計算機アーキテクチャ　63－2
(1986・11・20)

A Real-Time Image Processing using Optically-Connected 3D-VLSI Architecture

藤田　聡　　相原玲二　　阿江　忠
Satoshi FUJITA,　Reiji AIBARA　and　Tadashi AE
広島大学工学部
Faculty of Engineering, Hiroshima University

1. Introduction

Development of VLSI technologies is extremely fast, and it has already realized the VLSI chip with more than one million elements. But, as is well known, there are serious physical limitations about density of elements on a chip in two dimensional manner, and the quantitative enlargement of integrity may have an end in the near future. In order to break the current status many qualitative researches are attempted, and one of them is the realization of three-dimensional, multi active layered, VLSI.

In general, the three-dimensional VLSI seems to have the following advantages;

(1) improvement of density per unit volume on a chip, and
(2) improvement of interconnection among data on gate or memory cell.

The first feature is achieved not only by increasing of the number of layers, but also by decreasing of the wiring area of the whole chip using vertical wiring ingeneously. The latter is desirable, because the higher the density of chip is, the more serious the increasing of wiring area is. On the other hand, the second feature plays a role of massively parallel architecture with pipelined processing by assigning a stage to a layer, or with parallel processing by assigning a task to a layer. At the present, one of the most available applications of three-dimensional VLSI is two-dimensional data processing, since it needs denser vertical connection than other applications. One example of actually implemented two-dimensional data processing system with three-dimensional structure is Grinberg's 3D Computer[GRI.84]. This system adopts a conventional image processing architecture similar to CLIP-4[DUF. 78] or MPP[BAT.80], and realizes a 32x32 element array on a chip. It is, moreover, realized by a three-dimensional structure with stack of chips without laminating technique such as SOI(Silicon-On-Insulator), because of ease of implementation.

On the other hand, recently, researchs on three-dimensional VLSI, in which active layers are connected optically to each other, have been coming up. These attempts utilize optical property such as refraction, e.g., Hasegawa et al. shows that area-time complexity (AT^2) of FFT (Fast Fourier Transform) algorithm could be improved, by applying hologram to inter-stage routing of FFT network [HAS.86]. The interlayer connections using hologram or lense, however, are still not clear on complexity, and therefore, we concentrate on property of its isotropic radiation in this paper, instead of other optical properties.

In the following discussions, we attempt to characterize the optical interlayer connection of the three-dimensional optically-connected VLSI, and to show that it realizes several useful functions. Moreover, as an application of these functions, we propose a fast template matching algorithm, and estimate the avairability of this algorithm by simulation. And finally, we try to construct a real-time template matching system, using pipeline technique.

In the section 2, we give several notations and definitions for the consequent sections. In the section 3, we formulate interlayer functions of three-dimensional optically-connected VLSI. In the section 4, we give a new and fast template matching algorithm using functions formulated in the section 3, and estimate this algorithm by simulation. And in the section 5, we show a real-time three-dimensional optically-connected VLSI template matching system using pipelining.

2. Notations and Definitions

To formulate interlayer functions we follow the notations of picture operators defined by Yokoi, et al. [YOK.77a][YOK.77b], who have also shown its avairability for analysis of picture processing algorithms. In this paper therefore, we define algebraicly the picture and the operations for picture processing as in Yokoi, et al. Moreover, in the following discussions, we restrict the target of operations to binary pictures, in order to simplify the structure of processing elements associated with pixels, and to increase the density of processing elements on a chip, as possible as we can.

[Definition 2.1](binary picture)
A binary picture is defined by $I \times I \rightarrow \{0,1\}$, where I is set of integer. $(i,j) \in I \times I$ denotes a picture element, located at i-th low and j-th column, and $A = \{a_{ij}\}$ denotes the picture, on which value of picture element (i,j) is given by $a_{ij} \in \{0,1\}$. □

[Definition 2.2]
(equality/inequality of two pictures)
For any given pictures A and B, we define equivarence and inequivarence of pictures as follows;
for any $a_{ij} \in A$, $b_{ij} \in B$,
$a_{ij} = b_{ij}$ $A = B$ (equality)
$a_{ij} \leq b_{ij}$ $A \leq B$ (inequality).□

Next we note several binary picture operators shown in [YOK.77b], with their meanings;

[Definition 2.3](picture operators)
(a) unary operators
identity $B = I(A)$ $b_{ij}=a_{ij}$ for any $(i,j) \in I \times I$
nagation $B = N(A)$ $b_{ij}=1-a_{ij}$ for any $(i,j) \in I \times I$
shift $B = T[p,q](A)$ $b_{ij}=a_{i-p\ j-q}$ for any $(i,j) \in I \times I$
(b) binary operators (figure 1)
and $C = A \owedge B$ $c_{ij}=a_{ij}\ b_{ij}$ for any $(i,j) \in I \times I$
or $C = A \ovee B$ $c_{ij}=a_{ij}\ b_{ij}$ for any $(i,j) \in I \times I$
difference $C = A \ominus B$ $c_{ij}=a_{ij}-b_{ij}$ for any $(i,j) \in I \times I$
(where $a_{ij}-b_{ij} = a_{ij}$ when $b_{ij} = 0$
$= 0$ when $b_{ij} = 1$)□

[Definition 2.4]
(composition of operators)
For any operators O_1 and O_2, we define the associative composition of operators as follows;
$$(O_1 \circ O_2)(A) = O_1(O_2(A)),$$
$$O_1{}^2(A) = (O_1 \circ O_1)(A).$$ □

In the following discussions, we use operator R(right shift) and D(down shift) instead of operator T[i,j], i.e. $T[i,j] = R^p \circ D^q$, for simplicity. But, due to isotropic radiation of light, a shift operation for arbitrary direction cannot be realized optically. Hence the shift operation must be realized electrically.

