RATE-DISTORTION PERFORMANCE ANALYSIS OF AN ANALOG MOTION ESTIMATION ARRAY

Lauri Koskinen
Helsinki University of Technology,
Electronic Circuit Design Laboratory,
P.O. Box 3000, 02015 TKK, Finland

Jonne Poikonen, Mika Laiho, Ari Paasio
University of Turku,
Department of Information Technology,
University of Turku, 20014 Turku, Finland

ABSTRACT
Emerging 3D-integration enables integrating high quality image sensors with various massively parallel processing elements. Analog motion estimation is one potential application, which is likely to result in significant benefits in the form of low power or high frame-rate 3D-integrated image sensor-processors. The system-level operation of a proposed analog motion estimation array, enabling all various block sizes from 4x4 to 16x16 is examined. The analog motion estimation circuitry has been designed as a 32x32 test array in 0.13 \( \mu \)m CMOS technology. The transistor-level simulation results combined with H.264/AVC JM 14.2 show equivalent rate-distortion results with SAD as the error measure and an approximately 7% increase in bitrate with a slight increase in image quality for SSE.

Index Terms— Motion Estimation, Analog parallel processing, H.264

1. INTRODUCTION
Low power consumption is of utmost importance for video applications in mobile devices and essential for emerging applications such as wireless video sensor network nodes. Motion estimation (ME) is the main source of power consumption in conventional video coding algorithms such as H.264. For video sensors, H.264 with its high encoding complexity is not the optimal solution. Other video coding systems, such as Distributed Video Coding (DVC), try to limit the video encoding complexity by either reducing the ME complexity or abandoning ME altogether. However, the rate-distortion performance suffers if only intra-frame coding is used.

Analog motion estimation (AME) has the potential for high efficiency, which in turn can be converted either into low power consumption or very high processing speed. The processing elements can either be implemented on the focal plane, in each photo-sensor cell, or as a separate array. The major advantage of a fully parallel focal plane AME implementation is that motion estimation processing can be implemented locally within the image sensor array, saving time and power otherwise taken up A/D- and D/A-conversions and data transfer. However, if implemented within the sensor cell, the computational part in the pixel processor may increase the pixel cell size and degrade the image sensor quality or fill factor.

To combine best possible quality of the sensor and the benefits of the processor-per-pixel structure, the optimal way of implementing AME will be as a separate array, as is shown in Fig. 1. For highest possible efficiency, the data transfer between the computational array and the image sensor should be analog. The analog data transfer can be realized between different layers of a 3D-integrated integrated circuit, by using through-silicon vias [1]. Although A/D-conversion of the frame data is still required to provide standard format image stream output of the sensor-processor, and to store previous frames digitally, the current frame image may be applied directly from the sensor directly in analog form in the motion estimation. A 3D integrated structure also allows the required data converters and digital memory to be implemented more optimally within their own silicon layer.

Although 3D-integration enables an AME array having the same size as the photo-sensor, the AME array can also have a smaller spatial resolution and the array is then applied to the larger image frame in a windowed manner. For example, the number of rows or columns can be decreased or the whole array can be significantly smaller but still symmetrical. The correct selection between different possible configurations depends on the application dependent optimal price/speed/power consumption trade-off.
The proposed AME array was implemented as a 32×32 cell test array [2]. The chip is able to implement ME within a [8,8] search range, using all block sizes from 4×4 to 16×16. The simulated system level performance of the proposed massively parallel architecture is examined here. The hardware realization of the array will be discussed briefly in this paper, more detailed discussion can be found in [2].

2. ANALOG MOTION ESTIMATION

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

$$D_N(dx, dy) = \sum_{i=x, j=y}^{x+N_r, y+N_b} |ref(i + dx, j + dy) - cur(i, j)|^p,$$

(1)

where $D$ is the distortion, $N_r \times N_b$ 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 Squares for Error, SSE). Due to its efficient digital implementation the SAD is the most often used matching criterion, although SSE can have superior performance for certain types of sequences. The ME process can be broken into three basic operations: 1) Shift of reference pixel (reference block) data 2) The operation $|ref(i_2, j_2) - cur(i, j)|$ (SAD) or $(ref(i_2, j_2) - cur(i, j))^2$ (SSE), 3) Averaging the data over the current block. In an massively parallel analog array, the whole frame can be processed concurrently as opposed to a block-by-block manner, and thus each step of the ME process is executed simultaneously for the whole array. This SIMD-type operation also means that the controls to each cell are spatially invariant, which makes the realization of the global control simpler.

