Architecture for Analog Variable Block-Size Motion Estimation

Lauri Koskinen*, Joona Marku†, Ari Paasio†, Kari Halonen *
*Electronic Circuit Design Laboratory, Helsinki University of Technology
P.O. Box 3000, 02015 TKK, Finland, e-mail: lauri.koskinen@ecdl.tkk.fi
†University of Turku, Department of Information Technology, Microelectronics Laboratory

***Abstract*— A variable block-size analog motion estimation architecture is presented. The advantage of analog motion estimation is that it can be realized in connection with a sensor array. The reference data is stored within the processing array, and thus the memory transfers associated with digital realizations are not required. The architecture is capable of performing motion estimation with all of the seven different block sizes used in state-of-the-art video standards, such as H.264. By using a deep-submicron process, a novel interconnect strategy with 30 connections per cell has been developed. This interconnect strategy enables connecting the various block sizes. Also introduced is a analog motion estimation cell which is more robust against analog inaccuracies than previous implementations. Further, additional computational elements can be added so that the array can determine a block-size partition without the need for an initial search or Lagrange estimation.**

Keywords: Motion Estimation, Parallel Processing, Analog Processing, H.264

I. Introduction

Current mobile phones incorporate megapixel-sized cameras. Although these camera sizes would allow the capture of near-HD (High Definition) video, the cameras offer video with a considerably smaller frame size. Two main reasons for the small frame size can be found: the power consumption of the video encoding and available memory. Of these, power consumption is dominant. In the near future, technology scaling will enable the integration of a larger memory, but lowering the power consumption is more difficult, due to ever increasing frame size requirements. The main reason for large power consumption is motion estimation (ME), an integral part of all video standards. ME can take up to 80% of the power consumption of a video encoder. The motion estimation power consumption is compounded with variable block-size ME, used in state-of-the-art video standards such as H.264 and Windows Media 10. For small frame sizes, such as QCIF (Quarter Common Intermediate Format), the power consumption of ME can be decreased sufficiently with algorithmic solutions. However, for larger frame sizes, the power savings offered by algorithmic solutions are not sufficient and thus, fundamentally different hardware solutions are needed.

Presented here is an analog parallel processor array for variable block-size analog ME (AME). In the previous AME realizations [1], [2], variable block-size ME (VBSME) has not been feasible. The main reasons for this are the space restrictions of interconnect when using variable block-sizes. With deep-submicron technologies, such cell inter-connections can be achieved. The presented variable block-size AME array structure is designed in 130 nm. The drawbacks of deep-submicron design cell size are the increasing inaccuracies in the analog design. Thus, a new cell structure is also introduced. This structure enables significant improvements in cell complexity, and is more robust against the analog inaccuracies.

The major advantage of the AME is that it can be implemented in connection with the image sensor array, without need for A/D- and D/A-conversions. The processing elements can either be implemented in the photo-sensor cell or as a separate array. If implemented with the photo-sensor, as is done in [3], the computational part in the pixel processor may raise the pixel cell size and degrade the image sensor quality. In these cases, as is shown in Fig. 1, the AME will have to be implemented as a separate array. The separate array can then possibly be implemented in a more compact form. For a separate AME array to be feasible, the data transfer between the computational array and the image sensor has to be analog. This can be achieved by packaging together the AME and sensor, as is the case in Multi-chip Module (MCM), and System-in-Package (SIP) realizations. In both of these cases, the advantage of the AME is that power consuming memory transfers are not needed from the frame memory and that the original frame is stored in a digital memory and referenced with the output of the AME. Thus, as the original frame is used as the input to the video codec, the analog inaccuracies do not degrade the input frame and their only effect is on the ME result. However, some of the power savings in digital memory transfer will be offset by the need to transfer the image from the sensor to the AME array.

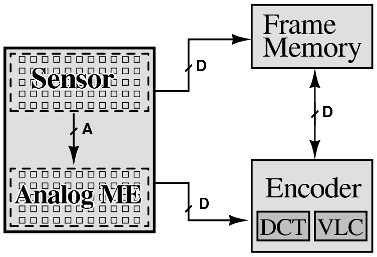


Fig. 1. Principle of suggested analog motion estimation by separate AME core. A/D = analog/digital data transfer.

