。第９章、１０章、１１章について有益な議論をいただいた認知機構研究室の研究員諸氏に厚く感謝致します。ＮＣＵＢＥ納入時に環境整備と立ち上げを助けていただいた住商電子システム山上裕司氏に御礼申し上げます。第３章の一部について御助力いただいた視覚研究室城和貴研究員に感謝します。国外では、いつも助言をいただくカリフォルニア工科大学の Prof.G.C.Fox ならびに Prof.J.Hopfield に感謝致します。最後に、いつも研究のサポートをしていただく岸本洋一課長をはじめとする当所企画課の皆様に感謝いたします。

ＮＣＵＢＥ　操作マニュアル

ＦＥＢ．２０．１９８９

ＶＥＲＳＩＯＮ　１．０

目次

第１章　システムメンテナンス
　１・１　システム立ち上げ・立ち下げ
　　１・１・１　システム立ち上げ
　　１・１・２　システム立ち下げ
　１・２　運転環境
　１・３　テープバックアップ
　　１・３・１　ＴＢＲ　ＭＯＤＥ
　　１・３・２　ＲＡＭ　ＭＯＤＥ
　１・４　ディスク管理
　１・５　ユーザー管理
　１・６　ファイルシステム＆プロテクション
　　１・６・１　ファイルシステム構造
　　１・６・１　ファイルシステム構造
　　１・６・２　ファイルプロテクション
　１・７　システム・ディスク

第２章　ソフトウェア　システム概念
　２・１　オペレーティング　システム
　　２・１・１　ホスト　オペレーティング　システム
　　２・１・１・１　ＲＯＭモニター
　　２・１・１・２　ＲＡＭモニター
　　２・１・１・３　ＡＸＩＳ
　２・２　エディタ
　　２・２・１　ＮＭＡＣＳ
　　２・２・１・１　ＮＳＨＥＬＬ　ウィンドウ

第１章　　システムメンテナンス

１・１　システム立ち上げ・立ち下げ

１・１・１　システム立ち上げ

まず、ＮＣＵＢＥに向かって左側のＢＯＸを　ＣＰＵ　ＢＯＸ　右側のＢＯＸを　ＰＥＲＩＰＨＥＲＡＬＳ　ＢＯＸ と呼びます。

（１）　ＣＰＵ　ボックス内部を上方より観察しパワーユニットが何台実装されているかを確認する。（最大４台まで実装が出来ます。．．現在２個実装されています。）

（２）　ＣＰＵ　ボックスの背面に回り　ＭＩＳＣ、次にパワーユニットの台数だけＭＡＩＮ　をＯＮにし、次に　ＦＡＮ，ＰＥＲＩＰＨＥＲＡＬＳ　とＯＮにする。

※　ブレイカー群は、ＭＩＳＣ　ＭＡＩＮ＃１，ＭＡＩＮ＃２，ＭＡＩＮ＃３，ＭＡＩＮ＃４　ＦＡＮ，ＰＥＲＩＰＨＥＲＡＬＳ、ＭＡＩＮ　と並んでいる。

（３）　ＰＥＲＩＰＨＥＲＡＬＳ　ボックスの背面に回り 上部のブレイカーをＯＮにする。

（４）　ＣＰＵ　ボックスの背面に回り 下部にあるブレイカー群の右端にある黒いブレイカー　ＭＡＩＮ　をＯＮにする。

（５）　ＣＰＵ　ボックスの前面にある　ＰＯＷＥＲ　ＤＯＷＮ のパイロットランプが点灯している事を確認する。

ＡＸＩＳの立ち上げ

（１）　ＰＯＷＥＲ　ＯＮ　というパイロットランプを２秒間押す。

（２）　ファンが作動し、ＰＯＷＥＲ　ＯＮ のパイロットランプが点灯する。

（３）　システムは自動的にセルフチェックに入り、障害が無い場合はログイン可能な状態になる。　障害が有る場合はダイアグノスティクモードに入る。

ダイアグノスティクモードの立ち上げ

（１）　ＤＩＡＧ　というパイロットランプを押す。

（２）　ＰＯＷＥＲ　ＯＮ　というパイロットランプを押す。

（３）　ファンが作動し、ＰＯＷＥＲ　ＯＮ　と　ＤＩＡＧ のパイロットランプが点灯する。

※　ＡＸＩＳ、ダイアグノスティクモード については、第２章以下を参照の事。

１・１・２　システム立ち下げ

システム立ち下げには下記の２通りがある。

① ソフトウェアシャットダウン　　（標準時）
② ハードウェアシャットダウン　　（緊急時）

① ソフトウェアシャットダウン

（１）　必ず、シリアルチャネル　＃０　につながっている端末より操作を行い、スーパーユーザーとしてログインする。

（２）　ｓｈｕｔ　ｏｐｔｉｏｎ ↵　　ｏｐｔｉｏｎ ┌ 未入力・・５分待
　　└ 入力　・・１分単位

ｓｈｕｔ　０↵　ですぐシャットダウンする。

（３）　プロンプト　が　＞　に変わる

（４）　Ｘｄｉｓｋ　Ｖｅｒｉｆｙ　ｄｉｓｋ　↵
（このコマンドはディスクの内容をチェックし、エラーがあればそれを表示する。エラー発生時は、（４）のコマンドを繰り返す。

（５）　Ｘｄｉｓｋ　Ｐｏｗｅｒ　ｄｏｗｎ　ｓｙｓｔｅｍ　↵

（６）　ＣＰＵ　ボックス　のパイロットランプ　ＰＯＷＥＲ　ＯＦＦ　が点灯ファンが３０秒程回り自動的に停止する。　マシン移動時等、完全に電源を切る場合は、両ボックスの全てのメインブレーカを切ってください。

② ハードウェアシャットダウン

※　ハードウェアシャットダウンは、緊急時　及びシステムが完全にハングアップしてしまった時行って下さい。

（１）　ＰＯＷＥＲ　ＯＦＦ　のパイロットランプ を１回押す。
（この状態においてはまだ、ＰＯＷＥＲ　ＯＮのパイロットランプが点灯しています。ここで、シリアルチャネル　＃０につながっている端末より、　Ｘｂ　↵　　でＡＸＩＳが再度立ち上がる。但し、ターミナルには　ｘｄｉｓｋ　ｂｏｏｔ　ｓｙｓｔｅｍ　ａｘｉｓ　と表示されます。）

（２）　再度 、ＰＯＷＥＲ　ＯＦＦ　のパイロットランプ を押すと　ＰＯＷＥＲ　ＯＦＦ　のパイロットランプが点灯し、ファンが３０秒程回り続けた後自動的に停止します。マシン移動時等、完全に電源を切る場合は、両ボックスの全てのメインブレーカを切ってください。

注意）　ＮＣＵＢＥシステムが稼働中に、メインブレーカを切るとシステムはそれをシステムクラッシュとみなし、最悪の場合ファイルシステムが破壊される事がありますので、出来るだけ避けるようにして下さい。

１・２　運転環境

（１）　コンピュータ室を必要とし、室内温度は２５度以下に設定して下さい。

（２）　ＣＰＵ　ボックス　の下方より吸気し、上方より排気を行いますので、これらを塞いだり、また吸気・排気の悪い場所での稼働は避けるようにして下さい。

（３）　ＣＰＵ　ボックス　排気温度が３８度を越えますと、マシンは自動的にシャットダウンを行いますが、温度の高いところでの稼働は絶対に行わないようにして下さい。

１・３　テープバックアップ

ＮＣＵＢＥのテープバックアップには　ＴＢＲ　ＦＯＲＭＡＴ（ＴＡＰＥ　ＢＡＣＫ　ＵＰ　ＡＮＤ　ＲＥＳＴＯＲＥ）　と　ＲＡＭ　ＦＯＲＭＡＴ（ＲＡＭとは、ＮＣＵＢＥシステムの内部にあるＥＰＲＯＭによって稼働するＲＡＭ　ＭＯＮＩＴＯＲ　より使用出来るためその名前がついている。）　の２通りがある。

ＴＢＲ　ＦＯＲＭＡＴは、ＵＮＩＸシステムの　ＴＢＲ　ＦＯＲＭＡＴに相当するものでＡＸＩＳが稼働中に使用できるフォーマットです。

ＲＡＭ　ＦＯＲＭＡＴは、システムが立ち上がってない場合、例えば　ディスクがイニシャライズされた後に、システムを再インストールするなどの目的でバックアップをとっておくためのものです。

従って、出来るだけ両方のフォーマットでバックアップを取ることをお薦めします。

１・３・１　テープ　フォーマット

１・３・１・１　ＲＡＭ　ＦＯＲＭＡＴ時

（１）　テープは３Ｍ製のＤＣ６００Ｈを使用する。

（２）　ｓｈｕｔ　０ ↵　（ＲＡＭ　モードに入る）

（３）　テープをドライブに挿入する。オートローディングが自動的に起動する。

（４）　Ｔａｐｅ　Ｃｅｒｔｉｆｙ　↵

（５）　ターミナルにトラックナンバーなどが表示されます。この処理は約一時間かかります。

（６）　プロンプトが戻ってきたら、Ｔａｐｅ　Ｕｎｌｏａｄ　↵

（７）　テープドライブで音が鳴ると、テープがドライブより取り出せます。

１・３・１・２　ＴＢＲ　ＦＯＲＭＡＴ時

（１）　スーパーユーザでＬＯＧＩＮする。

（２）　ｔｂｒ　－ｃ　↵　（約１時間所要）

１・３・２　ＴＢＲ　ＭＯＤＥ

（１）　スーパーユーザーとしてログインする。

（２）　ｃｄ　／／　↵　　でルートディレクトリを管理するディスクネームを確認する。

（３）　テープをドライブに挿入する。オートローディングが自動的に起動する。

（４）　ｔｂｒ　－ｂ　／／ディスクネーム　ａｓ　／／テープ上のディスクネーム　↵
（テープ上には新しいネームをつけることが出来るが、通常の場合はディスクネームをつけるようにする事）

（５）　もし、１本のテープに入らなかった場合（ｔａｐｅ　ｎｏｔ　ｆｏｒｍａｔ）は次のテープをドライブに挿入して下さいというメッセージがターミナルに表示されますので次のテープをドライブに挿入して下さい。

（６）　この２番目のテープに以前バックアップを取っていた場合　ＡＰＰＥＮＤ？と聞いてきます。　最初から上書きしてしまう場合は　Ｎ、　そうでない場合は　Ｙ　を入力する。

（７）　プロンプトが表示されたら終了

（８）　バックアップが終了したら、ｔｂｒ　－ｕ　↵

（９）　テープドライブで音が鳴ると、テープがドライブより取り出せます。

※　ＴＢＲ　は一つのディスクファイルシステムがテープの記憶容量（６５Ｍ）を越える場合でも、２つ以上のテープに連続して記憶する機能をサポートします　（１９８９年２月現在不良）

１・３・３　ＲＡＭ　ＭＯＤＥ

（１）　ＡＸＩＳ上でスーパーユーザーとしてログインする。

（２）　ｓｈｕｔ　０　↵　　（ＲＡＭ　モードに入る）
（パイロットランプのＰＯＷＥＲ　ＯＦＦを一度押してもＲＡＭ　ＭＯＯＤに入ることが出来る。）

（３）　テープをドライブに挿入する。オートローディングが自動的に起動する。

（４）　Ｔａｐｅ　Ｂａｃｋｕｐ　ｆｉｌｅｓ　／／ディスク　ファイルネーム　↵

（５）　ａｓ　／／テープ　ファイル　ネーム　↵
（テープ上には新しいネームをつけることが出来るが、通常の場合はディスク　ファイルネームをつけるようにする事）

（６）　ａｐｐｅｎｄ？　と聞いてきます。　最初から上書きしてしまう場合は　Ｎ、　そうでない場合は　Ｙ　を入力する。

（７）　バックアップが終了したら、Ｔａｐｅ　Ｕｎｌｏａｄ　↵

（８）　テープドライブで音が鳴ると、テープがドライブより取り出せます。

１・４　ディスク管理

通常、ＵＮＩＸシステムでは、システムを立ち上げる時点においては、ファイルシステムのチェックを行いますが，ＡＸＩＳはそれを省略しています。従って、ユーザーによるファイルシステムのチェックを推奨します。

（１）　ｓｈｕｔ　０　↵　　（ＲＡＭ　モードに入る）

（２）　Ｘｄｉｓｋ　Ｖｅｒｉｆｙ　↵
（このコマンドはディスクの内容をチェックし、エラーがあればそれを表示する。エラー発生時は、（２）のコマンドを繰り返す。）

（３）　ＡＸＩＳは、　Ｘｄｉｓｋ　Ｂｏｏｔ　ｓｙｓｔｅｍ　↵　　でブートします。

１・５　ユーザー管理

ユーザーの新規登録、削除、一覧、パスワード設定変更、ユーザー情報権利変更等の操作は、スーパーユーザーのみが出来ます。下記にその方法を示します。

（１）　まず、スーパーユーザーとしてログインする。

（２）　ｅｄｕｓｒ　↵

これで、ｅｄｕｓｒというユーザー管理のユーティリティプログラムが起動し、プロンプトが　＠　となる。　この時点でコマンド一覧表が見たければ　Ｈｅｌｐ　↵　で見れます。　下記に、そのメニューを紹介します。これらのコマンドは、　各コマンドの最初の英文字キャラクターをタイプするだけでかまいません。
また、各コマンドを使用中にキャンセルしたい時は、Ｃｔｌ　Ｕ　で抜けれます。
Ｃｔｌ　Ｃ　でｅｄｕｓｒ自体を終了させます。

①　新規登録

Ａｄｄ　ｕｓｅｒ　＜ＵＳＥＲ　ＮＡＭＥ＞

このコマンドを使用するとこの新規登録されるユーザーに関する情報及びログイン条件を次のように聞いてきます。

１）　Ｐａｓｓｗｏｒｄ　？　　パスワードを入力して下さい。

2) Superuser ? スーパーユーザーですか？ Y／Nで答えて下さい。
もし、Y なら USER DATA？以降の質問に行きます。
もし、N なら次の質問に行きます。

3) Create links ? このユーザーはリンクファイルを作る権利があるか？

4) Create directories ? このユーザーはディレクトリを作る権利があるか？

5) Change memory size ? このユーザーはメモリーサイズを変更する権利があるか？

6) Kill process ? このユーザーはプロセスを消滅する権利があるか？

7) Create process ? このユーザーはプロセスを生成する権利があるか？

8) Delete links ? このユーザーはリンクファイルを削除する権利があるか？

9) Create files ? このユーザーはファイルを作成する権利があるか？

注意： 上記の７個の質問には、必要がない限り Y を入力して下さい。必要となる例としては、ゲストとして部外者が使用する場合がありますが、この場合システム管理の意味で、Create process、Kill process 以外を N で答える場合が考えられます。

10) user data ? このユーザーに関する情報を英文字で最１２８文字登録

11) initial program ? ／BIN／NSH と 入力して下さい

12) file preface ? このユーザーにとってのルートディレクトリトリを入力。通常何も入力せずそのままリターンする。

13) working directory ? このユーザーのホームディレクトリを入力

② パスワード変更

Change password for user <USER NAME>

③ ＥＤＵＳＲ　コマンド一覧表

　　ｈｅｌｐ

④ 現在登録されているユーザーの表示

　　ｌｉｓｔ　ｕｓｅｒｓ

⑤ ログイン条件変更

　　ｍｏｄｉｆｙ　ｕｓｅｒ　＜ＵＳＥＲ　ＮＡＭＥ＞

⑥ ＥＤＵＳＲ終了

　　ｑｕｉｔ

⑦ ユーザー削除

　　ｒｅｍｏｖｅ　ｕｓｅｒ　＜ＵＳＥＲ　ＮＡＭＥ＞

⑧ ログイン条件表示

　　ｓｈｏｗ　ｕｓｅｒ　＜ＵＳＥＲ　ＮＡＭＥ＞

１・６　ファイルシステム＆プロテクション

ＡＸＩＳは大部分においてＵＮＩＸを採用していますが、ファイルシステムの概念においては、多少異なる形式を採用しています。普段の操作においては、あまり気にする必要はありませんが、システムを理解する上で重要ですので、下記にその概要を示します。

１・６・１　ファイル　システム概要

基本的には，ＵＮＩＸと同様のツリー構造を採用していますが、特に異なる点の詳細を下記に示します。

（１）　ｍｋｄｉｒ　について

新しいディレクトリーは指定すれば、強制的に作成されますので　ｍｋｄｉｒ　というコマンドを使用しません。　つまり、操作時において、既にそのディレクトリが存在しているものと仮定し操作を行うことにより、ディレクトリが作成されます。

（２）　ファイル　ディレクトリネームの変更

ＵＮＩＸにおいては、ファイルネームの変更には　ｍｖ　コマンドを使用しますが、ＡＸＩＳには、ｃｎ（ｃａｈｎｇｅ　ｎａｍｅ）というコマンドがあり、これがディレクトリネームの変更にも使用できます。　ｍｖも勿論使用できます。

（３）　ＰＲＥＦＡＣＥ　ＤＩＲＥＣＴＯＲＹ　について

ＵＮＩＸに存在しないプリフェース　ディレクトリ　と言う概念があります。これは、各ユーザーにとってのルートディレクトリを強制的に設定可能にするものです。　例えば、あるユーザーが　／ＵＳＲ／ＮＣＵＢＥ／ＧＵＥＳＴ　というプリフェース　ディレクトリをもっている場合、このユーザーが　ｃｄ／　としてルート　ディレクトリを指定した場合でもそれは、／ＵＳＲ／ＮＣＵＢＥ／ＧＵＥＳＴ　であると言うことになります。つまり、真のルート　ディレクトリへのアクセスを禁止することが可能になります。このプリフェース　ディレクトリは、ｅｄｕｓｒ　で設定が出来ます。

（４）　ＦＩＬＥ　ＰＲＯＴＥＣＴＩＯＮ　について

ＵＮＩＸにおけるファイル　プロテクションは、ファイル一つ一つに設定され、それは、ＲＥＡＤ、ＷＲＩＴＥ、ＥＸＣＵＴＥ、及び　ＤＥＬＥＴＥに対するプロテクションをされますが、ＡＸＩＳはこれを更に拡張し、ファイルのパスネームそのものに対するディレクトリ　プロテクションを設定する事が可能です。例えば、システムエリアのディレクトリには、一般ユーザーのアクセスそのものを禁止したり、アクセスを認めているファイルのリストを取る事が出来ても、そのディレクトリにはファイルを作成する事を禁止する事が可能です。　詳細は、次の１・６・２を参照して下さい。

１・６・２　ファイル　プロテクション

ｌｓ　－ｌ　↵　により得られるファイルのプロテクショオンには、下記のものが表示されます。

Ｄ：ＲＣＡＤＥＴＬＮ　　Ｆ：ＲＷＥＤ

Ｄ：　は、ディレクトリ　プロテクションを意味し、Ｆ：　は、ファイル　プロテクションを意味します。これらプロテクションは、そのファイルのオーナー及びスーパーユーザのみが無視が可能です。詳細は下記の通りです。

（１）　ディレクトリ　プロテクション

Ｒ：　ディレクトリ内容の表示
Ｃ：　ディレクトリ内で新しいエントリーの作成
Ａ：　アクセスライツの変更
Ｄ：　ディレクトリの削除
Ｅ：　ディレクトリ内でエントリーの削除
Ｔ：　ファイルルックアップ用のディレクトリのトラバース
（ｔｒａｖｅｒｓｅ　ｄｉｒｅｃｔｏｒｙ）

Ｌ：　リンクの作成と変更
Ｎ：　ディレクトリ内での名前変更

（２）　ファイル　プロテクション

Ｒ：　ファイルの読み込み
Ｗ：　ファイルの書き込み
Ｘ：　ファイルの実行
Ｄ：　ファイルの削除

※　これらファイル及びディレクトリプロテクションの設定は　ＣＨＭＯＤ　というコマンドにより行って下さい。

１・７　システム　ディレクトリ

ＡＸＩＳは下記のディレクトリを管理しています。

/dev　　ｼｽﾃﾑﾃﾞﾊﾞｲｽ ﾌｧｲﾙ ﾃﾞｨﾚｸﾄﾘ

/bin　　AXISｼｽﾃﾑ実行可能 ﾌｧｲﾙ ﾃﾞｨﾚｸﾄﾘ

/phy　　ﾊﾞｯﾄﾞｾｸﾀｰ、ﾃﾞｨｽｸﾃﾞｰﾀ、ﾌﾘｰﾌﾟｰﾙ、その他ｼｽﾃﾑﾃﾞｰﾀﾌｧｲﾙ ﾃﾞｨﾚｸﾄﾘ

/sys　　ｼｽﾃﾑ情報用のﾃﾞｨﾚｸﾄﾘを持つ ﾃﾞｨﾚｸﾄﾘ

- /acct　　各ﾕｰｻﾞｰのｱｶｳﾝﾄﾌｧｲﾙを持つ ﾃﾞｨﾚｸﾄﾘ
- /bin　　OSのｵﾌﾞｼﾞｪｸﾄ、ﾛｸﾞｲﾝ、ｽﾌﾟｰﾗｰ、EDUSRを持つ ﾃﾞｨﾚｸﾄﾘ
- /games　　ｹﾞｰﾑ　ﾃﾞｨﾚｸﾄﾘ
- /help　　ﾍﾙﾌﾟﾒｯｾｰｼﾞ用 ﾃﾞｨﾚｸﾄﾘ
- /spool.log　　ｽﾌﾟｰﾗｰ ﾛｸﾞｲﾝ ﾌｧｲﾙ
- /startup　　ｼｽﾃﾑｽﾀｰﾄｱｯﾌﾟ ﾌｧｲﾙ

/startup.log　ｽﾀｰﾄｱｯﾌﾟﾛｸﾞﾌｧｲﾙ

/sysmsg　エラーメッセージ ディレクトリ

/usrlist　ユーザーのリストを持つファイル

/usr　ユーザーエリア

/bin　システムとは直接関係ないが便利なコマンドである

/ed、vi	エディター
/einit	イーサーネットボードド初期化
/atgrafinit、matroxinit	簡易グラフィクボードド初期化
/ftp、ftprcv	イーサーネットプログラム
/kc、nm	nm キーマップ コンパイラ
/nroff	テキスト フォーマッティング
/ud	テキストファイルをバイナリーに変換する。hdと使用しバイナリーファイルのエディトを可能にする。

/ncube　NCUBE ワーキング ディレクトリー

/sample	サンプルプログラム
/handbook	NCUBE HANDBOOK ファイル
/graph、sgraph	グラフィックスを使用するサンプルプログラム

/spool/lpt　スプーラーにより、プリンター待ちのファイルをバッファリングするエリア

/sysv2　UNIX SYSTEM-V サポート ディレクトリ

第２章　　ソフトウェア　システム概念

ＮＣＵＢＥ　システムのソフトウェアは基本的に下記のように分類されます。

オペレーティングシステム
ＮＳＨＥＬＬ　シェル環境
エディター
アセンブラ
高級言語コンパイラー、リンカー、ロケータ
デバッガー
システムコール
ハイパーキューブ操作
各種ライブラリ
その他ユーティリティ　プログラム

各々の詳細を下記に示します。

２・１　オペレーティング　システム

ＮＣＵＢＥ　システムは、ハイパーキューブ　アレイを中心とし、それより放射状に接続されるＩ／Ｏ　ペリフェラル　サブシステムにより構成されています。ハイパーキューブアレイ、及び　ペリフェラル　サブシステムはマスター・スレーブという関係ではなく、各々が独立し、ホストよりのコントロールバス、及び　ハイパーキューブからのメッセージにより連絡しています。従って、各々個別のソフトウェア　システムを保持しています。詳細を下記に示します。

ハイパーキューブ　アレイ　－－　ＶＥＲＴＥＸ　ＳＹＳＴＥＭ

ＮＣＵＢＥにおけるハイパーキューブは、最大２＊＊１０までのノードにより構成されます。従って、最大１０次元のハイパーキューブを構築できます。これらのノードは、疎結合された３２ビットのプロセッサであり、各々が自己のオペレーティングシステムを持ちます。　このシステムをＶＥＲＴＥＸと呼びます。

Ｉ／Ｏペリフェラル　サブシステム

① ホストシステム　・・・　ＲＯＭモニタ、ＲＡＭモニタ、ＡＸＩＳ、ＶＯＲＴＥＸ

ホストシステムは、インテル８０２８６を中心としコミュニケーション用にＮＣＵＢＥプロセッサを１６個搭載したシステムで、シリアルチャネルを８本，ＥＳＭＤディスクインターフェースを４本、ＳＢＸ規格１６ビット　パラレルポートを３本を持ちます。

ホストシステムは、下記の機能をサポートします。

1. システム全体の立ち上げ、立ち下げ、ハイパーキューブの初期化、その他サブシステムの初期化

2．　ユーザーインターフェース、プログラム入力、コンパイル、アセンブル、

リンク、デバッグ、プログラム実行、ノードプログラム　ローディング、データＩ／Ｏ、その他．．．

ホストシステムは、システム全体の管理者として機能し、更にユーザーインターフェースを行います。また、ホストシステムでは、ＲＯＭモニタ、ＲＡＭモニタＡＸＩＳ、ＶＯＲＴＥＸ　という４つオペレーティングシステムが稼働します。この中で、ＶＯＲＴＥＸがハイパーキューブとのコミュニケーションを行うシステムであり、ＲＯＭモニタ、ＲＡＭモニタ、ＡＸＩＳは階層構造となっているＵＮＩＸ形式のオペレーティングシステムです。（詳細は２・１・１参照）

② グラフィクシステム　・・・　ＲＯＭモニタ、ＧＲＡＰＨＯＳ

グラフィクシステムは、インテル８０１８６、インテル８２７３０テキストプロセッサ、２Ｋ＊１Ｋ＊８ビットのフレームバッファ、日立製ＡＣＲＴＣグラフィクスコントローラ、及び　コミュニケーション　ローカル処理用のためのＮＣＵＢＥプロセッサ１６個により構成されているシステムです。

出力はＲＧＢ　６０ＨＺ高解像度ディスプレイにより行います。さらに、タブレット用にＲＳ２３２Ｃポート、ＳＢＸ規格パラレルポートを２本搭載しています。このポートよりホストシステムとのコミュニケーション、及び、２４ビット構成のフレームバッファを構成します。

グラフィクシステムは下記の機能をサポートします。

1．　ホストシステムよりグラフィクコマンドを受取り、各種グラフィクスをフレーンバッファに出力表示する。

2．　ホストシステムよりラスタデータの入出力を行い、またそれを表示する。

3．　ハイパーキューブ　アレイより直接ラスタデータの入出力を行い、またそれを表示する。

4．　１６個のＮＣＵＢＥプロセッサにより、ラスターデータのローカル処理を行います。例として、ピクセル色変換等、直接ハイパーキューブに担当させるまでもない処理を行います。

グラフィクシステムでは、ＲＯＭモニタ、ＧＲＡＰＨＯＳが稼働します。ＲＯＭモニタがグラフィクシステムの初期化、及び、全体の管理を行います。ＧＲＡＰＨＯＳは、１６個のＮＣＵＢＥプロセッサで各々稼働します。これはＶＥＲＴＥＸとよく似たシステムであり、ユーザープログラムを必要とします。これにより、ハイパーキューブアレイより来たラスターデータをフレーンバッファに転送、その時点、又はその後にラスター処理を行うことが可能です。

③　Ｎチャンネルシステム

Ｎチャンネルシステムは、ハイパーキューブアレイへの直接入出力を可能にします。一つのＮチャンネルシステムより１６本のシリアルチャネルが使用できます。このシリアルチャネルには、例として、ＥＳＭＤディスクインターフェースを取り付けることが可能ですので、１６台のディスクを同時に稼働させ、パラレルディスクインターフェース（ＤＩＳＫ　ＦＡＲＭ）とすることが可能です。外に、Ａ／Ｄコンバータ等により画像入力デバイスとしても使用できます。　ＡＴＲではＣＡＴＶ－ＤＩＧＩＴＡＬ－ＴＲＡＮＳＣＯＤＥを接続してＶＩＤＥＯ－ＣＡＭＥＲＡ入力デバイスを装備しています。ＡＸＩＳにＶＩＤＥＯ関係Ｏ／Ｓが付加される。

④　インターリンクシステム　（ＡＴＲのシステムには未装備）

インターリンクシステムは、複数のＮＣＵＢＥシステムのハイパーキューブアレイを直接結合し、これらアレイのパイプラインシステムを可能にするものです。

⑤　オープンボードシステム　（ＡＴＲのシステムには未装備）

オープンボードシステムは、ユーザーデバイスを直接ハイパーキュブアレイに結合させるボードです。コミュニケーション用に１６個のＮＣＵＢＥプロセッサが搭載され、必要最小限のインターフェース　ロジックが組み込まれていますが、残り　ボード上　約２／３はオープンスペースとなっています。

ここで、ユーザーにとって当面重要なシステムは、下記の通りです。

ＶＥＲＴＥＸ　ＳＹＳＴＥＭ	・・	ハイパーキューブアレイを構成しているＮＣＵＢＥノードプロセッサ上で稼働
ＶＯＲＴＥＸ　ＳＹＳＴＥＭ	・・	ホストシステム上に搭載されているコミュニケーション用ノードプロセッサ上で稼働
ＲＯＭ　ＭＯＮＩＴＯＲ	・・	システム立ち上げ時、ホストシステム上に搭載されているインテル８０２８６上稼働
ＲＡＭ　ＭＯＮＩＴＯＲ	・・	システム立ち上げ時、及びシステムメンテナンス時、ホストシステム上に搭載されているインテル８０２８６上稼働
ＡＸＩＳ	・・	ＮＣＵＢＥシステム稼働中、ホストシステム上に搭載されている８０２８６で稼働
ＧＲＡＰＨＯＳ	・・	グラフィックスシステム上に搭載されているＮＣＵＢＥプロセッサ上で稼働

上記システムをホストシステムのインテル８０２８６上で稼働するオペレーティングシステムと、ＮＣＵＢＥノードプロセッサ上で稼働するノードオペレーティングシステムとに分類し、詳細を下記に示します。

２・１・１　ホストオペレーティング　システム

ホストシステムのＣＰＵであるインテル８０２８６上では、下記の３種類のシステムが稼働します。これらは、階層構造をとっています。

ＲＯＭ　　モニタ
ＲＡＭ　　モニタ
ＡＸＩＳ　オペレーティングシステム

２・１・１・１　ＲＯＭ　モニタ

ＲＯＭモニタは、ホストシステム上のＥＰＲＯＭに格納されており、システムがパワーアップされる時最初に起動されるシステムです。このモニタは、シングルユーザーモードであり、シリアルチャネル０に接続されている端末を使用します。ＲＯＭモニタの基本的機能は、システムのダイアグノスティックスです。
システムを立ち上げた時、ＮＯＲＭＡＬモードにセットされていれば、ＲＯＭモニタはダイアグノスティックスを走らせた後、ＲＡＭモニタを起動します。（ＲＡＭモニタはその後、ＡＸＩＳをブートする事になります。）　もし、ＤＩＡＧモードにセットされていれば、ダイアグノスティックスを終了後、シングルユーザーモードとなり、プロンプト　＄　を端末に表示します。

このモードの設定は、パワーアップ時にＣＰＵ　ＢＯＸ側のＤＩＡＧ　と言うパイロットランプを押し、それが点灯した後ＰＯＷＥＲ　ＵＰ　を押すことによりＲＯＭモニタ　シングルユーザーモードとなります。このＤＩＡＧパイロットランプはトグルスイッチとなっており、もう一度押すと解除されます。

ＲＯＭモニタはメモリーを使用しませんので、非常に簡単なコマンドを持っています。基本的には、ホストシステムに故障の疑いのある場合以外は使用しませんが、ＲＯＭモニタの存在を知り、且つそのレベルからＡＸＩＳ起動までの手順を知る意味で、下記のＲＯＭモニタコマンドが重要になります。
下記にＲＯＭモニターコマンドを示します。

各コマンドは、コマンドネームの最初のキャラクタータイプする事により起動されます。リターンは新しいプロンプト　＄　を表示します。㊞＋Ｃ　により如何なる時でもコマンドを打ち切ることが可能です。コマンドのオペランドは下記の通りです。

ＡＤＤＲ　　２つの４桁の１６進数数字。コロンにより区分される。
最初の数字がセグメントセクタであり、次がオフセットを示す。

ＬＥＮＧＴＨ
ＩＯＡＤＤＲ　この３個は４桁の１６進数数字。
ＶＡＬＵＥ

ＳＥＧ　ＭＡＸ　テストをうけるメモリ内部の６４Ｋセグメントの数

① ｃｏｎｔｉｎｕｅ ↵

”ｄｅｂｕｇ”ストップ後、ディスク　オペレーティングシステムを起動する。

② ｄｉｓｐｌａｙ＜ＡＤＤＲ＞、＜ＬＥＮＧＨ＞ ↵

ＡＤＤＲから　ＡＤＤＲ＋ＬＥＮＧＴＨ－１のメモリ内容を下記のフォーマットで表示する。
ＡＤＤＲ　　ｈｈｈｈ　　・・・　　ｈｈｈｈ　＜ａｓｃｉｉ＞

ここで、ＡＤＤＲはビギニングアドレスであり、ｈｈｈｈは１６ビットワードを１６進数で示す。

③ ｇｏｔｏ　ｒａｍ　ｍｏｎｉｔｏｒ ↵

ＲＡＭモニタがブートされる。

④ ｈｅｌｐ ↵

ＲＯＭモニタコマンドのリストが表示されます。

⑤ ｉｎｐｕｔ＜ＩＯＡＤＤＲ＞ ↵

ＩＯＡＤＤＲにより示されるＩ／Ｏアドレスの値が表示される。
ｌｉｎｅｆｅｅｄはこのコマンドを繰り返し、リターンで終了する。

⑥ ｍｅｍｏｒｙ　ｔｅｓｔ＜ＳＥＧ　ＭＡＸ＞ ↵

ＳＥＧ　ＭＡＸにより示される６４Ｋバイトセグメント数がメモリアドレス　０よりテストされる。

⑦ ｏｕｔｐｕｔ＜ＩＯＡＤＤＲ＞、＜ＶＡＬＵＥ＞ ↵

ＩＯＡＤＤＲで示される　Ｉ／ＯアドレスにＶＡＬＵＥで示される値が書き込まれる。ｌｉｎｅｆｅｅｄはこのコマンドを繰り返し、リターンで終了する。

⑧ ｐｏｗｅｒ　ｄｏｗｎ ↵

システムがパワーダウンされる。

⑩ ▒ｓｅｔ＜ＡＤＤＲ＞ ↵

ＡＤＤＲで示されるメモリーの値が表示され、新しい値をタイプする事により変更が出来る。ｌｉｎｅｆｅｅｄにより、メモリー内部の次のワードが表示され、リターンで終了する。

２・１・１・２　ＲＡＭ　モニタ

ＲＡＭモニタはＲＯＭモニタ同様シングルユーザーであり、シリアルチャネル０に接続されている端末を使用します。ＲＡＭモニタはＥＰＲＯＭに格納されており、起動された後はシステムメモリ上にコピーされます。ＲＡＭモニタは下記の時に起動されます。ＲＡＭシステムの基本的機能は、ジェネラルコマンド、ディスク関係、テープ関係、及び、デバッキングです。ＮＣＵＢＥシステム管理者は必ずＲＡＭモニタの機能を熟知して下さい。

