amplifier, bit rate 450 Mbit/s and bit error rate $P_e = 10^{-9}$. Fig. 3 indicates that sensitivity improvement can be obtained by making the detectable area small, for example 80 μm to 30 μm diameter, 1.3 dB at 25°C and 2.4 dB at 45°C.

Measurement: Measured receiver sensitivity temperature dependences are shown in Figs. 4a and b. The Ge APD used is assembled in module with a single-mode optical fibre pigtai and a SELFOC® lens. The wavelength used is 1.525 μm and the bit rate is 450 Mbit/s (mark density is 1/2 and duty factor is 1/2). The preamplifier used is composed of bipolar transistors. Sensitivities achieved for the 30 μm-diameter device and for the 80 μm-diameter device at $P_e = 10^{-9}$ is $-39.4$ dBm and $-38.8$ dBm at 5°C, $-39.4$ dBm and $-38.3$ dBm at 25°C, $-39.9$ dBm and $-35.5$ dBm at 45°C, respectively. In the temperature elevation from 25°C to 45°C, the power penalty is 1.5 dB at the 30 μm-diameter device and 2.8 dB at the 80 μm-diameter device. The small detectable area device has less temperature dependence. The measurement results indicate that an obtained sensitivity improvement, caused by making the detectable area small, is 1.1 dB at 25°C and 2.4 dB at 45°C. These values almost coincide with the theoretical values in Fig. 3. In conclusion, a small detectable area (30 μm diameter) Hi-Lo Ge APD has a good temperature characteristic in addition to high sensitivity compared with an 80 μm-diameter device. This reflects the multiplied dark current improvement.

**Image Data Compression using 2-D Lattice Predictor**

**Indexing terms:** Image processing, 2-D lattice predictor

The outline of an approach for image data compression using a 2-D lattice predictor is presented. Preliminary results indicate that acceptable quality images (quantised to 15 levels) can be transmitted at a lower bit rate, for example 80 μm to 30 μm diameter, 1.3 to 2.4 dB at 45°C.

**Introduction:** The principle of a predictive method of image data compression is to exploit redundancy in the data and to represent the data by a new set of data with reduced redundancy such that they can be transmitted at a lower bit rate. From this new set of data, the original image data can be reconstructed with acceptable distortion.

In this contribution, we present the outline of an approach for image data compression using a 2-D lattice predictor. The 2-D lattice predictor employed has the advantages of guaranteed stability, fewer modelling parameters and reduced computation and implementation complexity as compared to the use of differential pulse code modulation (DPCM).

**Compression system:** The block diagram of the image data compression system is shown in Fig. 1. The 2-D lattice predictor is obtained by using the 2-D lattice modelling in the quarter plane case as described in Reference 1. The predictor exploits redundancy in the image data and represents them by uncorrelated parameters. At each lattice stage, three reflection coefficients (which minimise the expected mean value of the sum of the four squared prediction errors) and the resultant four prediction errors are computed. In practice, only the reflection coefficients and the final stage first prediction errors are stored.

The purpose of the nonuniform quantiser is to quantise the final stage first prediction errors to levels lower than the grey levels of the original image so as to compress the data. Experimental results show that the first prediction errors are characterised by a probability density function (PDF) which is of near-zero mean and small variance. Therefore, the least mean squared error nonuniform quantiser of B. Smith (eqns. 4 and 5 of Reference 2) is used in which the errors at the tail area of the PDF are quantised coarsely while the midrange errors are quantised finely.

At the output of the nonuniform quantiser a variable-length binary encoder is used. This is because the probabilities of occurrence of the various output levels of the quantiser are

---

**References**

not the same; the levels which occur more often can be assigned shorter code words. The type of variable length code employed is the Huffman code.\(^3\)

\[
\begin{align*}
& K_1 \times K_2 \text{ image data from the compressed data of the final } N\text{-stage } \\
& \text{first prediction errors, } e_{00}^{(i)}(k_1, k_2) \text{ for } k_1 = 1 \text{ to } K_1, k_2 = 1 \text{ to } K_2. \text{ The original image data are reconstructed from } \\
y(k_1, k_2) = - \sum_{i=0}^{N} \sum_{j=0}^{N} b_{00}^{(i,j)}(i, j)y(k_1 - i, k_2 - j) \\
+ e_{00}^{(i)}(k_1, k_2)
\end{align*}
\]

where the transfer function coefficients \( b_{00}^{(i,j)}(i, j) \) are computed from the reflection coefficients.

