The University of Osaka
Institutional Knowledge Archive

Title	並列光演算システムに関する研究
Author(s)	谷田, 純
Citation	大阪大学, 1986, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/1606
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

並列光演算システムに関する研究

昭和60年11月

谷田　純

並列光演算システムに関する研究

谷　田　　純

目　　次

緒　　論 …… 1

第１章　並列光論理演算法 …… 4

1.1　緒　　言 …… 4

1.2　並列２変数２値論理演算 …… 6

1.2.1　並列光論理演算法の概要と特徴 …… 6

1.2.2　入力画像の符号化 …… 8

1.2.3　多重投影光学系 …… 10

1.2.4　並列論理演算の実現 …… 11

1.2.5　並列シフト演算 …… 15

1.2.6　実験結果 …… 15

1.2.7　高密度化に対する考察 …… 19

1.3　並列光論理演算法の一般化 …… 20

1.4　結　　言 …… 25

第２章　符号化法と実時間２次元相関演算法 …… 26

2.1　緒　　言 …… 26

2.2　符号化法 …… 26

2.2.1　カラー符号化法 …… 26

2.2.2　ＣＧＨフィルタリング符号化法 …… 28

2.2.3　偏光面符号法 …… 31

2.2.4　複屈折符号化法 …… 33

2.3　実時間２次元相関演算法 …… 34

2.3.1　多重投影法 …… 35

2.3.2　多重結像法 …… 35

2.3.3　回折パターンサンプル法 …… 38

2.4　結　　言 …… 39

第３章　並列光論理演算法の画像処理への応用 …… 40

3.1　緒　　言 …… 40

3.2　並列２値画像処理 …… 40

3.2.1　並列光ディジタルフィルタリング …… 41

3.2.2　並列光論理テンプレートマッチング …… 43

3.3　並列多値画像処理 …… 46
3.3.1　ビット分散配置法 …… 46
3.3.2　ディジタル / アナログ変換法 …… 50
3.3.3　多値符号法 …… 51
3.4　結　　言 …… 54
第4章　光アレイロジック …… 55
4.1　緒　　言 …… 55
4.2　光アレイロジックの概念 …… 55
4.2.1　アレイロジックの拡張 …… 56
4.2.2　演算処理の指定 …… 60
4.2.3　並列光論理演算法による実現 …… 62
4.3　光アレイロジックによる並列処理 …… 65
4.3.1　エッジ検出 …… 65
4.3.2　並列2ビット加算器 …… 67
4.3.3　並列剰余演算器 …… 71
4.4　結　　言 …… 75
第5章　並列光演算システム：OPALS …… 76
5.1　緒　　言 …… 76
5.2　並列光演算システム ── OPALS ── の概念 …… 77
5.3　光 / 電子複合型OPALS …… 80
5.3.1　システム構成 …… 81
5.3.2　実験システム …… 82
5.4　純光学型OPALS …… 84
5.4.1　光機能素子 …… 85
5.4.1.1　符号化素子 …… 85
5.4.1.2　実時間2次元相関演算素子 …… 86
5.4.1.3　OR演算素子 …… 86
5.4.1.4　一時記憶素子 …… 87
5.4.2　システム構成 …… 88
5.5　結　　言 …… 89
第6章　OPALSのシステム構成に対する考察 …… 90
6.1　緒　　言 …… 90

6.2　光順序回路型OPALS …… 90
6.2.1　光順序回路技術 …… 91
6.2.2　OPALSの処理手順の順序回路化 …… 92
6.2.3　波長多重並列光論理演算法 …… 95
6.2.4　システムの構成 …… 97
6.2.5　計算機シミュレーション …… 101
6.3　OPALSの光コンピュータへの発展性 …… 102
6.3.1　問題点の認識と対応 …… 103
6.3.2　機能モジュール型OPALS …… 104
6.3.3　OPALSの能力限界 …… 107
6.4　結　　言 …… 109

総　　括 …… 110
謝　　辞 …… 113
参考文献 …… 114

緒　　論

光を情報媒体とする光情報処理技術は，処理能力の大きさと優雅な処理方法を特徴としており，次世代の新しい大容量高速情報処理システム開発の基礎技術として，各方面から期待されている。光による情報処理は並列性・高速性・非干渉性を特徴とし，特に大容量情報に対する並列処理能力は他に類がないものである。[13-15] この並列情報処理技術を基本とした光演算システムの能力は，現在，認識され始めた電子計算機の能力の限界を大幅に上回る可能性を持っている。それゆえ，光の特性を生かした新しい情報処理システムの開発研究は，情報処理の分野から現在，最も注目をあびている研究対象の1つとなっている。

光学情報処理技術を利用した新しい情報処理システム開発に対する期待の背景には，高度情報化社会における情報処理への需要の爆発的増大がある。たとえば，衛星写真や医用画像を対象とした画像処理，流体力学分野における偏微分方程式の計算，さらに，知識工学における大容量データの検索・処理等々で要求される大容量情報処理は，現在の電子計算機を基本とした情報処理システムで実行し得る処理能力の限界をすでに越えている。[16-18]

もちろん，現在の情報処理システムの中心は汎用大型計算機であり，それ自体，十分完成されたシステムである。また，電子計算機の高性能化，大容量情報処理能力向上への努力もハードウェア，ソフトウェア両面から活発に進められている。たとえば，ハードウェアの面ではジョゼフソン素子[19]やHEMT素子[20]などの超高速スイッチング素子の開発，パイプラインプロセッサ[21]やアレイプロセッサ[22]等の様々なレベルの並列処理アーキテクチャの実用化があげられる。また，ソフトウェアの面では高速演算アルゴリズムの開発，並列プログラミング法の研究などの努力が続けられている。このようなハードウェア，ソフトウェア両面からの努力により電子計算機の演算能力は日に日に高まっており，極めて高度な処理にも対応できるようになっている。そして，超高性能電子計算機 ― スーパーコンピュータ ― が開発されるに及んでいる。

しかし，上述した応用分野における大容量情報の高速演算の実行にはスーパーコンピュータでさえも，まだその能力が不足していることが認識されている。[18] 高性能の電子素子作成時に必要な超微細加工，極低温動作など極限技術の将来性，非常に複雑化したシステム構成，それを駆動するソフトウェアの煩雑さ，などを考えあわせれば，情報処理システムとしての電子計算機の飛躍的な能力向上はもはや限界に近づいたと考えることもできる。その結果，この演算能力限界の壁を乗越え得る新しい演算原理に基づいた情報処理システムへの要求が強まっている。その1つの候補として光学情報処理技術を活用した大容量情報処理システム，すなわち光コンピュータが非常に有力なものとしてクローズアップされてきた。[23-28]

現在の電子計算機の能力を制限している要因はVLSI(very large scale integrated cir-

cuit)内のゲート間，チップ間の信号通信能力の限界であり，この欠点を克服した構成を持つ大容量情報処理システムの開発が期待されている。[26,27,29] 電子計算機は膨大な数の電子回路素子により構成され，各素子は導体を用いた信号線で情報の授受を行っている。しかし，システムの高性能化に伴い，素子間の配線パターンは極度に複雑化しており，種々の問題が生じている。たとえば，信号線自体の通信容量の限界，von Neumann bottleneck, bus congestion, clock skew などである。これらは電子計算機のアーキテクチャから生じる問題であり，さらにつきつめると，電子を情報媒体とした情報処理技術自体から生じる基本的な問題でもある。それに対し，光を用いた並列信号伝送は非常に強力な通信能力を持ち，電子計算機内の信号通信で生じている様々な問題点を一掃し得る可能性を有している。たとえば，結像光学系や投影光学系は数十万から数百万の信号チャンネルをいとも簡単に実現し，しかもそれらはすべて非干渉である。光情報処理技術の本質は光の強力な信号伝送であり，この光の信号通信能力を有効に利用できれば電子計算機の能力限界をはるかに越えた情報処理システム —光コンピュータ— の構築も夢ではない。

それにもかかわらず，光コンピュータの具体的なシステム構築に関する研究は必ずしも活発には行われていなかった。これは，光コンピュータの持つ可能性，構想の自由度が余りにも大きすぎて，その手掛りをつかむことがかえって困難であったことが原因である。しかし，この問題は光コンピュータを既存の入出力装置，演算装置，記憶装置から成る電子計算機の概念に押込めようとするために生じたものである。したがって，種々の目的に適した様々な形式の光コンピュータがあると柔軟に考え，しかも目標設定さえ明確であれば，それらの開発研究は大きな意義を持ってくるものと思われる。その意味で，過去精力的に行われた光学情報処理分野での様々の研究は光コンピュータ開発の基礎研究として価値のあるものと考えられる。そして，その成果の有効な利用こそが光コンピュータ実現への近道と考えることもできる。

光コンピュータのシステム構成に大きな影響を与えるのは，如何なる演算方式を採用するかという点である。既に発表されている研究から，光学情報処理技術を利用した情報処理システムの演算方式を分類してみると次の3つの方式に分けることができる。[18]

1. 並列アナログ演算方式
2. 時系列ディジタル演算方式
3. 並列ディジタル演算方式

これらのうち，最も発展性があり，興味ある方式は3.の並列ディジタル演算方式である。なぜならば，並列ディジタル演算方式は光学情報処理技術の並列性と，ディジタル演算方式の柔軟性とをあわせ持つことが期待されるからである。さらに，この演算方式に基づくシステムは光学情報処理技術のみならず，今までの電子計算機の開発過程で蓄積されてきたディジタル演算方式に関するすべての資産を継承することができるからである。したがって，数ある光コンピュータの実現方式の中

で最も有望なものと考えられる。

著者は以上の点を留意し，新しい汎用並列光演算システムの構築を目的として，光学的に並列論理演算を行い得る並列光論理演算法を新しく考案した。そして，並列光論理演算法をもとにした並列画像処理法を考えると共に，アレイロジックの概念を導入して並列光近傍画素間論理演算法 — 光アレイロジック — の概念を確立した。さらに，光アレイロジックの概念を基盤として構築し得る汎用並列光演算システムOPALS(optical parallel array logic system)を提案し，種々の具体的なシステム構成に対する考察を行った。本論文はこれらの研究成果をまとめたものであり，以下に示す各章で構成している。

第1章では，新しく考案した並列光論理演算法の概要と並列2変数2値論理演算および並列シフト演算の実行方法を説明する。そして，多様な処理への対応が可能なように，並列光論理演算法の考え方を一般化する。

第2章では，並列光論理演算法の実行に必要な周辺技術として符号化法と動的2次元相関演算法を考案した。そして，それらを実行するために考案したいくつかの処理系について述べる。

第3章では，並列光論理演算法の応用として，並列画像処理を目的とした並列光演算法について述べる。そして，並列光論理演算法による並列光アナログ演算の可能性を検討する。

第4章では，並列光論理演算法とアレイロジックの類似点を積極的に利用した新しい演算原理，光アレイロジックについて説明し，実際の問題に適用した例を通してその有効性を示す。

第5章では，光アレイロジックをもとにした新しい汎用並列光演算システム，OPALSを提案し，その概念について説明する。そして，著者が考案した具体的なシステム構成法を2種類示し，その実現に必要な周辺技術を明確にする。

第6章では，OPALSのシステム構成の最適化を2つのアプローチから試みる。そして，OPALSの光コンピュータへの発展性を多面的に検討し，OPALSの新しい演算システム — 光コンピュータ — としての処理能力を評価する。

最後に，本研究の研究成果について総括し，今後の研究課題について述べる。

第１章　並列光論理演算法

1.1　緒　　言

光の有する高速・並列情報処理能力は非常に大きく，電気的システムでは実現困難な数々の大容量の演算を簡単に実行してしまう。[13-15] たとえば，２次元相関演算。これは２次元の広がりを持つ２つのデータ集合に対する膨大な数の積和演算であるが，光学的には簡単な投影系で実現できる。また，２次元フーリエ変換。これは光学におけるフラウンホーファー回折像そのものが演算結果を与える。このように光学情報処理技術の特長は大容量の２次元データに対する高速・並列処理である。しかし，一方，従来の光学的演算は精度が低い，処理に対する柔軟性が乏しい，さらに，実行可能な演算の種類が少ない等の点で問題があった。[30] これらの問題点は基本的な光演算がアナログ演算方式であることに起因している。電子計算機による演算がアナログ方式からディジタル方式に変化した経緯を考慮すれば，光学情報処理による演算にもディジタル演算の概念を持込むことは極めて自然である。

ディジタル演算方式の利点には，まず，処理の柔軟性の高さがあげられる。現在，電子計算機が広汎な分野に普及している理由は，幅広い問題に対応できるディジタル演算方式の柔軟性によるものである。次に，ディジタル演算方式の利点として精度の高さがあげられる。ディジタル演算方式ではビット数の増大により高い精度の演算を実行することができる。また，連続的に処理を実行する場合にも途中結果が離散的な値しか取り得ないため，信号は処理の各段で誤差補正され，その結果，誤差蓄積を排除することができる。さらに，ディジタル演算方式では離散的準位の信号を取扱うため，光学的非線形処理に不可欠な光機能素子に対する要求も，アナログ処理用のものより緩やかとなる。[26]

ディジタル演算方式とは情報を離散的準位の信号として表わし，それらに対する操作により各種の処理を行うものである。現在，用いられている大部分のディジタル演算処理系は０，１の２値で情報を表現する２値論理に基づくもので，ブール代数を基盤としている。[31] 被処理データは論理変数と呼ばれる０，１の２値だけを取り得る変数，あるいはその組として表現され，処理はそれらの論理変数に対する論理関数の組合せにより実行される。したがって，ディジタル処理系を構成するためには，まず，論理関数（論理演算とも呼ぶ）を実現する方法を確立することが不可欠である。

光学的に論理演算を実行する方法は既にいくつか提案されている。[26,27,32-36] それらのうち光の並列処理能力を生かしていない形式のものを除外すると，残りは大別して２つのグループに分類することができる。１つは非線形機能素子の機能を利用するもの［タイプ１と呼ぶ］であり，他は処理データに対する非線形操作と従来からある光学技術とを併用するもの［タイプ２と呼ぶ］である。

タイプ1の例としては液晶ライトバルブ（LCLV：liquid crystal light valve）[32]やPROM素子（Pockels readout optical modulator）[33]をはじめとする空間光変調素子，活性ガラスの微小ファイバーを束にしたファイバーレーザープレート[34]を利用した方式のものなどがあげられる。空間光変調素子は何らかの光学的媒質を介して，2次元光波信号（書込み光）で別の2次元光波信号（読出し光）を変調する機能素子である。LCLVやPROMでは光の偏光状態を変調し，偏光子，検光子の組合せによって強度変調信号を得ている。素子の非線形応答特性を利用すればいくつかの論理演算が実現でき，さらに素子を直列接続するとより多くの種類の論理演算が行える。[35] ファイバーレーザープレートは微小レーザーの束であり，書込み光でポンピングし，レーザー発振の閾値特性を利用して論理演算を実行する。入力信号として2つの2次元光信号およびバイアス光を重ねて入射してANDやOR演算を実行する。これらタイプ1の光並列論理演算法は素子自体の演算機能を利用するため，素子の性能が処理能力を決定する。しかし，1つの素子が1つの並列論理演算要素（並列論理ゲート）に対応するため，まとまった単位の要素から成る処理系を構成するには多数の素子が必要となる。したがって，現時点でのこれらの素子の製作技術やコストを考えれば実用的システムの基本要素としての利用には問題点が多い。

一方，タイプ2の論理演算法は基本的には既存の光学技術を有効に利用しようという思想に基づくものである。例としてθ変調法[36]や著者らの考案した投影系を用いた並列光論理演算法等があげられる。[1-3] θ変調法では入力データを画素ごとに分解し，各画素をその画素値に従った方向・周期を持つ格子として変調する。その後，変調画素を空間フィルタリング処理して各種の論理演算結果を得る。並列光論理演算法は画素値を特定の空間符号パターンに変換し，アレイ状の点光源列により符号化画像を多重投影して論理演算結果を得る方法である。これらタイプ2の論理演算法は被処理データの変調や符号化等の非線形操作を必要とするが，原理は非常に簡単であり，光の持つ並列情報処理能力を効果的に利用することができる。また，多数の論理ゲートを必要とする場合もタイプ1の方法とくらべ安価に処理系を構成することができる。しかし，連続してデータを処理する場合には動的な変調法や符号化法が要求され，結果的にはタイプ1の方法で必要であった非線形機能素子が必要となる。それにもかかわらず，タイプ2の論理演算法は光学情報処理の能力を十分に活用できる点，および必要とする非線形機能素子の数が少なくてよいという点で，より望ましい方法であると思われる。

本章では，著者らの考案した並列光論理演算法の原理を説明し，その演算原理を一般化することにより並列論理演算のみならず多様な並列演算処理に対応できるものとする。本演算法はタイプ2の論理演算法の特長を生かすことにより，新しいディジタル光演算の基礎技術となり得るものである。以下，1.2節では，投影系を用いた並列光論理演算法の概要とディジタル演算の基礎となる2変数2値論理演算および並列シフト演算の方法を述べる。1.3節では，並列光論理演算法の数式化

を行い，並列光論理演算法が多様な処理に利用できることを示す。

1.2　並列2変数2値論理演算

本研究では投影光学系を用いた新しい並列光論理演算法を考案した。これは処理画像データの符号化と点光源アレイによる多重投影技術を応用したもので，従来の電子的なスイッチング法を基にした方法やタイプ1の方法にはない種々の機能上の特徴を持つ。そこで，まず，並列光論理演算法の概要と特徴をまとめ，その後に，実行方法の手順を説明する。さらに，この方法の検証実験の結果を示し，被処理データの高密度化に対する考察を行う。

1.2.1　並列光論理演算法の概要と特徴

並列光論理演算法は並列論理ゲートを光学的に実現する技術である。この並列論理ゲートは本論文を通じての重要概念であるため，まずその概要を説明する(Fig.1.1)。被処理データはN×N（Nは自然数）の微小正方領域（画素と呼ぶ）で構成され，それぞれが2値（0または1）の情報を持つものとする。これを2値離散画像と呼ぶ。今，この画像をAとし，その(i, j)画素をa_{ij}で表わすとa_{ij}は論理変数とみなすことができる。したがって，画像Aは論理変数群a_{ij}（$i, j=1, \cdots, N$）の集合体と考えることができる。また，別に画像Bを考え，その(i, j)画素をb_{ij}とするとこれも論理変数群b_{ij}（$i, j=1, \cdots, N$）の集合体と考えられる。並列論理ゲートはこの2つの画像A，B内の同一位置にある対応画素同志（a_{ij}とb_{ij}）を入力変数とし，各画素ごとに演算の出力値c_{ij}を出力する。各画素間で実行される演算はすべて同じものである。したがって，出力値c_{ij}の集合体は1つの画像データCを形成する。この種の並列演算はSIMD(single instruction multi-data flow)形式と呼ばれ，shift-invariantな光学系の特長を生かせる

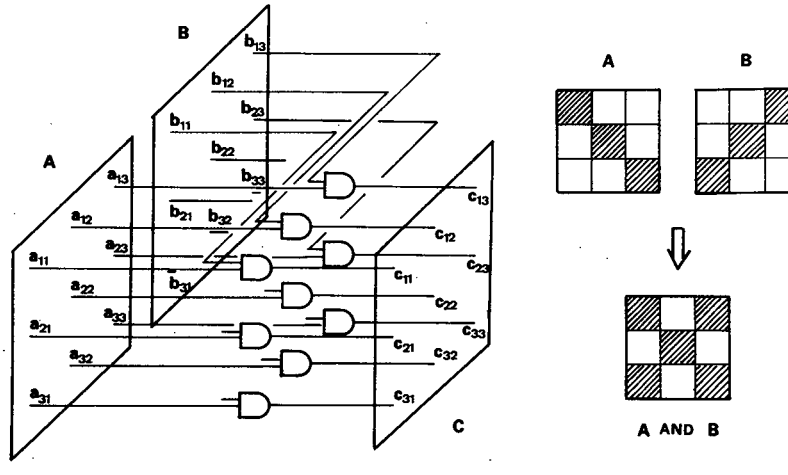


Fig.1.1　並列論理ゲート

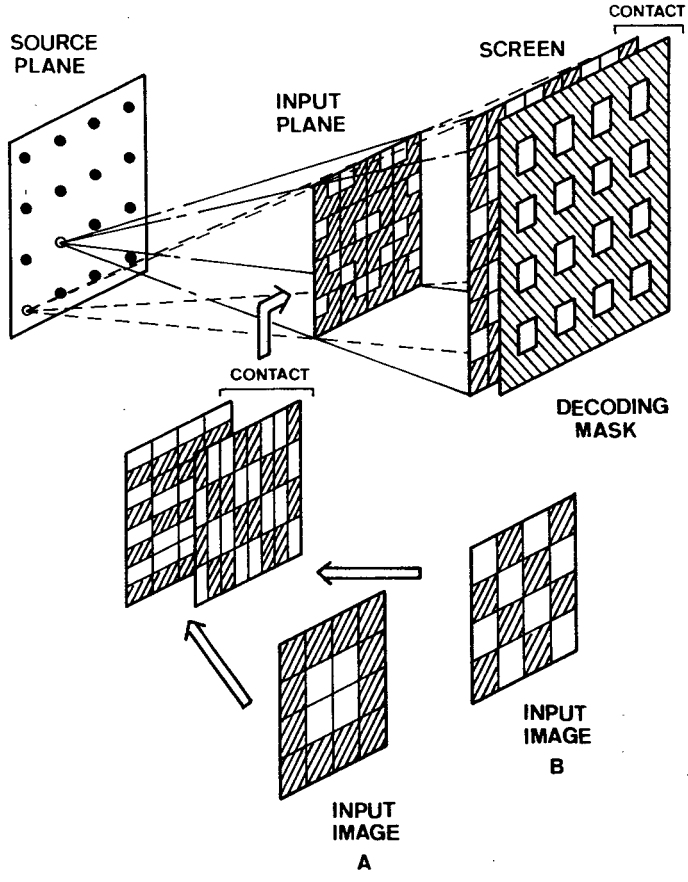


Fig.1.2　投影系による並列光論理演算法

ことから，ディジタル系演算系における有望な演算形式であると言える。以上の演算を数式で表現すると次のようになる。

$$c_{ij}=f(a_{ij},\ b_{ij}).\quad (i,\ j=1,\cdots,\mathrm{N}) \tag{1-1}$$

ここで，$f(a,\ b)$ は $a,\ b$ を入力変数とする２値論理関数を示す。

Fig.1.2 が投影光学系を用いた並列光論理演算法の概念図である。入力は２つの２値離散画像であり，それらを符号化した後，点光源アレイで照明する。その結果，スクリーン上に符号化画像の多重投影像が得られる。これを復号マスクで復号すれば，光の明暗信号の集合として並列論理演算の結果を得る。

並列光論理演算法の特徴を次に示す。

1. 使用する処理系は非常に簡単であり，性能に対する価格比が低い。
2. ２つの２値画像に対する全論理演算を並列的に実行可能。
3. プログラム可能であり，任意の組合せ論理を簡単に実行できる。

4. 点光源アレイの点滅状態の組合せにより論理関数の選択ができる。したがって，電気的・機械的走査は不要。

5. 各ゲートはSIMD形式の並列動作を行う。

6. 入力２値画像の空間的な符号化が必要であるが，復号は簡単な復号マスクにより行える。

1.2.2 入力画像の符号化

並列光論理演算法を実行するには，まず，入力画像の符号化を行う。入力画像Ａ，Ｂは入力変数 a_{ij} および b_{ij} の集合体で，各入力変数（すなわち画素）ごとに値（０または１）を有している。そこで，まず，それらの情報を光学処理系で取扱い可能な形に変換する。その変換には種々の方法が考えられるが，光学情報処理の並列性を生かすためにFig.1.3に示す符号パターンを用いる。符号パターンは透明部分と不透明部分で構成され，入力画像ＡまたはＢの種類および画素値によって異なる。すなわち，画像Ａ内の各画素はその値に応じてFig.1.3(a)の符号パターンで置換し，また画像Ｂ内の各画素はFig.1.3(b)の符号パターンで置換する。この操作を画像Ａ，Ｂ中の全画素にわたって行う。

次に，符号パターンにより構成された２つの画像を対応画素（たとえば a_{ij} と b_{ij})が同じ位置にくるように重ね合わせる。その結果，各画素位置には a_{ij}，b_{ij} の値の組合わせにしたがった4種類の符号の内の１つが得られる（Fig.1.3(c)）。これらの符号は画素区画の1/4大の窓を持つ符号であり，その窓の空間的位置によって a_{ij}，b_{ij} の値の組合せた情報を表わす。Fig.1.4は画像Ａ，Ｂに対する符号パターン変換の手順を示したものである。もとの入力画像から重ね合わせに

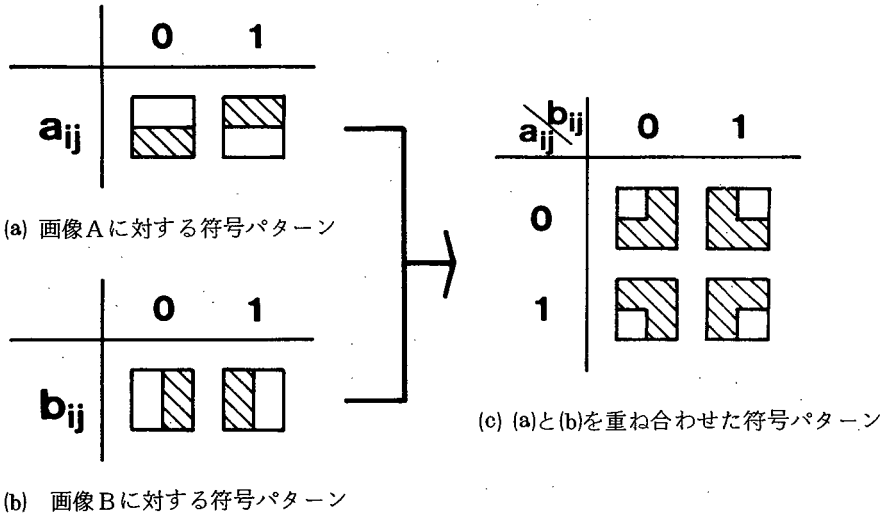


Fig.1.3 符号パターン

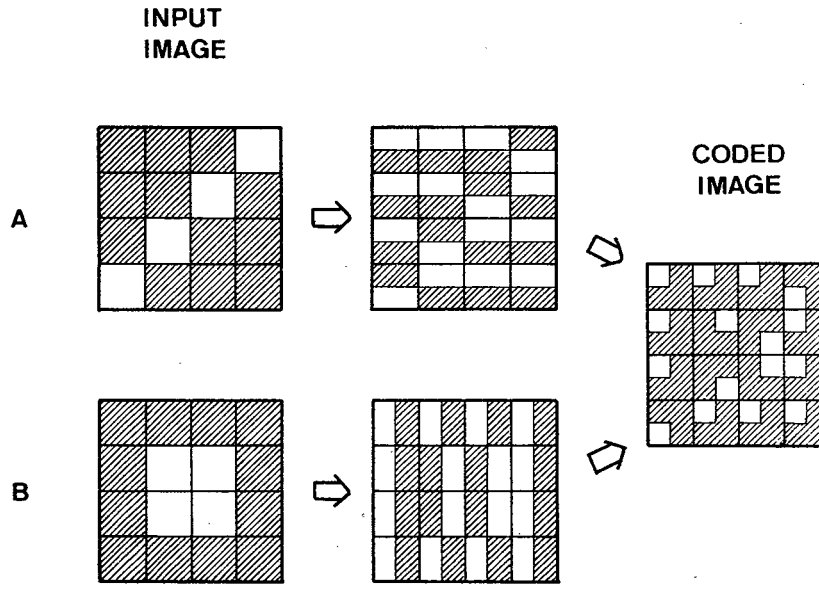


Fig.1.4　画像に対する符号化

よる最終パターンを得るまでの一連の操作を符号化と呼び，得られたパターンを符号化画像と呼ぶ。

この符号化操作をディジタル回路の機能と対応させると，次のような解釈が成立つ。符号パターンを1/4画素大の4つのサブパターンの集合体と考え，透過状態が1，不透明状態が0に対応するとすれば，変換操作はFig.1.5に示す電子回路の2ビットデコーダの処理と等価となる。[37] したがって，符号パターンは2ビットデコーダの出力信号を光学的情報（透明，不透明）として空間的に配置したものと考えることができる。2ビットデコーダの各出力信号は符号パターン内の各サブパターンに対応する。

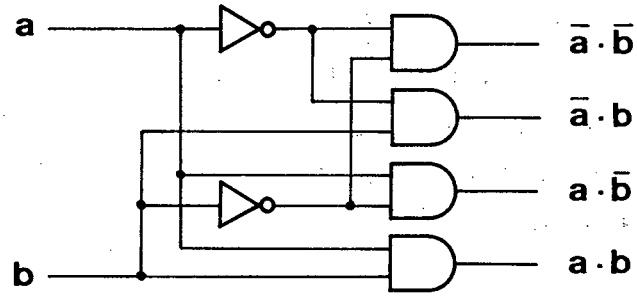
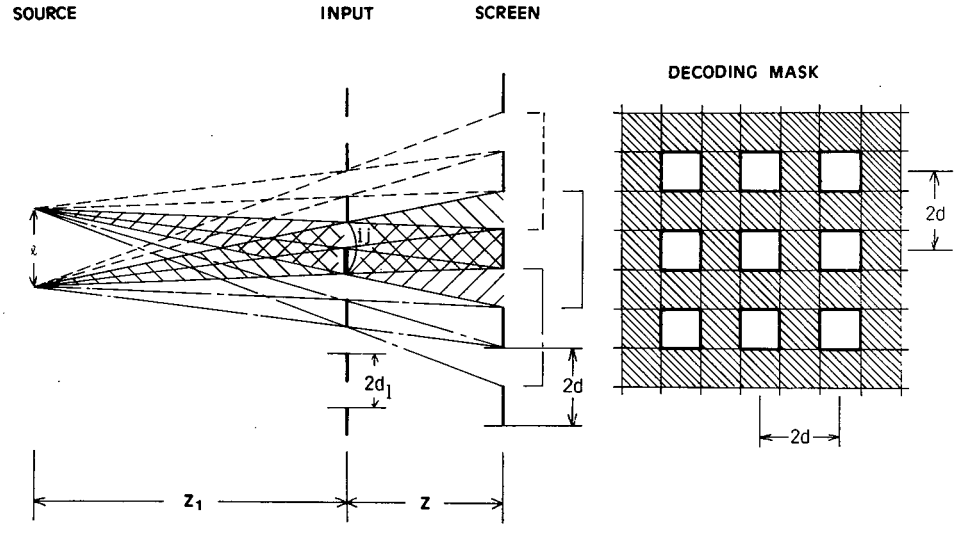


Fig.1.5　2ビットデコーダ

1.2.3　多重投影光学系

符号化画像をFig.1.2に示す多重投影光学系の入力面に置く。多重投影光学系は点光源アレイ，入力透過画像，スクリーン，復号マスクにより構成される。Fig.1.6，Fig.1.7は各部分の位置関係を示す。Fig.1.6は系を真横から見た図，Fig.1.7は特定の画素に対応する符号パターンの投影の様子を示した図である。



(a) 真横より見た図　　(b) 復号マスク

Fig.1.6　多重投影光学系

すべての点光源が同時に点燈すると符号パターンの影がスクリーン上に4重に重なって投影される。この重なりの状態は点光源間隔 l，光源面－入力面間隔 z_1，入力面－スクリーン面間隔 z，符号パターンの大きさ $2d_1$ に応じて変化するが，特に，

$$\frac{z}{z_1} = \frac{d_1}{l-d_1} \qquad (l > d_1) \qquad (1-2)$$

の場合には，Fig.1.7に示すように4つの点光源による符号パターンの投影像は，垂直・水平方向共ちょうど半画素ずれて重なる。多重投影光学系は（1－2）式の関係を満足するように設定されたものを用いるものとする。なお，（1－2）式は z と z_1 の比を規定するもので，実際の系を設計するには点光源の指向特性，回折現象を考慮する必要がある。入力画像の高密度化に対しては1.2.7項で考察する。

多重投影光学系を用いると符号化パターン内のサブパターンはスクリーン上の9区画に投影される。ここで，中央区画（太枠内）には4つのサブパターンが重なって投影されることに着目す

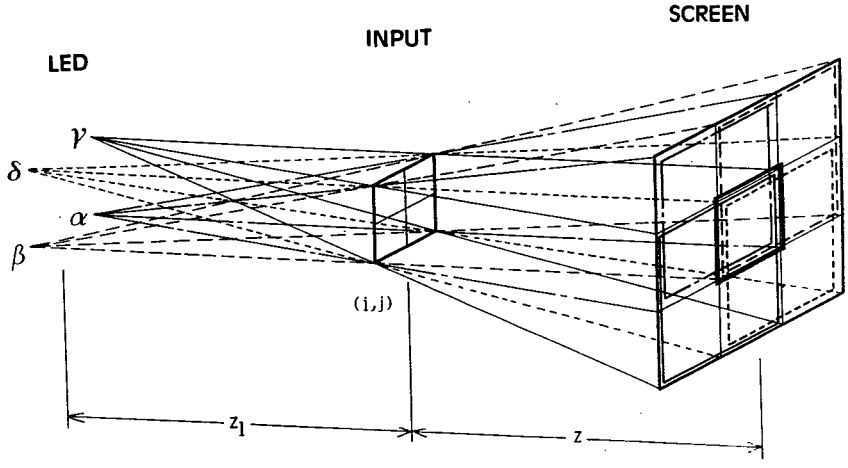


Fig.1.7　符号パターンの多重投影像

る。この時の区画の大きさ d は次式で与えられる。

$$d = \frac{d_1 l}{l - d_1} \cdot \qquad (l > d_1) \qquad (1-3)$$

Fig.1.7 は単一画素に対する投影の様子を示すが，他の符号化画像すべてについても同様な投影が行われる。重要なことは，各画素の符号パターン内の 4 つのサブパターンの重なる領域が画素ごとに存在し，しかも互いに干渉しあわないことである。並列光論理演算法ではこの領域内の信号のみが必要であり，その信号を取出すためにFig.1.6 に示す復号マスクをスクリーンに重ね，多重投影像を観測する。

1.2.4　並列論理演算の実現

2 値論理における基本的演算はTable 1.1 に示す 2 変数 2 値論理演算である。[31] これは入力変数 a, b の値の組合せに対し 0 か 1 の出力値を出す演算で，全部で 16 種類あり複雑な論理回路を構成する上での基本要素となる。

Fig.1.8 は多重投影系による論理演算実行の説明図である。点光源 β と γ が点燈している場合を考える。図は (i, j) 画素に関係した部分のみを抜出したもので，入力変数 a_{ij}, b_{ij} の値の組合せによる 4 通りの符号パターンに対する多重投影像の変化を示す。今，多重投影像において中央区画部分（太枠内）の投影像に注目すると，a_{ij}, b_{ij} の値の組合せにより暗あるいは明のいずれかの信号が得られていることがわかる。そこで，この出力信号強度に対し論理値 0 または 1 を対応させると論理ゲートが実現できる。光信号と論理値の対応には Table 1.2 に示すように 2 通りが考えられる。各論理体系を bright-true 論理および dark-true 論理と呼ぶ。Fig.1.8 の例に対し bright-true 論理を用いると出力値は入力変数 a_{ij}, b_{ij} の値が相異なる時にのみ 1 となり，

Table 1.1　2変数2値論理演算

INPUT		OUTPUT							
a	b	0	ab	$a\bar{b}$	a	$\bar{a}b$	b	$a \oplus b$	a+b
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
			AND					XOR	OR

$\bar{a}\bar{b}$	$\overline{a \oplus b}$	$\bar{b}$	$a+\bar{b}$	$\bar{a}$	$\bar{a}+b$	$\bar{a}+\bar{b}$	1
1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
NOR	EQU					NAND	

これはTable 1.1よりXOR演算の結果と一致する。

Fig.1.9は4つの点光源の点滅状態のすべての組合せに対する投影像の変化をまとめたものである。各列はある点光源の点滅パターン（光源パターンと呼ぶ）による入力信号の組合せに依存した4通りの符号パターンの投影像を示す。投影像の下の数字はbright-true論理による出力値である。これらの結果とTable 1.1の比較より，光源パターンの選択で，すべての2変数2値論理関数が実現できることがわかる。

以上の演算の原理は光インターコネクション技術の応用として説明することができる。1.2.2項でも述べたように並列光演算法で用いる符号パターンは2ビットデコーダの出力信号に対応するサブパターンの集まりと考えられる。これらのサブパターンは各点光源により投影像の中央区画へ投影されるが，ここで，サブパターンと点光源とは1対1に対応している。すなわち $a_{ij}\, b_{ij}$

Table 1.2　光信号と論理値の対応

OPTICAL SIGNAL	LOGICAL VALUE IN BRIGHT-TRUE LOGIC	LOGICAL VALUE IN DARK-TRUE LOGIC
BRIGHT	1	0
DARK	0	1

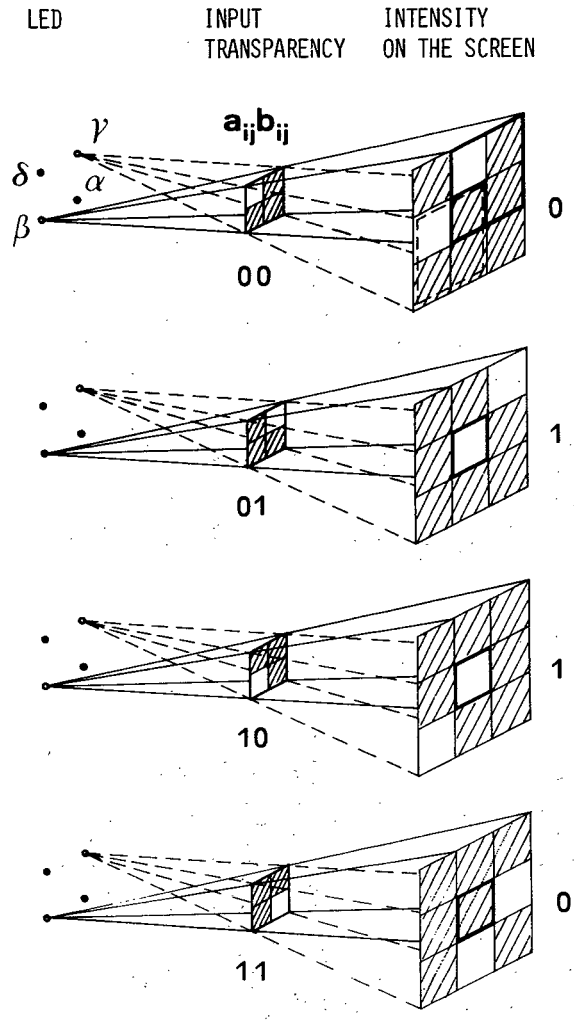


Fig.1.8　XOR演算の実行

を表わすサブパターンの信号は点光源αにより中央区画へ送られ，$a_{ij}\,\overline{b}_{ij}$は点光源$\beta$，$\overline{a}_{ij}\,b_{ij}$は点光源$\gamma$，$\overline{a}_{ij}\,\overline{b}_{ij}$は点光源$\delta$で送られる。これは，Fig.1.10に示すように，サブパターンの情報が各点光源の作る信号伝送路（光線）により関数出力位置へ送られていると考えることができる。各信号伝送路で送られる信号はインコヒーレント光を用いるかぎり互いに非干渉であり，独立に制御可能である。したがって，中央区画に到達する信号は次式で表現することができる。

$$c_{ij}=\alpha\,a_{ij}\,b_{ij}+\beta\,a_{ij}\,\overline{b}_{ij}+\gamma\,\overline{a}_{ij}\,b_{ij}+\delta\,\overline{a}_{ij}\,\overline{b}_{ij}\,. \qquad (1-4)$$

