*Results:* As an example, Fig. 2 presents the dependence of the amplitude of the attenuation factor on the dimensionless parameter $\rho = z(n + 1)^{3/2} = -2\pi^{1/2}(D/\lambda)^{1/2}D/R$. The negative values of the parameter $\rho$ correspond to the convex surface, and the positive values correspond to the concave surface. In this example the source has zero height. The results of computations for the constant-curvature surface match well with the data for a parabolic hill with a base of 203m, peak height of 5m and distance between half-screens of 29m [6, 7], obtained by the UTD and the parabolic equation method for a wavelength of 0.33m (frequency 900MHz). It is important that with a large number of obstacles, the field strength variation with the growth of $n$ in practice does not depend on the surface curvature and the path length. The obtained results enable the dependence of the loss on the building density and on the path profile of the terrain to be analysed separately.

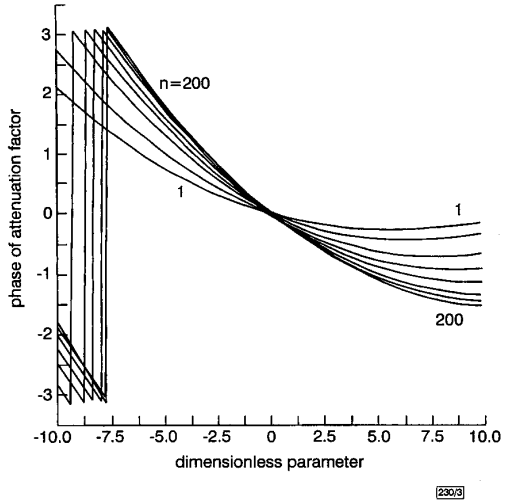


**Fig. 3** *Phase against* $\rho$ *and* $n$

Fig. 3 presents the dependence of the attenuation factor phase over length $[-\pi, \pi)$. At $\rho = 0$ the phase is equal to zero, irrespective of $n$. Consequently, for a flat surface where the source and the observation point have zero heights, the attenuation factor becomes real, which coincides with the solution [8] for the array of parallel half-planes bounded by a flat surface.

*Conclusion:* The developed model not only enables the speed of calculation to be increased for the prediction of the field strength in an urban environment, but also leads to a deeper understanding of the physical phenomenon of radiowave diffraction in a series of absorbing half-screens, the tops of which form a surface with a smoothly varying curvature. For the first time it has been analytically established that when the waves propagate at small glancing angles, the diffraction loss depends on the curvature of the surface, not on the height.


*7 September 1999*
*Electronics Letters Online No: 19991307*
*DOI: 10.1049/el:19991307*

A.S. Nastachenko (*Institute of Physics, Stachky 194, Rostov-on-Don, 344090, Russia*)

E-mail: nas@iphys.rnd.runnet.ru

## References

1 VOGLER, L.E.: 'An attenuation function for multiple knife-edge diffraction', *Radio Sci.*, 1982, **17**, (6), pp. 1541–1546

2 PIAZZI, L., and BERTONI, H.L.: 'Effect of terrain on path loss in urban environments for wireless applications', *IEEE Trans.*, 1998, **AP-46**, (8), pp. 1138–1147

3 TZARAS, C., and SAUNDERS, S.R.: 'Rapid, uniform computation of multiple knife-edge diffraction', *Electron. Lett.*, 1999, **35**, (3), pp. 237–239

4 BOERSMA, J.: 'On certain multiple integrals occurring in a waveguide scattering problem', *SIAM J. Math. Anal.*, 1978, **9**, (2), pp. 377–393

5 XIA, H.H., and BERTONI, H.L.: 'Diffraction of cylindrical and plane waves by an array of absorbing half-screens', *IEEE Trans.*, 1992, **AP-40**, (2), pp. 170–177

6 ANDERSEN, J.B.: 'UTD multiple-edge transition zone diffraction', *IEEE Trans.*, 1997, **AP-45**, (7), pp. 1093–1097

7 JANASWAMY, R., and ANDERSEN, J.B.: 'Path loss predictions in urban areas sprawling over irregular terrain'. Proc. IEEE Symp. PIMRC, Boston, MA, 8–11 September 1998, pp. 874–878

8 LEE, S.W.: 'Path integrals for solving some electromagnetic edge diffraction problems', *J. Math. Phys.*, 1978, **19**, (6), pp. 1414–1422

# Array address translation for SDRAM-based video processing applications

Hansoo Kim and In-Cheol Park

To increase the memory bandwidth of SDRAM (synchronous DRAM) that is commonly employed as external memory in video applications such as MPEG2, a memory address translation method is proposed for minimising the number of overhead cycles needed for row-activations and precharges. The features of SDRAM and the characteristics of memory accesses in video processing applications are considered to find a suitable address translation method. Experimental results show that the proposed method increases the memory bandwidth by 44% over that possible using conventional linear translation.