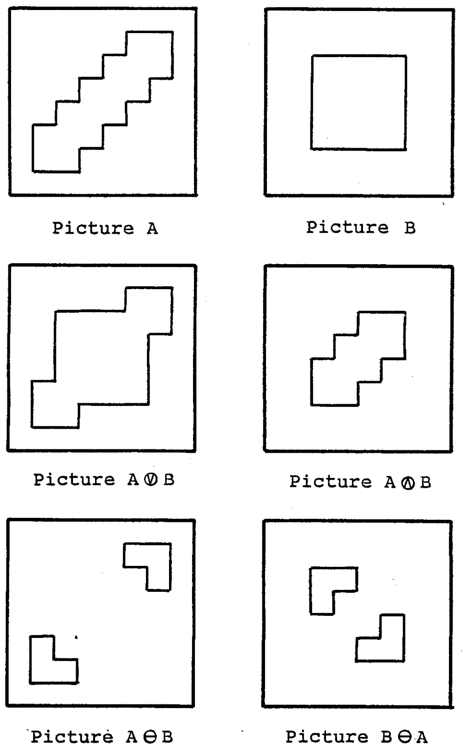


Figure 1 An Example of Binary Picture Operators

3. Optical Interlayer Functions

3.1 Physical Assumptions

Before modeling of the three-dimensional integrated circuit, we introduce several physical assumptions;

[Assumption 3.1](configuration)
We assume the three-dimensional integrated circuit to have k layers. LED(Light Emitting Diode) and PD(Photo Diode) are both associated with each of processing elements. These elements construct an n x n array on each layers, and this configuration is the same in all layers (figure 2). □

[Assumption 3.2]
The light emitted by any LED on any layers propagates to both upper

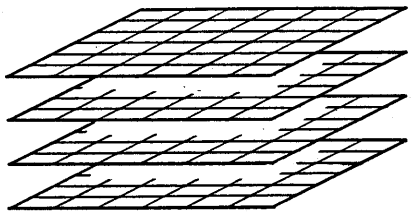


Figure 2 Configuration of three-dimensional VLSI

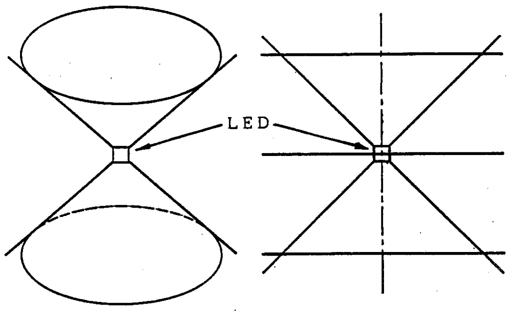


Figure 3 Directivity of LED

and lower directions with some isotropic radiation. This light can be detected by any PD contained in this radiation (figure 3). □

[Assumption 3.3]

SiO_2 and Si are both sufficiently transparent for the light emitted by LED. Hence, in the following discussions, we disregard the attenuation of light. □

3.2 Formulation of Interlayer Functions

Every element in this model has one bit information(memory), and is associated with picture element, respectively. Moreover each of layered n x n element arrays is associated with a distinct picture plane, i.e. the domain of binary pictures is restricted from I x I to U x U, where U = {1,...n}. Any element that does not belong to the picture plane has value 0, and any (i,j) element belongs to the picture plane has value 0 or 1. All elements with label (i,j), where $(i,j) \in U \times U$, are standing in vertical line. Each of elememts belongs to picture planes can exchange its information for that of several elements, which are neighborhood of (i,j) elements on the other layers. This neighborhood is a function of interlayer distance and is defined as follows using assumptions 3.1-3.3;

[Definition 3.1](neighborhood)

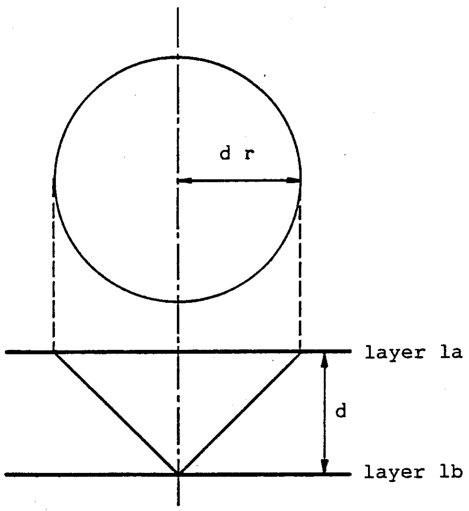


Figure 4 Definition of neighborhood

Suppose two layers, i.e., one layer l_a on which picture $A = \{a_{ij}\}$ is stored, and the other layer l_b on which picture $B = \{b_{ij}\}$ is stored. The distance between l_a and l_b is denoted by d. Then, by emission of layer l_a, an element $b_{ij} \in B$ on layer l_b can accept the information from elements $a_{i+p\ j+q} \in A$ on layer l_a (figure 4). Here integer p and q must satisfy the following condition;

$$p^2+q^2 \leq (dr)^2.$$

(r is a real constant given previously) □

By the property of PD, the function for information from neighborhood elements is actually restricted to some of threshold functions. In this paper, we assume this threshold function to be an OR function weighted uniformly.

