Review of the CLIP image processing system

by M. J. B. DUFF
University College London
London, England

INTRODUCTION

In this paper, the terms "pattern recognition" and "image processing" will be considered only in relationship to optical or visual images, which will be expressed as two dimensional arrays of numbers, each number representing the image intensity in the corresponding picture element, or "pixel." The image will be assumed to be adequately reproduced by means of an \( n \times n \) array of square picture elements whose intensities are chosen from a range of \( L \) discrete values which are uniformly spaced over a selected brightness range. Obviously, both \( n \) and \( L \) must be large enough to give acceptable spatial and grey-tone resolutions for the task to be performed on the image.

Many of the operations carried out during pattern recognition or image processing can be described as "local;" this implies that the new value of a pixel in the processed image is a function of the values of a limited and compact subset of pixels in the original image. A particularly useful subset is the immediate neighbourhood of each pixel, i.e. the three by three array of adjacent pixels surrounding and including each pixel. The implication of this statement is that extremely fast processing of images could be achieved by constructing an array of identical processors in one to one relationship with all the pixels of an image, each processor receiving inputs from its corresponding pixel and from the eight immediate neighbours of the pixel. The purpose of the CLIP (Cellular Logic Image Processor) programme of research has been to optimize a processor array structure for the range of operations commonly required in image processing, having due regard to the cost of implementing the system as an array of large scale integrated circuits. At the same time, it was recognized that, ideally, the array should be capable of executing all possible functions provided an appropriate sequence of allowed instructions is obeyed. In other words, the array should perform as a general purpose computer optimized for a typical range of image processing operations.

CLIP4 LOGIC

The basic unit of the array is the cell, being a combination of logic units (i.e., the processor) and memory. Details of the CLIP4 cell are shown in Figure 1(a) in which A, B and C are single bits of buffer memory and D is a 32 by 1 bit RAM. The boolean processor provides two program selected independent functions, \( D \) and \( N \), of the two binary inputs A and P, one of which (N) is used to form an output to all neighbouring cells. Inputs from neighbouring cells are individually gated and then OR'ed together to form the binary function T which combines with B to produce P. A and B are each single bits of the pixels of two images in the array location corresponding with the cell concerned. Other gates are included which can be used to generate the exclusive OR (EXOR) of B and T at P and also the AND of B and T which is then OR'ed with the N output of the boolean processor to form the interconnection output N*. These additional gates are of use when the processor is performing arithmetic operations and are brought into use by special instructions which put 1's on the R and C control inputs.

Before considering the details of the operations in the CLIP4 cell, it will be helpful to examine the organisation of the data storage in the \( n \times n \) cell array (see Figure 1(b)). Each address in the D memories can be visualised as a plane of single bit stores, \( d \). If the location of each cell is given by its \((x,y)\) coordinate in the array, then a bit plane will be the set of bit stores defined by:

\[
D_y = \{ d_{x,y} : x, y = 1, 2, \ldots n \}
\]

A pixel will be formed as a column of single bit stores passing through \( g \) bit planes. Thus the pixel at location \((x,y)\) is defined by:

\[
P_{xy} = \{ d_{jxy} : j = k, k+1, \ldots k+g-1 \}
\]

\( D_k \) will contain all the least significant bits in the binary numbers representing the grey levels in each pixel; \( D_{k+g-1} \) will contain the most significant bits. Note also that

\[
g = \log_2 L
\]

implying \( g \) binary bit planes are needed to represent \( L \) grey levels. The ordering and addressing of the bit planes in the 32 available addresses in the \( D \) memories is arbitrary and may be varied to suit the needs of the program.

When numerical calculations are to be performed on data in which there is no longer an exact correspondence between the \( n^2 \) pixels and the \( n^2 \) processors, it is sometimes convenient to represent data in the binary column mode, in which \( n \) binary numbers are stored in the \( n \) columns of a
single bit plane. A binary number will be represented by the column:

\[ d_{xy} : \ y = 1, 2, \ldots, n \]

Normally, the precision provided by \( n \) bits will be unnecessarily high for typical values of \( n \). Methods have been devised for using the available storage more effectively.

A binary image is composed of elements which are either black or white so that pixels have values 0 or 1 respectively. Usually, binary images are produced by thresholding a grey tone image. Binary images are stored in a single bit plane of the D memories, at any convenient address.

**USING THE CLIP4 CELL**

The five principal ways in which the CLIP4 cell can be configured are illustrated in the simplified logic diagrams of Figure 2(a)-(e). In each case, the symbols A, B, C and D refer to bit planes.

**Simple boolean operations (Figure 2(a))**

Binary image \( I_1 \) is loaded into A and binary image \( I_2 \) loaded into B. The direction gates 1 to 8 and inputs R and C are disenabled. Any of the 16 possible boolean functions of the two images can be produced and loaded into a selected address of the D memories. If \( I_1 \) is in \( D_1 \), and \( I_2 \) in \( D_2 \), and if the result address is \( D_3 \), the coding for the operation of taking the exclusive OR of \( I_1 \) and \( I_2 \) is (in the CLIP4 language: CAP4):

```
SET P@ A
LDA 1
LDB 2
PST 3
```

(Notes: @ implies EXOR; PST is a mnemonic for Process and Store)
A binary image is loaded into A and an interconnection function set up (for example: \( N = A \)). A selection of interconnection gates are opened (for example: those in directions such that outputs from neighbouring cells to the North, East, South and West are admitted, i.e., directions 2, 4, 6 and 8) and a boolean function for the output \( D \) is chosen (say: \( P = \bar{A} \)). The coding is:

\[
\begin{align*}
\text{SET} & \quad P + - A, [2 4 6 8] A \\
\text{LDA} & \quad 1 \\
\text{PST} & \quad 3 
\end{align*}
\]

(Notes: the negative sign implies NOT; the B pattern does not affect the operation).