*Introduction:* As the resolution of video processing applications increases, video signal processors need to cope with an increasingly large amount of data within a tightly bounded timeframe. The huge amount of video data are generally stored in off-chip memories that are usually slow. As the system performance strongly depends on the memory bandwidth between the signal processor and the external memory, newer DRAM families such as synchronous DRAM (SDRAM) are being widely used. To obtain high performance, SDRAM has two key features: the burst access mode and multiple bank architecture [1]. The burst access mode makes it possible to access a number of data by changing only column addresses. The multiple bank architecture of SDRAM can be used to hide memory cycles needed for row-activations and precharges by accessing different banks alternatively. Since the number of additional cycles needed for row-changes is usually considerable, we have to reduce or hide these cycles by effectively using the SDRAM features.

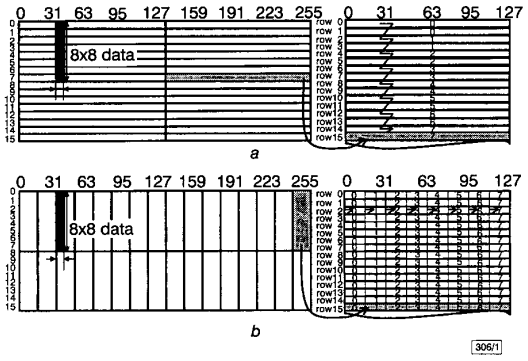


**Fig. 1** *Row-changes for two physical memory maps*

*a* Logical array with (1, 128)-windows, and physical memory map
*b* Logical array with (8, 16)-windows, and physical memory map

The authors of [2] showed that the order of memory accesses has a major effect on the total number of memory cycles and proposed pre-synthesis optimisation steps in which interleaved burst access is exploited. However, the work is limited to the reordering of memory accesses. If video data are stored in memory linearly in order of their logical addresses, the data accesses will usually

involve a large number of row-changes, which leads to a degradation of the performance. For example, consider a video processing algorithm in which video data are processed in 8 × 8 blocks, as shown in Fig. 1. If logical addresses are linearly mapped to physical addresses, seven row-changes are needed to access a block, as shown in Fig. 1*a*. To minimise row-changes, which degrade the system performance, we propose a translation of the logical addresses of multi-dimensional arrays into physical addresses as shown in Fig. 2. A logical array is partitioned into a set of rectangles called windows and each window is stored in a row of SDRAMs. The windows in Fig. 1*a* and *b* consist of 1 line × 128 data and 8 lines × 16 data, respectively. The address translation based on the (8, 16)-window requires no row-changes in accessing a block.

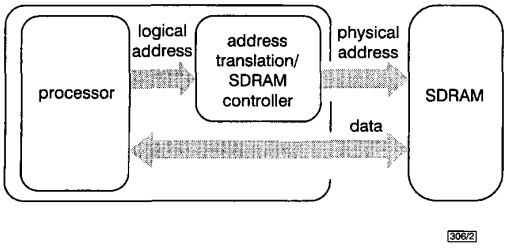


**Fig. 2** *Address translation method*

In this Letter, we propose an address translation method for increasing the memory bandwidth of SDRAM by reducing the number of overhead cycles and present an efficient algorithm for determining the correct window size.

*Proposed method:* Video data are usually represented by arrays and processed using nested loops. Bounds and strides of the loops as well as the dimensions and sizes of the arrays are statically determined. From the innermost loops, the number of data accessed sequentially can be determined. As the number of data accessed in an innermost loop is usually fixed, we call the memory access of the innermost loops 'block-type access'. The indices of outer loops determine the start address of a block-type access, but do not change the number of data accessed sequentially. Hence, the memory access patterns are determined by the array sizes, loop characteristics and indices of the outer loops. If we choose a window such that block-type data accessed in the innermost loops are stored in a row or in different banks block-type access does not induce any overhead cycles for row-activations. The conditions under which a window has no overhead cycles during a block-type access are as follows:

(i) All data for block-type access are contained in a window.