ジェネラルコマンドはコマンドネームの最初のキャラクターをタイプする事により起動されます。ディスク、テープ及びデバッキングコマンドは各々 ▒ｄｉｓｋ ▒ａｐｅ、▒ｄｅｂｕｇ の最初のキャラクターをタイプする事でコマンドが起動します。最初に、 ↵ を入力しますと プロンプト ＞ が出力され、コマンド受け入れ状態を示します。 ▒＋Ｃ は如何なる時でも入力可能でコマンドがキャンセルされます。

ＲＡＭモニタのオペランドは下記に示すとおりです。

ＦＩＬＥＮＡＭＥ	ファイルのパスネーム
ＦＩＬＥ　ＳＰＥＣ	ファイル名、及びディレクトリ名
ＳＣＣＮＵＭ	シリアルコミュニケーション　チャネルＮＯ.
ＳＲＣ　ＡＤＤＲ ＤＥＳＴ　ＡＤＤＲ	ＲＯＭモニタのＡＤＤＲと同じ
ＳＥＣＴＯＲ	ディスクアドレスを示す。２つの１６進数をコロンで句切り示す。最初がシリンダーＮＯ. であり次が、ヘッド及び、セクター（ハイオーダーバイトがヘッドＮＯ、ローオーダーバイトがセクター）を示す。
ＤＩＳＫ　ＮＵＭＢＥＲ	０～３の１０進数で示す、このホストボードに接続されているディスクＮＯ。
ＲＥＣ	インテル８０２８６レジスター名 （ＡＸ、ＢＸ，ＣＸ，ＤＸ，ＢＰ，ＳＩ，ＤＩ， ＳＰ，ＩＰ，ＣＳ，ＤＳ，ＳＳ，ＥＸ）
ＴＳＥＣＴＯＲ	テープアドレス　０～６４ＫＢ　を示す１６進数字によりテープ上のある１０２４バイトを示す。

注意　：　ディスク　０　以外のディスクを選択する場合、ディスクコマンドにある”▒ＮＩＴＩＡＬＩＺＥ　ＣＯＮＴＲＯＬＬＥＲ”を実行して下さい。

ＲＡＭモニタ　コマンド

1）　ａｓｃｉｉ　ｄｕｍｐ　＜ＦＩＬＥＮＡＭＥ＞　↵

ＦＩＬＥＮＡＭＥにより指定されたファイルの内容が下記のフォーマットで表示される。ｏｆｆｓｅｔ　ｈｈｈｈ　・・・　ｈｈｈｈ　＜ａｓｃｉｉ＞
ｏｆｆｓｅｔとは、ファイルの最初からのバイトオフセットであり、ｈｈｈｈは１６進数による１６ビットワードであり、ａｓｃｉｉはアスキーとして解釈された８ワードを示す。但し、表示できない分についてはピリオドとなる。

2）　ｂａｕｄ　ｒａｔｅ　＜ＳＣＣＮＵＭ＞，＜ＶＡＬＵＥ＞　↵

Ｉ／Ｏアドレス　ＩＯＡＤＤＲに格納されているボーレートが、ＶＡＬＵＥに変更される。但し、１０進数として入力する。１９．２ＫＢは１９２００として入力する。

3）　ｃｈａｎｇｅ　ｄｉｒｅｃｔｏｒｙ　＜ＦＩＬＥ　ＳＰＥＣＫ＞　↵

カレントディレクトリを　ＦＩＬＥ　ＳＰＥＣＫに修正する。もし　ＦＩＬＥ　ＳＰＥＣＫ　が、／　より始まる場合は、カレントディレクトリを変更する。もし　ＦＩＬＥ　ＳＰＥＣＫが、それを最初に持たない場合は、カレントディレクトリに追加される。もし、ＦＩＬＥ　ＳＰＥＣＫ　なしで　↵　を行うとカレントディレクトリが表示される。

4）　ｄｉｓｐｌａｙ　＜ＡＤＤＲ＞、＜ＬＥＮＧＴＨ＞　↵

ＡＤＤＲ　から　ＡＤＤＲ＋ＬＥＮＧＴＨ－１　のメモリの内容を下記のフォーマットで表示する。
ＡＤＤＲ　ｈｈｈｈ　・・・・　ｈｈｈｈ　＜ａｓｃｉｉ＞

5）　ｅｘｅｃｕｔｅ　＜ＦＩＬＥＮＡＭＥ＞　↵

エクゼキュータブルである　ＦＩＬＥＮＡＭＥ　により表示されるファイルがＡＸＩＳシングルプロセス環境により実行される。　カレントディレクトリ　と　／ＢＩＮ　がコマンドディレクトリとしてサーチされる。ある一部のコマンドが使用できる。これらは、ｎｍ、ａｓ、及び　ｅｔ等である。

6）　ｆｉｌｅ　ｓｙｓｔｅｍ　ｓｕｂｔｒｅｅ
ｒｅｍｏｖｅ　＜ＦＩＬＥＮＡＭＥ＞　↵

指定されたサブツリーを取り除く。

7) goto rom monitor ↵

rom monitor が起動される。メモリの内容を保証しながらROMコマンドが使用できる。

8) help ↵

RAMコマンドのリスト及びオペランド

9) input <IOADDR> ↵

IOADDRにより示されるI/Oアドレスの値が表示される。
linefeedにより、このコマンドが繰り返され、リターンで終了する。

10) jump <ADDR> ↵

<ADDR> 番地に枝分かれする。

11) list directory <FILE SPECK> ↵

<FILE SPECK> により示されるディレクトリの内容が下記のフォーマットで表示される。

system type、 rights file type、 #byte date create name

12) move memory <SRC ADDR>、<LENGTH>、<DEST ADDR> ↵

SRC ADDR より始まる LENGTH+1のメモリーが、DEST ADDR に移動する。

13) new time set ↵

↵ が入力された後、現在の時刻が表示される。次に、新しい時間を入力するためのプロンプトが表示される。(year、month、day、hour、minutes、second) 入力は10進数で行う。

注意 ・・ 数字入力なしで ↵ を入力すると 0 として入力されるので注意すること。

14) output <IOADDR>, <VALUE> ↵

IOADDR で示されるI/OアドレスにVALUEで示される値が書き込まれる。 linefeedはこのコマンドを繰り返し、[ESC] で終了する。

１５）　ｐｏｒｔ　＜ＳＣＣＮＵＭ＞　↵

指定されたポートにコネクションが行われる。ターミナルエミュレーションが可能になる。Ctrl＋ｓｐａｃｅｂａｒ　でコネクションが中断する。

１６）　ｒｅｍｏｖｅ　＜ＦＩＬＥＮＡＭＥ＞　↵

指定されたファイルがファイルシステムより除かれる。

１７）　ｓｅｔ　＜ＡＤＤＲ＞　↵

ＡＤＤＲ　により示されるメモリーの値が表示され、新しい値をタイプすることにより値を変更することが出来る。ｌｉｎｅｆｅｅｄによりメモリー内部の次のワードが表示され、コマンドを繰り返す。↵　により終了する。

１８）　ｔａｐｅ　＜ＣＯＭＭＡＮＤ＞　↵

”ｔ”　を入力後、ｔａｐｅが表示され次にテープコントロール　コマンドが入力されなければならない。

１９）　ｖｓｙｓｔｅｍ　ｔｅｍｐｅｒａｔｕｒｅ　↵

システムＢＯＸ　内部の温度が摂氏により表示される。

２０）　ｗｔｙｐｅ　＜ＦＩＬＥＮＡＭＥ＞　↵

指定されたファイルの内容がターミナルに表示される。Ctrl＋Ｓ、Ctrl＋Ｑ　によりスクリーンをコントロールする。

２１）　ｘｄｉｓｋ　＜ＣＯＭＭＡＮＤ＞　↵

ｘｄｉｓｋ　を入力後、ｘｄｉｓｋが表示され次にデバッグコントロールコマンドが入力されなければならない。
（次項以降のディスクコマンドを参照の事）

２２）　ｙｄｅｂｕｇ　＜ＣＯＭＭＡＮＤ＞　↵

ｙｄｅｂｕｇ　を入力後、ｙｄｅｂｕｇが表示され次にデバッグコントロールコマンドが入力されなければならない。
（次項以降のデバッキング　モニター　コマンド　を参照の事）

２３）　ｚｔｉｍｅ　↵

現在のクロックタイムが１６進、及び人間が理解できる形で表示される。

デバッキング　モニター　コマンド

デバッキング　モニター　コマンドは、最初に　ｙｄｅｂｕｇ　を入力した後に、下記に示すコマンドの最初のキャラクターをタイプする事により実行される。

1）　ｃｈａｎｇｅ　ｒｅｇｉｓｔｅｒ　＜ＲＥＧ＞

ＲＥＧにより指定されるレジスター値が表示され、新しい値を入力し変更が可能である。ｌｉｎｅｆｅｅｄ　により次のレジスターに対し同コマンドが繰り返されリターンでこのコマンドが終了する。

2）　ｈｅｌｐ

デバッキング　コマンド　及び、オペランドのリストが表示される。

3）　ｒｅｇｉｓｔｅｒｓ

全てのレジスターの現在値が表示される。

4）　ｓｔａｔｕｓ

ディスク　コントローラ　レジスターの現在のステータスを表示する。

5）　ｔｒａｃｅ　＜ＡＤＤＲ＞

最後に実行されたｔｒａｃｅによるＣＳ：ＩＰの値が表示され、新しいＣＳ：ＩＰの入力を待つ。数値なしでリターンすると　ＣＳ：ＩＰの値はそのままになる。そして、ＣＳ：ＩＰの値が更に表示され現在のレジスターの内容（ＡＸ，ＢＸ，ＣＸ，ＤＸ，ＢＰ，ＳＩ，ＤＩ，ＳＰ）の値が表示され、入力待ちとなる。ｌｉｎｅｆｅｅｄを入力すると、ＣＳ：ＩＰに存在するインストラクションが実行され、レジスターの内容が表示され、入力待ちとなる。リターンでＲＡＭ　ＭＯＮＩＴＯＲに戻る。

ディスク　コマンド

ディスクモニターコマンドは、最初にｘｄｉｓｋが入力された後に、下記に示すコマンドの最初のキャラクターをタイプする事により実行される。

1) ascii dump <SECTOR> ↵

SECTORにより指定されたディスクアドレスの内容が、16進数で表示される。表示フォーマットは、ascii dump<FILENAME>コマンドと同様である。

2) boot operating system ↵

事前にシステムクラッシュがあったかチェックを行い、ディスクのベリファイを行い、オペレーティングシステムを新しくブートする。クラッシュ以降のブート等に使用する。 ⇧+Cによりログインを行う。

3) clear disk state ↵

ディスク コントロール レジスターをリセットする。

4) disk format and init <DISK NUMBER> ↵

指定されたディスクがフォーマットされ、初期化される。これは、フィジカルフォーマットであるため、最初に、ディスク上の全てのデータが消去されるという警告メッセージが出力される。

注意 ・・ このコマンドはよほどの事がない限り使用してはならない。

5) format sector <SECTOR> ↵

SECTORにより指定されたセクターのみがフォーマットされる。他はそのままである。このコマンドはよほどの事がない限り使用してはならない。

6) help ↵

ディスクコントロールコマンドのリストが表示される。

7) initialize controller <DISK NUMBER> ↵

ディスクコントローラ ユニットが初期化され、指定されたディスクが選択される。

8) list bad sectors ↵

バッドセクターの数値（16進）とその場所を表示する。

9) new file system ↵

選択されたディスクドライブのロジカル初期化です。このコマンドは、”enter disk system name” とディスクネームを入力して下さ

いと表示し、入力待となります。２４文字までの名前を入力します。次に、ディスクは消去され、新しいファイルシステムを構築します。このコマンドはよほどの事がない限り使用してはならない。

１０）　ｐｏｗｅｒ　ｄｏｗｎ　ｓｙｓｔｅｍ ↵

システムがパワーダウンされます。

１１）　ｒｅａｄ　ｓｅｃｔｏｒ　＜ＳＥＣＴＯＲ＞，＜ＡＤＤＲ＞ ↵

ＳＥＣＴＯＲ　により指定されたディスクのデータ内容が、ＡＤＤＲより始まるメモリーに書き込まれる。

１２）　ｓｔａｔｕｓ ↵

ディスクコントロール　レジスターの内容を表示する。

１３）　ｔｅｓｔ　ｒｅａｄ ↵

全セクターが検査される。エラーが発見されれば、ディスクステータスが表示され、コマンドは終了する。

１４）　ｖｅｒｉｆｙ　ｄｉｓｋ ↵

ファイルシステムの検査を行い、エラーの大部分の修復を行う。システムクラッシュの後等に行う。

１５）　ｗｒｉｔｅ　ｓｅｃｔｏｒ ＜ＳＥＣＴＯＲ＞，＜ＡＤＤＲ＞ ↵

ＡＤＤＲにより指定されるメモリー上のデータブロック　１０２４バイトが、ＳＥＣＴＯＲで指定されるディスクアドレスに書き込む。このコマンドはよほどの事がない限り使用してはならない。

２・１・１・３　ＡＸＩＳ　モニタ

ＮＣＵＢＥのオペレーティング　システムであるＡＸＩＳは、ＵＮＩＸライクでありマルチタスク、マルチユーザーをサポートする。下記にその主要なコマンドを紹介する。

1)　ａｄｂ　＜ＯＰＴＩＯＮ＞

ホストプロセッサとノードプロセッサの状態を見ることが出来る。

（詳細はａｄｖａｎｃｅｄ　ｄｅｂｕｇｅｒを参照の事）

2)　ｃａｔ

ファイルを編集したり、端末に表示したりする事が出来る。

ｅｘ）

ｃａｔ　ｆｉｌｅ　・・・　ｆｉｌｅの内容をＣＲＴに表示する。

ｃａｔ　ｆｉｌｅ１　ｆｉｌｅ２　＞　ｆｉｌｅ３　・・・
ｆｉｌｅ１の後にｆｉｌｅ２をリンクしｆｉｌｅ３に代入する。

3)　ｃｄ　＜ｄｉｒｅｃｔｏｒｙ＞

ｄｉｒｅｃｔｏｒｙ　で指定されるディレクトリ内に移動する。デフォルトはホームディレクトリにセットされる。

4)　ｃｐ　［－ｒ］　ｆｉｌｅ１　ｆｉｌｅ２

ｆｉｌｅ１　を　ｆｉｌｅ２　にコピーする。

－ｒ　はオーバーライトする。

5)　ｄｆ

現在のディスクの空きブロックと使用ブロックを表示する。
（１０２４ｂｙｔｅ／１ｂｌｏｃｋ）

6)　ｄｕ　［－ｓ］［－ａ］［ｎａｍｅ］

ｎａｍｅで指定したディレクトリから下の使用ブロックを表示する。

－ｓ　ｎａｍｅで指定したディレクトリから下の使用ブロックの合計を表示する。

－ａ　ｎａｍｅで指定したディレクトリから下のファイルを１個づつ表示する。

7)　kill [-a -p -t -w] process no

process no で指定するjobをkillする。

-a　abort
-p　pause
-t　terminate
-w　wake

8)　logout

ログアウトする

9)　mv file1 file2

file1 を file2 に移動する。

10)　ninit

ノードプロセッサ
ノードプロセッサ+VIDEO
ノードプロセッサ+SUN　をイニシャルする。

11)　nm filename

filenameのファイルを開く。（オペレーションは2・2章参照）

12)　pr [-p -e -c] [-d -l] [-f] [file]

fileで指定するファイルをプリントアウトする。

-p　10chra/inch　-
-e　12chra/inch　| 文字の大きさを指定
-c　17chra/inch　-

-d　data　mode　-　文字の種類を指定
-l　letter　mode　-

-f　formatted　リストをフォーマットして出力
（ディレクトリー、時間、ページ）

13)　rm name

name で指定するファイルまたはディレクトリを消去する。

注意）ディレクトリの場合
そのディレクトリの下にファイル存在する場合は出来ない。
そのファイルを消去後、ディクトリを消去できる。

２・２　エディタ

２・２・１　ＮＭＡＣＳ

この章ではＡＸＩＳでの　ＳＣＲＥＥＮ　ＥＤＩＴＯＲ　”ｎｍ”（ＮＭＡＣＳ）のオペレーションについて説明します。

カーソル・ページ移動

＾Ａ	１つ前の行の先頭
＾Ｚ	１つ後の行の先頭
＾Ｗ	１つ左
＾Ｅ	１つ右
＾Ｇ	その行の最後尾の文字
＾Ｖ	その行の先頭の文字
＾Ｐ	次のページ
＾Ｕ	前のページ
＾Ｔ	ファイルの先頭
＾Ｂ	ファイルの最後

コピー

＾Ｙ	コピーする範囲を指定（始点、終点をタイプする）
＾ＸＣ	＾Ｙで指定した範囲をカーソルより後ろにコピーする

削除

＾Ｄ	１文字削除
＾Ｌ	１行削除
＾Ｈ	カーソルの１つ前の１文字削除
＾Ｘ＾Ｄ	次の単語の削除

サーチ

＾Ｎ	サーチ文字の入力
＾Ｆ	＾Ｎで入力された文字のサーチ（先頭から後尾へ）
＾Ｘ＾Ｆ	＾Ｎで入力された文字のサーチ（後尾から先頭へ）

ウインドウ

＾Ｘ＾Ｇ	ニューウインドウ　オープン　（ワークバッファ名とファイル名入力する）
＾Ｘ＾Ｎ	次のウインドウにカーソル移動
＾Ｘ＾Ｗ	前のウインドウにカーソル移動

^ Ｘ ^ Ｐ　　バッファにセーブしたい範囲を指定（始点、終点をタイプし　セーブバッファ名を入力する）
^ Ｘ ^ Ｒ　　^ ＸＰで指定したバッファから読み込む

^ Ｘ ^ Ｋ　　カーソルのあるウインドウをクローズ
^ Ｘ ^ Ａ　　同じファイルをウインドウに出す

その他

^ Ｊ　　コマンドのキャンセル
^]　　ファイルのセーブ＆終了
^ ＼　　ファイルにセーブせず終了

２・２・１・１　ＮＳＨＥＬＬ　ウインドウ

ｎｍ　－ｎ　　エディタの上にもう１つのシェルコマンド領域をつくる

^ Ｏ ^ ＼　で１つのシェルコマンド領域を終了する

4. 光線追跡法による画像生成

プログラム リスト

(1) ＮＣＵＢＥホスト系プログラム

ｈｏｓｔｒａｙｎ２ ： レイトレーシング処理

(2) ＮＣＵＢＥノード系プログラム

ｎｏｄｅｒａｙｎ２ ： 画像生成処理

```
*    ***********************************************************************
*    ***  SYSTEM           : ATR  Parallel Computing System          ***
*    ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*    ***  MODEL NO.        : NCUBE/10    (AXIS)                      ***
*    ***  LANGUAGE         : FORTRAN-77                              ***
*    ***  PROGRAM NAME     : Raytracing on NCUBE    (256)            ***
*    ***  PROGRAM FUNCTION : HOSTRAYN1.F  &  Subroutine color_set    ***
*    ***  ORDER NO.        : ---                                     ***
*    ***  LOT NO.          : ---                                     ***
*    ***  DOCUMENT NO.     : ---                                     ***
*    ***  MAKE OUT DATE    : 08-Jul-87  M.Maekawa (CSK)              ***
*    ***  VERSION NO.      : 02.00                                   ***
*    ***********************************************************************
*    ***    #    #####  ####      Advanced Telecommunication         ***
*    ***   # #     #    #   #      Reseach Institute International   ***
*    ***  #   #    #    ####      ---------------------------------- ***
*    ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*    ***  #   #    #    #   #      Research Laboratories             ***
*    ***********************************************************************

      program Hostrayn2

      integer*4 imsg_type
      integer*4 imsg_buf, imsg_len, isrc, istype
      integer*4 ichan, icnt, idim
      integer*4 istatus, icflag
      character*(*) cfile
      parameter (cfile = 'noderayn2')
      integer*1 iload_work(2**14)
      integer*4  isend_buf(4), inode_buf(0:255)
      integer*1  iread_buf(512*4,2)
      integer*4 iend, i, j, ierr, itime_val(8)
      real  rtime(8), rread_time(8), rave_time(8), rwork_time
      integer*4 timeval, timeval1, timeval2, timeval3, timeval4, timeval5
      character*20 timenam1, timenam2, timenam3, timenam4, timenam5
      character*6 ci

      call color_set
      idim = 8
      icnt=2**idim
      ichan = nopen(idim)
      call amove( 320, 80 )
      call dtextp(' NOW LOADING NODE PROGRAM ... ', 2, 0 )
      call gcmd
      istatus = nloadm(ichan, cfile, -icnt, icflag, iload_work, 2**13)
      call amove( 320, 80 )
      call dtextp('RETURN NODE CALCULATING DATA ... ', 2, 0 )
      call gcmd
      do 200 j = 0, icnt-1
          isrc = -1
          imsg_type = 41
          istatus = nread(ichan, rread_time, 32, isrc, imsg_type)
          imsg_len = 1*512*2
          imsg_type = 44
          istatus = nread(ichan, iread_buf, imsg_len, isrc, imsg_type)
```

```
            call ldrwp(256,128+2*isrc)
            call dwt(iread_buf,512,1,1,512,2,1)
            call gcmd
200       continue
          call amove( 300, 80 )
          call dtextp('NORMAL TERMINATION , ARE YOU OK ? ', 2, 0 )
          call amove( 320, 40 )
          call dtextp('RESULT CPU TIME(sec) :', 2, 0 )
          call amove( 600, 40 )
          write (ci,'(i4.4)') timeval
          call dtextp(ci,2,0)
          call gcmd
          istatus = nclose(ichan)
          stop ' data correct, normal termination'
          end

*     ***********************************************************************
*     ***  PROGRAM NAME     : color_set                                    ***
*     ***********************************************************************

          subroutine  color_set

             カラーテーブルセット処理省略

          return
          end
```

```
*  ********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System           ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs.  ***
*  ***  MODEL NO.        : NCUBE/10    (VERTEX)                     ***
*  ***  LANGUAGE         : FORTRAN-77                               ***
*  ***  PROGRAM NAME     : Raytrace on NCUBE  (256 Nodes use)       ***
*  ***  PROGRAM FUNCTION : NODERAYN2.F                              ***
*  ***  MAKE OUT DATE    : 08-Jul-87  M.Maekawa (CSK)               ***
*  ***  VERSION NO.      : 01.00                                    ***
*  ********************************************************************

      program noderayn2

      common /koku/ ick, hil2
      common /chr/ mcr
      dimension  it(3)
      dimension  ist1(500), ist2(500), ff(500)
      dimension  ax(500), ay(500), az(500), px(500), py(500), pz(500)
      dimension  uu(20), abc(20), rr(20), r2(20)
      dimension  xo(20), yo(20), zo(20), xvo(20), yvo(20), zvo(20)
      dimension  sn1(20), sn2(20), s1(20), s2(20), s3(20), s4(20)
      dimension  xvor(20), yvor(20), zvor(20)
      integer*4 imsg_type, imsg_len, icflag, istatus, isrc
      integer*4 id_node, id_process, id_host, icube_size
      integer*4 imsg_buf(8)
      integer*4 iasize, i, j
      integer*1 isend_buf(512,2), ipara
      integer*4 ired,iblue,igreen
      integer*1 iigreen,iired,iiblue

      call whoami(id_node, id_process, id_host, icube_size)
1000  continue
      xl = 250.
      yl = 400.
      zl = 600.
540   continue
      n = 1
      yk = 100.
      do 51 i = 1, n
          rl = 25.0
          rr(i) = rl
51    continue
      xp = 250.
      yp = 0.
      zp = 200.
      zq = 800.
      zqp = zq - zp
      zqp2 = zqp*zqp
      jeda = 0
      jea = 0
      jenda = 0
      jendm = 0
      ic = 1
      mcr = 2
      do 52 i = 1,n
```

```
100         continue
            xo1 = 260.
            yo1 = 170.
            zo1 = 200.
            xo(i) = xo1
            yo(i) = yo1
            zo(i) = zo1
            k = i -1
            if ( k .eq. 0 ) go to 52
            do 53 j = 1, k
                rrr = sqrt((xo(i)-xo(j))**2+(yo(i)-yo(j))**2+
     1              (zo(i)-zo(j))**2)-(rr(i)+rr(j))
                if ( rrr .le. 0. ) go to 100
53          continue
52      continue
110     continue
        arl = 350.
        tht = 30.
        fai = 70.
        rtht = 0.017453293*tht
        rfai = 0.017453293*fai
        stht = sin ( rtht )
        ctht = cos ( rtht )
        sfai = sin ( rfai )
        cfai = cos ( rfai )
        if ( stht .gt. 1.0 ) stht = 1.0
        if ( ctht .gt. 1.0 ) ctht = 1.0
        if ( sfai .gt. 1.0 ) sfai = 1.0
        if ( cfai .gt. 1.0 ) cfai = 1.0
        if ( stht .lt. -1.0 ) stht = -1.0
        if ( ctht .lt. -1.0 ) ctht = -1.0
        if ( sfai .lt. -1.0 ) sfai = -1.0
        if ( cfai .lt. -1.0 ) cfai = -1.0
        xv = arl*cfai*stht+xp
        yv = arl*sfai+yp
        zv = arl*cfai*ctht+zp
        do 54 i = 1, n
            xvo(i) = xv -xo(i)
            yvo(i) = yv -yo(i)
            zvo(i) = zv -zo(i)
54      continue
        do 55 i = 1, n
            uu(i) = rr(i)**2
            abc(i) = xvo(i)**2+yvo(i)**2+zvo(i)**2
            if (abc(i) .le. uu(i)) go to 110
55      continue
        vp = arl - sqrt(zqp2)
        do 56 i = 1, n
            ss1 = 1.52
            ss2 = ss1
            r = 0.0
            sn1(i) = ss1
            sn2(i) = ss2
            r2(i) = r
56      continue
```

```
      sh = 0.05
      rrl = 50.0
      call matrix( stht, ctht, sfai, cfai, ax1, ax2, ax3,
     1             ay1, ay2, ay3, az1, az2, az3 )
      do 57 i = 1, n
          s1(i) = sn1(i)
          s2(i) = sn2(i)/sn1(i)
          s3(i) = sn1(i)/sn2(i)
          s4(i) = 1.0/sn1(i)
57    continue
      iy_start = id_node*2 + 1
      iy_end = iy_start + 1
      iy_no = 0
      do 10 iy = iy_start, iy_end
          iy_no = iy_no + 1
          ix_no = 0
          y1 = iy - 1
          do 20 ix = 1, 512
              ix_no = ix_no + 1
              x1 = ix - 1
              ick = 0
              x1p = x1 - xp
              y1p = y1 - yp
              z1p = vp - zp
              xx1 = ax1*x1p+ax2*y1p+ax3*z1p+xp
              yy1 = ay1*x1p+ay2*y1p+ay3*z1p+yp
              zz1 = az1*x1p+az2*y1p+az3*z1p+zp
              xv1 = xx1 - xv
              yv1 = yy1 - yv
              zv1 = zz1 - zv
              do 30 jj = 1,500
                  ist1(jj) = 0
                  ist2(jj) = 0
30            continue
              jend = 1
              jr = 1
              xxx1 = 0.
              yyy1 = 0.
              zzz1 = 0.
              call check( xv1, yv1, zv1, xvo, yvo, zvo, xv, yv, zv,
     1                    ist1(jr), ist2(jr), uu, n, xxx1, yyy1, zzz1 )
              if ( ist2(jr) .eq. 1000 ) go to 120
              if ( yv .ge. yk ) go to 200
              if ( yv1 .eq. 0.0 ) go to 210
              tu = ( yk -yv )/yv1
              xxa = xv+xv1*tu
              zza = zv+zv1*tu
              if ( xxa .lt. 180.0 ) go to 200
              if ( xxa .ge. 340.0 ) go to 200
              if ( zza .lt.  80.0 ) go to 200
              if ( zza .ge. 320.0 ) go to 200
              go to 220
210   continue
              r = 0.0
              g = 0.0
```

```
                b = 0.0
                go to 230
200     continue
                ff(jr) = 1.0
                ax(jr) = xxx1 - xv
                ay(jr) = yyy1 - yv
                az(jr) = zzz1 - zv
                call unit( ax(jr), ay(jr), az(jr) )
                px(jr) = xxx1
                py(jr) = yyy1
                pz(jr) = zzz1
                r = 0.0
                g = 0.0
                b = 0.0
520     continue
                if ( jr .gt. 450 ) go to 230
                if ( ist2(jr) .eq. 0 ) go to 230
                if ( ist2(jr) .eq. -9999 ) go to 240
                if ( ist2(jr) .eq.  1000 ) go to 240
                do 58 i = 1,n
                    xvor(i) = px(jr) - xo(i)
                    yvor(i) = py(jr) - yo(i)
                    zvor(i) = pz(jr) - zo(i)
58              continue
                jm = ist1(jr)
                xn = px(jr) - xo(jm)
                yn = py(jr) - yo(jm)
                zn = pz(jr) - zo(jm)
260             call unit( xn, yn,zn )
                if ( ist2(jr) .lt. 3 ) go to 280
                xn = -xn
                yn = -yn
                zn = -zn
280     continue
                sn = s1(jm)
                if ( ist2(jr) .eq. 2 ) sn = s2(jm)
                if ( ist2(jr) .eq. 3 ) sn = s3(jm)
                if ( ist2(jr) .eq. 4 ) sn = s4(jm)
300             call frenel ( xn, yn, zn, ax(jr), ay(jr), az(jr),
     1                        sn, f )
                jh = jend + 1
                jd = jh + 1
                jend = jd
                ff(jh) = ff(jr)*f
                ff(jd) = ff(jr)*(1.0-f)
                if ( ff(jh) .lt. sh ) go to 330
                call hansha( xn, yn, zn, ax(jr), ay(jr), az(jr),
     1                       ax(jh), ay(jh), az(jh) )
                ist2(jh) = 6 - ist2(jr)
                if ( ist2(jh) .eq. 5 ) ist2(jh) = 100
                go to 360
330     continue
                ist2(jh) = -9999
360     continue
                if ( ff(jd) .lt. sh ) go to 380
```

```
            call refct( ax(jr), ay(jr), az(jr), xn, yn, zn,
     1                  sn, ax(jd), ay(jd), az(jd) )
            ist2(jd) = ist2(jr) + 1
            if ( ist2(jd) .eq. 5 ) ist2(jd) = 100
            go to 410
380     continue
            ist2(jd) = -9999
410     continue
            if ( ist2(jh) .eq. -9999 ) go to 430
            if ( ist2(jh) .eq.   100 ) go to 440
            call nextp( ax(jh), ay(jh), az(jh),
     1                  xo(jm), yo(jm), zo(jm),
     2                  px(jr), py(jr), pz(jr),
     3                  rr(jm), r2(jm), ist2(jh),
     4                  px(jh), py(jh), pz(jh) )
            ist1(jh) = ist1(jr)
            go to 430
440     continue
            ist1(jh) = ist1(jr)
            call check( ax(jh), ay(jh), az(jh), xvor, yvor, zvor,
     1                  px(jr), py(jr), pz(jr), ist1(jh), ist2(jh),
     2                  uu, n, px(jh), py(jh), pz(jh) )
430     continue
            if ( ist2(jd) .eq. -9999 ) go to 470
            if ( ist2(jd) .eq.   100 ) go to 480
            call nextp( ax(jd), ay(jd), az(jd),
     1                  xo(jm), yo(jm), zo(jm),
     2                  px(jr), py(jr), pz(jr),
     3                  rr(jm), r2(jm), ist2(jd),
     4                  px(jd), py(jd), pz(jd) )
            ist1(jd) = ist1(jr)
            go to 470
480     continue
            ist1(jd) = ist1(jr)
            call check( ax(jd), ay(jd), az(jd), xvor, yvor, zvor,
     1                  px(jr), py(jr), pz(jr), ist1(jd), ist2(jd),
     2                  uu, n, px(jd), py(jd), pz(jd) )
470     continue
            if ( ist2(jh) .eq. -9999 ) go to 510
            if ( ist2(jh) .ne.  1000 ) go to 510
            call drgb( ax(jh), ay(jh), az(jh),
     1                 px(jr), py(jr), pz(jr),
     2                 xl, yl, zl, rrl, yk, r0, g0, b0 )
            r = r + ff(jh)*r0
            g = g + ff(jh)*g0
            b = b + ff(jh)*b0
510     continue
            if ( ist2(jd) .eq. -9999 ) go to 240
            if ( ist2(jd) .ne.  1000 ) go to 240
            call drgb( ax(jd), ay(jd), az(jd),
     1                 px(jr), py(jr), pz(jr),
     2                 xl, yl, zl, rrl, yk, r0, g0, b0 )
            r = r + ff(jd)*r0
            g = g + ff(jd)*g0
            b = b + ff(jd)*b0
```

```
240     continue
560             jr = jr + 1
                go to 520
120     continue
                if ( yv1 .eq. 0.0 ) go to 530
                tc = ( yk - yv )/yv1
                xxa = xv + xv1*tc
                zza = zv + zv1*tc
                go to 220
530     continue
                r = 0.0
                g = 0.0
                b = 0.0
                go to 230
220     continue
                call back( xxa, zza, r, g, b )
                if ( yv .ge. yk ) go to 230
                r = r/10.0
                g = g/10.0
                b = b/10.0
230     continue
                r = r + 0.5
                g = g + 0.5
                b = b + 0.5
                if ( r .gt. 255.0 )  r = 255.0
                if ( g .gt. 255.0 )  g = 255.0
                if ( b .gt. 255.0 )  b = 255.0
                if ( jend .ne. 1 )  jenda = jenda + 1
                if ( jendm .lt. jend )  jendm = jend
                jeda = jeda + jend
                ired   = int(r)
                igreen = int(g)
                iblue  = int(b)
                call noset ( ired, igreen, iblue, ipara )
                isend_buf( ix_no, iy_no ) = ipara
20          continue
10      continue
        imsg_type = 41
        istatus = nwrite(rtime, 32, id_host, imsg_type, icflag)
        imsg_type = 44
        imsg_len = 1*512*2
        istatus = nwrite ( isend_buf, imsg_len, id_host, imsg_type, icflag )
        stop
        end
*   ********************************************************************
*   ***  PROGRAM NAME     : noset                                    ***
*   ********************************************************************
        subroutine  noset ( ired, igreen, iblue, ipara )

              カラーＮＯセット処理省略

        return
        end
*   ********************************************************************
*   ***  PROGRAM NAME     : drgb                                     ***
```

```
*    ********************************************************************
      subroutine drgb( xa,ya,za,xp,yp,zp,xl,yl,zl,rl,yk,r,g,b )

      common /koku/ ick, hil2

      if ( ya .ge. 0.0 ) go to  10
      tb = ( yk - yp )/ya
      xb = xp + xa*tb
      zb = zp + za*tb
      call back ( xb, zb, r, g, b )
      return
10    continue
      call light ( xl, yl, zl, xa, ya, za, xp, yp, zp, rl )
      if ( ick .eq. 0 ) go to 20
      r = 1000.0*hil2
      g = r
      b = r
      return
20    continue
      r = 0.0
      g = 0.0
      b = 0.0
      return
      end
*    ********************************************************************
*    ***  PROGRAM NAME     : unit                                     ***
*    ********************************************************************
      subroutine unit ( x, y, z )

      r = sqrt( x**2 + y**2 + z**2 )
      x = x/r
      y = y/r
      z = z/r
      return
      end
*    ********************************************************************
*    ***  PROGRAM NAME     : frenel                                   ***
*    ********************************************************************
      subroutine frenel ( xn,yn,zn,xe,ye,ze,sn,f )

      xee = -xe
      yee = -ye
      zee = -ze
      c = xn*xee + yn*yee + zn*zee
      p = sn**2 + c**2 - 1.0
      if ( p .lt. 0.0 )  p = 0.0
      g = sqrt( p )
      a1 = g - c
      a2 = g + c
      a = ( a1**2 )/( a2**2 )
      d1 = c*a2 - 1.0
      d2 = c*a1 + 1.0
      b1 = d1**2
      b2 = d2**2
      b = b1/b2 + 1.0
```

```
      f = a*b/2.0
      return
      end
*     ***********************************************************************
*     ***  PROGRAM NAME     : hansha                                      ***
*     ***********************************************************************
      subroutine  hansha( xn, yn, zn, xe, ye, ze, xh, yh, zh )

      a = abs( xe*xn + ye*yn + ze*zn )
      xa = xe/a
      ya = ye/a
      za = ze/a
      xh = xa + 2.0*xn
      yh = ya + 2.0*yn
      zh = za + 2.0*zn
      call unit( xh, yh, zh )
      return
      end
*     ***********************************************************************
*     ***  PROGRAM NAME     : light                                       ***
*     ***********************************************************************
      subroutine  light( xl,yl,zl,xh,yh,zh,xs,ys,zs,rl )

      common /koku/ ick, hil2

      a = xh**2 + yh**2 + zh**2
      b = xh*(xs - xl) + yh*(ys - yl) + zh*(zs - zl)
      c = (xs-xl)**2 + (ys-yl)**2 + (zs-zl)**2 - rl**2
      d = b**2 - a*c
      if ( d .lt. 0. ) return
      ick = 1
      e = sqrt( d )
      t1 = ( -b + e )/a
      t2 = ( -b - e )/a
      t = amin1( t1, t2 )
      px = xs + t*xh
      py = ys + t*yh
      pz = zs + t*zh
      xn = px - xl
      yn = py - yl
      zn = pz - zl
      call unit( xn, yn, zn )
      xi = -xh
      yi = -yh
      zi = -zh
      call unit( xi, yi, zi )
      hil2 = xn*xi + yn*yi + zn*zi
      return
      end
*     ***********************************************************************
*     ***  PROGRAM NAME     : nextp                                       ***
*     ***********************************************************************
      subroutine nextp( dx,dy,dz,xo,yo,zo,xs,ys,zs,rr,r2,is,xn,yn,zn )

      a = dx**2 + dy**2 + dz**2
```

```
      b = dx*(xs-xo) + dy*(ys-yo) + dz*(zs-zo)
      go to ( 10, 20, 30, 40 ) ,is
10    continue
      return
20    continue
      c = rr**2 - r2**2
      dd = b**2 - a*c
      if ( dd .lt. 0.0 ) go to 100
      d = sqrt( dd )
      t = ( -b - d )/a
      go to 333
100   continue
      is = 4
      t = -2.0*b/a
      go to 333
30    continue
      t = -2.0*b/a
      go to 333
40    continue
      c = r2**2 - rr**2
      dd = b**2 - a*c
      d = sqrt( dd )
      t = ( -b + d )/a
333   continue
      xn = xs + t*dx
      yn = ys + t*dy
      zn = zs + t*dz
      return
      end
*   ***********************************************************************
*   ***  PROGRAM NAME     : matrix                                      ***
*   ***********************************************************************

      subroutine  matrix( stht, ctht, sfai, cfai, ax1, ax2, ax3,
     1                    ay1, ay2, ay3, az1, az2, az3 )

      ax1 = ctht
      ax2 = -stht*sfai
      ax3 = stht*cfai
      ay1 = 0.
      ay2 = cfai
      ay3 = sfai
      az1 = -stht
      az2 = -ctht*sfai
      az3 = ctht*cfai
      return
      end
*   ***********************************************************************
*   ***  PROGRAM NAME     : refct                                       ***
*   ***********************************************************************
      subroutine  refct( xi,yi,zi,xxil,yyil,zzil,sn,rx,ry,rz )

      vi = xi*xi + yi*yi + zi*zi
      vr = xxil*xxil + yyil*yyil + zzil*zzil
      vv = sqrt( vr )
```

```
      xn = xxil/vv
      yn = yyil/vv
      zn = zzil/vv
      svi = sqrt( vi )
      xui = xi/svi
      yui = yi/svi
      zui = zi/svi
      vn = xui*xn + yui*yn + zui*zn
      xni = vn*xn
      yni = vn*yn
      zni = vn*zn
      sx = (xui - xni)/sn
      sy = (yui - yni)/sn
      sz = (zui - zni)/sn
      sa = 1.0 - (sx*sx + sy*sy + sz*sz )
      if ( sa .gt. 1.0 )  sa = 1.0
      if ( sa .lt. 0.0 )  sa = 0.0
      sb = sqrt( sa )
      rx = sx - sb*xn
      ry = sy - sb*yn
      rz = sz - sb*zn
      call unit( rx, ry, rz )
      return
      end
*     ***********************************************************************
*     ***  PROGRAM NAME     : back                                        ***
*     ***********************************************************************
      subroutine  back( xx, zz, rj, gj, bj )

      if (( xx .ge. 340.0 ).or.( xx .lt. 180.0 )) go to 20
      if (( zz .ge. 320.0 ).or.( zz .lt.  80.0 )) go to 20
      kx = xx/40.0
      xm = xx - 40.0*kx
      kz = zz/40.0
      zm = zz - 40.0*kz
      if ( zm .ge. 20.0 ) go to 50
      if ( xm .ge. 20.0 ) go to 30
      go to 40
50    continue
      if ( xm .ge. 20.0 ) go to 40
      go to 30
40    rj = 150.0
      gj = 0.0
      bj = 0.0
      call plate( xx, zz, rj, gj, bj )
      go to 10
30      rj = 150.0
      gj = 150.0
      bj = 150.0
      call plate( xx, zz, rj, gj, bj )
      go to 10
20    rj = 0.0
      gj = 0.0
      bj = 0.0
10    return
```

```
      end
*     ***********************************************************************
*     ***  PROGRAM NAME     : plate                                       ***
*     ***********************************************************************
      subroutine  plate( x, z, r, g, b )

      common /chr/ mcr

      if (( z .lt. 170.0 ).or.( z .gt. 210.0 )) return
      if (( x .lt. 210.0 ).or.( x .gt. 310.0 )) return
      t = 0.75
      r = 128.0*t + r*(1.0 - t)
      g = 128.0*t + g*(1.0 - t)
      b = 0.0*t + b*(1.0 - t)
      if ( mcr .eq. 2 ) go to 10
      call mie( x, z, r, g, b )
      return
10    continue
      call meidai( x, z, r, g, b )
      return
      end
```