ここで，α, β, γ, δはそれぞれ点光源α, β, γ, δの点燈状態を示す論理変数である。α, β, γ, δの値の組合せで任意の2変数2値論理関数が実現できることがわかる。特に注意すべきこと

は，光波伝播の並列性より，Fig.1.10と同様の信号伝送が符号化画像中の全符号パターンに対しても並列・独立に行われることである。その結果，並列論理演算が実行でき，しかも点光源の制御により画像中の全画素に対する演算を簡単に選択することができる。

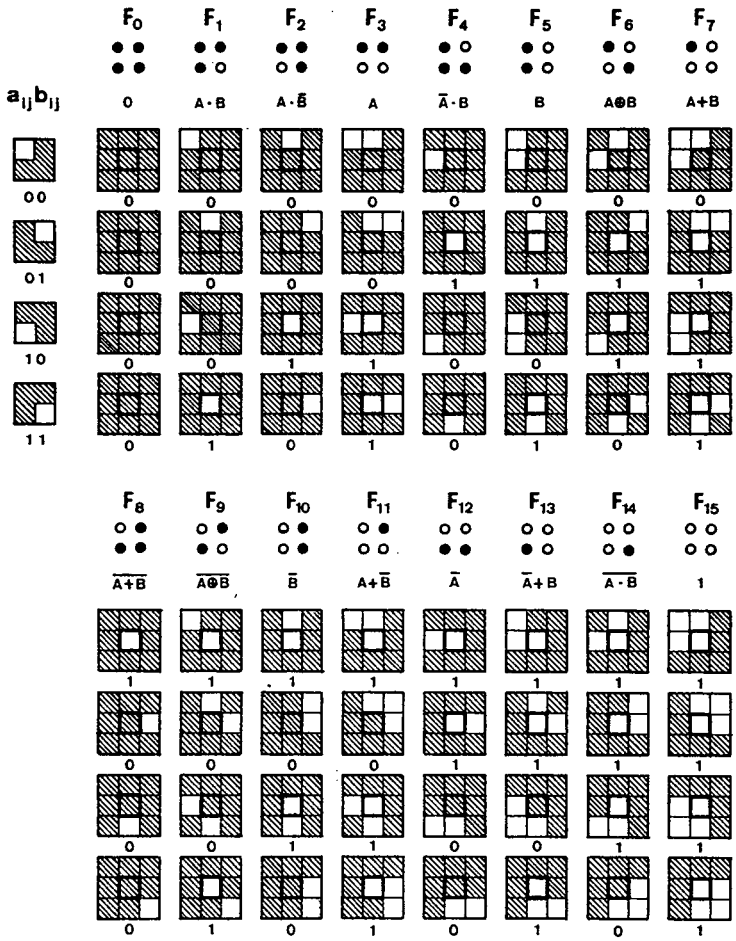


Fig.1.9　光源パターンと多重投影像の関係

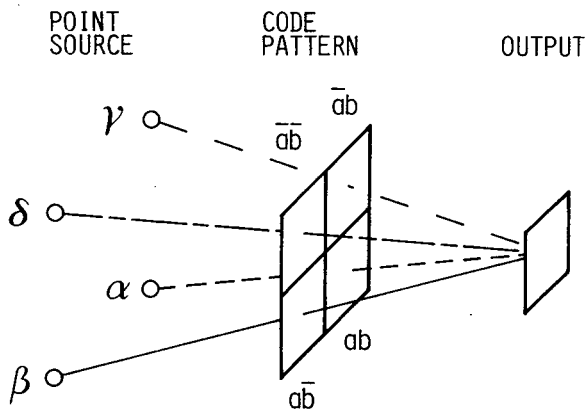


Fig.1.10　光インターコネクション

1.2.5 並列シフト演算

ディジタル演算では論理演算と共にシフト演算が重要な役割をはたす。このことは2次元データの並列処理でも同様であり，並列シフト演算が重要な役割を持つ。シフト演算は2次元データ全体をある画素分だけ上下・左右に平行移動させる操作で，その結果，異なった位置の画素間（たとえば a_{ij} と $b_{i,j-1}$）の演算が可能となり，処理の自由度が広がる。

並列光論理演算法では点光源の増設によりシフト演算を容易に実現することができる。Fig.1.11にその光学系を示す。光源面の左半分（グループ1）の4光源が符号化入力に対して論理演算を実行し，結果を出力面の (i, j) 画素に出力する。ここで，光源面の右側に別の点光源（グループ2）を増設する。この光源も符号化入力に対し論理演算を実行するが，結果を $(i, j-1)$ 画素，すなわち j 方向に -1 画素シフトした位置に出力する。この演算はシフト演算と2変数2値論理演算を組合せたもので，並列光論理演算法の特徴的な演算の1つである。シフトの方向とシフト量は点光源グループの位置により決定される。

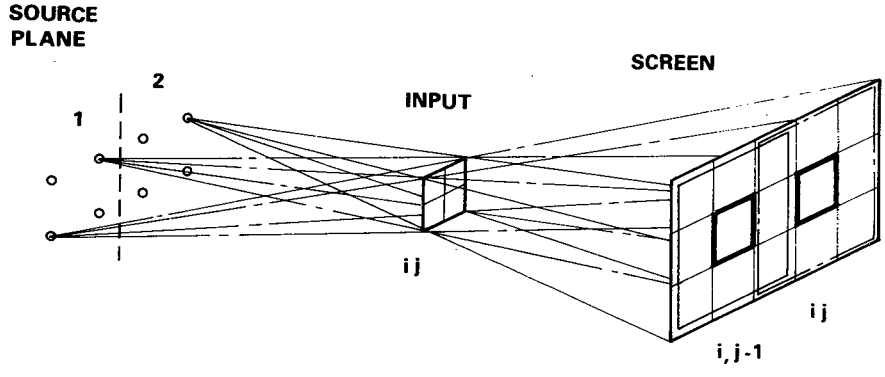
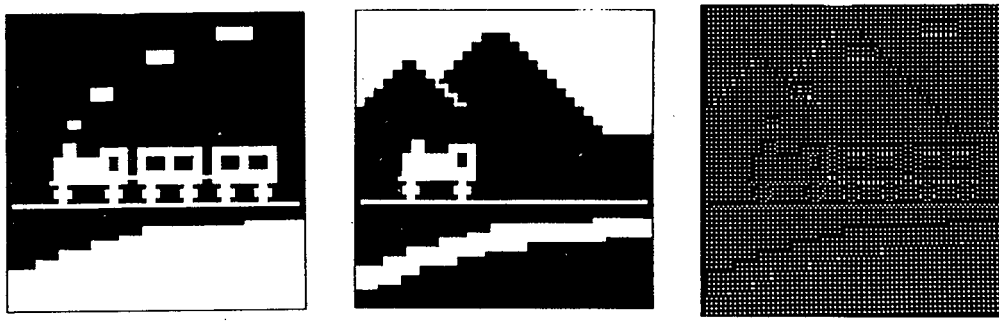


Fig.1.11　並列シフト演算の原理

1.2.6 実験結果

ここでは投影光学系を用いた並列光論理演算法による実験結果を示す。Fig.1.12(a), (b)はそれぞれ64×64の画素構成を持つ2値入力画像である。Fig.1.12(c)が符号化画像であり，これを18㎜×18㎜（$d_1 = 0.14$ ㎜）の大きさでリスフィルムに記録して入力像として用いた。使用した光学系は光源間隔 $l = 5.08$ ㎜，光源面－入力面間隔 $z_1 = 33$ ㎜で，符号化画像とスクリーンは厚さ1.7㎜のスライドグラスを間にはさんで固定し，復号マスクはスクリーンと密着させた。点光源には東芝製の黄色LED TLY-108を用い，パーソナルコンピュータPC-9801F2(日本電気製）により点滅を制御した。Fig.1.13が全2変数2値論理演算の実験結果である。各出力画像はLEDの点滅パターンを変えて得たものである。この処理系の場合，4096チャンネルの論理演算を並列に実行したことになる。

Fig.1.14はシフト演算の実験結果である。(a), (b)が入力画像で，16×16の画素構成を持つ。(c)～(f)が出力画像でそれぞれ1画素および2画素左シフトした位置に所定の論理演算の結果を出力している。



(a) INPUT A　　(b) INPUT B　　(c) CODED INPUT

Fig.1.12　入力画像と符号化画像

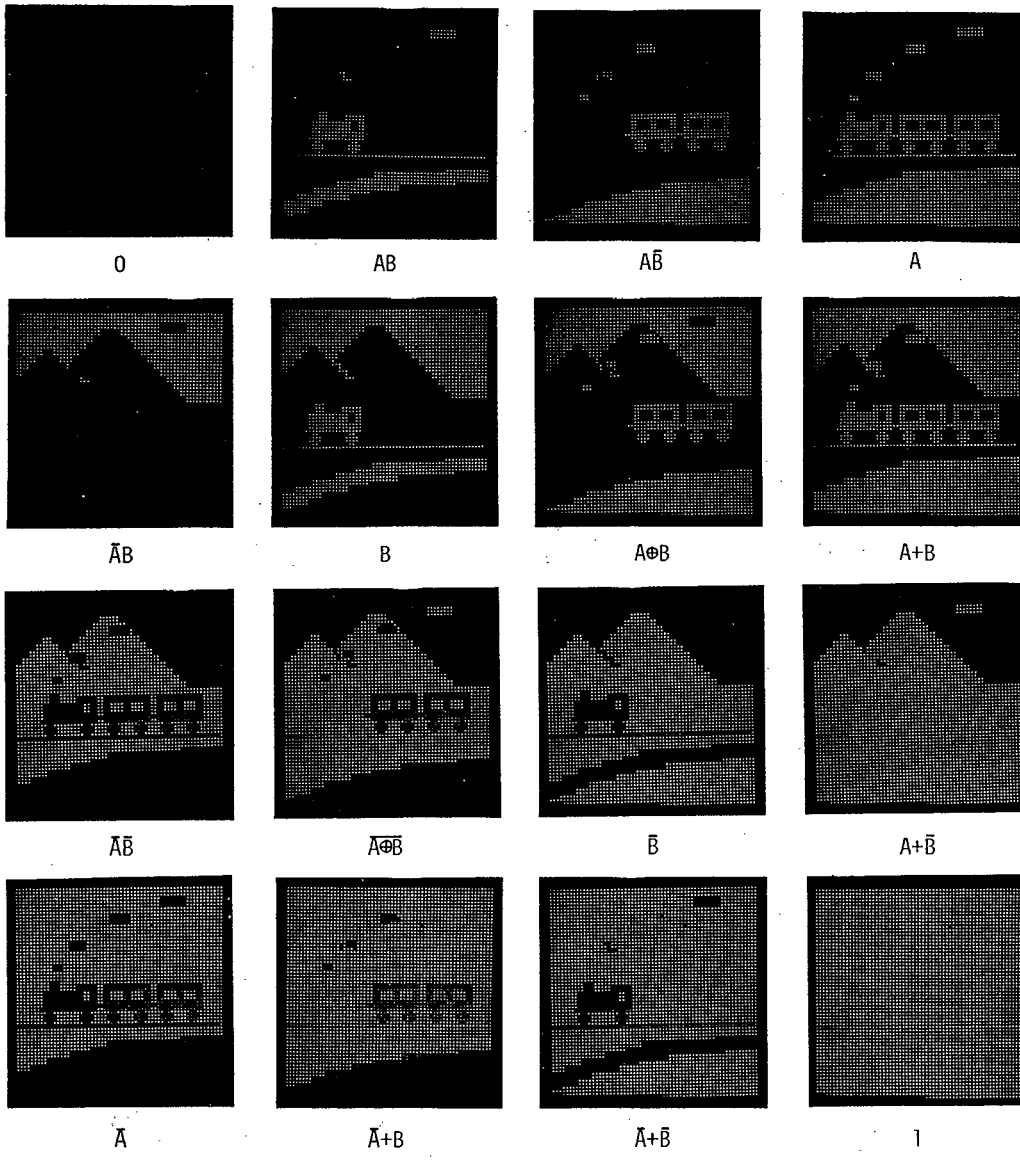
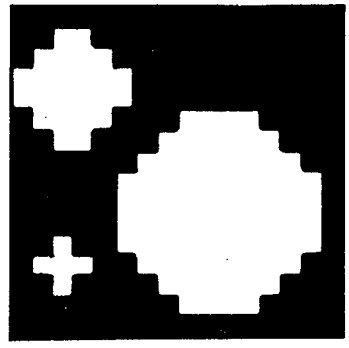
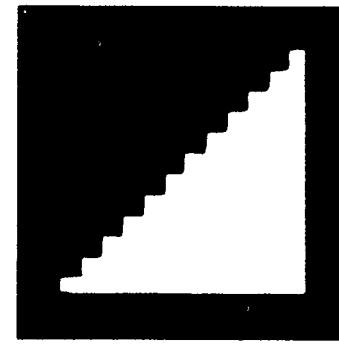


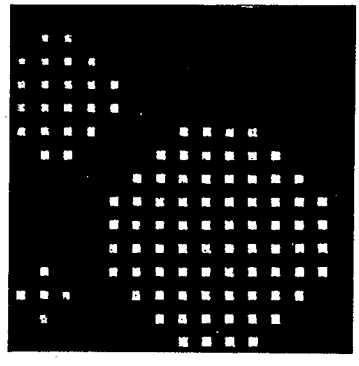
Fig.1.13　2変数2値論理演算の実験結果



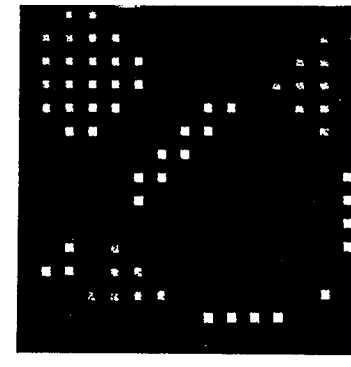
(a) INPUT A



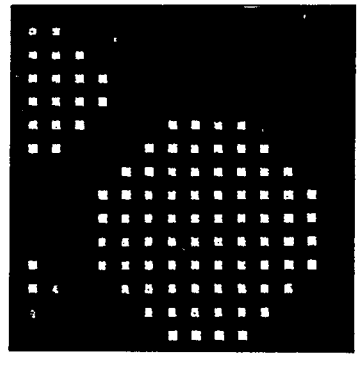
(b) INPUT B



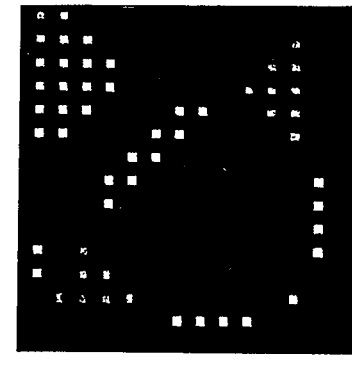
(c) A + SHIFT 1



(d) XOR + SHIFT 1



(e) A + SHIFT 2



(f) XOR + SHIFT 2

Fig.1.14 並列シフト演算の実験結果

1.2.7 高密度化に対する考察

並列論理ゲートの演算能力は同時に実行できる演算チャンネルの数（入力画像の画素数）に大きく依存する。並列論理ゲート全体の２次元的拡がりには限界があることを考えあわせると，論理ゲート数を増やすには各論理要素（画素）の高密度化が必要となる。しかし，投影系による並列光論理演算の基本原理は幾何光学に基づくため，高密度化を行う場合には回折の影響を考慮しなければならない。そこで，parageometrical 光学の手法[38]により回折計算を行い，幾何光学近似の成立する範囲を求めた。

Fig.1.15 に示すように，$z=z_1$ にある幅 d_1 のスリットが，$(x_p, 0)$ にある単色光源からの発散光で照明されている場合を考える。この時，$x=x$ での振幅分布 $u(x, z)$ は Huygens-Fresnel-kirchhoff の２乗近似式を用いて

$$u(x, z)=\int_{-\frac{d_1}{2}}^{\frac{d_1}{2}} \exp\left\{ \frac{i\pi}{\lambda}\left[\frac{(x_1-x_p)^2}{z_1}+\frac{(x_1-x)^2}{z}\right]\right\} dx_1 \qquad (1-5)$$

と表わせる。ここで，λ は光源の波長である。

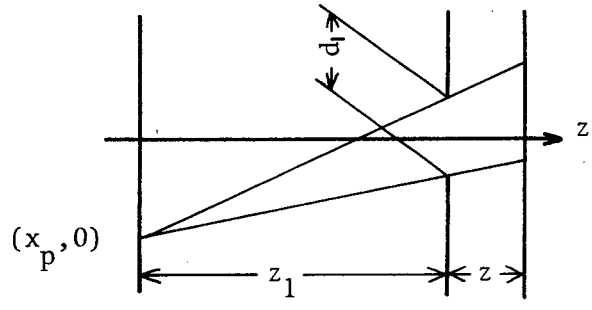


Fig.1.15　スリットによる回折

（1−5）式を停留位相原理[38]を用いて評価し，幾何光学近似の成立範囲を求めると次式を得る。

$$4\lambda < z \ll \frac{d_1^2}{\lambda}\left(\frac{z+z_1}{z_1}\right)^3. \qquad (1-6)$$

さらに（1−2）式を用いて

$$4\lambda < z \ll \frac{d_1^2}{\lambda}\frac{l^3}{(l-d_1)^3}. \qquad (1-7)$$

ここで一般に

$$l \gg d_1 \qquad (1-8)$$

と設定できるため（1−7）式は

$$4\lambda < z \ll \frac{d_1^2}{\lambda} \qquad (1-9)$$

となる。

（1−9）式より，d_1 の変化に対する z のとるべき値と20㎜×20㎜ 領域内の解像点を評価した結果をTable 1.3に示す。2重不等号の成立範囲は10倍以上として計算した。

Table 1.3　解像点数の評価

d_1	d_1^2/λ	z $(=d_1^2/10\lambda)$	resolution points in 20 mm × 20 mm
5 μm	0.05 mm	0.005 mm	2000 × 2000
10	0.2	0.02	1000 × 1000
25	1.25	0.125	400 × 400
50	5	0.5	200 × 200
100	20	2.0	100 × 100
250	125	12.5	40 × 40

d_1: slit width;　λ: wavelength of light

1.3　並列光論理演算法の一般化

1.2節においては投影系を用いた並列光論理法の概要を説明したが，この論理演算法は任意の相関光学系によっても実現できる。そこで，本章では並列光論理演算法の演算手順を数式化し，それを実行する光学処理系（相関光学系）が持つべき演算機能を整理しておく。

投影法による並列光論理演算法の実行手順をまとめると，次のようになる。

1. 入力画像の符号化
2. 点光源アレイによる多重投影
3. 復号マスクによる復号

ここで，2.の多重投影が投影光学系による特殊な処理であるが，この処理は別の見方をすれば符号化画像と点光源アレイの空間的なパターン（光源パターン）との2次元相関演算に他ならない。また，投影系では倍率の変化を併なうが本質的なものではない。これらのことを考慮して並列光論理演算法を数式化する。

まず，入力画像A，Bを次式で表わす。

$$\mathrm{A}: \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\ \mathrm{rect}\left(\frac{x-2id}{2d},\ \frac{y-2jd}{2d}\right) \qquad (1-10)$$

$$\mathrm{B}: \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij}\ \mathrm{rect}\left(\frac{x-2id}{2d},\ \frac{y-2jd}{2d}\right). \qquad (1-11)$$

ここで，a_{ij}, b_{ij} は0または1の値を持つ論理変数，2d は画素の大きさ，N^2は画素数である。また，rect 関数は次式で定義される。

$$\mathrm{rect}(x,\ y)=\begin{cases}1 & (|x|\leqq\frac{1}{2}\ ,\ |y|\leqq\frac{1}{2})\\ 0 & (\mathrm{otherwise}).\end{cases} \tag{1-12}$$

入力画像A，Bから作られる符号化画像 $i(x, y)$ は次式で表わせる。

$$i(x,y)=\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{rect}\left[\frac{x-(a_{ij}+2i)d}{d}\ ,\ \frac{y-(b_{ij}+2j)d}{d}\right]. \tag{1-13}$$

この符号化画像に対し，光源パターン $f(x, y)$ との間で2次元相関演算を実行する。光源パターンを次式で表わす。

$$f(x,y)=\sum_{p=-2L}^{2L+1}\sum_{q=-2L}^{2L+1}w_{pq}\delta(x-pd,\ y-qd). \tag{1-14}$$

ここで，$(4L+2)^2$ は点光源の数で，w_{pq} は各点光源の点滅状態を表わす論理変数である。また，$\delta(x, y)$ はDirac のデルタ関数である。(1-13)，(1-14)式を用いると相関画像 $c(x, y)$ は次式で与えられる。

$$\begin{aligned}c(x,y)&=\iint_{-\infty}^{\infty}i(u,v)f(u-x,v-y)\,du\,dv\\&=\sum_{p=-2L}^{2L+1}\sum_{q=-2L}^{2L+1}w_{pq}\left\{\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{rect}\left[\frac{x-(a_{ij}+2i-p)d}{d}\ ,\ \frac{y-(b_{ij}+2j-q)d}{d}\right]\right\}.\end{aligned} \tag{1-15}$$

一方，相関画像に重ねる復号マスク $g(x, y)$ は

$$g(x,y)=\sum_{m=1}^{N}\sum_{n=1}^{N}\mathrm{rect}\left(\frac{x-2md}{d}\ ,\ \frac{y-2nd}{d}\right) \tag{1-16}$$

と表わせる。

(1-15)式と(1-16)式の比較より

$$m=i+\frac{1}{2}(a_{ij}-p) \tag{1-17}$$

$$n=j+\frac{1}{2}(b_{ij}-q). \tag{1-18}$$

ここで，

$$a_{ij}-p=-2k \qquad (k\in\mathrm{integer}) \tag{1-19}$$

$$b_{ij}-q=-2l \qquad (l\in\mathrm{integer}) \tag{1-20}$$

と置く。a_{ij}, b_{ij} が論理変数であることから，符号化画像内の rect 関数が復号マスクの窓の位置に重なるのは次の4つの場合である。

$$a_{ij}=0\ ,\quad b_{ij}=0\ \text{のとき}\quad p=2k\ ,\qquad q=2l \tag{1-21}$$

$$a_{ij}=0\ ,\quad b_{ij}=1\ \text{のとき}\quad p=2k\ ,\qquad q=2l+1 \tag{1-22}$$

$$a_{ij}=1\ ,\quad b_{ij}=0\ \text{のとき}\quad p=2k+1\ ,\quad q=2l \tag{1-23}$$

$$a_{ij}=1\ ,\quad b_{ij}=1\ \text{のとき}\quad p=2k+1\ ,\quad q=2l+1\ . \tag{1-24}$$

また，（1－17）～（1－20）式より

$$m = i - k \tag{1-25}$$

$$n = j - l\ . \tag{1-26}$$

これらの関係から出力画像 $o(x, y)$ を求め，整理すると次式が得られる。

$$\begin{aligned} o(x, y) &= c(x, y)\ g(x, y) \\ &= \sum_{m=1}^{N}\sum_{n=1}^{N}\Big[\sum_{k=-L}^{L}\sum_{l=-L}^{L}(w_{2k,2l}\,\overline{a}_{m+k,n+l}\,\overline{b}_{m+k,n+l} + w_{2k,2l+1}\,\overline{a}_{m+k,n+l}\,b_{m+k,n+l} \\ &\quad + w_{2k+1,2l}\,a_{m+k,n+l}\,\overline{b}_{m+k,n+l} + w_{2k+1,2l+1}\,a_{m+k,n+l}\,b_{m+k,n+l})\Big] \\ &\quad \times \mathrm{rect}\left(\frac{x-2md}{d}\ ,\ \frac{y-2nd}{d}\right). \end{aligned} \tag{1-27}$$

ただし

$$\overline{a}_{mn} = 1 - a_{mn} \tag{1-28}$$

$$\overline{b}_{mn} = 1 - b_{mn}\ . \tag{1-29}$$

この（1－27）式の各 m，n に対する係数が，各画素における論理関数の結果を与える。

たとえば，$k=0$，$l=0$ の場合，（1－27）式の m，n に対する係数部分は

$$w_{00}\overline{a}_{mn}\overline{b}_{mn} + w_{01}\overline{a}_{mn}b_{mn} + w_{10}a_{mn}\overline{b}_{mn} + w_{11}a_{mn}b_{mn} \tag{1-30}$$

となる。これは（1－4）式の右辺と等価で2変数2値論理関数の結果を与える。

また，$k=1$，$l=0$ の場合には

$$w_{20}\overline{a}_{m+1,n}\overline{b}_{m+1,n} + w_{21}\overline{a}_{m+1,n}b_{m+1,n} + w_{30}a_{m+1,n}\overline{b}_{m+1,n} + w_{31}a_{m+1,n}b_{m+1,n} \tag{1-31}$$

となり，出力面 (m, n) の位置に被処理画像の $(m+1, n)$ 画素の結果を出力するシフト論理演算を表わす。

以上の例から，2次元相関演算に用いる光源パターン（w_{pq}）の組合せにより種々の演算が実行できることがわかる。ここで，任意の相関光学系による並列光論理演算を記述するため，光源パターンをカーネルと呼ぶ。（1－27）式からわかるように，1組の (k, l) に対応する4つの $w_{2k,2l}$，$w_{2k,2l+1}$，$w_{2k+1,2l}$，$w_{2k+1,2l+1}$ は (m, n) 画素を中心とした近傍画素 $a_{m+k,n+l}$ と $b_{m+k,n+l}$ との間の論理演算の種類を決定する。(k, l) の値は (m, n) 画素を中心とした近傍領域内における演算対象画素の位置を示す。そこで，各 (k, l) に対する4つの w_{pq} をカーネルユニットと呼び，演算指定の基本単位と考える。Table 1.4 にカーネルユニットのパターンと対応する論理演算を示す。

Table 1.4　カーネルユニットと論理演算の対応表

KERNEL UNIT	FUNCTION	KERNEL UNIT	FUNCTION
	0		$\bar{a}\bar{b}$
	ab		$\overline{a \oplus b}$
	$a\bar{b}$		$\bar{b}$
	a		$a+\bar{b}$
	$\bar{a}b$		$\bar{a}$
	b		$\bar{a}+b$
	$a \oplus b$		$\bar{a}+\bar{b}$
	$a+b$		1

Fig.1.16に示すように，カーネルはカーネルユニットにより構成される。重要なことは，各カーネルユニットの指定する論理演算の集まりとして近傍領域内画素に対する論理演算が実現できることである。カーネル内でのカーネルユニットの位置と近傍領域内での演算対象画素の位置は1対1に対応する。また，カーネルユニットの数，$(2L+1)^2$ は近傍領域の大きさを決定する。

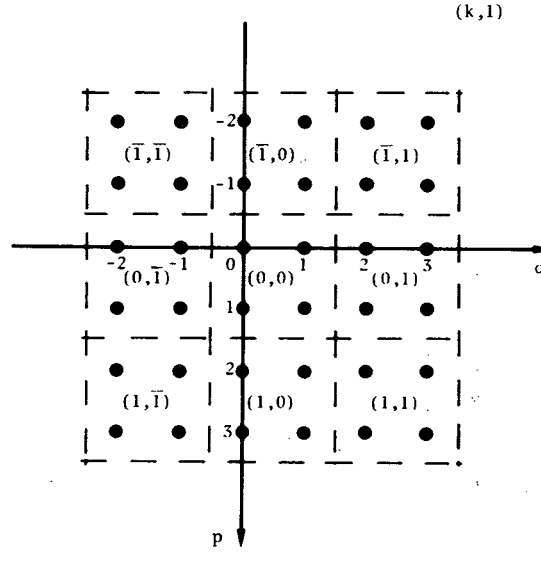


Fig.1.16　カーネル

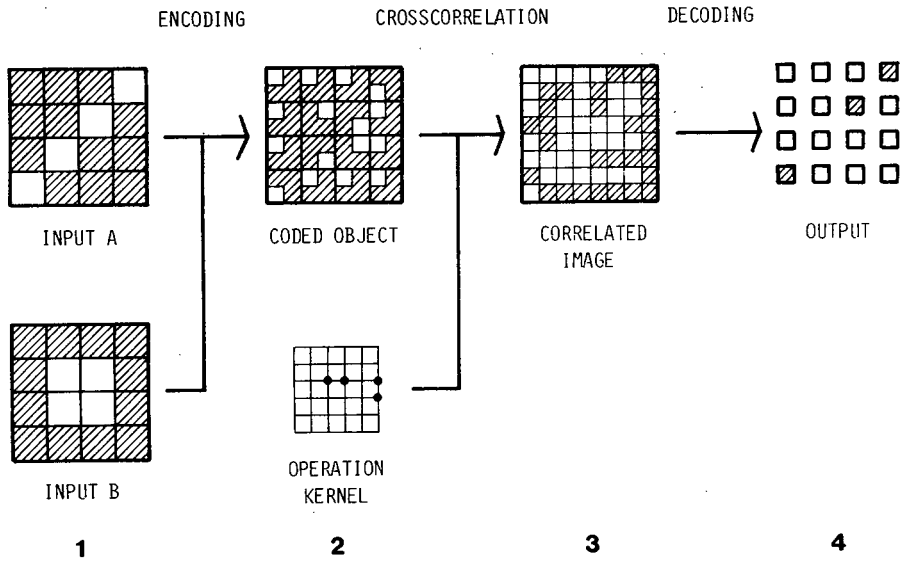
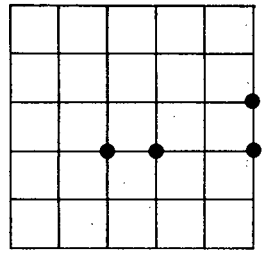


Fig.1.17　並列光論理演算法の実行手順

Fig.1.17は並列光論理演算法の手順を図示したものである。相関演算のカーネルは格子点上の点群として与えられる。本論文ではカーネルの表記法として，格子点パターンと行列表記の2つを使う。カーネルを構成する各点に強度分布がない場合には格子点パターンを，ある場合には行列表記を用いる。Fig.1.18に示す2つの表記は同一のカーネルを表わす。カーネルに強度分布を持たせることにより，アナログ処理を行うこともできる。この例については3章で説明する。

(1-27)式はbright-true論理に基づく論理演算の結果を与えているが，dark-true論理を用いると非常に興味ある結果を得ることができる。(1-27)式の係数部を論理変数とみなし，その否定を求め，De Morganの定理[31]を用いると次式を得る。



(a) 格子点パターン

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(b) 行列

Fig.1.18　カーネルの表記法

$$\prod_{k=-L}^{L}\prod_{l=-L}^{L}\{[\overline{w}_{2k,2l}+(a_{m+k,n+l}+b_{m+k,n+l})][\overline{w}_{2k,2l+1}+(a_{m+k,n+l}+\overline{b}_{m+k,n+l})]$$
$$\times[\overline{w}_{2k+1,2l}+(\overline{a}_{m+k,n+l}+b_{m+k,n+l})][\overline{w}_{2k+1,2l+1}+(\overline{a}_{m+k,n+l}+\overline{b}_{m+k,n+l})]\}. \tag{1-32}$$

これはエレクトロニクスの分野で研究されているアレイロジックの中で，2ビットデコーダを利用する系の論理表現と同一のものである。[39,40] そして，アレイロジックにおける議論より，(1-32)式中のすべての論理変数に対するAND演算を実現することができる。さらに，アレイロジックとの対応関係を積極的に応用すれば近傍領域内の画素に対する任意の並列論理演算も実行できる。この点については4章で詳しく議論する。

1.4 結　言

本章では，著者らの考案した並列光論理演算法の実行方法を説明し，実際の光学系による処理結果を示した。処理結果は非常に良好で，本並列光論理演算法の原理の確証と共にその処理能力に対する知見が得られた。また，回折理論を用いて論理ゲート密度の限界を評価した。1.3節では，並列光論理演算法の原理を数式で記述し，その一般化を試みた。その結果，演算の内容と光学的操作の関係を明らかにすることができ，並列光論理演算法が論理演算のみならず，アナログ処理や近傍画素を対象とした演算にも応用できることがわかった。

第2章　符号化法と実時間2次元相関演算法

2.1　緒　　言

1.3節で示したように，並列光論理演算法は3つの操作 ── 入力画像の符号化，符号化画像とカーネルとの2次元相関演算，相関画像に対する標本化 ── から構成される。相関画像の標本化は1.2節で用いた復号マスクを用いると簡単に行えるが，符号化操作と2次元相関演算を実行するには種々の工夫が必要である。なぜなら，並列光論理演算法を実用化するにはそれらを実時間で行わなければならないからである。そこで，著者は実時間で並列光論理演算法が実現できるいくつかの方法を考案した。これらの内で最終的に利用されるのは高速で高密度化可能な方法であろうが，その選択は将来開発される光機能素子等の性能によって左右され，現時点ではいずれの方法が最良であるかは判断できない。本章では，著者が考案した種々の符号化法，実時間2次元相関演算法の列挙にとどめる。そのいずれを選択するかは将来，利用可能になる素子の特性に依存することになろう。

2.2　符号化法

並列光論理演算法における符号化とは，1.2.2項で述べたように，入力画像データの符号パターンへの変換操作であり，2種類の符号化操作法が考えられる。1つの方法は，1つの入力画像の各画素をFig.1.3(a)あるいは(b)のいずれかの符号パターンで置換するもの［レベル1の符号化と呼ぶ］である。他の方法は，2つの入力画像の対応画素の組合せに応じてFig.1.3(c)の符号パターンを選び，それらより符号化画像を直接得る方法［レベル2の符号化と呼ぶ］である。レベル1の符号化は比較的簡単であるが，各入力画像に対する符号化像は光学系内で重ね合わせる必要があり，使用する相関光学系は複雑なものとなる。ここでは，まず，レベル1の符号化法を4種類示し，その内でレベル2の符号化が可能なものについてはその実現法を示す。

2.2.1　カラー符号化法

最も簡単な符号化法として，2つの異なった波長の光を透過させる色フィルタを用いる方法を考えた。この方法を用いるとレベル1の符号化が簡単に行える。Fig.2.1はその符号化の原理を示したもので，符号化の手順は次の通りである。

まず，入力画像の値（0か1）に対し相異なる波長（λ_0とλ_1）を割り当て，入力画像の各画素を，その画素値に対応する波長の光のみを透過する色フィルタに置換する。すなわち，入力画像中，値0の画素は波長λ_0の光を通す色フィルタで，値1の画素は波長λ_1の光を通す色フィルタで表示する。次に，図に示すように，画素サイズを周期とするカラーRonchi格子を画素置換に用いた色フィルタで作製し（たとえば，赤緑赤緑……から成るRonchi格子），色フィルタに

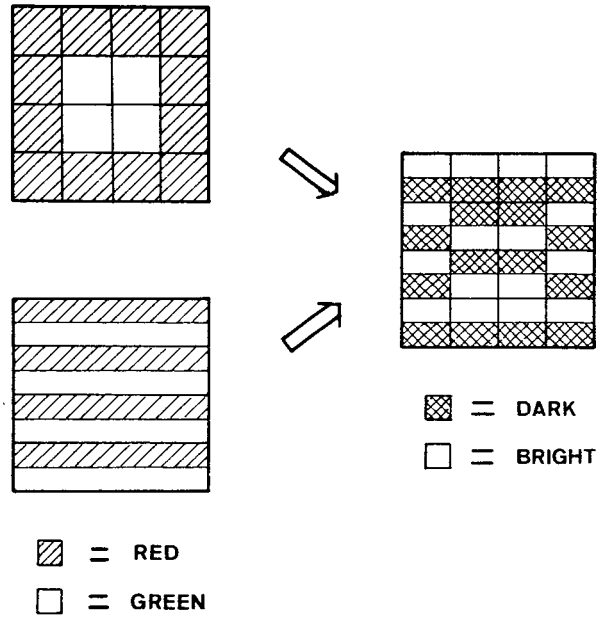


Fig.2.1　カラー符号化法

変換した画像と画素区画が一致するように重ねる。この重畳画像を背面から白色光で照明し，出力を光の明暗として検出すれば，レベル１の符号化像を得る。この操作は色フィルタによるＥＱＵ演算（Table 1.1 参照）を利用したものである。