(ii) All data for block-type access are contained in windows that are contiguous horizontally and the horizontal size of windows is greater than the minimum number of data needed for hiding the row-change overhead cycles.

(iii) When the block-type access data are contained in several windows that are contiguous horizontally and vertically, there should be a sufficient number of data which are stored in other banks between the last line of the upper window and the first line of the lower window.

It is apparent that the first condition induces no row-changes. The window in Fig. 1*b* satisfies this condition and thus the access of an 8 × 8 block needs no row-changes. The second condition means that a row of the next contiguous window can be activated during the burst access of the preceding window, if the next contiguous windows are assigned to different banks. The last condition is the most general case. To determine the number of data in other banks between the accesses of two different rows in the same bank, we calculate two physical locations of the last data of the upper window and of the first data of the lower window.

To determine the best window, memory penalty simulations for all the candidate windows are required. Since repetitive simulations are very time-consuming, we present an efficient algorithm consisting of three steps.

The C description of an application and parameters of the SDRAM such as the number of banks and the size of one row are given as the inputs to the algorithm. First, through evaluating the above conditions as well as analysing the loops and arrays contained in the C description, the algorithm selects candidate windows which do not cause row-changes for any block-type access of the innermost loops. Secondly, a simple penalty simulation is performed for the candidate windows selected in the first step to consider the variation of outer loop indices. In this simulation, we calculate the physical location of each memory access and estimate the number of overhead cycles. By restricting the penalty simulation to only the candidate windows selected in the first step, the CPU time can be significantly reduced. In addition, we do not simulate all the memory accesses but the variation information of the outer loop indices, because block-type accesses of the innermost loops do not induce any overhead cycles. Finally, the algorithm chooses the best window associated with minimum penalty.

*Experimental results:* MPEG-2 video decoding [3] is used for two-dimensional video data and involves several kernel operations such as VLD (variable length decoding), dequantisation, IDCT (inverse discrete cosine transformation), MC (motion compensation), and BA (block addition). In particular, the MC routine and the BA routine extensively access the video data stored in external memory. The access patterns of the routines are almost deterministic, i.e. the burst length is determined to be 8, 16, or 17 and the stride of the vertical axis is determined to be 1 or 2 according to the motion compensation mode.

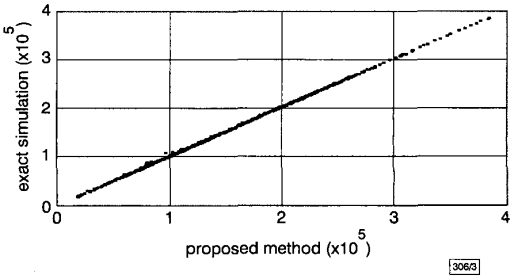


**Fig. 3** *Penalties computed by proposed method and exact simulation*

**Table 1**: Comparison of memory cycles

| MPEG2 data | Number of memory accesses | Cycles in linear translation | Cycles in proposed translation | Speedup |
|---|---|---|---|---|
| | | | | % |
| CAR | 11031205 | 15701330 | 11048208 | 42.12 |
| CHEER | 11789289 | 16367664 | 11807747 | 38.62 |
| FTBALL | 12021093 | 18058208 | 12038523 | 50.00 |
| MOBILE | 12751142 | 17891835 | 12753371 | 40.29 |
| SUSIE | 12965622 | 19264802 | 12967097 | 48.57 |
| Average | 12111670 | 17456768 | 12122989 | 44.00 |

**Table 2**: Comparison of CPU time

| MPEG2 data | CPU time of proposed method | CPU Time of exact simulation | Speedup |
|---|---|---|---|
| | s | s | % |
| CAR | 1035 | 63554 | 61 |
| CHEER | 1077 | 67522 | 63 |
| FTBALL | 1089 | 68909 | 63 |
| MOBILE | 1150 | 72397 | 63 |
| SUSIE | 1137 | 87624 | 77 |
| Average | 1098 | 72001 | 66 |