[Definition 3.2]

(optical interlayer functions)

By using notations defined above, we can formulate optical interlayer functions as follows;

$$\{ \underset{(p,q) \in Sd}{\circledvee} (R^p \circ D^q) \}$$

Furthermore if nagation is permitted, we can also obtain the following operator;

$$N \circ \{ \underset{(p,q) \in Sd}{\circledvee} (R^p \circ D^q) \} \circ N$$
$$= \{ \underset{(p,q) \in Sd}{\circledvee} (R^p \circ D^q) \}.$$

Here $Sd = \{(i,j) \mid i^2 + j^2 \leq (dr)^2 \}$. □

[Example 3.1]

If the size of r is given as $\sqrt{2}/2 \leq r < 1$, then we get

(a) $b_{ij}=a_{ij}$ for any $(i,j) \in I \times I$, when d is equal to 1 (1)

(b) $b_{ij} = \bigvee_{p=-1}^{1} \bigvee_{q=-1}^{1} a_{i+p\ j+q}$ (2)

for any $(i,j) \in I \times I$, when $d = 2$.

Here, V denotes the OR function for all value of $-1 \leq p \leq 1$ and $-1 \leq q \leq 1$. In (2), if we replace a_{ij} and b_{ij} by a_{ij} and b_{ij}, then we get

(c) $b_{ij} = \bigwedge_{p=-1}^{1} \bigwedge_{q=-1}^{1} a_{i+p\ j+q}$ (3)

for any $(i,j) \in I \times I$, when $d = 2$.

Here $\wedge$ denotes the AND function for all value of $-1 \leq p \leq 1$, $-1 \leq q \leq 1$. (1) represents identical function. From (2) and (3), we obtain the following formula;

$$B = \Big(\mathop{\textcircled{V}}_{p=-1}^{1} \mathop{\textcircled{V}}_{q=-1}^{1} R^{p} \circ D^{q} \Big)(A) = M(A)$$

$$B = \Big(\mathop{\textcircled{V}}_{p=-1}^{1} \mathop{\textcircled{V}}_{q=-1}^{1} R^{p} \circ D^{q} \Big)(A) = P(A),$$

here, M and P are called the basic disjunction filter and the basic conjunction filter, respectively [YOK. 77a](Figure 5, Appendix 1). □

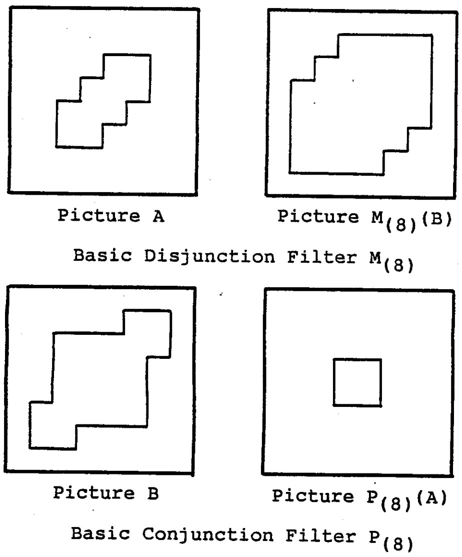

Picture A
Picture $M_{(8)}(B)$
Basic Disjunction Filter $M_{(8)}$
Picture B
Picture $P_{(8)}(A)$
Basic Conjunction Filter $P_{(8)}$

Figure 5 Basic conjunction(disjunction) filter

Actually, it seems possible to realize the interlayer distance as d =2, considering the sensitivity of PD and the power of LED. Therefore, in the following discussions, we use the only functions in the example described above. In the figure 6, we show a basic element configuration, necessary for implementation of above functions. As shown, alternative

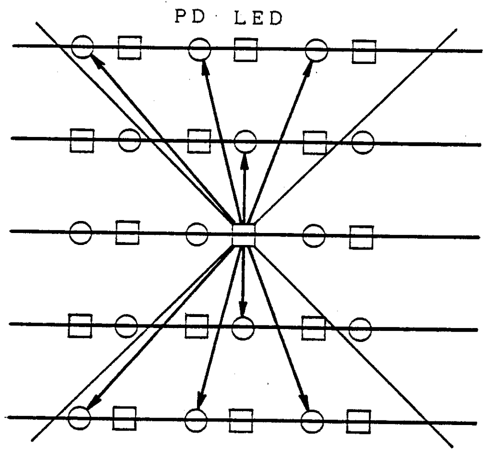


Figure 6 Arrangement of LED and PD

placement of LED and PD make it possible to implement above functions.

3.3 Properties of Optical Interlayer Functions

From the section 3.3, we obtain following properties on the optical interlayer function;

(1) Optical interlayer function is isotropic.
(2) The larger the distance between the emitting layer and the detecting layer is, the larger the neighborhood is.
(3) The figure of neighborhood is a circle.

(1) means that it does not permit anisotropic operations such as shift operations, and it becomes a large demerit for algebraic structure of operations. On the contrary, since it includes shift operations for all directions, it will be expected to decrease, in some case, the number of steps of operation. On (2) we expect the decreasing of steps by using more layers, or by simultaneous execution of different operations. Moreover, (3) is expected for the special applications that require a degitaized circle.

4. Applications for Template Matching

As shown in the section 3, isotropic operators, such as the basic disjunction filter M, the basic conjunction filter P and identical operator I, are realized naturally using optical interlayer connection. In this section, we concentrate on template matching algorithm as an application to show the avairability of these operators. Since proposed algorithm operates in shift and judge manner, basicaly, it fits for shifted pictures, but does not fit for rotated or scaled pictures.