5. sobel-filteringによる輪郭抽出

プログラム リスト

(1) ＮＣＵＢＥホスト系プログラム

ｈｏｓｔ＿ｓｇ　　：　ｓｏｂｅｌ＿ｇａｕｓｓ　Ｉ／Ｆ処理

(2) ＮＣＵＢＥノード系プログラム

ｎｏｄｅ＿ｓｏｂｅｌ　：　ｓｏｂｅｌ処理

ｎｏｄｅ＿ｇａｕｓｓ　：　ｇａｕｓｓ処理

```
*************************************************************************
*       Main program for Host of (--) NCUBE Node communication
*               this program is host original picture send to nodes(256)
*               mask and gaus function (2) process or aplicate sobel operator
*               and return to nodes , then draw picture
*
*               picture size ( x*y ) = ( 256*256 )
*               masking size ( x*y ) = (   3*  3 )
*               interval     ( x*y ) = (   1*  1 )
*************************************************************************
      program host_sg
      integer*4  idim, iop_num
      idim  = 8
      call ginit
      call tinit
      call ldtable('grey')
      call gcmd
   10 continue
      read(5,*) iop_num
      if     (iop_num .eq. 0) then
          go to 9999
      else if(iop_num .eq. 1) then
          call gauss(idim)
      else if(iop_num .eq. 2) then
          call sobel(idim)
      end if
      go to 10
 9999 stop
      end
*************************************************************************
*       Sub program for Host of (--) NCUBE Node communication
*               this program is host original picture send to nodes(256)
*               mask and gauss function (2) process
*               and return to nodes , then draw picture
*               picture size ( x*y ) = ( 256*256 )
*               masking size ( x*y ) = (   3*  3 )
*               interval     ( x*y ) = (   1*  1 )
*************************************************************************
      subroutine gauss(idim)
      integer*4 idim
*       message types and return codes
        integer*4 imsg_type, imsg_buf, isrc, istype
*       ichan is the channel number of the allocated hypercube
        integer*4 ichan, icnt , idim ,iop_num
*       istatus is the status returned from message primitives
        integer*4 istatus, icflag
*       cfile is the name of the program to load and run on the hypercube
        character*(*) cfile
        parameter (cfile = 'node_gauss')
        integer*1 iload_work(2**14)
*       Send and return data
        integer*4 isend_size, iread_size,ithres
        integer*1 iread_buf(256)
        real param(4)
        real thres
```

```
*       Picture data
        integer*1 dat1
        common /dat1/ dat1(2**16)
        character*(54) picfile
        character*(*) pic1
        parameter (pic1 = '/usr/miyake/edge/image_data/sekiyama1.gre')
        character*(*) pic2
        parameter (pic2 = '/usr/miyake/edge/image_data/miura1.gre')
          一部省略
*       Work parameter
        integer*4  jxpos, jypos
        integer*4  kxpos, kypos, kxpos1, kypos1
        integer*4  i, j, k, largex, largey , n_image
        real  thres
        character*6 ci
        parameter ( jxpos = 96)
        parameter ( largex = 256, largey = 256)
        parameter (jypos = 120 )
*       time count parameters
        integer*4    timeval(10)
        character*20 timenam(10)
*****************************************************************************
*        Start program
        idim  = 8
        ierr = iclock( 0, timenam(1), timeval(1) )
******* Allocate and open a hypercube
        icnt  = 2**idim
        ichan = nopen(idim)
******* Load programs to nodes
        istatus = nloadm(ichan, cfile, -icnt, icflag, iload_work, 2**16)
        ierr = iclock( 0, timenam(2), timeval(2) )
******* Picture choice , read file and parameter
10      continue
        read (5,*) ichenge
        if ( ichenge .lt. 0 ) go to 10
        if ( ichenge .eq. 0 ) go to 9990
        if ( ichenge .ge.10 ) go to 10
        call ldcl1(127)
        call amove (jxpos,jypos)
        call rfrct(largex,largey)
        call gcmd
        call ldcl1(255)
        call amove(jxpos + 30, jypos - 30)
        call dtextp('ORIGINAL PICTURE', 2,0)
        call gcmd
        if ( ichenge .eq. 1 ) then
            picfile = pic1
        elseif ( ichenge .eq. 2 ) then
            picfile = pic2
            一部省略
        endif
        read (5,*) sigx
        read (5,*) sigy
        read (5,*) thres
        ierr = iclock( 0, timenam(3), timeval(3) )
```

```
      ifile = ifopen( 3, picfile, length )
      ir = ifread( ifile, length, dat1, 0, 0 )
      ir = ifclose( ifile )
      ierr = iclock( 0, timenam(4), timeval(4) )
      call ldrwp ( jxpos, jypos )
      call dwt(dat1,largex,1,largey,largex,-largey,1)
      call gcmd
******* Send picture's parameters to each node
      param(1) = thres
      param(2) = sigx
      param(3) = sigy
      param(4) = 0.0
      ierr = iclock( 0, timenam(5), timeval(5) )
      isend_size = icnt
      do 120 i = icnt-1, 0, -1
          if ( i .eq. 0 )  isend_size = icnt
          ii = icnt * i + 1
          imsg_type = 48
          istatus = nwrite(ichan, dat1(ii), isend_size, i, imsg_type)
          imsg_type = 47
          istatus = nwrite(ichan, param,      16,       i, imsg_type)
          isend_size = 3 * icnt
120       continue
      ierr = iclock( 0, timenam(6), timeval(6) )
      do 140 n = 1,2
          nn = n * 280
          call ldcl1(127)
          call amove( jxpos + nn , jypos )
          call rfrct( largex, largey )
          call gcmd
          if ( n .eq. 1) then
              call ldcl1 (255)
              call amove( jxpos + nn + 40 , jypos - 30 )
              call dtextp('SMOOTHED DATA', 2, 0 )
          else if ( n .eq. 2) then
              call ldrwp ( jxpos + nn  , jypos )
              call dwt(dat1,largex,1,largey,largex,-largey,1)
              call gcmd
              call ldcl1 (255)
              call amove( jxpos + nn + 5 , jypos - 27)
              call dtextp('GAUSS-FILTERED RESULT', 2, 0 )
              call amove( jxpos + nn + 5, jypos - 54 )
              call dtextp(' THRESHOLD =',2 ,0 )
              call amove (jxpos + nn + 150, jypos - 54 )
              call dtextp(ci,2,0)
              call amove( jxpos + nn + 5 , jypos - 81 )
              call dtextp(' SIGMA-X   =', 2, 0 )
              call amove( jxpos + nn + 150 , jypos-81 )
              call dtextp(ci,2,0)
              call amove( jxpos + nn + 5 , jypos - 104 )
              call dtextp(' SIGMA-Y   =', 2, 0 )
              call amove( jxpos + nn + 150 , jypos - 104 )
              call dtextp(ci,2,0)
          end if
          call gcmd
```

```
            ierr = iclock( 0, timenam(2*n+5), timeval(2*n+5) )
            id = n + 50
            iread_size = icnt
            do 130 i = icnt-1, 0, -1
                isrc = -1
                imsg_type = id
                iread_size = icnt
                istatus = nread(ichan,iread_buf,iread_size,isrc,imsg_type)
                call ldrwp( jxpos + nn , jypos + 255 - isrc )
                call dwt(iread_buf,icnt,1,1,icnt,1,1)
                call gcmd
  130        continue
        ierr = iclock( 0, timenam(2*n+6), timeval(2*n+6) )
  140   continue
        go to 10
******* finally, close and deallocate the hypercube
9990    istatus = nclose(ichan)
        return
        end

*****************************************************************
*       Main program for Host of <--> NCUBE Node communication
*               this program is host original picture send to nodes(256)
*               mask and gaus function (2) process or aplicate sobel operator
*               and return to nodes , then draw picture
*               picture size ( x*y ) = ( 256*256 )
*               masking size ( x*y ) = (   3*  3 )
*               interval     ( x*y ) = (   1*  1 )
*               demo2 - souroutinne -> sobel
*****************************************************************************
        subroutine sobel(idim)
        integer idim
*       message types and return codes
        integer*4 imsg_type, imsg_buf, isrc, istype
*       ichan is the channel number of the allocated hypercube
        integer*4 ichan, icnt , idim ,iop_num
*       istatus is the status returned from message primitives
        integer*4 istatus, icflag
*       cfile is the name of the program to load and run on the hypercube
        character*(*) cfile
        parameter (cfile = 'node_sobel')
        integer*1 iload_work(2**14)
*       Send and return data
        integer*4 isend_size, iread_size,ithres
        integer*1 iread_buf(256)
        real thres
*       Picture data
        integer*1 dat1
        common /dat1/ dat1(2**16)
        character*(54) picfile
        character*(*) pic1
        parameter (pic1 = '/usr/miyake/edge/image_data/sekiyama1.gre')
        character*(*) pic2
        parameter (pic2 = '/usr/miyake/edge/image_data/miura1.gre')
            一部省略
```

```
*       Work parameter
        integer*4  jxpos, jypos
        integer*4  kxpos, kypos, kxpos1, kypos1
        integer*4  i, j, k, largex, largey , n_image
        real  thres
        character*6 ci
        parameter ( jxpos = 96)
        parameter ( largex = 256, largey = 256)
        parameter (jypos = 450 )
*       time count parameters
        integer*4    timeval(10)
        character*20 timenam(10)
*************************************************************************
*        Start program
        idim  = 8
        ierr = iclock( 0, timenam(1), timeval(1) )
******* Allocate and open a hypercube
        icnt  = 2**idim
        ichan = nopen(idim)
******* Load programs to nodes
        istatus = nloadm(ichan, cfile, -icnt, icflag, iload_work, 2**16)
        ierr = iclock( 0, timenam(2), timeval(2) )
******* Picture choice , read file and parameter
10      continue
        read (5,*) ichenge
        if ( ichenge .lt. 0 ) go to 10
        if ( ichenge .eq. 0 ) go to 9990
        if ( ichenge .ge.10 ) go to 10
        call ldcl1(127)
        call amove (jxpos,jypos)
        call rfrct(largex,largey)
        call gcmd
        call ldcl1(255)
        call amove(jxpos + 30, jypos - 30)
        call dtextp('ORIGINAL PICTURE', 2,0)
        call gcmd
        if ( ichenge .eq. 1 ) then
            picfile = pic1
        elseif ( ichenge .eq. 2 ) then
            picfile = pic2
              一部省略
        endif
        read (5,*) thres
        ierr = iclock( 0, timenam(3), timeval(3) )
        ifile = ifopen( 3, picfile, length )
        ir = ifread( ifile, length, dat1, 0, 0 )
        ir = ifclose( ifile )
        ierr = iclock( 0, timenam(4), timeval(4) )
        call ldrwp ( jxpos, jypos )
        call dwt(dat1,largex,1,largey,largex,-largey,1)
        call gcmd
******* Send picture's parameters to each node
        ierr = iclock( 0, timenam(5), timeval(5) )
        isend_size = icnt
        do 120 i = icnt-1, 0, -1
```

```
            if ( i .eq. 0 )  isend_size = icnt
            ii = icnt * i + 1
            imsg_type = 48
            istatus = nwrite(ichan, dat1(ii), isend_size, i, imsg_type)
            imsg_type = 47
            istatus = nwrite(ichan,     thres,          4, i, imsg_type)
            isend_size = 3 * icnt
120     continue
        ierr = iclock( 0, timenam(6), timeval(6) )
******* Read converted image from nodes
        do 140 n = 1,2
            nn = n * 280
            call ldcl1(127)
            call amove( jxpos + nn , jypos )
            call rfrct( largex, largey )
            call gcmd
            if ( n .eq. 1) then
                call ldcl1 (255)
                call amove( jxpos + nn + 40 , jypos - 30 )
                call dtextp('SMOOTHED DATA', 2, 0 )
            else if ( n .eq. 2) then
                call ldrwp ( jxpos + nn  , jypos )
                call dwt(dat1,largex,1,largey,largex,-largey,1)
                call gcmd
                call ldcl1 (255)
                call amove( jxpos + nn + 5 , jypos - 27)
                call dtextp('SOBEL-FILTERED RESULT', 2, 0 )
                call amove( jxpos + nn + 5, jypos - 54 )
                call dtextp(' THRESHOLD =',2 ,0 )
                call amove (jxpos + nn + 150, jypos - 54 )
                call dtextp(ci,2,0)
            end if
            call gcmd
            ierr = iclock( 0, timenam(2*n+5), timeval(2*n+5) )
            id = n + 50
            iread_size = icnt
            do 130 i = icnt-1, 0, -1
                    isrc = -1
                imsg_type = id
                iread_size = icnt
                istatus = nread(ichan,iread_buf,iread_size,isrc,imsg_type)
                call ldrwp( jxpos + nn , jypos + 255 - isrc )
                call dwt(iread_buf,icnt,1,1,icnt,1,1)
                call gcmd
  130        continue
        ierr = iclock( 0, timenam(2*n+6), timeval(2*n+6) )
  140   continue
        go to 10
******* finally, close and deallocate the hypercube
9990    istatus = nclose(ichan)
9999    write(6,*)  ' data correct, normal termination ( sobel ) '
        return
        end
```

```
*********************************************************************
*     program node_gauss
*       dez/87
*       somoothing (chose media of 9 values),
*       node communicatin and
*       gauss filtering
*********************************************************************
      program node_gauss
      common sigx,x0,sigy,sigx22
      integer*4 id_node, id_host, id_process, icube_size
      integer*4 icrc,iwgt1,iwgt2,icx1,icx2,icy1,icy2,k,ix
      integer*4 iread_size,isend_size
      integer*1 iread_buf(3*256), isend_buf(256)
      integer*4 ires(256),ires1(256),ires2(256),ithres,iread(3*256)
      dimension iwgt1(3,6),iwgt2(3,6)
      real param(4) , rthes(3*256),thres
      integer*1 mmask
      real xm(16*16)
*************************************************************
*        starting operatin
      call whoami(id_node,id_process,id_host,icube_size)
   10 continue
      if ((id_node .eq. 0) .or. (id_node .eq.255)) then
        iread_size = 256
      else
        iread_size = 3 * 256
      endif
***** getting data
      isrc = -1
      imsg_type = 48
      istatus = nread(iread_buf,iread_size,isrc,imsg_type,icflag)
      isrc = -1
      imsg_type = 47
      istatus = nread( param, 16, isrc, imsg_type, icflag)
      if ((id_node .eq. 0) .or. (id_node .eq. 255)) then
        isend_size = 256
        imsg_type = 51
        istatus = nwrite(iread_buf,isend_size,id_host,imsg_type,icflag)
        do 20 ix = 1, 256
           isend_buf(ix) = -1
   20   continue
        go to 300
      end if
***   ajustment
      do 30 ix = 1,768
        iread(ix) = iread_buf(ix)
        if (iread(ix) .lt. 0)  iread(ix) = iread_buf(ix) + 256
   30 continue
        do 40 k = 1, 3*256
            rthes(k) = param(1)
 40     continue
        mxhalf = 1
        myhalf = 1
        sigx   = param(2)
        sigy   = param(3)
```

```
        x0     = param(4)
        ixintv = 1
        iyintv = 1
        sigx22 = 2.0*sigx*sigx
        ixstat = ixintv+mxhalf
        ixend  = 256-mxhalf
C*************************************************************************
C       Caluculation module
C*************************************************************************
      do 41 ix = 2,255
         kk = med(iread(ix-1) , iread(ix) ,   iread(ix+1),
     1            iread(ix+255),iread(ix+256),iread(ix+257),
     2            iread(ix+511),iread(ix+512),iread(ix+513) )
         ires(ix) = kk
         if ( kk .ge. 127) then
             isend_buf(ix) = kk -256
         else
             isend_buf(ix) = kk
         end if
   41 continue
      isend_buf(1)   = iread(257)
      isend_buf(256)= iread(512)
***** returning smoothed data
      isend_size = 256
      imsg_type = 51
      istatus = nwrite(isend_buf,isend_size,id_host,imsg_type,icflag)
***** node - node communications
      if (mod(id_node,2) .eq. 0) then
           isrc = id_node - 1
           imsg_type = 100 + id_node - 1
           status = nwrite(ires,256*4,isrc,imsg_type,icflag)
           if(id_node .ne. 254) then
                isrc = id_node + 1
                imsg_type = 100 + id_node
                istatus = nread(ires1,256*4,isrc,imsg_type,icflag)
                isrc = id_node + 1
                imsg_type = 500 + id_node + 1
                status = nwrite(ires,256*4,isrc,imsg_type,icflag)
           end if
           isrc = id_node - 1
           imsg_type = 500 + id_node
           istatus = nread(ires2,256*4,isrc,imsg_type,icflag)
      else
          isrc = id_node + 1
          imsg_type = 100 + id_node
          istatus = nread(ires1,256*4,isrc,imsg_type,icflag)
          if ( id_node .ne.1) then
                isrc = id_node - 1
                imsg_type = 100 + id_node - 1
                status = nwrite(ires,256*4,isrc,imsg_type,icflag)
                isrc = id_node - 1
                imsg_type = 500 + id_node
                istatus = nread(ires2,256*4,isrc,imsg_type,icflag)
          end if
          isrc = id_node + 1
```

```
          imsg_type = 500 + id_node + 1
          status = nwrite(ires,256*4,isrc,imsg_type,icflag)
      end if
      if (id_node .eq. 254 ) then
        do 50 ix = 2, 255
          iread(ix+256) = ires( ix)
          iread(ix+512) = ires2(ix)
          ires1(ix) = 0
          ires2(ix) = 0
   50   continue
      else if (id_node .eq. 1 ) then
        do 60 ix = 2, 255
          iread(ix    ) = ires1(ix)
          iread(ix+256) = ires( ix)
          ires1(ix) = 0
          ires2(ix) = 0
   60   continue
      else
        do 70 ix = 2, 255
          iread(ix    ) = ires1(ix)
          iread(ix+256) = ires( ix)
          iread(ix+512) = ires2(ix)
          ires1(ix) = 0
          ires2(ix) = 0
   70   continue
      end if
      do 80 ix = 1, 256
        isend_buf(ix) = -1
   80 continue
c***** Do convolution on image
        do 150 i = 0, 35
            call mask( i, mxhalf, myhalf, xm )
            mmask = i
            do 151 nnx = ixstat, ixend
                temp = 0.0
                inxm = 1
                inxg = nnx-ixintv
                do 152 mmy = myhalf, -myhalf, -1
                    do 153 mmx = -mxhalf, mxhalf
                        temp = temp+float(iread(inxg))*xm(inxm)
                        inxm = inxm+1
                        inxg = inxg+1
  153               continue
                    inxg=inxg+253
  152           continue
                if ( temp .ge. rthes(nnx) ) then
                    isend_buf(nnx) = mmask*3
                    rthes(nnx) = temp
                endif
  151       continue
  150   continue
      isend_buf(  1) = -1
      isend_buf(256) = -1
  300 continue
***** returning result
```

```
      isend_size = 256
      imsg_type = 52
      istatus = nwrite(isend_buf,isend_size,id_host,imsg_type,icflag)
      go to 10
 9999 stop
      end
************************************************************************
*       function returning median value
      function med(k1,k2,k3,k4,k5,k6,k7,k8,k9)
      integer k1,k2,k3,k4,k5,k6,k7,k8,k9
      integer m(9)
      logical id
      m(1) = k1
      m(2) = k2
      m(3) = k3
      m(4) = k4
      m(5) = k5
      m(6) = k6
      m(7) = k7
      m(8) = k8
      m(9) = k9
 10   continue
      id = .true.
      do 20 i = 1, 5
          if( m(i) .lt. m(i+4) ) then
              call change ( m(i), m(i+4) )
              id = .false.
          end if
 20   continue
      do 30 i = 1, 4, 2
          if( m(i) .lt. m(i+1) ) then
              call change ( m(i), m(i+1) )
              id = .false.
          end if
 30   continue
      if ( .not. id) go to 10
      med = m(5)
      return
      end
**************
*       change values
      subroutine change( m1, m2)
      integer m1,m2
      integer m
      m = m1
      m1 = m2
      m2 = m
      return
      end
*******
      subroutine mask( mmask, mxhalf, myhalf, xm )
      real xm(16*16)
      pi    = 3.14159265358979323846264338327795
      siita = float(mmask)*pi/18.0
      inx=1
```

```
      do 20 mm1 = myhalf, -myhalf, -1
          do 30 mm2 = -mxhalf, mxhalf
              x = float(mm2)
              y = float(mm1)
              xx =  x*cos(siita)+y*sin(siita)
              yy = -x*sin(siita)+y*cos(siita)
              xm(inx) = fmx(xx/4.)*fmy(yy/4.)
              inx = inx+1
30        continue
20    continue
      return
      end
********
      function fmx(xx)
      common sigx, x0, sigy, sigx22
      fmx = exp(-(xx-x0)**2/sigx22)*xx/sigx/sigx
      return
      end
********
      function fmy(yy)
      common sigx, x0, sigy, sigx22
      fmy = exp(-yy*yy/2.0/sigy/sigy)
      return
      end
```

```
*****************************************************************
*     program node_sobel
*       dez/87
*       (ver 3.0)
*       somoothing (chose media of 9 values),
*       node communicatin and
*       sobel filtering
*****************************************************************
      program node_sobel
      integer*4 id_node, id_host, id_process, icube_size
      integer*4 icrc,iwgt1,iwgt2,icx1,icx2,icy1,icy2,k,ix
      integer*4 iread_size,isend_size
      integer*1 iread_buf(3*256), isend_buf(256)
      integer*4 ires(256),ires1(256),ires2(256),ithres,iread(3*256)
      dimension iwgt1(3,6),iwgt2(3,6)
      real thres
      data iwgt1 /  -1,  1, -1,
     &              -1,  0, -2,
     &              -1, -1, -1,
     &               1,  1,  1,
     &               1,  0,  2,
     &               1, -1,  1   /
      data iwgt2 /  -1,  1,  1,
     &               0,  1,  2,
     &               1,  1,  1,
     &              -1, -1, -1,
     &               0, -1, -2,
     &               1, -1, -1   /
*****************************************************************
*        starting operatin
      call whoami(id_node,id_process,id_host,icube_size)
   10 continue
      if ((id_node .eq. 0) .or. (id_node .eq.255)) then
        iread_size = 256
      else
        iread_size = 3 * 256
      endif
***** getting data
      isrc = -1
      imsg_type = 48
      istatus = nread(iread_buf,iread_size,isrc,imsg_type,icflag)
      isend_size = 256
      imsg_type = 47
      istatus = nread(   thres,  4,   isrc,imsg_type,icflag)
      if ((id_node .eq. 0) .or. (id_node .eq. 255)) then
        isend_size = 256
        imsg_type = 51
        istatus = nwrite(iread_buf,isend_size,id_host,imsg_type,icflag)
        do 20 ix = 1, 256
           isend_buf(ix) = -1
   20   continue
        go to 300
      end if
***   ajustment
      do 30 ix = 1,768
```

```
      iread(ix) = iread_buf(ix)
      if (iread(ix) .lt. 0)  iread(ix) = iread_buf(ix) + 256
   30 continue
      do 40 ix = 2,255
         kk = med(iread(ix-1) , iread(ix) ,   iread(ix+1),
     1            iread(ix+255),iread(ix+256),iread(ix+257),
     2            iread(ix+511),iread(ix+512),iread(ix+513) )
         ires(ix) = kk
         if ( kk .ge. 127) then
             isend_buf(ix) = kk -256
         else
             isend_buf(ix) = kk
         end if
   40 continue
      isend_buf(1) = iread_buf(1)
      isend_buf(256) = iread_buf(256)
***** returning smoothed data
      isend_size = 256
      imsg_type = 51
      istatus = nwrite(isend_buf,isend_size,id_host,imsg_type,icflag)
***** node - node communications
      if (mod(id_node,2) .eq. 0) then
           isrc = id_node - 1
           imsg_type = 100 + id_node - 1
           status = nwrite(ires,256*4,isrc,imsg_type,icflag)
           if(id_node .ne. 254) then
                isrc = id_node + 1
                imsg_type = 100 + id_node
                istatus = nread(ires1,256*4,isrc,imsg_type,icflag)
                isrc = id_node + 1
                imsg_type = 500 + id_node + 1
                status = nwrite(ires,256*4,isrc,imsg_type,icflag)
           end if
           isrc = id_node - 1
           imsg_type = 500 + id_node
           istatus = nread(ires2,256*4,isrc,imsg_type,icflag)
      else
          isrc = id_node + 1
          imsg_type = 100 + id_node
          istatus = nread(ires1,256*4,isrc,imsg_type,icflag)
          if ( id_node .ne.1) then
                isrc = id_node - 1
                imsg_type = 100 + id_node - 1
                status = nwrite(ires,256*4,isrc,imsg_type,icflag)
                isrc = id_node - 1
                imsg_type = 500 + id_node
                istatus = nread(ires2,256*4,isrc,imsg_type,icflag)
          end if
          isrc = id_node + 1
          imsg_type = 500 + id_node + 1
          status = nwrite(ires,256*4,isrc,imsg_type,icflag)
      end if
      if (id_node .eq. 254 ) then
        do 50 ix = 2, 255
          iread(ix+256) = ires( ix)
```

```
          iread(ix+512) = ires2(ix)
          ires1(ix) = 0
          ires2(ix) = 0
   50   continue
      else if (id_node .eq. 1 ) then
        do 60 ix = 2, 255
          iread(ix    ) = ires1(ix)
          iread(ix+256) = ires( ix)
          ires1(ix) = 0
          ires2(ix) = 0
   60   continue
      else
        do 70 ix = 2, 255
          iread(ix    ) = ires1(ix)
          iread(ix+256) = ires( ix)
          iread(ix+512) = ires2(ix)
          ires1(ix) = 0
          ires2(ix) = 0
   70   continue
      end if
***** Sobel filtering ******************
      do 100 k = 1,6
        iwt1 = iwgt1(3,k)
        iwt2 = iwgt2(3,k)
        icy1 = (iwgt1(2,k) + 1 ) * 256
        icy2 = (iwgt2(2,k) + 1 ) * 256
        do 110 ix = 2, 255
           j1 = icy1 + ix + iwgt1(1,k)
           j2 = icy2 + ix + iwgt2(1,k)
           ires1(ix) = ires1(ix) + iwt1*iread(j1)
           ires2(ix) = ires2(ix) + iwt2*iread(j2)
  110   continue
  100 continue
      do 200 ix = 2, 255
        res = real(iabs(ires1(ix)) + iabs(ires2(ix)))
        if (      res .ge. thres * 5.0 ) then
            isend_buf(ix) = 2
        else if ( res .ge. thres * 4.0) then
            isend_buf(ix) = 15
        else if ( res .ge. thres * 3.0 ) then
            isend_buf(ix) = 30
        else if ( res .ge. thres * 2.0 ) then
            isend_buf(ix) = 45
        else if ( res .ge. thres       ) then
            isend_buf(ix) = 60
        else
            isend_buf(ix) = -1
        end if
  200 continue
  300 continue
      isend_buf(  1) = -1
      isend_buf(256) = -1
***** returning result
      isend_size = 256
      imsg_type = 52
```

```
      istatus = nwrite(isend_buf,isend_size,id_host,imsg_type,icflag)
      go to 10
 9999 stop
      end
*****************************************************************
*       function returning median value
      function med(k1,k2,k3,k4,k5,k6,k7,k8,k9)
      integer k1,k2,k3,k4,k5,k6,k7,k8,k9
      integer m(9)
      logical id
      m(1) = k1
      m(2) = k2
      m(3) = k3
      m(4) = k4
      m(5) = k5
      m(6) = k6
      m(7) = k7
      m(8) = k8
      m(9) = k9
 10   continue
      id = .true.
      do 20 i = 1, 5
          if( m(i) .lt. m(i+4) ) then
              call change ( m(i), m(i+4) )
              id = .false.
          end if
 20   continue
      do 30 i = 1, 4, 2
          if( m(i) .lt. m(i+1) ) then
              call change ( m(i), m(i+1) )
              id = .false.
          end if
 30   continue
      if ( .not. id) go to 10
      med = m(5)
      return
      end
**************
*       change values
      subroutine change( m1, m2)
      integer m1,m2
      integer m
      m = m1
      m1 = m2
      m2 = m
      return
      end
```