We use a set of MPEG2 video bit-streams, each of which contains 15 frames. Assuming that the number of cycles needed for each row-activation and precharge, the number of banks, and the size of a row are 6, 4, and 1024, respectively, we calculate the memory cycles of two types of address translation; one where the video data are stored linearly in order of logical addresses (linear translation), and the other where the logical addresses are correctly translated for the MPEG routine (proposed translation). As

shown in Table 1, video decoding under linear translation consumes a large number of cycles for row-activations, while the proposed translation requires almost no additional cycles. On average, the speedup of the proposed translation over linear translation is 44%.

Fig. 3 shows that the penalty computed by the proposed searching algorithm is consistent with the result of exact simulation. As indicated in Table 2, the proposed algorithm is ~60 times faster than that possible using exact simulation.

*Conclusions:* We have proposed a memory address translation method for reducing the number of memory cycles in multi-dimensional video processing applications and have presented an algorithm that searches for the window size that can be stored in a row of SDRAM. Experimental results show that the proposed translation is very effective at increasing the memory bandwidth of SDRAM by reducing the number of memory cycles required for row-changes and precharges.


*13 September 1999*
*Electronics Letters Online No: 19991300*
*DOI: 10.1049/el:19991300*

Hansoo Kim and In-Cheol Park (*Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-gu, Taejon, 301-701, Korea*)

### References

1 TAKAI, Y., NAGASE, M., KITAMURA, M., KOSHIKAWA, Y., YOSHIDA, N., KOBAYASHI, Y., OBARA, T., FUKUZO, Y., and WATANABE, H.: '250 Mbyte/s synchronous DRAM using a 3-stage-pipelined architecture', *IEEE J. Solid-State Circuits*, 1994, **SSC-29**, (4), pp. 426–431

2 KHARE, A., RANJAN PANDA, P., DUTT, N.D., and NICOLAU, A.: 'High-level synthesis with synchronous and RAMBUS DRAMs',Proc. Workshop Synthesis and System Integration of Mixed Technologies, October 1998, pp. 186–193

3 'Generic coding of moving pictures and associated audio information: Video'. ISO/IEC 13818-2: Draft International Standard, November 1993

## Method for estimating lossless image compression bound

Ning Zhang, Yujin Zhang, Qingdi Lin and Xinggang Lin

A practical method for estimating the lossless image compression bound based on high-order conditional entropy analysis is proposed. This analysis can be performed within an ordinary image as opposed to dealing with an enormous training set. Its feasibility and reliability have been verified on a large group of images naturally acquired by various imaging sensors.

*Introduction:* Lossless image compression is used in many applications. Much research has been carried out into methods for obtaining higher compression ratios [1, 2], but improvements have been limited. What would be the compression bound? How can we compute or estimate it? To our knowledge, no such bound has been established and no reliable method for estimating it has been proposed. In this Letter we propose a practical method for estimating the lossless image compression bound.

*Theoretical background:* In terms of information theory, an image $w \times h \times N$ (width × height × $N$bit) is modelled as an information source. The $w \times h$ pixels of a random variable $X$ are considered with $2^N$ possible values $\{0, 1, ..., 2^N - 1\}$. While being coded, the pixels are arranged in a certain order (usually raster scan order) as a sequence $\{x_1, x_2, ..., x_{w\times h}\}$, and processed one by one. The optimal code length of the sequence in bits is given by [3]: $L_{min} = -(\log_2 p(x_1) + \log_2 p(x_2/x_1) + \log_2 p(x_3/x_1, x_2) + ... + \log_2 p(x_n/x_1, x_2 ... x_{n-1})$. To estimate $L_{min}$, the image can be further approximated as a $K$-order Markov source $(X|S_K)$ that is characterised by the conditional probability density function $P(X/S_K)$. Here $S_K = \{s_1, s_2, ..., s_K\}$ is a vector of $K$ pixels that are scanned before $X$. The conditional entropy of the Markov source $H_K(X|S) = -\Sigma_{X,SK} p(x, s)\log_2 p(x|s)$ is traditionally considered as the bound for lossless compression, i.e. the minimal number of compressed bits per pixel can be estimated as $CBPP_{min} = L_{min}/SIZE \simeq H_K(X|S)$. To obtain an accurate estimation, the order $K$ must be large enough to precisely match the real statistical properties of the image. However, as $K$ grows, the large number of Markov states makes it difficult to estimate conditional probabilities by frequency counts within a single image. This problem is called 'context dilution' [3]. This is also why in most previous research, when the information content of images was analysed, only first or second-order conditional entropy was discussed. Roger [4] introduced a lossless compression bound in terms of additive noise caused by imaging sensors. However, no evidence can be shown that the information content inherited in an image, i.e. information other than imaging noise, is not considerable. So this bound would be too conservative. The best way to estimate the bound should be to estimate the high order conditional entropy directly.

