tional assumptions concerning the reciprocity or symmetry of the error network are made, or if at least one of the leakage paths can be neglected. It may well be that such an assumption has been automatically carried out by the SVD algorithm. The authors even mention that a direct solution did not give satisfactory results, probably due to (two) singularities.

Calibration standards: The most common calibration standards used with the network analyser measurements are: thru (T), delay (D), match (M), short (S) and open (O). Five two-port calibration standards could be achieved taking pairs M-M, S-S and O-O in addition to the thru and the delay line (M-M means matched loads simultaneously at port one and at port two). This combination of standards unfortunately does not allow the determination of the error parameters. Several other combinations are possible.

We first assume that one of the standards is the thru and four standards are chosen from the pairs consisting M, S or O: M-M, S-S, O-O, S-O, O-S, S-M, M-S, O-M, M-O. There are 126 such possibilities of which 46 can be shown to be singular.

If we take the thru and the delay line and choose only three standards from the preceeding list, we obtain 84 different combinations. Only eight of them produce a singular combination. To save space only the nonsingular combinations are listed in Table 1; i.e. these combinations of standards can be used for calibration of the 16-term error model. Because the short and the open represent dual cases, they are handled together denoting A = short, B = open or vice versa.

If combination T, A-A, B-B, A-B, M-A is nonsingular, then of course, T, A-A, B-B, B-A, A-M is also a nonsingular combination. These cases have not been repeated in Table 1. If offset shorts or opens are used, some of the otherwise singular combinations could be usable, but there will exist the same kind of bandwidth problems as with, for example, the LRL method. Also the delay lines may cause a bandwidth limitation. There must always be an impedance reference: a load or a transmission line. The line may exist in a form of a delay line or in a one-port standard (an offset short or an offset open circuit). D, A-A, B-B, A-B, B-A is an allowed combination, while this combination does not give a solution, if delay D is replaced by a zero-length thru (T).

Table 1 NONSINGULAR COMBINATIONS OF FIVE TWO-PORT CALIBRATION STANDARDS IN CONJUNCTION WITH 16-TERM ERROR MODEL

T	A-M	M-A	A-A	B-B	T	M-M	A-A	B-B	A-B
T	A-M	B-A	A-A	B-B	T	M-M	A-A	B-B	A-M
T	A-M	B-M	A-A	B-B	T	M-M	A-A	A-B	B-A
T	A-M	M-A	B-M	A-A	T	M-M	A-A	A-B	M-A
T	A-M	M-B	B-M	A-A	T	M-M	A-A	A-B	M-B
T	A-M	B-A	B-M	A-A	T	M-M	A-A	A-M	M-A
T	A-M	B-A	M-B	A-A	T	M-M	A-A	A-M	M-B
T	A-M	B-A	A-B	B-B	T	M-M	A-A	B-M	M-B
T	A-M	M-A	A-B	B-B	T	M-M	A-B	A-M	B-A
T	A-M	B-M	A-B	B-B	T	M-M	A-B	A-M	M-A
T	A-M	M-B	B-A	B-B	T	M-M	A-B	A-M	M-B
T	A-M	M-B	B-M	B-B	T	M-M	A-B	B-M	M-A
T	A-M	M-A	B-M	B-B	T	D	A-A	A-B	M-M
T	A-M	M-A	B-M	A-B	T	D	A-A	A-M	M-M
T	A-M	M-A	B-M	B-A	T	D	A-A	B-M	M-M
T	A-M	M-B	A-B	B-A	T	D	A-B	A-M	M-M
T	A-M	M-B	B-M	A-B	T	D	A-B	M-A	M-M
T	A-M	M-B	B-M	B-A	T	D	A-M	M-A	M-M
T	A-M	B-M	A-B	B-A	T	D	A-M	M-B	M-M
					T	D	A-A	A-B	B-M
					T	D	A-A	A-M	B-M
					T	D	A-B	A-M	B-M
					T	D	A-M	M-A	B-M

T = thru; M-M = match-match can be replaced by D = delay;
A = short, B = open or A = open, B = short

Equation set: The equations are chosen according to the standards in use. One possibility is to choose eqns. 1–4 for the first and the second standard, eqns. 1–3 for the third standard and eqns. 2 and 3 for the two remaining standards. This allows a solution for every combination listed in Table 1.

If error term t_{12} [1] is set equal to 1, t_{15} will correspond to the parameter k in Reference 3. The same sets of equations can also be used, if the leakage is zero. The 8-term model can thus be handled as a special case of the 16-term error model with the same algorithm. If the singular value decomposition (SVD) or other least squares techniques are used, the choice of equation is not important. On the other hand, the unused equations give several possibilities for selfcalibration.

The usefulness of the different combinations is dependent on the available calibration standards. They may also possess different error sensitivities, when used with different kinds of measuring system.