**Labelled propagating operations (Figure 2(c))**

In this case propagation can be started from any cell in which B has the value 1, i.e., from any white pixel in \( I_2 \). Using the same values as in the previous example, the coding becomes

\[
\begin{align*}
\text{SET} & \quad P + - A, [2 4 6 8] A \\
\text{LDA} & \quad 1 \\
\text{LDB} & \quad 2 \\
\text{PST} & \quad 3 
\end{align*}
\]

(Notes: the connection to the B store is enabled by inserting the symbol B at the end of the direction list on the SET instruction).

Propagation can also be initiated at the edges of the array by adding an E at the end of the SET instruction. This has the effect of supplying a 1 to all interconnection inputs of array edge cells which cannot connect to missing neighbours lying outside the limits of the array.

**Bit plane arithmetic operations (Figure 2(d))**

If A is a particular digit of one binary number, B the corresponding digit of another binary number, and C the carry digit resulting from the addition of the previous digits of less significance, then the expressions for the sum digit and new carry digit are:

\[
\begin{align*}
\text{SUM} & = A \oplus B \oplus C \\
\text{CARRY} & = (B \oplus C) \lor B \land C
\end{align*}
\]
When the inputs R and C are enabled, the CLIP4 cell will implement this arithmetic function for bit planes loaded into A and B, using the boolean processor to provide the operations N=P.A and D=P.EXOR.A. If two grey tone images are loaded into D₁ to D₄ and D₅ to D₁₂ respectively, then the part of the coding which adds the second least significant digits is:

```
SET P@A, [B] P.A, RC
LDA 2
LDB 8
PST 14
```

Giving a summed image in D₁₃ to D₁₉.

**Binary column arithmetic operations (Figure 2(e))**

Two planes of binary column numbers in D₁ and D₄ can be summed to form a new plane of binary column numbers in D₃ by writing the following short program:

```
SET P@A, [6B] P.A,R
LDA 1
LDB 2
PST 3
```

(Notes: the carry function is automatic when R is enabled; no use is made of the C plane stores; direction gate 6 is opened to allow carries to be propagated along the binary columns).

Other arithmetic functions can be developed from the summing operation in the usual way. In principle, floating point arithmetic can also be performed, although full details of the algorithms involved have still to be worked out.

**OTHER HARDWARE FEATURES**

In the CLIP4 system now being constructed, a 96 by 96 cell array is being assembled using custom-designed NMOS integrated circuits. Each circuit comprises eight cells arranged in two rows of four. The A elements are serially connected to form 96 shift registers, each of length 96 and lying along the array rows. Single bit planes of data are loaded serially into a 9216 bit shift register external to the array; this shift register can then be reconfigured into 96 short shift registers which then discharge their data into the A rows. The whole system interfaces to a television camera and monitor via two shift register memories, both of which can store a 96 by 96, six bit grey tone image. Analogue to digital conversion and digital to analogue conversion between these memories and the television camera and monitor respectively proceeds at the full video rate.

The array is operated by a controller which extracts the 16 or 32 bit instruction words from the controller memory, decodes them and drives the array and peripherals. Included in the controller are 14 general purpose 16-bit registers. A fifteenth register has a special additional function: a bit plane may be shifted from the array through a tree of parallel adders, thus permitting a rapid count of the 1-bits set in the A stores. This count appears in register 15. Furthermore, another instruction can be used to output the contents of...
register 15 back into the array by means of an operation similar to the E instruction.

CLIP4 INSTRUCTION TIMES

CLIP4 is a Single Instruction, Multiple Data stream computer and performs image processing tasks at speeds which are quite unobtainable in conventional machines. The integrated circuit operates with a four phase, 400 ns clock cycle; load instructions require 8 cycles, PST instructions 10 cycles and SET instructions 12 cycles. A further 3 cycles are required for each propagation step through a cell when the selected boolean function is such as to cause propagation. Branch and register instruction require 4 or 8 cycles (the longer time being for memory reference register instructions).

Interleaving of these instructions is used to reduce complete array operations to just under 9 μs plus 1.2 μs/cell for propagation beyond the immediate neighbourhood of each

Figure 2(d)—Bit plan arithmetic logic

Figure 2(e)—Binary column arithmetic logic
cell. In general, the complete operation will comprise the instructions SET, LDA, LDB and PST. A bit plane of data can be entered into the array in under 4 ms. However, this may increase up to 10 ms since the operation is synchronised with the television scan cycle.

SYSTEM STATUS

A small scale pilot model of the system (CLIP3) has been in operation since 1973. The array comprises 16 by 12 cells and can be used on images of the same size or, by scanning the array across a 96 by 96 pixel image, can be used to simulate the larger CLIP4 array. Additional hardware and software is incorporated to handle sector to sector propagation when CLIP3 is used in the scanned mode. Details of the CLIP3 system are provided in References 1-5, which also include bibliographies of earlier studies both at University College and elsewhere.

At the time of writing (late January 1978), the CLIP4 chip design is complete and prototype chips are expected to be delivered before the end of March 1978. All other parts of the hardware and software for the image processing system are complete, although some modifications and improvements are being carried out. Negotiations are in hand to arrange for CLIP4 systems to be manufactured and made commercially available. Preliminary estimates suggest that the first systems will be ready during mid to late 1979. The University College prototype is scheduled for operation in Autumn 1978.

REFERENCES