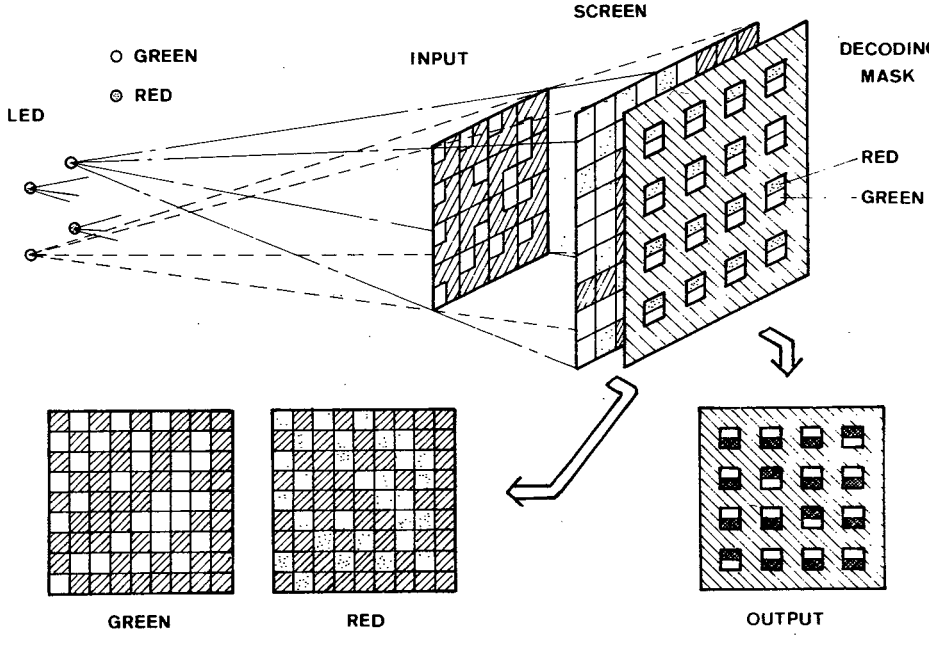


Fig.2.2　出力結果の符号化

この符号化法の応用としては，並列光論理演算によって得られた出力結果の符号化が考えられる。Fig.2.2に原理図を示す。この光学系はFig.1.2の多重投影光学系を少し変更したもので，各点光源は点滅のかわりに波長λ_0かλ_1いずれかの光を発し，復号マスクの各窓は波長λ_0とλ_1の光を通す色フィルタで構成される。図はXOR演算実行の様子を示す。波長λ_1を発する光源はXOR演算を実行し，波長λ_1の色信号で結果を出力する。復号マスクの波長λ_0とλ_1の光を通す色フィルタはEQU演算を行い，その結果，下半分が明るい符号パターンを得る。一方，波長λ_0の光を発する光源はEQU演算（XOR演算の否定）を実行し，波長λ_0の色信号で結果を出力する。この結果は色フィルタにより上半分が明るい符号パターンとなる。この操作で出力された画素はXOR演算の結果が0の画素であり，その結果，図に示すようにレベル1の符号化像を得ることができる。

2.2.2 CGHフィルタリング符号化法

計算機ホログラム（computer generated hologram；CGH）を用いたフィルタリング技術[41]を応用してレベル1およびレベル2の符号化を実現する方法を考えた。この方法は光の干渉によるXOR演算と光波の透過によるAND演算を併用したもので，CGHフィルタリング符号化法と呼ぶ。

Fig.2.3にレベル1の符号化を実行するCGHフィルタリング処理光学系を示す。CGHはFig.2.3(b)の像のフーリエ変換型ホログラムである。P_1上の画像を平面波照明すると，P_1上の画像とP_2上のCGHに記録した像（Fig.2.3(b)）との相互相関像とコンボリューション像がP_3上に得

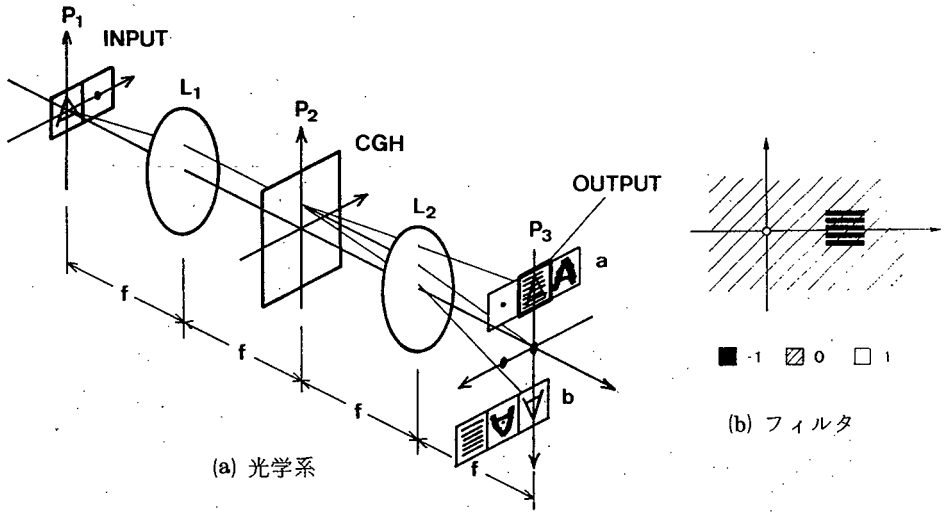


Fig.2.3 CGHフィルタリング符号化法（レベル1）

られる。Fig.2.3に示すように，P_1上の画像は被符号化２値画像Ａと原点シフトしたデルタ関数であり，CGH上に記録する像はデルタ関数と原点シフトしたRonchi 格子のフーリエ変換像である。ここで，CGHから再生されるデルタ関数とRonchi 格子の位相はπずれるように設定しておく。以上の系で得られたP_3上のコンボリューション像に注目すると，中央部分に入力画像Ａと，そのＡと位相がπずれたRonchi 格子Grとが重なる領域が存在する。この領域内では，ＡかGrのいずれか一方が明部であれば，光波が到達した対応点は明るくなる。しかし，Ａ，Gr両方が明部ならば，互いに位相がπシフトした２つの光が対応点にやってくるため互いの光は干渉により打消され暗くなる。もちろん，Ａ，Gr共，暗部ならば対応点には光が届かず暗くなる。この操作によってＡとGrとの並列XOR演算が実現できることになる。レベル１の符号化像はこの入力画像とRonchi 格子とのXOR演算結果として得られる。

レベル２の符号化法としては２つの方法が考えられる。第１の方法はFig.2.4に示すもので，レベル１の符号化像をインコヒーレント的に重ね合せる方法である。Fig.2.3の系と比較するとP_1面の入力像とP_2上のCGHにフーリエ変換して記録する像（Fig.2.4(b)）が異なる。P_1面に２つの入力画像Ａ，Ｂと２つの原点シフトしたデルタ関数を置き，入力画像とデルタ関数を１組として，異なる光源を用いた２系統のコヒーレント照明系で平面波照明する。各照明系はそれぞれ入力画像Ａと入力画像Ｂに対しレベル１の符号化を行う。ここで，P_3上のコンボリューション像の中央部分に，２つのレベル１の符号化像が重なる領域が存在するように系を設定しておく。

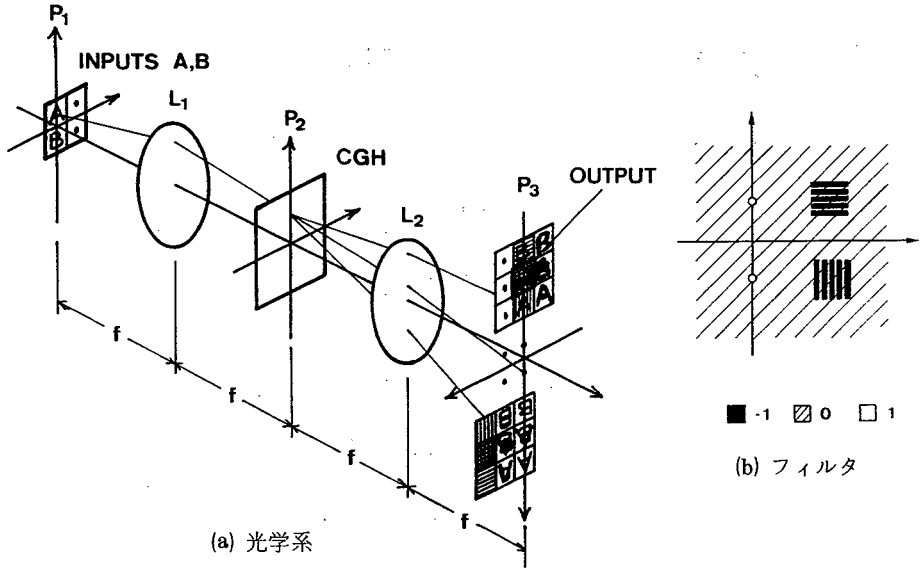


Fig.2.4　CGHフィルタリング符号化法（レベル２－Ｉ）

それぞれの符号化像は異なる光源を用いた別系統のコヒーレント照明像であり，互いにインコヒーレントなため，再生像が重なった領域では3レベルの強度値が存在する。そこで，最高強度のみを取出す閾値処理を行えばレベル2の符号化像を得る。閾値処理には空間光変調素子を利用することが考えられる。[26]

第2のレベル2の符号化法としてレベル1の符号化法に用いたXOR演算とAND演算とを組合せた方法を考えた。Fig.2.5に(i, j)画素の符号化の手順を示す。この手順は次の3つの処理からなる。

1. 入力データa_{ij}と符号パターン■とのXOR演算
2. 1.の結果と入力データb_{ij}とのAND演算，および1.の結果と符号パターン◧のAND演算
3. 2.の2つの結果に対するXOR演算

以上の処理は，(2−1)〜(2−4)式で表現される4種類の論理演算を空間的に異なる4つの場所で実行していることに相当する。

$$\bar{a}_{ij}\bar{b}_{ij}=[(a_{ij}\oplus 1)\ b_{ij}]\oplus[(a_{ij}\oplus 1)\cdot 1] \qquad (2-1)$$

$$\bar{a}_{ij}b_{ij}=[(a_{ij}\oplus 1)\ b_{ij}]\oplus[(a_{ij}\oplus 1)\cdot 0] \qquad (2-2)$$

$$a_{ij}\bar{b}_{ij}=[(a_{ij}\oplus 0)\ b_{ij}]\oplus[(a_{ij}\oplus 0)\cdot 1] \qquad (2-3)$$

$$a_{ij}b_{ij}=[(a_{ij}\oplus 0)\ b_{ij}]\oplus[(a_{ij}\oplus 0)\cdot 0] \qquad (2-4)$$

ここで，⊕はXOR演算子を示す。

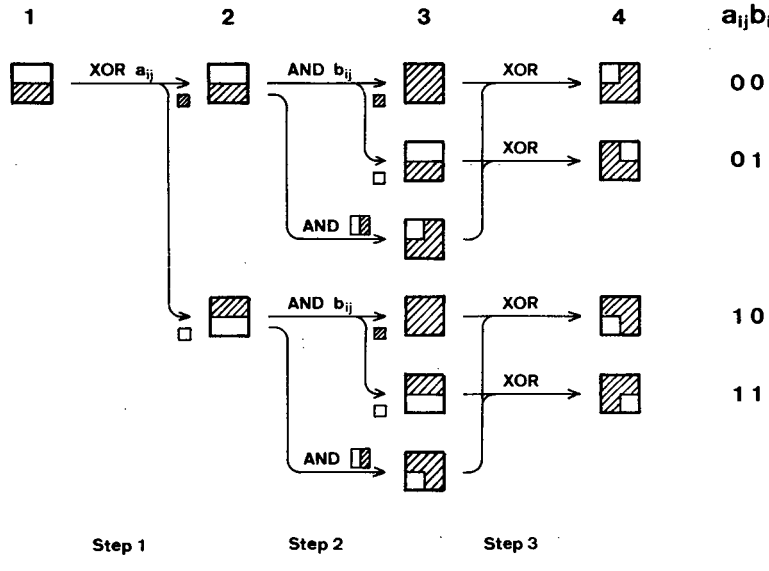


Fig.2.5　XOR演算とAND演算による符号化

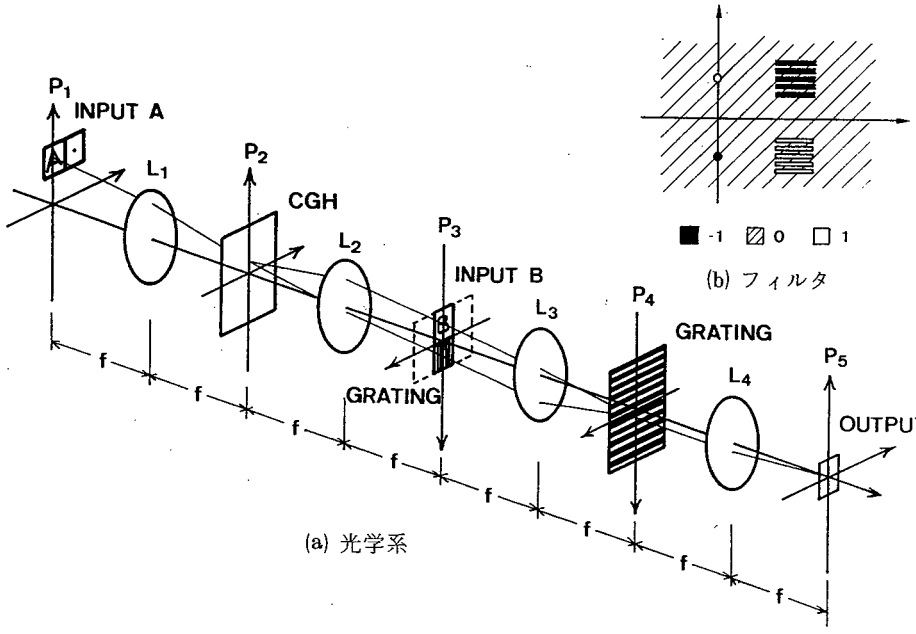


Fig.2.6　CGHフィルタリング符号化法（レベル2－II）

Fig.2.6(a)が上述の処理を実現する光学系である。P_1 面に入力画像Aと原点シフトしたデルタ関数を置き，コヒーレントな平面波で照明する。P_2 上のCGHはFig.2.6(b) に示す像のフーリエ変換像を記録したものである。P_3 面直前には互いに位相がπずれた2つのレベル1の符号化像が出力される（処理1）。各符号化像はP_3 上に置かれた入力画像BまたはRonchi格子を透過し，それらとのAND演算を実行する（処理2）。各演算結果はP_4 上の格子により偏向され，P_5上で重畳される。重畳する2つの像は互いに位相差πを持つのでXOR演算が実行される（処理3）。この結果，レベル2の符号化像を得る。

計算機ホログラムを用いた符号化法の問題点は，ホログラムの合成方法である。回折効率が高く，高品質の像を再生するフーリエ変換型ホログラムを作成するには広いダイナミックレンジと高い分解能を有する記録材料が必要である。このためには最近，技術の進歩の著しい電子ビーム描画法[42]などが利用できるものと思われる。ここで示した方法はやや複雑ではあるが，光学処理の並列性を生かせるという点では有望な符号化法の1つである。

2.2.3　偏光面符号法

光の持つ特性のうち情報処理に利用されるべきものの1つとして偏光状態の操作があげられる。2値情報を対象とする場合，直交する2つの直線偏光によって情報を表現すれば，従来からある種々の偏光処理技術を用いることができる。[43,44]

Fig.2.7に偏光面を利用した符号化法を示す。図中の矢印が直線偏光の偏光面の方向を示し，

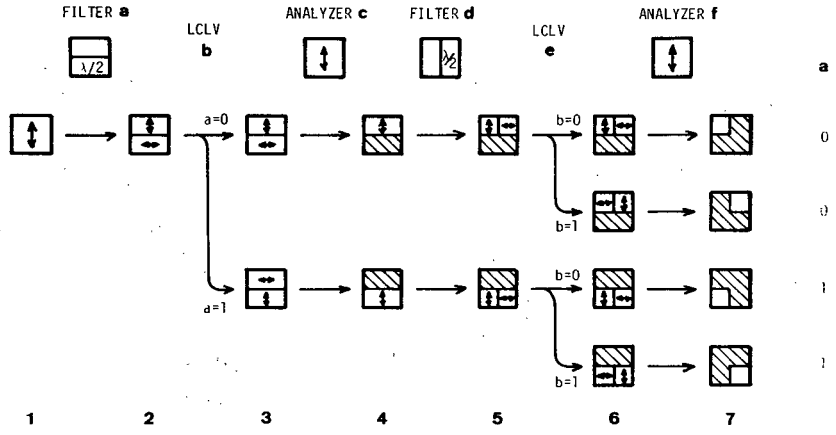
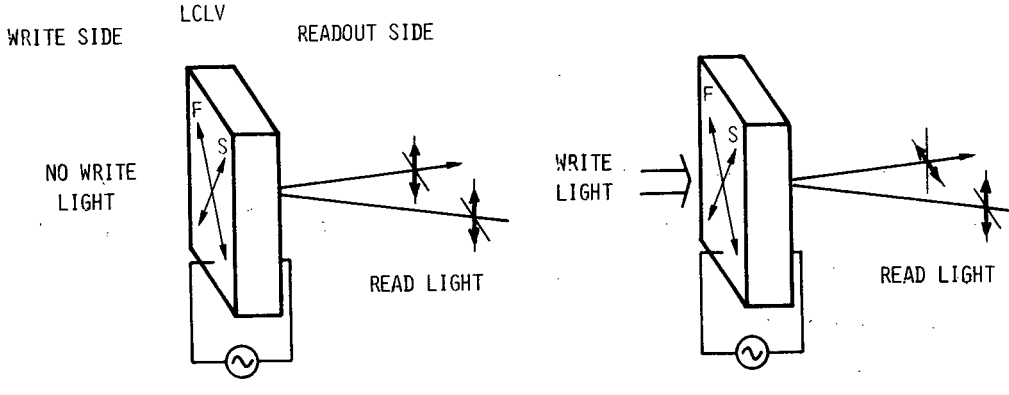


Fig.2.7　偏光面符号法

ハッチングは暗信号を表わす。この符号化法では半波長板による偏光面の90°回転と，検光子による偏光面 / 光強度変換を基礎技術として利用する。また，動的符号化のために液晶ライトバルブ（liquid crystal light valve；LCLV）[26,32]の使用を仮定している。LCLVは2次元光波信号制御可能な反射型半波長板とみなせる。したがって，書込み光の有無により，素子各部からの読出し光の偏光状態の変化（偏光面の90°回転）を制御することができる（Fig.2.8）。

Fig.2.7に示した各手順を説明する。まず，偏光子により垂直方向の直線偏光を作り，フィルタ**a**に入射させる（手順1）。フィルタ**a**は2つの部分より成り，その上半分が等方性ガラス，下半分は45°方位の半波長板である。フィルタ**a**透過後の光信号の偏光面は図に示すように，上下各領域で直交した状態となる（手順2）。この光信号をLCLV**b**の読出し光として用いると，



(a) 書込み光がある場合　　(b) 書込み光がない場合

Fig.2.8　LCLVの機能

LCLVの書込み信号光の状態により，2種類の出力光の状態が得られる。ここで，入力信号 a をLCLV**b**の書込み光とすると，a の値にしたがい偏光面の方向が変化する（手順3）。検光子**c**は偏光面情報を光強度に変換し，その結果，レベル1の符号化像を得る（手順4）。上と同様な手順を用いて，フィルタ**d**，LCLV**e**，検光子**f**で符号化を行うとレベル2の符号化像が得られる（手順7）。ただし，フィルタ**d**は左半分が等方性ガラス，右半分が45°方位の半波長板から成っている。また，LCLV**e**の書込み光には入力信号 b を用いる。

もちろん，この符号化処理は2つの入力信号を各LCLVの書込み信号として用いるので画像単位で並列に実行できる。その場合，フィルタ**a**およびフィルタ**d**は半波長板と等方性ガラスから成るRonchi格子となり，その格子定数が処理点数（画素数）を決定する。

2.2.4 複屈折符号化法

偏光面情報を利用する符号化法として単軸結晶による複屈折現象[45]を応用した方法を考案した。単軸結晶中では入射光は常光線と異常光線に分かれ，それぞれ異なった方向へ伝播する。結晶光学によれば，常光線，異常光線はそれぞれ，入射光波の偏波成分であり，結晶の光学軸と入射光のなす角度および偏光状態により2つの屈折光の伝播方向および強度比を制御することができる。

Fig.2.9は単軸結晶平行平板に光波が垂直入射する場合を描いたものである。入射光の方向ベクトルと光学軸が張る平面（主平面）に対して垂直に振動する偏波成分は，結晶に入射後常光線としてそのまま直進する。一方，主平面に平行に振動する偏波成分は異常光線として結晶内を斜

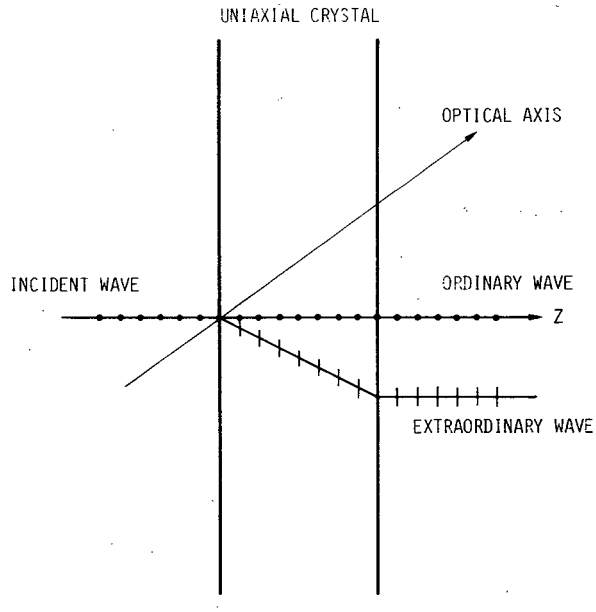


Fig.2.9 単軸結晶平行平板における複屈折

めに進行する。この２つの光波は結晶端面で再び屈折し，結晶通過後，常光線，異常光線は再び入射光の進行方向と平行な方向に進む。しかし，異常光線は空間的にシフトしている。このシフトの状態を有効に利用すれば，レベル１の符号化像を得ることができる。たとえば，LCLV等の偏波面制御素子を用いて入力画像の２値情報を単軸結晶の主平面に垂直な方向と水平な方向に振動する２つの偏光に変換する。その直線偏光を単軸結晶に入射させると，出力光によってレベル１の符号化が実現できる。

レベル２の符号化は上述の方法を直列接続して行う。Fig.2.10はその実現方法の例で透過型の偏波面制御素子を利用する場合を示す。ここで，直列接続する単軸結晶は互いに主平面が直交するように配置する。図の右端に描いたグラフは２つの入力信号 a，b に対するシフトの結果を示す。すなわち，a, b の値の組合せにより図中のいずれかの位置に光信号が得られる。これはFig. 1.3(c)に示す符号パターンと等価なものであるが，a, b の組合せと位置との対応関係は異なる。したがって，このままでは本論文で説明した並列光論理演算法に適合させることはできない。しかし，これは単に符号パターンの定義の問題であり，Table 1.4等に示すカーネルユニットの組合せを変更することが解決でき，本質的な問題ではない。

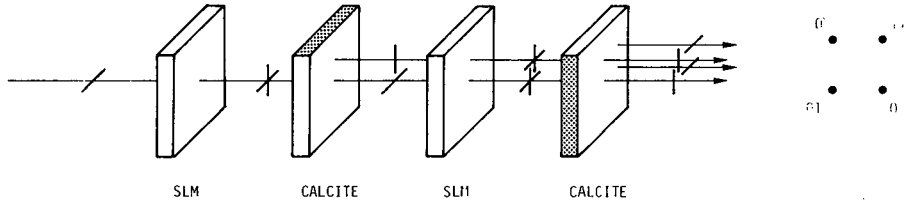


Fig.2.10　複屈折符号化法

2.3　実時間２次元相関演算法

並列光論理演算法の中で用いられる２次元相関演算は一般の相互相関演算とはやや異ったものである。すなわち，（1－15）式で示すように演算対象データの一方はN^2画素区画よりなる離散画像であるが，他方は$(4L+2)^2$の離散点で構成される点状関数（カーネル）である。実際の処理系では，

$$N^2 \gg (4L+2)^2 \qquad (2-5)$$

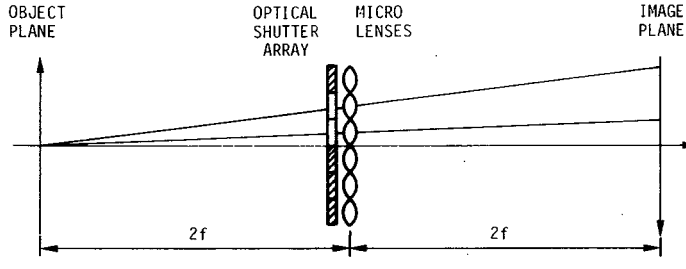
であるため，この２次元相関演算の実行には多くの方法が考えられる。しかし，並列光論理演算法で連続的に処理する場合や，後で述べるシステム化の際には実時間性が要求される。そこで，本章では，並列光論理演算法を目的として考案した実時間制御可能な２次元相関演算法を説明する。

2.3.1　多重投影法

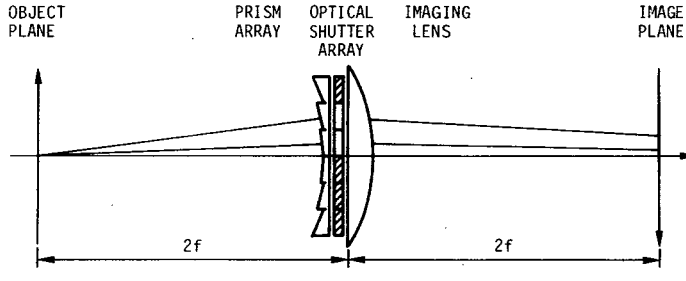
最も簡単で拡張性のある方法としてはFig.1.2に示す多重投影光学系を用いた方法があった。この方法は1.2節で詳細に説明したように，非常に簡単な光学系で行え，しかも点光源アレイの点滅で処理の内容を容易に制御できるという特長を持っている。この方法は幾何光学に基づいているものの，1.2.7項で評価したように，情報の高密度化に対しても十分対応できる方法である。しかし，問題点として，相関演算の対象画像の入力に高解像点の透過型空間光変調素子が必要なこと，カーネルが大きくなった場合（４Ｌ＋２が大きい場合）点光源のshading現象により投影像の中心部と周辺部で照度差が生じ，その結果出力結果のS/N比が低下すること等があげられる。したがって，解像点数が極端に多くなく，小規模のカーネルを用いる場合に有効な方法と言える。

2.3.2　多重結像法

並列光論理演算法で用いる２次元相関演算法は，多数のシフトした符号化画像の重ね合わせと考えることができる。シフト像は結像系の中に偏向素子を入れたoff-axis結像系（空間的シフト像を与える結像系）によって得られる。したがって，このoff-axis結像系を多数用意すれば所望の２次元相関演算を実現することができる。もちろん，各off-axis結像系内の偏向素子はそれぞれ異なったシフト像を与えるように設計する必要がある。



(a) マイクロレンズアレイを利用するもの



(b) プリズムアレイを利用するもの

Fig.2.11　多重結像光学系Ⅰ

Fig.2.11に多重結像系の例を2つ示す。(a)はマイクロレンズアレイと光シャッタアレイを組合せたもの，(b)は結像レンズ，プリズムアレイ，光シャッタアレイを組合せたものである。動作原理は共に同じで，物体の後側よりインコヒーレント照明を行い，光軸をシフトして配置したマイクロレンズやプリズムの偏向作用を用いて像面上に種々のシフト量を持つ画像を結像させる。この時，個々のマイクロレンズの位置や個々のプリズムの頂角，斜面の方位により各像のシフト量が決定される。ここで，各マイクロレンズやプリズムを通過する光線を光シャッタアレイにより制御すれば選択された任意のシフト像だけを像面上で重ねることができ，その結果，目的の2次元相関演算が行える。この場合，光シャッタアレイの開閉パターンが相関演算のカーネルに対応し，その設定で所望の2次元相関演算，ひいては並列光論理演算が実現できる。

ここで，Fig.2.11(a)の系と(b)の系，それぞれの特徴を述べる。(a)の方式では，必要な光学要素は微小光学技術の応用により容易に作成できることがあげられる。[46] しかし，図からわかるように，マイクロレンズの間隔がシフト量を決定するため，処理画像が高密度であれば各マイクロレンズは非常に小さなものとなり，透過光量，結像性能の点が問題となる。一方，(b)の方式は自由に像のシフト量を設定できるため，処理に必要なシフト像だけを生成するようにプリズムアレイを設計・製作すれば効率的な2次元相関演算系を作ることができる。しかし，精度の高い微小プリズムアレイの作成は現在のところまだ困難で，周辺技術の発達を待たなければならない。もちろん，計算機ホログラム素子を用いて同等な系を構成することもできるが，この場合にもホログラム作製技術の向上が必須条件となる。

Fig.2.11の2つの系は光シャッタアレイにより各シフト像の重ね合わせを制御するが，高速で2次元制御可能な光シャッタアレイの利用が困難な場合も考えられる。その時には光学系は少々複雑になるが，光シャッタアレイを用いずに多重結像系を構成することもできる。Fig.2.12に照明系を利用した多重結像法を示す。光源面上の点光源は物面上の符号化画像を平行照明し，プリズムアレイに光源自身の像を結ぶ。プリズムに入射した光はプリズムの偏向作用を受け，像面上ではシフトした符号化像の像を作る。この系では，1つの点光源に1つのシフト画像が対応する

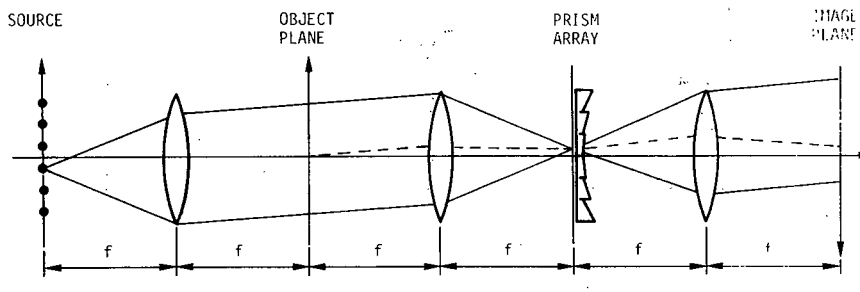


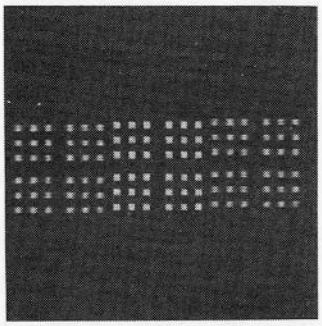
Fig.2.12　多重結像光学系II

ため，点光源の点滅制御により実時間２次元相関演算が実現できる。点光源アレイの制御は光シャッタアレイの制御にくらべ容易であり，高速化も期待できるという点で有望な方法である。

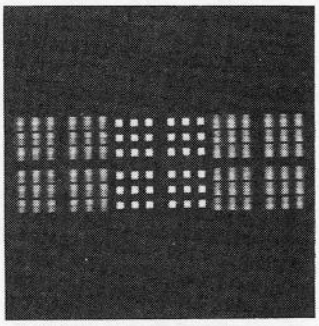
Fig.2.13はFig.2.12の系による２次元相関演算の実験結果である。ここでは，プリズムアレイのかわりに回折格子アレイを用いた。回折格子アレイは計算機制御されたドラフタで描画したものを写真縮小し，リスフィルムに記録し，さらに，漂白して作製した。回折格子アレイは４つのRonchi 格子で構成し，各部分の空間周波数はFig.2.13(a)に示すものとした。光学系は f＝300㎜のレンズを用い，点光源にはスタンレー製の高輝度LED，EBY−3402Sを使用した。入力画像は２㎜×２㎜とした。Fig.2.13(b)〜(d)が光源アレイの発光パターンをかえて得た出力結果である。回折格子を利用したため，＋１次，０次，－１次の像が出ているが，必要な結果は＋１次（左側）のものである。

	0	1
0	ν = -0.49 μ = 15.3	ν = 0.49 μ = 15.3
1	ν = -0.49 μ = 14.3	ν = 0.49 μ = 14.3

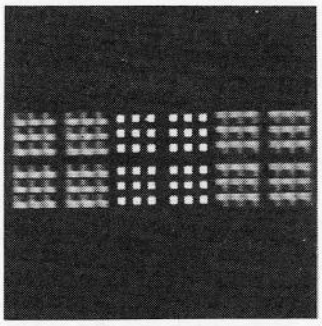
(a) 回折格子アレイの空間周波数［line/㎜］



(b) 出力像１



(c) 出力像２



(d) 出力像３

Fig.2.13 多重結像法による実験結果

2.3.3 回折パターンサンプル法

2次元回折格子による回折を利用した実時間2次元相関演算法を考案した。Fig.2.14にその光学系を示す。この光学系は一種のコヒーレントフィルタリング処理系で，物面上の符号化画像が処理された後，その結果が像面に出力される。ここで，物面より距離zにある2次元回折格子が符号化画像の回折光を多方向へ分解・偏向するため，スペクトル面では符号化画像のスペクトルの島が格子状に多数配列する。このスペクトルの島は後側のレンズにより像面上で符号化画像の像に変換されるが，2次元回折格子が物面内にないため，スペクトル面上の個々のスペクトルの島からはシフトした画像が再生される。そこで，スペクトル面上の各スペクトルの透過を制御すると演算に必要なシフト像だけを像面上で重ねることができる。

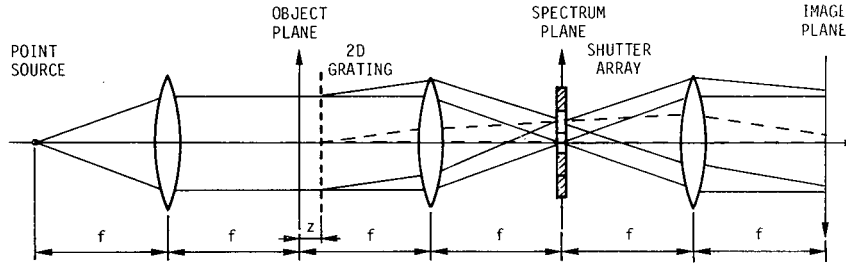


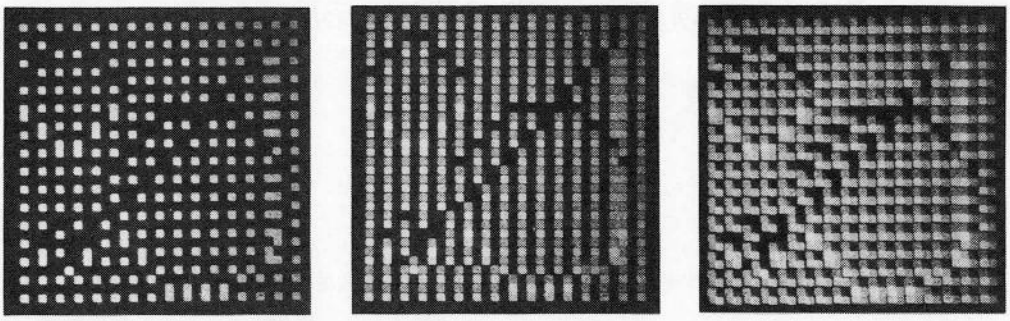
Fig.2.14 回折パターンサンプル法

この方法では，符号化画像が高密度のものになるとスペクトル面で個々の符号化画像のスペクトルが広がってしまうため，取扱える処理点数が限定される。また，像面では，各シフト像がコヒーレント的に重ねられるので，干渉の影響も無視できない。しかし，2.3.2項の多重結像法のようにマイクロレンズアレイやプリズムアレイ等，特殊な素子を必要としない点では有用な方法である。しかも，シフト画像のシフト量$\triangle x$は

$$\triangle x = \frac{z\lambda}{d} \qquad (2-6)$$

で与えられるため，2次元格子を設定する位置により自由に制御できるという特徴を持つ。ここでλは光波の波長，dは格子定数である。

Fig.2.15は回折パターンサンプル法による処理結果である。実験にはHe-Neレーザーを光源として用いた。回折格子の空間周波数は40 line/mmで入力像の大きさは18㎜×18㎜である。演算の種類はスペクトル面に置くマスクの種類によって変更した。出力結果を見る限り干渉の影響はほとんど見られない。



(a) 出力像 1　　(b) 出力像 2　　(c) 出力像 3

Fig.2.15 回折パターンサンプル法の実験結果

2.4 結　言

並列光論理演算法の実現に必要な符号化と実時間2次元相関演算に対し，著者はいくつかの新しい方法を考案し，その概要を説明した。2.2節では，2値画像データを符号化画像へ変換する手法として色フィルタによるEQU演算を利用するもの，計算機ホログラムによるフィルタリングを用いるもの，偏光面情報を利用するもの，単軸結晶による複屈折現象を利用するものの4つを考案し，個々について説明した。また，2.3節では，符号化画像の実時間2次元相関演算法として，多重投影法，多重結像法，回折パターンサンプル法の3つを考案し，その概要および特徴を述べた。いずれの方法も従来の光学技術を応用したもので，光の持つ並列処理能力を十分に生かしたものである。これらの方法を用いることにより並列光論理演算法の持つ能力を十分に引き出すことが期待できる。連続的に並列論理演算を実行する目的には空間光変調素子や点光源アレイ等，実時間制御可能な能動素子の開発が不可欠である。本章の議論は今後開発されるべき空間光変調素子や能動結像素子等のデバイス開発の方向づけに対し有用な情報を与えるものと思われる。

第3章　並列光論理演算法の画像処理への応用

3.1　緒　　言

並列光論理演算法はSIMD（single instruction multi-data flow）形式の並列演算を効率よく実現する方法であるが，このSIMD並列演算に適した応用として画像処理があげられる。画像処理は画像* の形で得られる膨大な2次元情報から人間または機械にとって有用な情報を抽出する技術であり，その応用分野は極く身近かなものからリモートセンシング像（各種衛星写真，航空写真など）による資源開発や医学における医用画像処理（X線写真，超音波像，X線CT《computed tomography》像など）など広範囲かつ高度な応用にまでおよんでいる。[47-49] 画像処理において必要な演算は目的とする処理の内容に大きく依存するが，低レベルの画像処理の段階では同一の演算を，画像を構成する全画素に対し実行する必要性が頻繁に起こり，その効率的な実行が，画像処理技術の実用化の鍵となっている。この要求はまさに光コンピュータ開発の1つの動機でもあり，1章で述べた並列光論理演算法に恰好の応用の場を与えてくれる。

現在，ディジタル画像処理の分野で広く用いられている基本的な演算法の1つにディジタルフィルタリング[47,48]がある。これは，M×M画素の配列で表現した点像分布関数（ディジタルフィルタ）と被処理画像との離散的コンボリューションを実行する処理である。ディジタルフィルタのとり方により差分処理や2次微分等，様々な処理を実現することができる。電子計算機ではこの演算をすべて2値情報に変換したデータに対して実行するが，光学技術の特長である並列情報伝送能力を生かした並列光論理演算法を用いると，従来とは異った特徴を持つ並列ディジタル画像処理を実現することが可能である。著者はこの観点に立って，並列光論理演算法をもとにディジタル画像処理を目的としたいくつかの新しい並列光演算法を考案した。本章では，2値画像処理と多値画像（濃淡画像）処理に分け，新しく考案した光学的な並列ディジタル画像処理法を説明する。