The size of the array can be the same as the image size or smaller. For example, one possibility would be for one dimension of the array to be the same as the image sensor dimension and the other dimension would be specified by the search range (i.e. a 352x32 or 288x32 array for CIF-sized sensor and a [-15,16] search range). This kind of an implementation would require more memory, as each cell would have to store more than one pixel of reference data.

II. Analog Variable Block-Size Motion Estimation

The block-based motion estimation (ME) process can be described with

$$D_N(dx,dy) = \sum_{i=x,j=y}^{x+N_v,y+N_h} |ref(i+dx,j+dy) - cur(i,j)|^p, \quad (1)$$

where D is the distortion, N_vxN_h the block size, (dx,dy) the current predictor candidate, and (x,y) the top-left pixel of the macroblock. The most common values for p are p=1 (Sum of Absolute Differences, SAD) and p=2 (Sum of Squared Differences, SSD). The SSD is

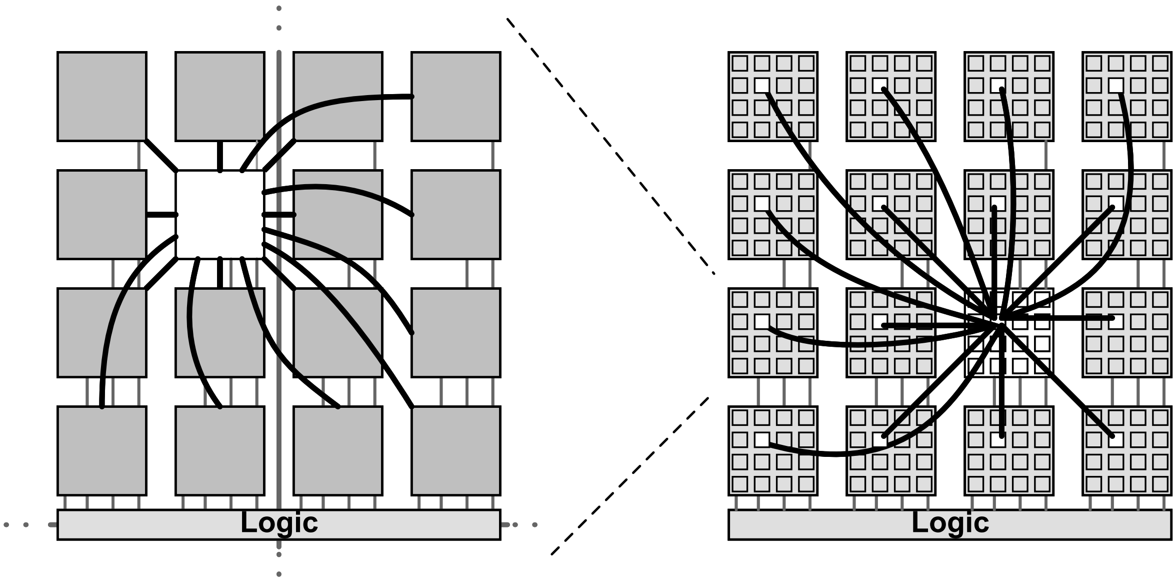


Fig. 2. Inter-cell connection for H.264 variable block-size ME. a) First neighborhood. b) Second neighborhood. The analog connections are shown in black and the digital connections in gray.

known to be superior to the SAD. However, due to its efficient digital implementation the SAD is the most often used matching criterion. The ME process can be broken into three basic operations:

1) Shift of reference pixel (reference block) data $ref(i_2, j_2) = ref(i + dx, j + dy)$.
2) The operation $|ref(i_2, j_2) - cur(i, j)|$ (SAD) or $(ref(i_2, j_2) - cur(i, j))^2$ (SSD).
3) Averaging the data over the current block $\sum_{i=x0,j=y0}^{x0+M-1,y0+N-1}$.