6. ヒトの網膜における情報処理とそのモデル

プログラム　リスト

(1) ＮＣＵＢＥホスト系プログラム

ｈｏｓｔｄｓｐ４　　：　網膜モデル処理

(2) ＮＣＵＢＥノード系プログラム

ｎｏｄｅｄｓｐ５　　：　不均一網膜モデル演算処理
ｎｏｄｅｄｓｐ６　　：　不均一網膜モデル演算処理

```
*  ********************************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System                       ***
*  ***  PROJECT          : Auditory & Visual perception Res. Labs.              ***
*  ***  MODEL NO.        : NCUBE/10                                             ***
*  ***  LANGUAGE         : FORTRAN-77                                           ***
*  ***  PROGRAM NAME     : hostdsp4 (used node 256 version)                     ***
*  ***  PROGRAM FUNCTION : ---                                                  ***
*  ***  ORDER NO.        : ---                                                  ***
*  ***  LOT NO.          : ---                                                  ***
*  ***  DOCUMENT NO.     : ---                                                  ***
*  ***  MAKE OUT DATE    :09-JUN-88                                             ***
*  ***  VERSION NO.      :3.0                                                   ***
*  ********************************************************************************
*  ***    *    *****  ****     Advanced Telecommunication                      ***
*  ***   * *     *    *   *     Research Institute International              ***
*  ***  *   *    *    ****     ----------------------------------------       ***
*  ***  *****    *    *   *    ATR Auditory and Visual Perception             ***
*  ***  *   *    *    *    *    Research Laboratories                         ***
*  ********************************************************************************

      program     hostdsp4

      integer*4 imsg_type, imsg_buf, isrc, istype
      integer*4 ichan, icnt, idim
      integer*4 istatus
      logical*1 lcflag1(0:255), lcflag2(0:255)
      integer*2 iik
      character*(*) cfile1, cfile2
      parameter (cfile1 = 'nodedsp5')
      parameter (cfile2 = 'nodedsp6')
      integer*1 iload_work(2**14)
      real  rsend_buf(1:8,0:255)
      real  rread_buf(1:3,1:17), rg(1:3,1:16,0:31), rmax, rmin
      real  rrtime(0:127), rrgo(0:127), ggmax(0:127)
      integer*4 iend, i, j, ierr, itime(8)
      integer*2 ix_buf(32),iy_buf(32),ix_buf1(32),iy_buf1(32),ixray,iyray
      integer*4 timeval, timeval1, timeval2, timeval3, timeval4,timeval5
      real  rtime(8), rread_time(8), rave_time(8), rwork_buf
      character*20 timenam1, timenam2, timenam3, timenam4, timenam5
      character*2 ci
      real  e, p, q, th, dist, rdist, rgo, gmax, vrad
      real  gol, dil, diisl
      real*4  rdis

      call ginit
      call tinit
      call ldtable('h/s')
      call ldcolor(0,1,1,1)
      idim = 8
      ichan = nopen(idim)
10    continue
      call ldcl1 (2)
      call gcmd
      do 20 i = 1, 8
         rtime(i) = 0.
```

```
          rave_time(i) = 0.
          itime(i) = 0
20      continue
        do 30 i = 5, 8
          rtime(i) = 9999.
30      continue
        do 40 iik = 0,255
           lcflag1(iik) = 0
           lcflag2(iik) = 0
40      continue
        rwork_time = 0.
        call distance ( rdis, iend )
        if ( iend .eq. 1 ) go to 9999
        e = rdis
        p = 21.0
        q = 0.116
        th = 0.0
        in   = 0
        inr  = 0
        icon = 0
        dist  = 0.0
        rdist = 0.0
        rgo  = 0.0
        gmax = 0.0
        icount = 0
200     continue
        rdist = rdist + dist
        vrad =( 0.55*rdist/60.0 + 4.15 )/2.0
        if ( rdist .lt. e - p/2.0 - vrad*4.0 ) go to 230
        if ( rdist .gt. e + p/2.0 + vrad*4.0 ) go to 220
        if ( in .eq. 0 ) then
           inr = 1
        else
           inr = in*6
        endif
        rsend_buf( 1, icount ) = 1.0
        rsend_buf( 2, icount ) = e
        rsend_buf( 3, icount ) = p
        rsend_buf( 4, icount ) = q
        rsend_buf( 5, icount ) = th
        rsend_buf( 6, icount ) = float(inr)
        rsend_buf( 7, icount ) = rdist
        rsend_buf( 8, icount ) = vrad
        icount = icount + 1
230     dist = 3.3*( 1.0+0.009833*rdist )/sqrt( 10.0 )
        in = in + 1
        go to 200
220     continue
        icnt = icount
        do 50 j = 0, icnt-1
           rsend_buf(1,j) = float(icnt)
50      continue
        do 60 ik = 0,icnt-1
           lcflag1(ik) = 1
60      continue
```

```
      do 70 ik = icnt,255
         lcflag2(ik) = 1
70    continue
      istatus = nloadm(ichan, cfile1, 256, lcflag1, iload_work, 2**13)
      istatus = nloadm(ichan, cfile2, 256, lcflag2, iload_work, 2**13)
      do 120 i = 0, icnt-1
         imsg_type = 48
         istatus = nwrite(ichan, rsend_buf(1,i), 32, i, imsg_type)
120   continue
      rmax = 0.0
      rmin = 99999.99
      do 130 i = 0, icnt-1
         isrc = -1
         imsg_type = 42
         istatus = nread(ichan, rread_buf, 3*17*4, isrc, imsg_type)
         imsg_type = 41
         istatus = nread(ichan, rread_time, 16, isrc, imsg_type)
         rwork_time = rwork_time + rread_time(4)
         rrtime(isrc) = rread_time(4)
         rtime(4) = max( rtime(4), rread_time(4) )
         rtime(8) = min( rtime(8), rread_time(4) )
         gmax = max( gmax, rread_buf(2,1) )
         rgo  = rgo + rread_buf(3,1)
         ggmax(isrc) = rread_buf(2,1)
         rrgo(isrc)  = rread_buf(3,1)
         do 140 jk = 1, 16
            do 145 jj = 1, 3
               rg( jj,jk,isrc ) = rread_buf(jj,jk+1)
145         continue
140      continue
130   continue
      call crtout ( rg, icnt )
      go to 10
9999  istatus = nclose(ichan)
      stop ' data correct, normal termination'
      end

      subroutine  crtout ( rg, icnt )

      implicit    integer*4 (i-n)
      real        rg( 1:3, 1:16, 0:31 )
      character*2 ci
      dimension   iadr(1:2,0:40,0:40)

      call thradr ( iadr )
      do 303 i = icnt-1, 0, -1
         ip = int( rg(1,16,i) )
         do 304 ki = 2, ip
            if ( rg(3,ki,i) .ge. 100.0 ) then
               rg(3,ki,i) = 0.0
            endif
304      continue
303   continue
      rmax = 0.0
      rmin = 999999.99
```

```
      do 301 i = icnt-1, 0, -1
          ip = int( rg(1,16,i) )
          do 302 ki = 2, ip
              rmax = max( rmax, rg(3,ki,i) )
              rmin = min( rmin, rg(3,ki,i) )
302       continue
301   continue
      idy1 = nint( 10.0 * rmax    / 16.0 ) + 1
      idy2 = nint( 10.0 * (-rmin) / 16.0 ) + 1
      imdl = (icnt-1) / 2
      ibx = 645
      iby = 559
      idx = 5
      idy = 5
      iht1= 15 + (icnt/2)
      iht = ibx + idx*(icnt/2)
      do 300 i = icnt-1, 0, -1
          call ldcl1 (15+16*i)
          call gcmd
          ix = int( rg(1,1,i) )
          ixpos = 170 + ix
          iy = int( rg(2,1,i) )
          ip = int( rg(1,16,i) )
          call amove( ixdel*(ix-ixpos)+i*15, iydel*iy+100 )
          do 150 ki = ip,2,-1
              ixray = iht
              iyray = iby + (ki-1)*idy
              call colset1( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
              ist = 19 - (ki-2)
              ixray = iadr(1,ist,iht1)
              iyray = iadr(2,ist,iht1)
              call colset( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
150       continue
              ixray = iht
              iyray = iby
          call colset1( ixray, iyray, idy1, idy2, int(10.0*rg(3,1,i)) )
              ixray = iadr(1,20,iht1)
              iyray = iadr(2,20,iht1)
              call colset( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
          do 155 ki = 2,ip
              ixray = iht
              iyray = iby - (ki-1)*idy
              call colset1( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
              ist = 19 + ki
              ixray = iadr(1,ist,iht1)
              iyray = iadr(2,ist,iht1)
              call colset( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
155       continue
          if ( i .eq. imdl ) then
             do 160 ki = ip,2,-1
                 ist = 19-(ki-2)
                 ixray = iadr(1,ist,iht1) - 100
                 iyray = iadr(2,ist,iht1) - 350
                 call colset( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
160          continue
```

```
            ixray = iadr(1,20,iht1) - 100
            iyray = iadr(2,20,iht1) - 350
            call colset( ixray, iyray, idy1, idy2, int(10.0*rg(3,1,i)) )
            do 165 ki = 2,ip
                ist = 19+ki
                ixray = iadr(1,ist,iht1) - 100
                iyray = iadr(2,ist,iht1) - 350
                call colset( ixray, iyray, idy1, idy2, int(10.0*rg(3,ki,i)) )
165         continue
          endif
        iht1 = iht1 - 1
        iht = iht - idx
300     continue
999     return
        end

*  ***************************************************************************
*  ***  PROGRAM NAME     : thradr                                           ***
*  ***  PROGRAM FUNCTION : 二次元画像データ－－＞三次元画像データ変換      ***
*  ***************************************************************************

        subroutine thradr ( iadr )

        implicit integer*4 (i-n)
        dimension  iza1(2), iza2(2), iza3(2), iza4(2), iza5(2), iza6(2)
        dimension   iadr(1:2,0:40,0:40)

        call ldcolor( 0,0,0,0,0 )
        call ldcolor( 1,100,100,100 )
        call ldcl1 (1)
        call gcmd
        iza1(1) = -685
        iza1(2) = 1487
        iza2(1) = 1179
        iza2(2) = 444
        iza3(1) = 830
        iza3(2) = 444
        isa1 = iza2(1) - iza3(1)
        iza4(1) = -10225
        iza4(2) = -7184
        iza5(1) = 810
        iza5(2) = 654
        iza6(1) = 1135
        iza6(2) = 654
        isa2 = iza6(1) - iza5(1)
        call ldcl1 ( 3 )
        ix1 = iza1(1)
        iy1 = iza1(2)
        ix3 = iza4(1)
        iy3 = iza4(2)
        id1 = isa1 / 30
        id2 = isa2 / 40
        icnt=0
        do 300 i = 30,0,-1
           ix2 = iza2(1) - (icnt*id1)
```

```
            iy2 = iza3(2)
            a1  = float((iy1-iy2)) / float((ix1-ix2) )
            b1  = float(iy1) - ( float((iy1-iy2)*ix1) / float(ix1-ix2) )
            jcnt = 0
            do 310 j = 0,40
               ix4 = iza5(1) + (jcnt*id2)
               iy4 = iza5(2)
               a2  = float(iy3-iy4) / float(ix3-ix4)
               b2  = float(iy3) - ( float((iy3-iy4)*ix3) / float(ix3-ix4) )
               iadr(1,j,i) = nint((b2-b1) / (a1-a2))
               iadr(2,j,i) = nint(((b2-b1)*a1)/(a1-a2) + b1)
               jcnt = jcnt + 1
 310        continue
            icnt = icnt + 1
 300     continue
 999     return
         end

*  ************************************************************************
*  ***  PROGRAM NAME     : colset                                        ***
*  ***  PROGRAM FUNCTION : カラーＮＯ．セット                            ***
*  ************************************************************************

         subroutine  colset ( ixray, iyray, idy1, idy2, irg )

             カラーＮＯセット処理省略

 999     return
         end

*  ************************************************************************
*  ***  PROGRAM NAME     : colset1                                       ***
*  ***  PROGRAM FUNCTION : カラーＮＯ．セット                            ***
*  ************************************************************************

         subroutine  colset1 ( ixray, iyray, idy1, idy2, irg )

             カラーＮＯセット処理省略

 999     return
         end

*  ************************************************************************
*  ***  PROGRAM NAME     : distance                                      ***
*  ***  PROGRAM FUNCTION : 初期画面出力　＆　Ｅｃｃｅｎｔｒｉｃｉｔｙ入力処　***
*  ************************************************************************

         subroutine  distance ( rdis, iend )

             初期画面出力　＆　Ｅｃｃｅｎｔｒｉｃｉｔｙ入力処理省略

  999    return
         end
```

```
*  ********************************************************************************
*  ***  SYSTEM          :  ATR  Parallel Computing Sytem                        ***
*  ***  PROJECT         :  Auditory & Visual Perception Res. Labs.              ***
*  ***  MODEL NO.       :  NCUBE/10                                             ***
*  ***  PROGRAM NAME    :  FORTRAN-77                                           ***
*  ***  MEKE OUT DATE   :  nodedsp5                                             ***
*  ***  VERSION NO.     :  01.10                                                ***
*  ********************************************************************************

      program     nodedsp5

C     message types
      integer*4 imsg_type, icflag, istatus, isrc
      integer*4 id_node, id_process, id_host, icube_size
      real rmsg_buf(8)
      integer*4 kk, inr, inx, ikx, is_node
      real  a, e, p, q, g, th, pi, gs
      real  pnintv, rdist, rgo, gmax, vrad
      real  xx1, yy1, oex, oinh, plimit, pcomp
      real  rresult(1:3,1:17), sbuf(10), rbuf(4)
      integer*4 itime1, itime2, itime3, itime4
      real  rtime(4)

      call whoami(id_node, id_process, id_host, icube_size)
      do 20 i = 1, 4
         rtime(i) = 0.
20    continue
      pi = 3.14159
      pnintv = 0.5
      rgo  = 0.0
      gmax = 0.0
      inx = 0
      ikx = 0
      oex  = 0.0
      oinh = 0.0
      isrc = -1
      imsg_type = 48
      istatus = nread( rmsg_buf, 32, isrc, imsg_type, icflag)
      e  = rmsg_buf(2)
      p  = rmsg_buf(3)
      q  = rmsg_buf(4)
      th = rmsg_buf(5)
      inr   = int( rmsg_buf(6) )
      rdist = rmsg_buf(7)
      vrad  = rmsg_buf(8)
      wrad  = vrad*3.0
      a =( 2.075/vrad )**1.23
      plimit = (p/2.0+vrad*4.0)**2
      ilnode =( 255-int(rmsg_buf(1)) )/int(rmsg_buf(1))
      sbuf(1) = float(ilnode)
      sbuf(2) = rmsg_buf(1)
      sbuf(3) = e
      sbuf(4) = p
      sbuf(5) = pnintv
      do 100 kk = 0, inr/2
```

```
            xx1 = rdist*cos( pi*2.0*kk/inr )
            yy1 = rdist*sin( pi*2.0*kk/inr )
            pcomp =( xx1-e )**2 + yy1**2
            if (((xx1-e)**2+yy1**2).gt.((p/2.0+vrad*4.0)**2)) go to 200
            sbuf(6) = vrad
            sbuf(7) = wrad
            sbuf(8) = xx1
            sbuf(9) = yy1
            is_node = ilnode*id_node + int(rmsg_buf(1))
            do 110 i = 0, ilnode-1
                imsg_type = 31
                istatus = nwrite(sbuf,36,is_node+i,imsg_type,icflag)
110         continue
            do 120 j = 0, ilnode-1
                isrc = -1
                imsg_type = 32
                istatus = nread( rbuf, 16, isrc, imsg_type, icflag)
                oex  = oex + rbuf(1)
                oinh = oinh + rbuf(2)
120         continue
            oex  = oex*pnintv*pnintv
            oinh = oinh*pnintv*pnintv
            gs   = a*( oex - q*oinh )
            gmax = max( gmax, gs )
            gr   = gs
            if ( gs .lt. th ) gs = 0.0
            if (( kk .eq. 0 ) .or. ( kk .eq. inr/2)) then
                g = gs
            else
                g  = gs*2.0
            endif
            rresult(1,2+kk) = xx1
            rresult(2,2+kk) = yy1
            rresult(3,2+kk) = gr
            rgo = rgo + g
100     continue
200     continue
        rresult(1,1) = float(id_node)
        rresult(2,1) = gmax
        rresult(3,1) = rgo
        rresult(1,17) = float(kk)
        rresult(2,17) = plimit
        rresult(3,17) = pcomp
        imsg_type = 42
        istatus = nwrite(rresult, 3*17*4, id_host, imsg_type, icflag)
        imsg_type = 41
        istatus = nwrite(rtime, 4*4, id_host, imsg_type, icflag)
        stop
        end
```

```
*  ********************************************************************************
*  ***  SYSTEM          :  ATR  Parallel Computing Sytem                        ***
*  ***  PROJECT         :  Auditory & Visual Perception Res. Labs.              ***
*  ***  MODEL NO.       :  NCUBE/10                                             ***
*  ***  PROGRAM NAME    :  FORTRAN-77                                           ***
*  ***  MEKE OUT DATE   :  nodedsp6                                             ***
*  ***  VERSION NO.     :  01.20                                                ***
*  ********************************************************************************
      program     nodedsp6

      integer*4 imsg_type, icflag, istatus, isrc
      integer*4 id_node, id_process, id_host, icube_size
      real rmsg_buf(10)
      integer*4 kk, inr, iposi
      real  el, p, q, g, th, pi, gs, x, y, x1, y1
      real  bx, ex, by, ey, bx_stat, bx_end, sho_no
      real  vwintv, vwrad, pnintv, rdist, svwrad
      real*8  oex, oinh, opt, ops, dc, dp, sp
      real  rresult(4)

      call whoami(id_node, id_process, id_host, icube_size)
1000  continue
      ops = 0.
      opt = 0.
      isrc = -1
      imsg_type = 31
      istatus = nread( rmsg_buf, 36, isrc, imsg_type, icflag)
      iposi = 0
      el = rmsg_buf(3)
      p  = rmsg_buf(4)
      vwintv = rmsg_buf(5)
      vwrad  = rmsg_buf(6)
      x1 = rmsg_buf(8)
      y1 = rmsg_buf(9)
      bx = el - p/2.0 - vwintv
      ex = el + p/2.0 + vwintv
      by = -p/2.0 - vwintv
      ey =  p/2.0 + vwintv
      sho_no =( ex - bx )/rmsg_buf(1)
      imod_no = mod( (id_node-int(rmsg_buf(2))), int(rmsg_buf(1)) )
      if ( imod_no .eq. 0 ) then
          bx_stat = bx
      else
          bx_stat = bx + sho_no*(float(imod_no)) + vwintv
      endif
      if ( imod_no .eq. (int(rmsg_buf(1))-1) ) then
          bx_end = ex
      else
          bx_end = bx + sho_no*(float(imod_no+1))
      endif
100   continue
      sp = p*p/4.0
      svwrad = vwrad**2
      ops = 0.0
      do 10 x = bx_stat, bx_end, vwintv
```

```
            do 15 y = by, ey, vwintv
                dp =( x - el )**2 + y**2
                if ( dp .gt. sp ) then
                    opt = 0.0
                else
                    dc =( x - x1 )**2 + ( y - y1 )**2
                    opt = exp( -dc/svwrad )
                    if ( dc .gt. svwrad*12.25 ) opt = 0.0
                endif
                ops = ops + opt
15          continue
10      continue
        iposi = iposi + 1
        if ( iposi .eq. 1 ) then
            vwrad = rmsg_buf(7)
            rresult(1) = ops
            go to 100
        else
            rresult(2) = ops
            rresult(3) = 0.0
            rresult(4) = 0.0
        endif
        imsg_type = 32
        isrc = (id_node-int(rmsg_buf(2)))/int(rmsg_buf(1))
        istatus = nwrite(rresult, 16, isrc, imsg_type, icflag)
        go to 1000
        stop
        end
```

7. エネルギー最小化による画像復元とエネルギー学習

プログラム　リスト

(1) ＮＣＵＢＥホスト系プログラム

ｓｒｈｏｓｔｒ　：　画像復元処理

(2) ＮＣＵＢＥノード系プログラム

ｓｒｎｏｄｅ　：　画像復元神経回路モデル処理

```
*  ***********************************************************************
*  ***    SYSTEM          : ATR Parallel Computing System          ***
*  ***    PROJECT         : Auditory & Visual Perception Res. Labs.***
*  ***    MODEL NO.       : NCUBE/10 (AXIS)                        ***
*  ***    LANGUEGE        : FORTRAN-77                             ***
*  ***    MODULE NAME     : SRHOSTR  画像復元処理                  ***
*  ***    MODULE FUNCTION : IMAGE RESTORATION by NEURAL NETWORK    ***
*  ***                    : for REAL DEMONSTRATOR                  ***
*  ***    DOCUMENT NO.    : ATR-SR-H-4-04-D                        ***
*  ***    MAKE OUT DATE   : 18-MAY-88                              ***
*  ***    VERSION         : 4.04                                   ***
*  ***********************************************************************
*  ***      #    #####  ####    Advanced Telecommunication         ***
*  ***     # #     #    #   #    Reseach Institute International   ***
*  ***    #   #    #    ####    ---------------------------------- ***
*  ***    #####    #    #  #    ATR Auditory and Visual Perception ***
*  ***    #   #    #    #   #    Research Laboratories             ***
*  ***********************************************************************
*
      program         srhostr
      common          /nodecom/ iload_work, nbuf
      integer*1       iload_work(2**14)
      integer*1       nbuf(0:257)
      common          /pixcom/ pix
      integer*1       pix(0:255, 0:255)
      character*1     pixdf(0:255, 0:255)
      equivalence     ( pix(0,0) , pixdf(0,0) )
        ＣＲＴ関係テーブル群省略
      real            simtim  ,dt     ,s
      integer         n       ,nn     ,nm     ,exp2
      integer         iter    ,dcnt   ,itercnt        ,lp     ,mode
      integer         di      ,dl     ,bl
      integer         r       ,g      ,b      ,p
      real            parms(0:6)
      integer         node(0:255)     ,xnode(0:255)
      integer         imtyp   ,imbuf  ,isrc   ,istyp  ,ich
      integer         icnt    ,idim   ,istat
      logical*1       icf(0:255)
      integer*1       xdata(0:511)
      character*1     xdatadf(0:511)
      equivalence     ( xdata(0) , xdatadf(0) )
      integer         basex   ,basey  ,undskw ,undskr ,uncslw ,uncslr
      integer         grpopt
      integer*2       xary(4) ,yary(4)
      real            gpara(4)
      real            thetax  ,thetay ,api
      character*6     ndprog
      character*10    mst     ,bas    ,org    ,nis
      character*16    c16
      character*32    c32
      integer         ixpt(64)
      data            ndprog / 'srnode' /
      data            mst    / 'srmast.dat' /
      data            bas    / 'srbase.dat' /
      data            org    / 'srorig.dat' /
```

```
      data    nis    / 'srnois.dat' /
      data    undskr /  9 /
      data    undskw / 10 /
      data    uncslr /  5 /
      data    uncslw /  6 /
      data    ixpt  / 40*112 , 24*160 /
      data    ixmx  / 54 /
c     ------<< PROCESS >>-----------------------------------------
      grpopt = 0
      api    = 2.0 * 3.14159265 / 360.0
      thetax =  20.0 * api
      thetay = 160.0 * api
      gpara(1)  = cos ( thetax )
      gpara(2)  = cos ( thetay )
      gpara(3)  = sin ( thetax )
      gpara(4)  = sin ( thetay )
        初期設定処理一部省略
      open ( unit=undskr , iostat=ists , err=9910
     #       , file=mst , status='old' , access='direct'
     #       , form='formatted' , recl=25 )
      read ( unit=undskr , fmt=8010 , iostat=ists , err=9920 , rec=1 )
     #       simtim , dt , s , lp , bl , di , dl
 8010 format (3f5.2,2i2,i4,i2)
      close ( unit=undskw )
c                                                  / setup initial picture /
      open ( unit=undskr , iostat=ists , err=9930
     #       , file=bas , status='old' , access='direct'
     #       , form='formatted' , recl=256 )
      do  1260  j = 0 , 255
        read ( unit=undskr , fmt=8020 , iostat=ists , err=9940 , rec=j+1 )
     #          (pixdf(i,j),i=0,255)
 8020   format (256a1)
 1260  continue
      close ( unit=undskw )
C                                                  / Node program setup /
      idim = 8
      icnt = 2**8
      ich = nopen(idim)
      istat = nloadm(ich, ndprog, -icnt, icf, iload_work, 2**16)

      node(0) = 0
      exp2 = 1
      do 1420 i = 1, idim
        do 1410 nm = 0, exp2 - 1
          node(exp2 * 2 - nm - 1) = node(nm) + exp2
 1410   continue
        exp2 = exp2 * 2
 1420 continue
      do 1430 i = 0, 255
        xnode(node(i)) = i
 1430 continue

      parms(0) = simtim
      parms(1) = dt
      parms(2) = s
```

```
      parms(3) = lp
      parms(4) = di
      parms(5) = dl
      parms(6) = bl
      do 1450 nn = 0, 8 - 1
        do 1440 nm = 0, (icnt / 8) - 1
          n = nm * 8 + nn
          istat = nwrite(ich, parms, 7 * 4, node(n), 10)
 1440   continue
 1450 continue

      do 1470 nn = 0, 8 - 1
        do 1460 nm = 0, (icnt / 8) - 1
          n = nm * 8 + nn
          istat = nwrite(ich, pix(0, n), 256, node(n), 11)
 1460   continue
 1470 continue
        ＣＲＴ表示処理省略
      itercnt = int(simtim / dt) / di
      if ( itercnt .gt. 1 ) then
        do  4000  i = 1 , itercnt-1
          if ( grpopt .eq. 0 ) then
            call  restor ( ich , nbuf , node , xnode , i )
          else
            call  grpout ( ich , nbuf , node , xnode , i ,gpara )
            ists = isleep ( 10 )
          endif
 4000   continue
      endif
      call  grpout ( ich , nbuf , node , xnode , itercnt , gpara )
      istat = nclose(ich)
        ＣＲＴ表示処理一部省略
c                                                  / original image disp. /
      open ( unit=undskr , iostat=ists , err=9950
     #       , file=org , status='old' , access='direct'
     #       , form='formatted' , recl=512 )
      do  1340  j = 0 , 511
        read ( unit=undskr , fmt=8030 , iostat=ists , err=9960 , rec=j+1 )
     #          (xdatadf(i),i=0,511)
 8030   format (512a1)
          ＣＲＴ表示処理一部省略
 1340  continue
 1350 continue
      close ( unit=undskr )
 9900 stop ' Normal end'
c                                                  / error process /
 9910 continue
        異常処理時省略
 9920 continue
        異常処理時省略
 9930 continue
        異常処理時省略
 9940 continue
        異常処理時省略
 9950 continue
```

```
        異常処理時省略
 9960   continue
        異常処理時省略
 9970   continue
        異常処理時省略
 9980   continue
        異常処理時省略
 9999   continue
        stop ' Abnornal end'
        end
*   ***********************************************************************
*   ***     MODULE NAME    : RESTOR      復元画像受信処理                 ***
*   ***********************************************************************
        subroutine  restor ( ich , nbuf ,node , xnode , dspi )
c                                                   / parameter symbol /
        integer          ich      ,dspi
        integer*1        nbuf(0:257)
        integer          node(0:255), xnode(0:255)
c       〈〈 PROCESS 〉〉
          ＣＲＴ表示処理省略
        do  100  nodeno = 255 , 0 , -1
          nno = node(nodeno)
          istat = nread  ( ich , nbuf , 258 , nno , 20 )
  100   continue
        return
        end
*   ***********************************************************************
*   ***     MODULE NAME    : GRPOUT      復元画像表示処理                 ***
*   ***********************************************************************
        subroutine  grpout ( ich , nbuf ,node , xnode , dspi , gpara )
c                                                   / parameter symbol /
        integer          ich      ,dspi
        integer*1        nbuf(0:257)
        integer          node(0:255), xnode(0:255)
        real             gpara(4)
c       〈〈 PROCESS 〉〉
          ＣＲＴ表示処理省略
        do  100  nodeno = 255 , 0 , -1
          nno = node(nodeno)
          istat = nread  ( ich , nbuf , 258 , nno , 20 )
          call  cbgrp    ( nodeno , nbuf , basex , basey , gain , gpara )
  100   continue
        return
        end

        subroutine  cbgrp  ( iy0 , xbuf , basex , basey , gainz , gpara )
          ２次元→３次元変換表示処理省略

        subroutine  binasc ( int , str )
          ２進数→ＡＳＣＩＩ変換処理省略
```

```
*  ***********************************************************************
*  ***    SYSTEM          : ATR Parallel Computing System          ***
*  ***    PROJECT         : Auditory & Visual Perception Res. Labs.***
*  ***    MODEL NO.       : NCUBE/10 (VERTEX)                      ***
*  ***    LANGUEGE        : FORTRAN-77                             ***
*  ***    MODULE NAME     : SRNODE   画像復元神経回路モデル処理    ***
*  ***    MODULE FUNCTION : IMAGE RESTORATION by NEURAL NETWORK    ***
*  ***                    : for NODE PROGRAM                       ***
*  ***    DOCUMENT NO.    : ATR-SR-H-4-04-D                        ***
*  ***    MAKE OUT DATE   : 18-MAY-88                              ***
*  ***    VERSION         : 4.04                                   ***
*  ***********************************************************************
*  ***      #    #####  ####    Advanced Telecommunication         ***
*  ***     # #     #    #   #    Reseach Institute International   ***
*  ***    #   #    #    ####    -------------------------------- ***
*  ***    #####    #    #  #    ATR Auditory and Visual Perception ***
*  ***    #   #    #    #   #    Research Laboratories             ***
*  ***********************************************************************
*
      program         srnode
      integer iter, itercnt
      integer*1 nbuf(0:257)
      real simtim, dt, s, rm, cv, cl, cp, cc, cg, k, bd
      real parms(0:6)
      real d(0:257), ud(0:257), dd(0:257), dx(0:257), d2(0:257)
      real h(0:256), uh(0:256), dh(0:256)
      real v(0:256), uv(0:256), dv(0:256)
      real mm(0:256), um(0:256), dm(0:256), m2(0:256)
      real nn(0:256), un(0:256), dn(0:256), n2(0:256)
      integer i, n, nm, exp2, rand, rn, up, down, down2, nd, lp, di, dl
      integer bl
      real sigmoid
      integer node(0:255)
c     -----<< PROCESS >>-----------------------------------------
      call whoami(idn, idp, idh, ics)
      istat = nread(parms, 7 * 4, idh, 10, icf)
      istat = nread(nbuf, 256, idh, 11, icf)
      n = 8
      node(0) = 0
      exp2 = 1
      do 1200 i = 1, n
          do 1210 nm = 0, exp2 - 1
              node(exp2 * 2 - nm - 1) = node(nm) + exp2
1210      continue
          exp2 = exp2 * 2
1200  continue
       do 1220 i = 1, 254
          if (node(i).eq.idn) then
              up = node(i - 1)
              down = node(i + 1)
          endif
1220  continue
*      -----   special node marking. -----
      if(idn.eq.0) then
          up = -1
```

```
        down = node(1)
      endif
      if(idn.eq.node(255)) then
        up = node(254)
        down = -1
      endif
      if(idn.eq.node(254)) then
        down2 = -1
      else
        down2 = 0
      endif
*      -----   setup parameters. -----
      simtim = parms(0)
      dt = parms(1)
      s  = parms(2)
      lp = parms(3)
      di = parms(4)
      dl = parms(5)
      bl = parms(6)
      cd = 0.2
      rm = 16.0
      av = 0.0
      itercnt = int(simtim / dt)
      cv = 0.35
      cp = 17.0
      cc = 15.0
      cg = 2.5
      cl = 40.0
*      ----- local envirements initialize. -----
      d(0)    = 0.0
      d(257)  = 0.0
      dd(0)   = 0.0
      dd(257) = 0.0
      ud(0)   = 0.0
      ud(257) = 0.0
      do 1400 i = 0, 256
        if(up.eq.-1) then
          uh(i) = 1.0
          um(i) = 100.0
        else
          uh(i) = 0.5
          um(i) = 0.0
        endif
        if(down.eq.-1) then
          h(i) = 1.0
          mm(i) = 100.0
        else
          h(i) = 0.5
          mm(i) = 0.0
        endif
        if(down2.eq.-1) then
          dh(i) = 1.0
          dm(i) = 100.0
        else
          dh(i) = 0.5
```

```
            dm(i) = 0.0
        endif
        uv(i) = 0.5
        v(i)  = 0.5
        dv(i) = 0.5
        un(i) = 0.0
        nn(i) = 0.0
        dn(i) = 0.0
1400    continue
        uv(0)   = 1.0
        uv(256) = 1.0
        v(0)    = 1.0
        v(256)  = 1.0
        dv(0)   = 1.0
        dv(256) = 1.0
        un(0)   = 100.0
        un(256) = 100.0
        nn(0)   = 100.0
        nn(256) = 100.0
        dn(0)   = 100.0
        dn(256) = 100.0
        rand = 12345678+idn+idp+idh+ics
        do 1100 i = 0, 255
            r = 0.0
            if(nbuf(i).lt.0) then
                d(i+1) = real(256+nbuf(i))
            else
                d(i+1) = real(nbuf(i))
            endif
            r = 0.0
            do 1110 rn = 1, 12
                rand = mod(163 * rand + 656329, 12518383)
                r = r + real(rand) / 12518383.0
1110        continue
            d(i+1) = d(i+1) + s * (r - 6.0) + av
            dx(i+1) = d(i+1)
1100    continue
        if(bl.ne.0) goto 1130
        if(up.ge.0) istat = nwrite(d(1), 256 * 4, up, 30, icf)
        if(down.ge.0) istat = nwrite(d(1), 256 * 4, down, 31, icf)
        if(up.ge.0) istat = nread(ud(1), 256 * 4, up, 31, icf)
        if(down.ge.0) istat = nread(dd(1), 256 * 4, down, 30, icf)
        if(up.lt.0) then
            do 1140 i = 1, 256
                ud(i) = d(i)
1140        continue
        endif
        if(down.lt.0) then
            do 1150 i = 1, 256
                dd(i) = d(i)
1150        continue
        endif
        ud(0)   = ud(1)
        dd(0)   = dd(1)
        ud(257) = ud(256)
```

```
      dd(257) = dd(256)
      d(0)    = d(1)
      d(257)  = d(256)
      do 1120 i = 1, 256
          bd = ud(i-1)+ud(i)+ud(i+1)+d(i-1)+d(i+1)+dd(i-1)+dd(i)+dd(i+1)
          bd = bd / 16 + d(i) / 2
          d(i) = bd
          dx(i) = bd
1120  continue
1130  continue
      do 1000 iter = 1, itercnt
          k = iter * dt
          do 1010 i = 0, 255
              if(dl.eq.0) then
                  if(d(i+1).gt.254.0) then
                      nbuf(i) = 254
                  else if(d(i+1).lt.1.0) then
                      nbuf(i) = 1
                  else
                      nbuf(i) = int(d(i+1))
                  endif
              else
                  nbuf(i) = (h(i+1) + v(i+1)) * 127 + 1
                  if((h(i+1)+v(i+1))*127+1.gt.254) nbuf(i) = 254
              endif
1010      continue
          nbuf(256) = mod(iter, 100)
          nbuf(257) = int(iter / 100)
          if((mod(iter,di).eq.1).or.(iter.eq.itercnt)
     +        .or.(iter.eq.1).or.(di.eq.1)) then
              istat = nwrite(nbuf, 258, idh, 20, icf)
          endif
          if(up.ge.0) istat = nwrite(d(1), 256 * 4, up, 30, icf)
          if(down.ge.0) istat = nwrite(d(1), 256 * 4, down, 31, icf)
          if(up.ge.0) istat = nread(ud(1), 256 * 4, up, 31, icf)
          if(down.ge.0) istat = nread(dd(1), 256 * 4, down, 30, icf)
          if (up.eq.-1) then
              do 1310 i = 1, 256
                  d2(i) = d(i)+dt*2*((d(i)-dd(i))*(h(i)-1)
     +                    +(d(i)-d(i-1))*(v(i-1)-1)
     +                    +(d(i)-d(i+1))*(v(i)-1)-cd*(d(i)-dx(i)))
1310          continue
          elseif (down.eq.-1) then
               do 1320 i = 1, 256
                  d2(i) = d(i)+dt*2*((d(i)-ud(i))*(uh(i)-1)
     +                    +(d(i)-d(i-1))*(v(i-1)-1)
     +                    +(d(i)-d(i+1))*(v(i)-1)-cd*(d(i)-dx(i)))
1320          continue
          else
              do 1330 i = 1, 256
                  d2(i) = d(i)+dt*2*((d(i)-ud(i))*(uh(i)-1)
     +                    +(d(i)-dd(i))*(h(i)-1)
     +                    +(d(i)-d(i-1))*(v(i-1)-1)
     +                    +(d(i)-d(i+1))*(v(i)-1)-cd*(d(i)-dx(i)))
1330          continue
```

```
        endif
        do 1300 i = 1, 256
            d(i) = d2(i)
1300    continue
*       ----- line process progress. -----
        if ((mod(iter, 10) .eq. 1).and.(lp.ne.1)) then
            if (up.ge.0) then
                istat = nwrite(mm(1), 255 * 4, up, 40, icf)
                istat = nwrite(nn(1), 255 * 4, up, 41, icf)
            endif
            if (down.ge.0) then
                istat = nwrite(mm(1), 255 * 4, down, 42, icf)
                istat = nwrite(nn(1), 255 * 4, down, 43, icf)
            endif
            if (up.ge.0) then
                istat = nread(um(1), 255 * 4, up, 42, icf)
                istat = nread(un(1), 255 * 4, up, 43, icf)
                endif
            if (down.ge.0) then
                istat = nread(dm(1), 255 * 4, down, 40, icf)
                istat = nread(dn(1), 255 * 4, down, 41, icf)
            endif
            do 1500 i = 1, 255
                uh(i) = sigmoid(rm, um(i))
                uv(i) = sigmoid(rm, un(i))
                dh(i) = sigmoid(rm, dm(i))
                dv(i) = sigmoid(rm, dn(i))
1500        continue
            if(up.eq.-1) then
            else if(down.eq.-1) then
            else if(down2.eq.-1) then
            else
                do 1520 i = 1, 255
                    m2(i) = -cv*(1-2*h(i))-cp*(uh(i)+dh(i))-cc
                    m2(i) = m2(i)-cl*((1-h(i+1)-v(i)-dv(i))**2)
                    m2(i) = m2(i)-cl*((1-h(i-1)-v(i-1)-dv(i-1))**2)
                    m2(i) = m2(i)-cl*(-2*h(i-1)*(1-h(i)-v(i-1)-dv(i-1)))
                    m2(i) = m2(i)-cl*(-2*h(i+1)*(1-h(i)-v(i)-dv(i)))
                    m2(i) = m2(i)+2*cl*(1-dv(i)-h(i)-h(i+1))
                    m2(i) = m2(i)+2*cl*(1-dv(i-1)-h(i-1)-h(i))
                    m2(i) = m2(i)+2*cl*(1-v(i)-h(i)-h(i+1))
                    m2(i) = m2(i)+2*cl*(1-v(i-1)-h(i-1)-h(i))
                    m2(i) = m2(i)/k+(dd(i)-d(i))**2-cg*mm(i)
                    m2(i) = mm(i)+dt*m2(i)
                    n2(i) = -cv*(1-2*v(i))-cp*(v(i-1)+v(i+1))-cc
                    n2(i) = n2(i)-cl*((1-dv(i)-h(i)-h(i+1))**2)
                    n2(i) = n2(i)-cl*((1-uv(i)-uh(i)-uh(i+1))**2)
                    n2(i) = n2(i)-cl*(-2*uv(i)*(1-v(i)-uh(i)-uh(i+1)))
                    n2(i) = n2(i)-cl*(-2*dv(i)*(1-v(i)-h(i)-h(i+1)))
                    n2(i) = n2(i)+2*cl*(1-h(i+1)-v(i)-dv(i))
                    n2(i) = n2(i)+2*cl*(1-uh(i+1)-uv(i)-v(i))
                    n2(i) = n2(i)+2*cl*(1-h(i)-v(i)-dv(i))
                    n2(i) = n2(i)+2*cl*(1-uh(i)-uv(i)-v(i))
                    n2(i) = n2(i)/k+(d(i+1)-d(i))**2-cg*nn(i)
                    n2(i) = nn(i)+dt*n2(i)
```

```
1520                continue
                endif
                do 1510 i = 1, 255
                    mm(i) = m2(i)
                    nn(i) = n2(i)
                    h(i) = sigmoid(rm, mm(i))
                    v(i) = sigmoid(rm, nn(i))
1510            continue
            endif
1000    continue
9999    stop
        end

        real function sigmoid(rm, x)
        real rm, x, xx
        xx = -2.0 * rm * x
        if (xx .lt. -10.0) then
            xx = -10.0
        else if (xx .gt. 10.0) then
            xx = 10.0
        endif
        if (xx .lt. 0.0) then
            sigmoid = 1.0 / (1.0 + (1 / (2.71828**(-xx))))
        else
            sigmoid = 1.0 / (1.0 + 2.71828**xx)
        endif
        return
        end
```