3.2　並列2値画像処理

画像処理の対象となるデータは0, 1の情報よりなる2値画像と0～mの値を持つ多値画像（濃淡画像）に分類できる。しかし，通常，並列ディジタル計算機では多値画像も2値画像に分解して処理する方式が一般的であり，[50] 2値画像に対する処理が実現できれば多値画像処理への拡張は容易である。そこで，まず，本節では2値画像を処理対象とした2つの並列演算法を説明する。1つは並

＊本論文では‘画像’を広義の意味（2次元平面に配列した情報）に用いているが，本章に限り狭義の意味（写真や絵のように2次元配列により初めて意味を持つ情報）で用いる。

列光論理演算法で用いる処理系をアナログ処理に応用した並列光ディジタルフィルタリング処理である。他方は並列光論理演算法の並列論理演算を利用する並列光論理テンプレートマッチング処理である。いずれも光学処理の並列性を十分にいかした処理であり，多値画像処理への拡張性も有している。

3.2.1 並列光ディジタルフィルタリング

並列光論理演算法で実行する処理は本質的に，２次元並列ディジタルフィルタリング処理と等価である。[47,48] 並列光論理演算法のカーネルは２次元ディジタルフィルタそのものに対応し，それらは共に被処理画像とのコンボリューション（あるいは相関演算）に供される。２次元ディジタルフィルタを構成する単一要素はカーネル内の１つのカーネルユニット*に対応する。Table 3.1 にカーネルユニットとディジタルフィルタの要素の値との対応関係を示す。ここで，被処理画像は並列光論理演算の入力画像Aであると考え，カーネルユニットの各項は任意の正値をとり得るものとする。ディジタルフィルタの負値はカーネルユニットのパターンを変えて実現することができる。Fig.3.1 はディジタルフィルタと対応するカーネルの例を示している。(a)は入力画像の反転，(b)は横方向の１次微分，(c)はラプラシアン演算用のカーネルである。ディジタルフィルタの要素数が大きい場合にはカーネルユニットの数を増やすことにより対応できる。

並列光ディジタルフィルタリングの実行は並列光論理演算法の処理手順にしたがって行う。すなわち，被処理画像の符号化，Fig.3.1 に示すカーネルを用いた２次元相関演算，復号の順で処

Table 3.1 ディジタルフィルタ処理用カーネルユニット

KERNEL UNIT	ELEMENT OF DIGITAL FILTER	MASK
$\begin{pmatrix} 0 & 0 \\ p & p \end{pmatrix}$	p	-
$\begin{pmatrix} p & p \\ 0 & 0 \end{pmatrix}$	-p	-
$\begin{pmatrix} 0 & 0 \\ 0 & p \end{pmatrix}$	p	$b_{ij} = 1$
$\begin{pmatrix} 0 & p \\ 0 & 0 \end{pmatrix}$	-p	$b_{ij} = 1$
		$p \geqq 0$

* カーネルユニットは 1.3 項で述べたものと同一のものである。ただし，カーネルユニットの各要素は２値とは限らず，非負の実数である。

DIGITAL FILTER

$$\begin{matrix}0&0&0\\0&-1&0\\0&0&0\end{matrix}\qquad\begin{matrix}0&0&0\\0&1&-1\\0&0&0\end{matrix}\qquad\begin{matrix}0&-1&0\\-1&4&-1\\0&-1&0\end{matrix}\qquad\begin{matrix}0&0&0\\0&1&-1\\0&0&0\end{matrix}$$

masked by B

KERNEL

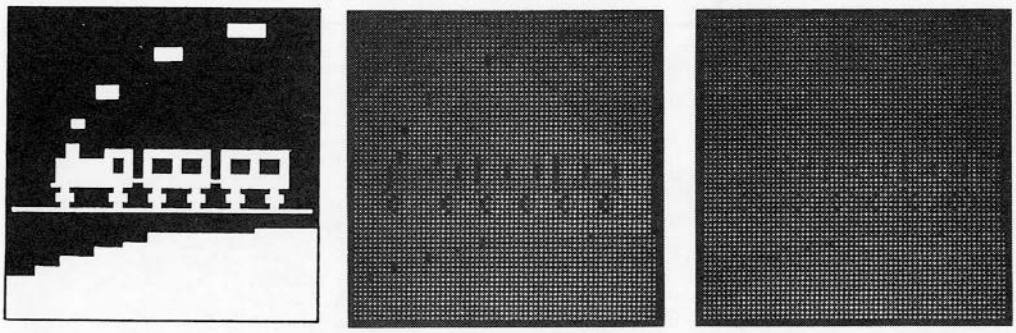
$$\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}\quad\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&1&1\\0&0&1&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}\quad\begin{pmatrix}0&0&1&1&0&0\\0&0&0&0&0&0\\1&1&0&0&1&1\\0&0&4&4&0&0\\0&0&1&1&0&0\\0&0&0&0&0&0\end{pmatrix}\quad\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}$$

(a) (b) (c) (d)

Fig.3.1 ディジタルフィルタとカーネル

理する。この場合，出力結果は多値画像として得られる。また，ディジタルフィルタ内に負の要素が含まれる場合には出力結果は，ディジタルフィルタ内の全部の負の要素の絶対値総和に等しいバイアスを含む。たとえば，Fig.3.1(c)では出力結果はバイアス4が加わったものとして得られる。このようなバイアスの付加は処理系がインコヒーレント光学系であることに起因している。

先述したように，並列光論理演算法では2つの入力画像が同時に取扱える。そこで，ディジタルフィルタリングの被処理画像とは別に，処理の制御用に他の画像信号Bを使用することができる。その結果，この制御画像（マスク画像と呼ぶ）が1の値を持つ画素に限りフィルタリング処理を実行させることが可能となり，SIMDの並列処理においてもある程度の局在処理を実行することができる。Table 3.1にはこの局在処理用のカーネルユニットもあわせて示してある。Fig. 3.1(d)は入力画像Bをマスク画像とする横方向の1次微分のカーネルである。この場合，マスク



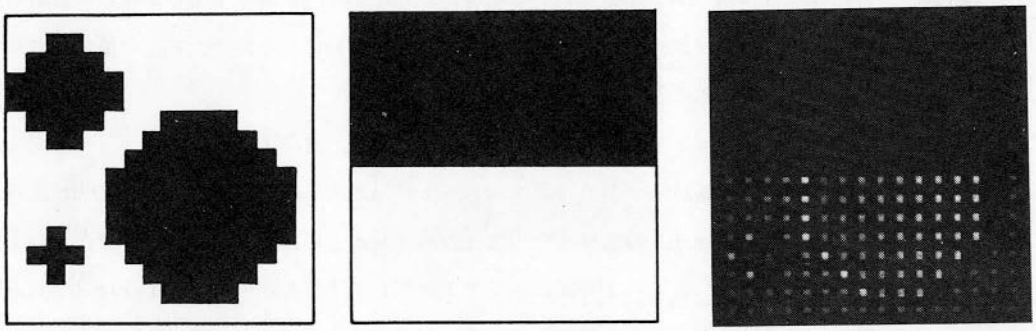
(a) 被処理画像　(b) 横方向1次微分　(c) ラプラシアン演算

Fig.3.2 並列光ディジタルフィルタリングの処理結果

画像中で値1を持つ画素に対してのみ1次微分処理を実行する。

Fig.3.2は並列光ディジタルフィルタリングによる実験結果である。(a)が被処理像で64×64画素から成る。(b)は横方向の1次微分，(c)はラプラシアンの結果を示す。演算の実行にはLEDアレイを点光源とする多重投影光学系を用いた。各処理はそれぞれFig.3.1(b)，(c)のカーネルを用いて行ったものである。Fig.3.3はFig.3.1(d)のカーネルで局在1次微分処理を行った結果である。(a)が対象画像，(b)がマスク画像，(c)が出力画像を示す。

この並列光ディジタルフィルタリングでは光学処理の並列性を生かすと共に光信号のダイナミックレンジを十分に利用している。その結果，1章で述べた論理演算法とは異なった体系の処理が可能となっている。



(a) 被処理画像　　(b) マスク画像　　(c) 処理結果

Fig.3.3　局在1次微分

3.2.2　並列光論理テンプレートマッチング

並列光論理演算法の並列演算能力を生かした2値画像処理法として並列光論理テンプレートマッチングを考えた。テンプレートマッチングとはテンプレートと呼ばれる特定のパターンと同じパターンを被処理画像中から抽出する処理で，画像の特徴抽出等に有効な処理である。[47,48] 普通，この処理は被処理画像とテンプレートとの相関演算で行われるが，各画素ごとに近傍画素の値を調べる手法を用いても実現できる。たとえば，Fig.3.4のテンプレートのパターンを画像内で探すことは

$$c_{ij} = \bar{a}_{i-1,j-1}\, a_{i-1,j}\, \bar{a}_{i-1,j+1}\, a_{i,j-1}\, a_{i,j}\, a_{i,j+1}\, \bar{a}_{i+1,j-1}\, a_{i+1,j}\, \bar{a}_{i+1,j+1} \qquad (3-1)$$

なる論理演算を画像中の全画素について実行することと等価である。ここで，出力値1の画素がテンプレートパターンの存在する場所を示す。したがって，並列光論理演算法による論理演算を

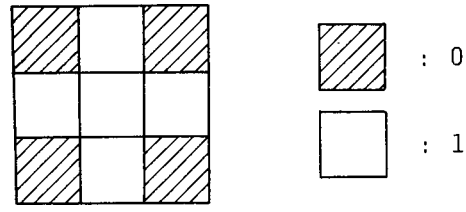


Fig.3.4　テンプレートパターン

用いて，テンプレートパターンマッチングを行うことができる。

論理テンプレートマッチングを実行するためには（3−1）式で示すように，近傍画素データに対する論理積演算が必要である。この目的には出力信号にdark-true 論理を用いた並列光論理演算法が有用となる。（1−32）式からわかるように，dark-true 論理を用いた場合には着目画素とその近傍画素の論理演算はすべて論理積の形式で接続されているため，w_{pq} の値の組合せにより（3−1）式を実行することができる。

カーネルはカーネルユニットの集合体として構成されるが，論理テンプレートマッチングではテンプレートの1つの要素が1つのカーネルユニットに対応する。ただし，dark-true 論理を利用しているため，Table 1.4 の演算をそのまま対応させることはできない。そこで，論理テンプレートマッチングで用いるカーネルユニットとテンプレート要素との関係をTable 3.2 に示す。ここで，被処理画像は並列光論理演算の入力画像Aと考えている。Fig.3.5はテンプレートと対応するカーネルユニットの例を示したものである。大きなサイズで表わされるテンプレートもカーネルユニットの数を増やすことによって実現できる。また，並列光ディジタルフィルタリング

Table 3.2　論理テンプレートマッチング用カーネルユニット

KERNEL UNIT	ELEMENT OF TEMPLATE	MASK
		-
		-
	DON'T CARE	-
		$b_{ij}=1$
		$b_{ij}=1$

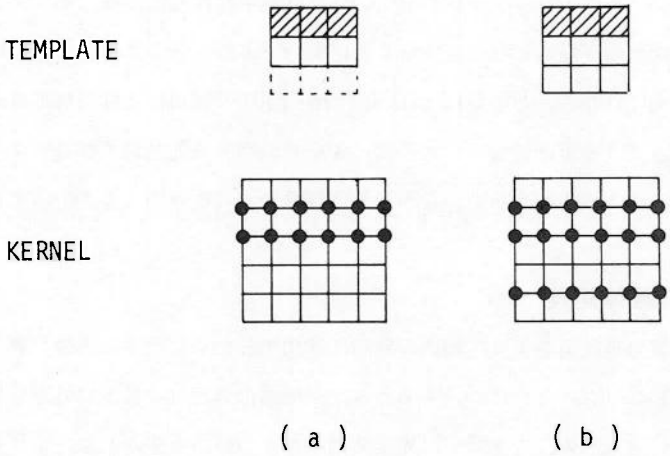
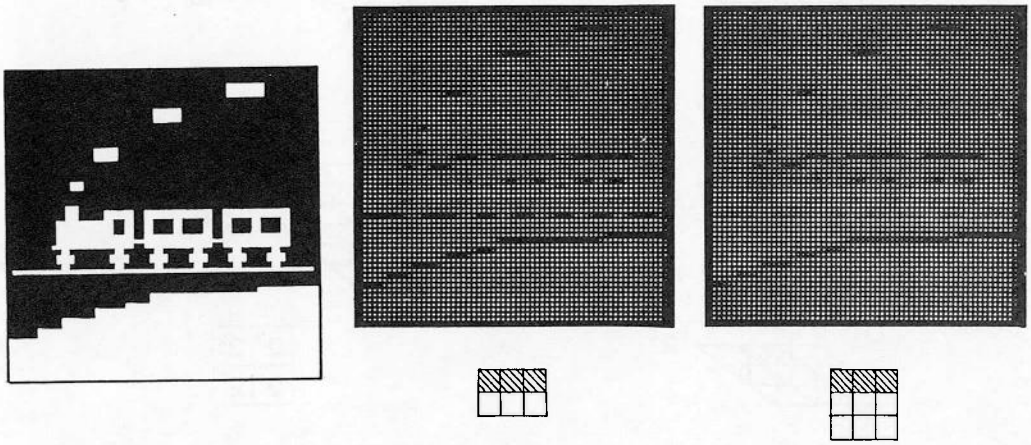


Fig.3.5 テンプレートとカーネル

と同様，マスク画像による局在処理も可能である。

論理テンプレートマッチングの実行は並列光論理演算法の手順にしたがって，被処理画像を符号化し，Fig.3.5に示すカーネルを用いた2次元相関演算により行う。dark-true 論理のため，出力画像内の暗信号がテンプレートと同一のパターンの存在する場所を示す。

Fig.3.6に並列光論理テンプレートマッチングの実験結果を示す。(a)が入力画像で，64×64画素より成る。(b)，(c)はそれぞれ異なったテンプレートに対する処理結果である。テンプレートは各処理結果の下に示すものを用い，実際の実験は多重投影光学系を用いて行った。所望の結果が得られていることがわかる。



(a) 被処理画像　　(b) 出力像 1　　(c) 出力像 2

Fig.3.6 並列光論理テンプレートマッチングの実験結果

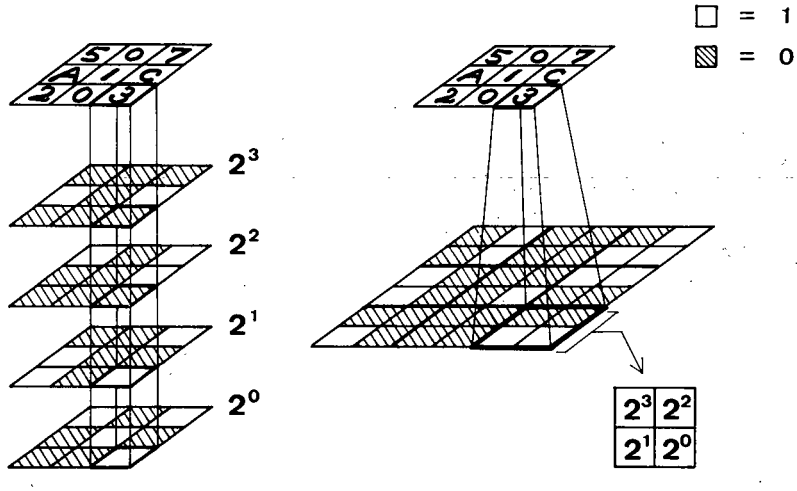
並列光論理テンプレートマッチングの最大の特長は簡単な操作により，テンプレートマッチング処理を被処理画像の全画素に対し，完全並列に実行できることである。また，検出の精度がよいことも特長にあげられる。この処理はdark-true論理の利用により近傍画素データに対する論理積を実現することで可能になったもので，4章で説明する光アレイロジックの特殊な例と考えられる。詳しくは4章で述べるが，並列光論理演算法の有用性の大きさを示す1つの例である。

3.3 並列多値画像処理

前節で示した画像処理技術をより広範囲の問題に適用するためには多値画像（濃淡画像）に対する処理方法を検討しなければならない。もちろん，多値画像処理は2値画像処理を単純に拡張しても実現できるが，拡張方法の工夫次第では光学処理の種々の長所を生かした，より効率的な処理法が実現可能である。そこで，本節では，並列ディジタル処理で多ビット情報を取扱う際に有用なビット分散配置法を説明し，次に，アナログ的な方法としてディジタル/アナログ変換法および多値符号法について述べる。

3.3.1 ビット分散配置法

並列ディジタル計算機を用いた多値画像処理では，多値画像をビットプレーンと呼ばれる2値画像に分解し，各ビットプレーンごとに処理を行う方法が用いられる。[50] Fig.3.7(a)は画像が4ビット数で構成されている時のビットプレーン展開の様子を表わしたもので，各画素の情報は4つのビットプレーンに分けられる。ここで，各ビットプレーンごとに並列光論理演算を行えば多値



(a) ビットプレーン展開　　(b) ビット分散配置

Fig.3.7 多値画像のビットプレーン展開

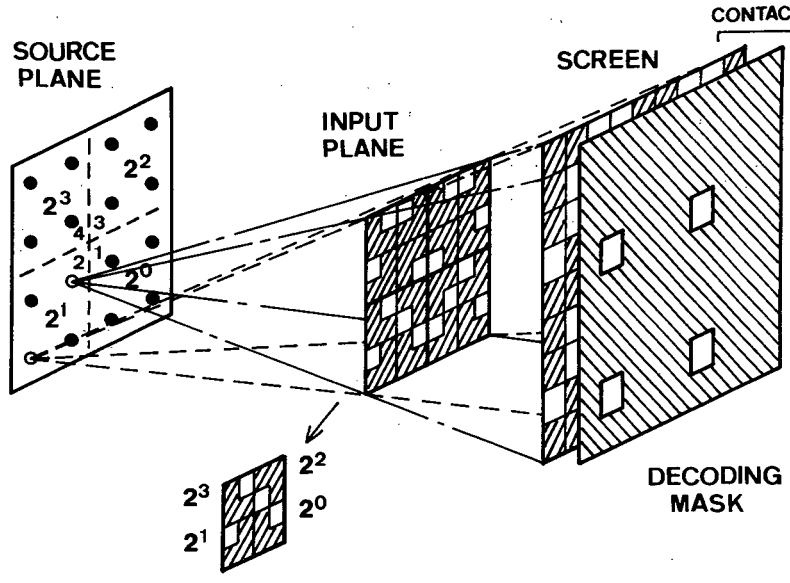


Fig.3.8　ビット分散配置画像に対する処理系

画像処理が実現できることになる。しかし，各ビットプレーン間の桁上げ等の情報交換の方法については十分考慮しなければならない。そこで，光の持つ強力な情報伝送能力を生かせる多値画像の2値画像化法を考案した。

Fig.3.7(b)に示すように，各画素の多値情報を互いに隣接する4つの画素に2値化して分散配置する。その結果，各ビット間の情報の授受は並列シフト演算により容易に行える。この場合，もとの多値画像の4倍の拡がりを持つ2値画像を取扱わなければならないが，並列光論理演算法で用いる光学系が扱える物理的限界内ならば処理点数を容易に増やせること，処理画像全体にわたって並列処理が行えること，シフト演算が簡単に実行できること等を考慮すると，このビット展開法は非常に有効な方法である。この2値画像化の方法をここではビット分散配置法と呼ぶ。この方法は4.3.2項で述べるように画像以外の多値データの取扱いの基礎技術としても応用できる。

Fig.3.8はビット分散配置法によって2値画像に変換した多値画像に対する処理を実行する系の例である。これは，Fig.1.2の多重投影光学系を変形したもので，Fig.1.2のものと比べると復号マスクの窓間隔が異なる。カーネルは4つのカーネルユニットにより構成され，各カーネルユニットが1つのビットプレーン上のデータに対して実行する処理を決める。たとえば，図中のカーネルの右下のカーネルユニット（1とラベル付けしてある）は2^0ビット位置のデータに対する処理を実行し，左下のもの（2とラベル付け）は2^1ビットのデータに対する処理を実行する。Table 3.3はカーネルとそれに対応する演算の例を示したものである。これらのカーネルを用い

Table 3.3 ビット分散配置法で用いるカーネルの例

KERNEL	FUNCTION	KERNEL	FUNCTION	KERNEL	FUNCTION
	A_0		B_0		A_0+B_0
	A_1		B_1		A_1+B_1
	A_2		B_2		A_2+B_2
	A_3		B_3		A_3+B_3

てビット分散配置画像を処理すれば出力として表中の演算結果を得る。

Fig.3.9はビット分散配置法による実験結果を示す。(a)が多値画像、(b)がそのビット分散配置画像を示す。このビット分散配置画像を符号化し，Fig.3.8の多重投影系で処理した結果が(c)～(f)である。それぞれ，異なったビットプレーン上のデータを出力させたもので，所望の結果が得られている。

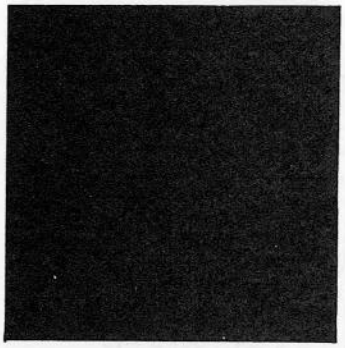
注意すべきことは，ビット分散配置法は単に多値画像データの2値画像データへの変換法を提供するにすぎないことである。したがって，実際に多値画像処理を行うには隣接画素間の演算が必要となり，より大きなサイズのカーネルを用意しなければならない。また，2値画像に対する複雑な論理演算の設計が要求され，その実現は容易ではない。しかし，原理的には可能であり，4章で述べる光アレイロジックを利用すれば多値画像処理システムの構成も決して不可能なものではない。むしろ，光の並列処理能力を生かせる方式として，今後有望な方式とも考えられる。

0	0	0	0	0	0	0	0
0	2	2	2	2	2	2	0
0	2	4	4	4	4	2	0
0	2	4	8	8	4	2	0
0	2	4	8	8	4	2	0
0	2	4	4	4	4	2	0
0	2	2	2	2	2	2	0
0	0	0	0	0	0	0	0

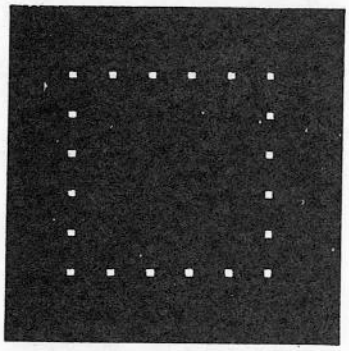
(a) 被処理画像

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	0
0	0	0	0	0	1	0	1	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	1	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

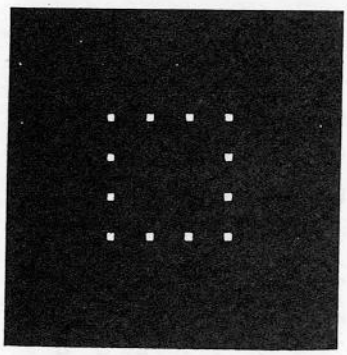
(b) ビット分散配置画像



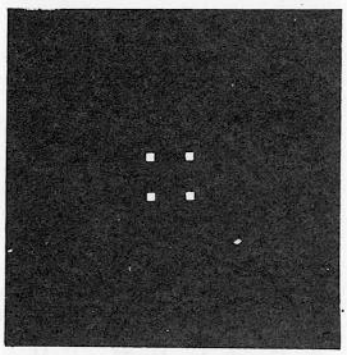
(c) 2^0bit



(d) 2^1bit



(e) 2^2bit



(f) 2^3bit

Fig.3.9 ビット分散配置法による実験結果

3.3.2 ディジタル／アナログ変換法

ここで考案したディジタル／アナログ変換法はビット分散配置法を応用したもので，出力信号をアナログ量として取扱うものである。処理光学系はビット分散配置法と同一で，たとえばFig. 3.8の系を用いる。原理は2値画像内に分散配置した各ビット情報をそれぞれのビットの重みをかけて同時に出力面へ呼出す方式をとる。各ビットごとに演算を指定すれば，その出力結果はアナログ光強度として得ることができる。Table 3.4はカーネルとそれに対応する演算の例である。これらは4ビット数の処理例である。

Fig.3.10は4ビット数で構成される2つの多値画像である。これらを入力画像としてディジタル／アナログ変換法による処理を行った結果がFig.3.11である。(a)は入力画像Aのアナログ出力像，(b)はA＋Bの並列演算結果のアナログ出力像を示す。

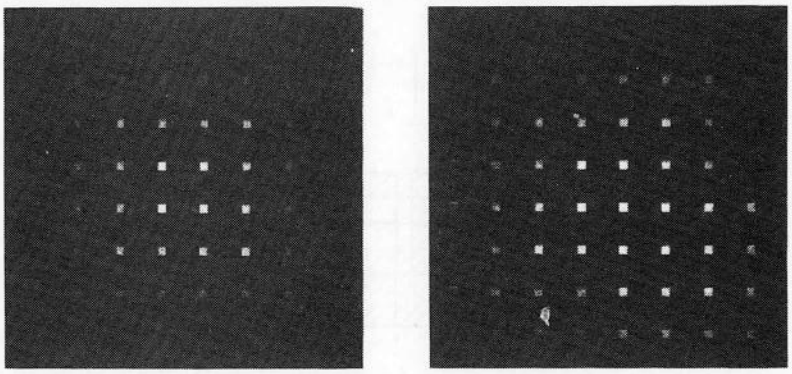
Table 3.4 ディジタル/アナログ変換法で用いるカーネルの例

KERNEL	FUCNTION	KERNEL	FUNCTION
$\begin{pmatrix} 0&0&0&0 \\ 8&8&4&4 \\ 0&0&0&0 \\ 2&2&1&1 \end{pmatrix}$	A	$\begin{pmatrix} 0&8&0&4 \\ 0&8&0&4 \\ 0&2&0&1 \\ 0&2&0&1 \end{pmatrix}$	B
$\begin{pmatrix} 0&8&0&4 \\ 8&16&4&8 \\ 0&2&0&1 \\ 2&4&1&2 \end{pmatrix}$	A+B	$\begin{pmatrix} 8&0&4&0 \\ 16&8&8&4 \\ 2&0&1&0 \\ 4&2&2&1 \end{pmatrix}$	A-B+15

$$\begin{matrix} 0&0&0&0&0&0&0&0 \\ 0&2&2&2&2&2&2&0 \\ 0&2&4&4&4&4&2&0 \\ 0&2&4&8&8&4&2&0 \\ 0&2&4&8&8&4&2&0 \\ 0&2&4&4&4&4&2&0 \\ 0&2&2&2&2&2&2&0 \\ 0&0&0&0&0&0&0&0 \end{matrix} \qquad \begin{matrix} 0&0&0&0&1&1&1&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&1&1&1&1 \\ 2&2&2&2&4&4&4&4 \\ 2&2&2&2&4&4&4&4 \\ 2&2&2&2&4&4&4&4 \\ 2&2&2&2&4&4&4&4 \end{matrix}$$

INPUT A　　INPUT B

Fig.3.10 入力に用いた多値画像



(a) A　　(b) A＋B

Fig.3.11　ディジタル/アナログ変換法による実験結果

このディジタル／アナログ変換法は各画素値が強度変調を受けているため出力画像の表示に適している。しかし，この手法を用いて連続的に処理を行う場合にはアナログ出力像から再度ビット分散画像を得るための新しいアナログ／ディジタル変換法が必要となる。もし，このアナログ／ディジタル変換法が開発できれば桁上げの問題は解決され，簡単な構成の多値画像処理システムが実現できる。

3.3.3　多値符号法

並列光論理演算法で用いる符号パターンはFig.1.3(c)に示した4種類のものである。これは論

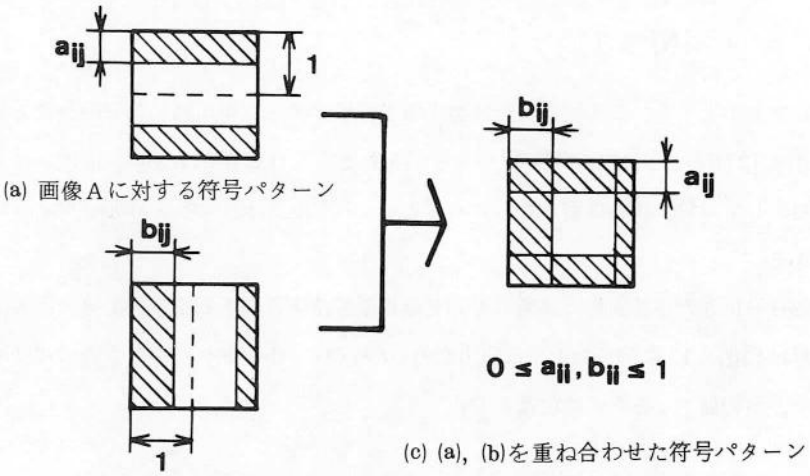


Fig.3.12　多値符号化

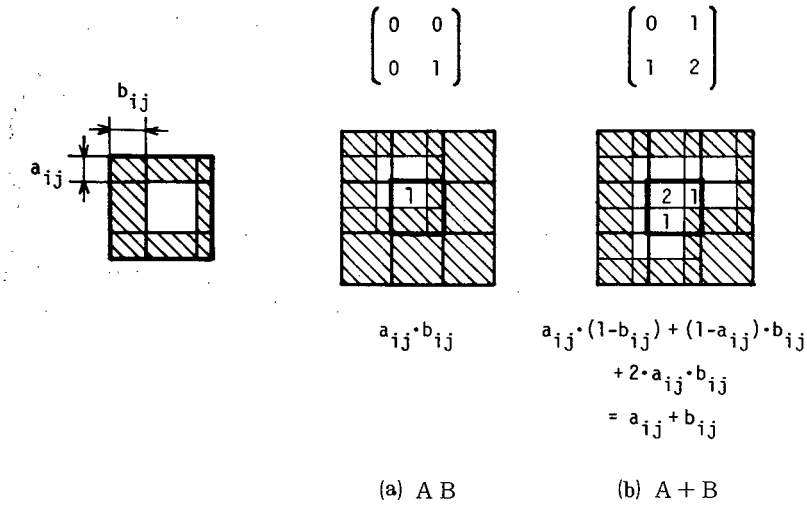


Fig.3.13　多値符号法の出力信号

理演算で取扱うデータが0, 1の2値しかとらないことによる。そこで，符号パターン自体を多値データの表現が可能な形に拡張すれば効率的な多値データ処理法の実現が期待できる。この考えに基づいた多値画像処理法を多値符号法と呼ぶ。

Fig.3.12に多値符号化の手順を示す。被処理データ a_{ij}, b_{ij} はそれぞれ

$$0 \leqq a_{ij} \leqq 1 \tag{3-2}$$

$$0 \leqq b_{ij} \leqq 1 \tag{3-3}$$

を満たすものとする。この時，各データの値を符号パターン中の透明部分のシフト量に変換するとFig.3.12(a)および(b)の符号パターンが得られる。これはFig.1.3(a), (b)のレベル1の符号化像に対応しており，両者を重ね合わせることによりFig.3.12(c)の多値符号パターンを得ることができる。

多値符号化した多値画像は通常の並列光論理演算法実行の手順を用いて処理する。この時，出力信号はFig.3.13に示したように出力窓内（太枠内）で空間的な拡がりを持つ強度の分布を持つ。そこで，出力値 c_{ij} を次式で定義する。

$$c_{ij} = \int_{S_{ij}} I(S)\, dS. \tag{3-4}$$

ここで，$I(S)$ は S における光信号強度，S_{ij} は (i, j) 出力窓の領域を表わす。この定義を用いるとFig.3.13の下に記した出力値が得られる。Table 3.5は多値符号法で実行できる演算の例

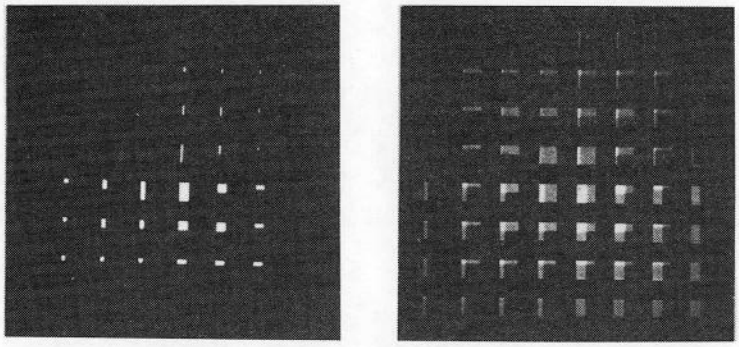
Table 3.5　多値符号法で用いるカーネルの例

KERNEL	FUNCTION	KERNEL	FUNCTION	KERNEL	FUNCTION
	0		(1-A)(1-B)	$\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$	A + B
	AB		1-(A+B)+2AB	$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$	A - B + 1
	A(1-B)		1-B	$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$	B - A + 1
	A		1-B+AB	$\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$	2 - A - B
	(1-A)B		1-A	$\begin{pmatrix} 0 & b \\ a & a+b \end{pmatrix}$	aA + bB
	B		1-A+AB	$\begin{pmatrix} b & 0 \\ a+b & a \end{pmatrix}$	aA-bB+b
	A+B-2AB		1-AB	$\begin{pmatrix} a & a+b \\ 0 & b \end{pmatrix}$	bB-aA+a
	A+B-AB		1	$\begin{pmatrix} a+b & a \\ b & 0 \end{pmatrix}$	(a+b)-aA-bB

をカーネルと共にまとめたものである。表からわかるように，この多値符号法は特に並列算術演算に適していると言える。

Fig.3.14 は多値符号法を用いた処理結果である。入力画像はFig.3.10 に示したもので，これを多値符号に変換して処理を行った。Fig.3.14(a)がＡＢ，(b)がＡ＋Ｂの出力信号である。

多値符号法には複雑な符号化の操作，厳密な光学系設定および調節，並列光検出器を必要とす



(a) ＡＢ　　(b) Ａ＋Ｂ

Fig.3.14　多値符号法による実験結果

る欠点がある。しかし，別な見方をすれば，それだけ高密度な情報を符号パターンが持ち，処理系自体の情報処理能力を十分に使用していると言える。その点で，この多値符号法は光による高密度演算の能力を引き出していると考えられ，新しい並列光演算体系を実現しているということができる。

3.4 結　言

並列光論理演算法をもとに画像処理を目的とした新しい並列光演算法をいくつか考案した。3.2節では，2値画像を対象した演算法として並列光ディジタルフィルタリングと並列光論理テンプレートマッチングについて述べ，それぞれの処理結果を示した。3.3節では，多値画像処理のため諸手法を説明し，処理結果を示した。本章で示した各演算法は並列光論理演算法の手順や処理を一部変更して実現できたもので並列光論理演算法の自由度の大きさを示すものである。光学情報処理技術の1つの特長としてディジタル量とアナログ量が同一の処理系で取扱えることがあげられるが，本章の各演算法はその特長を十分にいかした方法でもある。ディジタル演算とアナログ演算の使い分けは光演算技術を発展させる上で不可欠なものと考えられる。

第4章　光アレイロジック

4.1　緒　　言

光学情報処理技術は大容量情報の超高速・超並列演算の有力な手段を提供している。たとえば2次元相関演算や2次元フーリエ変換はその最もよい例である。しかし，これらの演算以外のものは光学情報処理では実現が難しく，その柔軟性の乏しさが長い間この技術の応用範囲を限定してきた。

一方，現在隆盛を誇っているディジタル電子計算機はディジタル演算方式による処理の柔軟性を特長とし，その柔軟性ゆえに，様々な分野で利用されるようになっている。ディジタル演算方式を用いると任意の処理はいくつかの基本操作（コマンド）の組合せ（プログラム）で記述でき，そのプログラムを遂行して所望の処理を達成することができる。処理効率さえ問題にしなければ殆んどすべての処理がディジタル計算機で実現できる。

1章で述べた並列光論理演算法は光による並列情報処理とディジタル演算方式双方の特徴を兼備えたものであり，簡単な光学処理系を使用して並列2変数2値論理演算や並列シフト演算等，演算内容の選択を行うことができる。この方法を用いると並列演算の種類は符号化画像との2次元相関演算に用いるカーネルの指定で簡単に選択することができる。この選択の簡便さが先に提案した並列光論理演算法の持つ大きな特長の1つとなっている。

さらに重要な特長として著者は並列光論理演算法がエレクトロニクス分野において開発・応用されているアレイロジック[39,40]と密接な関係を持つことを見出した。アレイロジックは入力信号を符号化し，その符号化信号の組合せにより任意の論理演算を実現する技術である。この考え方は扱うデータの並列性という点を除けば並列光論理演算法の基本的な考え方と同一のものである。そこでこの対応関係を積極的に活用すれば，並列光論理演算法の応用範囲の飛躍的な拡大が期待できると思われる。

本章では，アレイロジックを並列光論理演算法による並列近傍画素間論理演算の取扱い可能な形に拡張した新しい演算原理 ── 光アレイロジック ── について説明する。この光アレイロジックの概念を用いると任意の並列情報処理をプログラムによって簡単に実行することができ，光演算による汎用並列処理の具体的な手段を得ることができる。4.2節では，新しく提案する光アレイロジックの概念を説明し，並列光論理演算法による実現方法を述べる。さらに4.3節では，それを実際の問題に適用した例を示し，その有用性を確証する。

4.2　光アレイロジックの概念

並列光論理演算法における各画素ごとの論理演算実行の手順はアレイロジックの実行手順と1対

1に対応する。すなわち，並列光論理演算法の画素信号の符号化がアレイロジックの入力信号のデコード* に対応し，符号化画像に対して行う2次元相関演算のカーネルがアレイロジックのデコード信号の組合せに対応する。この関係を積極的に利用すれば，アレイロジックを用いて，並列光論理演算法による近傍画素間論理演算を記述できる。

アレイロジックで表わせる論理演算はある1つの画素に対するもので次式で表現できる。

$$c_{ij} = R(\mathbf{a}_{ij}, \mathbf{b}_{ij}). \tag{4-1}$$

ここで，ベクトル$\mathbf{a}_{ij}$，$\mathbf{b}_{ij}$は被処理画像AおよびB内の(i, j)画素a_{ij}，b_{ij}とそれらの近傍画素であり，$R(\mathbf{x})$は論理変数列$\mathbf{x}$に対する論理関数，c_{ij}は出力画像中の(i, j)画素を表わす。注意すべきことは，SIMD(single instruction multi-data flow)形式の並列演算の場合，(4-1)式だけで並列演算を記述できることである。すなわち，SIMD形式の演算はある代表画素に対する演算だけで，全体の並列演算を記述できる。したがって，アレイロジックで表現した近傍画素間演算は，並列光論理演算法による並列近傍画素間論理演算を表現するものである。