In analog processing arrays[1], the processing is achieved with the cell logic and by convolution operations between the central cell and it's neighbors. These spatially invariant (i.e. the convolution mask is the same for every cell in the array) convolution masks are called templates. For a 8-connected first neighborhood, a single shift operation (operation 1, above) can be achieved with the template

$$B_{sh} = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & 0 & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} \qquad (2)$$

where $b_{0,0}$ is the central cell, $b_{i=m,j=n}$=1, $b_{i\neq m,j\neq n}$=0, and (m,n) is the direction of the current shift. For operation 2) dedicated circuits are needed in the cell logic. Operation 3) is achieved with the template

$$B_{ave} = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \qquad (3)$$

III. Variable block-size AME array

A. Inter-cell connection for variable block-size motion estimation

The variable block-size AME is achieved with novel connection between the cells. This inter-cell connection for a single cell is shown in Fig. 2. The first neighborhood[2], shown in Fig. 2a, connects a 4x4 block to the central cell[3]. These 4x4 blocks are then further interconnected with the second neighborhood, shown in Fig. 2b. Thus, total number of connections per cell is 30. It follows from the number of connections that, for this array, the size of the templates in Eq. 2 and Eq. 3 is 30x30. As is elaborated below, this connection enables the processing of 16x16 area with two concurrent templates. Without this connection, the number of connections per cell would be 16^2-1=255 which would be impossible to implement even with the most current deep-submicron technologies.

Also shown in Fig. 2 is the digital logic which controls the motion estimation process and implements the digital part of the algorithm of [5]. There is a single logic block per each 4x4 group of cells. These logic blocks are further inter-connected to the logic blocks of the adjacent 4x4 cell groups. The 4x4 logic blocks are further controlled by a single block on the 16x16 cell level.

B. The motion estimation process

In array processing, due to the state invariant templates, each step of the ME process is executed simultaneously for the whole array. Thus, each cell is controlled by the same control signals.

In the shift step, the whole reference array is shifted the amount of the current shift so that the current and reference arrays are aligned. By joining together consecutive inter-cell connections, different search ranges can be achieved. When a single inter-cell connection in the first neighborhood and a inter-cell connection in the second neighborhood are joined, the search range is [-9,6]. In this case the search range is asymmetrical. This can be seen in Fig. 2b, where the [-9,6] search range includes all of the cells in the right side of the

[1] The presented array can be thought to be a special case of the CNNUM [4], where the averaging is done with B-template and other operations by specific circuits.

[2] The terms first and second neighborhood used here do not correspond to the customary concepts associated with parallel processing.

[3] The choice of the center cell is arbitrary within the four cells in the center of the 4x4 block. Here, the center cell is chosen to minimize the length difference of the connections in each neighborhood.

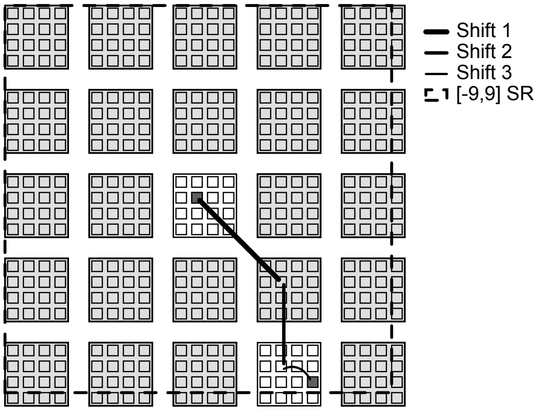


Fig. 3. Example of a [8,6] shift, which is achieved with three connections. A [-9,9] search range (SR) is also outlined.

figure. By enabling the joining of three consecutive connections, a symmetrical search range of [-9,9] can be achieved. The [-9,9] search range is shown in Fig. 3. Also shown in Fig. 3 is an example of a [8,6] shift. In Fig. 3, the first shift is from [8,6] to [7,4], the second shift from [7,4] to [3,3], and the final shift from [3,3] to [0,0]. For several shifts, there are more than one possible routes, the only restriction is that no wire can be used twice in any sequence of shifts. In practice, the sequence of shifts amounts to applying the template of Eq. 2 successively.