Some very interesting sets of standards consist only of one matched load, one short circuit and one open circuit in addition to the thru connection: e.g. T, S-M, O-M, S-O, O-S. The method can be considered a 16-parameter application of the old SOLT calibration method.

16th June 1993

K. J. Silvonen (*Helsinki University of Technology, Otakaari 5A SF-02150 Espoo, Finland*)

References

1 BUTLER, J. V., RYTTING, D. K., ISKANDER, M. F., POLLARD, R. D., and VAN DEN BOSSCHE, M.: '16-term error model and calibration procedure for on-wafer network analysis measurements', *IEEE Trans.*, 1991, **MTT-39**, pp. 2211–2217

2 SPECIALE, R. A.: 'A generalization of the TSD network-analyzer calibration procedure, covering n-port scattering-parameter measurements, affected by leakage errors', *IEEE Trans.*, 1977, **MTT-25**, pp. 1100–1115

3 SILVONEN, K. J.: 'A general approach to network analyzer calibration', *IEEE Trans.*, 1992, **MTT-40**, pp. 754–759

SEQUENTIAL IMAGE CODING BASED ON MULTIRESOLUTION TREE ARCHITECTURE

J. Li and X. Lin

Indexing terms: Image coding, Motion estimation

A new image sequence coding scheme based on the multiresolution tree structure is proposed. It not only guarantees reliable and homogeneous motion field, but also reduces the amount of computation necessary in the motion vector search. With nearly the same decoded image quality, 25–30% bit rate reduction can be obtained compared with the MPEG-I algorithm.

Introduction: Fixed block size motion compensation is widely used in current image sequence coding techniques, e.g. MPEG-I [1] and H.261; it splits the image frame into fixed size blocks. Motion estimation is performed for every block separately. This scheme is quite simple, but has some inherent limitations:

(i) Fixed size block cannot adapt itself for complex scenery. The block size is often too small for large background areas and too large for moving object boundaries.

(ii) The homogeneous characteristic of the motion field is not used in fixed block size motion estimation. Therefore the calculation can easily fall into a local minimum and obtain a false vector.

(iii) The fixed size scheme only estimates the motion field in one single resolution. The amount of computation needed in the estimation procedure increases rapidly when the search region becomes large.

Research has already been carried out to solve the above problems. Chan [2] introduced a bin-tree structure to organise and estimate the motion field. He did not consider in any detail the homogeneous characteristic of the motion field, so the algorithm is slow. Bierling [3] has also introduced a famous hierarchical block matching motion estimation scheme. A reliable and accurate motion field is obtained during computation. However, he does not try to organise the motion field and the MCP (motion compensated prediction) error, nor does he consider the computation speed and motion field transmission.

Algorithm: To solve the inherent limitation in fixed block size motion estimation, a multiresolution tree structure is proposed. The multiresolution tree structure is used to estimate the motion field, to identify the moving area where the frame difference signal should be coded, and to transmit the motion field and MCP error. It can adapt the spatial resolution of the local motion field according to the image scenery. The image is split into blocks with variable block size at different resolutions. A motion vector and an MCP error coding scheme is associated with each block. The construction of the multiresolution tree structure runs top–down as follows:

(i) *Step 1: A multiresolution tree is built for the images:* The current image frame f and the previous image frame g are lowpass filtered and subsampled to generate a series of different resolution images $f^0, f^1, \ldots, f^{level}$, and $g^0, g^1, \ldots, g^{level}$, that is, two image pyramids.

(ii) *Step 2:* The lowest resolution image f^{level} is split into fixed size blocks. Each block is processed in step (iii).

(iii) *Step 3:* The block is processed at level i. This can be further divided into several substeps.

(*a*) *Motion detection:* The frame difference signal is tested to determine if it is small enough for the block to be considered as a static block.

(*b*) *Motion estimation:* A 2-D logarithmic search [4] or exhaustive search is carried out for the block. The motion vector of the parent block serves as a prediction in the estimation procedure. The search is only performed over a small range $(-M_i, M_i) \times (-M_i, M_i)$ around the prediction vector.

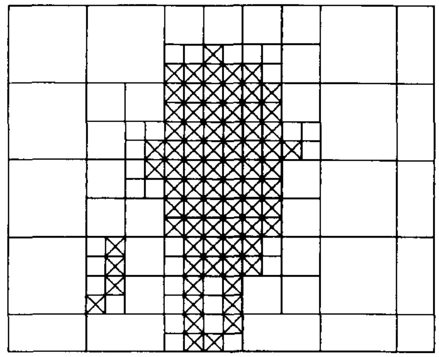
(*c*) The motion vector is prediction coded.

(*d*) The decision procedure is terminated.

The MCP error is tested to determine if it is small enough. If the result is positive, the coding procedure is terminated. Otherwise the encoding procedure runs to step (iii)*e*.