8. ネオコグニトロンによる手書き数字認識

プログラム リスト

(1) ＮＣＵＢＥホスト系プログラム

ｎｅｏｈｏｓｔ ： ネオコグニトロンＩ／Ｆ処理

(2) ＮＣＵＢＥノード系プログラム

ｓｃｎｏｄｅ ： Ｓ－Ｃ処理

ｖｎｏｄｅ ： ｖ ｃｅｌｌ処理

```
*       ***********************************************************************
*       ***  SYSTEM           : ATR  Parallel Computing System          ***
*       ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*       ***  MODEL NO.        : NCUBE/10    (AXIS)                      ***
*       ***  LANGUAGE         : FORTRAN-77                              ***
*       ***  PROGRAM NAME     : NEOHOST                                 ***
*       ***  PROGRAM FUNCTION : NEOCOGNITRON host program               ***
*       ***  VERSION NO.      : 4.00                                    ***
*       ***********************************************************************
*       ***    #    #####  ####      Advanced Telecommunication         ***
*       ***   # #     #    #   #      Reseach Institute International   ***
*       ***  #   #    #    ####      ---------------------------------- ***
*       ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*       ***  #   #    #    #   #      Research Laboratories             ***
*       ***********************************************************************
*
        PROGRAM NEOHOST

c       NEOHOST.F V4.00
c       Added to 4-data(pattern)by 128 nodes and 512 node          22/Mar/1988
c       Changed to load different node programs to each layer      27/Jan/1988
c       and main routine is devided into several subs.
c       Added loading procedures.                                  05/Jan/1988
c       neohost.f(NCUBE) Ver 1.0   Rewrite by M.Maekawa            03/Dec/1987
c       neohost.for V1.00       written by T. Ito and K. Fukushima 16-Nov-1987
c       NEOCOGNITRON host program on Ncube

c       ----- NEOCOGNITRON Parameters of Host. -----
$include(hostparam.inc)
c       ----- [Variables] -----
        integer         Nstart(4), Ndim(4)
        data            Nstart/Nstart1, Nstart2, Nstart3, Nstart4/
        data            Ndim  /  Ndim1,   Ndim2,   Ndim3,   Ndim4/
c       ----- Tables of S responses. -----
        real            U0tab(NU02,4)
        common /Utab/   U0tab
        real            U4tab( Kc4 )
        integer*2       Kctab( Kc4 )
        real            rtime(4)
c       *********************************
c       ****** Initialize all nodes ******
c       *********************************
c       ----- Open the data file. -----
        call OPENDATA
c       ----- Open 256 nodes. ------
        ichan = nopen(Ndim0)
            if (ichan.le.0) stop ' Node open error ( main ) '
            write (6,'(1x,a,i3)') ' Open hypercube channel ', ichan
c       ----- Load programs to each node. -----
        call Loadpgms (ichan)
c       ----- Read data and load them to each node. -----
        call DataSend (ichan)
c       ---------- Set Display taitol. ----------
        call gamen_set(0)
c       ---------------------------------------
```

```
c       ---------- Start calculation. ----------
c       ----------------------------------------
        isetf = 0
20      continue
            write(6,'(a)') ' Please tablet key input ! '
        call tablet_set ( iend_flag )
        if ( iend_flag .ne. 0 ) go to 80
c       ----- Trigger for starting timer counter -----
        do 30 iketa = 1, 4
            NVno = Nstart(4)+128*(iketa-1)
            istat = nwrite(ichan, idum, 4, NVno, 40)
            if (istat .ne. 0) stop ' Nwrite error in sending Trigger to V'
30      continue
c       ----- Send input data to Vnode1 -----
        do 40 iketa = 1, 4
            NVno = Nstart(1)+128*(iketa-1)
            istat = nwrite(ichan,U0tab(1,iketa),NU02b,NVno,Msg_UCV)
            if (istat .ne. 0) stop ' Nwrite error in sending U0tab to V'
40      continue
c       ----- Get the result. --------
        do 50 iketa = 1,4
            isrc = -1
            length1 = nread(ichan, U4tab, Kc4*4, isrc, Msg_UCH)
            length2 = nread(ichan, Kctab, Kc4*2, isrc, Msg_KCH)
            iposi = isrc/128
            iposi = iposi + 1
c           -----  Call dispUc ( CRT OUTPUT ) -----
            call dispUc ( iposi, length1/4, Kctab)
50      continue
c       ----- Result Working time from Vnode. -----
        do 60 iketa = 1,4
            isrc = -1
            length3 = nread(ichan, rtime_buff, 4, isrc, 41)
            rtime(isrc/128+1) = rtime_buff
60      continue
        write(6,'(a,4f11.6)') ' Result working times(Sec) :',(rtime(i),i=1,4)
        go to 20
c       ----- Continue or End ? -----
80      continue
        write (6,'($,a)') ' NEOCOGNITRON : Continue(=1) or End(=0) ? ===> '
        read (5,*) iyes
        if ( iyes .eq. 1 ) go to 20
c       ----- Normal end program routine for neohost. -----
90      continue
        do 100 i = 1,4
            close (unit=15+i)
100     continue
        write (6,'(1x,a,i3)') ' Closing cube on channel ',ichan
        istat = nclose(ichan)
            if ( istat .ne. 0 ) stop ' Nclose error in Neohost '
        stop ' Normal termination ( neohost-main ) '
        end

c       --------------------
        subroutine OPENDATA
```

```
c       -------------------
c       Open data files.
        ierr = 10
        OPEN (UNIT=16, IOSTAT=is, ERR=999, FILE='neopac01.dat',
     1       STATUS='OLD', ACCESS='SEQUENTIAL', FORM='UNFORMATTED')
        ierr = 20
        OPEN (UNIT=17, IOSTAT=is, ERR=999, FILE='neopac02.dat',
     1       STATUS='OLD', ACCESS='SEQUENTIAL', FORM='UNFORMATTED')
        ierr = 30
        OPEN (UNIT=18, IOSTAT=is, ERR=999, FILE='neopac03.dat',
     1       STATUS='OLD', ACCESS='SEQUENTIAL', FORM='UNFORMATTED')
        ierr = 40
        OPEN (UNIT=19, IOSTAT=is, ERR=999, FILE='neopac04.dat',
     1       STATUS='OLD', ACCESS='SEQUENTIAL', FORM='UNFORMATTED')
        return

c       ----- error routine for OPENDATA. -----
999     write (6,'(1x,a,i4,a,i4)')
     1      ' (neohost.opendata) File OPEN  ERR Status =',
     1       is,' ERR posi',ierr
        do 10 i = 1,4
            close (unit=15+i)
10      continue
        write (6,'(1x,a,i3)') ' Closing cube on channel ',ichan
        istat = nclose(ichan)
            if ( istat .ne. 0 ) stop ' Nclose error in Neohost.opendata '
        STOP ' Abnormal stop ( Neohost - OPENDATA ) '
        end

c       ---------------------------
        subroutine Loadpgms (ichan)
c       ---------------------------
$include(hostparam.inc)
$include(hostvar.inc)
        integer         Nstart(4), Ndim(4)
        data            Nstart/Nstart1, Nstart2, Nstart3, Nstart4/
       .data            Ndim  /  Ndim1,   Ndim2,   Ndim3,   Ndim4/
        character*(9)   cfile, cfile1, cfile2, cfile3, cfile4
        parameter       (cfile1 = 'scnode1', cfile2 = 'scnode2')
        parameter       (cfile3 = 'scnode3', cfile4 = 'scnode4')
        character*(8)   vfile, vfile1, vfile2, vfile3, vfile4
        parameter       (vfile1 = 'vnode1', vfile2 = 'vnode2')
        parameter       (vfile3 = 'vnode3', vfile4 = 'vnode4')
        integer*1       iload_work(2**14)
        logical*1       nflags(0:Nnode)
c       ----- Loading SCnode...
        do 40 iketa = 1,4
            do 30 L = 1, 4
                do 10 i = 0, Nnode
                    nflags(i) = .false.
10              continue
                kstart = Nstart(L)+1+128*(iketa-1)
                kend   = Nstart(L)+Ks(L)+128*(iketa-1)
                do 20 K = kstart, kend
                    NodeNo = ngray(K, Nstart(L)+128*(iketa-1), Ndim(L))
```

```
                nflags(NodeNo) = .true.
20              continue
                if (L .eq. 1) then
                    cfile = cfile1
                elseif (L .eq. 2) then
                    cfile = cfile2
                elseif (L .eq. 3) then
                    cfile = cfile3
                else
                    cfile = cfile4
                endif
                istat = nloadm(ichan,cfile,Nnode+1,nflags,iload_work,2**13)
                    if (istat .ne. 0) stop 'Error in loading Scnode.'
30          continue
40      continue
        write(6,*) ' Loading end Scnode ...'
c       ----- Loading Vnode...
        do 70 iketa = 1, 4
            do 60 L = 1, 4
                Kstart = Nstart(L)+128*(iketa-1)
                nflags(Kstart) = .false.
                if (L .eq. 1) then
                    vfile = vfile1
                elseif (L .eq. 2) then
                    vfile = vfile2
                elseif (L .eq. 3) then
                    vfile = vfile3
                else
                    vfile = vfile4
                endif
                istat = nload(ichan, vfile, Kstart, iload_work, 2**13)
                    if (istat .ne. 0) stop ' Error in loading Vnode.'
60          continue
70      continue
        write(6,*) ' Loading end Vnode ...'
        return
        end

c       ---------------------------
        subroutine DataSend (ichan)
c       ---------------------------
$include(hostparam.inc)
$include(hostvar.inc)

        integer         Nstart(4), Ndim(4)
        data            Nstart/Nstart1, Nstart2, Nstart3, Nstart4/
        data            Ndim  /  Ndim1,   Ndim2,   Ndim3,   Ndim4/
        do 10 L = 1, 4
            call disk( a, ia0, ia, b, c, d, join, iamax,
     1                 Kc(L-1), Ks(L), Na(L), Nd(L), L )
            call send(ichan, L, Nstart(L), Ndim(L),
     1                 a, ia0, ia, b, c, d,
     1                 join, iamax, Kc(L-1), Ks(L),
     1                 (2*Na(L)+1)**2, Nd(L))
            write(6,'(1x,a,i3)') ' Call and Return (disk/send) Layer = ',L
```

```
10      continue
        return
        end

c       ------------------------------------------------------------
        subroutine disk ( a, ia0, ia, b, c, d, join, iamax,
     1                    KcMax, KsMax, Na, Nd ,L )
c       ------------------------------------------------------------
        real            a(-Na:Na, -Na:Na, KcMax*KsMax), b(KsMax)
        real            c(-Na:Na, -Na:Na), d(-Nd:Nd, -Nd:Nd)
        integer*2       ia(KcMax,KsMax)
        integer         ia0(KsMax+1), join(KsMax)
        ierr = 120 + L
        read (UNIT=15+L, IOSTAT=IS, ERR=998)
     1       IAMAX,
     2      (Ia0(M), M=1,KsMax+1),
     3     ((Ia(M,N), M=1,KcMax), N=1,KsMax),
     4      (join(M), M=1,KsMax),
     1     ((c(M,N), M=-NA,NA), N=-NA,NA),
     2     ((d(M,N), M=-ND,ND), N=-ND,ND),
     3      (b(M), M=1,KsMax),
     4    (((a(M,N,K), M=-NA,NA), N=-NA,NA), K=1,IAMAX)
        return
c       ----- Error routine for DISK. -----
998     write (6,'(1x,a,i4,a,i4)')
     1       ' (neohost) File READ  ERR Status =',is,' ERR posi',ierr
        do 100 i = 1,4
            CLOSE (UNIT=15+i)
100     continue
        STOP ' Abnormal Stop ( neohost_sub DISK ) '
        end

c       ------------------------------------------------------------
        subroutine send( ichan, L, Nstart, Ndim,
     1                   a, ia0, ia, b, c, d,
     1                   join, iamax, KcMaxs, KsMaxs, Naa, Nd )
c       ------------------------------------------------------------
        real            a(Naa, *), b(KsMaxs)
        real            c(Naa), d(-Nd:Nd, -Nd:Nd)
        integer         ia0(KsMaxs+1)
        integer*2       ia(KcMaxs,KsMaxs)
        integer*4       join(KsMaxs)
        integer*4       ibuf(10)
        logical*4       joined, joining
c       ----- NEOCOGNITRON Parameters of Host. -----
$include(hostparam.inc)
c       ---- Send parameter and data to V node ----
        do 20 iketa = 1, 4
            Kstart = Nstart+128*(iketa-1)
            istat = nwrite(ichan, c, Naa*4, Kstart, Msg_CNV)
                if (istat.ne.0) stop ' Nwrite error in sending c to V '
            Kc = 0
            do 30 Ks = 1, KsMaxs
                if (join(Ks) .eq. Ks) then
                    Kc = Kc + 1
```

```
                joined = .false.
            else
                joined = .true.
            endif
            if (join(Ks+1) .eq. join(Ks)) then
                joining = .true.
            else
                joining = .false.
            endif
            ibuf(1) = Ks
            ibuf(2) = joining
            ibuf(3) = joined
            ibuf(4) = Kc
            NodeNo  = ngray(Ks+Kstart, Kstart, Ndim)
c           ---- Sending integer parameter SCnode ----
            istat = nwrite(ichan, ibuf, 16, NodeNo, Msg_INS)
            if (istat.ne.0) stop ' Nwrite error in sending int data to S '
c           ---- Sending ia ----
            istat = nwrite(ichan, ia(1,Ks), KcMaxs*2, NodeNo, Msg_IANS)
            if (istat.ne.0) stop ' Nwrite error in sending ia to S '
c           ---- Sending a and b ----
            iastart = ia0(Ks) + 1
            tempa   = a(1, ia0(Ks+1)+1)
            a(1, ia0(Ks+1)+1) = b(Ks)
            Nall = ((ia0(Ks+1)-ia0(Ks))*Naa+1)*4
            istat = nwrite(ichan, a(1,iastart), Nall, NodeNo, Msg_ANS)
            if (istat.ne.0) stop ' Nwrite error in sending a to S '
            a(1, ia0(Ks+1)+1) = tempa
30          continue
c       ---- Send d to V, which will broadcast to all nodes in subcube ---
            istat = nwrite(ichan,d,((2*Nd+1)**2)*4,Kstart,Msg_DNV)
                if (istat.ne.0) stop ' Nwrite error in sending d to V '
20      continue
        return
        end

c---------------------------------------
        function ngray(Myid, Nstart, Ndim)
c---------------------------------------
        Id    = Myid-Nstart
        ngray = (Id .neqv. ishft(Id, -1)) + Nstart
        return
        end
```

```
*       ************************************************************
*       ***  SYSTEM           : ATR  Parallel Computing System      ***
*       ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*       ***  MODEL NO.        : NCUBE/10    (VERTEX)                ***
*       ***  LANGUAGE         : FORTRAN-77                          ***
*       ***  PROGRAM NAME     : SCNODE1                             ***
*       ***  PROGRAM FUNCTION : NEOCOGNITRON calculate S and C responses***
*       ***  VERSION NO.      : 3.00                                ***
*       ************************************************************
*       ***    #    #####  ####      Advanced Telecommunication     ***
*       ***   # #     #    #   #      Reseach Institute International ***
*       ***  #   #    #    ####      ------------------------------ ***
*       ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*       ***  #   #    #    #   #      Research Laboratories         ***
*       ************************************************************
*
      PROGRAM SCNODE1

c     scnode.f Ver3.00  Created on 26-Jan-1988 by T. Ito and K. Fukushima
c     NEOCOGNITRON node program to calculate S and C responses.
c     Each node calculate a pair of S-C plane response.
c     All nodes in the same layer should be arranged in a subcube.
c     The node of the lowest number in the subcube is supposed to be a V-cell.

c     ----- NEOCOGNITRON Parameters of Node. -----
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c     ----- Node allocation flags -----
      logical*1       nflags(0:Jnode)
c     ----- Connection weights of S-C and C-S -----
      real            a(Naall), d(Nd12)
c     ----- Connection table of a -----
c     if connected ia(Ka) = start address in array a
      integer*2       ia(Kc0)
c     ----- Inhibitory connection for this node. -----
      real            b, brr
      common /conct/  brr, a, b, d, ia
      integer*1       MaxNa(-Nc0:Nc0), MinNa(-Nc0:Nc0)
      integer*1       MaxNd(-Ns1:Ns1), MinNd(-Ns1:Ns1)
      common /maxnm/  MaxNa, MinNa, MaxNd, MinNd
c     ----- Array for cell response. -----
      real            Us(-Ns1:Ns1,-Ns1:Ns1), V(-Ns1:Ns1,-Ns1:Ns1)
      common /Uss/    Us
      common /UsV/    V
      logical*4       NonZeroUs
      common /flags/  NonZeroUs
c     ----- Tables of C responses. -----
      real            Uctab(MaxUctab)
      common /TabUc/  Uctab
      integer*2       Kctab(MaxKctab)
      common /Tabbuf/ KcMax, Kctab
c     ----- join control flag. -----
      logical*4       joining, joined
      integer*4       iparam(10)
c             ************************
```

```
c               ***** Initialization *****
c               *************************
        call whoami(Myid, Idproc, Idhost, Ncube)
        iketa = Myid/128
        iketa = 128*iketa
        kstart = Nstart + iketa
        call Flagset (Ks1, Kstart, Ndim, Jnode, nflags)
        istatus = nread(iparam, 16, Idhost, Msg_INS, icflag)
        Ks      = iparam(1)
        joining = iparam(2)
        joined  = iparam(3)
        Kc      = iparam(4)
        Nia2 = nread(ia, Kc0*2, Idhost, Msg_IANS, icflag)
        Nall4 = nread(a, Naallb+4, Idhost, Msg_ANS, icflag)
        Nia  = Nia2/2
        Nall = Nall4/4
        call brdcast(Myid, Kstart, Ndim, d, idum, nflags, Msg_R_B)
        b = a(Nall)
        brr = b*rr
        call creMNindex(MaxNa, MinNa, Nc0, Ns1, Na1, 1)
        call creMNindex(MaxNd, MinNd, Ns1, Nc1, Nd1, Nt1)
        if (joining) joinfrom = neibid(Myid, Kstart, Ndim, 1)
        if (joined) jointo = neibid(Myid, Kstart, Ndim, -1)
c               ****************************
c               ***** Start calculation *****
c               ****************************
10      continue
        call brdcast(Myid, Kstart, Ndim, Uctab, NUc, nflags, Msg_UCT)
        KcMax = 1
        Kctab(1) = 1
c       ----- Calculate numerators of S resopnses ------
        call Sres1
c       ----- Get V plane response. -----
        call brdcast(Myid, Kstart, Ndim, V, idum, nflags, Msg_SVP)
c       ----- Get final S responses -------
        if (NonZeroUs) call Sres2
c       ----- Joining and Joined -------
        if (joining) then
            length = nread(Uctab, Ns12b, joinfrom, Msg_JOU, icflag)
            if (length .ne. 0) then
                call AddUs
                NonZeroUs = .true.
            end if
        endif
        if (joined) then
            if (NonZeroUs) then
                istatus = nwrite(Us, Ns12b, jointo, Msg_JOU, icflag)
            else
                istatus = nwrite(Us, 0, jointo, Msg_JOU, icflag)
            end if
            NonZeroUs = .false.
        endif
c       ----- Get Uc responses if not all-zero.------
        if (NonZeroUs) then
            call Cres
```

```
            Nuc = Nc12b
            Kctab(1) = Kc
            NKc = 2
        else
            Nuc = 0
            NKc = 0
        end if
        call gather(Myid, Kstart, Ndim, Uctab, Nuc, nflags, Msg_UCG)
        call gather(Myid, Kstart, Ndim, Kctab, NKc, nflags, Msg_KCG)
        goto 10
        end

c       -----------------
        subroutine Sres1
c       -----------------
c       Get the first half of S calculation from Uc responses.
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c       ----- Connection weights of S-C and C-S -----
        real            a(Na12,Kc0), d(-Nd1:Nd1,-Nd1:Nd1)
c       ----- Connection table of a -----
c       if connected ia(Ka) = start address in array a
        integer*2       ia(Kc0)
c       ----- Inhibitory connection for this node. -----
        real            b, brr
        common /conct/  brr, a, b, d, ia
        integer*1       MaxNa(-Nc0:Nc0), MinNa(-Nc0:Nc0)
        integer*1       MaxNd(-Ns1:Ns1), MinNd(-Ns1:Ns1)
        common /maxnm/  MaxNa, MinNa, MaxNd, MinNd
c       ----- Array for cell response. -----
        real            Us(Ns12)
        common /Uss/    Us
        logical*4       NonZeroUs
        common /flags/  NonZeroUs
c       ----- Tables of C responses. -----
        real            Uctab(Nc02, Kc0)
        common /TabUc/  Uctab
        integer*2       Kctab(MaxKctab)
        common /Tabbuf/ KcMax, Kctab
c       --- Get S plane response in direct for Layer 1 ---
        do 10 M = 1, Ns12
            Us(M) = 0.0
10      continue
        NonZeroUs = .false.
        do 30 Kcp = 1, KcMax
            Kc = Kctab(Kcp)
            iaKc = ia(Kc)
c           ----- Get S plane from one C plane. -----
            if (iaKc .ne. 0) then
                NonZeroUs = .true.
                call conv(Uctab(1,Kcp), a(1,iaKc))
            endif
30      continue
        return
        end
```

```
c       -------------------------
        subroutine conv(Uctab, a)
c       -------------------------
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c
        real            Us(-Ns1:Ns1,-Ns1:Ns1)
        common /Uss/    Us
        integer*1       MaxNa(-Nc0:Nc0), MinNa(-Nc0:Nc0)
        integer*1       MaxNd(-Ns1:Ns1), MinNd(-Ns1:Ns1)
        common /maxnm/  MaxNa, MinNa, MaxNd, MinNd
        real            Uctab(-Nc0:Nc0,-Nc0:Nc0)
        real            a(-Na1:Na1,-Na1:Na1)
        do 70 Inc = -Nc0, Nc0
            MinNaIn = MinNa(Inc)
            MaxNaIn = MaxNa(Inc)
            do 60 Imc = -Nc0, Nc0
                UUc = Uctab(Imc, Inc)
                if (UUc .gt. 0.0) then
                    MinNaIm = MinNa(Imc)
                    MaxNaIm = MaxNa(Imc)
                    do 50 NN = MinNaIn, MaxNaIn
                        INN = Inc - NN
                        do 40 MM = MinNaIm, MaxNaIm
                            aa = a(Imc-MM, INN)
                            if (aa .ne. 0.0) then
                                Us(MM,NN) = Us(MM,NN) + aa*UUc
                            end if
40                      continue
50                  continue
                endif
60          continue
70      continue
        return
        end

c       -----------------
        subroutine Sres2
c       -----------------
c       Get the last half of S calculation and create the S response table.
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c       ----- Array for cell response. -----
        real            Us(Ns12), V(Ns12)
        common /Uss/    Us
        common /UsV/    V
        real            brr
        common /conct/  brr
        logical*4       NonZeroUs
        common /flags/  NonZeroUs
c       ----- Get Us responses. -----
        do 10 M = 1, Ns12
            if (Us(M) .gt. 0.0) then
                Us(M) = r * ((sig+Us(M))/(sig+brr*V(M))-1.0)
```

```
            endif
10      continue
c       ----- Threshold. -----
        NonZeroUs = .false.
        do 20 M = 1, Ns12
            if (Us(M) .gt. 0.0) then
                NonZeroUs = .true.
            else
                Us(M) = 0.0
            end if
20      continue
        return
        end

c       -----------------
        subroutine addUs
c       -----------------
c       Add joined Us table to the array Us.
$include(nodeparam1.inc)
$include(nodeparam0.inc)
        real            Uctab(Ns12)
        common /TabUc/  Uctab
        integer*2       Kctab(MaxKctab)
        common /Tabbuf/ KcMax, Kctab
c       ----- Array for cell response. -----
        real            Us(Ns12)
        common /Uss/    Us
        logical*4       NonZeroUs
        common /flags/  NonZeroUs
        if (NonZeroUs) then
            do 10 M = 1, Ns12
                if (Uctab(M) .ne. 0.0) then
                    Us(M) = Us(M) + Uctab(M)
                end if
10          continue
        else
            do 20 M = 1, Ns12
                Us(M) = Uctab(M)
20          continue
        end if
        return
        end

c       ---------------
        subroutine Cres
c       ---------------
c       Get C plane responses
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c       ----- Connection weights of S-C and C-S -----
        real            a(-Na1:Na1,-Na1:Na1,Kc0), d(-Nd1:Nd1,-Nd1:Nd1)
c       ----- Connection table of a -----
c       if connected ia(Ka) = start address in array a
        integer*2       ia(Kc0)
c       ----- Inhibitory connection for this node. -----
```

```
      real              b, brr
      common /conct/    brr, a, b, d, ia
      integer*1         MaxNa(-Nc0:Nc0), MinNa(-Nc0:Nc0)
      integer*1         MaxNd(-Ns1:Ns1), MinNd(-Ns1:Ns1)
      common /maxnm/    MaxNa, MinNa, MaxNd, MinNd
c     ----- Array for cell response. -----
      real              Us(-Ns1:Ns1,-Ns1:Ns1)
      common /Uss/      Us
      logical*4         NonZeroUs
      common /flags/    NonZeroUs
c     ----- Tables of C responses. -----
      real              Uctab(-Nc1:Nc1,-Nc1:Nc1)
      common /TabUc/    Uctab
      integer*2         Kctab(MaxKctab)
      common /Tabbuf/   KcMax, Kctab

      do 4 N = -Nc1, Nc1
          do 2 M = -Nc1, Nc1
              Uctab(M,N) = 0.0
2         continue
4     continue
      if (Nt1 .eq. 1) then
          do 40 Inc = -Ns1, Ns1
              MinNdIn = MinNd(Inc)
              MaxNdIn = MaxNd(Inc)
              do 30 Imc = -Ns1, Ns1
                  UUs = Us(Imc, Inc)
                  if (UUs .gt. 0.0) then
                      MinNdIm = MinNd(Imc)
                      MaxNdIm = MaxNd(Imc)
                      do 20 NN = MinNdIn, MaxNdIn
                          INN = Inc - NN
                          do 10 MM = MinNdIm, MaxNdIm
                              Uctab(MM,NN) = Uctab(MM,NN)
     1                                       + d(Imc-MM,INN)*UUs
10                        continue
20                    continue
                  endif
30            continue
40        continue
      else
          do 80 Inc = -Ns1, Ns1
              MinNdIn = MinNd(Inc)
              MaxNdIn = MaxNd(Inc)
              do 70 Imc = -Ns1, Ns1
                  UUs = Us(Imc, Inc)
                  if (UUs .gt. 0.0) then
                      MinNdIm = MinNd(Imc)
                      MaxNdIm = MaxNd(Imc)
                      do 60 NN = MinNdIn, MaxNdIn
                          INN = Inc - NN*2
                          do 50 MM = MinNdIm, MaxNdIm
                              Uctab(MM,NN) = Uctab(MM,NN)
     1                                       + d(Imc-MM*2,INN)*UUs
50                        continue
```

```
60                      continue
                    endif
70                continue
80            continue
          end if
          do 100 N = -Nc1, Nc1
              do 90 M = -Nc1, Nc1
                  UUc = Uctab(M,N)
                  if (UUc .gt. 0.0) then
                      Uctab(M,N) = UUc/(UUc+alpha)
                  end if
90            continue
100       continue
          return
          end

c         ------------------------------------------
          function neibId ( Myid, Nstart, Ndim, idir )
c         ------------------------------------------
c         Get neighbor node in gray code order.
c         idir: direction ... 1 ---> Upper neighbor
c                         ...-1 ---> Lower neighbor
c         Degray Myid
          neibId = Myid - Nstart
          idbuff = neibId/2
10        neibId = idbuff .neqv. neibId
          idbuff = idbuff/2
          if (idbuff .ne. 0) goto 10
          neibId = neibId + idir
c         ----- Gray of neibId. -----
          neibId = (neibId .neqv. (neibId/2)) + Nstart
          return
          end
```

```
*       ************************************************************
*       ***  SYSTEM           : ATR  Parallel Computing System     ***
*       ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*       ***  MODEL NO.        : NCUBE/10    (VERTEX)               ***
*       ***  LANGUAGE         : FORTRAN-77                         ***
*       ***  PROGRAM NAME     : VNODE                              ***
*       ***  PROGRAM FUNCTION : V cell of NEOCOGNITRON             ***
*       ***  VERSION NO.      : 4.00                               ***
*       ************************************************************
*       ***    #    #####  ####      Advanced Telecommunication    ***
*       ***   # #     #    #   #      Reseach Institute International ***
*       ***  #   #    #    ####      ------------------------------ ***
*       ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*       ***  #   #    #    #   #      Research Laboratories        ***
*       ************************************************************
*
        PROGRAM VNODE1

c       Vnode1.f Ver. 4.0 rewrite by M. Maekawa              on 22/Mar/1988
c       vnode.for V3.00  Created on 27-Jan-1988  by T. Ito and K. Fukushima
c       V cell node program of NEOCOGNITRON

c       ----- NEOCOGNITRON Parameters of Node. -----
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c       ----- Node allocation flags -----
        logical*1       nflags(0:Jnode)
c       ----- Recieve and Send node table. -----
        integer*4       ircv(4), isnd(4)
        data            ircv / -1, Nstart1, Nstart2, Nstart3 /
        data            isnd / Nstart2, Nstart3, Nstart4, -1 /
c       ----- Connection weights of C-V and V-C -----
        real            c(Na12), d(Nd12)
        common /conct/  c, d
        integer*1       MaxNa(-Nc0:Nc0), MinNa(-Nc0:Nc0)
        integer*1       MaxNd(-Ns1:Ns1), MinNd(-Ns1:Ns1)
        common /maxnm/  MaxNa, MinNa, MaxNd, MinNd
c       ----- Array for cell response. -----
        real            V(-Ns1:Ns1,-Ns1:Ns1)
        common /VUc/    V
c       ----- Tables of C responses. -----
        real            Uctab(MaxUctab)
        integer*2       Kctab(MaxKctab)
        common /Tabbuf/ KcMax, Kctab, Uctab
c       ---------------------------------------
c       ----------  Initialization  ------------
c       ---------------------------------------
        call whoami(Myid, Idproc, Idhost, Ncube)
        iketa = Myid/128
        iketa = 128*iketa
        kstart = Nstart + iketa
        call Flagset (Ks1, Kstart, Ndim, Jnode, nflags)
        Ncc4 = nread(c, Na12b, Idhost, Msg_CNV, icflag)
        Ndd4 = nread(d, Nd12b, Idhost, Msg_DNV, icflag)
        call brdcast(Myid, Kstart, Ndim, d, Ndd4, nflags, Msg_R_B)
```

```
        call creMNindex(MaxNa, MinNa, Nc0, Ns1, Na1, 1)
        call creMNindex(MaxNd, MinNd, Ns1, Nc1, Nd1, Nt1)
        idrcv = Idhost
        idsnd = Isnd(Layer) + iketa
        Msg_type1 = Msg_U(Layer)
        Msg_type2 = Msg_M(Layer)
        Msg_type3 = Msg_U(Layer+1)
        Msg_type4 = Msg_M(Layer+1)

c       ------------------------------------------
c       ------------  Start calculation ------------
c       ------------------------------------------
10      continue
        NUc = nread(Uctab, MaxUctabb, idrcv, Msg_Type1, icflag)
        KcMax = 1
        call brdcast(Myid, Kstart, Ndim, Uctab, NUc, nflags, Msg_UCT)
c       ------- Get V response -------
        call Vres
c       ----- Send V plane response. -----
        call brdcast(Myid, Kstart, Ndim, V, Ns12b, nflags, Msg_SVP)
c       ----- Receive Us responses tables from all other nodes in this subcube.
        Nuc = 0
        NKc = 0
        call gather(Myid, Kstart, Ndim, Uctab, Nuc, nflags, Msg_UCG)
        call gather(Myid, Kstart, Ndim, Kctab, NKc, nflags, Msg_KCG)
c       ----- Send them to the root of the next subcube. -----
        istatus = nwrite(Uctab, Nuc, idsnd, Msg_Type3, icflag)
        istatus = nwrite(Kctab, NKc, idsnd, Msg_Type4, icflag)
        goto 10
        end

c       -----------------
        subroutine Vres
c       -----------------
c       Get C response and the first half of S calculation.
$include(nodeparam1.inc)
$include(nodeparam0.inc)
c       ----- Connection weights of C-V -----
        real            c(-Na1:Na1,-Na1:Na1)
        common /conct/  c
        integer*1       MaxNa(-Nc0:Nc0), MinNa(-Nc0:Nc0)
        integer*1       MaxNd(-Ns1:Ns1), MinNd(-Ns1:Ns1)
        common /maxnm/  MaxNa, MinNa, MaxNd, MinNd
c       ----- Array for cell response. -----
        real            V(-Ns1:Ns1,-Ns1:Ns1)
        common /VUc/    V
c       ----- Tables of Uc responses. -----
        real            Uctab(-Nc0:Nc0,-Nc0:Nc0, Kc0)
        integer*2       Kctab(MaxKctab)
        common /Tabbuf/ KcMax, Kctab, Uctab
        do 20 N = -Ns1,Ns1
            do 10 M = -Ns1,Ns1
                V(M,N) = 0.0
10          continue
20      continue
```

```
c       ----- Get squared Uc responses -----
c       ----- The results are returned in the first plane of Uctab.
        do 70 Inc = -Nc0, Nc0
            MinNaIn = MinNa(Inc)
            MaxNaIn = MaxNa(Inc)
            do 60 Imc = -Nc0, Nc0
                UUc = Uctab(Imc, Inc,1)
                if (UUc .gt. 0.0) then
                    MinNaIm = MinNa(Imc)
                    MaxNaIm = MaxNa(Imc)
                    do 50 NN = MinNaIn, MaxNaIn
                        IncNN = Inc-NN
                        do 40 MM = MinNaIm, MaxNaIm
                            V(MM,NN) = V(MM,NN) + c(Imc-MM,IncNN)*UUc
40                      continue
50                  continue
                endif
60          continue
70      continue
        do 90 N = -Ns1, Ns1
            do 80 M = -Ns1, Ns1
                V(M,N) = sqrt(V(M,N))
80          continue
90      continue
        return
        end
```