Second, the SSD between the shifted current pixel and the reference pixel is computed within the cell logic. Finally, the average of these values is computed with the template of Eq. 3. In practice, the average is first taken in the first neighborhood and then in the second neighborhood. The first averaging operation is implemented between the 4x4 blocks. The second average is then implemented if the current ME block size is larger than 4x4. Thus, the 30x30 sized template is split into two 15x15 templates. After these operations, the SSD value of the block can be read out from the cell.

C. Structure of the Cell

The AME cell structure is shown in Fig. 4. The hardware in the cell consists of the following blocks: both reference and current pixel memories, shift, averaging template, and the absolute value and quadratic circuits. 1) Shift: Unlike the previous realization [1], the shift hardware has been separated from the averaging hardware. This approach is more robust against analog inaccuracies. Instead of using current mirrors to shift the pixel value as a current, voltages are used in the shift, as is done in [6]. In the shift process the voltage value is transferred from current pixel memory to the target cell through 0, 1 or 2 switches, depending on the prevailing search position. The advantage of this method is that voltage value does not have to be stored in the intermediate cells. 2) Quadratic Circuit: For SSD calculation, a quadratic circuit [7] is needed. This circuit was originally intended for cubic behavior but, by changing the bias voltages, can be made to function as a quadratic approximation. As the circuit only accepts positive currents, an absolute value circuit [8] is also required. 3) Average template: The averaging template hardware is an improved version from the linear non-propagative cell presented in [1]. The division by 16 is performed in three steps. Using multiple steps scales down the mismatch effects of a single-step division by 16.

In the cell, the reference pixel data from the previous frame is stored within the cell memory. If the array size is smaller than the frame size, then are more than one reference memories. The memory block for the current pixel data in Fig. 4 could also be an interface to a photo-detector cell.

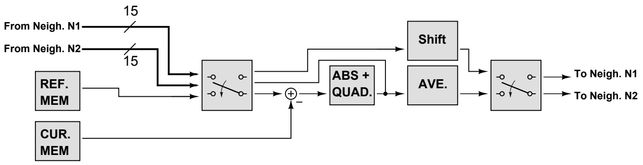


Fig. 4. Cell structure. Cell blocks: reference and current pixel memories, shift, averaging template, and the absolute value and quadratic circuits.

D. Error sources and effect on the ME result

Whereas digital operations can be performed with sufficient precision, analog operations always include different error sources. The main error sources are device mismatch [11], linearity error, charge injection (clock feedthrough) [12], and leakage currents [13]. Of these, device mismatch is the largest single error source in the structure. The mismatch is inversely proportional to the size of the transistors. The leakage currents can also be reduced by using larger transistors. Thus, the minimizing of the error in the AME is a trade-off between cell size and injected error. Unfortunately, linearity error limits the range of currents and thus the transistor sizes inside the cell. Charge injection and clock feedthrough effects were found to be minor compared to the mismatch and linearity error. All the error sources were thoroughly taken into account when designing the cell.