In this section, we denote a template picture by A, and an input picture for matching by B. Moreover we assume that equivalence/inequivalence judgement of two pictures could be done in constant time, not affected by its size. Here any 1-elements in picture A and B are assumed to be included in picture plane.

4.1 Searching of Same Pictures

We can judge whether input picture B is a shifted picture of template picture A, by following algorithm.

<u>Algorithm 4.1</u>

```
begin
  $x \leftarrow 0$; { initialization of counting }

  { enlarging phase }
  $\mathbf{C} \leftarrow I(\mathbf{A})$; $c \leftarrow 1$;
   while $\mathbf{B} \ominus \mathbf{C} \neq \mathbf{0}$
     do  $\mathbf{D} \leftarrow \mathbf{C}$; $\mathbf{C} \leftarrow \mu_{(8)}(\mathbf{D})$; $c \leftarrow c+1$
     od;

  { matching phase }
   SEARCH( c , A , B );

   Exit.: if $x \neq 0$  then " matched "
                         else " not matched "
          fi
end
```

<u>procedure SEARCH(c , A, B);</u>

```
begin
  { move picture A to init. position }
  $\mathbf{A} \leftarrow T[-c,-c](\mathbf{A})$; $p \leftarrow -c$; $q \leftarrow -c$;

  { move picture A along to circle with
    radius = c and compare with B }

  while $q \leq c$
   do if $\mathbf{A} = \mu_{(8)}(\mathbf{A}) \owedge \mathbf{B}$ then $x \leftarrow x+1$
      fi
      $q \leftarrow q+1$;  $\mathbf{A} \leftarrow R(\mathbf{A})$
   od;
  while $p \leq c$
   do if $\mathbf{A} = \mu_{(8)}(\mathbf{A}) \owedge \mathbf{B}$ then $x \leftarrow x+1$
      fi
      $q \leftarrow q+1$;  $\mathbf{A} \leftarrow D(\mathbf{A})$
   od;
  while $q \geq -c$
   do if $\mathbf{A} = \mu_{(8)}(\mathbf{A}) \owedge \mathbf{B}$ then $x \leftarrow x+1$
      fi
      $q \leftarrow q-1$;  $\mathbf{A} \leftarrow R^{-1}(\mathbf{A})$
   od;
  while $p \geq -c$
   do if $\mathbf{A} = \mu_{(8)}(\mathbf{A}) \owedge \mathbf{B}$ then $x \leftarrow x+1$
      fi
      $p \leftarrow p-1$;  $\mathbf{A} \leftarrow D^{-1}(\mathbf{A})$
   od
end
```

Figure 7 (a) Algorithm 4.1

Algorithm 4.1 is composed of two phases, the enlarging phase and the matching phase. In the enlarging phase, we continue to operate basic disjunction filter M on picture A until the result picture covers input picture B, i.e., until $M^c(A) \leq B$ where c is the times of M operator has been applied. In the matching phase, we shift picture A to the position, where is away from original position with 8-neighbor distance c, then, continue the shift and the match operation keeping this distance from original position. To verify correctness of this algorithm, we will show following three lemmas.

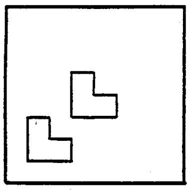
[lemma 1] $M^n(A) = \{\ovee_{p=-n}^{n} \ovee_{q=-n}^{n} (R^p \circ D^q)\}(A)$

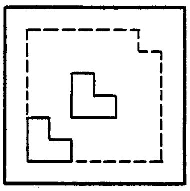
(Proof) We prove it by induction.
(i) When n = 1, formula (1) is true.
(ii) If (1) is true when n = k, then when n = k+1,

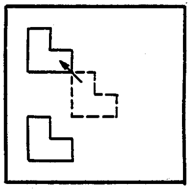
$$M^{k+1}(A) = (M \circ M^k)(A)$$
$$= \{(\ovee_{i=-1}^{1} \ovee_{j=-1}^{1} (R^i \circ D^j)\,(\ovee_{p=-k}^{k} \ovee_{q=-k}^{k} (R^p \circ D^q))\}(A)$$
$$= \{\ovee_{p=-(k+1)}^{k+1} \ovee_{q=-(k+1)}^{k+1} (R^p \circ D^q)\}(A)$$

From (i)(ii), formula (1) is true for any $n \in N$. □

[lemma 2] $(R^p \circ D^q)(A) \ominus M^r(A) = 0$


(1) Template A and Input Picture B


(2) After Enlargement Phase: Picture drawn by broken line is minimum $M_{(8)}{}^{c}(A)$ covering picture B.


(3) Initialization in Matching Phase: Move template A to the edge of circle with radius=c.

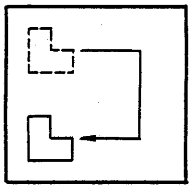

(4) Matching Phase: Detect matched picture by shifting on the circle.