近傍画素間論理演算はセルラロジックとも呼ばれ，並列画像処理の基礎概念としてディジタル画像処理の分野で活発に研究されている。[51,52] このセルラロジックの利用により各種の画像処理をはじめとする2次元データ処理が可能となる。したがって，並列近傍画素間論理演算は，2次元データの汎用処理の基本演算といえ，アレイロジックを通して並列光論理演算法で実現できることは汎用並列光ディジタル演算の手段を獲得したことに等しい。このように，画素単位の論理演算をアレイロジックで表わし，その集合体として並列論理演算を光学的に実現する技術をここでは光アレイロジックと呼称する。

4.2.1 アレイロジックの拡張

アレイロジックは入力信号を符号化し，その結果得られた符号化信号の組合せにより任意の論理回路を実現する技術である。[39,40] これは集積型半導体記憶素子と同等のアレイ構造を持つ回路（ロジックアレイと呼ぶ）により構成される。

Fig.4.1は半加算回路をランダムロジックとアレイロジックとによって表わしたものである。半加算回路の機能は次の論理式で表現できる。[53]

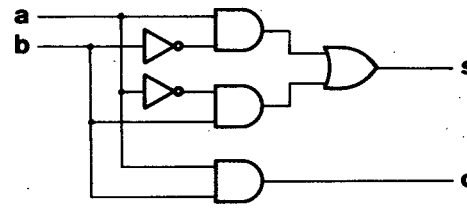
$$s = a\bar{b} + \bar{a}b \tag{4-2}$$

$$c = ab. \tag{4-3}$$

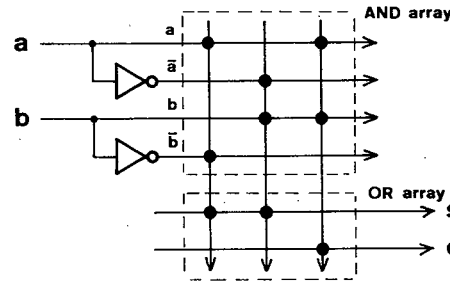
Fig.4.1からわかるように，(a)のランダムロジックはAND, OR, NOTの論理ゲートとそれらの

---

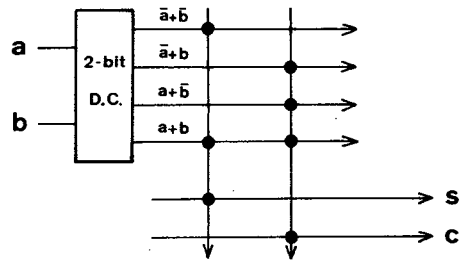
*アレイロジックにおける‘デコード’（復号）が並列光論理演算法の符号化に対応する。混同を避けるため，アレイロジックの場合には‘デコード’と表記する。



(a) ランダムロジックによる表現



(b) アレイロジックによる表現（１ビットデコーダ）



(c) アレイロジックによる表現（２ビットデコーダ）

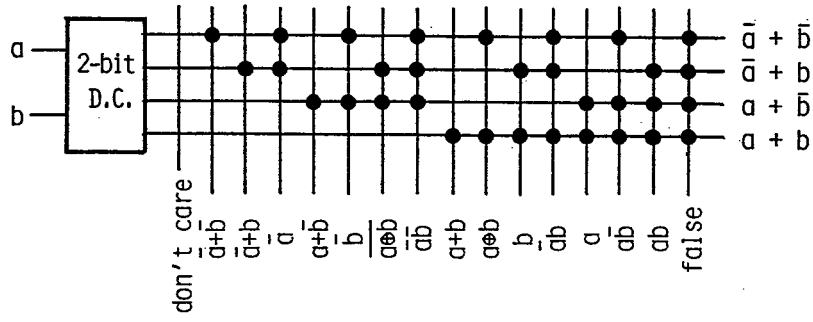
Fig.4.1　半加算回路

間の配線により（4−2），（4−3）式の論理を実現している。一方，(b)のアレイロジックでは入力信号 a，b をそれぞれ反転信号 $\overline{a}$，$\overline{b}$ と非反転信号 a，b にデコードし，デコード信号間の論理積（AND演算）をとって積項を求め，さらにその積項間の論理和（OR演算）をとることによって所望の演算結果を得ている。ここで，積項とは論理変数 p_i とその否定 $\overline{p}_i$ の論理積で表現される論理関数である。一般に，任意の論理関数は積項の論理和として表現されるので，アレイロジックによって任意の論理関数が実現できる。図中のANDアレイ内で積項演算が，ORアレイ内で論理和が行われ，各アレイ回路内の黒丸が積項演算および論理和の対象を示す。アレイロジックでは，これらの組合せ方を変えて所望の論理を設計する。

Fig.4.1(c)は同じアレイロジックでも入力２変数を１組にしてデコードする２ビットデコード

a	b	$\bar{a}+\bar{b}$	$\bar{a}+b$	$a+\bar{b}$	$a+b$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

(a) デコード信号の真理値表



(b) デコード信号の組合せで可能な論理演算

Fig.4.2　2ビットデコーダ

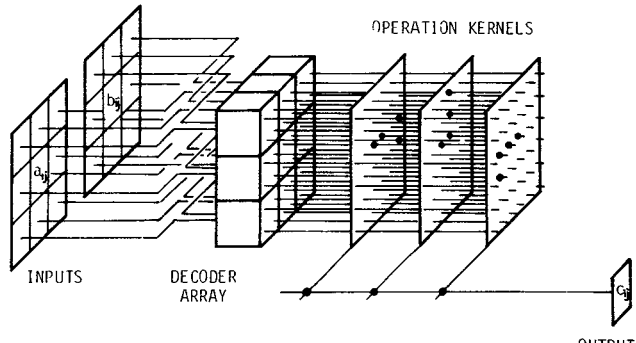
方式のものである。この場合，デコード信号に対する1回の論理積で入力2変数に対する16種類の2値論理関数すべてが実現できる。Fig.4.2はその様子を示したもので，(a)が2ビットデコーダ出力の真理値表，(b)はそれらによって得られる関数を示す。ここで注目すべきことは，1章で述べた並列光論理演算法の各画素の演算手順がこの2ビットデコーダを用いたアレイロジックの演算原理と非常によく対応することである。すなわち，入力画像の画素の符号化は2ビットデコーダの機能に，符号化画像とカーネルの相関演算は積項演算に対応する。これらの事実を踏まえ，2ビットデコーダを用いたアレイロジック（以下では単にアレイロジックと呼ぶ）の概念を並列光論理演算法に適合する形に拡張することは自然であると考えられる。

並列光論理演算法の並列処理がSIMD形式であることは既に述べた。そこで，2つの被処理画像の代表画素とその近傍画素 $\mathbf{a}_{ij}$，$\mathbf{b}_{ij}$ 間の論理演算を取出し，アレイロジックで表現すれば全体の並列処理をうまく記述することができる。しかも，記述された並列処理は並列光論理演算法により容易に実行することができる。したがって，以下では代表画素に対する演算のアレイロジック化を考える。

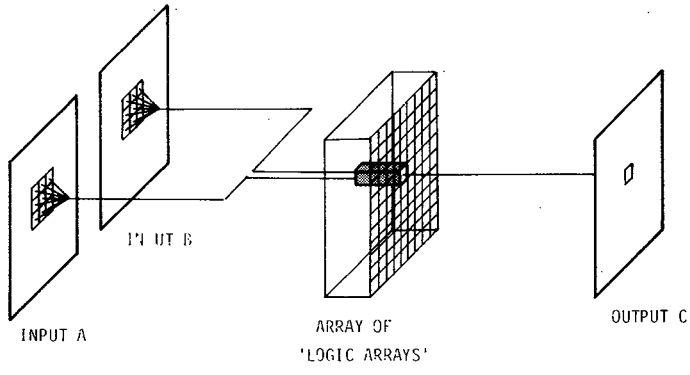
代表画素に対する演算の入力変数は $\mathbf{a}_{ij}$，$\mathbf{b}_{ij}$ であり，出力は c_{ij} である。$\mathbf{a}_{ij}$，$\mathbf{b}_{ij}$ の範囲は必要とされる処理により決められるが，ここでは a_{ij}，b_{ij} とそれらの8隣接近傍画素を仮定する。

これらの入力変数は画像を構成する画素そのものであり２次元的広がりを持つため，Fig.4.3(a)に示す３次元的ロジックアレイを考える。この３次元的ロジックアレイは２つの入力画像の対応する画素を１組としてデコードし，その４つのデコード信号を正方形状に配列して出力する。Fig.4.3(a)では近傍領域を８近傍画素としているが，近傍領域の拡張はデコーダの増設により可能である。ANDアレイは図に示すように立体構造を持ち，積項演算を指定する黒点は積項面上の２次元パターンとなる。この２次元パターンにより任意の近傍画素間の積項演算が指定できるため，光アレイロジックでは演算の記述にこの２次元パターンを用いる。なお，詳細は4.2.2項で述べる。さらに，この２次元パターンは並列光論理演算法の出力をdark-true論理で解釈することにより並列光論理演算法におけるカーネルとまったく等価にでき，その結果，光アレイロジックで記述した演算を光学的に実行することができる。これについては4.2.3項で述べる。所定の積項演算で得た結果に対する論理和が近傍画素間論理演算の結果を与える。

光アレイロジックは画素単位の演算を行うアレイロジックの２次元的な集合体として構成される。Fig.4.3(b)はその様子を示したものである。１つの画素に対する処理はFig.4.3(a)の３次元的ロジックアレイで表現され，それと同種のロジックアレイが被処理画像の全画素に対して並列



(a) (i, j) 画素に対するロジックアレイ



(b) 画像全体における(a)のロジックアレイの位置

Fig.4.3 光アレイロジックの概念

に存在する。各ロジックアレイでの演算はすべて同じであり，それは積項演算を指定する２次元パターンで決定される。これらはすべて並列に動作するのでＳＩＭＤ形式の並列処理を行うことになる。

4.2.2 演算処理の指定

光アレイロジックの演算の種類はすべて３次元的ロジックアレイ内の積項面上の２次元パターンによって定義できる。すなわち，任意の近傍画素間論理演算は近傍画素間積項演算の論理和の形に展開でき，したがって，ある論理演算を構成する積項演算の実行に必要な２次元パターンの組として，その論理演算を定義することができる。ここで積項面上の２次元パターンを演算カーネルと呼び，Fig.4.4に示す格子点上のパターンで表わす。

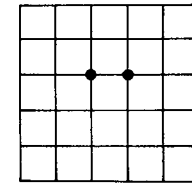


Fig.4.4　演算カーネルの表記法

このように，演算カーネルは光アレイロジックと直接結びつき，さらに4.2.3項で述べるように並列光論理演算法による実現においても重要な役割をはたす。しかし，単なる点パターンとしてではその機能がわかりにくく，プログラムにおいても不便が予想される。そこで，より実際の処理に則した記述法を決めておく。ここで示す記述法と演算カーネルの関係は電子計算機におけるニーモニックコードとマシンコードの関係に対応しており，光アレイロジックによるプログラムの簡便さを目的とするものである。

演算カーネルはFig.4.3(a)に示すように正方形状に配列した２ビットデコーダからのデコード信号線と積項面の交点の集合として得られる。１つの２ビットデコーダからの信号は近傍画素領域内のある対応画素の組（たとえば，$a_{i,j+1}$と$b_{i,j+1}$）の情報しか含まず，したがって，演算カーネルを２ビットデコーダの信号ごとに分解すれば，対応画素の組ごとの演算に分けることができる。Fig.4.2(b)に示すように，２ビットデコーダの出力信号の組合せにより，２変数２値論理関数をすべて実現できるが，ここでの場合，正方状に配列したパターンの組合せが演算を指定する。このパターンは，1.3節で考えたカーネルユニットと等価なものである。Table 4.1はパターンと論理演算との関係をまとめたものである。演算カーネルが表わす近傍画素間積項演算は，このパターンが示す各画素ごとの２変数２値論理演算の結果の論理積として得られる。すなわち，その関係は

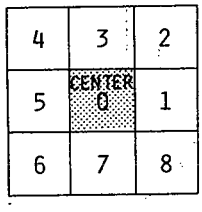


Fig.4.5　８隣接近傍画素

$$c_0 = f_0(a_0, b_0) f_1(a_1, b_1) f_2(a_2, b_2) f_3(a_3, b_3) f_4(a_4, b_4) \cdot f_5(a_5, b_5) f_6(a_6, b_6) f_7(a_7, b_7) f_8(a_8, b_8) \qquad (4-4)$$

と表現できる。ここで，fは任意の２変数２値論理関数であり，各

Table 4.1 光アレイロジック用カーネルユニット

KERNEL UNIT	FUNCTION	SYMBOL	KERNEL UNIT	FUNCTION	SYMBOL
	1	..		a+b	PP
	$\bar{a}+\bar{b}$	NN		$a \oplus b$	UU
	$\bar{a}+b$	NP		b	.1
	$\bar{a}$	0.		$\bar{a}b$	01
	$a+\bar{b}$	PN		a	1.
	$\bar{b}$	.0		$a\bar{b}$	10
	$\overline{a \oplus b}$	EE		ab	11
	$\bar{a}\bar{b}$	00		0	DD

変数の添字はFig.4.5に示す近傍画素の位置を示す。

今，各画素ごとの演算をTable 4.1に併せて記したシンボルで表わし，[54] 対象画素の幾何学的配置に対応した位置に置いた表記法を考える。各シンボルは2つの文字よりなり，前が入力変数 a，後が b に対応する。演算の内容は文字が表わし，AND演算の場合は1，0，. を用い，OR演算ではP，Nを用いる。また，1，Pはtrueを0，Nはfalseを示し，. はdon't careを意味する。ただしEE，UU，DDは特別でそれぞれEQU，XOR，禁止を示す。この表記法により1つの近傍画素間積項演算の記述ができる。いくつかの表記例を次に示す。

例1　$c_0 = a_0\, b_0$　(4-5)

$$\Rightarrow \begin{bmatrix} .. & .. & .. \\ .. & 11 & .. \\ .. & .. & .. \end{bmatrix} \qquad (4\text{-}6)$$

例2　$c_0 = a_0 \bar{a}_1 \bar{a}_3 \bar{a}_5 \bar{a}_7$　(4-7)

$$\Rightarrow \begin{bmatrix} .. & 0. & .. \\ 0. & 1. & 0. \\ .. & 0. & .. \end{bmatrix} \qquad (4\text{-}8)$$

例3　$c_0 = (a_0 + b_0)(a_1 \oplus b_1)$　(4-9)

$$\Rightarrow \begin{bmatrix} .. & .. & .. \\ .. & \text{PP} & \text{UU} \\ .. & .. & .. \end{bmatrix} \tag{4-10}$$

例3は厳密には積項演算ではないが，光アレイロジックでは積項演算として取扱う。

一般に任意の論理演算は複数の積項演算の論理和として表わすことができる。したがって，光アレイロジックの記法でも演算カーネルの論理和として表現する。その結果，任意の論理演算が記述できる。

例4　$c_0 = a_0 b_0 + a_0 \overline{a}_1 \overline{a}_3 \overline{a}_5 \overline{a}_7$　(4-11)

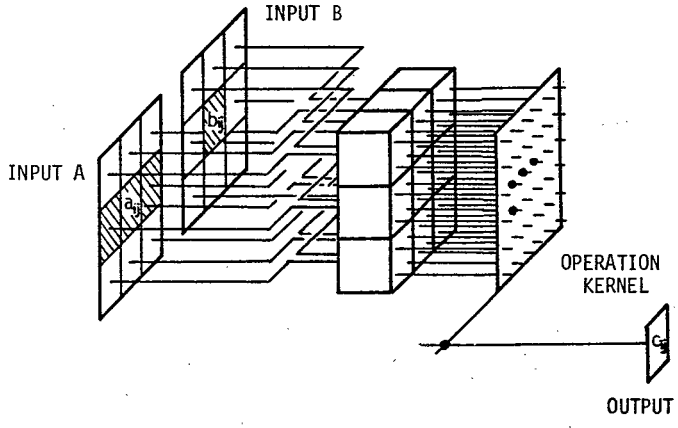
$$\Rightarrow \begin{bmatrix} .. & .. & .. \\ .. & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 0. & .. \\ 0. & 1. & 0. \\ .. & 0. & .. \end{bmatrix} \tag{4-12}$$

以上の表記法は代表画素に対する演算を記述するものであるが，光アレイロジックのSIMD構造を考えあわせれば，これらの表示は並列近傍画素間論理演算をも記述していることになる。したがって，上の表示は一種の並列論理の記述式とみなすこともできる。さらに特筆すべきことは，これらの表記法はTable 4.1，Fig.4.5の関係より演算カーネルに変換でき，4.2.3項で述べるように並列光論理演算法で直接実行が可能なことである。以上の理由により，光アレイロジックの記述には上述の表記法を用いる。なお，演算カーネルの大きさは対象とする近傍領域の範囲により決定される。上述の例では3×3の近傍領域を考えたが，近傍領域が大きくなる時にはその範囲に対応した配置でTable 4.1のシンボルを置くことで記述することができる。このように，この表記法を用いると処理の種類，近傍領域の大きさなどの記述に普遍性を持たすことができる。

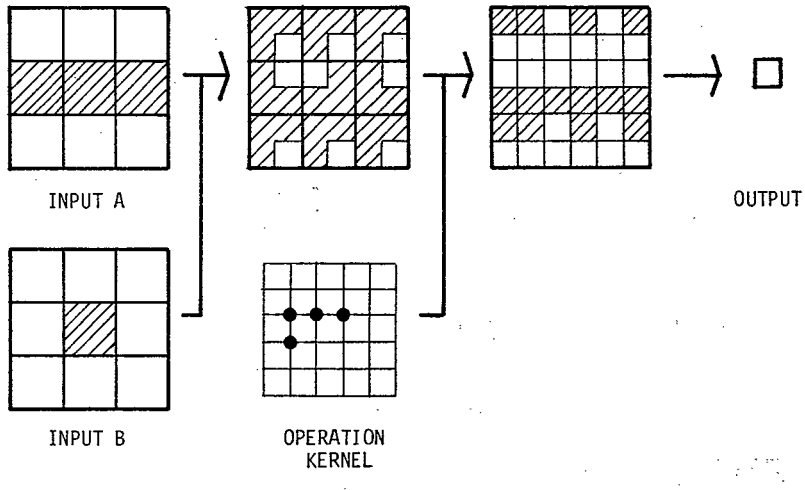
4.2.3　並列光論理演算法による実現

4.2.1項および4.2.2項ではおもに光アレイロジックの概念について説明した。光アレイロジックは並列処理に適応した演算技術であるが，最大の特長は並列光論理演算法を用いることによって，光学的に容易に実行できることである。Fig.4.6は画素単位の演算に対する3次元的ロジックアレイとそれに対応する並列光論理演算法の概念図である。

さきに述べたように並列光論理演算法における画素ごとの符号化の操作は3次元的ロジックアレイの画素ごとの2ビットデコードに対応し，相関演算に用いるカーネルは演算カーネルに対応している。ここで，1.3項で述べたように，並列光論理演算法での出力信号をdark-true 論理で解釈すればアレイロジックと1対1対応した演算を実現できる。すなわち，(1-32)式で $k =$



(a) 3次元的ロジックアレイ



(b) 並列光論理演算法

Fig.4.6 3次元的ロジックアレイと並列光論理演算法の対応

0, $l=0$ の場合を考えると

$$[\overline{w}_{00}+(a_{mn}+b_{mn})][\overline{w}_{01}+(a_{mn}+\overline{b}_{mn})][\overline{w}_{10}+(\overline{a}_{mn}+b_{mn})]$$
$$\cdot[\overline{w}_{11}+(\overline{a}_{mn}+\overline{b}_{mn})] \qquad (4-13)$$

が得られるが，この式の各項は a_{mn}, b_{mn} を入力とする2ビットデコーダの出力信号であり，Fig.4.2(a)に示すものと等価なものである。したがって，w_{00}, w_{01}, w_{10}, w_{11} の組合せにより任意の論理演算が実現できることがわかる。また，(k, l) が複数組存在する場合には(1-32) 式の各項はAND演算として接続され，(4-4) 式の関数を実現できる。その結果，近傍画素を対象とした積項演算が行えることがわかる。ただし，得られた結果は dark-true 論理で表わされ

るため，この結果に対する論理和は暗信号を検出するものとなる。

以上の論理演算の結果を求める手順は次の通りである。

1. 対象とする論理演算をいくつかの積項演算に分解し，それぞれを符号化画像と演算カーネルとの相関演算により実行する。
2. 各相関画像のうちで1回でも暗信号が出力された画素は対象とする論理演算に対し'1'の値を持つ。その他の場合には'0'の値を持つと解釈する。
3. 各相関演算の出力信号に対して，暗信号の検出を行う。

Fig.4.7は並列光論理演算法による光アレイロジック実現の概念図である。入力画像データに対して完全並列に処理を行える様子がわかる。2次元相関演算を効率よく実行するには演算カーネルを動的に制御する必要がある。この目的には2.3節で示した種々の方法が利用できる。出力画像は演算系の接続性を考え，入力画像と同じbright-true論理に変換する。この部分は光アレイロジックにおける論理和の実現方法に依存する。

一般に，近傍画素間論理演算の実行には複数の近傍画素間積項演算が必要であり，できればそれらを同時に実行することが望ましい。この場合，光アレイロジックの演算能力は最大限に発揮される。しかし，実際にシステムを構成する場合，現時点で利用し得る光機能素子を考えあわせると同時実行方式は必ずしも最善の方法とは言えない。なぜなら，それを実現するには使用素子数，素子の使用効率，システムの大きさ等を考慮しなければならないが，現在のところこの目的にかなう特性を持つ光機能素子が存在しないからである。そこで，本論文では積項演算は逐次的

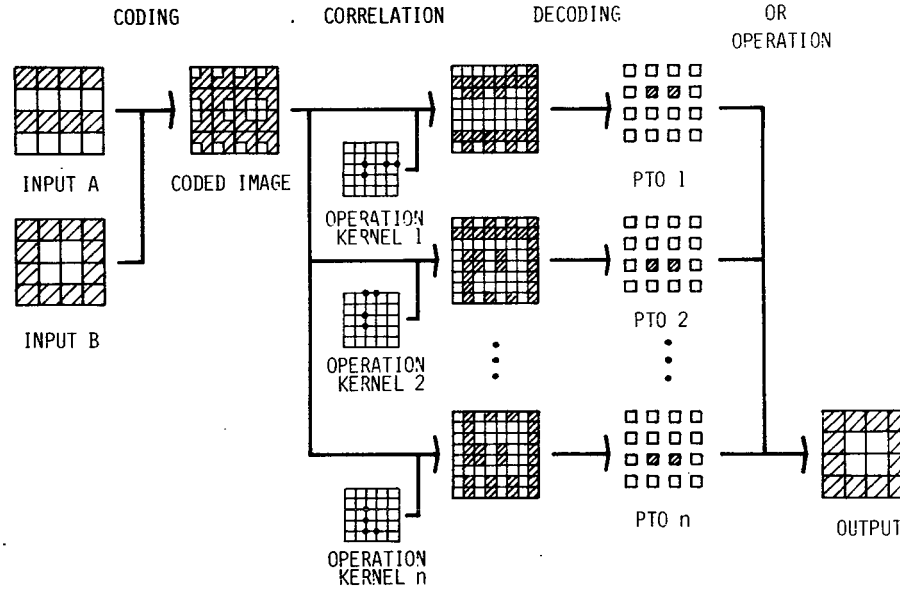


Fig.4.7 並列光論理演算法による光アレイロジック

に行うものと仮定して議論を進める。積項演算同時実行方式の可能なシステム例は6.3.2項で示す。

4.3 光アレイロジックによる並列処理

本節では，4.2節で述べた光アレイロジックの概念に基づいた処理例を示す。光アレイロジックで実行可能な処理は2値画像データに対する並列近傍画素間演算である。そこで，4.3.1項では画像データを対象とする処理例として画像のエッジ点検出を取上げ，その処理方法・処理結果を示す。また，光アレイロジックは被処理データの配置を工夫することにより画像以外のデータ処理にも適用することができる。すなわち，2進数で表わされた多値データを3.3.1項のビット分散配置法を利用して処理する方法である。この具体的な応用例として，4.3.2項では並列2ビット加算器を取上げて説明する。さらに桁上げを生じない特長を持つため，並列処理に適した演算方式として注目されている剰余演算系の光アレイロジックによる実現法を4.3.3項で述べる。

なお，本節では2.3.1項の投影光学系と写真技術を応用して光アレイロジックを実行した。写真技術はdark-true論理の信号に対する論理和の操作を実現するために用いた。すなわち，各積項演算に対する相関画像出力をリスフィルムに個別に記録し，それらを重ねあわせて観察する方法をとった。得られた複数枚の相関画像のうちで1度でも暗信号が出力された画素は黒くなり，論理値1を表わす。すなわち，出力はdark-true論理として得られる。もちろん，これらの処理は処理画像内の全画素に対し完全並列に実行される。

4.3.1 エッジ検出

光アレイロジックの概念を用いると入力画像データに対して種々の論理演算を実行することができる。ここでは1例としてN×N画素から成る2値入力画像のエッジ画素の検出処理を取上げる。エッジ画素とはある図形の輪郭を構成する画素で，その抽出は画像強調や画像理解のための前処理などにとって重要である。

エッジ画素抽出法としては各種のテンプレートと被処理画像との相関を用いたテンプレートマッチング法が有名である。[47] ここでは光アレイロジックの特徴を生かした近傍画素間論理演算によるエッジ画素抽出法について述べる。この方法は他の処理にも応用が可能であり，拡張性が高いものである。

まず，処理を記述する論理関数を求める。ここで，エッジ画素とは次の条件を満たすものとする。

1. 着目画素 a_0 の値は1.
2. a_0 の4隣接近傍画素 a_1，a_3，a_5，a_7 のうち少なくとも1つが0.

この時，次式を満足するものがエッジ画素である。

$$c_0 = a_0(\overline{a}_1 + \overline{a}_3 + \overline{a}_5 + \overline{a}_7). \tag{4-14}$$

次に，(4－14) 式を光アレイロジックで実現可能な形，すなわち，近傍画素間積項演算の論理和の形式に変換する。

$$c_0 = a_0\overline{a}_1 + a_0\overline{a}_3 + a_0\overline{a}_5 + a_0\overline{a}_7. \tag{4-15}$$

ここで，式の変形には論理関数の分配則[31]を用いた。(4－15) 式は光アレイロジックの記法では

$$\begin{bmatrix} .. & .. & .. \\ .. & 1. & 0. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 0. & .. \\ .. & 1. & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ 0. & 1. & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ .. & 1. & .. \\ .. & 0. & .. \end{bmatrix} \tag{4-16}$$

と表現でき，この各項に対応した演算カーネルにより順次相関演算を実行すれば，所望の結果が得られる。

Fig.4.8 は投影光学系による処理結果である。(a)は 16×16 の画素により構成された入力画像である。この入力画像を符号化したものを 18mm×18mm の大きさでリスフィルム上に記録し，(4－16) 式の各項の演算カーネルに対応した発光パターンを持つＬＥＤアレイで照明した。(b)～(e)

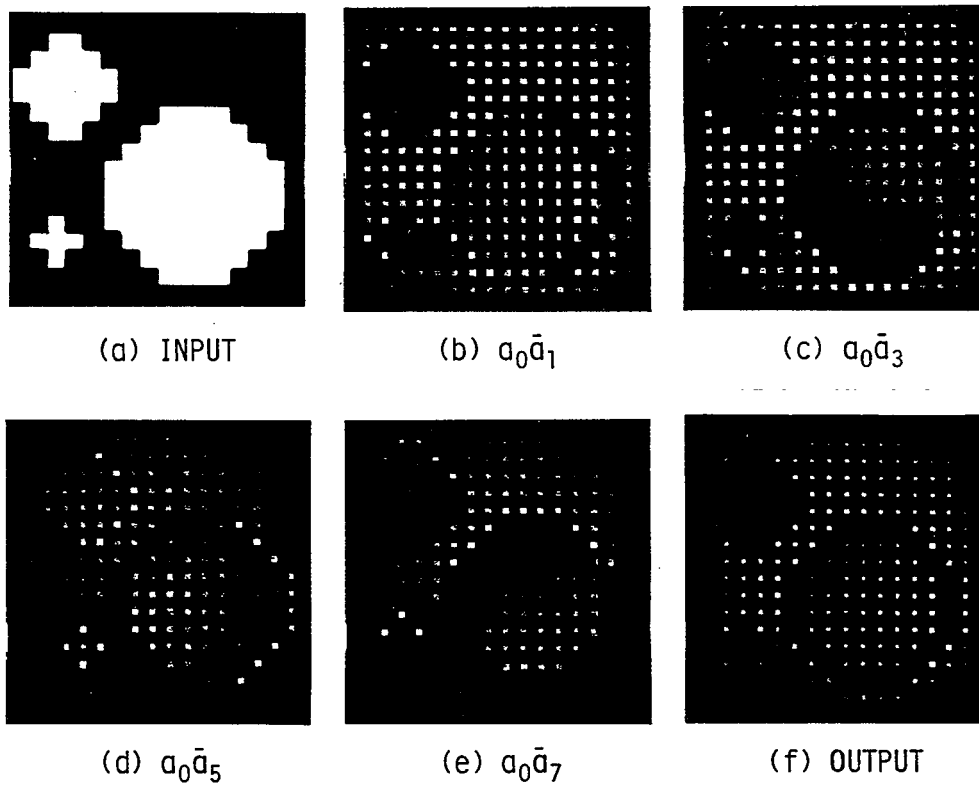


Fig.4.8 エッジ検出処理の実験結果（Ｉ）

が各積項演算の結果である。(4－15) 式の各積項の条件を満たす画素が順次，暗信号として得られている。(f)は(b)～(e)の結果に対して論理和を行った結果である。(4－15) 式を満足する点，すなわちエッジ画素が暗信号として得られている。

(4－14) 式は別の方法でも実現可能である。すなわち，(4－14) 式の否定をとり，負論理により処理を行っても実現できる。(4－14) 式の否定は De Morgan の定理[31]を用いて次式で表現できる。

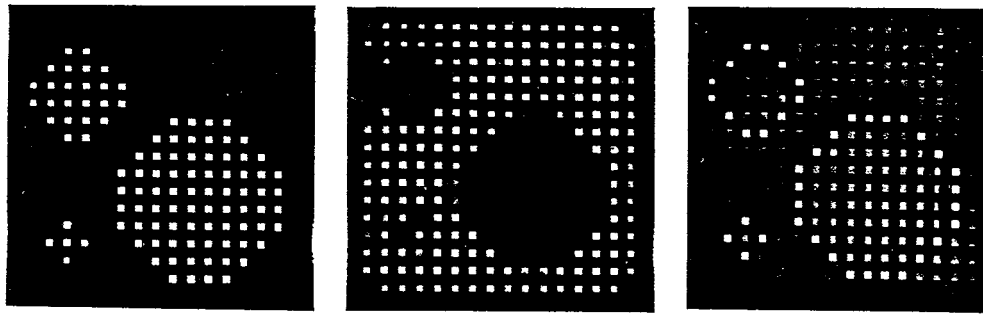
$$\overline{c}_0 = \overline{a}_0 + (\overline{\overline{a}_1 + \overline{a}_3 + \overline{a}_5 + \overline{a}_7}) = \overline{a}_0 + a_1 a_3 a_5 a_7 . \qquad (4-17)$$

この式は光アレイロジックでは

$$\begin{bmatrix} .. & .. & .. \\ .. & 0. & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 1. & .. \\ 1. & .. & 1. \\ .. & 1. & .. \end{bmatrix} \qquad (4-18)$$

となる。Fig.4.9 は (4－18) 式にしたがって行った実験結果である。(a)，(b)は (4－17) 式の各項の演算結果で，(c)はそれらに対する論理和の結果である。この場合，非エッジ画素が暗信号(1)として得られ，(4－15) 式をもとにして行った結果の反転画像が出力されている。

このエッジ抽出の例では負論理による方法〔(4－18) 式〕の方が正論理による方法〔(4－16) 式〕にくらべ，積項演算の数を少なくすることができる。一般に，どちらの論理が効率的であるかは実行させる論理関数に依存するが，光アレイロジックはどちらの方法にも対応できる自由度を有していることがわかる。



(a) $\bar{a}_0$　　(b) $a_1 a_3 a_5 a_7$　　(c) OUTPUT

Fig.4.9　エッジ検出処理の実験結果(II)

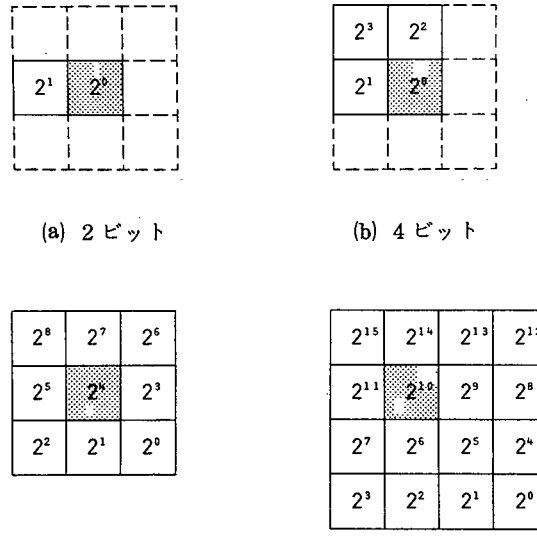
4.3.2　並列2ビット加算器

光アレイロジックの処理対象は2次元離散データである。これは画像データ以外に，2次元平

面上にパターンとして表現された数値データをも処理対象に含んでいることを意味する。ディジタル処理において数値データは，0, 1 のビット符号の集合として表現されるため，それをパターンとして表わしても何ら問題はない。しかも，この場合，同一平面上に置かれた数値データは同時並列に処理され，その並列度を上げることにより実効的に単位データ当たりの処理速度を上げることができる。したがって，光アレイロジックは画像だけでなく一般の数値演算にも利用することができる。この場合，数値データを表現するパターンのとり方により種々の応用に適用が可能となる。3.3.1 項のビット分散配置法による多値画像処理もこの技術を応用したものである。パターンの選択により表現できる数値のビット数は任意に選べ処理の自由度は高い。ここでは，2 ビット数を処理する系として 2 ビット加算器を取上げ，その実現方法を述べる。

数値データを光アレイロジックで取扱うには，まず数値を表現するパターンを決める必要がある。すなわち，パターンを構成するセルの数とその配置である。ここで，セルとは 2 値画像データにおける画素に対応するものである。セルの数は数値データのビット数に等しく，要求される仕様を満足するようにとる。セルの配置は任意であるが，近傍画素間論理演算と数値データに対する処理との関係に大きく影響を持つので，ある程度，配置法を決めておく方が望ましい。Fig. 4.10 は 2, 4, 9, 16 ビット数の配置例を示したものである。本論文ではこの図に示す配置法を用いるものとする。ここでは，2 ビット数値を処理するので Fig.4.10 (a) の配置パターンを用いる。

数値データをパターンに変換した後は，4.3.1 項で示した手順により光アレイロジックを設計する。すなわち，2 ビット加算器は



(a) 2 ビット　(b) 4 ビット

(c) 9 ビット　(d) 16ビット

Fig.4.10　多ビット数値のビット配置法

$$s^0 = a^0 \oplus b^0 \tag{4-19}$$

$$s^1 = (a^1 \oplus b^1)(\overline{a}^0 + \overline{b}^0) + (\overline{a^1 \oplus b^1})\, a^0 b^0 \tag{4-20}$$

$$s^2 = a^1 b^1 + (a^1 + b^1)\, a^0 b^0 \tag{4-21}$$

なる論理式で表現できるため,[53] これらを光アレイロジックの記述式に変換すればよい。ここで, a, b が入力データ, s が出力データであり, 肩文字はビットの位置を示す。今, Fig. 4.10(a)のパターン配置を用いるので, 入力データの各ビットと光アレイロジックにおける代表画素およびその近傍画素との間には次の関係がある。

$$a^0 = a_0 \tag{4-22}$$

$$a^1 = a_5 \tag{4-23}$$

$$b^0 = b_0 \tag{4-24}$$

$$b^1 = b_5 . \tag{4-25}$$

そこで, (4-19)～(4-21)式は次の近傍画素間論理演算に変換される。

$$s^0 = a_0 \oplus b_0 \tag{4-26}$$

$$s^1 = (a_5 \oplus b_5)(\overline{a}_0 + \overline{b}_0) + (\overline{a_5 \oplus b_5})\, a_0 b_0 \tag{4-27}$$

$$s^2 = a_5 b_5 + (a_5 + b_5)\, a_0 b_0 . \tag{4-28}$$

これらは光アレイロジックの記述法では

$$S^0 : \begin{bmatrix} \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \cdot\cdot & \mathrm{UU} & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot \end{bmatrix} \tag{4-29}$$

$$S^1 : \begin{bmatrix} \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \mathrm{UU} & \mathrm{NN} & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot \end{bmatrix} + \begin{bmatrix} \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \mathrm{EE} & 11 & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot \end{bmatrix} \tag{4-30}$$

$$S^2 : \begin{bmatrix} \cdot\cdot & \cdot\cdot & \cdot\cdot \\ 11 & \cdot\cdot & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot \end{bmatrix} + \begin{bmatrix} \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \mathrm{PP} & 11 & \cdot\cdot \\ \cdot\cdot & \cdot\cdot & \cdot\cdot \end{bmatrix} \tag{4-31}$$

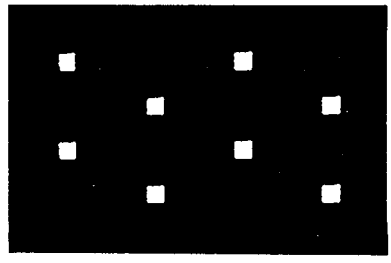
となり, (4-26)～(4-28)式は直接光学系で実現できることがわかる。ただし, 2ビット加算器の場合, 出力は S^0, S^1, S^2 の3つであり, これらを同時に出力するためには3組の処理系が必要となる。*

---

* 6.2.3項で述べる波長多重並列光論理演算法を利用すれば, 効率的な処理系を実現できる。

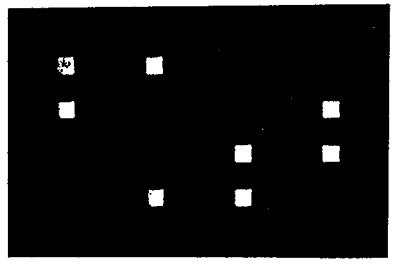
00	01	10	11	00	00	00	00
00	01	10	11	01	01	01	01
00	01	10	11	10	10	10	10
00	01	10	11	11	11	11	11