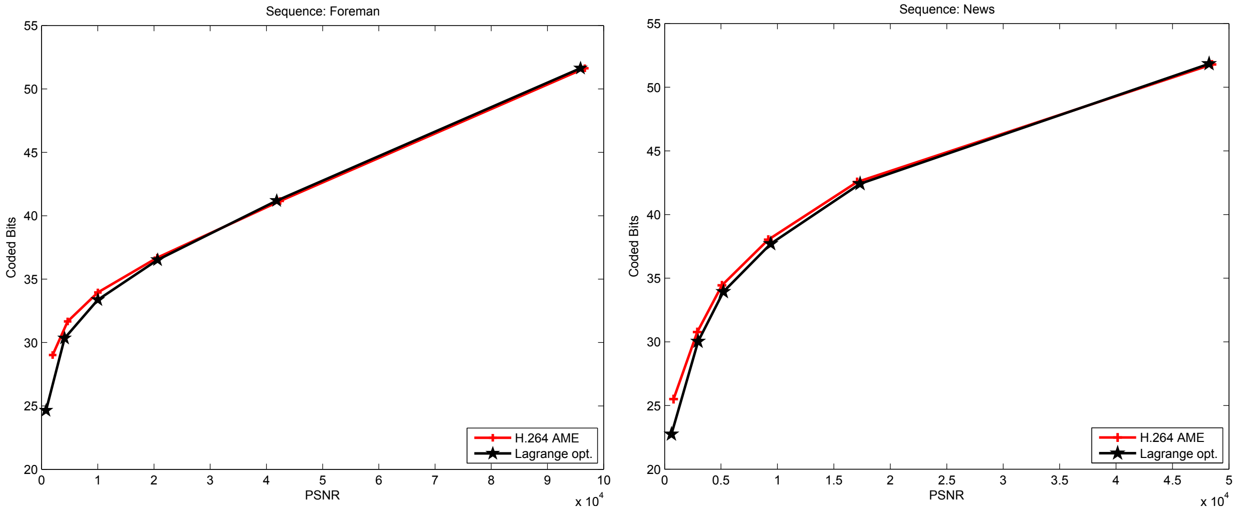
The effects of the these error sources in AME result (the found motion vector) is similar to the reducing the word-length of the pixels [14]. The effect of the error is non-linear due to: 1) Any amount of error in the ME has no effect if the block with the minimum matching criterion (the correct motion vector) is still chosen as the outcome of the search. 2) Even if an MV pointing to a larger matching criterion block is chosen, the difference in rate distortion (R/D) does not depend linearly on the AME error. This is due to the fact that the minimum block and the chosen block can be very similar in terms of PSNR and bit-rate. Also, an optimal matching criterion is input-data-dependent which means that, even with induced error, the ME result can be better in terms of rate distortion than the error-free ME result. Naturally, the probability that this will happen decreases with a larger error.

IV. Simulation Results

A. Simulation Methods and Results

All circuit level simulations are performed in 130 nm standard CMOS technology. The circuit-level simulations are performed by doing DC sweep analysis and 100 Monte Carlo simulations for all the blocks in the cell. The generated error deviations were inserted to a proprietary H.264 codec. The largest mismatch errors occurred in the averaging process.

The rate-distortion (R-D) results for the sequences Foreman and News are shown in Fig. 5. In the simulation, the coding sequence was IPPP... and Q_p was constant for the whole sequence. The used values for $QP_{H.264}$ were 10, 22, 28, 33, 38, and 50. As the purpose of this research was to study various motion estimation methods and their effect on picture quality and bit-rate, syntax generation and rate-control were not implemented. Thus, only the payload bits (i.e. bits output from the VLC) are used in the results. The output bits are referred to as “Coded Bits” in the results. Most results in literature



(a)

(b)

Fig. 5. Rate-distortion results. a) Foreman. b) News.

include the syntax bits, which has to be taken into account in possible comparisons. In Fig. 5, it can be seen that for Q_p= 10-33 the AME results are comparable to the error-free results. For Q_p= 38, 50, there is a slight drop in performance. In all likelihood, this is due to some other motion vector is chosen instead of the correct zero vector.

The static power consumption of a single cell is simulated at 83.3 μW. For low-power applications the AME can be put into an idle state after the computation, as is done in [15]. Thus, for a frame-sized AME array, the final power consumption is the product of the power consumption during the on-time and the on-time percentage. The power consumption of the array is 0.77 mW (QCIF@15fps) and 6.4 mW (CIF@30fps).

V. Conclusions

A novel variable block-size analog motion estimation array is introduced. Although the analog processing injects error into the motion estimation result, the final H.264 rate-distortion result does not exhibit significant error. The AME array is intended for low-power, real-time codecs. Future work includes combining the designed cell with the hardware previously introduced in [8] (gray-scale) and [10] (black & white). Thus, in addition to motion estimation, the array would implement the algorithm of [5] which derives the block-size partition of H.264.