9. 3層逆伝播学習ネットワークによる画像データ圧縮

プログラム リスト

(1) ＮＣＵＢＥホスト系プログラム

ｎｃｂ３００ ： ニューロ・コーデック学習処理

(2) ＮＣＵＢＥノード系プログラム

ｎｃｃ３００ ： ＮＣ画像変換処理

ｎｃｎ３００ ： ＮＣ神経回路モデル処理

```
*  ***********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System          ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*  ***  MODEL NO.        : NCUBE/10    (AXIS)                      ***
*  ***  LANGUAGE         : FORTRAN-77                              ***
*  ***  PROGRAM NAME     : NCB300    ニューロ・コーデック学習処理  ***
*  ***  PROGRAM FUNCTION : NEURO CODEC (BACKPROPAGATE TYPE)        ***
*  ***  DOCUMENT NO.     : ATR-NC-A-3-00-D                         ***
*  ***  MAKE OUT DATE    : 02-SEP-88                               ***
*  ***  VERSION NO.      : 03.00                                   ***
*  ***********************************************************************
*  ***    #    #####  ####      Advanced Telecommunication         ***
*  ***   # #     #    #   #      Reseach Institute International   ***
*  ***  #   #    #    ####      ---------------------------------- ***
*  ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*  ***  #   #    #    #   #      Research Laboratories             ***
*  ***********************************************************************
*
      program          ncb300
$      include         (nccomh.h)
c     --------<< PROCESS >>-----------------------------------------
       ists = 0
       call  ncbs100 ( ists )
       if ( ists .ne. 0 ) goto 9910
       call  ncbs200 ( ists )
       if ( ists .ne. 0 ) goto 9910
       if ( iwset_mod .eq. 1 ) then
         call  ncbs300 ( ists )
         if ( ists .ne. 0 ) goto 9910
       else
         call  ncbsg00 ( ists )
         if ( ists .ne. 0 ) goto 9910
       endif
       if ( icrt_mod .ge. 1 ) then
         call  ssclrs  ( 0 )
         call  ncbs400 ( 1 , ists )
         if ( ists .ne. 0 ) goto 9910
       endif
       do  1100  ip_cnt = 1 , iport_cnt
         call  ncbs500 ( ip_cnt , ists )
         if ( ists .ne. 0 ) goto 9910
         call  ncbs600 ( ip_cnt , ists )
         if ( ists .ne. 0 ) goto 9910
 1100  continue
       call  ncbs700 ( ists )
       if ( ists .ne. 0 ) goto 9910
       if ( ibpn_ans .eq. 0 ) goto 3100
       iv_cnt = 0
 2100  continue
         call  ncbs800 ( ists )
         if ( ists .ne. 0 ) goto 9910
         if ( istop .ne. 0 ) goto 3100
         if ( iout_mod .eq. 1 ) then
           call  ncbs900 ( ists )
           if ( ists .ne. 0 ) goto 9910
```

```
          call  ncbsa00 ( ists )
          if ( ists .ne. 0 ) goto 9910
          if ( icrt_mod .eq. 1 ) then
            call  ncbsb00 ( ists )
            if ( ists .ne. 0 ) goto 9910
            call  ncbsc00 ( ists )
            if ( ists .ne. 0 ) goto 9910
          endif
        endif
        if ( icrt_mod .eq. 1 ) then
          call  ncbsd00 ( ists )
          if ( ists .ne. 0 ) goto 9910
        endif
        call  ncbse00 ( ists )
        if ( ists .ne. 0 ) goto 9910
        goto 2100
3100    continue
      if ( ibpn_ans .eq. 0 ) then
        call  ncbsi00 ( ists )
        if ( ists .ne. 0 ) goto 9910
      endif
      if ( iwsave_mod .eq. 1 ) then
        call  ncbsf00 ( ists )
        if ( ists .ne. 0 ) goto 9910
        call  ncbsh00 ( ists )
        if ( ists .ne. 0 ) goto 9910
      endif
      goto  9990
9910    continue
        異常時処理省略
        stop 'PROGRAM ABNORMAL END'
9990    continue
      ists = nclose ( ichn )
      stop 'PROGRAM NORMAL END'
     END
*  ********************************************************************
*  ***    MODULE NAME     : NCBS200  ノードセットアップ処理          ***
*  ********************************************************************
     subroutine       ncbs200 ( status )
      integer*4       status
$     include         (nccomh.h)
      parameter       ( iwork_size  = 2 ** 16 )
      common          / nwork / iload_work(iwork_size)
      integer*1       iload_work
      character*8     nodec           ,noden
      integer*1       icflag(0:max_node-1)
      data            nodec   / 'ncc300  '    /
      data            noden   / 'ncn300  '    /
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
      status = 0
      iuse_node = ifep_nodes + inet_nodes
      call  sshcubed ( iuse_node , ihcube )
      ichn = nopen ( ihcube )
      if ( ichn .lt. 0 ) goto 9940
      nodes_cnt   = 2 ** ihcube
```

```
      inode_tb(1) = 0
      inode_tb(2) = ifep_nodes
      inode_tb(3) = inode_tb(2) + iblocks
      inode_tb(4) = -1
      call  sshcubes ( ihcube , ipnode , ilnode , ists )
      do  1210  i = 0 , nodes_cnt-1
        icflag(i) = 0
1210  continue
      do  1220  i = inode_tb(1) , inode_tb(2)-1
        id_node = ipnode(i)
        icflag(id_node) = 1
1220  continue
      ists = nloadm ( ichn , nodec , nodes_cnt
     +              , icflag , iload_work , iwork_size )
      if ( ists .ne. 0 ) goto 9910
      do  1310  i = 0 , nodes_cnt-1
        icflag(i) = 0
1310  continue
      do  1320  i = inode_tb(2) , inode_tb(3)-1
        id_node = ipnode(i)
        icflag(id_node) = 1
1320  continue
      ists = nloadm ( ichn , noden , nodes_cnt
     +              , icflag , iload_work , iwork_size )
      if ( ists .ne. 0 ) goto 9920
      id_node = ipnode(inode_tb(1))
      msg_no = 1100
      ists = nwrite ( ichn , iparam , iparam_len1 , id_node , msg_no )
      if ( ists .ne. 0 ) goto 9930
      id_node = ipnode(inode_tb(2))
      msg_no  = 3100
      ists = nwrite ( ichn , iparam , iparam_len1 , id_node , msg_no )
      if ( ists .ne. 0 ) goto 9930
      goto 9990
9910  continue
        異常時処理省略
9920  continue
        異常時処理省略
9930  continue
        異常時処理省略
9940  continue
        異常時処理省略
9990  continue
      return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCBS600  入力画像送信処理                ***
*  ***********************************************************************
      subroutine       ncbs600 ( ip_cnt , status )
      integer*4        ip_cnt
      integer*4        status
$     include          (nccomh.h)
c     -----<< PROCESS >> ----------------------------------------
      status = 0
      isnd_len = ivx_len * ivy_len
```

```
       id_node = ipnode(inode_tb(1)+ip_cnt-1)
       msg_no  = 1200
       ists = nwrite ( ichn , iv_tbl , isnd_len , id_node , msg_no )
       if ( ists .ne. 0 ) goto 9910
       goto 9990
 9910  continue
         異常時処理省略
 9990  continue
       return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCBS700  制御信号送信処理                  ***
*  ***********************************************************************
      subroutine       ncbs700 ( status )
       integer*4       status
$      include         (nccomh.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status   = 0
       iv_cnt   = 0
       ib_cnt   = 0
       irun_mod = 0
       if ( ibpn_ans .eq. -1 ) then
         iout_mod = -1
       elseif ( ibpn_ans .gt. 0 ) then
         iout_mod = ibpn_ans
       else
         iout_mod = 0
       endif
       it_cnt   = log(2)
       istop    = 0
       isnd_len = 4 * ictl_len4
       id_node  = ipnode(inode_tb(1))
       msg_no   = 1300
       ists = nwrite ( ichn , ctl_tbl , isnd_len , id_node , msg_no )
       if ( ists .ne. 0 ) goto 9910
        goto 9990
 9910  continue
         異常時処理省略
 9990  continue
       return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCBS800  パラメータ受信処理                ***
*  ***********************************************************************
      subroutine       ncbs800 ( status )
       integer*4       status
$      include         (nccomh.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status  = 0
       isync   = 2100
       idx     = mod ( iv_cnt , iport_cnt ) + 1
       id_node = ipnode(inode_tb(1)+idx-1)
       msg_no  = 1001
       ists = nwrite ( ichn , isync , 4 , id_node , msg_no )
       if ( ists .ne. 0 ) goto 9920
```

```
      ircv_len = 4 * 32
      msg_no   = 2100
      id_node  = -1
      ists = nread ( ichn , lnkbuf1 , ircv_len , id_node , msg_no )
      if ( ists .le. 0 ) goto 9910
      istop    = lnkbuf4(01)
        受信情報編集処理省略
      sn_10a   = lnkbufr(09)
      if ( istop .eq. 0 ) then
        log(1)   = log(1) + 1
          履歴編集処理省略
      endif
      goto  9990
 9910 continue
        異常時処理省略
 9920 continue
        異常時処理省略
 9990 continue
      return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCBS900  出力画像受信処理              ***
*  ***********************************************************************
      subroutine       ncbs900 ( status )
      integer*4        status
$     include          (nccomh.h)
c     -----<< PROCESS >> ----------------------------------------
      status   = 0
      ircv_len = ivx_len * ivy_len
      msg_no   = 2200
      id_node  = -1
      ists = nread ( ichn , ov_tbl , ircv_len , id_node , msg_no )
      if ( ists .le. 0 ) goto 9910
      goto  9990
 9910 continue
        異常時処理省略
 9920 continue
        異常時処理省略
 9990 continue
      return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCBSA00  入力画像受信処理              ***
*  ***********************************************************************
      subroutine       ncbsa00 ( status )
      integer*4        status
$     include          (nccomh.h)
c     -----<< PROCESS >> ----------------------------------------
      status   = 0
      ircv_len = ivx_len * ivy_len
      msg_no   = 2300
      id_node  = -1
      ists = nread ( ichn , iv_tbl , ircv_len , id_node , msg_no )
      if ( ists .le. 0 ) goto 9910
      goto  9990
```

```
9910  continue
        異常時処理省略
9920  continue
        異常時処理省略
9990  continue
      return
      END

      subroutine        ncbs100 ( status )
        パラメータ読み込み処理省略

      subroutine        ncbs300 ( status )
        荷重セットアップ処理省略

      subroutine        ncbs400 ( icaller , status )
        ＣＲＴセットアップ処理省略

      subroutine        ncbs500 ( ip_cnt , status )
        入力画像読み込み処理省略

      subroutine        ncbsb00 ( status )
        入力画像表示処理省略

      subroutine        ncbsc00 ( status )
        出力画像表示処理省略

      subroutine        ncbsd00 ( status )
        Ｓ／Ｎ比表示処理省略

      subroutine        ncbse00 ( status )
        Ｓ／Ｎ比格納処理省略

      subroutine        ncbsf00 ( status )
        荷重受信処理省略

      subroutine        ncbsg00 ( status )
        履歴ファイル作成処理省略

      subroutine        ncbsh00 ( status )
        実行履歴格納処理省略

      subroutine        ncbsi00 ( status )
        パラメータ受信処理（２）省略
```

ｎｃｂ３００各種テーブル群（一部省略）

```
*  ***********************************************************************
*  ***  PROGRAM NAME     : NCCOMH.H                                      ***
*  ***  PROGRAM FUNCTION : NEURO CODEC COMMON AREA (HOST)                ***
*  ***********************************************************************
$      include          (nccoma.h)
       parameter        ( iparamh_len1 = 80+80*(ivfile_cn+7) )
       parameter        ( iparamh_len4 = iparamh_len1 / 4 )
       common           / iparamh / iparamh(iparamh_len4)
       integer*4        iparamh
         ファイルＩ／Ｏテーブル群省略
       parameter        ( lnk_len4       = max_3d        )
       parameter        ( lnk_len1       = 4*lnk_len4    )
       common           / lnkbuf / lnkbuf1(lnk_len1)
       integer*1        lnkbuf1
       integer*4        lnkbuf4(lnk_len4)
       real*4           lnkbufr(lnk_len4)
       equivalence      ( lnkbuf1        ,lnkbuf4        ,lnkbufr )
       common           / node_tbl / node_tbl(max_node+16)
       integer*2        ipnode(0:max_node-1)    ,ilnode(0:max_node-1)
       integer*4        inode_tb(4)
       equivalence      ( node_tbl(001)          ,ipnode          )
       equivalence      ( node_tbl(max_node/2+1),ilnode          )
       equivalence      ( node_tbl(max_node+1)  ,ichn            )
       equivalence      ( node_tbl(max_node+2)  ,id_node         )
       equivalence      ( node_tbl(max_node+3)  ,icube_odrer     )
       equivalence      ( node_tbl(max_node+4)  ,inodes          )
       equivalence      ( node_tbl(max_node+5)  ,inode_tb        )
       common           / crt_tbl / crt_tbl(256)
       integer*4        crt_tbl
         ＣＲＴ関係テーブル群省略
       common           / iv_tbl / iv_tbl(ivx_len,ivy_len)
       integer*1        iv_tbl
       character*1      iv_tbldf(ivx_len,ivy_len)
       equivalence      ( iv_tbl , iv_tbldf )
       common           / ov_tbl / ov_tbl(ivx_len,ivy_len)
       integer*1        ov_tbl
       character*1      ov_tbldf(ivx_len,ivy_len)
       equivalence      ( ov_tbl , ov_tbldf )
       parameter        ( isn_len        = 2 )
       common           / sn_tbl / sn_tbl(16)
       integer*4        sn_tbl
       real*4           snlog(isn_len)
       character*4      snlogdf(isn_len)
       real*4           sn_10            ,sn_10a
       equivalence      ( sn_tbl         ,snlog          ,snlogdf        )
       equivalence      ( sn_tbl(15)     ,sn_10a         )
       equivalence      ( sn_tbl(16)     ,sn_10          )
       parameter        ( ifw_blk        = 256 )
       common           / w_tbl / w_tbl(max_3d)
       real*4           w_tbl
       character*4      w_tbldf(max_3d)
       equivalence      ( w_tbl , w_tbldf )
       common           / wth_tbl / wth_tbl(max_2d)
       real*4           wth_tbl
```

```
      character*4     wth_tbldf(max_2d)
      equivalence     ( wth_tbl , wth_tbldf )
      common          / log_tbl / log_tbl(32)
      integer*4       log_tbl
      integer*4       log(32)
      equivalence     ( log_tbl       ,log    )
*  ***************************************************************
*  ***  PROGRAM NAME     : NCCOMA.H                              ***
*  ***  PROGRAM FUNCTION : NEURO CODEC COMMON AREA (MASTER)      ***
*  ***************************************************************
      parameter       ( max_3d        =  16*1024      )
      parameter       ( max_2d        =   4*1024      )
      parameter       ( max_node      =  512          )
      parameter       ( max_layer     =  16           )
      parameter       ( ivx_len       =  128          )
      parameter       ( ivy_len       =  128          )
      parameter       ( ivfile_cn     =   64          )
      parameter       ( isn_buf_max   = 4000          )
      parameter       ( iparam_len4   = 1024          )
      parameter       ( iparam_len1   = 4*iparam_len4 )
      common          / iparam / iparam
      integer*4       iparam(iparam_len4)
      integer*4       nd_x(max_layer) ,nd_y(max_layer)
      integer*4       nd(max_layer)   ,nds(max_layer)
      real*4          alpha(max_layer)
      integer*4       ith_mod(max_layer),ilinear(max_layer)
      equivalence     ( iparam(001)   ,iport_x        )
      equivalence     ( iparam(002)   ,iport_y        )
      equivalence     ( iparam(003)   ,iblock_x       )
      equivalence     ( iparam(004)   ,iblock_y       )
      equivalence     ( iparam(005)   ,iblockv_x      )
      equivalence     ( iparam(006)   ,iblockv_y      )
      equivalence     ( iparam(007)   ,iovl_mod       )
      equivalence     ( iparam(008)   ,ln             )
      equivalence     ( iparam(021)   ,ifep           )
      equivalence     ( iparam(022)   ,cgain          )
      equivalence     ( iparam(023)   ,cbase          )
      equivalence     ( iparam(025)   ,iq_mod         )
      equivalence     ( iparam(026)   ,iq_step        )
      equivalence     ( iparam(027)   ,q_offset       )
      equivalence     ( iparam(033)   ,iwset_mod      )
      equivalence     ( iparam(034)   ,iwsave_mod     )
      equivalence     ( iparam(035)   ,inz_mod        )
      equivalence     ( iparam(036)   ,rand_u         )
      equivalence     ( iparam(037)   ,rand_l         )
      equivalence     ( iparam(038)   ,w_g0           )
      equivalence     ( iparam(041)   ,epcil          )
      equivalence     ( iparam(042)   ,iunub          )
      equivalence     ( iparam(069)   ,istop_cnt      )
      equivalence     ( iparam(070)   ,iport_cnt      )
      equivalence     ( iparam(071)   ,ibpn_cnt       )
      equivalence     ( iparam(072)   ,ibpn_ans       )
      equivalence     ( iparam(073)   ,iwrk_mod       )
      equivalence     ( iparam(074)   ,icrt_mod       )
      equivalence     ( iparam(081)   ,ifep_nodes     )
```

```
equivalence     ( iparam(082)    ,inet_nodes      )
equivalence     ( iparam(083)    ,iblocks         )
equivalence     ( iparam(084)    ,i2d_len         )
equivalence     ( iparam(085)    ,i3d_len         )
equivalence     ( iparam(129)                     ,nd      )
equivalence     ( iparam(129+  max_layer)         ,nds     )
equivalence     ( iparam(129+2*max_layer)         ,nd_x    )
equivalence     ( iparam(129+3*max_layer)         ,nd_y    )
equivalence     ( iparam(129+4*max_layer)         ,ilinear)
equivalence     ( iparam(129+5*max_layer)         ,alpha   )
equivalence     ( iparam(129+6*max_layer)         ,ith_mod)
equivalence     ( iparam(iparam_len4-2) ,idbg_nd          )
equivalence     ( iparam(iparam_len4-1) ,idbg_msg         )
equivalence     ( iparam(iparam_len4)   ,ipara_no         )
parameter       ( ictl_len4     = 8     )
common          / ctl_tbl / ctl_tbl(ictl_len4)
integer*4       ctl_tbl
equivalence     ( ctl_tbl(01)    ,iv_cnt          )
equivalence     ( ctl_tbl(02)    ,ib_cnt          )
equivalence     ( ctl_tbl(03)    ,irun_mod        )
equivalence     ( ctl_tbl(04)    ,iout_mod        )
equivalence     ( ctl_tbl(05)    ,it_cnt          )
equivalence     ( ctl_tbl(06)    ,istop           )
```

```
*   ***********************************************************************
*   ***  SYSTEM           : ATR  Parallel Computing System              ***
*   ***  PROJECT          : Auditory & Visual Perception Res. Labs.     ***
*   ***  MODEL NO.        : NCUBE/10    (VERTEX)                        ***
*   ***  LANGUAGE         : FORTRAN-77                                  ***
*   ***  PROGRAM NAME     : NCC300    画像変換処理（ＮＯＤＥ．１）     ***
*   ***  PROGRAM FUNCTION : NEURO CODEC (NODE.1)  (CONVERTER)           ***
*   ***  DOCUMENT NO.     : ATR-NC-A-3-00-D                             ***
*   ***  MAKE OUT DATE    : 05-SEP-88                                   ***
*   ***  VERSION NO.      : 03.00                                       ***
*   ***********************************************************************
*   ***    #    #####  ####      Advanced Telecommunication           ***
*   ***   # #     #    #   #      Reseach Institute International     ***
*   ***  #   #    #    ####      -------------------------------- ***
*   ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*   ***  #   #    #    #   #      Research Laboratories               ***
*   ***********************************************************************
*
      program          ncc300
$      include          (nccomc.h)
c     --------<< PROCESS >>-----------------------------------------
       call  nccs100 ( ists )
       if ( ists .ne. 0 ) goto 9910
       call  nccs200 ( ists )
       if ( ists .ne. 0 ) goto 9910
       call  nccs300 ( ists )
       if ( ists .ne. 0 ) goto 9910
 1100  continue
         call  nccs400 ( ists )
         if ( ists .ne. 0 ) goto 9910
         call  nccs500 ( ists )
         if ( ists .ne. 0 ) goto 9910
         if ( istop .ne. 0 ) goto 2100
         if ( ibpn_ans .ne. 0 ) then
           call  nccs600 ( ists )
           if ( ists .ne. 0 ) goto 9910
           call  nccs700 ( ists )
           if ( ists .ne. 0 ) goto 9910
           call  nccs800 ( ists )
           if ( ists .ne. 0 ) goto 9910
           if ( irun_mod .eq. 0 ) then
             if ( iout_mod .gt. 0 ) then
               call  nccs900 ( ists )
               if ( ists .ne. 0 ) goto 9910
               call  nccsa00 ( ists )
               if ( ists .ne. 0 ) goto 9910
             endif
           else
             call  nccs900 ( ists )
             if ( ists .ne. 0 ) goto 9910
           endif
         else
           call  nccsb00 ( ists )
           if ( ists .ne. 0 ) goto 9910
         endif
```

```
        goto 1100
 2100   continue
        if ( ibpn_ans .ne. 0 ) then
          call  nccs800 ( ists )
          if ( ists .ne. 0 ) goto 9910
        else
          call  nccsc00 ( ists )
          if ( ists .ne. 0 ) goto 9910
        endif
        goto  9990
 9910   continue
          異常時処理省略
 9990   continue
        stop
       END
*  ********************************************************************
*  ***    MODULE NAME     : NCCS100  自己セットアップ処理           ***
*  ********************************************************************
       subroutine       nccs100 ( status )
        integer*4       status
$       include         (nccomc.h)
c      -----〈〈 PROCESS 〉〉 ----------------------------------------
        status = 0
        call  whoami ( id_node , id_process , id_host , icube_odrer )
        nodes_cnt = 2 ** icube_odrer
        call  sshcubes ( icube_odrer , ipnode , ilnode , ists )
        msg_no  = 1100
        ir_node = -1
        ists = nread  ( iparam , iparam_len1 , ir_node , msg_no , icflag )
        if ( ists .le. 0 ) goto 9910
        inode_tb(1) = 0
        inode_tb(2) = iport_cnt
        inode_tb(3) = inode_tb(2) + iblocks
        inode_tb(4) = -1
        il_node = ilnode(id_node)
        ig_node = il_node - inode_tb(1)
        ihd_no  = ig_node + 1
        if ( ihd_no .lt. iport_cnt ) then
          msg_no   = 1100
          is_node  = ipnode(il_node+1)
          ists = nwrite ( iparam , iparam_len1 , is_node , msg_no , icflag )
          if ( ists .ne. 0 ) goto 9920
        endif
        goto  9990
 9910   continue
          異常時処理省略
 9920   continue
          異常時処理省略
 9990   continue
        return
       END
*  ********************************************************************
*  ***    MODULE NAME     : NCCS200  入力画像受信処理               ***
*  ********************************************************************
       subroutine       nccs200 ( status )
```

```
      integer*4       status
$     include         (nccomc.h)
c     -----<< PROCESS >> ----------------------------------------
      status   = 0
      msg_no   = 1200
      ircv_len = ivx_len * ivy_len
      ir_node  = -1
      ists = nread  ( iv_tbl , ircv_len , ir_node , msg_no , icflag )
      if ( ists .le. 0 ) goto 9910
      goto 9990
 9910 continue
        異常時処理省略
 9990 continue
      return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCCS400  制御信号受信処理              ***
*  ***********************************************************************
      subroutine      nccs400 ( status )
      integer*4       status
$     include         (nccomc.h)
c     -----<< PROCESS >> ----------------------------------------
      status   = 0
      msg_no   = 1300
      ircv_len = 4 * ictl_len4
      ir_node  = -1
      ists = nread  ( ctl_tbl , ircv_len , ir_node , msg_no , icflag )
      if ( ists .le. 0 ) goto 9910
        受信情報編集処理省略
      goto 9990
 9910 continue
        異常時処理省略
 9990 continue
      return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCCS500  入力信号送信処理              ***
*  ***********************************************************************
      subroutine      nccs500 ( status )
      integer*4       status
$     include         (nccomc.h)
c     -----<< PROCESS >> ----------------------------------------
      status    = 0
      isnd_len  = 4 * ( 16 + nd(1) )
      isnd2_len = 4 * ictl_len
      lnkbuf4(01) = ib_cnt
        送信HEAD情報編集処理省略
      do  1310  ibno = iblocks , 1 , -1
        call  nccs510 ( ibno , ix , iy )
        ib  = 0
        ixe = ix + iblockv_x - 1
        iye = iy + iblockv_y - 1
        do  1220  jy = iy , iye
          do  1210  jx = ix , ixe
            ib = ib + 1
```

```
            if (     ( jy .ge. 1 .and. jy .le. iport_y )
     +          .and. ( jx .ge. 1 .and. jx .le. iport_x ) ) then
              lnkbufr(ib+16) = ci_tbl(jx,jy)
            else
              lnkbufr(ib+16) = 0.0
            endif
 1210      continue
 1220    continue
         is_node = ipnode(ibno+inode_tb(2)-1)
         msg_no  = 5100
         ists = nwrite ( lnkbuf1 , isnd_len , is_node , msg_no , icflag )
         if ( ists .ne. 0 ) goto 9910
 1310  continue
       goto 9990
 9910  continue
         異常時処理省略
 9990  continue
       return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCCS600  出力信号受信処理                  ***
*  ***********************************************************************
      subroutine       nccs600 ( status )
       integer*4       status
$      include         (nccomc.h)
c     -----<< PROCESS >> ----------------------------------------
       status  = 0
       isync   = 6100
       msg_no  = 5001
       is_node = ipnode(inode_tb(2))
       ists = nwrite ( isync , 4 , is_node , msg_no , icflag )
       if ( ists .ne. 0 ) goto 9910
       ircv_len = 4 * ( 16 + nd(ln) )
       sn_i2  = 0.0
       sn_io2 = 0.0
       msg_no = 6100
       do  1320  i = 1 , iblocks
         ir_node = -1
         ists = nread ( lnkbuf1 , ircv_len , ir_node , msg_no , icflag )
         if ( ists .le. 0 ) goto 9920
         sn_blk  = lnkbufr(9)
         sn_i2   = sn_i2  + lnkbufr(10)
         sn_io2  = sn_io2 + lnkbufr(11)
         ibno = ilnode(ir_node) - inode_tb(2) + 1
         call  nccs610 ( ibno , ists )
         if ( ists .ne. 0 ) goto 9930
 1320  continue
c      --------------------------------<< RETURN >>--------------------
       goto  9990
 9910  continue
         異常時処理省略
 9920  continue
         異常時処理省略
 9930  continue
         異常時処理省略
```

```
 9990  continue
       return
      END
*  ********************************************************************
*  ***    MODULE NAME      : NCCS800  制御信号送信処理                ***
*  ********************************************************************
      subroutine       nccs800 ( status )
       integer*4       status
$      include         (nccomc.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status = 0
       do  1010  i = 1 , 32
         lnkbuf4(i) = 0
 1010  continue
       if ( istop .eq. 0 ) then
         is       = mod ( ihd_no , iport_cnt ) + 1
         is_node  = ipnode(is-1+inode_tb(1))
         msg_no   = 1300
         isnd_len = 4 * ictl_len4
         ists = nwrite ( ctl_tbl , isnd_len , is_node , msg_no , icflag )
         if ( ists .ne. 0 ) goto 9910
       endif
       lnkbuf4(01) = istop
         送信情報編集処理省略
       lnkbufr(09) = sn_10a
       msg_no  = 1001
       ir_node = -1
       ists = nread ( isync , 4 , ir_node , msg_no , icflag )
       if ( ists .le. 0 ) goto 9920
       isnd_len = 4 * 32
       msg_no   = 2100
       ists = nwrite ( lnkbuf1 , isnd_len , id_host , msg_no , icflag )
       if ( ists .ne. 0 ) goto 9930
       goto 9990
 9910  continue
         異常時処理省略
 9920  continue
         異常時処理省略
 9930  continue
         異常時処理省略
 9990  continue
       return
      END
*  ********************************************************************
*  ***    MODULE NAME      : NCCS900  出力画像送信処理                ***
*  ********************************************************************
      subroutine       nccs900 ( status )
       integer*4       status
$      include         (nccomc.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status   = 0
       isnd_len = ivx_len * ivy_len
       msg_no   = 2200
       ists = nwrite ( ov_tbl , isnd_len , id_host , msg_no , icflag )
       if ( ists .ne. 0 ) goto 9910
```

```
      goto 9990
 9910 continue
        異常時処理省略
 9990 continue
      return
      END
*  ********************************************************************
*  ***    MODULE NAME     : NCCSA00  入力画像送信処理              ***
*  ********************************************************************
      subroutine       nccsa00 ( status )
       integer*4       status
$      include         (nccomc.h)
c     -----<< PROCESS >> ----------------------------------------
       status   = 0
       isnd_len = ivx_len * ivy_len
       msg_no   = 2300
       ists = nwrite ( iv_tbl , isnd_len , id_host , msg_no , icflag )
       if ( ists .ne. 0 ) goto 9910
       goto 9990
 9910 continue
        異常時処理省略
 9990 continue
      return
      END
*  ********************************************************************
*  ***    MODULE NAME     : NCCSB00  制御信号送信処理              ***
*  ********************************************************************
      subroutine       nccsb00 ( status )
       integer*4       status
$      include         (nccomc.h)
 c     -----<< PROCESS >> ----------------------------------------
       status  = 0
       msg_no  = 6001
       ir_node = -1
       ists = nread ( isync , 4 , ir_node , msg_no , icflag )
       if ( ists .lt. 0 ) goto 9910
       if ( istop .eq. 0 ) then
         is       = mod ( ihd_no , iport_cnt ) + 1
         is_node  = ipnode(is-1+inode_tb(1))
         msg_no   = 1300
         isnd_len = 4 * ictl_len4
         ists = nwrite ( ctl_tbl , isnd_len , is_node , msg_no , icflag )
         if ( ists .ne. 0 ) goto 9920
       endif
       goto 9990
 9910 continue
        異常時処理省略
 9920 continue
        異常時処理省略
 9990 continue
      return
      END
*  ********************************************************************
*  ***    MODULE NAME     : NCCSC00  パラメータ送信処理            ***
*  ********************************************************************
```

```
      subroutine        nccsc00 ( status )
       integer*4        status
$      include          (nccomc.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status = 0
       do  1010  i = 1 , 32
         lnkbuf4(i) = 0
 1010  continue
       lnkbuf4(01) = istop
         送信情報編集処理省略
       lnkbufr(08) = sn_10
       isnd_len = 4 * 32
       msg_no   = 2110
       ists = nwrite ( lnkbuf1 , isnd_len , id_host , msg_no , icflag )
       if ( ists .ne. 0 ) goto 9910
       goto 9990
 9910  continue
         異常時処理省略
 9990  continue
       return
      END

      subroutine        nccs300 ( status )
          入力画像変換処理省略

      subroutine        nccs510  ( ibno , ix , iy )
          入力ブロックインデックス計算処理省略

      subroutine        nccs610 ( ibno , status )
          出力信号編集処理省略

      subroutine        nccs700 ( status )
          出力画像変換処理省略

      subroutine        id_cnv1 ( id , id_x , id_y , mtlx_x , mtlx_y )
          2次元→1次元変換処理省略

      subroutine        id_cnv2 ( id , id_x , id_y , mtlx_x , mtlx_y )
          1次元→2次元変換処理省略

      integer*4 function    idx2d   ( i , n )
          ニューロエリアインデックス計算処理省略

      integer*4 function    idx3d   ( i , j , n )
          荷重エリアインデックス計算処理省略
```