**Results:** Experiments were carried out on an image of dimensions 256 \( \times \) 256 and quantised to 6 bits per pixel. For each of the lattice stages (i.e. 1, 3, 5), with the first prediction error quantised to 15 levels, the information rate (IR) of the Huffman encoded data per encoded word, and the signal/noise ratio (SNR) of acceptable quality images are summarised in Table 1. (The BR includes the transmission of all 32-bit reflection coefficients, a match table, and then the Huffman-encoded first prediction errors. The match table consists of 15 \( \times \) 15 entries of 26 bits; each entry consists of four bits each for the first and second quantisation levels, and four bits and 14 bits, respectively, for the length and binary pattern of a Huffman encoded word.)

**Table 1**

<table>
<thead>
<tr>
<th>Lattice stage</th>
<th>1</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR, bpp</td>
<td>1-16</td>
<td>1-19</td>
<td>1-38</td>
</tr>
<tr>
<td>BR, bpp</td>
<td>1-19</td>
<td>1-33</td>
<td>1-40</td>
</tr>
<tr>
<td>SNR, dB</td>
<td>20-6</td>
<td>21-6</td>
<td>22-5</td>
</tr>
</tbody>
</table>

**Conclusions:** The outline of the use of a 2-D lattice predictor for image data compression has been described. The feasibility of the approach has been demonstrated by some results. The purpose of the 2-D inverse predictor is to reconstruct the original \( K_1 \times K_2 \) image data from the compressed data of the final \( N \)-stage first prediction errors, \( e_{00}^{(i)}(k_1, k_2) \) for \( k_1 = 1 \) to \( K_1, k_2 = 1 \) to \( K_2 \). The original image data are reconstructed from

\[
y(k_1, k_2) = - \sum_{i=0}^{N} \sum_{j=0}^{N} b_{00}^{(i,j)}(i, j)y(k_1 - i, k_2 - j) + e_{00}^{(i)}(k_1, k_2)
\]

Satisfactory results have been obtained for image data compression having been described. The feasibility of the approach has been demonstrated by some results.

**SYSTOLIC MULTIPLIER**

Indexing term: Logic and logic design

A systolic multiplier based on the shift-and-add algorithm is presented. With this array the minimum possible number of cells is obtained.

McCanny and McWhirter\(^1,2\) have demonstrated how to factorise the multiplication problem in such a way that the design of a complete circuit is reduced to that of a single cell. They have applied the concept of systolic arrays\(^3\) to previously proposed pipelined Carry-Save multipliers.\(^4\)

This letter presents a similar systolic-multiplier array, but based on the shift- and-add algorithm.\(^4\) It can be seen as the 'systolisation' of the pipelined ripple-carry multiplication array.\(^5\)

**Fig. 1 Systolic multiplier array**

Fig. 1 depicts the circuit for the pipelined multiplication of two continuous streams of 4 bit numbers. The elementary cell of the lattice is the so-called inner-product step processor which performs the following operations on the binary information:

- (i) copy \( a \) and \( b \)
- (ii) transfer \( a \) and \( b \) (delay)
- (iii) multiply \( a \) and \( b \) to obtain product \( P \)
- (iv) add incoming sum \( S \), product \( P \) and incoming carry \( C \) to obtain outgoing sum \( S \) and outgoing carry \( C \).

All input and output signals are fully synchronised, obtained by skewing. The arrows represent shifts at discrete time intervals. The rectangles represent delays which enable multiplications to be carried out in a completely pipelined and parallel way, which is the fundamental reason why very high throughputs are feasible.

The numbers to be multiplied are fed in parallel from the left side and shifted through the array. The carries are shifted parallel with the \( a \). At the bottom the sums \( S(0), S(1) \) etc. are obtained. The products \( a(0) \cdot b(0) \) and \( a(1) \cdot b(1) \) are calculated at the crossings of bit streams \( a \) and \( b \). These products are obtained in the following time order:

(i) first \( a(0) \cdot b(0) \)
(ii) then \( a(0) \cdot b(1) \)
(iii) next, at the same time, \( a(0) \cdot b(2) \) and \( a(1) \cdot b(0) \), and so on.

**References**

2. **HABIBI, A.:** 'Comparison of nth-order DPCM encoder with linear transformations and block quantization techniques', *ibid.*, 1971, COM-19, pp. 948-956
4. **McCanny, J., and Mc WHIRTER, J. G.:** 'A systolic multiplier'.
5. **Pokfulam Road, Hong Kong**