(*e*) Split the block or code of the remaining MCP error is split. If the current level i it is not 0, the encoding procedure runs down to level $i-1$, to a higher resolution level. The block is split into four high resolution sub-blocks. Each sub-block is processed again through step (iii). Otherwise a special MCP error coder (a DCT codec with macro-block structure, just the same as that in the MPEG–I standard) is called to transmit the remaining MCP error.

An example of the multiresolution tree structure is shown in Fig. 1. Only the MCP error in block X needs coding.



502/1

Fig. 1 *Sample multiresolution tree structure for 'Claire' sequence*

In the multiresolution tree structure, the spatial resolution for motion estimation and motion vector transmission increases in a top–down manner. In the higher level, the image resolution is low. A small block stands for quite a large area, and a small vector also represents a large displacement. Therefore large motion can be estimated with little computation, and transmitted with fewer bits. Because at a high level, a relatively large area is incorporated in motion estimation, a reliable and homogeneous motion field can be esnured. At a lower level, the image resolution is increased. However, motion estimation is restricted to a small region around the prediction vector. The computation burden is small, and the estimated motion field is still homogeneous. The number of bits used to transmit the difference vector is also quite small because the maximum difference is restricted by the search region.

Result and discussion: Several experiments have been carried out on the 'Claire' and 'Miss America' standard image sequences. The data is in CIF format, which has a resolution of 352×288 pixels per frame, and 10 frames per second.

First, the speed and efficiency for motion estimation are compared. The search range for motion estimation is 16. A three level multiresolution tree structure is used for comparison; the search range for each level is $M_0 = 2$, $M_1 = 4$, $M_2 = 4$. The results are shown in Table 1. Note that because a high level block carries many low level blocks, the calculation carried out at a high level is divided by the subsample factor.

The prediction gain in Table 1 is the difference between the motion compensated image f' and the current image f; the value is represented by the peak signal to noise ratio:

$$PSNR = 10 \log \frac{255^2}{E\{(f'_{i,j} - f_{i,j})^2\}} \tag{1}$$

From Table 1, we can see that the prediction gain of the multiresolution tree structured motion estimation is nearly the

Table 1 COMPARISON BETWEEN DIFFERENT MOTION ESTIMATION METHODS

Motion estimation method	Search point	Prediction gain for 'Claire'	Prediction gain for 'Miss America'
		dB	dB
Fixed block size, exhaustive search	963	41·14	39·06
Fixed block size, 2-D logarithm search	33	39·34	37·22
Multiresolution, exhaustive search	24·3	40·90	38·50
Multiresolution, 2-D logarithm search	14·3	40·80	38·34

Table 2 COMPARISON BETWEEN DIFFERENT ENCODING METHOD

Coding method	Results for 'Claire' Bit rate for motion field	Total bit rate	PSNR	Results for 'Miss America' Bit rate for motion field	Total bit rate	PSNR
	bit/pixel	bit/pixel	dB	bit/pixel	bit/pixel	dB
MPEG-I	0·0108	0·0463	42·03	0·0279	0·0808	38·56
Multiresolution	0·0069	0·0345	42·25	0·0157	0·0582	38·71

same as the fixed block size scheme with exhaustive search, and much higher than the traditional fast motion estimation scheme, the 2-D logarithm search. The speed of the multiresolution scheme is even faster than for the 2-D logarithm search. The use of exhaustive search or logarithm search in the multiresolution tree structure makes little difference.

The MPEG-I image sequence coding standard was then compared with our multiresolution tree structured codec. The fixed block size 2-D logarithm search is used in the MPEG-I standard. The results are shown in Table 2.

The effect of the multiresolution tree structure is quite obvious. The total bit rate for image sequence coding and the bit rate used in motion field transmission are both reduced. With nearly the same decoded image quality, a 25–30% reduction in bit rate can be obtained.

Conclusion: A sequential image coding scheme based on the multiresolution tree architecture is proposed. The multiresolution tree structure is used to estimate the motion vector field, to identify the moving area where the frame difference signal should be coded, and to transmit the motion field and MCP error. A homogeneous and reliable motion field is obtained with very little computation, and the coding bit rate is also reduced by ~25–30% compared with the MPEG-I standards.