ｎｃｃ３００各種テーブル群（一部省略）

```
*  ****************************************************************
*  ***  PROGRAM NAME     : NCCOMC.H                                ***
*  ***  PROGRAM FUNCTION : NEURO CODEC COMMON AREA (NODE_EDIT)     ***
*  ****************************************************************
$      include          (nccoma.h)
       parameter        ( iparamn_len4 = 8 )
       common           / iparamn       / iparamn(iparamn_len4)
       integer*4        iparamn
       integer*4        irun_mode
       equivalence      ( iparamn(01)   ,irun_mode      )
       equivalence      ( iparamn(02)   ,ihd_no         )
       equivalence      ( iparamn(iparamn_len4) ,idn    )
       parameter        ( lnk_len1      = ivx_len*ivy_len + 64)
       parameter        ( lnk_len4      = lnk_len1 / 4  )
       common           / lnkbuf / lnkbuf1(lnk_len1)
       integer*1        lnkbuf1
       integer*4        lnkbuf4(lnk_len4)
       real*4           lnkbufr(lnk_len4)
       equivalence      ( lnkbuf1       ,lnkbuf4        ,lnkbufr )
       common           / node_tbl      / node_tbl(max_node+16)
       integer*2        ipnode(0:max_node-1)    ,ilnode(0:max_node-1)
       integer*4        inode_tb(4)
       equivalence      ( node_tbl(001)         ,ipnode         )
       equivalence      ( node_tbl(max_node/2+1),ilnode         )
       equivalence      ( node_tbl(max_node+ 1) ,id_host        )
       equivalence      ( node_tbl(max_node+ 2) ,id_node        )
       equivalence      ( node_tbl(max_node+ 3) ,inodes         )
       equivalence      ( node_tbl(max_node+ 4) ,id_process     )
       equivalence      ( node_tbl(max_node+ 5) ,icube_odrer    )
       equivalence      ( node_tbl(max_node+ 6) ,inode_tb       )
       equivalence      ( node_tbl(max_node+10) ,il_node        )
       equivalence      ( node_tbl(max_node+11) ,ig_node        )
       common           / iv_tbl / iv_tbl(ivx_len,ivy_len)
       integer*1        iv_tbl
       common           / ov_tbl / ov_tbl(ivx_len,ivy_len)
       integer*1        ov_tbl
       common           / ci_tbl / ci_tbl(ivx_len,ivy_len)
       common           / co_tbl / co_tbl(ivx_len,ivy_len)
       parameter        ( iout_len      = 8 )
       common           / out_tbl / out_tbl(iout_len)
       equivalence      ( out_tbl(1)    ,sn_blk         )
       equivalence      ( out_tbl(2)    ,sn_i2          )
       equivalence      ( out_tbl(3)    ,sn_io2         )
       equivalence      ( out_tbl(4)    ,sn_10          )
       equivalence      ( out_tbl(5)    ,sn_10a         )
```

```
*  ***********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System          ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*  ***  MODEL NO.        : NCUBE/10    (VERTEX)                    ***
*  ***  LANGUAGE         : FORTRAN-77                              ***
*  ***  PROGRAM NAME     : NCN300    ニューラル・ネットワーク処理  ***
*  ***  PROGRAM FUNCTION : NEURO CODEC (NODE.2)  (NEURAL NETWORK)  ***
*  ***  DOCUMENT NO.     : ATR-NC-A-3-00-D                         ***
*  ***  MAKE OUT DATE    : 05-SEP-88                               ***
*  ***  VERSION NO.      : 03.00                                   ***
*  ***********************************************************************
*  ***    #    #####  ####      Advanced Telecommunication        ***
*  ***   # #     #    #   #      Reseach Institute International   ***
*  ***  #   #    #    ####      ---------------------------------- ***
*  ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*  ***  #   #    #    #   #      Research Laboratories             ***
*  ***********************************************************************
*
      program          ncn300
$      include          (nccomn.h)
c     --------<< PROCESS >>-----------------------------------------
       call  ncns100 ( ists )
       if ( ists .ne. 0 ) goto 9910
       if ( iwset_mod .eq. 0 ) then
         call  ncns200 ( ists )
         if ( ists .ne. 0 ) goto 9910
       else
         call  ncns300 ( ists )
         if ( ists .ne. 0 ) goto 9910
       endif
 1100  continue
         call  ncns400 ( ists )
         if ( ists  .ne. 0 ) goto 9910
         if ( istop .ne. 0 ) goto 2100
         if ( irun_mod .eq. 0 ) then
           do  1200  i = 1 , ibpn_cnt
             call  ncns500 ( ists )
             if ( ists .ne. 0 ) goto 9910
             call  ncns600 ( ists )
             if ( ists .ne. 0 ) goto 9910
 1200      continue
           call  ncns700 ( ists )
           if ( ists .ne. 0 ) goto 9910
           if ( ibpn_ans .ne. 0 ) then
             call  ncns800 ( ists )
             if ( ists .ne. 0 ) goto 9910
           else
             call  ncnsb00 ( ists )
             if ( ists .ne. 0 ) goto 9910
           endif
         elseif ( irun_mod .eq. 1 ) then
           call  ncns500 ( ists )
           if ( ists .ne. 0 ) goto 9910
           call  ncns700 ( ists )
           if ( ists .ne. 0 ) goto 9910
```

```
          call  ncns800 ( ists )
          if ( ists .ne. 0 ) goto 9910
          call  ncnsa00 ( ists )
          if ( ists .ne. 0 ) goto 9910
        endif
        goto 1100
 2100  continue
       if ( iwsave_mod .eq. 1 ) then
         call  ncns900 ( ists )
         if ( ists .ne. 0 ) goto 9910
         if ( ibpn_ans .eq. 0 ) then
           call  ncnsc00 ( ists )
           if ( ists .ne. 0 ) goto 9910
         endif
       endif
       goto  9990
 9910  continue
         異常時処理省略
 9990  continue
       stop
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCNS100  自己セットアップ処理          ***
*  ***********************************************************************
      subroutine        ncns100 ( status )
       integer*4       status
$      include         (nccomn.h)
c     -----<< PROCESS >> ----------------------------------------
       status = 0
       call  whoami ( id_node , id_process , id_host , icube_odrer )
       nodes_cnt = 2 ** icube_odrer
       call  sshcubes ( icube_odrer , ipnode , ilnode , ists )
       msg_no  = 3100
       ir_node = -1
       ists = nread  ( iparam , iparam_len1 , ir_node , msg_no , icflag )
       if ( ists .le. 0 ) goto 9910
       inode_tb(1) = 0
       inode_tb(2) = iport_cnt
       inode_tb(3) = inode_tb(2) + iblocks
       inode_tb(4) = -1
       il_node = ilnode(id_node)
       ig_node = il_node - inode_tb(2)
       ihd_no  = ig_node + 1
       if ( ihd_no .lt. iblocks ) then
         msg_no   = 3100
         is_node  = ipnode(il_node+1)
         ists = nwrite ( iparam , iparam_len1 , is_node , msg_no , icflag )
         if ( ists .ne. 0 ) goto 9920
       endif
       iu      = 0
       isn_cnt = 0
       goto  9990
 9910  continue
         異常時処理省略
 9920  continue
```

```
          異常時処理省略
 9990  continue
       return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCNS400  入力信号受信処理                  ***
*  ***********************************************************************
      subroutine        ncns400 ( status )
       integer*4       status
$      include         (nccomn.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status = 0
       msg_no   = 5100
       ircv_len = 4 * ( 16 + nd(1) )
       ir_node  = -1
       ists = nread  ( lnkbuf1 , ircv_len , ir_node , msg_no , icflag )
       if ( ists .le. 0 ) goto 9910
       ib_cnt    = lnkbuf4(01)
         受信情報編集処理省略
       iret_node = lnkbuf4(09)
       do  3100  i = 1 , nd(1)
         t_tbl(i) = lnkbufr(i+16)
 3100  continue
       goto 9990
 9910  continue
 9990  continue
       return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCNS500  ニューラル・ネットＦＦ処理     ***
*  ***********************************************************************
      subroutine       ncns500 ( status )
       integer*4       status
$      include         (nccomn.h)
       integer*4       ists
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
       status = 0
       if ( iq_mod .eq. 1 ) then
         iqs2 = 2 ** iq_step
         rqs2 = float ( iqs2 )
         aqs2 = 1.0 / rqs2
       endif
       do  1100  i = 1 , nd(1)
         s_tbl(i) = t_tbl(i)
 1100  continue
       do  2300  n = 2 , ln-1
         j3 = idx3d ( 1 , 1 , n ) - 1
         j2 = idx2d ( 1 , n ) - 1
         do  2200  j = 1 , nd(n)
           rnet = 0.0
           i2 = idx2d ( 1 , n-1 ) - 1
           do  2100  i = 1 , nd(n-1)
             i2 = i2 + 1
             j3 = j3 + 1
             rnet = rnet + w_tbl(j3) * s_tbl(i2)
```

```
2100      continue
          j2 = j2 + 1
          rnet = rnet + wth_tbl(j2) * sth_tbl(n)
          if ( ilinear(n) .eq. 0 ) then
            if ( rnet .lt. -87.0 ) rnet = -87.0
            if ( rnet .gt.  87.0 ) rnet =  87.0
            sdata = 1.0 / ( 1.0 + exp ( -rnet ) )
          else
            sdata = rnet
          endif
          if ( iq_mod .eq. 0 ) then
            s_tbl(j2) = sdata
          else
            isdata = rqs2 * sdata
            s_tbl(j2) = aqs2 * float ( isdata ) + q_offset
          endif
2200    continue
2300  continue
      j3 = idx3d ( 1 , 1 , ln ) - 1
      j2 = idx2d ( 1 , ln ) - 1
      do  3200  j = 1 , nd(ln)
        rnet = 0.0
        i2 = idx2d ( 1 , ln-1 ) - 1
        do  3100  i = 1 , nd(ln-1)
          i2 = i2 + 1
          j3 = j3 + 1
          rnet = rnet + w_tbl(j3) * s_tbl(i2)
3100    continue
        j2 = j2 + 1
        rnet = rnet + wth_tbl(j2) * sth_tbl(ln)
        if ( ilinear(ln) .eq. 0 ) then
          if ( rnet .lt. -87.0 ) rnet = -87.0
          if ( rnet .gt.  87.0 ) rnet =  87.0
          sdata = 1.0 / ( 1.0 + exp ( -rnet ) )
        else
          sdata = rnet
        endif
        s_tbl(j2) = sdata
        o_tbl(j)  = sdata
3200  continue
      goto  9990
9910  continue
        異常時処理省略
9990  continue
      return
     END
*  ***********************************************************************
*  ***    MODULE NAME     : NCNS600  ニューラル・ネットＦＢ処理     ***
*  ***********************************************************************
     subroutine       ncns600 ( status )
      integer*4       status
$     include         (nccomn.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
      status = 0
      if ( ilinear(ln) .eq. 0 ) then
```

```
        i2 = idx2d ( 1 , ln ) - 1
        do  1100  l = 1 , nd(ln)
          dlts = ( t_tbl(l) - o_tbl(l) )
     +           * ( o_tbl(l) * ( 1.0 - o_tbl(l) ) )
          i2 = i2 + 1
          d_tbl(i2) = dlts
1100    continue
      else
        i2 = idx2d ( 1 , ln ) - 1
        do  1200  l = 1 , nd(ln)
          dlts = t_tbl(l) - o_tbl(l)
          i2 = i2 + 1
          d_tbl(i2) = dlts
1200    continue
      endif
      do  2300  n = ln-1 , 2 , -1
        j2 = idx2d ( 1 , n ) - 1
        do  2200  j = 1 , nd(n)
          a = 0.0
          j3 = idx3d ( j , 1 , n+1 ) - nd(n)
          k2 = idx2d ( 1 , n+1 ) - 1
          do  2100  k = 1 , nd(n+1)
            k2 = k2 + 1
            j3 = j3 + nd(n)
            a = a + d_tbl(k2) * w_tbl(j3)
2100      continue
          j2 = j2 + 1
          if ( ilinear(n) .eq. 0 ) then
            sdata = s_tbl(j2)
            dlts = ( sdata * ( 1.0 - sdata ) ) * a
          else
            dlts =  a
          endif
          d_tbl(j2) = dlts
2200    continue
2300  continue
      do  3300  n = 2 , ln
        i2 = idx2d ( 1 , n ) - 1
        do  3200  i = 1 , nd(n)
          i2 = i2 + 1
          ddata = d_tbl(i2)
          j3 = idx3d ( 1 , i , n ) - 1
          i21 = idx2d ( 1 , n-1 ) - 1
          do  3100  j = 1 , nd(n-1)
            j3 = j3 + 1
            i21 = i21 + 1
            dwdata = alpha(n) * dw_tbl(j3)
     +             + ( 1.0 - alpha(n) ) * ddata * s_tbl(i21)
            dw_tbl(j3) = dwdata
3100      continue
          dwthdata = alpha(n) * dwth_tbl(i2)
     +           + ( 1.0 - alpha(n) ) * ddata * sth_tbl(n)
          dwth_tbl(i2) = dwthdata
3200    continue
3300  continue
```

```
      iu = iu + 1
      if ( iu .ge. iunub ) then
        iu = 0
        do  4300  n = 2 , ln
          j3 = idx3d ( 1 , 1 , n ) - 1
          i2 = idx2d ( 1 , n ) - 1
          do  4200  i = 1 , nd(n)
            do  4100  j = 1 , nd(n-1)
              j3 = j3 + 1
              wdata = w_tbl(j3) + epcil * dw_tbl(j3)
              w_tbl(j3) = wdata
              if ( iunub .gt. 1 )  dw_tbl(j3) = 0.0
4100        continue
            i2 = i2 + 1
            wthdata = wth_tbl(i2) + epcil * dwth_tbl(i2)
            wth_tbl(i2) = wthdata
            if ( iunub .gt. 1 )  dwth_tbl(i2) = 0.0
4200      continue
4300    continue
      endif
9990  continue
      return
      END
*  ***********************************************************************
*  ***    MODULE NAME     : NCNS800  出力信号送信処理                  ***
*  ***********************************************************************
      subroutine       ncns800 ( status )
      integer*4        status
$     include          (nccomn.h)
c     -----〈〈 PROCESS 〉〉 ----------------------------------------
      status = 0
      lnkbuf4( 1) = 0
        送信情報編集処理省略
      lnkbufr(11) = sn_io2
      if ( irun_mod .eq. 1
     +    .or. ( irun_mod .eq. 0 .and. iout_mod .gt. 0 ) ) then
        isnd_len = 4 * ( 16 + nd(ln) )
        do  1200  i = 1 , nd(ln)
          lnkbufr(i+16) = o_tbl(i)
1200    continue
      else
        isnd_len = 4 * 16
      endif
      msg_no  = 5001
      ir_node = -1
      ists = nread  ( isync , 4 , ir_node , msg_no , icflag )
      if ( ists .le. 0 ) goto 9910
      msg_no  = 6100
      is_node = iret_node
      ists = nwrite ( lnkbuf1 , isnd_len , is_node , msg_no , icflag )
      if ( ists .ne. 0 ) goto 9920
      if ( ihd_no .lt. iblocks ) then
        msg_no   = 5001
        is_node  = ipnode(il_node+1)
        ists = nwrite ( isync , 4 , is_node , msg_no , icflag )
```

```
        if ( ists .ne. 0 ) goto 9930
      endif
      goto 9990
9910  continue
        異常時処理省略
9920  continue
        異常時処理省略
9930  continue
        異常時処理省略
9990  continue
      return
      END

      subroutine          ncns200 ( status )
        荷重初期設定処理省略

      subroutine          ncns300 ( status )
        荷重受信処理省略

      subroutine          ncns700 ( status )
        S／N比計算処理省略

      subroutine          ncns900 ( status )
        荷重送信処理省略

      subroutine          ncnsa00 ( status )
        隠れ層信号送信処理省略

      subroutine          ncnsb00 ( status )
        S／N比送信処理省略

      subroutine          ncnsc00 ( status )
        S／N比送信処理（２）省略
```

ｎｃｎ３００各種テーブル群（一部省略）

```
*  ****************************************************************
*  ***  PROGRAM NAME     : NCCOMN.H                                ***
*  ***  PROGRAM FUNCTION : NEURO CODEC COMMON AREA (NODE_NNW)      ***
*  ****************************************************************
$      include         (nccoma.h)
       parameter       ( iparamn_len4 = 8 )
       common          / iparamn       / iparamn(iparamn_len4)
       equivalence     ( iparamn(01)   ,ihd_no         )
       equivalence     ( iparamn(02)   ,iret_node      )
       equivalence     ( iparamn(03)   ,iu             )
       equivalence     ( iparamn(04)   ,iv             )
       equivalence     ( iparamn(iparamn_len4) ,idn    )
       parameter       ( lnk_len4      = max_3d        )
       parameter       ( lnk_len1      = 4*lnk_len4    )
       common          / lnkbuf / lnkbuf1(lnk_len1)
       integer*1       lnkbuf1
       integer*4       lnkbuf4(lnk_len4)
       real*4          lnkbufr(lnk_len4)
       equivalence     ( lnkbuf1       ,lnkbuf4        ,lnkbufr )
       common          / node_tbl      / node_tbl(max_node+16)
       integer*2       ipnode(0:max_node-1)    ,ilnode(0:max_node-1)
       integer*4       inode_tb(4)
       equivalence     ( node_tbl(001)         ,ipnode         )
       equivalence     ( node_tbl(max_node/2+1),ilnode         )
       equivalence     ( node_tbl(max_node+ 1) ,id_host        )
       equivalence     ( node_tbl(max_node+ 2) ,id_node        )
       equivalence     ( node_tbl(max_node+ 3) ,inodes         )
       equivalence     ( node_tbl(max_node+ 4) ,id_process     )
       equivalence     ( node_tbl(max_node+ 5) ,icube_odrer    )
       equivalence     ( node_tbl(max_node+ 6) ,inode_tb       )
       equivalence     ( node_tbl(max_node+10) ,il_node        )
       equivalence     ( node_tbl(max_node+11) ,ig_node        )
       parameter       ( iout_len      = 8 )
       common          / out_tbl / out_tbl(iout_len)
       equivalence     ( out_tbl(1)    ,sn_blk         )
       equivalence     ( out_tbl(2)    ,sn_i2          )
       equivalence     ( out_tbl(3)    ,sn_io2         )
       common          / t_tbl         / t_tbl(max_2d)
       common          / s_tbl         / s_tbl(max_2d)
       common          / sth_tbl       / sth_tbl(max_layer)
       common          / o_tbl         / o_tbl(max_2d)
       common          / w_tbl         / w_tbl(max_3d)
       common          / wth_tbl       / wth_tbl(max_2d)
       common          / d_tbl         / d_tbl(max_2d)
       common          / dw_tbl        / dw_tbl(max_3d)
       common          / dwth_tbl      / dwth_tbl(max_2d)
       common          / sn_tbl / isn_cnt , sn_tbl(3,isn_buf_max)
```

１０．前注意過程における図形の特徴点抽出モデルと画像復元

プログラム　リスト

(1) ＮＣＵＢＥホスト系プログラム

ｐｉｃｈｏｓｔ９　：　特徴点抽出モデル処理

(2) ＮＣＵＢＥノード系プログラム

ｐｉｃｎｏｄｅ９１　：　特徴点検出処理

ｐｉｃｎｏｄｅ９２　：　輪郭線検出処理

```
*  ***********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System          ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*  ***  MODEL NO.        : NCUBE/10    (VERTEX)                    ***
*  ***  LANGUAGE         : FORTRAN-77                              ***
*  ***  PROGRAM NAME     : pichost9                                ***
*  ***  PROGRAM FUNCTION :                                        ***
*  ***  ORDER NO.        : ---                                     ***
*  ***  LOT NO.          : ---                                     ***
*  ***  DOCUMENT NO.     : ---                                     ***
*  ***  MAKE OUT DATE    : 20-DEC-88                               ***
*  ***  VERSION NO.      : 10.0                                    ***
*  ***********************************************************************
*  ***    #    #####  ####      Advanced Telecommunication        ***
*  ***   # #     #    #   #      Reseach Institute International  ***
*  ***  #   #    #    ####      ---------------------------------- ***
*  ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*  ***  #   #    #    #   #      Research Laboratories            ***
*  ***********************************************************************

      program         pichost9

       call   open_load ()
       call   send1 ()
       call   title1 ()
       call   image_in ()
       call   send2 ()
       call   read1 ()
       call   send3 ()
       call   read2 ()
       stop
       end

*  ***********************************************************************
*  ***  PROGRAM NAME     : open_load                               ***
*  ***  PROGRAM FUNCTION : hypercube open & load subroutine        ***
*  ***********************************************************************

       subroutine          open_load ()

       common   /in_cnt/in_cnt, /ichan/ichan
       logical*1 lcflag(0:511)
       character*(*) cfile1, cfile2
       parameter (cfile1 = 'picnode91')
       parameter (cfile2 = 'picnode92')
       integer*1 iload_work(2**14)

*-----    Allocate and open a hypercube. -----
       idim  = 9
       icnt  = 2**idim
       ichan = nopen(idim)
       do 100 i = 0, 511
          lcflag(i) = 0
100    continue
       do 150 i = 0, 510, 2
```

```
         lcflag(i) = 1
150     continue
        istatus = nloadm(ichan, cfile1, icnt, lcflag, iload_work, 2**16)
        do 200 i = 0, 511
           lcflag(i) = 0
200     continue
        do 230 i = 1, 511, 2
           lcflag(i) = 1
230     continue
        istatus = nloadm(ichan, cfile2, icnt, lcflag, iload_work, 2**16)
999     return
        end

*  ********************************************************************
*  ***  PROGRAM NAME     : send1                                      ***
*  ***  PROGRAM FUNCTION : input parameter send subroutine            ***
*  ********************************************************************

      subroutine          send1 ()

       real*4   param(24)
```

パラメーター入力処理省略

```
*-----   send parameter to all nodes     -----
        do 100 i = 510, 0, -2
            isrc = i+1
            imsg_type = 51
            istatus = nwrite(ichan, param, 4*24, isrc, imsg_type)
            isrc = i
            istatus = nwrite(ichan, param, 4*24, isrc, imsg_type)
100     continue
999     return
        end

*  ********************************************************************
*  ***  PROGRAM NAME     : title1                                     ***
*  ***  PROGRAM FUNCTION : title graphics out subroutine              ***
*  ********************************************************************

      subroutine          title1 ()
```

固定画面グラッフィク出力処理省略

```
999     return
        end

*  ********************************************************************
*  ***  PROGRAM NAME     : image_in                                   ***
*  ***  PROGRAM FUNCTION : image input subroutine                     ***
*  ********************************************************************

      subroutine          image_in ()
```

原画像データ入力処理省略

```
999     return
        end

*   ********************************************************************
*   ***  PROGRAM NAME     : send2                                      ***
*   ***  PROGRAM FUNCTION : image data send subroutine                 ***
*   ********************************************************************

       subroutine          send2 ()

        common   /in_cnt/in_cnt,   /ichan/ichan
        common   /ileng/ileng(6),  /sigy1/sigy1, /iloop/iloop(2)
        integer*1 dat1, dat2
        common  /dat1/dat1(256,256), /dat2/dat2(256,256)

*----- Send to nodes picture parameter. -----
        ibase  = 256
        ryhalf = (sigy1*4.0)/2.0 -1.0
        iadr   = ibase - nint(ryhalf)
        do 100 i = 510, 0, -2
           imsg_type = 52
           do 110 j = 1, iloop(1)
              istatus = nwrite(ichan, dat1(1,iadr+(j-1)*119),
     1                         ileng(j), i+1, imsg_type)
              istatus = nwrite(ichan, dat1(1,iadr+(j-1)*119),
     1                         ileng(j), i   , imsg_type)
110        continue
           iadr = iadr - 1
100     continue
999     return
        end

*   ********************************************************************
*   ***  PROGRAM NAME     : read1                                      ***
*   ***  PROGRAM FUNCTION : zero crossing result data read             ***
*   ***                                                 subroutine     ***
*   ********************************************************************

       subroutine          read1 ()

        common    /in_cnt/in_cnt, /ichan/ichan, /ileng/ileng(6)
        common    /iend/iend, /sig_max/sig_max, /iloop/iloop(2)
        integer*1   icheck(256,256), irec(256)
        integer*2 icol1, icol2, icol3
        common    /icheck/icheck, /ima/ima, /ifno/ifno
        real*4     rtem(256)

        do 100 i =  510, 0, -2
            isrc = i
            imsg_type  = 151
            istatus = nread(ichan, irec, 256, isrc, imsg_type)
            do 150 j = 1, 256
               icheck(j,isrc/2+1) = irec(j)
150         continue
```

```
100     continue
999     return
        end

*   ********************************************************************
*   ***  PROGRAM NAME     : send3                                      ***
*   ***  PROGRAM FUNCTION : zero crossing data send subroutine        ***
*   ********************************************************************

       subroutine          send3 ()

        integer*1   icheck(256,256)
        real*4        param(24), sigy2
        common   /in_cnt/in_cnt,  /ichan/ichan, /icheck/icheck
        common   /ileng/ileng(6), /sigy2/sigy2, /param/param

*-----  Send to nodes picture parameter. -----
        ibase  = 256
        iadr   = ibase - int(sigy2)
        do 200 i = 510, 0, -2
           imsg_type = 53
           istatus = nwrite(ichan, icheck(1,iadr),ileng(4),i, imsg_type)
           iadr = iadr - 1
200     continue
999     return
        end

*   ********************************************************************
*   ***  PROGRAM NAME     : read2                                      ***
*   ***  PROGRAM FUNCTION : result data read subroutine               ***
*   ********************************************************************

       subroutine          read2 ()

        integer*1   dat2(256,256)
        real*4        param(24)
        common    /in_cnt/in_cnt, /ichan/ichan
        common    /iend/iend, /sig_max/sig_max, /iloop/iloop(2)
        common    /ileng/ileng(6), /dat2/dat2, /param/param
        integer*2 icol1, icol2, icol3, iza(2,200)
        common    /jxpos2/jxpos2 , /jypos2/jypos2
        common    /icol1/icol1 ,   /icol2/icol2,    /icol3/icol3
        common    /ishiro/ishiro, /ikuro/ikuro, /iall/iall, /ihow/ihow
        integer*1   iread_bufa(256,3), iori(256,3), ib(256,47), ir(256)
        real*4      rread_bufa(256), rtd(256,9), rtm(256), rdata(2,256)

        ishiro = 0
        ikuro  = 0
        icup = 1
*----    receive calculate data from nodes    ----
        do 100 i = 510, 0, -2
            isrc = -1
            imsg_type  = 161
            istatus = nread(ichan, ir, 256, isrc, imsg_type)
            do 101 j = 1, 256
```

```
            dat2(j,isrc/2+1) = ir(j)
 101        continue
            imsg_type  = 162
            istatus = nread(ichan, iori, 256*3, isrc, imsg_type)
            imsg_type  = 1234
            istatus = nread(ichan, rdata(1,isrc/2+1), 4*2,isrc,imsg_type)
            if ( icup .eq. 1 ) then
               call ldcl1(icol1)
               call amove( jxpos2+1, jypos2 )
               call rfrct( jxarea, jyarea )
               call gcmd()
               icup = icup + 1
            endif
*----     graphics out calculate data     ----
            do 300 k = 3, 252
               if ( dat2(k,isrc/2+1) .eq. -1 )  then
                  call ldcl1 (icol2)
                  call amove ( jxpos2+2+k*2, jypos2+isrc)
                  call crcl  (1)
                  call paint (0)
                  call gcmd  ()
                  ishiro = ishiro + 1
                elseif ( dat2(k,isrc/2+1) .eq. 0 )  then
                  ikuro = ikuro + 1
               endif
 300         continue
 100    continue
            do 333 isrc = 510, 0, -2
               do 333 k = 3, 252
                  if ( dat2(k,isrc/2+1) .eq. -1 )  then
                   elseif ( dat2(k,isrc/2+1) .eq. 0 )  then
                     call ldcl1 (icol3)
                     call amove ( jxpos2+2+k*2, jypos2+isrc)
                     call crcl  (4)
                     call paint (0)
                     call gcmd  ()
                  endif
 333         continue
 999    return
        end
```

```
*  ********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System           ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs.  ***
*  ***  MODEL NO.        : NCUBE/10    (VERTEX)                     ***
*  ***  LANGUAGE         : FORTRAN-77                               ***
*  ***  PROGRAM NAME     : picnode91                                ***
*  ***  MAKE OUT DATE    : 20-DEC-88                                ***
*  ***  VERSION NO.      : 10.1                                     ***
*  ********************************************************************
      program         picnode91
       real*4     param(24)
       integer*1  iread_buf(256,256)
       common    /id_node/id_node, /id_process/id_process,
     1            /id_host/id_host, /icube_size/icube_size
       common    /param/param, /iread_buf/iread_buf
       integer*4  ileng(6), iloop(2)
       call whoami(id_node, id_process, id_host, icube_size)
10     continue
       isrc = -1
       imsg_type = 51
       istatus = nread( param, 4*24, isrc, imsg_type, icflag)
                  入力パラメーターセット処理省略
       isrc = -1
       imsg_type = 52
       do 100 i = 1, iloop(1)
          istatus = nread( iread_buf(1,(i-1)*119+1), ileng(i),
     1                                    isrc, imsg_type, icflag)
100    continue
       call conv1 ()
       isrc = -1
       imsg_type = 53
       istatus = nread( iread_buf(1,1),ileng(4),isrc,imsg_type,icflag)
       call conv3 ()
       go to 10
999    stop
       end
*  ********************************************************************
*  ***  PROGRAM NAME     : conv1                                    ***
*  ***  PROGRAM FUNCTION : zero crossing subroutine                 ***
*  ********************************************************************
      subroutine          conv1 ()
       integer*1  iread_buf(256,256), icheck(256)
       real*4     tempd(256,9), param(24)
       common    /param/param, /iread_buf/iread_buf, /id_node/id_node
       common     /inod/inod, /tempd/tempd, /icheck/icheck
       real*4    tempb(256), tempc(256,3), rdata(4), rconst(79*79)
       common    /tempb/tempb, /rconst/rconst, /tempc/tempc
                   入力パラメーターセット処理省略
       raby = ryhalf + 1.0
       pai     = 3.141592653
       ic = 1
       rdat1 = 1.0 / (pai*(sigm**4))
       rdat4 =  2.0*(sigm**2)
       do 50 mmy = myhalf, -myhalf, -1
          do 51 mmx = -mxhalf, mxhalf
```

```
            rad   = float(mmx**2 + mmy**2)
            rdat2 = 1.0 - ( rad / rdat4 )
            rwk = -1*rad / rdat4
            if ( rwk .le. -87.0 )  rwk = -87.0
            if ( rwk .ge. 88.0 )  rwk = 88.0
            rdat3 = exp(rwk)
            rconst(ic)= rdat1*rdat2*rdat3
            ic = ic + 1
51         continue
50       continue
           radr = raby
           do 200 nnx = 1, 128, 1
              temp = 0.0
              ic = 1
              do 300 mmy = myhalf, -myhalf, -1
                 iy = int(radr) + mmy
                 do 400 mmx = -mxhalf, mxhalf
                   ix = nnx + mmx
                      temp = temp+float(iread_buf(ix,iy))*rconst(ic)
                      ic = ic + 1
400              continue
300           continue
              tempd(nnx,2) = temp
200        continue
       isrc =  id_node + 1
       imsg_type = 70
       istatus = nread( tempd(129,2), 128*4, isrc, imsg_type, icflag)
       if ( id_node .eq. 0 )  then
          istatus = nwrite(tempd(1,2), 256*4, id_node+2, 71, icflag)
          istatus = nread( tempd(1,3), 256*4, id_node+2, 71, icflag)
       elseif ( id_node .eq. 510 )  then
          istatus = nwrite(tempd(1,2), 256*4, id_node-2, 71, icflag)
          istatus = nread( tempd(1,1), 256*4, id_node-2, 71, icflag)
       else
          istatus = nwrite(tempd(1,2), 256*4, id_node-2, 71, icflag)
          istatus = nwrite(tempd(1,2), 256*4, id_node+2, 71, icflag)
          istatus = nread( tempd(1,1), 256*4, id_node-2, 71, icflag)
          istatus = nread( tempd(1,3), 256*4, id_node+2, 71, icflag)
       endif
       do 800 nnx = 1, 256, 1
          do 810 idy = 1, -1, -1
             do 820 idx = -1, 1
                if ( tempd(nnx,2) .lt. 0 )  then
                   iax = nnx + idx
                   iay = 2 + idy
                   if ( tempd(iax,iay) .gt. 0 )  icheck(nnx) = 1
                endif
820          continue
810       continue
800    continue
       isrc = 151
       istatus = nwrite(icheck, 256, id_host, isrc, icflag)
999    return
       end
*  ***********************************************************************
```

```
*  ***  PROGRAM NAME     : conv3                                          ***
*  ***********************************************************************
      subroutine          conv3 ()
       integer*1  iread_buf(256,256), icheck(256)
       real*4     param(24), tempd(256,9), tempc_max(256)
       common    /param/param, /iread_buf/iread_buf, /icheck/icheck
       common    /sigx/sigx,   /x0/x0,  /sigy/sigy,  /sigx22/sigx22
       common    /inod/inod, /tempd/tempd, /iend/iend,/id_host/id_host
       real*4    xm(256*256), tempb(256,36), tempc(256,36)
       integer*1  isend_bufa(256,3), isend_bufb(256,3), iori(256,3)
       real*4    rdat(-4:1,-15:15)
              入力データセット処理省略
       pai = 3.141592
          ic = 1
           rmax = 0.0
          do 300 i = 0, 35, isitta
              do 400 nnx = 10, 246, 1
                temp = 0.0
                if ( icheck(nnx) .eq. 1 ) then
                    sitta = float(i)*pai / 18.0
                    rcos  = cos(sitta)
                    rsin  = sin(sitta)
                    do 500 rmy = ryhalf, -ryhalf, -1.0
                       yy1 = rmy
                       do 600 rmx = -4.0, 1.0, 1.0
                          xx1 = rmx
                          xx =  xx1*rcos + yy1*rsin + (nnx+0.0)
                          yy = -xx1*rsin + yy1*rcos + ryab
                          ix1 = (xx*10.0+5.0)/10.0
                          iy1 = (yy*10.0+5.0)/10.0
                          if ( iread_buf(ix1,iy1) .eq. 1 )  then
                             temp = temp + 1.0
                          endif
600                     continue
500                 continue
                  if ( temp .gt. rmax )  rmax = temp
                  tempc(nnx,ic) = temp
            endif
400         continue
            ic = ic + 1
300       continue
1000    continue
       do 1400 i = 1, 256
          if ( tempc_max(i) .lt. param(11) )  then
             isend_bufa(i,2) = 0
          else
             isend_bufa(i,2) = -1
          endif
1400    continue
       imsg_type = 161
       istatus = nwrite(isend_bufa(1,2),256,id_host,imsg_type,icflag)
       imsg_type = 162
       istatus = nwrite(iori, 256*3, id_host, imsg_type, icflag)
999    return
       end
```

```
*  ********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System          ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*  ***  MODEL NO.        : NCUBE/10    (VERTEX)                    ***
*  ***  LANGUAGE         : FORTRAN-77                              ***
*  ***  PROGRAM NAME     : picnode92                               ***
*  ***  MAKE OUT DATE    : 02-DEC-88                               ***
*  ***  VERSION NO.      : 10.2                                    ***
*  ********************************************************************

      program         picnode92

       integer*4  ileng(6), iloop(2)
       real*4     param(24)
       integer*1  iread_buf(256,256), icheck(256,3)
       common    /id_node/id_node, /id_process/id_process,
     1            /id_host/id_host, /icube_size/icube_size
       common    /param/param, /iread_buf/iread_buf, /icheck/icheck

       call whoami(id_node, id_process, id_host, icube_size)
10     continue
       imsg_type = 51
       istatus = nread( param, 4*24, -1, imsg_type, icflag)
                 入力データセット処理省略
       imsg_type = 52
       do 100 i = 1, iloop(1)
          istatus = nread( iread_buf(1,(i-1)*119+1), ileng(i),
     1                                    -1, imsg_type, icflag)
100    continue
       call conv2 ()
       go to 10
999    stop
       end

*  ********************************************************************
*  ***  PROGRAM NAME     : conv2                                   ***
*  ********************************************************************

      subroutine          conv2 ()

       real*4     param(24), tempd(128), rdata(4), rconst(79**2)
       integer*1  iread_buf(256,256)
       common    /param/param, /iread_buf/iread_buf, /id_host/id_host

       call whoami(id_node, id_process, id_host, icube_size)
       if ( id_node .lt. 6 ) then
          irstr = id_node*6
          irend = irstr + 5
       else
          isho  = id_node / 6
          imod  = id_node - (isho*6)
          irstr = imod * 6
          irend = irstr + 5
       endif
                 入力データセット処理省略
```

```
      isrc   =  int(id_node / 6)
      isrc   =  isrc * 6
      raby = ryhalf + 1.0
      pai     = 3.141592653
c-----    Do convolution on image. -----
      ic = 1
      rdat1 = 1.0 / (pai*(sigm**4))
      rdat4 =  2.0*(sigm**2)
      do 50 mmy = myhalf, -myhalf, -1
          do 51 mmx = -mxhalf, mxhalf
             rad   = float(mmx**2 + mmy**2)
             rdat2 = 1.0 - ( rad / rdat4 )
             rwk = -1.0*rad / rdat4
             if ( rwk .le. -87.0 )  rwk = -87.0
             if ( rwk .ge. 88.0 )  rwk = 88.0
             rdat3 = exp(rwk)
             rconst(ic)= rdat1*rdat2*rdat3
             ic = ic + 1
51        continue
50     continue
         radr = raby
         do 100 nnx = 129, 256, 1
            temp = 0.0
            ic = 1
            do 200 mmy = myhalf, -myhalf, -1
               iy = int(radr) + mmy
               do 300 mmx = -mxhalf, mxhalf
                  ix = nnx + mmx
                  temp = temp+float(iread_buf(ix,iy))*rconst(ic)
                  ic = ic + 1
300            continue
200         continue
            tempd(nnx-128) = temp
100      continue
      imsg_no = 70
      isrc    = id_node - 1
      istatus = nwrite(tempd, 128*4, isrc, imsg_no, icflag)
 999  return
      end
```