Figure 7 (b) Explanation of algorithm 4.1

iff $(p,q) \in \{(i,j) \mid -r \le i \le r,\ -r \le j \le r\}$

(Proof)($\leftarrow$) It is trivial from lemma 1.
($\rightarrow$) We verify it by contradiction. In the other word, we will lead inconsistency from ;

$$(R^p D^q)(A) \ominus M^r(A) = 0$$
$$(p,q) \in \{(i,j) \mid i < -r\} \cup \{(i,j) \mid r < i\} \cup \{(i,j) \mid j < -r\} \cup \{(i,j) \mid r < j\}.$$

Now we put $B = (R^p \circ D^q)(A) = \{b_{ij}\}$, $C = M^r(A) = \{c_{ij}\}$. Then in the case of $(p,q) \in \{(i,j) \mid r < i\}$,

$$\begin{aligned} &\max(i \mid b_{ij} = 1) \\ &= \max(i \mid a_{ij} = 1) + 1 \\ &> \max(i \mid a_{ij} = 1) + r \\ &= \max(i \mid c_{ij} = 1). \end{aligned}$$

Hence $B \ominus C = 0$, and it is inconsistent. For the other cases, we can also get inconsistency in the same way. Therefore lemma 2 has been verified. □

[lemma 3] $(R^p \circ D^q)(A) \ominus M^{r-1}(A) = 0$ and $(R^p \circ D^q)(A) \ominus M^r(A) = 0$
$$(p,q) \in \{(i,j) \mid -r \le i \le r,\ j = \pm r\} \cup \{(i,j) \mid i = \pm r,\ -r \le j \le r\}$$
(Proof) It is trivial from lemma 2 □

To estimate the time complexity of algorithm 4.1, we introduce several assumptions about hardware (detail of these assumptions has shown in Appendix 2).

[Assumption 4.1]

(1) Comparison(equality/inequality) of two pictures is performed in one step, independent of other operations.
(2) Interlayer operators (I, M, N) are perfoemed in one step. But when operataor M is applied repeatedly, it takes three steps because it needs to pump up the picture to original position.
(3) Binary operators are all performed in one step. Here operator and M are able to performed in parallel.
(4) The shift operator R and D are performed in two steps. □

From Lemma 1-3 and Assumption 4.1, we obtain following theorem about algorithm 4.1.

[Theorem 1] Algorithm 4.1 can judge correctly whether picture A is a shifted picture of picture B, with O(n) time complexity. Here n is the size of the picture plane edge.

(Proof) Since correctness of the algorithm is obvious from lemma 3, here, we verify about time complexity. If the template picture is larger than picture 0, the result picture of n times M operations covers the whole picture plane. Therefore the enlarging phase is repeated at most n times, and it requires at most 3n steps(from Assumption 4.1). On the other hand, in the matching phase, shift operation is repeated at most 9c times, where c is the times of enlarging and c is smaller than n/2. Hence the matching phase requires at most $(5+2)\text{x}(9/2)\text{xn} = 31.5\text{n}$ steps. Therefore this algorithm can judge correctly whether picture A is a shifted picture of picture B, with O(n) time complexity.

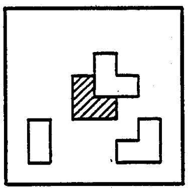
Algorithm 4.1 performs comparison between template A and whole input picture B. In general, however, input picture B is composed of several connected components, and it is often required to compare template A to each connected component of input picture B. In the last half of this section, therefore, we show an extended algorithm applicable to such case.

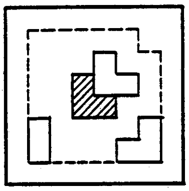
Algorithm 4.2 is mainly composed of three phases, the enlarging phase, the matching phase, and the extracting phase. The enlarging and the matching phases are similar to algorithm 4.1. The extracting phase is based on lemma 3, i.e., when it satisfies enlarging condition for input picture B and we are searching on the circle of radius c, template A is compared to any connected component of B that has 1-element on this circle. Hence

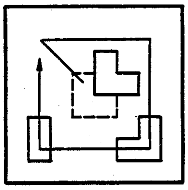
Algorithm 4.2

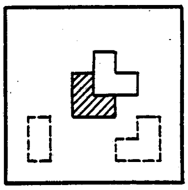
```
begin
   $x \leftarrow 0$;  { initialize of counting }
   while $\mathbf{B} \neq \mathbf{0}$
     do   { enlarging phase }
          $\mathbf{C} \leftarrow \mathrm{I}(\mathbf{A})$; $c \leftarrow 1$;
          while $\mathbf{B} \ominus \mathbf{C} \neq \mathbf{0}$
            do  $\mathbf{D} \leftarrow \mathbf{C}$;  $\mathbf{C} \leftarrow \mu_{(8)}(\mathbf{D})$;
                $c \leftarrow c+1$
            od;

          { matching phase }
          SEARCH( c , A , B );

          { extracting phase }
          $\mathbf{C} \leftarrow \mathbf{C} \ominus \mathbf{D}$;
          $\mathbf{D} \leftarrow \mathbf{C} \otimes \mathbf{B}$;
          $\mathbf{C} \leftarrow \mu_{(8)}(\mathbf{D}) \otimes \mathbf{B}$;
          while $\mathbf{C} \neq \mathbf{D}$
            do  $\mathbf{D} \leftarrow \mathbf{C}$;  $\mathbf{C} \leftarrow \mu_{(8)}(\mathbf{D}) \otimes \mathbf{B}$
            od;
          $\mathbf{B} \leftarrow \mathbf{B} \ominus \mathbf{C}$;
     od;

 { x is the number of connected components
   matched with  template A }

end
```

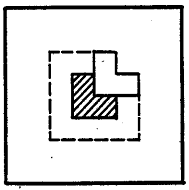
Figure 8 (a) Algorithm 4.2


(1) Template A (shown with diagonal line) and Input Picture B


(2) After Enlargement Phase


(3) Matching Phase


(4) Extraction Phase: Components drawn by broken line are deleted.