2.1. Shift implementation with limited neighborhood connections

In the designed AME array, the shift step is realized by shifting the whole reference array with a very efficient hierarchical scheme. Fig. 2 shows the neighborhood connections available for each cell of the array. The processor cells are 4-connected to the 5th and 2nd neighborhoods and 8-connected to the 1st neighborhood. The shifts to the diagonal directions in the 4-connected neighborhoods are implemented by using the same cardinal neighborhood connections twice in the same shift operation, first by either N or S direction, and then by E or W connection of the same neighborhood. After this, the signal can either be taken into the target cell or to a lower neighborhood, from 5th to 2nd or from 2nd to the 1st neighborhood. By correct combinations of connections in 5th, 2nd and 1st neighborhoods, all cells within an 8-cell search area can be accessed (from 1 to 5 + 2 + 1). Also, e.g. a shift with a length of 4 can be implemented simply by moving into the opposite direction in the lower neighborhood: East(4) = East(5) – West(1). Two examples are shown in Fig. 2.

Each physical connection between cells operates bidirectionally, so that the actual number of neighborhood wires per cell is only half of the number of possible neighborhood connections. This approach, with 16 connections per cell, is significantly simplified from the 30 neighborhood connections of [3]. The implemented hierarchical shift network requires only roughly 20 global control signals. At approximately 130 transistors, even this very simple shift circuitry is a major contribution to the complexity of the cell itself. However, with this approach it is possible to implement any image shift within the search area in one step, without any iteration.

2.2. In-cell analog processing

Excluding the the shift, the rest of the AME operations are achieved with the cell analog processing circuitry shown in Fig. 3. The difference operation can be performed with a very simple current subtraction. The absolute value (ABS) and quadratic (QUAD) circuitry is effectively the same as in an earlier proposed AME designs [3], however, the cell circuitry has been optimized for the new array design. The SAD/SSE operation can be chosen by connecting either of these circuits to the output of the cell (the signals and associated switches are not shown in Fig. 3). The summing of the macroblock output values is simply realized by connecting all the output nodes of selected cells to a global output wire. Each cell can be selected separately. Activating 16 through 256 of the cells realizes the averaging of the various block sizes (4×4 –16×16).

Additionally, for testing purposes, each cell of the array
includes the C-frame and R-frame DACs, which are NMOS-type 8-bit current mode binary-weighted converters, and 16-bit SRAM memory elements for storing the pixel values. The cell circuitry is described in detail in [2].

3. IMPLEMENTATION AND PERFORMANCE EVALUATION

The 32×32 test array was realized in 0.13 μm 6 metal digital CMOS technology. The size of the chip is approximately 1.5 × 1.7 mm² and the size of a single cell is approximately 30 × 35 μm² [2]. The analog processing within the array is implemented with current-mode circuitry, this enables the shift operation to be implemented very efficiently by using simple digitally controlled current-steering switches. However, the physical implementation of the shift network is still the most complicated part of the analog motion estimation network, due to the potentially prohibitive wiring complexity.

Whereas digital operations can be always performed with sufficient precision, analog operations inevitably include various error sources, which makes the evaluation of the analog system-level performance more complex. The main error sources in the proposed circuitry are resistive effects, device mismatch, and linearity errors. As analog memories are not used for storing images, related effects such as charge injection and memory retention problems due to leakage are not an issue. The inputs provided by the in-cell DACs are static and robust. In order to perform system level simulations which can predict the behavior of the analog realization, the most significant physical performance parameters were extracted by first simulating the circuit operation at the transistor level with the applied 0.13 μm CMOS technology (operating voltage of VDD = 1.2 V).