References

[1] Koskinen, L.; Paasio, A.; Halonen, K.; “3-neighborhood motion estimation in CNN silicon architectures” Circuits and Systems, 2004. ISCAS ’04. Proceedings of the 2004 International Symposium on Volume 5, 23-26 May 2004 Page(s):V-708 - V-711 Vol.5

[2] Tomasini, A.; Brattoli, M.; Chioffi, E.; Colli, G.; Gerna, D.; Pasotti, M.; “B/W adaptive image grabber with analog motion vector estimator at 0.3 GOPS” IEEE International Solid-State Circuits Conference, Digest of Technical Papers, 1996, Page(s):94-95, 425.

[3] Dudek, P.; Hicks, P.J.; “A general-purpose processor-per-pixel analog SIMD vision chip” IEEE Transactions on Circuits and Systems I: Regular Papers, Volume 52, Issue 1, Jan. 2005 Page(s):13 - 20

[4] Roska T.; Chua L. O.; “The CNN Universal Machine: An Analogic Array Computer.” IEEE Transactions on Circuits and Systems-II. 1993, Vol. 40, Page(s):163-173

[5] Koskinen L.; Laiho M.; Halonen K.; “CNN Algorithm for H.264 Motion Estimation Partitions” Proceedings of the 2004 8th IEEE International Workshop on Cellular Neural Networks and Their Applications, (CNNA 2004) , 2004, Page(s):346-351

[6] Poikonen, J.; Paasio, A.; “Mismatch-tolerant asynchronous grayscale morphological reconstruction” Cellular Neural Networks and Their Applications, 2005 9th International Workshop on, 28-30 May 2005 Page(s):266 - 269

[7] Laiho M.; Paasio A.; Kananen A.; Halonen K.; “Realization of Couplings in a Polynomial Type Mixed-Mode CNN” Proceedings of the 2002 7th IEEE International Workshop on Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). pp. 436-443

[8] Kananen, A.; Laiho, M.; Halonen, K.; Paasio, A.; “Nx16 cellular test chips for low-pass Filtering large images” Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS ’04. Volume 5, 23-26 May 2004 Page(s):V-461 - V-464

[9] Koskinen L.; Laiho M.; Halonen K.; “Motion Estimation Matching Criterion Computation with Analog Circuits” Proceedings of the 2004 8th IEEE International Workshop on Cellular Neural Networks and Their Applications, (CNNA 2004) , 2004, Page(s):216-221

[10] Flak J.; Laiho M.; Paasio A.; Halonen K.; “VLSI Implementation of a Binary CNN: First Measurement Results”, Proc. of CNNA 2004 Page(s):129-134

[11] Pelgrom, M.J.M.; Duinmaijer, A.C.J.; Welbers, A.P.G.; “Matching properties of MOS transistors” *IEEE Journal of Solid-State Circuits*, Volume 24, Issue 5, Oct 1989 Page(s):1433 - 1439

[12] Balachandran, G.K.; Allen, P.E.; “Switched-current circuits in digital CMOS technology with low charge-injection errors” *IEEE Journal of Solid-State Circuits,* Volume 37, Issue 10, Oct. 2002 Page(s):1271 - 1281

[13] Roy, K.; Mukhopadhyay, S.; Mahmoodi-Meimand, H.; “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits” *Proceedings of the IEEE* Volume 91, Issue 2, Feb. 2003 Page(s):305 - 327

[14] Baek, Y.; Oh, H.-S.; Lee, H.-K.; “Block-matching criterion for efficient VLSI implementation of motion estimation” *Electronics Letters* Volume 32, Issue 13, 20 June 1996 Page(s):1184 - 1185

[15] Paasio A.; Kananen A.; Halonen K. “A QCIF Resolution Binary I/O CNN-UM Chip.” *Journal of VLSI Signal Processing.* Vol. 23, No. 2/3, 1999, Page(s):281-290.