(5) After Enlargement Phase

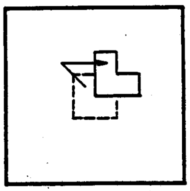

(6) Matching Phase

Figure 8 (b) Explanation of algorithm 4.2

after searching on this circle, we may delete these connected components from the target of comparison. We operate this extraction using picture propergation algorithm shown in [YOK.77b]. In this way, the candidate area, where connected components could be exist, could be restricted, and the number of connected components could be obtained.

Although this method is similar to binary search method in a sense of area restriction, it is more avairable than binary search method, especially for the picture with many connected components. By applying this method, execution time for judgement could be shortened, for some picture B satisfying following conditions.

[Theorem 2] Algorithm 4.2 can judge correctly with $O(n \times \min(m,n) + \min(md,n^2))$ time complexity, how many connected components of input picture B exist, that is a shifted picture of template A. Here n is the size of picture plane, m is the number of connected components of B, and d is the maximum diameter of each connected component.

(Proof) Since correctness of the algorithm is obvious from lemma 3, here, we verify about time complexity. When m is smaller than n, the enlarging phase is repeated at most m times, and when m is larger and equal than n, this phase is repeated at most n times. This phase requires at most 3n steps per one repetition, hence, it requires at most $3n \times \min(m,n)$ steps(from Assumption 4.1). It is the same for the matching phase as the enlarging phase, i.e., it requires at most 61.5n steps.

On the other hand, since picture propergation algorithm has at most $|B|$ repetition, where $|B|$ is the number of 1-elements on the input picture, it requires at most $6|B|$ steps. The extracting phase, therefore, requires at most $6\min(md, n^2)$ steps, here n^2 is the number of all elements in the picture plane.

Hence theorem 2 has been verified.

4.2 Evaluation by Simulation

As is shown in the theorem 2, algorithm 4.2 is expected to be faster than conventional methods. In this section, we evaluate the difference of execution time among two methods by simulation. The size of picture plane is set to 32x32 (n=32) and input picture B is made by producing random number. It is assumed that template A is 2x2 square, and that the algorithm stops after completing of judgement for all connected components.

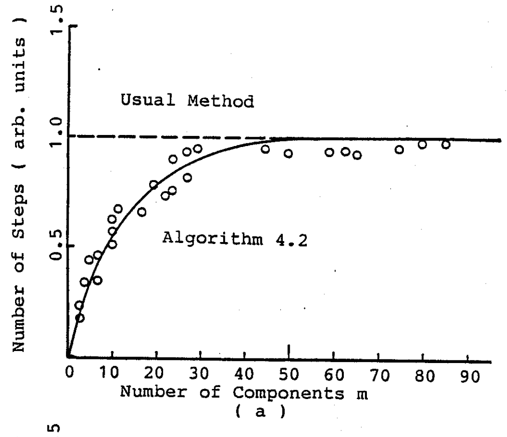
Figure 9(a) shows the number of equivalency comparison (the matching phase) for both algorithms, setting number of connected component m on horizontal axis. Although the number of equivalency comparison is constant(30x30) not affected by m in conventional method, in algorithm 4.2, this number is asymptotic to this constant value according to increase of m. On the other hand, figure 9(b) compares the whole steps of two algorithms, based on Assumption 4.1. Since algorithm 4.2 executes more operations than usual method such as enlarging or extracting, it becomes inefficient for large m ($\geq$ mc). It is, however, expected that mc increases for larger n and this algorithm would be sufficiently effective. Moreover if the matching phase and the other phases are executed concurrently (in pipelining), execution time will be shorten and the curve in figure 9(b) approaches to one shown in figure 9(a).

5. Real-time oparation using pipeline

As is well known, pipeline is effective method for data processing with many input data. In this section, we linearly allocate several three-dimensional VLSI, which performs above algorithm, and verify its availability by estimating response time on pipeline.

Let template size is smaller than k x k(k $\leq$ n). Since the number of template positions is at most n^2, if we could perform template matching in parallel with parallelism n^2, all operations might be completed in constant time. In fact, however, considering about the propergation delay of wire and the other several factors, pipeline seems rather realistic than such parallel processing.

Figure 10 shows the three-dimensional pipeline system with direction for vertical axis. On the three-dimensional VLSI, it is possible to make bandwidth for vertical direction to be n^2, therefore, when the size of data is n^2, the time required for inter-stage communication might be constant. Moreover from Assumption 4.1, comparison to the template with



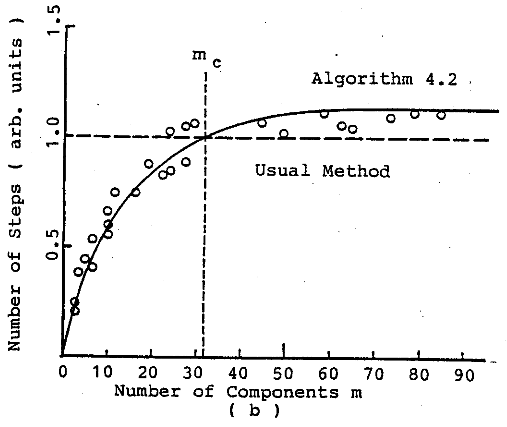


Figure 9 Result of Simulation

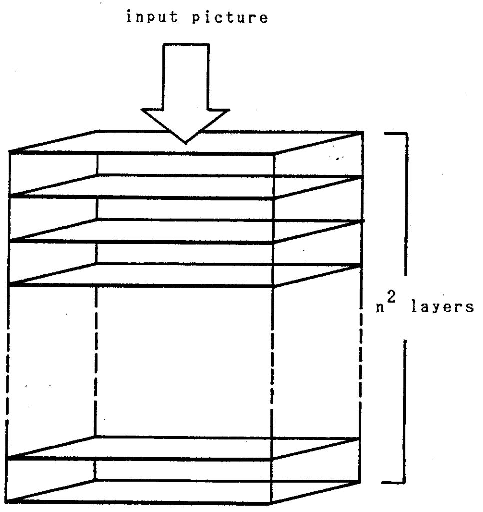


Figure 10 The vertical pipeline system

fixed position takes constant time. Hence we can obtain response time = n^2, throughput = n^2 in this vertical pipelining, however, this system requires $O(n^2)$ layers and it is infeasible.