24th June 1993

J. Li and X. Lin (*Image Processing Division, Electronic Engineering Department, Tsinghua University, Beijing, 100084, People's Republic of China*)

References

1 ISO/IEC JTC1/SC29/WG11: 'Coding of moving pictures and associated audio for digital storage media at up to about 1·5 Mbit/s'. ISO CD 11172, November 1992

2 CHAN, M. H., SU, Y. B., and CONSTANTINIDES, A. G.: 'Variable size block matching motion compensation with application to video coding', *IEE Proc. I*, 1990, **137**, (4), pp. 205–212

3 BIERLING, M.: 'Displacement estimation by hierarchical block matching'. Visual Communication and Image Processing '88, Cambridge, Massachusetts, 9th–11th November 1988, pp. 942–951

4 KOGA, T., IINUMA, K., HIRANO, A., IIJIMA, Y., and ISHIGURO, T.: 'Motion-compensated interframe coding for video conference'. Proc. of NTC 81, New Orleans, LA, December 1991, pp. G.5.3.1–G.5.3.5

SYSTOLIC COMPUTATION OF THE RUNNING MIN AND MAX

A. R. Butz

Indexing terms: Filtering and prediction theory, Systolic arrays

A method of computing the running min and max, requiring no fan-in and hence compatible with systolic implementation, is proposed. The time period complexity for the computation of the running min and max, or for more general rank order functions expressible in terms of min and max functions, is O(b), where b is the word length.

Introduction: Proposed implementations of median and related filters sometimes call for computation of the running min and/or max of input values. The modified recursive median filter [1, 2], which has the same input–output relationship as the standard recursive median filter [3], does not compute the median; rather for effective odd length L it computes both the min and the max of the most recent $l = (L + 1)/2$ inputs and then combines the two results to form an output value. Thus Ko *et al.* [4] proposed an implementation of the modified recursive median filter that employed standard fan-in computations to compute the min and max. Fitch [5] suggested the direct computation of a general running rank order function as a max of min values, also by a method requiring fan-in computations. Because fan-in is undesirable in terms of systolic implementation in VLSI, the purpose of this Letter is to show how the running min and max may be computed in a one dimensional array in which inputs enter only the first stage or cell of the array, and outputs exit only from the same stage or cell, with communication between only neighbouring cells. The time period complexity is O(b), where b is the number of bits in the fixed point data representation. This time period complexity holds not only for the computation of individual running minima and maxima, but also (with a qualification) for the more elaborate functions, e.g. running medians and modified recursive median filters, that may be built up from them.

Algorithm: The basic algorithm appeared in Reference 6 in the context of more general issues; it is best defined by example. Suppose the input sequence is $\{u(n)\}$ and the output sequence is $\{y(n)\}$, where it is required ($\wedge$ stands for min) that

$$\begin{aligned} y(n) = u(n) \wedge u(n-1) \wedge u(n-2) \wedge u(n-3) \\ \wedge u(n-4) \wedge u(n-5) \wedge u(n-6) \end{aligned} \quad (1)$$

Define a sequence of state vectors $\{x(n)\}$, where $x(n) = [x_0(n), x_1(n), x_2(n), x_3(n), x_4(n), x_5(n), x_6(n)]$; the output value $y(n) = x_0(n)$. Assuming that $x(n-1)$ has been computed at time $n-1$, the algorithm requires that $x_0(n)$, $x_2(n)$, $x_4(n)$, $x_6(n)$ be computed in time interval or half cycle $[n-1, n-1/2]$, and $x_1(n)$, $x_3(n)$, $x_5(n)$ be computed in time interval or half cycle $[n-1/2, n]$, thus:

$$\begin{aligned} &x_0(n) := u(n) \wedge x_1(n-1) && x_1(n) := u(n) \wedge x_2(n) \\ &x_2(n) := u(n-1) \wedge x_3(n-1) && x_3(n) := u(n-1) \wedge x_4(n) \\ &x_4(n) := u(n-2) \wedge x_5(n-1) && x_5(n) := u(n-2) \wedge x_6(n) \\ &x_6(n) := u(n-3) && \end{aligned} \quad (2)$$

It is easily confirmed that eqn. 2 implies eqn. 1. If rather than eqn. 1 it is desired to compute

$$y(n) = u(n) \wedge u(n-1) \wedge u(n-2) \wedge u(n-5) \wedge u(n-6) \quad (3)$$

then in eqn. 2 the references to $u(n-1)$ and $u(n-2)$ are removed in the computations of $x_3(n)$ and $x_4(n)$, respectively ($x_3(n) := x_4(n)$ and $x_4(n) := x_5(n-1)$). The change necessary to compute running maxima, rather than running minima, is obvious.

Implementation: The implementation of eqn. 2 could be as a linear array processing seven stages or cells that compute the seven $x_i(n)$ in place, with communication only between adjacent cells. The input sequence $\{u(n)\}$ enters at the first $[x_0(n)]$ cell and the output sequence $\{y(n)\}$ is made available at the same cell. Thus there is no fan-in.

Each min or max of two values as in eqn. 2 can be computed in a bit serial, most significant bit leading, manner. If the number of bits is b, and the state vector has dimension N ($N = 7$ for eqn. 1 or 3), then the time delay complexity is O(Nb). Because a new output value is made available in every