(a) INPUT A, B



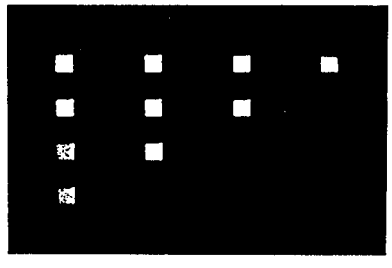
0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0

(b) OUTPUT S^0



0	0	1	1
0	1	1	0
1	1	0	0
1	0	0	1

(c) OUTPUT S^1



0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1

(d) OUTPUT S^2

Fig.4.11 並列２ビット加算器の実験結果

Fig.4.11 (a)は光アレイロジックによる2ビット加算器の動作を確認するために用いたデータである。4×4，計16個の2ビット数を2組用意し，それらを入力データとして加算した。Fig.4.11 (b)～(d)はその結果である。(b)～(d)はそれぞれS^0, S^1, S^2の結果を示す。ここでの各出力信号はdark-true論理で表示されている。写真の右側には得られるべき値を記してあり，実験データとこの数値を比較すると，光アレイロジックによって所望の結果が得られていることがわかる。

4.3.3 並列剰余演算器

剰余演算系は加減乗法に対し，桁上げを生じないという特性を持っており並列処理に適した演算方式として注目されている。[55-57] 剰余演算系では，各数値は法と呼ばれる互いに素である正整数の組で対象とする数値を割った時の余りの組として表わされる。

法をr個の正整数m_i ($i=1, \cdots, r$)とすると，剰余系では

$$0 \leqq x \leqq \prod_{i=1}^{r} m_i - 1 \tag{4-32}$$

なる範囲の整数xを取扱うことができる。(4-32)式より，rおよびm_iが大きいほど広い範囲の数値を取扱える。

整数xはr個の剰余の組で表現する。

$$(\mathrm{mod}(x/m_1), \cdots, \mathrm{mod}(x/m_r)). \tag{4-33}$$

ここで，$\mathrm{mod}(a/b)$は正整数aを正整数bで割った時の剰余を表わす。各剰余については

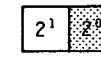
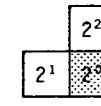
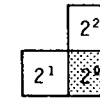
$$0 \leqq \mathrm{mod}(x/m_i) \leqq m_i - 1 \tag{4-34}$$

なる関係がある。

この剰余系は加・減・乗法について閉じているため，各剰余数ごとに処理を行うことができ，しかも各処理系は独立しているため並列的に実行可能である。もし各演算の結果が(4-32)式の範囲内にあれば10進数に変換することができる。

今，例として3，5，7を法とする剰余演算系を光アレイロジックにより構成することを考える。この場合，それぞれの法による剰余数を異なる3つの光アレイロジックで処理する。(4-34)式からわかるように，各法に対する光アレイロジックでは2ビットないし3ビットの数値の演算を行う必要が生じる。そこでFig.4.12に示すセル配置を用いて各処理系を設計する。

Fig.4.13は法3,5,7の剰余数に関する加法演算規則表である。例として，法3に関する加法の出力値をビットごとに論理関数で表わすと次式となる。

DECIMAL SYSTEM	RESIDUE SYSTEM	BINARY CODED RESIDUE SYSTEM	CODE PATTERN
	$a_3 = \mathrm{mod}(a/3)$	$a_3^1\, a_3^0$	
a	$b_3 = \mathrm{mod}(b/3)$	$b_3^1\, b_3^0$	
	$a_5 = \mathrm{mod}(a/5)$	$a_5^2\, a_5^1\, a_5^0$	
b	$b_5 = \mathrm{mod}(b/5)$	$b_5^2\, b_5^1\, b_5^0$	
	$a_7 = \mathrm{mod}(a/7)$	$a_7^2\, a_7^1\, a_7^0$	
	$b_7 = \mathrm{mod}(b/7)$	$b_7^2\, b_7^1\, b_7^0$	

($0 \leqq a,\ b \leqq 3 \cdot 5 \cdot 7 - 1$)

Fig.4.12 剰余演算処理用セル配置法

Modulus 3

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Modulus 5

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Modulus 7

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

Fig.4.13 法3，5，7の剰余加法規則表

$$s_3^0 = \overline{a}^1 \overline{b}^1 (a^0 \oplus b^0) + a^1 b^1 \tag{4-35}$$

$$s_3^1 = \overline{a}^1 \overline{b}^1 a^0 b^0 + (a^1 \oplus b^1)\, \overline{a}^0 \overline{b}^0 . \tag{4-36}$$

ここで，下付き文字は法，肩文字はビット位置を示す。これらを光アレイロジックで記述すると，

$$S_3^0 : \begin{bmatrix} .. & .. & .. \\ 00 & UU & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ 11 & .. & .. \\ .. & .. & .. \end{bmatrix} \tag{4-37}$$

$$S_3^1: \begin{bmatrix} .. & .. & .. \\ 00 & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ UU & 00 & .. \\ .. & .. & .. \end{bmatrix} \tag{4-38}$$

他に法に対する加法も同様にして表現できる。法5に対する加法は次式で表わせる。

$$S_5^0: \begin{bmatrix} .. & 00 & .. \\ NN & UU & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ UU & 00 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ 11 & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 11 & .. \\ .. & .. & .. \\ .. & .. & .. \end{bmatrix} \tag{4-39}$$

$$S_5^1: \begin{bmatrix} .. & 00 & .. \\ UU & NN & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ UU & UU & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ 00 & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} . & 11 & .. \\ .. & .. & .. \\ .. & .. & .. \end{bmatrix} \tag{4-40}$$

$$S_5^2: \begin{bmatrix} .. & 00 & .. \\ UU & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ 11 & 00 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ 00 & 00 & .. \\ .. & .. & .. \end{bmatrix}. \tag{4-41}$$

また，法7に対する加法は次式で表現できる。

$$S_7^0: \begin{bmatrix} .. & 00 & .. \\ .. & UU & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 11 & .. \\ .. & EE & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ 11 & 00 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ 00 & UU & .. \\ .. & .. & .. \end{bmatrix}$$

$$+ \begin{bmatrix} .. & UU & .. \\ UU & 11 & .. \\ .. & .. & .. \end{bmatrix} \tag{4-42}$$

$$S_7^1: \begin{bmatrix} .. & 00 & .. \\ UU & NN & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 00 & .. \\ EE & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & .. & .. \\ UU & 00 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ 00 & 11 & .. \\ .. & .. & .. \end{bmatrix}$$

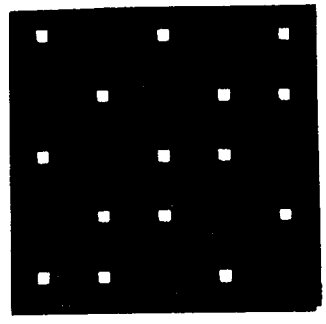
$$+ \begin{bmatrix} .. & UU & .. \\ 11 & UU & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 11 & .. \\ 00 & PP & .. \\ .. & .. & .. \end{bmatrix} \tag{4-43}$$

$$S_7^2: \begin{bmatrix} .. & UU & .. \\ 00 & .. & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & EE & .. \\ 11 & .. & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & 00 & .. \\ UU & 11 & .. \\ .. & .. & .. \end{bmatrix} + \begin{bmatrix} .. & UU & .. \\ UU & 00 & .. \\ .. & .. & .. \end{bmatrix}$$

$$+ \begin{bmatrix} .. & 11 & .. \\ UU & UU & .. \\ .. & .. & .. \end{bmatrix}. \tag{4-44}$$

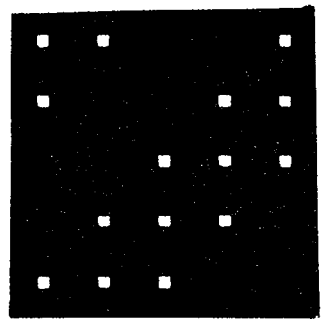
0 0 0 0 0	0 1 2 3 4
1 1 1 1 1	0 1 2 3 4
2 2 2 2 2	0 1 2 3 4
3 3 3 3 3	0 1 2 3 4
4 4 4 4 4	0 1 2 3 4

(a) INPUT A, B



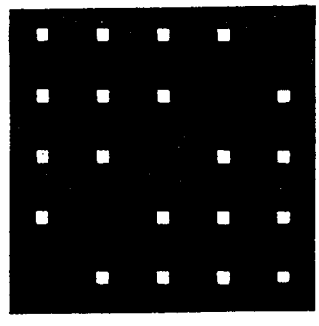
0	1	0	1	0
1	0	1	0	0
0	1	0	0	1
1	0	0	1	0
0	0	1	0	1

(b) OUTPUT S_5^0



0	0	1	1	0
0	1	1	0	0
1	1	0	0	0
1	0	0	0	1
0	0	0	1	1

(c) OUTPUT S_5^1



0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

(d) OUTPUT S_5^2

Fig.4.14 法５の並列剰余加算の実験結果

Fig.4.14(a)は5×5の数値データで，これを光アレイロジックによる剰余処理系の入力データとして用いた。Fig.4.14(b)~(d)は5を法とする剰余系加算の光学系による出力結果である。写真の右に得られるべき値を付記してあり，所望の結果が得られていることがわかる。もちろん，減法，乗法等の他の演算や他の法に対する処理も演算カーネルやセル配置を変えることにより実現できる。

4.4 結　言

光アレイロジックの最大の特長は並列処理の記述が簡単にでき，その実行が光学系を用いて容易に行えることである。光アレイロジックの概念を利用すると任意の並列論理演算の実行に並列光論理演算法が適用可能となる。この光アレイロジックは光学処理の最大の利点である大容量データの並列演算とディジタル演算方式の利点である処理の柔軟性とをあわせ持つ新しい演算方式を提供するものである。4.2節では、この新しい光アレイロジックの概念およびその記述法，実現法について説明した。ここで示した論理体系は'論理'と呼ぶにははなはだ不完全なものではあるが4.3節の結果からもわかるように，実用的見知からは何ら不足のないものとなっている。何にもまして，光アレイロジックにより記述された並列論理演算が実際の相関光学系によって簡単に実行できることは光アレイロジックの最大の特長となっている。このように論理体系と直接結びつき，しかも具体的実行手段を有する光演算法は他に類を見ない。4.3節では，光アレイロジックが実際の処理に対してどのように適用されるかを示した。光アレイロジックの処理対象は2次元2値データであるが，数値データ等もパターン化することにより取扱うことができる。この考え方は汎用演算処理に対する光アレイロジックの適用可能性を示唆するものであり，事実，4.3節におけるいくつかの例で，その適用性を示した。さらに，光アレイロジックによる演算は入力画像上のすべてのデータに対して完全並列に行われるため，従来とはまったく異った新しい演算形態への発展性もあわせ持っているように思われる。このように，光アレイロジックは実際の処理に対して具体的な並列演算方式を提供するものであり，今後，広範な分野での応用が期待される概念である。

第5章　並列光演算システム：OPALS

5.1　緒　　言

現在の高度情報化社会においては電子計算機のはたす役割は極めて大きく，科学技術計算をはじめ各種情報処理，システム制御，データベース管理等々，どの応用分野をとって見ても電子計算機なしではそれらの機能はまったくはたせないと言っても過言ではない。しかし，その反面，計算・処理に対する需要は電子計算機の利用分野の拡大と共に爆発的に増加しており，電子計算機の部分的改良だけでは対処しきれないという問題も明白になっている。この問題解決への期待は特に科学技術計算分野から強く，それに応えるべく，従来の方式とは異った演算原理に基づく大容量情報の高速演算システムの開発研究が精力的に進められている。

特に，科学技術分野で要望の強い大容量情報の計算の特徴として，2次元あるいは多次元配列データを対象とするものが多いことがあげられる。そのため配列データを効率的に処理し得る新しい大容量情報の高速演算方式の開発が期待されており，様々の方式のものが検討されている。その中でも光の物理的特性である高速並列情報処理・伝送能力を生かした演算方式は最も有望なものの1つとして注目をあびている。

光の並列特性を利用した情報処理システムの構想は比較的古く，1977年に提案されたTseコンピュータに端を発する。[23] Tseコンピュータの'Tse'とは'字'の中国音を表わしたもので，表意文字である漢字，すなわち図形を対象としたコンピュータという意味である。Tseコンピュータは2次元平面上のデータに対し，膨大な数の論理ゲートを並列動作可能なように配列し，ファイバー束による並列伝送路でそれらを接続して構築される。このような並列処理系実現の手法は電子回路技術によって蓄積された計算機のアーキテクチャやソフトウェアなどの資産をそのまま継承できるという点では有利であるが，世界最初のディジタル電子計算機ENIACでさえ20,000個もの論理ゲート素子を必要としたことを考慮すれば，[53] この方法は非現実的なものと言わざるを得ない。Tseコンピュータの意義は光学処理にディジタル演算の概念を持ち込んだ点にある。

現在，光演算システム研究の主たる対象はディジタル光演算方式のシステムであり，すでにいくつかの有望なシステムが提案されている。たとえば，ベル研究所のHuangは単一機能並列論理ゲートとインターコネクションユニットによる古典的有限状態マシンの提案を行い，[24] この思想は南カリフォルニア大学のSawchukらにより発展されて実動する光順序論理回路にまで至っている。[42,58] また，同じくHuangにより可変機能論理ゲートを利用した光パイプラインプロセッサが提案されている。[27] これらのシステムはいずれも光の強力な並列情報伝送能力を生かしたもので，光機能ゲートアレイと光インターコネクション系により構成されている。システム構成および動作は個々のシ

ステムで異なるが，いずれもTseコンピュータの問題点を踏まえ，効率的なシステム構成，高い発展性等を有しているのが特徴である。

著者も同じディジタル演算方式の思想に基づき，光の並列情報処理能力を生かした汎用並列ディジタル光演算システムOPALS(optical parallel array logic system)を提案し，具体的なシステム化の方法を発表した。[9,10] OPALSは4章で述べた光アレイロジックの特長を生かしたシステムで，2次元データに対する汎用並列ディジタル処理を目的としたものである。OPALSの特徴は，並列ディジタル演算方式の概念を基本に光の強力な情報伝送能力を有効に利用していること等は前述のシステムと共通しているが，さらに具体的なシステム化の方法とそれを用いた並列情報処理の方法が明確に確立されている点があげられる。これは光アレイロジックによる問題解法の手順がそのままOPALSでの処理手順に対応し，具体的な演算の実行ができるからである。他のシステムが演算システム構築のためにこれから複雑なインターコネクション素子を設計し，具体的な処理方法を確立しなければならないことを考えれば，OPALSの研究ではすでにそれらの問題は解決済みであり，システム構築の具体的な段階にまで進んでいるという大きな特徴点を持っている。

以下，5.2節では，OPALS開発の目的，その機能上の特徴をあげ，その概念を記述する。5.3節，5.4節では実際に考案したOPALSの具体的なシステム構成方式を2種類説明する。これらは光/電子複合型OPALSと純光学型OPALSで，前者は従来の計算機と純光学型OPALSの中間形態的役割を持ち，後者は将来の新しい並列光演算システム — 光コンピュータ — の候補となり得るものである。

5.2 並列光演算システム — OPALS — の概念

5.1節でも述べたように新しい演算原理に基づく大容量演算システムの開発は，現在，大容量高速情報処理を必要とする分野が最も待ち望んでいるものである。本論文の1章から4章までに説明してきた並列光論理演算法および光アレイロジックはその扱えるデータ量，処理の柔軟性などにおける自由度の大きさから種々の情報処理分野への応用が期待される。そこで，その並列演算能力を十分に生かした汎用演算システム開発への活用は最も有望なものである。著者はこの観点より光アレイロジックの概念を演算原理の中心として新しい汎用並列ディジタル光演算システムOPALSを開発した。

Fig.5.1はOPALSのシステム構成を概念的に示したものである。システムは2値画像データを処理対象とし，光アレイロジックによる処理を反復的に実行する。OPALSは画像データの符号器，演算カーネルの動的制御可能な2次元相関器，dark-true 論理データに対する論理和を実行するOR演算器，出力データの復号器の4つのモジュールとそれらを接続する並列伝送路から構成される。

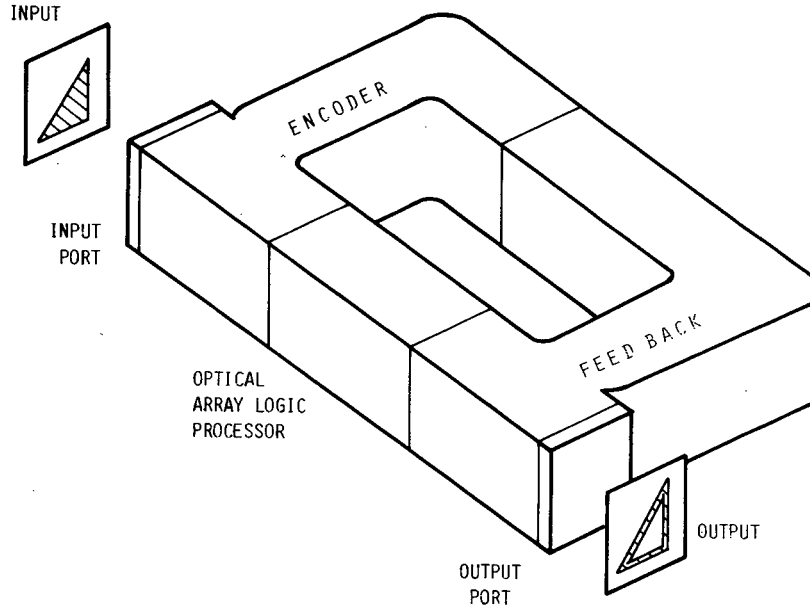


Fig.5.1　OPALSの概念図

OPALSの機能上・動作上の特徴を列挙すると，

1. 2次元2値データに対する並列画素間演算・並列近傍画素間演算の実行が可能。
2. データ入出力，演算処理，記憶，表示，すべてを2次元データの配列を保ったまま実行する。
3. 光アレイロジックによるプログラムが可能で，汎用並列演算システムとしての機能を備えている。
4. 並列帰還系により同一データに対する反復処理が可能。

以下，各項について補足説明をする。

第1の特徴はOPALSの演算原理の中核である光アレイロジックの性質により得られたものである。4.2節で述べたように，1つの演算カーネルが近傍画素間積項演算に対応するため，その演算カーネルを自由に制御できる2次元相関器を用いれば，任意の近傍画素間積項演算が実行できる。さらにその積項演算の結果に対してOR演算を行うと近傍画素間論理演算の結果を得る。

第2の特徴はTseコンピュータの思想を受継ぐものであり，並列光情報処理システムOPALSの最も強調すべき点である。OPALSを構成する各モジュールは画素単位の機能要素を2次元並列に配置したもので，符号器，OR演算器，復号器のいずれにおいても処理は各画素単位で行われる。これらの処理では近傍画素との接続が不要であり，画素単位の機能要素の集合体としてモジュールを構成することができる。2次元相関器は複雑な並列情報伝送を必要とするが，光学系を利用すれ

ばきわめて簡単に実現することができる。帰還系および各モジュール間の信号接続は結像系で並列に行う。このようにOPALS内のすべての情報処理および信号伝送は並列性を保ったまま実行される。

第3の特徴は光アレイロジックの動的制御・実行が可能なシステム構成を取ることによって生じるものである。4.2節で述べたように，任意の並列近傍画素間演算は光アレイロジックの簡単な記述法により設計することができる。並列近傍画素間演算は画像処理の分野で応用されているセルラロジックの概念[51,52]を包含するため，画像データを対象とする処理であれば簡単にプログラムすることができる。各プログラムは演算カーネルの組合せで記述されるため，プログラムは演算カーネルを時系列あるいは並列に制御できる実時間2次元相関器を用いて逐次実行できる。さらに，4.3.2項および4.3.3項で示したように，任意の数値データの2次元パターン化により汎用演算への利用も可能となる。

第4の特徴はOPALSがループ型の構成を持つシステムとして実現されることによって生ずる。画像データ処理を対象とするセルラロジックでは同一種類の演算の反復実行が要求されるが，OPALSはそれを実現し得るシステム構成を持っている。この反復処理では入力画像データに対する近傍画素間演算の結果を次段の演算の入力データとして用い，同様の処理を繰返し実行する。セルラロジックを用いた画像処理に関しては多くの研究がなされているため，その成果をOPALSでより高速に実現することも可能である。

Fig.5.2はOPALS内の信号の流れを描いたものである。光アレイロジックでは2つの入力画像に対して処理を行い，1つの画像を出力するが，各データの置き場所として並列レジスタを使用する。並列レジスタは被処理画像を一連の近傍画素間積項演算の実行中保持し，OR演算の結果が得られた後，書換えられる。OPALSでは光アレイロジックの出力信号を次の演算の一方の入力信号として帰還する構成をとっている。したがって，Fig.5.2中のレジスタBの内容が処理の進行と共に変化していく。OPALS外部とのインターフェースは入力ポート，出力ポートを通して行われる。

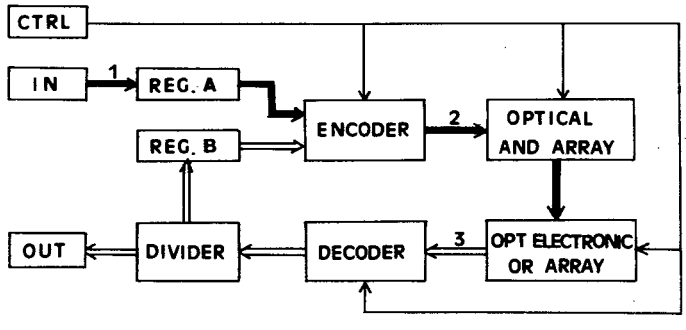


Fig.5.2 OPALS内の信号の流れ

入力ポートは光アレイロジックの残りの入力（レジスタA）と直結されており，反復処理途中のデータと外部から入力したデータとの間の演算が可能であり多様な処理にも対応できる。処理結果はレジスタBへの帰還出力の一部を分配器を介して常にモニターしながら出力ポートで得る。

以上のOPALSにおける処理手順をまとめると次の様になる。

1. レジスタA（入力ポート）より被処理データをレジスタB（反復レジスタ）へ転送する。
2. 別のデータが処理に必要であればレジスタA（入力ポート）に入力する。
3. レジスタAとレジスタBのデータに対し近傍画素間積項演算を行い，OR演算器へ転送する。
4. OR演算器では送られてきたデータと内部のデータとのOR演算を実行し，結果を内部に保存する。
5. 必要回数，3.および4.を繰返す。
6. OR演算器の内部データをレジスタBおよび出力ポートへ転送する。OR演算器の内部データをクリアする。
7. 必要ならば1.へ戻る。

OPALSの具体的な実現には種々のシステム構成法が考えられる。OPALSの中核である光アレイロジックの構成には2章で述べた各種の符号化法と実時間2次元相関演算法の技術が利用できる。しかし，OR演算を実行する方法は新たに考えなければならず，各レジスタ用にも並列記憶素子が必要となる。また，光アレイロジックにおける積項演算の実行方式についても考えなければならない。4.2.3項で述べたように，現時点で利用できる光機能素子の種類とその能力の制限により，本論文では逐次積項演算実行方式を仮定するが，新しいOPALSを設計する場合には同時実行方式も考慮する必要がある。そこで，これらの問題解決法を含めて，OPALSの具体的な構成例を2種類考案した。1つは符号化，OR演算，記憶等光学処理では実現が難しい非線形処理を電子処理系で実行する光/電子複合型システムである。他は未開発の特殊機能を持つ光機能素子を利用する純光学型システムである。いずれもOPALSの1つの実現可能な形態のシステムであって最終的なシステムではないが，OPALSの基本機能を有している。

5.3 光／電子複合型OPALS

光学処理系によって光アレイロジックに要求される各種の処理を実現するには複雑な系の構成が必要である。一般に光学処理は符号化，記憶，逐次信号に対するOR演算等の非線形処理に不向きであり，それを克服するために種々の技術を導入しなければならない。一方，電子回路技術を利用する電子処理系は非線形処理を容易に実行することができる。これは，電子回路ではトランジスタやダイオード等の非線形素子が簡単に使用できるからである。しかし，電子処理系は信号伝送の制

限より大容量信号の複雑なパターンの接続が難しいという欠点を持つ。そこで，光学処理に適していない処理のみを電子処理系で実行する光 / 電子複合処理方式のシステムが考えられる。この光 / 電子複合型OPALSは現時点では最も実現が容易なOPALSであり，システムの性能評価の目的にも有用なものである。

5.3.1 システム構成

Fig.5.3に光 / 電子複合型OPALSの１つの概念図を示す。OPALSを構成するモジュールのうち実時間２次元相関器のみを光学系で構成し，他は液晶アレイ，フォトダイオードアレイをインターフェース素子とした電子処理系で実現する。電子処理系は画像レジスタ，符号器，符号化画像表示器（液晶アレイ素子），復号器兼光学信号取込器（フォトダイオードアレイ），ＯＲ演算器により構成する。各モジュールは画素単位の機能要素の並列の集合体であり，電子処理系の弱点である複雑なパターンの配線を回避したものとなっている。

Fig.5.3には (i, j) 画素に対する電子処理系の信号の流れのみを描いてある。この図より明らかなように，個別の電子処理系では近接した画素との信号通信は一切必要としない。まず，着目画素データ（a_{ij} および b_{ij}）を符号化し，光学系に入力する。次に，光学系によって行った相関演算結果の対応点の光学信号を取込み，遂次ＯＲ演算を実行する。フィードバック系を構成するために，ＯＲ演算の出力は符号化器の入力端子に接続してある。各処理は光アレイロジック実行の手順にしたがって制御する。

光学処理系は符号化画像に対して実時間２次元相関演算を実行する。使用できる光学技術は電

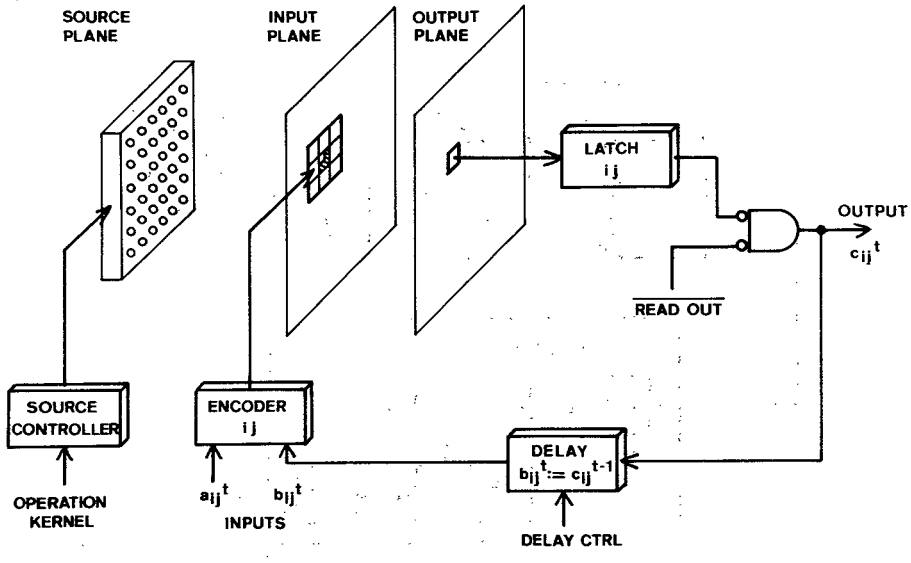


Fig.5.3　光 / 電子複合型OPALS

子処理系とのインターフェース素子により制約を受ける。たとえば，Fig.5.3に示すシステムの場合，透過型液晶アレイ素子を電子処理系から光学系への入力インターフェース素子として用いるため，2.3.1項の多重投影系が最も簡単でかつ制御が容易である。しかし，インターフェース素子として発光型や反射型素子を利用する場合には他の2次元相関系を利用する必要がある。したがって，場合に応じて使用する光学系を選択しなければならない。

この光/電子複合型OPALSの性能は電子処理系の能力により決定される。特に，インターフェース素子の果たす役割は大きく，光/電子複合型システムの存在意義自体を左右する。変換効率が悪ければ，わざわざ複合システムを作る必要はないからである。光学系への入力素子としては液晶やBSO空間光変調素子をはじめ，GaAs等の発光素子が，光学信号の取込みにはpinフォトダイオード，CCD素子等が利用できる。[29,59] いずれもOPALSで処理する画像の画素数だけの素子が必要であり，高集積化ができることも不可欠な条件である。先に述べたように，電子処理系は画素ごとの処理要素の並列集合体であるので，インターフェース素子を含め高集積化ができれば実用的システムとしても十分利用可能である。特に，複合型OPALSは将来の高集積素子として期待される3次元LSI(large scale integrated circuit)[60,61] の主要な活用分野としても十分検討の余地があろう。

5.3.2 実験システム

OPALSによる並列処理の動作確認を目的として光/電子複合方式の予備実験システムを試作した。Fig.5.4は試作システムのブロック図である。電子処理系はパーソナルコンピュータ PC-9801F2（日本電気製）およびフレームメモリ IMAGE-PC（エディック製）を中心にし，

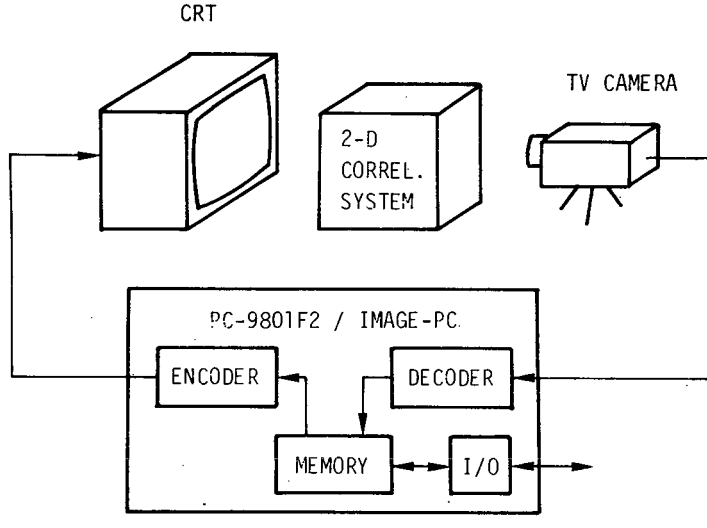


Fig.5.4 予備実験システム

CRTモニタとTVカメラをインターフェース素子として用いた。光学系は多重結像レンズを使用する。

この予備実験システムでは電子処理系は並列ではなく時系列的に逐次動作する。すなわち，パーソナルコンピュータを駆動するソフトウェアによって処理を実現している。したがって，完全並列処理というOPALSの最大の特徴が失われ，OPALS本来の機能を発揮することはできない。しかし，5.3.1項で述べたような並列電子処理系は，たとえ予備システムと言えども膨大な数の回路素子やインターフェース素子が必要なこと，OPALSによる並列処理の動作確認の目的には速度の遅いシステムでも十分なこと，処理系の変更が自由にできることなどを考えあわせれば，Fig.5.4のシステムは予備実験用としては妥当なものである。

予備実験として，CRT上の画像を結像系を介してTVカメラでとり込み，反復処理のシュレーションを行った。光学処理系が行うべき実時間2次元相関演算は，符号化画像を種々のシフト量で多数重ねて，CRT上に表示することで実現した。ここで，符号化画像の重ね方が演算カーネルのパターンと対応する。復号はフレームメモリ内の特定アドレスの内容のみを取り出すことによって行った。

実行した処理は，初期値として与えた画素を含む4連結領域の抽出である。この処理は次のアルゴリズムで実現できる。

1. (object) → A
2. (initial pixel) → B^0
3. $b_0^t = a_0 b_0^{t-1} + a_0 a_1 b_1^{t-1} + a_0 a_3 b_3^{t-1} + a_0 a_5 b_5^{t-1} + a_0 a_7 b_7^{t-1}$
4. 必要回数，3.を繰返す

ここで，レジスタAの内容は変化せず，レジスタBの内容が処理と共に変化して行く。そこで各時刻におけるBの内容を示すために肩文字を付記してある。

Fig.5.5に処理結果を示す。対象画像と反復1，2，3，5，8，14回目のレジスタBの内容を表わしている。反復1回目の出力は，初期値として与えた画素を示している。この処理は反復14回目以後，定常状態に入った。所望の結果が得られていることがわかる。1回の反復に要する時間は約50秒であった。大部分は符号化画像のCRT表示に費されたもので，この処理の場合，1回の反復中に23回(演算カーネル内のドット総数)の表示が必要なためである。しかし，光学的相関演算が利用できれば，符号化画像の表示は1回だけで済み，大幅な処理時間の短縮が期待できる。

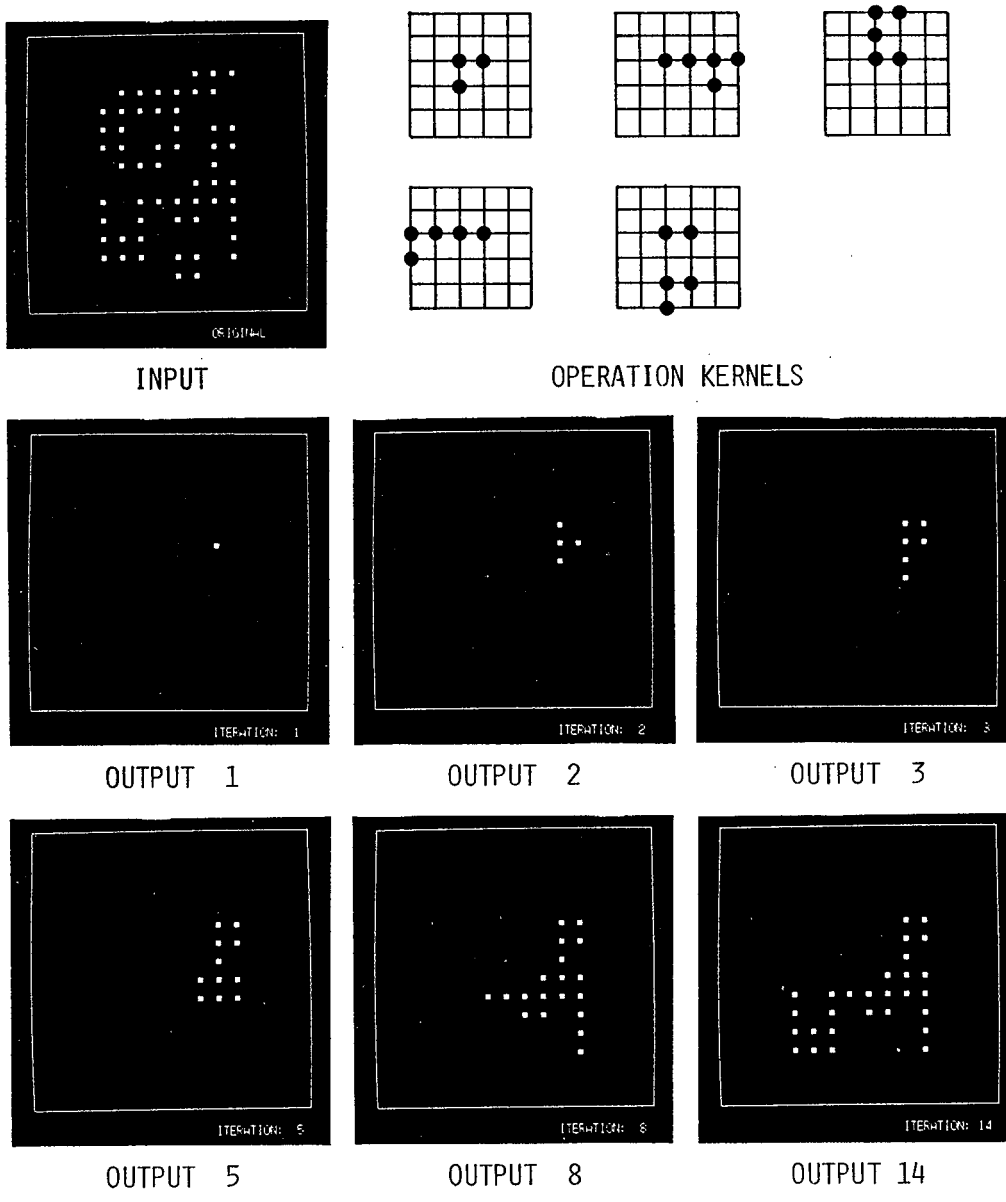


Fig.5.5　4連結領域の抽出

5.4　純光学型OPALS

光／電子複合型OPALSは光学処理系と電子処理系，それぞれの長所を生かしたシステムで興味深い。しかし，並列光演算システムの将来性という見地からは光／電子変換（インターフェース）素子の利用は必ずしも望ましいものではない。なぜなら，インターフェース素子でのエネルギーの消費や時間の浪費に対する何らかの補償が要求されるからである。また，光の情報処理の潜在能力

は非常に大きいのにもかかわらず，２次元相関演算への利用だけでは光の持つ並列情報処理・伝送能力を十二分に発揮しているとは言えないからである。以上のことを考えあわせれば純光学型 OPALSに対する期待は大きい。ここで，'純光学型'とは被処理情報がすべて光を媒体として運ばれ，処理される形態を意味する。

純光学型OPALSの実現に際して最も大きな問題となるのは光アレイロジックで要求される符号化やOR演算等の非線形処理をいかにして実行するかという点である。2.2節でも述べたが，符号化法を始めとする非線形処理の実行には光機能素子の利用が避けられない。そこで，純光学型 OPALSに要求される光機能素子の機能および特性について 5.4.1 項で議論する。さらに，その結果をもとに現在考えうる１つの純光学型システムの構成例を 5.4.2 項で示す。

5.4.1　光機能素子

２次元平面に広がった光波で他の光波を２次元的に制御し得る空間光変調素子は従来の光学技術のみでは実現が困難であった各種の非線形処理を可能にする素子として，現在，活発に研究されている。[15,62,63] この素子を含め，純光学型OPALSを構成する場合には種々の光機能素子が必要となる。そこで，以下，符号化，実時間２次元相関演算，OR演算，一時記憶の順に純光学型OPALSで要求される光機能素子の機能，特性について議論する。