In this paper, we concentrate on more feasible system, at which bandwidth is restricted to n (figure 11). As this system requires O(n) communication time, processing time in each processor may be O(n), i.e., operations for each low (or each column) could be done. It reduces the number of processors from n^2 to n. Moreover if it might perform input/output, enlarging, matching and extracting concurrently, the number of processors could be reduced from n to m by applying algorithm 4.2, where m is the number of connected components of input picture. The timing chart in usual method is shown in figure 12(a), and that in algorithm 4.2 is shown in figure 12(b). The number of steps in figure 12 are based on Assumption 4.1.

Let the time required for matching t_{search}(= 3n), input/output t_i = t_o(= 2n), enlarging t_{enl}($\leq$ 3n), and extracting t_{ext}($\leq$ $6k^2$). The responce time of usual system(shown in figure 12(a)) is represented by $(nt_i)+t_{search}$ = $O(n^2)$, and of new system(shown in figure 12(b)) is represented by m x $(t_i+t_{enl}+t_{ext})+t_{search}$ = $O(m(n+k^2)+n)$. Hence sufficient condition for that the response time of new system is shorter than usual one, is given as follows;

$$\begin{aligned} m &\leq (t_i \text{ x } n)/(t_i + t_{enl} + t_{ext}) \\ m &\leq 2n^2/(2n + 3n + 6k^2) \\ &= 2n/\{ 5 + (6k^2/n) \} \\ \therefore m &\leq 2n/(5 + 6k^2) \end{aligned}$$

Therefore when $m \leq 2n/(5 + 6k^2)$, it is effective to apply algorithm 4.2 reducing the response time. On the other hand, since each operations are permitted to perform concurrently, throughput is bounded by the longest operation t_{search}(= 3n), and it is the same as usual method, i.e. $O(n^2)$.

Extention for the case of many templates is quite easy. Extended system construction is shown in figure 13. Since the size of any templates are smaller than k x k, we adopt the k x k square as a common template. All templates share their north and west edges with this square, and the common template is put on south-east corner of the picture plane to be enlarged. In the extended system, therefore, the candidate area is restricted to north and west direction. When the number of templates is h and the response time of the system for one template is given by t_{res}, the response time for extended system is at most t_{res} + 3k(h-1). Here k is the size of common template. Throughput of this system is the same as above systems.

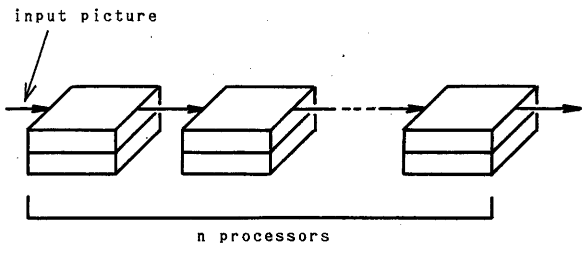


Figure 11 The horizontal pipeline system

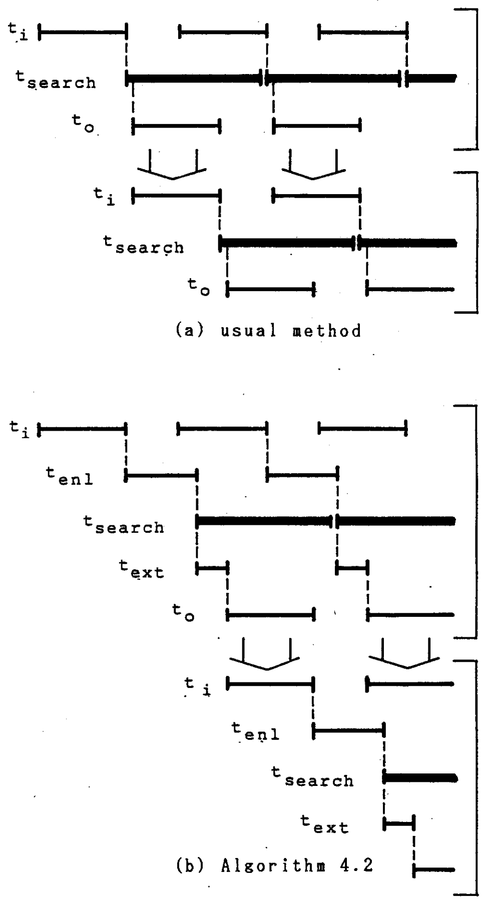


Figure 12 Time chart of operations

6. Conclusion

In this paper we concentrated on isotropic interlayer functions of three-dimensional optically-connected VLSI, and showed template matching algorithm and its implementation, as an application.

To realize this system, there are many problems to be solved, e.g., the establishment of SOI technique, the radiation of heat, and the control of emitting angle of LED, and so on. But, since this fabrication technique requires no vertical wiring essentially, it has a great deal of advantages for system construction, e.g. two times more layers can be realized with same laminating technology. In addition to this merit, the algorithm proposed in this paper is meaningful because it uses only the essential property of optical interlayer connection. Moreover, the interlayer function is one of memory-to-memory operation, and is avairable as an architecture.