１１．エネルギー学習をする局所並列確率アルゴリズムを用いた輪郭抽出プログラム　リスト

(1) ＮＣＵＢＥホスト系プログラム

ｌｅａｒｎｈ　：　輪郭線抽出処理

(2) ＮＣＵＢＥノード系プログラム

ｌｅａｒｎｎ　：　エネルギー学習処理

```
*  ***********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System          ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*  ***  MODEL NO.        : NCUBE/10    (AXIS)                      ***
*  ***  LANGUAGE         : FORTRAN-77                              ***
*  ***  PROGRAM NAME     : learnh (used node 128 version)          ***
*  ***  PROGRAM FUNCTION :                                        ***
*  ***  ORDER NO.        : ---                                     ***
*  ***  LOT NO.          : ---                                     ***
*  ***  DOCUMENT NO.     : ---                                     ***
*  ***  MAKE OUT DATE    : 20-DEC-88                               ***
*  ***  VERSION NO.      : 01.00                                   ***
*  ***********************************************************************
*  ***    #    #####  ####      Advanced Telecommunication        ***
*  ***   # #     #    #   #      Reseach Institute International  ***
*  ***  #   #    #    ####      ---------------------------------- ***
*  ***  #####    #    #  #      ATR Auditory and Visual Perception ***
*  ***  #   #    #    #   #      Research Laboratories            ***
*  ***********************************************************************
*
      program         learnh

       IMPLICIT INTEGER*2 (A-Z)
       PARAMETER(MAINSIZE=128)
       PARAMETER(MMSIZE=128)
       PARAMETER(CLASSIZE=10)
       COMMON /LUN/ LUN, /STRTXY/ IXST,IYST
       INTEGER*4 NOL,CH,BK,IMGSIZE,FRAME,LGTH,WDTH,TURBO,ESTNUM
       INTEGER*4 CUNTR1,CUNTR2,LOGNUM,BOL,STL,STEG,STLX,STEGX,I,J,K,L
       INTEGER*4 SEED, CLASS(CLASSIZE,2), O_LPNUM(CLASSIZE,2)
       INTEGER*4 B_LPNUM(CLASSIZE,2), C_LPNUM(CLASSIZE,2)
       REAL*4 EPS,PV0,PV2,LEG,ENG,TMPB,TMPE,C_TMP,V2DET
       REAL*4 C_ENERGY(2),C_PENVAL(CLASSIZE),GRBUF(MAINSIZE,MAINSIZE)
       integer*1 LPBGN(MAINSIZE,MAINSIZE,2)
       integer*1 LPMRF1(MAINSIZE,MAINSIZE,2)
       integer*1 LPMRF3(MAINSIZE,MAINSIZE,2)
       integer*1 IPBUF(MMSIZE,MMSIZE),IPMM(MMSIZE,MMSIZE)
       CHARACTER*100 FNAME
       CHARACTER*80 STRMEMO
       CHARACTER*1 DBUF(MAINSIZE,MAINSIZE/8,2)
       CHARACTER*1 CBUF(MAINSIZE*MAINSIZE/8,2)
       common  /GRBUF/GRBUF, /LPBGN/LPBGN, /LPMRF1/LPMRF1
       common  /IPBUF/IPBUF, /IPMM/IPMM, /LPMRF3/LPMRF3,/ichan/ichan
       integer*1 iload_work(2**14)
       character*(*)   cfile1
       parameter  (cfile1 = 'learnn' )
       integer*4 iidim, irtn, ichan, imsg_type, isrc, icnt, id
       logical*1 lcflag(0:511)
       dimension ip_map(0:511), il_map(0:511)
       real*4    rdata(8), const
       integer*1 igraf(512,4), idata(128*3*2)
       character*3     csta
       integer*4    iffchk, ios, ir1, ir2, ir3, ir4, ir5,in
       integer*4    iup, iflg
```

```
                イニシャル処理省略
      CALL PARLOAD(C_PENVAL,TMPB,TMPE,CLASSIZE,STRMEMO,
     &         BOL,NOL,EPS,PV0,CUNTR1)
      PV2 = C_PENVAL(2)
      const  = 2.0 / PV2
      FNAME = 'k_sekiyama.dat'
      CALL GRLOAD(GRBUF,IPBUF,MAINSIZE,IMGSIZE,MMSIZE,FNAME)
      FNAME = 'lpkaoruini.dat'
      CALL LPINIT(LPMRF3,DBUF,MAINSIZE,IMGSIZE,FNAME)
      CALL LPCOUNT(LPMRF3,0_LPNUM,CLASS,MAINSIZE,IMGSIZE,CLASSIZE)
*-----    Allocate and open a hypercube. -----
      idim  = 7
      icnt  = 2**idim
      ichan = nopen(idim)
*-----     Load  learnn  programs  to  nodes. -----
      do 1101 i = 0, 511
         lcflag(i) = 0
1101  continue
      do 1100 i = 0, icnt-1
         lcflag(i) = 1
1100  continue
      istatus = nloadm(ichan, cfile1, icnt, lcflag, iload_work, 2**16)
      iidim = idim
      call  sshcubes  ( iidim, ip_map, il_map, irtn )
               入力データセット処理省略
      do 2210 i = icnt-1, 0, -1
          imsg_type = 1000
          isrc = ip_map(i)
          istatus = nwrite(ichan, rdata,8*4, isrc, imsg_type)
          iadr = iadr - 1
2210  continue
      iadr = 127
      do 2200 i = icnt-1, 0, -1
          imsg_type = 1010
          isrc = ip_map(i)
          istatus = nwrite(ichan,GRBUF(1,iadr),128*3*4,isrc,imsg_type)
          iadr = iadr - 1
2200  continue
      iflg = 0
      do 3100 k = BOL, NOL
         if ( k.eq.iup*2+1 )  then
             iflg = 1
            iup = iup + 1
         else
            iflg = 0
         endif
         C_PENVAL(2) = PV2*const*TMPB*LOG(1.0)
         do 3200 i = icnt-1, 0, -1
            isrc = ip_map(i)
            imsg_type  = 1020
            istatus = nwrite(ichan, C_PENVAL, 10*4, isrc, imsg_type)
3200     continue
         do 3300 i = icnt-1, 0, -1
            isrc = ip_map(il_map(i))
            id = il_map(i)
```

```
            imsg_type  = 1040
            istatus = nread(ichan,LPMRF1(1,id+1,1),128,isrc,imsg_type)
            imsg_type  = 1050
            istatus = nread(ichan,LPMRF1(1,id+1,2),128,isrc,imsg_type)
3300      continue
          if ( iflg .eq. 1)  then
             call  lp_file  ( LPMRF1, ir5, iup, MAINSIZE, k )
          endif
          CALL LPCOUNT(LPMRF1,C_LPNUM,CLASS,MAINSIZE,IMGSIZE,CLASSIZE)
          LOGNUM = CUNTR1
          CALL LRNEXE(0_LPNUM,C_LPNUM,C_PENVAL,EPS,CLASSIZE,k+1,LOGNUM)
          call whg_file ( ir3, C_PENVAL, CLASSIZE, k )
          call  clp_file2  ( ir4, k, C_LPNUM, CLASSIZE )

                グラフィック出力処理省略

3100     continue
999     continue
        istatus = nclose(ichan)
        stop ' data correct normal termination ( learnh ) '
        end
C--------------------------------------
C--------- Load Gray Level Image ---------
        SUBROUTINE GRLOAD(GR,IPBUF,MAINSIZE,IMGSIZE,MMSIZE,FNAME)
                画像データ読み込み処理省略
999     RETURN
        END
C--------------------------------------
C------- PARAMETER LOAD ROUTINE ---------
        SUBROUTINE PARLOAD(C_PENVAL,TMPB,TMPE,CLASSIZE,STRMEMO,
     &          BOL,NOL,EPS,PV0,LOGNUM)
                入力パラメータ読み込み処理省略
999     RETURN
        END
C--------------------------------------
C----------- Load Micro Edges -----------
        SUBROUTINE LPINIT(LP,DBUF,MAINSIZE,IMGSIZE,FNAME)
                教師信号読み込み処理省略
999     RETURN
        END
C-------------------------------------
C---- Count Objective Line Processes ----
        SUBROUTINE LPCOUNT(LPMRF,LPNUM,CLASS,MAINSIZE,IMGSIZE,CLASSIZE)
        INTEGER*4 MAINSIZE,IMGSIZE,CLASSIZE,I,J,K,L
        INTEGER*4 CLASS(CLASSIZE,2),LPNUM(CLASSIZE,2)
        integer*1 LPMRF(MAINSIZE,MAINSIZE,2)
        DO 100 I=1,2
            DO 200 J=1,7
               LPNUM(J,I) = 0
200         continue
100     continue
        DO 300 I=1,2
            DO 400 J=2,IMGSIZE-1
                DO 500 K=2,IMGSIZE-1
                    CALL MECLASS(LPMRF,CLASS,MAINSIZE,CLASSIZE,J,K,I)
```

```
                DO 600 L=1,7
                    LPNUM(L,I) = LPNUM(L,I)+CLASS(L,I)
600             continue
500         continue
400     continue
300 continue
999 RETURN
    END
C--------------------------------------
C------- MICROEDGE CLASSIFICATION -------
    SUBROUTINE MECLASS(LP,CLASS,MAINSIZE,CLASSIZE,Y,X,FUNC)
    INTEGER*4 MAINSIZE,CLASSIZE,X,Y,FUNC,I,J,XP,YP,XM,YM,NBUF
    INTEGER*4 CLASS(CLASSIZE,2)
    LOGICAL*1 LP(MAINSIZE,MAINSIZE,2)
    XP = X+1
    YP = Y+1
    XM = X-1
    YM = Y-1
    DO 100 I=1,2
            DO 200 J=1,CLASSIZE
                    CLASS(J,I) = 0
200         continue
100 continue
    IF(FUNC.EQ.1) THEN
        NBUF = LP(X,Y,1) + LP(XP,YM,2) + LP(X,YM,1) + LP(X,YM,2)
        IF(NBUF.EQ.0) THEN
            CLASS(1,1) = CLASS(1,1)+1
        ELSE IF(NBUF.EQ.1) THEN
            CLASS(2,1) = CLASS(2,1)+1
        ELSE IF(NBUF.EQ.2) THEN
            IF ( ((LP(XP,YM,2).EQ.1).AND.LP(X,YM,2).EQ.1) .OR.
     &             ((LP(X,YM,1).EQ.1).AND.(LP(X,Y,1).EQ.1)) )  THEN
                CLASS(4,1) = CLASS(4,1)+1
            ELSE
                CLASS(3,1) = CLASS(3,1)+1
            ENDIF
        ELSE IF(NBUF.EQ.3) THEN
            CLASS(5,1) = CLASS(5,1)+1
        ELSE IF(NBUF.EQ.4) THEN
            CLASS(6,1) = CLASS(6,1)+1
        ELSE
        ENDIF
        IF((LP(X,Y,1).EQ.1).AND.(LP(XM,Y,1).EQ.1)) THEN
            CLASS(7,1) = CLASS(7,1)+1
        ENDIF
        NBUF = LP(X,Y,1)+LP(X,Y,2)+LP(X,YP,1)+LP(XP,Y,2)
        IF(NBUF.EQ.0) THEN
            CLASS(1,1) = CLASS(1,1)+1
        ELSE IF(NBUF.EQ.1) THEN
            CLASS(2,1) = CLASS(2,1)+1
        ELSE IF(NBUF.EQ.2) THEN
            IF((LP(X,Y,2).EQ.1).AND.(LP(XP,Y,2).EQ.1).OR.
     &             (LP(X,Y,1).EQ.1).AND.(LP(X,YP,1).EQ.1)) THEN
                CLASS(4,1) = CLASS(4,1)+1
            ELSE
```

```
            CLASS(3,1) = CLASS(3,1)+1
        ENDIF
    ELSE IF(NBUF.EQ.3) THEN
        CLASS(5,1) = CLASS(5,1)+1
    ELSE IF(NBUF.EQ.4) THEN
        CLASS(6,1) = CLASS(6,1)+1
    ELSE
    ENDIF
    IF((LP(X,Y,1).EQ.1).AND.(LP(XP,Y,1).EQ.1)) THEN
        CLASS(7,1) = CLASS(7,1)+1
    ENDIF
ELSE IF(FUNC.EQ.2) THEN
    NBUF = LP(X,Y,2)+LP(X,YP,1)+LP(XP,Y,2)+LP(X,Y,1)
    IF(NBUF.EQ.0) THEN
        CLASS(1,2) = CLASS(1,2)+1
    ELSE IF(NBUF.EQ.1) THEN
        CLASS(2,2) = CLASS(2,2)+1
    ELSE IF(NBUF.EQ.2) THEN
        IF((LP(X,YP,1).EQ.1).AND.(LP(X,Y,1).EQ.1).OR.
&               (LP(X,Y,2).EQ.1).AND.(LP(XP,Y,2).EQ.1)) THEN
            CLASS(4,2) = CLASS(4,2)+1
        ELSE
            CLASS(3,2) = CLASS(3,2)+1
        ENDIF
    ELSE IF(NBUF.EQ.3) THEN
        CLASS(5,2) = CLASS(5,2)+1
    ELSE IF(NBUF.EQ.4) THEN
        CLASS(6,2) = CLASS(6,2)+1
    ELSE
    ENDIF
    IF((LP(X,Y,2).EQ.1).AND.(LP(X,YM,2).EQ.1)) THEN
        CLASS(7,2) = CLASS(7,2)+1
    ENDIF
    NBUF = LP(X,Y,2)+LP(XM,Y,1)+LP(XM,Y,2)+LP(XM,YP,1)
    IF(NBUF.EQ.0) THEN
        CLASS(1,2) = CLASS(1,2)+1
    ELSE IF(NBUF.EQ.1) THEN
        CLASS(2,2) = CLASS(2,2)+1
    ELSE IF(NBUF.EQ.2) THEN
        IF((LP(XM,Y,1).EQ.1).AND.(LP(XM,YP,1).EQ.1).OR.
&               (LP(X,Y,2).EQ.1).AND.(LP(XM,Y,2).EQ.1)) THEN
            CLASS(4,2) = CLASS(4,2)+1
        ELSE
            CLASS(3,2) = CLASS(3,2)+1
        ENDIF
    ELSE IF(NBUF.EQ.3) THEN
        CLASS(5,2) = CLASS(5,2)+1
    ELSE IF(NBUF.EQ.4) THEN
        CLASS(6,2) = CLASS(6,2)+1
    ELSE
    ENDIF
    IF((LP(X,Y,2).EQ.1).AND.(LP(X,YP,2).EQ.1)) THEN
        CLASS(7,2) = CLASS(7,2)+1
    ENDIF
ENDIF
```

```
      do 2222 ijk = 1,2
              CLASS(4,ijk) = CLASS(4,ijk) +   CLASS(3,ijk)
              CLASS(3,ijk) = 0
2222  continue
999   RETURN
      END
C-------------------------------------------------------
C--- Learn Existentian Probabilities of Line Processes ---
      SUBROUTINE LRNEXE(O_LPNUM,C_LPNUM,C_PENVAL,EPS,
     1                  CLASSIZE,BOL,LOGNUM)
      INTEGER*4 CLASSIZE,BOL,LOGNUM,O_NBUF,C_NBUF,I
      integer*4 C_LPNUM(CLASSIZE,2), O_LPNUM(CLASSIZE,2)
      REAL*4 C_PENVAL(CLASSIZE),EPS
       DO 100 I=4,7
          O_NBUF = O_LPNUM(I,1)+O_LPNUM(I,2)
          C_NBUF = C_LPNUM(I,1)+C_LPNUM(I,2)
          IF(O_NBUF.EQ.0) THEN
              C_PENVAL(I) = C_PENVAL(I) - (EPS*float(O_NBUF-C_NBUF))
          ELSE
              C_PENVAL(I) = C_PENVAL(I) -
     &                      (EPS*float(O_NBUF-C_NBUF)/float(O_NBUF))
          ENDIF
100   continue
999   RETURN
      END
*  ********************************************************************
*  ***  PROGRAM NAME     : lp_file                                    ***
*  ***  PROGRAM FUNCTION : LPMRF1 output to file subroutine          ***
*  ********************************************************************
      subroutine          lp_file  ( LPMRF1, ir5, iup, MAINSIZE, k )
                ラインプロセス　ファイル出力処理省略
999   return
      end
*  ********************************************************************
*  ***  PROGRAM NAME     : whg_file                                   ***
*  ***  PROGRAM FUNCTION : WHIGHT output to file subroutine          ***
*  ********************************************************************
      subroutine          whg_file ( ir3, C_PENVAL, CLASSIZE, k )
                重みファイル出力処理省略
999   return
      end
*  ********************************************************************
*  ***  PROGRAM NAME     : clp_file2                                  ***
*  ***  PROGRAM FUNCTION : C_LPNUM output to file subroutine         ***
*  ********************************************************************
      subroutine          clp_file2  ( ir4, k, C_LPNUM, CLASSIZE )
                ﾗｲﾝﾌﾟﾛｾｽ ｵﾘｴﾝﾃｰﾃｮﾝ ﾌｧｲﾙ出力処理省略
999   return
      end
```

```
*  ********************************************************************
*  ***  SYSTEM           : ATR  Parallel Computing System          ***
*  ***  PROJECT          : Auditory & Visual Perception Res. Labs. ***
*  ***  MODEL NO.        : NCUBE/10    (VERTEX)                    ***
*  ***  LANGUAGE         : FORTRAN-77                              ***
*  ***  PROGRAM NAME     : LEARNN                                  ***
*  ***  MAKE OUT DATE    : 20-DEC-88                               ***
*  ***  VERSION NO.      : 01.1                                    ***
*  ********************************************************************
*
      program         learnn

       IMPLICIT INTEGER*2 (A-Z)
       PARAMETER(MAINSIZE=128)
       PARAMETER(MMSIZE=128)
       PARAMETER(CLASSIZE=10)
       integer*4   id_node, id_process, id_host, icube_size
       common    /id_node/id_node, /id_process/id_process,
     1            /id_host/id_host, /icube_size/icube_size
       COMMON /STRTXY/ IXST,IYST, /LUN/ LUN
       INTEGER*4 NOL,CH,BK,IMGSIZE,FRAME,LGTH,WDTH,ESTNUM
       INTEGER*4 CUNTR1,CUNTR2,LOGNUM,BOL,STL,STEG,STLX,STEGX,I,J,K,L
       INTEGER*4 CLASS(CLASSIZE,2), SEED, O_LPNUM(CLASSIZE,2)
       INTEGER*4 C_LPNUM(CLASSIZE,2), B_LPNUM(CLASSIZE,2)
       REAL*4 EPS,PV0,LEG,TMPB,TMPE,C_TMP,V2DET,E_CNST,PV2, const
       REAL*4 C_ENERGY(2), ENG, TURBO, C_PENVAL(CLASSIZE)
       REAL*4 GRBUF(MAINSIZE,3), rbuf(MAINSIZE), rdata(8)
       integer*1 LPMRF1(MAINSIZE,3,2), LPBGN(MAINSIZE,3,2)
       integer*2   ip_map(0:511), il_map(0:511)
       integer*4   isrc, imsg_type, imy_node, iup_node, idw_node
       integer*4   icflag, in0, in127, iretn

                イニシャル処理省略
       TURBO = 100.0
       ESTNUM = 1000
       call  sshcubes  ( 7, ip_map, il_map, iretn )
       call whoami(id_node, id_process, id_host, icube_size)
       isrc = -1
       imsg_type = 1000
       istatus = nread( rdata, 8*4, isrc, imsg_type, icflag)
                読み込みデータセット処理省略
*-----     gazou  parameter  recieve     -----
       isrc = -1
       imsg_type = 1010
       istatus = nread( GRBUF, 128*3*4, isrc, imsg_type, icflag)
       10     continue
 *-----     C_PENVAL recieve from host   -----
          isrc = -1
          imsg_type = 1020
          istatus = nread( C_PENVAL, 10*4, isrc, imsg_type, icflag)
           C_TMP = TMPB
           SEED = 12345679
           CALL LPRST(LPBGN,LPMRF1,MAINSIZE,IMGSIZE,SEED)
            do 100 J = 1, ESTNUM
              if ( C_TMP .LT. TMPE )  GO TO 900
```

```
              STL = 0
              STEG = 0
              CALL MRFEXE(GRBUF,LPMRF1,CLASS,C_ENERGY,PV0,
     &                    C_PENVAL,MAINSIZE,IMGSIZE,MMSIZE,
     1                                 CLASSIZE,C_TMP,SEED)
              call  node_comu ( LPMRF1, ip_map, il_map, MAINSIZE )
              ENG = C_ENERGY(1) + C_ENERGY(2)
              C_TMP = TMPB/LOG(TURBO*float(J-1)+2.71828)
              C_PENVAL(2) = PV2*const*TMPB*LOG(float(J))/
     &                           LOG(TURBO*float(J-1)+2.71828)
100         continue
900     continue
           imsg_type = 1040
           istatus = nwrite(LPMRF1(1,2,1),128,id_host,imsg_type,icflag)
           imsg_type = 1050
           istatus = nwrite(LPMRF1(1,2,2),128,id_host,imsg_type,icflag)
         go to 10
999     stop
        end
C--------------------------------------
C---- Line Processes Reset Routine ---
        SUBROUTINE LPRST(LPBGN,LPBUF,MAINSIZE,IMGSIZE,SEED)
C
        integer*4   id_node, id_process, id_host, icube_size
        common    /id_node/id_node, /id_process/id_process,
     1            /id_host/id_host, /icube_size/icube_size
        INTEGER*4 MAINSIZE,IMGSIZE,I,J,K,l, SEED
        REAL*4 NBUF
        integer*1 LPBUF(MAINSIZE,3,2), LPBGN(MAINSIZE,3,2)
        DO 100 I=1,2
            DO 200 J=1,3
                DO 300 K=1,IMGSIZE
                    do 400 l = 1, id_node+1
                       NBUF = RAND1(l)
400                 continue
                    IF(NBUF.LT.0.5) THEN
                        LPBUF(K,J,I) = 0
                    ELSE
                        LPBUF(K,J,I) = 1
                    ENDIF
300             continue
200         continue
100     continue
999     RETURN
        END
C--------------------------------------
C--- Assume Existential Expectations of Line Processes ---
        SUBROUTINE MRFEXE(GRBUF,LPMRF1,CLASS,ENERGY,PV0,
     &          C_PENVAL,MAINSIZE,IMGSIZE,MMSIZE,CLASSIZE,TMP,SEED)

         integer*4   id_node, id_process, id_host, icube_size
        common    /id_node/id_node, /id_process/id_process,
     1            /id_host/id_host, /icube_size/icube_size
        INTEGER*4 MAINSIZE,IMGSIZE,MMSIZE,CLASSIZE,I,J,K,L
        INTEGER*4 SEED, ijk,i1,i0, CLASS(CLASSIZE,2)
```

```
      REAL*4 PV0, rwk, ENERGY(2), U0, U1, C_PENVAL(CLASSIZE)
      REAL*4 PBUF0,PBUF1,P0,P1,DICE,NTMP,TMP, GRBUF(MAINSIZE,3)
      integer*1 LPMRF1(MAINSIZE,3,2), LPBUF
      i1 = 0
      i0 = 0
      ENERGY(1) = 0.0
      ENERGY(2) = 0.0
      SEED  =  12345679
      do 888 iik = 1, id_node+1
         DICE = RAND1(SEED)
 888  continue
          DO 100 J=2,2
              DO 200 K=2,IMGSIZE-1
                  I = 1
                  U0 = 0.0
                  U1 = 0.0
                  LPMRF1(K,J,I) = 0
                  CALL MECLASS(LPMRF1,CLASS,MAINSIZE,CLASSIZE,J,K,I)
                  DO 300 L=1,7
                      U0 = U0+(C_PENVAL(L)*float(CLASS(L,I)))
300               continue
                  rwk = GRBUF(K+1,J)-GRBUF(K,J)
                  U0 = U0+(PV0*rwk*rwk)
                  LPMRF1(K,J,I) = 1
                  CALL MECLASS(LPMRF1,CLASS,MAINSIZE,CLASSIZE,J,K,I)
                  DO 400 L=1,7
                      U1 = U1+(C_PENVAL(L)*float(CLASS(L,I)))
400               continue
                  PBUF0 = 1
                  NTMP = (U0-U1)/TMP
                  IF(NTMP.ge.88.0) THEN
                      PBUF1 = 1.7E38
                  ELSE IF(NTMP.le.-87.0) THEN
                      PBUF1 = 0.0
                  ELSE
                      PBUF1 = EXP(NTMP)
                  ENDIF
                  P0 = PBUF0/(PBUF0+PBUF1)
                  DICE = RAND1(SEED)
                  DICE = (DICE+1.0) / 2.0
                   IF(DICE.GE.P0) THEN
                      LPMRF1(K,J,I) = 1
                      ENERGY(I) = ENERGY(I)+U1
                      i1 = i1 + 1
                  ELSE
                      LPMRF1(K,J,I) = 0
                      ENERGY(I) = ENERGY(I)+U0
                      i0 = i0 + 1
                  ENDIF
          I=2
                  U0 = 0.0
                  U1 = 0.0
                  LPMRF1(K,J,I) = 0
                  CALL MECLASS(LPMRF1,CLASS,MAINSIZE,CLASSIZE,J,K,I)
                  DO 500 L=1,7
```

```
                    U0 = U0+(C_PENVAL(L)*float(CLASS(L,I)))
500                 continue
                    rwk = GRBUF(K,J+1)-GRBUF(K,J)
                    U0 = U0+(PV0*rwk*rwk)
                    LPMRF1(K,J,I) = 1
                    CALL MECLASS(LPMRF1,CLASS,MAINSIZE,CLASSIZE,J,K,I)
                    DO 600 L=1,7
                        U1 = U1+(C_PENVAL(L)*float(CLASS(L,I)))
600                 continue
                    PBUF0 = 1
                    NTMP = (U0-U1)/TMP
                    IF(NTMP.ge.88.0) THEN
                        PBUF1 = 1.7E38
                    ELSE IF(NTMP.le.-87.0) THEN
                        PBUF1 = 0.0
                    ELSE
                        PBUF1 = EXP(NTMP)
                    ENDIF
                    P0 = PBUF0/(PBUF0+PBUF1)
                    P1 = PBUF1/(PBUF0+PBUF1)
                    DICE = RAND1(SEED)
                    DICE = (DICE+1.0) / 2.0
                    IF(DICE.GE.P0) THEN
                        LPMRF1(K,J,I) = 1
                        ENERGY(I) = ENERGY(I)+U1
                    ELSE
                        LPMRF1(K,J,I) = 0
                        ENERGY(I) = ENERGY(I)+U0
                    ENDIF
200             continue
100         continue
        DO 1000 iv = 1 , 2
            DO 1100 iy = 1, 3
                LPMRF1(1,iy,iv) = 0
                LPMRF1(128,iy,iv) = 0
1100        continue
1000    continue
999     RETURN
        END
C-------------------------------------
C------- MICROEDGE CLASSIFICATION -------
        SUBROUTINE MECLASS(LP,CLASS,MAINSIZE,CLASSIZE,Y,X,FUNC)

                    l e a r n hのMECLASSと同じ
999     return
        end
*   ********************************************************************
*   ***  PROGRAM NAME     : node_comu                                  ***
*   ***  PROGRAM FUNCTION : up & down node comunication subroutine  ***
*   ********************************************************************
      subroutine          node_comu ( LPMRF1, ip_map, il_map, MAINSIZE )

        implicit    integer*4   (i-n)
        common      /id_node/id_node
        integer*2   ip_map(0:511), il_map(0:511)
```

```
      integer*1   LPMRF1(MAINSIZE,3,2)
            imy_node = il_map(id_node)
            iup_node = ip_map(imy_node+1)
            idw_node = ip_map(imy_node-1)
            imsg_type = 1030
            igu = mod(imy_node,2)
          if ( id_node .eq. ip_map(1) )  then
            istatus=nwrite(LPMRF1(1,2,1),128,iup_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,3,1),128,iup_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,2),128,iup_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,3,2),128,iup_node,imsg_type,icflag)
            do 1000 ix = 1, 128
               LPMRF1(ix,1,1) = 0
               LPMRF1(ix,1,2) = 0
1000        continue
          elseif ( id_node .eq. ip_map(126) )  then
            istatus=nread(LPMRF1(1,1,1),128,idw_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,1),128,idw_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,1,2),128,idw_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,2),128,idw_node,imsg_type,icflag)
            do 1100 ix = 1, 128
               LPMRF1(ix,3,1) = 0
               LPMRF1(ix,3,2) = 0
1100        continue
          else
            if ( igu .eq. 1 )   then
            istatus=nwrite(LPMRF1(1,2,1),128,iup_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,3,1),128,iup_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,2),128,iup_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,3,2),128,iup_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,1),128,idw_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,1,1),128,idw_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,2),128,idw_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,1,2),128,idw_node,imsg_type,icflag)
            else
            istatus=nread(LPMRF1(1,1,1),128,idw_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,1),128,idw_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,1,2),128,idw_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,2),128,idw_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,3,1),128,iup_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,1),128,iup_node,imsg_type,icflag)
            istatus=nread(LPMRF1(1,3,2),128,iup_node,imsg_type,icflag)
            istatus=nwrite(LPMRF1(1,2,2),128,iup_node,imsg_type,icflag)
            endif
         endif
999   return
      end
```