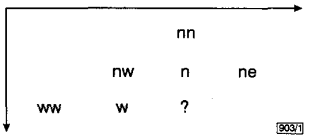


**Fig. 1** *Labelling of neighbouring pixels used in conditional vector selection*

*Multi-scale conditional entropy and memory measurement:* To obtain a relatively accurate estimation of high order conditional entropy and control the potential explosion in the number of Markov states, a multi-scale analysis method is proposed.

(i) *Step 1: Scale image pixel values at different levels*: At level $j$, only $j$ significant bits of the image pixel values are selected, i.e. for an image of $N$ bits, pixel value $x$ is scaled as: $x^j = [x/2^{N-j}]$, $j = 1, 2, ..., N$. Accordingly, vectors $S_K = \{s_1, s_2, ..., s_K\}$ are scaled as $S_K^j = \{s_1^j, s_2^j, ..., s_K^j\}$, where $s_i^j = [s_i/2^{N-j}]$, $i = 1, 2, ..., K$. Also, zero order entropy and conditional entropy are, respectively, defined as follows: $H_0^j(X) = -\Sigma_X p(x^j)\log_2 p(x^j)$, $H_K^j(X|S) = -\Sigma_{X,SK} p(x^j, s^j) \log p(x^j|s^j)$.

(ii) *Step 2: Obtain a multi-scale conditional entropy sequence with credible accuracy*: For $H_K^j(X|S)$, the number of Markov states is $(2^j)^K$, and the number of possible values under each state is $2^j$. $H_K^j(X|S)$ can be estimated without significant 'context dilution' when $(w \times h)/(2^j)^K > 2^j$.

As an example, for the aerial image Band4 (near IR component of JPEG test image Band, 736 × 736 × 8 bit acquired by linear array), the conditional vectors of 1–6 order Markov models are, respectively, selected as: $S_1 = \{w\}$, $S_2 = \{w, n\}$, $S_3 = \{w, n, ne\}$, $S_4 = \{w, n, ne, nw\}$, $S_5 = \{w, n, ne, nw, ww\}$, $S_6 = \{w, n, ne, nw, ww, nn\}$ (see Fig. 1). The conditional entropy sequence $\{H_K^j(X|S)\}$ is listed in Table 1.

**Table 1**: Conditional entropy sequence of Band4

| $j$ | $H_0^j$ | $H_1^j$ | $H_2^j$ | $H_3^j$ | $H_4^j$ | $H_5^j$ | $H_6^j$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.4262 | 0.2071 | 0.1604 | 0.1546 | 0.1532 | 0.1523 | 0.1509 |
| 2 | 1.3544 | 0.3892 | 0.2871 | 0.2735 | 0.2704 | 0.2681 | 0.2640 |
| 3 | 2.2101 | 0.7513 | 0.6011 | 0.5766 | 0.5671 | 0.5557 | |
| 4 | 2.9692 | 1.1734 | 0.9958 | 0.9530 | – | – | – |
| 5 | 3.6616 | 1.7192 | 1.5064 | – | – | – | – |
| 6 | 4.4170 | 2.3450 | 2.0730 | – | – | – | – |
| 7 | 5.3540 | 3.0207 | – | – | – | – | – |
| 8 | 6.3163 | 3.7325 | – | – | – | – | – |

(iii) *Step 3: Analyse multi-scale conditional entropy sequence* $\{H_K^{j\,(X|S)}\}$ *using memory measurement:* To explore potential regularities in $\{H_K^j(X|S)\}$, a variable called memory measurement is defined. At level $j$, memory measurement is defined as the ratio of zero order entropy to conditional entropy, i.e.