5.4.1.1　符号化素子

2.2節で述べたように，並列光論理演算法あるいは光アレイロジックにおける符号化には種々の方法が考えられる。しかし，いずれの方法をとるにしろ連続して並列信号の符号化を行うためには空間光変調素子を使用する必要がある。[63] 符号化の方式により偏光面制御，波長制御等，空間光変調素子に要求される機能は異なるが，共通した性能上の要求として速い応答速度および閾値特性があげられる。光アレイロジックでは１回の論理演算ごとに被処理データの符号化が必要であり，その効率的な実行が直接システムの処理能力を左右する。したがって，できる限り高速の素子が望まれる。また，光アレイロジックは反復処理を行うための誤差の蓄積が予想される。そこで，反復処理中に誤差補正の効果を持つ特性があればさらに都合がよい。この目的にはFig. 5.6に示す閾値特性を有する素子が有用である。この素子では0,1以外の入力状態は強制的に０または１の出力状態に変換され，その結果，符号化ごとに誤差補正が行われ誤差蓄積による誤りが防げる。その他，解像点数が高いこと，温度変化に対する安定性，高い再現性，空間的均一性，低消費エネルギー等の条件も要求される。[26] これらの条件については以下の素子につい

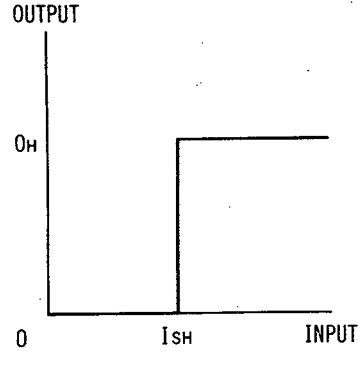


Fig.5.6　閾値特性

ても共通である。双安定特性はあってもよいが必ずしも必要としない。望まれる特性をTable 5.1に示す。ここでの数値は6.3.3項での議論に基づくものである。

Table 5.1　符号化素子の仕様

素子の大きさ	20㎜×20㎜
解像点数	＞1000×1000
応答時間	＜50μsec
状態分離度	＞1：1000

5.4.1.2　実時間2次元相関演算素子

実時間2次元相関演算法については2.3節でいくつかの方法を説明した。それらはいずれも光シャッタアレイや点光源アレイ等の使用を仮定したものである。したがってこれらの素子がなければ光アレイロジックは実現できない。実時間2次元相関演算は光アレイロジックでの積項演算に対応し，最も頻繁に実行される演算である。この演算の速度が光アレイロジックの基本サイクル時間を決定するため，OR演算素子と並んで最も速い応答速度が要求される。また，高い消光比を有する素子があればシステムの対雑音性を向上させることができる。望まれる特性はTable 5.2に示すものである。

Table 5.2　実時間2次元相関演算素子の仕様

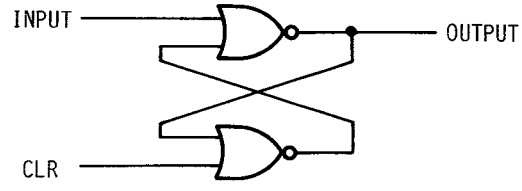
素子の大きさ	20㎜×20㎜
アレイ要素数	＞10×10
応答時間	＜50μsec
消光比	＞1：1000

5.4.1.3　OR演算素子

光アレイロジックでは積項演算の結果に対する論理和（OR演算）が要求される。この演算は，積項演算の結果がdark-true論理で出力されるため，たとえ同時に複数の積項演算が実行できるとしても信号の重ね合わせ等の簡単な方法では実現できない。4.2.3項で述べたように，本論文では遂次実行方式の積項演算を仮定しているため，このOR演算は時系列信号に対する演算となる。OR演算出力は次段の光アレイロジックとの接続性を考えるとbright-true論理で得られるものが望ましい。また，素子の初期化が簡単に行える必要もある。以上の条件を満たす素子としてディジタル回路におけるR-Sフリップフロップがあげられる。[31] Fig.5.7に

INPUT	CLR	OUTPUT
BRIGHT	BRIGHT	NOT CHANGE
BRIGHT	DARK	DARK
DARK	BRIGHT	BRIGHT
DARK	DARK	NOT STABLE

(a) 入出力信号の関係



(b) ランダムロジックによる表現

Fig.5.7 OR演算素子

はこの素子の入出力信号の関係とランダムロジックによる表記を示してある。入力に関してはFig.5.6に示す閾値特性も必要である。このOR演算素子の動作も実時間2次元相関演算用素子と同じく最高速の応答速度が要求される。Table 5.3は素子の特性の希望値をまとめたものである。

Table 5.3 OR演算素子の仕様

素子の大きさ	20㎜×20㎜
解像点数	＞1000×1000
応答時間	＜50μsec

5.4.1.4 一時記憶素子

光アレイロジックの積項演算を遂次実行する場合，被処理データを一連の積項演算が終了するまで一時記憶する必要がある。この目的には，光メモリ素子を利用する。[62] 低消費エネルギー，高速応答性，安定性，再現性等々，他の素子と共通の特性が要求される。OPALS内の一時記

憶素子は符号化素子が双安定特性を有する場合には不要であるが，OPALSの外部記憶としても有用な機能素子である。Table 5.4 は一時記憶素子に希望する特性をまとめたものである。

Table 5.4　一時記憶素子の仕様

素子の大きさ	20 mm × 20 mm
解像点数	＞1000 × 1000
応答時間	＜50 μsec
記憶時間	＞1 msec

5.4.2　システム構成

Fig.5.8 に考案した純光学型OPALSのシステム図を示す。各機能素子として 5.4.1 項で述べた素子を用いる。符号化法は 2.2.3 項で説明した偏光面符号法を利用する。この偏光面符号法で必要な光学素子にはFig.2.7 と共通の記号を表示してある。符号化用の光機能素子にＬＣＬＶを仮定しているが，これは現時点で製品として入手可能な素子であることによる。したがって，必ずしもこの素子である必要はない。

以下，このシステムの動作手順を示す。被処理データはLCLV **e** の書込み側へ入力する。この信号は光源から，偏光子，半波長板格子**a**，LCLV **b**，検光子**c**，半波長板格子**d**を経てきた光波により読出され，検光子**f**直後に符号化画像を形成する。この符号化画像に対し，可変多重結像レンズが２次元相関演算を実行し，演算結果をR−Sフリップフロップ素子に入力する。R−Sフリップフロップはその内部状態に応じた出力信号を送出し，その結果を一時記憶素子（Dフリップフロップ）およびシステム外部へ送る。一時記憶素子は積項演算の組が終了した後，書換えられ，その記憶信号をLCLV **b** の書込み側へ入力する。LCLV **b** に書込まれた信号は，光源

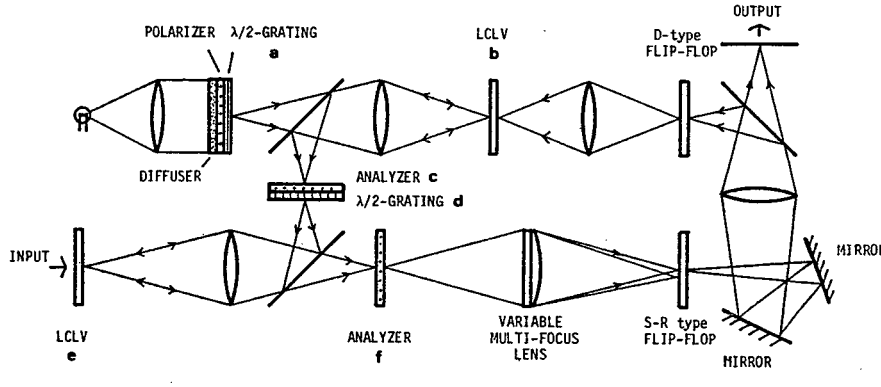


Fig.5.8　純光学型OPALS

から発せられた光により再び同じ経路を通り，反復処理に利用される。処理結果はR−Sフリップフロップの出力信号を外部でモニタすることにより得る。

純光学型OPALSでは，内部の信号伝送はすべて光による並列伝送である。先に述べたように，結像系の持つ情報伝送能力は非常に強力であり，電気的な伝送方式ではまったく及ばないものである。その結果，システム内の情報容量は各機能素子で決定されると考えてよい。たとえば，Fig.5.8のシステムの演算能力を計算すると約9.0×10^8 AND operation/secが得られる。これはLCLV上の処理点数，応答時間をそれぞれ1000×1000，20msecと仮定し，2つの画像の8近傍画素間の積項演算を行った場合の数値である。空間光変調素子の性能向上がシステムの演算能力に与える効果は明らかである。したがって，各機能素子の高分解能化をつき進め，たとえば，分子レベルでの現象を利用できるならば，光/電子複合型OPALSとはまったく比較にならない程の大容量データを処理し得る演算システムの構成も可能である。その意味で，純光学型OPALSは将来の新情報処理システムと成り得る発展性を兼ね備えたものであると言えよう。

5.5 結　言

本章では光の有する並列・高速情報処理能力を生かした汎用並列光演算システムOPALSの提案とその具体的なシステム化の方策を述べた。5.2節では光アレイロジックの利用による新しい並列光ディジタル演算システムOPALSの概念を説明し，その機能上・動作上の特徴を列挙し，有用性を示した。OPALSは概念的システムであり，それを具体的なシステムとして実現するには様々のレベルのシステム構成法が考えられる。そこで，著者は具体的なシステム化の例として2つのレベルのシステム構成を考案した。5.3節では光/電子複合型システムを，5.4節では純光学型システムを提案して，システム化の方法を詳述した。いずれのシステムも現時点では構想の域を出ないものであるが，近い将来，実現し得る汎用並列ディジタル演算システムとして有望視されるものと考えている。特に，光アレイロジックの利用により得られる並列プログラム可能性は現在の電子計算機が有する最大の特徴であり，OPALSが汎用演算システムとなるための十分条件でもある。他で提唱されているディジタル演算システム上では未だプログラムの方法についての具体的な方策が確立されていないことを考えあわせれば，OPALSは i) 具体的なシステム化の方法が提起されていること，ii) プログラムの駆動方法や演算実行方法が確立していること等の理由により，他のシステムよりも一歩先んじた新しい並列光演算システムとなっていることがわかる。この意味からも，OPALSは将来の新しい並列演算システムの有力な候補となり得るものと言えよう。

第6章　OPALSのシステム構成に対する考察

6.1　緒　　言

5章では光アレイロジックによる演算原理に基づき，任意の並列論理演算を効率よく実行する汎用並列光演算システムOPALSを提案した。OPALSは光アレイロジックの処理手順をそのままシステム構成に生かした汎用並列光演算システムである。その動作は光アレイロジックの動作と1対1に対応する。すなわち，光アレイロジックの処理手順である画像の符号化，積項演算，OR演算のそれぞれに対して1つのモジュールが対応し，各モジュールが符号化，2次元相関演算，OR演算を実行する。その結果，OPALSにおける処理・演算は制御しやすく，また各モジュールを別々に開発して，それらを組合せてシステムを構成することができるというシステム構築上の利点も生ずる。

しかし，処理効率の面から見た場合，OPALSのこのシステム構成は必ずしも最適なものとは言えない。なぜなら，各モジュールごとに光機能素子を必要とする結果，多くの種類の光機能素子がシステム内に存在し，それらの間のインターフェースを考えるとシステムはかえって複雑なものとなるからである。また，5章で述べたOPALSでは積項演算を逐次実行する方法を採用したためデータの流れの中に一時記憶要素が存在し，情報媒体である光波の高速伝播を妨げてしまう。これらの欠点を克服するには，積項演算を並列的に実行するシステム構成を考える必要がある。もしこのようなシステム構成をとることができればより高性能のシステムとなることは十分期待できる。

処理効率がよく，かつすぐれたシステム構成を持つOPALSを設計するためには様々なアプローチが考えられる。著者はOPALSの実現性・将来性の観点から2つのアプローチを用いて，新しい構成を持つシステムを考案した。第1のアプローチは現在の機能素子製作技術のレベルを認識した上で考案したものであり，必要最小限の光機能素子を用いてシステムを構成することを目標としたものである。第2のアプローチは電子処理技術の積極的な利用を仮定し，実用的なシステムへの発展性に重点を置いたものである。これらのアプローチの方向は必ずしも一致しないが，明確な概念が確立されていない光演算システムの研究の現状を考えあわせると，種々の方式の検討があってしかるべきであると思われる。そこで，本章では，これら2つのアプローチによるOPALSの最適システム化の方策を提案し，OPALSの光コンピュータとしての発展性について議論する。

6.2　光順序回路型 OPALS

OPALSを実現するためには4種類の光機能素子が必要である。すなわち，i) 符号化素子，ii) 実時間相関演算素子，iii) OR演算素子，iv) レジスタ用一時記憶素子である。それぞれの光機能

素子に要求される特性や素子動作に課せられる条件は5.4.1項で議論したが，現時点での光機能素子の開発状況を考えれば，できる限りシステム作成に使用する素子の種類を減らすことが望ましい。この観点に立ち，ここでは光アレイロジックの処理手順を再検討し，OR演算素子と一時記憶素子を排除したシステム構成を持つOPALSを提案する。

記憶素子を排除したOPALSを構築するための基本技術はフィードバック光学系を利用した光順序回路技術であり，ディジタル演算システムの原点に戻るものである。[24] 光順序回路を用いると一時記憶素子は不要となり，さらにOR演算素子も不要となる。しかし，OPALSに要求される機能をすべて実現するには3系統の光順序回路が必要となる。並列光論理演算法で用いる光学処理技術はspace-invariantなものであり，3つの独立した処理を同時に実行することは難しいが，波長多重技術を導入してこの問題を解決することができる。その結果，得られたシステムは光波の波長独立性を効果的に利用し，かつ光の情報処理能力を最大限に引出したものとなっている。

6.2.1 光順序回路技術

ディジタル演算方式を用いたすべての処理は論理関数の組合せとして実現する。論理関数を組合せたものは論理回路と呼ばれ，ディジタル演算の実行やディジタル計算機を設計することは究極的にこの論理回路を設計することに帰着する。論理回路は組合せ回路と順序回路に分類される。[31] 記憶要素を持ってない回路を組合せ回路と呼び，記憶要素を持つ回路を順序回路と呼ぶ。別の定義によると，組合せ回路とは入力$I(t)$だけで出力$O(t)$が決定される回路で，その機能は

$$O(t)=f(I(t)) \tag{6-1}$$

と表現できる。$I(t)$，$O(t)$はそれぞれ時刻tにおける入力変数列，出力変数列でfは出力関数である。一方，順序回路とは入力$I(t)$と内部状態$S(t)$から出力$O(t)$が決定されるもので，その機能は

$$O(t)=f(I(t),\ S(t)) \tag{6-2}$$

$$S(t+\Delta t)=g(I(t),\ S(t)) \tag{6-3}$$

と表現できる。ここで，$S(t)$は時刻tにおける内部状態変数列で，gは状態遷移関数と呼ばれる。

ディジタル演算方式で実行される複雑な処理はすべて順序回路を利用したものであり，ディジタル電子計算機の基本回路もこの順序回路である。順序回路の特徴は内部状態$S(t)$が存在し，それが保存されることである。したがって，順序回路は組合せ回路に記憶要素を付加して構成される。しかし，(6-3)式からわかるように，時刻$t+\Delta t$における内部状態$S(t+\Delta t)$は時刻tにおける内部状態$S(t)$の関数である。その結果，順序回路を構成するにはΔtの間だけ内部状

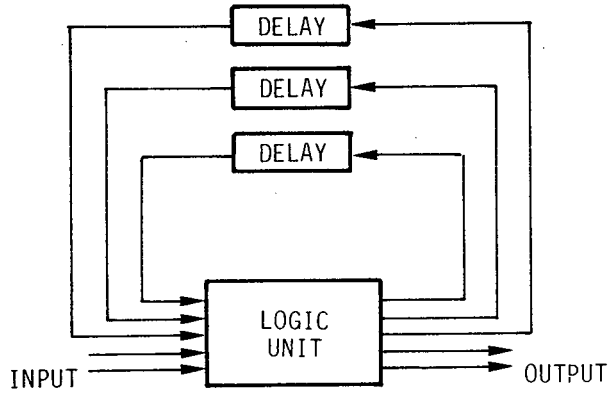


Fig.6.1 順序回路

態を保持できれば十分である。

簡単のため，Δt が一定である順序回路を考える。この回路は同期順序回路と呼ばれ，時刻はビットタイムと呼ばれる整数値で表現できる。[31] この場合，Δt 時間内の記憶は信号が伝送路を伝播する際に生じるΔt の時間遅れ（遅延）で実現可能である。したがって，特別な記憶素子を用いなくともFig.6.1に示すフィードバック系を利用することにより順序回路を構成することができる。

以上の考え方を光ディジタル演算処理システムの構成に応用すると並列論理演算要素と光インターコネクション技術のみで，システム内の順序回路が実現できる。Huang や Sawchuk らは並列論理演算要素として単一機能論理ゲートアレイを使用したシステムを提案しているが，[24,26] 並列光論理演算法を並列論理演算要素として用いる光ディジタル演算システムを構成すれば，さらに効率的なシステムが構築可能である。すなわち，光アレイロジックによる並列処理をこの光順序回路システム上で実現すれば，最小限の光機能素子により，5章で述べたOPALSと同等な性能を持つ光ディジタル演算システムを構成することができる。

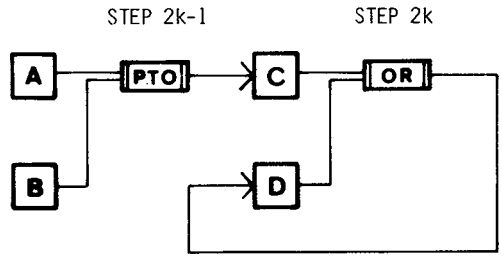
6.2.2 OPALSの処理手順の順序回路化

Fig.6.2は光アレイロジックの処理手順を模式的に表現したものである。演算の入力と出力を明確にするため論理変数列A, B, C, Dを考える。A, B, C, Dはそれぞれ，各画素が論理変数の画像データである。Fig.6.2中のPTOは積項演算（product term operation），ORはOR演算を表わし，ステップはビットタイムを示す。

この処理は次の順序論理式で表現できる。

$$C^{2k-1} = PTO_k[A^{2k-2},\ B^{2k-2}] \tag{6-4}$$

$$D^{2k} = C^{2k-1} + D^{2k-1}. \tag{6-5}$$



(a) 処理の流れ

STEP	REG. A	REG. B	REG. C	REG. D
1	a	b	PTO 1	
2	a	b	–	PTO 1
3	a	b	PTO 2	PTO 1
4	a	b	–	PTO 1+PTO 2
⋮	⋮	⋮	⋮	⋮
2n-1	a	b	PTO n	$\sum_{k=1}^{n-1}$ PTO k
2n	a	b	–	$\sum_{k=1}^{n}$ PTO k

(b) 各ステップにおけるレジスタの内容

Fig.6.2 光アレイロジックの処理手順

ここで A^k, B^k, C^k, D^k はそれぞれ第 k ビットタイムにおけるA, B, C, Dの値を示し，PTO_k は対象とする論理演算中の第 k 番目の積項演算を表わす。また，各データの内容を保存するために次の処理も必要である。

$$A^k = A^{k-1} \tag{6-6}$$

$$B^k = B^{k-1} \tag{6-7}$$

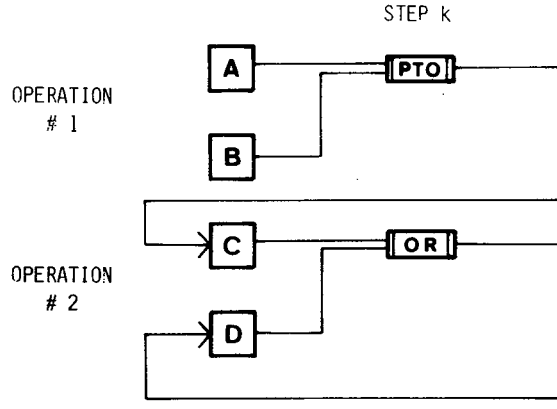
$$D^{2k+1} = D^{2k}. \tag{6-8}$$

5章で述べたOPALSは（6−4）式にしたがった処理のみを光学処理系で実行し，他の処理はすべて光機能素子を利用して行うシステム構成をとっていた。したがって，ここでの問題は，いかにして（6−4）～（6−8）式を光順序回路で実現するかということである。注意すべきことは，（6−4）～（6−8）式はともに2次元データに対する処理を表わしており，その実行には並列論

理回路が必要なことである。

まず，（6−6）式に注目する。Fig.5.2に示すように，OPALSのレジスタAは入力ポートの役割を持ち，レジスタAに置数されるデータは外部より供給される。したがって，そのデータは光順序回路システムの外部で保持されると仮定しても問題なく，（6−6）式の処理は光順序回路化の対象から除外できる。

次に，（6−4），（6−5）式を検討する。この2つの処理はFig.6.2に示した処理手順ではビットタイムごとに交互に行われるが，もし同時に実行すれば処理時間の短縮ができる（Fig.6.3）。たとえば，対象とする論理演算が n 個の積項で構成される場合，（6−4），（6−5）式を交互に



(a) 処理の流れ

STEP	REG. A	REG. B	REG. C	REG. D
1	a	b	PTO 1	
2	a	b	PTO 2	PTO 1
3	a	b	PTO 3	PTO 1+PTO 2
⋮	⋮	⋮	⋮	⋮
n	a	b	PTO n	$\sum_{k=1}^{n-1}$ PTO k
n+1	-	-	-	$\sum_{k=1}^{n}$ PTO k

(b) 各ステップにおけるレジスタの内容

Fig.6.3 パイプライン式光アレイロジックの処理手順

実行する方法ではその実行に $2n$ ビットタイム必要であったものが，同時実行方式では $n+1$ ビットタイムで完了する。したがって，もし $n>2$ ならば，処理時間を短縮することができる。この演算方式は電子計算機の処理速度向上法として用いられているパイプライン処理と等価である。[21]

以上を整理して（6−4）〜（6−8）式を書直すと次式を得る。

$$B^k = B^{k-1} \tag{6-9}$$

$$C^k = PTO_k[A^{k-1},\ B^{k-1}] \tag{6-10}$$

$$D^k = C^{k-1} + D^{k-1}. \tag{6-11}$$

これらはOPALSの動作を順序回路としての動作で表現したもので，Fig.6.1に示す光順序回路システムで実行できれば新しい形式のOPALSが構築できることになる。これらの処理の実行には並列光論理演算法を論理要素として用いる光順序回路システムを利用すればよく，その結果，記憶素子を用いずにOPALSを構成することができる。

ここで，(6−10) 式の処理は複数の画素を対象とするAND演算（近傍画素間積項演算）であることに注意を要する。Huangらが提案した単一機能論理ゲートアレイを論理要素とした光順序回路[24,26]を用いて同様の演算を実行するには非常に多くの論理ゲートを必要とし，処理対象画像を構成する処理点数（画素数）が限定されてしまう。この点から，ここで提案した並列光論理演算法を用いた光順序回路システムはより効率的な処理を実現していると言うことができる。

6.2.3 波長多重並列光論理演算法

（6−9）〜（6−11）式で示される3つの処理を同時に実行するためには3つの独立した光順序回路を使用しなければならない。そのために，3つの光学処理系を用いてシステムを構成する方法が考えられるが，各処理系間の情報交換の問題や光学系の複雑化の点で望ましい方法ではない。そこで，光波の波長多重技術と処理データの分散配置を利用した新しい波長多重並列光論理演算法を考案した。

Fig.6.4に波長多重並列光論理演算法の概念図を示す。図は (i, j) 画素に対する処理を抜出して示したものである。この図にはP, Q, R, S, T, U, V, Wの8画像データを入力とする4種類の論理演算法を同時に実行し，各論理演算の結果をX, Y, Z, Ωの4画像として出力する場合を示してある。各論理演算は相異なる4波長の光を用いた並列光論理演算法で実行される。それぞれの出力結果は対応波長のみを透過する波長フィルタで分離する。異なった波長の光は干渉性がないため，独立した4種類の論理演算が同時に実行できる。Fig.6.4からわかるように，この処理系は3.3.1項で述べたビット分散配置法を4つ組合せたものと等価である。光学処理系は全体として2入力，1出力のシステムで各データはFig.6.5に示すように4つの画像データを分散配置したものである。

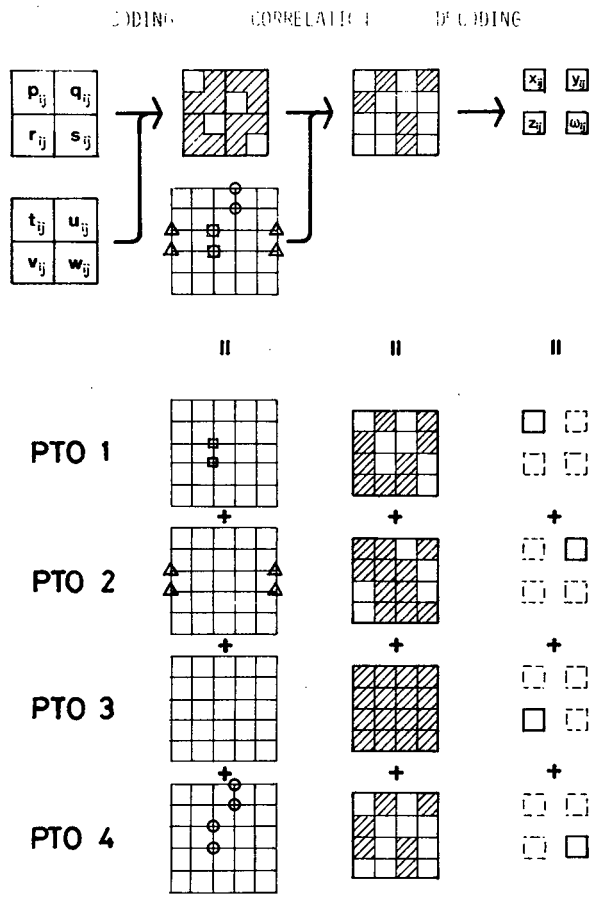


Fig.6.4　波長多重並列光論理演算法

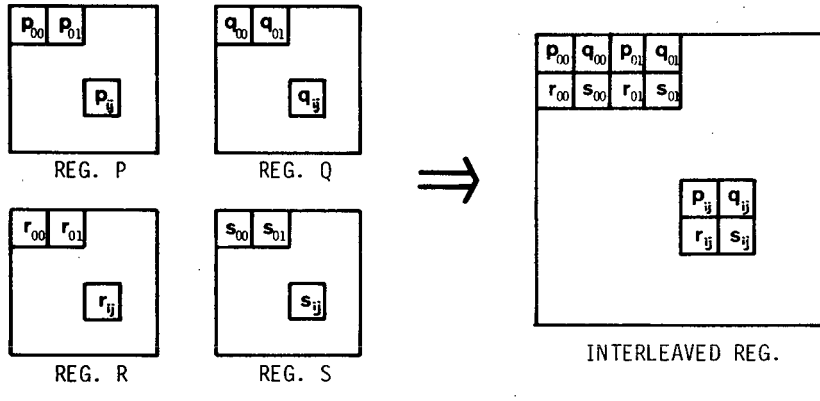


Fig.6.5　多重演算用分散配置画像

波長多重並列光論理演算法で実行する処理は次の論理式で記述できる。

$$X^k = PTO_1[P^k, Q^k, R^k, S^k, T^k, U^k, V^k, W^k] \quad (6-12)$$

$$Y^k = PTO_2[P^k, Q^k, R^k, S^k, T^k, U^k, V^k, W^k] \quad (6-13)$$

$$Z^k = PTO_3[P^k, Q^k, R^k, S^k, T^k, U^k, V^k, W^k] \quad (6-14)$$

$$\Omega^k = PTO_4[P^k, Q^k, R^k, S^k, T^k, U^k, V^k, W^k]. \quad (6-15)$$

ここでは dark-true 論理により積項演算を実行すると考えている。並列光論理演算法は組合せ論理を実現する手段であり，順序回路の構成にはFig.6.1 に示すフィードバック系による遅延が必要である。

Fig.6.6 は波長多重並列光論理演算を実行する光学系の例である。4 波長の発光スペクトルを独立に制御できる点光源と 4 種類の波長フィルタを組込んだ復号マスクが特徴点である。点光源の制御により，独立した 4 つの論理演算結果を同時に得ることができる。

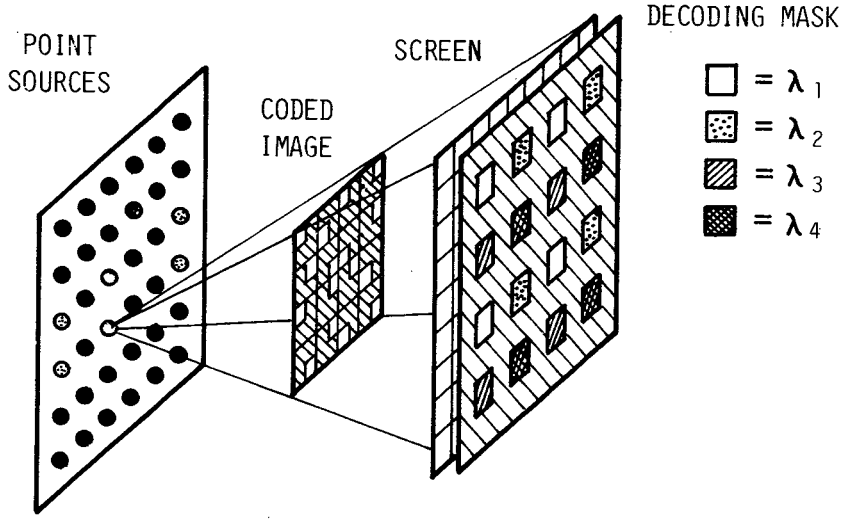


Fig.6.6 投影光学系を用いた波長多重並列光論理演算法

6.2.4 システムの構成

波長多重並列光論理演算法を論理要素とした光順序回路システムを考える。波長多重並列光論理演算法では，光学処理系への入出力の単位は，4 つの画像データを分散配置した 2 次元データであり，全体としては 2 入力，1 出力の処理系となっている。そこで，フィードバック系構成のため，出力データを 2 番目の入力データとして戻す方式をとる。Fig.5.2 のOPALSにならって，1 番目の入力データはシステム外部から与える。このフィードバック系では画像データ X, Y, Z,

Ωがそれぞれ画像データT, U, V, Wとして戻される(Fig.6.4参照)。フィードバック系での遅延時間(=光波伝播時間+符号化素子の応答時間)をビットタイムの長さにとれば，次式が成立する。

$$T^k = X^{k-1} \tag{6-16}$$

$$U^k = Y^{k-1} \tag{6-17}$$

$$V^k = Z^{k-1} \tag{6-18}$$

$$W^k = \Omega^{k-1}. \tag{6-19}$$

これらの結果と(6-12)～(6-15)式の処理を組合せると次式を得る。

$$T^k = PTO_1[P^{k-1}, Q^{k-1}, R^{k-1}, S^{k-1}, T^{k-1}, U^{k-1}, V^{k-1}, W^{k-1}] \tag{6-20}$$

$$U^k = PTO_2[P^{k-1}, Q^{k-1}, R^{k-1}, S^{k-1}, T^{k-1}, U^{k-1}, V^{k-1}, W^{k-1}] \tag{6-21}$$

$$V^k = PTO_3[P^{k-1}, Q^{k-1}, R^{k-1}, S^{k-1}, T^{k-1}, U^{k-1}, V^{k-1}, W^{k-1}] \tag{6-22}$$

$$W^k = PTO_4[P^{k-1}, Q^{k-1}, R^{k-1}, S^{k-1}, T^{k-1}, U^{k-1}, V^{k-1}, W^{k-1}]. \tag{6-23}$$

(6-20)～(6-23)式が示すように，以上の操作は4つの順序回路を同時に実現していることに他ならず，OPALSの構成に必要な処理を実行していることになる。したがって，この波長多重光順序回路により(6-9)～(6-11)式の処理を実行すれば新しいシステム構成を持つOPALSを構築することができる。

具体的には，波長多重光順序回路での4系統の処理系のうち3つを使い，次式にしたがって，OPALS内の処理データを表わす4つの論理変数列A, B, C, Dを(6-20)～(6-23)式の変数列に割当てる。

$$A = P \tag{6-24}$$

$$B = T \tag{6-25}$$

$$C = U \tag{6-26}$$

$$D = W. \tag{6-27}$$

また，各処理系での演算は次のように定義する。

$$PTO_1 \equiv T \tag{6-28}$$

$$PTO_2 \equiv PTO[P, T] \tag{6-29}$$

$$PTO_4 \equiv \overline{U}W. \tag{6-30}$$

したがって，(6-20)～(6-30)式より

$$B^k = B^{k-1} \tag{6-31}$$

$$C^k = PTO[A^{k-1}, B^{k-1}] \qquad (6-32)$$

$$D^k = \overline{C}^{k-1} D^{k-1} \qquad (6-33)$$

を得る。これらは（6－9）～（6－11）式と等価であり，OPALSを実現する光順序回路の動作を表わす。ここで，（6－33）式は（6－11）式と形が異なるが，De Morganの定理[31]を用いれば

$$\overline{D}^k = C^{k-1} + \overline{D}^{k-1} \qquad (6-34)$$

となり，Dの内容が負論理で表わされていることを除き，（6－11）式と一致する。これは，（6－11）式の論理和を積項演算で行うために用いた技術である。

Fig.6.7は1つの画素について構成される光順序回路の信号の流れを抽出したものである。波長多重並列光論理演算法を用いた多重光順序回路では，図と同じ回路が処理画像内の全画素に対して並列に構成され，その結果，並列光論理演算を実現する。

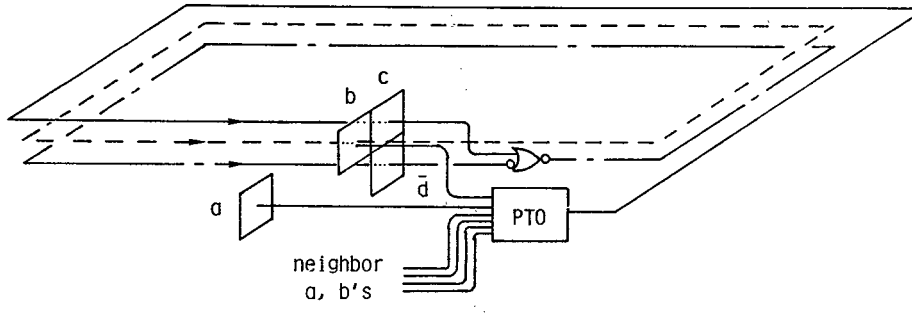


Fig.6.7 光順序回路型OPALS内の信号の流れ

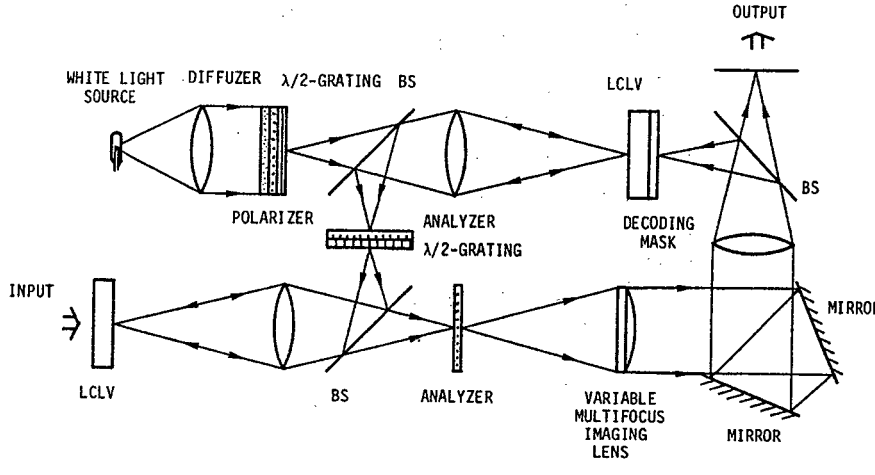
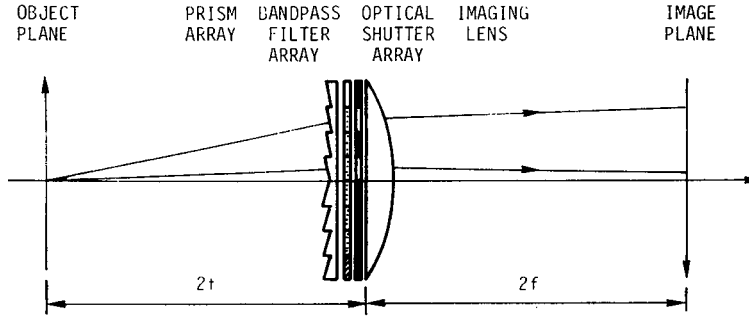


Fig.6.8 光順序回路型OPALS

以上の手順に基づいた処理を行う光学系をFig.6.8に示す。これは，Fig.5.8のシステムからOR演算素子と一時記憶素子を取除いたもので，符号化法には偏光面符号法を利用している。波長多重論理演算はFig.6.9に示す波長選択フィルタアレイ組込型可変多重結像レンズで実行する。この素子は，2.3.2項で述べた多重結像系を変形したもので，Fig.2.11 (b) の多重結像レンズと波長選択フィルタを組合せたものである。波長ごとに異なった2次元相関像を出力することができる。

Fig.6.9(b)はプリズムアレイおよび波長選択フィルタアレイの設計例で，8近傍画素間論理演算用のものである。#1，#2，#4は波長チャンネルを示し，(p, q)はプリズムによるシフト量を表わす。この(p, q)は(1−14)式中に現われる並列光論理演算法のカーネル成分 w_{pq} に対応する。



(a) 構成

#1 (0,0)	#1 (0,1)	#1 (1,0)	#1 (1,1)	#1 $(2,\bar{1})$	#1 $(2,\bar{2})$	#1 $(3,\bar{1})$	#1 $(3,\bar{2})$
#2 $(\bar{6},\bar{4})$	#2 $(\bar{6},\bar{3})$	#2 $(\bar{5},\bar{4})$	#2 $(\bar{5},\bar{3})$	#2 $(\bar{6},0)$	#2 $(\bar{6},1)$	#2 $(\bar{5},0)$	#2 $(\bar{5},1)$
#2 $(\bar{6},4)$	#2 $(\bar{6},5)$	#2 $(\bar{5},4)$	#2 $(\bar{5},5)$	#2 $(2,\bar{4})$	#2 $(2,\bar{3})$	#2 $(1,\bar{4})$	#2 $(1,\bar{3})$
#2 $(\bar{2},0)$	#2 $(\bar{2},1)$	#2 $(\bar{1},0)$	#2 $(\bar{1},1)$	#2 $(\bar{2},4)$	#2 $(\bar{2},5)$	#2 $(\bar{1},4)$	#2 $(\bar{1},5)$
#2 $(2,\bar{4})$	#2 $(2,\bar{3})$	#2 $(3,\bar{4})$	#2 $(3,\bar{3})$	#2 (2,0)	#2 (2,1)	#2 (3,0)	#2 (3,1)
#2 (2,4)	#2 (2,5)	#2 (3,4)	#2 (3,5)	#2 (0,0)	#2 (0,1)	#2 (1,0)	#2 (1,1)
#4 (0,0)	#4 (0,1)	#4 (1,2)	#4 (1,3)				