For the image processing, we need to discuss a lot of problems such as rotation/scaling of input pictures and similarity detection algorithms such as SSDA method.

[Acknowledgement]

The authors wish to thank research staffs of Integrated System Research Center of Hiroshima University, especially, Professors M. Hirose, M. Yamanishi for discussion of device technology, and to thank K. Ijichi for his simulating works.

[References]

[BAT.80] K. E. Batcher, " Design of a Massively Parallel Processor ", IEEE Trans Comput., Vol.c-29, No.9, pp.836-840 (Sep. 1980).

[DUF.78] M. J. B. Duff, " Review of the CLIP image processor system ", AFIPS Conf. Proc., Vol.47, pp1055-1060 (1978).

[GRI.84] J. Grinberg, et al., " A Cellular VLSI Architecture ", COMPUTER, Vol.17, No.1, pp.69-81 (Jan. 1984).

[HAS.86] M. Hasegawa and Y. Shigei, " $AT^2=O(N\log^4N)$, $T=O(\log N)$ Fast Fourier Transform in A Light Connected 3-Dimensional VLSI ", IEEE Conf. Proc. of the 13th Annual Int. Symp. on Comput. Architecture, pp.252-260 (June 1986).

[HAY.83] Y. Hayashi, " Expectation of Three Dimensional IC ", J. of IECE, Vol.66, No.8, pp.831-834 (Aug. 1983).

[YOK.77a] S. Yokoi, et al., " Algebraic Structure of the Operation System of Degitized Pictures and its Application (1)--Formulation and Properties of Fundermental Operations--", Trans IECE, Vol.J60-D, No.6, pp.411-418 (June 1977).

[YOK.77b] S. Yokoi, Doctral dissertation, Nagoya University, 1977.

Appendix 1

<u>the basic disjunction filter M and the basic conjunction filer P</u>

Let set of binary pictures is P_B. We define that the disjunction filter is the operator, which transforms picture A to picture B (for any A, B P_B) according to formula(1).

$$B = LS(A) \qquad (1)$$
$$b_{ij} = \bigvee_{(p,q)\in S} a_{i-p\ j-q}$$
$$\text{for any } (i,j)\in I \times I \quad (S \subset I \times I)$$

Similary we define that the conjunction filter is the operator, which transforms picture A to picture B according to formula(2).

$$B = LP(A) \qquad (2)$$
$$b_{ij} = \bigwedge_{(p,q)\in S} a_{i-p\ j-q}$$
$$\text{for any } (i,j)\in I \times I \quad (S \subset I \times I)$$

On (1) and (2), if S is given by
$S = \{(i, j)|\ \ |i| \leq 1,\ |j| \leq 1\ \}$,
basic disjunction filter $M_{(8)}$ (for (1)), and basic conjunction filter $P_{(8)}$ (for (2)) are obtained, respectively. In this paper, subscription (8) that means 8-neighborhood is omitted for simplicity.

Appendix 2

<u>assumptions about hardware</u>

In this paper we assumed the three-dimensional VLSI to has 5 layers. Although interlayer communication among arbitral layers is performed in 1 step, distance of layers could communicate is restricted to 2 (figure 13(a)).

Since binary picture operators are realized electrically except for the OR operation, before operation, one operand must be loaded to temporal register on operational layer (b)(the fourth layer is assigned). If M operator is applied repeatedly, two times of I operations are required to pump up the picture to original position, then it take 3 steps totally (c). Moreover since shift operations for arbitral directions are required in this algorithm, we assigned this function to the second layer (D^{-1}, D) and to the third layer (R^{-1}, R), respectively. Each shift operation takes 2 steps(d).

The comparison of two pictures is done by reading out the result of difference(inequivalency) or excusive-or(equivalency) on the fourth layer. This operation uses wired-or mechanism, and it takes only 1 step(d).

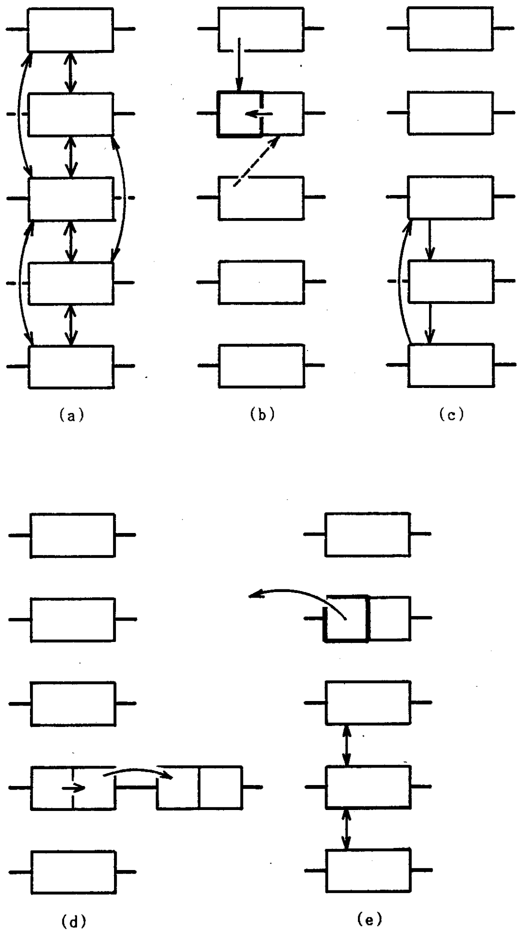


Figure 13 Primitive operations