Shifting of the reference value as a current signal can make the particular implementation vulnerable to resistive drops in the large number of series-connected switches used for the [8,8] range neighborhood connectivity. The number of switches in the current propagation path can vary from 2 ([0,0] shift) to more than 20 ([8,8] shift). The resistance of the switches can therefore lead to a deterministic shift-dependent offset, which may cause errors in the motion estimation process. However, simulations showed that the resistive loss in the switches should not be a limiting design issue for the proposed circuitry, since even with the maximum input value, the resulting difference in the shifted currents is small [2]. Other common-mode nonlinearity errors caused by the analog processing circuitry, which are not dependent on shift distance, should not be critical to the AME operation as long as they do not affect the ordering of the macroblocks in the subsequent comparison stage.

The most significant source of errors in the proposed motion estimation architecture is the mismatch between the analog transistors in the processor cells. Because the achieved analog accuracy is proportional to the area of a transistor, the circuitry should be made as simple as possible, which then enables sizing the transistors as large as possible. The mismatch variation in the different parts of the cell circuitry was simulated with a Monte Carlo simulator, with mismatch models provided by the foundry, and the different mismatch contributions were added to a system-level simulation model of the motion estimation array.

The simulated output standard deviation of the input DAC at the full signal output of 5 μA was approximately 0.5%. This is shown in Fig. 3 as σ1. The simulated relative output standard deviation of the absolute value current, without input mismatch, with the maximum input was approximately 0.6% (σ2). The simulated standard deviation in the quadrature output, which also includes the absolute value variation, but not input signal mismatch, was approximately 2.3% (σ3). As the circuit in the cell is only an approximation of an ideal quadratic function the I/O curve of the circuit, shown in Fig. 4 was also added in the simulations (σQ).

4. SIMULATION RESULTS

The simulated error deviations (σ1 − σ3, σQ in Fig. 3) were inserted in the H.264/AVC reference encoder v.14.2. The encoding parameters are shown in Table 1. These results were compared to the original JM 14.2 results.

The effect of the error sources can be simulated by adding noise (with the exception of σQ, which is modeled by changing ρ in Eq. 1) into the ME computation and the impact of this noise is analog to reducing the word-length of the pixels [4]. The effect of the error on the AME result (the found motion vector) is non-linear due to: 1) Any amount of error in the ME has no effect if the block with the minimum matching crite-
The rate-distortion (R-D) results for the sequences Container, Foreman, and News are shown averaged in Table 2 and detailed in Fig. 5. From the results, it can be seen that image quality varied from -0.01dB to 0.18dB, which can be considered as equivalent. The corresponding bitrate is increased between 0-8%. For SAD, the bitrate is increased between 0.2% and 1.3% with negligible PSNR differences. For SSE, the bitrate is increased between 6.7% and 8.3% with slightly higher PSNR differences. Some of the SSE increase might derive from JQ’s implementation of low-complexity mode decision. However, as RDO is not likely to be used in low-power implementations, RDO was not used here.

The quadratic circuit takes up approximately a third of the silicon area of each cell and also greatly increases the power consumption per cell. Therefore, based on the simulated performance the justification of the inclusion of the quadratic circuitry should be seriously considered in further research, since the device area could be used to improve the matching of the absolute value circuitry or to include digital state machines.

### Table 2. Averaged proportional bitrate and PSNR difference

<table>
<thead>
<tr>
<th>Container</th>
<th>Foreman</th>
<th>News</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitrate (%)</td>
<td>SAD</td>
<td>SSE</td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>0.02</td>
<td>0.10</td>
</tr>
</tbody>
</table>

### 5. CONCLUSIONS

The performance of a proposed cellular analog processor array for use in variable block-size motion estimation was examined with system-level simulations. The proposed motion estimation array implementation is based on very simple analog processing circuitry and includes a new efficient reference shift method, which enables all shifts within a [8,8] search area to be performed in a single step, with simple digital controls. The statistical error behavior of the processor cells was extracted from transistor-level simulations. The simulation results in JM 14.2 show equivalent rate-distortion results with SAD as the error measure and an approximately 7% increase in bitrate with a slight increase in image quality for SSE. Near future research includes experimental validation of the speed/power/accuracy performance of a designed 32×32 cell parallel analog test array for variable block-size motion estimation with a maximum block-size of 16×16.

### 6. REFERENCES