(b) 8近傍画素間論理演算用偏向素子アレイ

Fig.6.9 波長選択性多重結像レンズ

6.2.5 計算機シミュレーション

波長多重光順序回路によるOPALSの動作を確認するためにシミュレーション実験を行った。実行した処理は入力像のエッジ検出で，4.3.1項の(4－15)式で表わされる処理である。処理の実行手順は次の通りである。

1. 処理対象データを入力ポート（レジスタA）に置く。
2. レジスタAの内容を内部レジスタ（レジスタB）に送る。
3. レジスタBの内容に対し積項演算を行う。ただし，OR演算はパイプライン的に実行する。
4. 必要な回数だけ3.を繰返す。
5. 演算結果を出力する。

ここで，手順3が(6－31)～(6－33)式の光順序回路に対応する処理である。他の処理も同様な光順序回路で実行できる。

Fig.6.10にシミュレーション結果を示す。この処理は以下に示すプログラムを実行したものである。

STEP 1:　(object) → A

STEP 2:　A → B ;　0 → C ;　1 → D

STEP 3:　B → B ;　$b_0\bar{b}_1$ → C ;　$\bar{C}D$ → D

STEP 4:　B → B ;　$b_0\bar{b}_3$ → C ;　$\bar{C}D$ → D

STEP 5:　B → B ;　$b_0\bar{b}_5$ → C ;　$\bar{C}D$ → D

STEP 6:　B → B ;　$b_0\bar{b}_7$ → C ;　$\bar{C}D$ → D

STEP 7:　0 → B ;　0 → C ;　$\bar{C}D$ → D

STEP 8:　$\bar{D}$ → B ;　0 → C ;　1 → D

ここで，矢印は処理結果の送り先を示し，$b_0\bar{b}_1$ 等は近傍画素間積項演算の内容を示す。図は光学処理系への2つの入力データ（A，B，C，Dを分散配置したもの），出力データ（B，C，Dが分散配置されたもの），演算カーネル（全波長チャンネルに対するもの），さらに出力データを構成する3つの画像B，C，Dをステップ毎に表示したものである。演算カーネルは各波長チャンネルの信号を4 bit数の各桁に対応させて16進数で表現してある。たとえば，$3=(0011)_2$ はチャンネル1およびチャンネル2に値1が存在することを示す。

一連の処理結果から被処理データが画像Bとして保存されていること，出力結果が画像Dとして反転した状態で逐次得られていくことがわかる。この結果から，光順序回路を用いて構成したOPALSの動作の妥当性が確認できた。しかし，各素子の過渡特性に対する検討は行っておらず，今後の研究が必要である。

Fig.6.8のシステムは符号化および実時間相関演算用に光機能素子を必要とするものの，シス

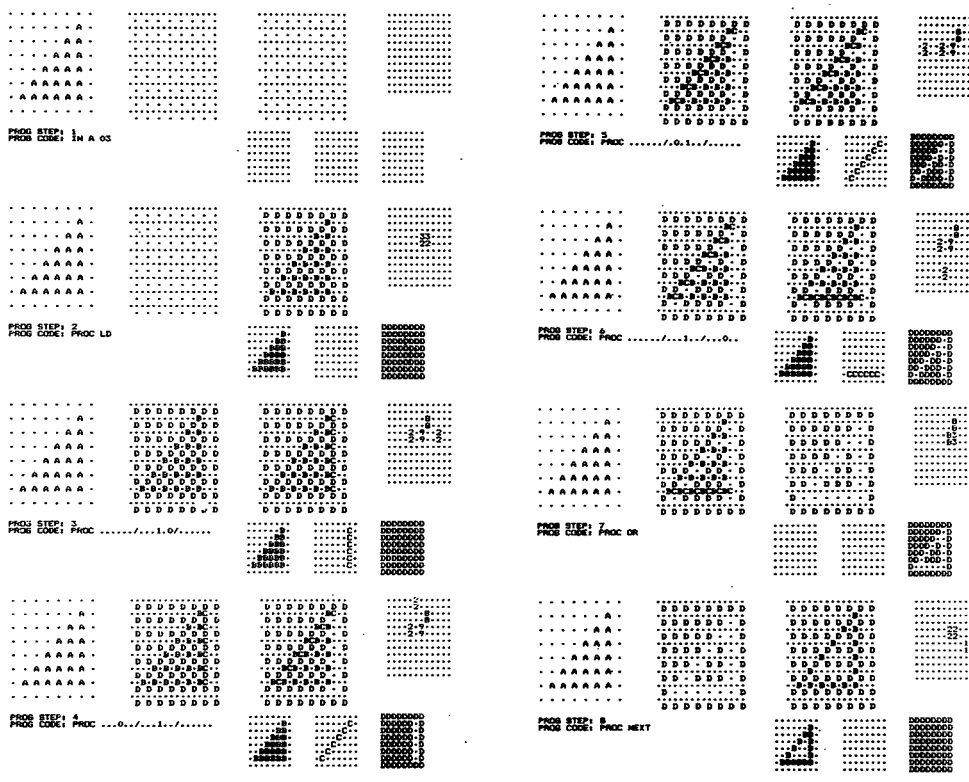


Fig.6.10　光順序回路型OPALSの動作シミュレーションの結果

テム全体としては単純な構成となっている。現在の光機能素子製作技術のレベルから考えると使用する機能素子が少ないことはシステム構成上非常に望ましいことである。光機能素子が少ない分だけシステムの制御は複雑になるが，システム機能の高度化に対応するためにはある程度はやむを得ない。しかし，波長多重光順序回路が持つ高密度・並列演算能力を考えたとき，同等の演算機能を他の演算システムで実現する際に生じる様々の問題点とは比較にならぬ程解決が容易なものである。この観点から光順序回路型OPALSは汎用並列光ディジタル演算システムの一つの雛型として有望なものであると考えられる。

6.3　OPALSの光コンピュータへの発展性

OPALS開発の最終目的は現在のディジタル電子計算機を凌駕し得る新しい大規模情報処理システム，すなわち光コンピュータを実現することである。しかし，一言で光コンピュータと言っても，ディジタル電子計算機が現在の地位を獲得するまでの過程を考えると，その実現がそう簡単でないことは明白である。システムの汎用性，信頼性，柔軟性，操作性等の点でディジタル電子計算機に比肩し得るシステムを作るためには膨大な技術の蓄積が要求され，やっと動き始めた光コンピュー

タ開発を目ざした基礎研究の成果だけでは到底実現が不可能のように思われる。

それにもかかわらず光コンピュータに対する要求，期待は近年増々強くなっている。これは現在のディジタル電子計算機の高性能化に限界が見え始めたことと共に，光通信をはじめとする各種の光技術[59]の発達に伴う技術面での向上によってある程度の光演算システムが開発できるのではないかという期待感が生じてきたことによるものと考えられる。光通信技術は光波伝送の利点を電子処理技術の中に取込んだ結果，成功した技術であるが，この成功を足場とすれば電子計算機よりもより高度な情報処理を実現し得るシステムが構築できるのではないかという予測が成立つ。この光コンピュータに対する考え方は電子技術が主体であり，光学技術は補助的な役割を受持つというものである。先に光学技術を主体とした光コンピュータの実現は非常に困難であると述べたが，電子技術の基盤の上でそれが開発される場合には事態が変わってくる。すなわち，光通信に付随する光技術や電子技術において蓄積されてきた知識や資産を利用すると光コンピュータの開発は現実に近いものとなってくる。ただし，このアプローチが電子回路の導線を単に光導波路で置換するということを意味するものではないことは銘記しておかなければならない。

著者は本論文で電子技術の利用を認めつつもできる限り光学処理技術を応用した演算システムが望ましいとする立場をとってきた。これは，光学処理技術の提供する並列処理能力が電子技術の提供する演算能力をはるかに上回るとの見地からである。6.2節の光順序回路型OPALSはこの思想をつき進めた1つの形態のシステムであるが，先に述べたように光通信をはじめとする光/電子総合技術による成果は無視できず，第1世代の光コンピュータはまず電子技術主体のシステムとして出現するものと考えられる。そこで，本節では，電子技術を主体とした1つの型のOPALSを考え，光コンピュータへの発展性を検討する。

6.3.1 問題点の認識と対応

OPALSを光コンピュータシステムに発展させるためには多くの問題を克服しなければならない。現時点で考えられる問題の中で特筆すべきものを列挙すると次の通りである。

1. 光機能素子の開発
2. システムの安定性・信頼性の確保
3. 制御方法の確立
4. 有効な利用分野の探索

以下，それぞれについて補足説明をし，問題解決に要求される技術と条件を考察する。

第1の問題は，OPALS自体の実現に関するものである。光/電子複合型OPALSにしろ，純光学型OPALSにしろ，光アレイロジックの手順にしたがって処理を行うためには光機能素子が必要である。光機能素子に対する要求は5.4.1項でも列挙した。しかし，現在，利用可能な素子がほとんど無いことを考えあわせれば，新たな光機能素子の開発がOPALSを実現する上で最も

必要なことであり，OPALSを光コンピュータとして機能させるための第1条件である。後述するように，技術面から考えると機能素子の開発には3次元LSI技術の応用が期待できる。[60,61]

第2の問題は実用的な見地から出されるものである。定盤の上に光学素子を並べて作ったシステムは研究室レベルでは利用できても，一般の使用には耐えない。精密な調整を要求される光学系では十分な安定性・信頼性を得ることは難しいからである。この問題を解決するためには微小光学技術[44,46,66]の応用が有望である。各光学素子を1つの基盤上に固定できればシステムの安定性・信頼性を確保できる。ただし，現在の微小光学技術は平面上の2次元光回路の作製を目的としたものであるが，光の情報処理・伝送能力を活用するためには立体形状の3次元光回路に対する技術開発が不可欠である。[64]

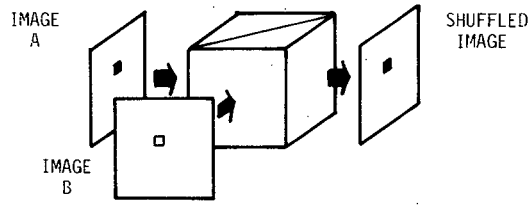
第3の問題はシステムの高性能化に伴なって付随してくるものと予想される。OPALSの主要な動作は2次元相関演算に用いる演算カーネルによって決定されるが，システムで複雑な処理を行うには演算カーネルの要素数が大きくなり，その制御方法が問題となる。演算カーネルの制御には並列性・高速性が要求され，電子処理系は不適である。そこで，光波の2次元信号を用いた制御方式が考えられる。たとえば，OPALSで処理した結果の一部分を演算カーネルの制御用として用いることができれば処理の高速性が達成できると共に，新しい演算方式の可能性が開けてくることにもなる。具体的にどのような演算が実現できるかは今後の課題である。

第4の問題は光コンピュータの応用分野に関するものである。OPALSはSIMD形式の並列処理を実行するが，実際の問題に適用させるためには種々の並列処理アルゴリズムの研究が不可欠である。幸いこの問題は現在の並列電子計算機が抱えている問題と同じであり，既に活発な研究が始まっている。したがって，その成果を利用することが期待できる。しかし，先に述べたシステム出力を用いてシステム自体の制御を行う演算形態のシステムはOPALS独自のものであり，新たな研究が必要である。

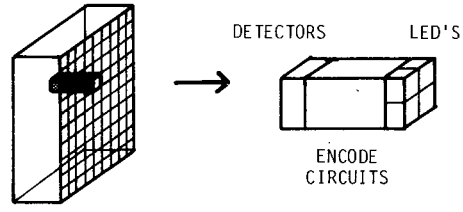
6.3.2 機能モジュール型OPALS

ここでは，前項での議論をもとに，現在および近未来の電子技術を応用する1つの型のOPALSを提案する。これはOPALSの構成要素を機能モジュールとして作製し，それらの組合せによりシステムを実現するものである。

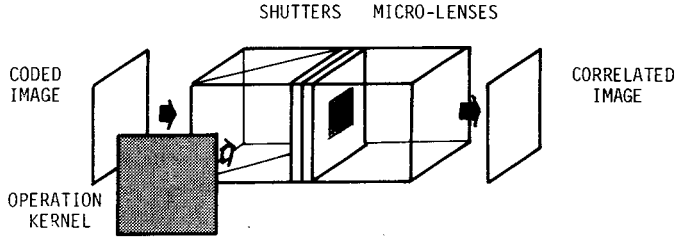
機能モジュールはFig.6.11に示す3種類を用意する。すなわち，i）並列結合・分配器，ii）並列符号器，iii）並列相関器である。並列結合・分配器は2つの画像を画素ごとにシャッフルして1つの画像へ変換する機能を持ち，並列符号器に入力データを転送する働きを持つ。また，信号を逆方向に通すと画像の分配も行える。並列符号器は並列結合・分配器でシャッフルした2画像を符号化し，符号化画像を出力する。並列相関器は符号化画像に対して2次元相関演算を実行するものである。相関演算に用いる演算カーネルは外部から画像信号で与える。



(a) 並列結合・分配器



(b) 並列符号器

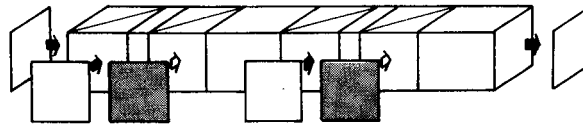


(c) 並列相関器

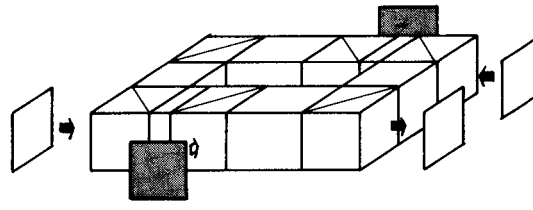
Fig.6.11 OPALS用機能モジュール

ここで，並列相関器モジュールについて説明する。これは，多重結像光学系をモジュール内に閉込めた構造を持ち，光波の自由空間伝播を利用して相関演算を実行する機能を持つモジュールである。この種の微小光学素子はまだ開発されていないが，光の並列処理能力を利用する上での最重要素子である。光シャッタアレイは外部からの画像信号で動作し，マイクロレンズアレイへの入射光の選択を行う。

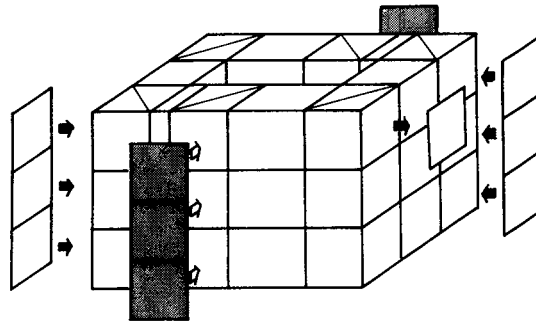
機能モジュール型OPALSの最大の特徴は，機能モジュールの組合せにより種々の形態の光演算システムが簡単に構築できる点にある。Fig.6.12 にいくつかのシステム構成例を示す。(a) は機能モジュールを直列接続したもので反復処理を必要としない場合に有効である。(b)は反復処理



(a) 直列システム



(b) 反復システム



(c) 多重反復システム

Fig.6.12 機能モジュール型OPALS

が可能なシステムでOPALSの処理系を2つ組合せた形になっている。ただし，これらの系には記憶要素がないため積項演算を逐次実行する形式の光アレイロジックは実行できない。そこで，光アレイロジックを実行するためには，一時記憶用の機能モジュールを新たに用意するか，あるいは6.2節で述べた光順序回路技術を用いるかのいずれかの方法が考えられる。機能モジュール形式の利点を生かすことを考えれば，Fig.6.12(c)のシステム構成も可能である。これは光アレイロジックにおける積項演算が並列的に実行でき，しかも記憶素子を必要としないで処理速度の飛躍的な向上が期待できるシステムである。

このように，機能モジュール型OPALSは光学情報処理技術を特長を生かしつつ，電子技術で利用できるものはできうる限り利用しようという立場に立って考案したものである。このシステムはすぐれた発展性を有し，光コンピュータとしても十分，性能を発揮し得るものと考えられる。各機能モジュールの作製には3次元LSI技術の応用が期待できる。[60,61] 3次元LSIは，従来平面上に形成されていたLSIを積ね合わせた多層構造を持つ半導体素子である。高機能化，高集積

化が可能であり，次世代電子素子の主流と目されている高集積素子である。よく知られているように，3次元LSI自体がすでに1つの演算システムであり，明確な思想に基づいた素子の設計・開発が必要である。その意味で，OPALS用の機能モジュールの設計は，まさに，3次元LSIに恰好の応用の場を与えるものであり，その開発は電子技術と光学処理技術の接点として大きな意義を持ってくるものと思われる。

ただし，機能モジュール型OPALSは本質的に光/電子複合型システムであり，光/電子，電子/光変換によるエネルギー損失，時間遅延を避けられず，性能には上限が存在するが，実現可能な第1世代の光コンピュータとして有望なシステムであろう。

6.3.3 OPALSの能力限界

本節ではOPALSの性能の上限を与える諸要因を考察する。考えられる要因を列挙すると次のようになる。

1. 光機能素子・機能モジュールのSBWP(space-bandwidth product)[38]
2. 光学処理系のSBWP
3. 光機能素子・機能モジュールの演算速度
4. 光波伝播による時間遅れ
5. 消費エネルギー・熱
6. SIMDアーキテクチャ．

1.および2.は空間並列処理可能な画素点数の上限を与え，3.および4.は単位時間内における並列処理の限界を与える。また，5.はシステム実装上の制約となる。6.は応用できる演算の種類を制限する。以下，各項について検討し，現時点で到達可能な数値を示す。

まず，第1の要因について，光機能素子，機能モジュールのいずれも処理し得る点数は空間分解能によって規制される。たとえば，光機能素子では光の変調に用いる光学的媒質の厚みが空間分解能を制約し，機能モジュールの場合には機能要素を構成する半導体の集積度が空間分解能を決定する。また，素子自体の製造法の問題から素子をむやみに大きくすることはできず，結局，各素子，モジュールにおける処理点数は制限をうける。本論文のシステムで使用を仮定したLCLVの場合，現在，素子の大きさ20㎜×20㎜で解像点数は約1000×1000である。[63] もちろん，半導体集積素子でも同程度の数値は実現できる。

第2の要因は，特に，使用する相関演算光学系による制限である。OPALSでは実時間2次元相関演算を実現するために，光路を分割し各光路を通過する光を制御する方法を用いている。その結果，各光路で扱い得る情報量は非常に限定されてしまう。この情報量は演算カーネルの要素数と関係し，要素数が多くなればアレイを構成する単位光学系の情報容量は減る。マイクロレンズアレイによる多重結像系の場合，演算カーネルの要素数を100×100とすると20㎜×20㎜内で

5000×5000 点の分解が可能である。これは1.の制限より緩やかなものである。ただし，この場合，マイクロレンズの開口数の制限より，多重結像系の入力面と出力面は100㎜以上離す必要がある。

第3の要因は光機能素子，機能モジュールの物理的特性に依存する。演算速度は5.の消費エネルギーと関連しており，高速に演算を実行しようとすると，それだけ多くのエネルギーを消費する。LCLVの場合，応答時間は20msec程度と非常に遅いが，強誘電性液晶を利用すると50μsecは達成できる見込みがある。[65] 半導体素子では光/電子，電子/光の変換時間が演算速度に大きく影響する。しかし，半導体レーザ，アバランシェフォトダイオードの利用で1nsecも不可能ではない。[59]

第4の要因は光速が有限であることに起因する。たとえば，先に述べたマイクロレンズアレイによる多重結像系では入力面から出力面への距離は100㎜であり，光波が伝播するのに333psec要する。これは，光機能素子の応答速度に比較するとはるかに速いものであるが，光コンピュータシステムの究極的な能力限界を示す数値である。

第5の要因は演算速度，素子密度と関連するもので，光機能素子，機能モジュールの動作原理に大きく依存する。しかし，一般的に言って，光/電子，電子/光変換を要する機能モジュール型OPALSの方が消費エネルギーの点では不利である。

最後の要因はアーキテクチャと処理アルゴリズムの関係に依存する。これはすべての種類の計算機が背負う宿命的な問題であり，SIMD形式のアーキテクチャであるが故に生ずるものではない。すべての計算機はそのアーキテクチャに適した処理アルゴリズムを用いて初めて本来の処理能力を発揮する。

以上を総合し，OPALSの性能評価を行った結果をTable 6.1に示す。これはLCLVを用いた光順序回路型OPALS，機能モジュール型OPALS，さらに理想的光演算システムの3つにつ

Table 6.1 OPALSの性能評価

	LCLVによる光順序回路型OPALS	機能モジュール型OPALS	理想的光演算システム**
処理点数	1000×1000	1000×1000	5000×5000
サイクル時間	20msec	1nsec	333 psec
演算能力*	9.0×10^{8}	1.8×10^{16}	1.8×10^{19}
処理データの大きさ	20㎜×20㎜	20㎜×20㎜	20㎜×20㎜
システムの大きさ	3000㎜×3000㎜	200㎜×200㎜	200㎜×200㎜

* 演算能力の単位は［AND operation/sec］
** noiseless

いて，処理の様式，演算能力*，システムの規模を試算したものである。理想的光演算システムとは，光波の物理的性質の限界を利用したシステムで，光演算システムの１つの上限値とみなすことができる。直接的な比較は難しいが，スーパーコンピュータの開発目標値，1.0×10^{10} FLOPS［floating operation / sec］(1992年時点)[17] をTable 6.1中の演算能力単位に換算すれば，** 2.0×10^{12} AND operation / sec を得る。この値と表中の演算能力値とを比較すれば，OPALSが新しい情報処理システムとして十分期待できるものであることが結論できる。

6.4 結　言

OPALSは並列論理演算を効率よく実行する汎用並列光演算システムであるが，そのシステム構成には検討すべき点が多い。そこで，本章では，２つのアプローチから処理効率のよいシステム構成を考案した。6.2節では，必要最小限の光機能素子でシステムが構成できる，光順序回路によるOPALSを提案した。これは波長多重技術の使用を含め，光の情報処理能力を最大限に活用したシステムであり，光学処理技術を単独に利用してもシステムが構築し得る可能性を示した。6.3節では，別のアプローチとして電子技術を積極的に利用すると仮定して，機能モジュール構成のOPALSを提案した。これは将来の光コンピュータとしての発展性を考慮して開発したシステムで，第１世代の光コンピュータの雛型と成り得るものと考えられる。最後に，OPALSの性能の上限を与える要因を考察し，OPALSの達成し得る性能を計算した。その結果，OPALSが新しい大容量情報処理システムとして十分期待が持てるものであるとの結論を得た。

---

* OPALSの基本演算は積項演算であり，これは複数のAND演算により構成される。そこで演算能力の評価単位には１秒間当たりのAND演算の実行回数を用いた。

** 浮動小数点演算を実行するのにAND演算200回を要すると仮定した。

総　　括

本論文では，光の有する並列・高速演算処理能力を有効に生かした並列光論理演算法をもとに，２次元画像に対する任意の並列近傍画素間演算を実行し得る汎用並列光演算システムOPALSの開発研究の成果について述べた。本研究により得られた知見および成果を以下，各章毎に総括し，今後の研究課題について考察する。

第１章では，光の並列性を生かした並列光論理演算法に関する研究を行った。まず，投影光学系を用いた並列光論理演算法の実行方法について述べ，確証実験によりその原理を確認した。そして，回折理論から論理ゲートの集積密度の限界を評価した。また，並列光論理演算法の原理を一般化し，その結果から本演算法の並列光演算に対する拡張性，応用性などを示した。並列光論理演算法の持つ特徴は次の通りである。

1. 使用する処理系は非常に簡単であり，性能に対する価格比が低い。
2. ２つの２値画像に対する全論理関数を並列的に実行可能。
3. プログラム可能であり，任意の組合せ論理を並列に，しかも簡単に実行できる。
4. 論理関数の選択は相関演算に用いるカーネルの組合せで簡単に実現可能。
5. 各ゲートはSIMD形式で並列動作する。
6. 入力２値画像の空間的な符号化が必要であるが，復号は簡単な復号マスクで行える。

第２章では，並列光論理演算法の実行に必要な符号化法と実時間２次元相関演算法を考案し，それらを実行する新しい光学処理系を考案した。符号化法としては次の４つの技術を利用した方法を考えた。

1. 波長選択性フィルタによるEQU演算
2. 計算機ホログラムを用いたフィルタリング処理
3. 偏光板と半波長板による偏光面制御
4. 単軸結晶における複屈折現象

いずれも光学処理の特徴である光の並列性を有効に生かしたものである。また，実時間２次元相関演算法は次の３つの方法を考案し，その概要を述べた。

1. 点光源アレイを用いた多重投影法
2. 複数の結像系による多重結像法
3. コヒーレント処理技術を応用した回折パターンサンプル法

これらの方法を用いると相関演算の内容・種類を簡単に制御できるという特徴を持つ。各符号化法，実時間２次元相関演算法ともに空間光変調素子等の光機能素子を使用すると実時間処理が行える。

その結果，実時間動作可能な並列光論理ゲートの構成方法に対する知見を得た。

第３章では，並列光論理演算法をもとにした並列画像処理法を考案した。まず，２値画像を対象として，並列光ディジタルフィルタリングと並列光論理テンプレートマッチングの２方法を説明し，それぞれの特徴を述べ，処理結果を示した。そして，多値画像処理への拡張方法として，ビット分散配置法，ディジタル / アナログ変換法，多値符号法の３方法について述べた。これらの方法は並列光論理演算法の原理を並列光アナログ演算に応用したもので，並列光論理演算法の拡張性の大きさを示すものである。

第４章では，並列光論理演算法を汎用並列論理演算と結びつける新しい概念，光アレイロジックを考案した。光アレイロジックは並列光論理演算法とアレイロジックとの類似性を積極的に利用したもので，次の特徴を持つ。

1. SIMD形式の種々の並列処理が簡単に記述（プログラム）できる。
2. 処理の種類の記述には処理の内容と結びついた専用の記号を用いる。
3. 記述記号は並列光論理演算法で用いる演算カーネルと１対１に対応し，記述された並列処理は光学系で簡単に実行できる。
4. 画像データを対象とした場合には任意の並列近傍画素間演算を行うことができる。

この章では特に，光アレイロジックの概念，記述法，実現法について記述した。そして，実際の問題を光アレイロジックの手順にしたがって実行し，光アレイロジックの概念が汎用並列処理に対し有効であることを確めた。

第５章では，光アレイロジックをもとにした新しい汎用並列光演算システムOPALSを提案し，その概念を説明した。OPALSの機能上・動作上の特徴は次の通りである。

1. ２次元２値データに対する並列画素間演算・並列近傍画素間演算の実行が可能。
2. データ入出力，演算処理，記憶，表示すべてを２次元データの配列を保ったまま実行する。
3. 光アレイロジックによるプログラミングが可能で，汎用並列演算システムとしての機能を備えている。
4. 並列帰還系により同一データに対する反復処理が可能。

このOPALSは概念的システムであり，具体的なシステムの実現には様々のレベルのシステム構成法が考えられる。そこで，光 / 電子複合型システムと純光学型システムを提案し，それぞれの特徴を述べ，構成に必要な技術，光機能素子を検討した。その結果から，いずれのシステムも近い将来実現し得る汎用並列ディジタル光演算システムとして有望視できることを示した。

第６章では，OPALSのシステム構成をより現実的なものにするために，２つのアプローチによる最適化を試みた。１つは必要最小限の光機能素子によるシステム構成を目標とし，他方は実用的なシステムへの発展性に重点を置いたものである。そして，それぞれの成果として光順序回路技術

によるOPALSと機能モジュール型OPALSを提案した。光順序回路型OPALSは必要最小限の光機能素子によるシステム構成をとるもので，次の特徴を有する。

1. 符号化用と実時間2次元相関演算用の光機能素子のみで構成でき，一時記憶素子は不要。
2. 波長多重技術により光の情報処理能力を最大限に活用できる。
3. OPALSの基本機能はすべて備えており，その特徴をすべてあわせ持つ。

また，機能モジュール型OPALSは実用的なシステムへの発展性を考えたもので，次の特徴を持つ。

1. OPALSの構成要素ごとに機能モジュール化を行い，それらの組合せでシステムを構成する。
2. 各機能モジュールの開発には電子技術を積極的利用するものと仮定している。
3. モジュールの組合せ方により，OPALSはもとより多様なシステムとしての機能をもたすことができる。
4. 第1世代の光コンピュータのプロトタイプと成り得る可能性を有する。

以上のシステム構成を考慮した上で，OPALSの性能の上限を与える要因を考察し，OPALSの達成可能な性能を試算した。その結果，現在構成可能なシステムで，9.0×10^{8} AND operation/sec，機能モジュール型OPALSで1.8×10^{16} AND operation/sec，さらに理想的な光演算システムでは1.8×10^{19} AND operation/secを得た。これらの数値より，OPALSが新しい大容量並列情報処理システムとして十分期待し得る性能を持つものであるとの知見を得た。

光を情報媒体とする情報処理システム — 光コンピュータ — に対する具体的な研究は始まったばかりである。この分野ではシステム機能の観点に立つシステム側の研究とシステムに必要な光機能素子開発に拘るデバイス側からの研究に大別することができる。しかし，現時点では，それぞれの立場における光コンピュータの概念にはかなりの相違が見られる。緒論でも述べたように，種々の目的に対する様々な型式の光コンピュータがあって然るべきであるが，システムおよびデバイス側両者の研究協力がない限り実用可能な光コンピュータの実現は困難であろう。したがって，数ある光コンピュータの概念の中から，システムおよびデバイス側両者が目標とすべき第1世代の光コンピュータの概念の明確化こそ光コンピュータ実現への第1歩であり，解決を急がなければならない課題である。その点で，本研究の成果であるOPALSは第1世代光コンピュータの1候補として有意義なものであり，今後デバイス側からのOPALSの実現可能性に対する検討が必要となろう。

謝　　辞

本研究は，大阪大学工学部応用物理学教室において，一岡芳樹教授の御指導の下に行ったものである。

終わりに臨み，終始懇切なる御指導と御助言を賜りました本学・一岡芳樹教授に深く感謝の意を表わすと共に厚く御礼申し上げます。

本学応用物理学教室・南　茂夫教授，同学精密工学教室・牧之内三郎教授，同学電子工学教室・西原　浩教授には，論文作成にあたり御検討頂き，貴重な御示唆，御教示を頂きました。ここに深く感謝いたします。

著者が本研究を遂行するにあたり，有益なる御助言を頂きました横関俊介講師，雑賀憲昭助手，実験に際し多くの協力を頂いた日高　忍氏，吉原一博氏をはじめ，終始激励と援助を頂きました本学応用物理学教室第5講座の方々に心から感謝いたします。

参　考　文　献

著者発表論文

第1章－第3章

1. 一岡芳樹，谷田　純，“投影法を用いた並列光論理演算法，”第13回画像工学コンファレンス論文集，pp.113-116（1982）.
2. J. Tanida and Y. Ichioka, “Optical logic array processor using shadowgrams,” J. Opt. Soc. Am. 73, 800-809 (1983).
3. J. Tanida and Y. Ichioka, “Optical logic array processor,” Proc. 10th Int. Opt. Comput. Conf. (MIT, Cambridge, MA, Apr. 6-8, 1983), pp. 18-23.
4. J. Tanida and Y. Ichioka, “Image encoding by a computer generated holographic filter,” Proc. Soc. Photo-Opt. Instrum. Eng. 437, 119-124 (1983).
5. Y. Ichioka and J. Tanida, “Optical parallel logic gates using a shadow-casting system for optical digital computing,” Proc. IEEE 72, 787-801 (1984).
6. J. Tanida and Y. Ichioka, “Optical-logic-array processor using shadowgrams. II. Optical parallel digital image processing,” J. Opt. Soc. Am. A 2, 1237-1244 (1985).

第4章，第5章

7. 谷田　純，一岡芳樹，“並列光ロジックアレイプロセッサ，”第14回画像工学コンファレンス論文集，pp.213-216（1983）.
8. 谷田　純，一岡芳樹，“ハイブリッド並列光ロジックアレイプロセッサ，”第15回画像工学コンファレンス論文集，pp.303-306（1984）.
9. J. Tanida and Y. Ichioka, “Optical-logic-array processor using shadowgrams. III. Parallel neighborhood operations and an architecture of optical digital computing system,” J. Opt. Soc. Am. A 2, 1245-1253 (1985).
10. J. Tanida and Y. Ichioka, “OPALS: Optical parallel array logic system,” Appl. Opt. 24 (in press).

第6章

11. J. Tanida and Y. Ichioka, “Optical parallel array logic system. 2: A new system architecture without memory elements,” Appl. Opt. (to be

published).

解　説

12. 一岡芳樹，谷田　純，“ディジタル光演算処理 ― 並列光論理演算法 ―，”（稲場文男編，“光コンピュータ，”オーム社，1985年），pp.102-111.

その他の文献

13. J. W. Goodman, “Introduction to Fourier Optics,” (McGraw-Hill, 1968).
14. 辻内順平，村田和美編，“光学情報処理，”（朝倉書店，1974）.
15. S. H. Lee, ed., “Optical Information Processing Fundamentals,” (Springer-Verlag, 1981).
16. I. E. サザーランド，C. A. ミード，別冊サイエンス，Scientific American 日本版 No.61, 172 (1983).
17. 川合敏雄，星野　力，電子通信学会誌 67, 1290 (1984).
18. 一岡芳樹，応用物理 54, 1019 (1985).
19. 早川尚夫，応用物理 53, 464 (1984).
20. T. Mimura, S. Hiyamizu, T. Fujii, and K. Nanbu, Jpn. J. Appl. Phys. 19, L225 (1980).
21. H. T. Kung, Computer 15 (1), 37 (1982).
22. L. S. Haynes, R. L. Lau, D. P. Siewiorek, and D. W. Mizel, Computer 15 (1), 9 (1982).
23. D. H. Schaefer and J. P. Strong, Ⅲ., Proc. IEEE 65, 129 (1977).
24. A. Huang, Proc. Soc. Photo-Opt. Instrum. Eng. 232, 119 (1980).
25. 石原　聰，島田潤一，桜井健二郎，電子通信学会誌 64, 89 (1981).
26. A. A. Sawchuk and T. C. Strand, Proc. IEEE 72, 758 (1984).
27. A. Huang, Proc. IEEE 72, 780 (1984).
28. 谷田貝豊彦，光学 14, 2 (1985).
29. J. W. Goodman, F. J. Feonberger, S-Y. Kung, and R. A. Athale, Proc. IEEE 72, 850 (1984).
30. 島田潤一，石原　聰，光学 10, 237 (1981).
31. 尾崎　弘，藤原秀雄，“論理数学の基礎，”（オーム社，1980）.
32. J. Grinberg, A. D. Jacobson, W. P. Bleha, L. Miller, L. Fraas, D. Boswell, and G. Myer, Opt. Eng. 14, 217 (1975).

33. J. Feinleib and D. S. Oliver, Appl. Opt. 11, 2752 (1972).

34. A. Seko and A. Sasamori, Appl. Opt. 18, 2052 (1979).

35. M. T. Fatehi, K. C. Wasmundt, and S. A. Collins, Jr., Appl. Opt. 20, 2250 (1981).

36. H. Bartelt, A. W. Lohmann, and E. E. Sicre, J. Opt. Soc. Am. A1, 944 (1984).

37. 岡村廸夫, "解析ディジタル回路," (CQ出版, 1976).

38. A. W. Lohmann, "Optical Information Processing," (Lohmann, Erlangen, Federal Republic of Germany, 1979).

39. H. Fleisher and L. I. Maissel, IBM J. Res. Develop. 19, 98 (1975).

40. 南谷 崇, "PLAの使い方," (産報出版, 1978).

41. W. H. Lee, Progress in Optics 16, 121 (1978).

42. B. K. Jenkins, A. A. Sawchuk, T. C. Strand, R. Forchheimer, and B. H. Soffer, Appl. Opt. 23, 3455 (1984).

43. W. A. Shurcliff, "Polarized Light, Production and Use," (Harvard University Press, 1962).

44. 小山次郎, 西原 浩, "光波電子工学," (コロナ社, 1978).

45. M. Born and E. Wolf, "Principles of Optics," (Pergaman Press, 1974).

46. 伊賀健一, 及川正尋, 三沢成喜, O plus E No.50, 82 (1984).

47. A. Rosenfeld and A. C. Kak, "Digital Picture Processing," (Academic Press, 1976).

48. W. K. Pratt, "Digital Image Processing," (Wiley, 1977).

49. 一岡芳樹, 応用物理 51, 573 (1982).

50. 木戸出正継, 坂出勝彦, 日経エレクトロニクス 295, 179 (1982).

51. K. Preston, Jr., M. J. B. Duff, S. Levialdi, P. E. Norgren, and Jun-ichiro Toriwaki, Proc. IEEE 67, 826 (1979).

52. A. Rosenfeld, Computer 16 (1), 14 (1983).

53. 渡部弘之, "コンピュータ設計技術[I]," (CQ出版, 1973).

54. J. C. Logue, N. F. Brickman, F. Howley, J. W. Jones, and W. W. Wu, IBM J. Res. Develop. 19, 110 (1975).

55. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, Appl. Opt. 18, 149 (1979).

56. D. Psaltis and D. Casasent, Appl. Opt. 18, 163 (1979).

57. C.C.Guest and T.K.Gaylord, Appl. Opt. 19, 1201 (1980).

58. B.K. Jenkins, P. Chavel, R.Forchheimer, A.A.Sawchuk, and T.C.Strand, Appl. Opt. 23, 3465 (1984).

59. 末田　正, "光エレクトロニクス," (昭晃堂, 1985).

60. 赤坂洋一, 西村　正, 日経エレクトロニクス 379, 229 (1985).

61. 中野元雄, 応用物理 54, 652 (1985).

62. 梅恒真祐, 光学 14, 11 (1985).

63. 窪田恵一, 西田信夫, 光学 14, 19 (1985).

64. K. Iga, M.Oikawa, S.Misawa, J.Banno, and Y.Kokubun, Appl. Opt. 21, 3456 (1982).

65. 日経エレクトロニクス 376, 169 (1985).

66. 西原　浩, 春名正光, 栖原敏明, "光集積回路," (オーム社, 1985).