

Different possibilities for realizing the bipolar image processing tasks in CNN field

Ari Paasio* Asko Kananen* Lauri Koskinen* and Kari Halonen*

Abstract — In this paper we compare different possibilities to implement bipolar image processing tasks required in an algorithm that is proposed for video image segmentation. It is shown that many simple operations are required to be executed many times and that these operations can be implemented with logic circuits more effectively than using conventional CNN analog approaches.

1 Introduction

In many image processing algorithms suggested to be evaluated with a Cellular Nonlinear Network Universal Machine [1] there are only few gray scale operations after which the actual processing and analysis is performed with bipolar image masks. One of these algorithms has been reported in [2] where the processing result is a segmentation result from a video image sequence. Because the gray-scale operations often have stricter requirements for coefficient accuracies, a good alternative to achieve high spatial resolution in the image processor is to separate the tasks into different categories and then to assign a separate structure for each of these tasks. Such a structure has been suggested to be used in the construction of parallel arrays in [3]. Because in the algorithms the processing results are used as initial values for further steps it is advantageous to have the processor grid size equal to the size of the image. This is rather difficult to achieve with gray-scale processing blocks, but the methods overviewed in this paper make the integration of a full size B/W processor possible.

Here we will concentrate on analyzing the requirements from the hardware point of view to realize the algorithm of [2]. In our overview the suggested templates are divided into categories depending on their complexities. The roles of more complex templates in the performance of the algorithm are analyzed and some replacements of a single template by a set of more simple templates are suggested.

A comparison is made between different hardware solutions where it will be shown that if only simple B/W operations are to be executed, solutions originating from digital approach are the best solutions

in every considered measure.

2 Required operations

Here we will list and categorize the B/W templates required in the video image segmentation algorithm [2]. The following categories are used. B/W operations with nonmonotonic outputs are assigned to *Class1* and the actual threshold logic operations are assigned to *Class2*. The rest of the templates fall into the last category *Class3*. The division follows a principle where all the considered alternatives for hardware can process *Class3* templates. The other classes require a more complicated hardware, the *Class1* operations being most difficult to perform. In the Motion Detection part of the algorithm only one template is required. This template is SMALLKILLER, and it is categorized to belong to *Class1*. At first, the existence of this type of template in the algorithm might suggest, that there are not much possibilities in the selection of the hardware realization. However, instead of trying to implement the complex dynamics of SMALLKILLER, we suggest that the task performed by the SMALLKILLER is divided into small tasks that belong to *Class3*. In this case we can use simple binary morphology to achieve the nearly same result. By analyzing the purpose of the SMALLKILLER template in the algorithm it can be concluded that it is not important to obtain exactly the same result that would be obtained by using the original template. As far as small objects from the image are removed the task of SMALLKILLER is achieved. An attractive feature of binary morphological operations is that they belong to *Class3* offering more possibilities in selection of the calculation platform. In the task for Remarkable Feature Extraction the following templates are used. Templates that belong to *Class3* are bipolar morphology operations and RESTORATION. A more complex template (HOLLOW, *Class2*) is also required. Again, by inspecting the purpose of using the HOLLOW template we can conclude that nearly the same result is obtained by using consecutive binary morphological operations. As will be discussed later, it can be both faster and more power efficient to use many simple operations to replace one more complex task if at the same time a possibility to use faster pro-

*Helsinki University of Technology, Electronic Circuit Design Laboratory, Department of Electrical and Communications Engineering, P.O.Box 3000, FIN-02015 HUT, Finland. E-mail:apa@ecdl.hut.fi, Tel: +358-9-4515013, Fax: +358-9-4512269.

cessor structure is allowed.

The third major section in the algorithm under investigation is Inter and Intra Frame Segmentation. Here, *Class3* operations are SKELETON, RESTORATION, EDGE and bipolar morphology. There are also *Class2* operations, namely the previously discussed HOLLOW and a new operation called LINE REMOVAL. The HOLLOW can be replaced in the same manner as described previously. However, there is not a simple way to replace the LINE REMOVAL by *Class3* templates. There are two possibilities to approach the problem. The first solution is to use a more complicated processor structure and the second one is to alter the algorithm more thoroughly so that the desired result is achieved. The realization of the second alternative is not described here because it is out of the scope of this paper.

Additionally to the tasks overviewed above we have found that when realizing hardware for MPEG-4 useful operations easily performed by the parallel processor array are selecting one object out of many objects so that they can be coded one at the time. Moreover, normally a minimum bounding rectangle has to be determined for the extracted objects [4]. These two tasks are easily performed by applying SHADOW templates and local pattern matching in right order. These templates belong to *Class3*.

3 Hardware alternatives

3.1 Asynchronous all digital solution

A two dimensional digital hardware realization was proposed in [5] that can realize *Class3* tasks categorized in the previous section. Some of these operations are discussed in [7] in more detail.

The design is based purely on conventional digital building blocks and therefore the physical size of the realization can be considered to scale down as smaller and smaller device features will be taken into use. The good features of this approach, additionally to the scalability, are fast programmability and the fact that there are no accuracy requirements for any 'analog' coefficients. The convergence speed is superior compared to the other approaches because only few gate delays are required for the evaluation if there is no propagation in the processing task. Because the operation is asynchronous, also the equivalent 'time constant' normally used in the CNN literature can be expressed in some hundreds of picoseconds. We have also found that the power consumed by this type of network is rather small due to two facts. First, the DC power of a logic gate is very small and power is mainly consumed during the evaluation when a gate

switches its state. Because the processing tasks are mainly such that the state of the cell changes its polarity only once or not at all, the switching power is not very high. The second power efficient feature comes from the fact that the processing task is controlled by digital signals and therefore once the preferred control signals are supplied over the array it does not take power to keep the pure logic levels during the evaluation.

3.2 Positive range high gain structure

A slightly more flexible structure for B/W image processing has been introduced in [8]. In this approach neighborhood logic, e.g. [9], with asynchronous propagation is implemented with a current mode threshold logic [10]. With this structure the restrictions in possible CNN templates are not so strict because also *Class2* templates can be realized. However, apart from the larger spectrum of possible operations, this design is worse in performance in others when compared to the first all digital approach. The threshold logic approach is not as scalable as the all digital approach because a certain accuracy level must be guaranteed for correct operation. Moreover, due to the requirements in the coefficient accuracies it may be that not all the templates that can be realized in theory, work properly. Fortunately, though, this type of positive range structure allows more spread in the coefficient values compared to the original unity gain CNN model while still preserving correct operation [11].

3.3 Other hardware solutions

Other possibilities to implement bipolar image processing tasks in the parallel processor field are at least different CNNs that process only bipolar images, e.g. [12] and Content Addressable Memory (CAM) [13]. Out of these the CAM does not offer fast asynchronous information propagation and therefore it will not be considered as a good choice for our purposes. The common factor for the CNN approaches is that they use more or less computation models that are originally proposed for mainly gray scale processing. The bipolar image processing tasks should be supposed to be 'additional' features available in the CNN. However, the B/W processing results constitute the main proportion of tasks reported with the CNN designs and no real analysis has been given on the gray scale measurement results. Because in our approach we assign different hardware for gray-scale processing we are only interested in the B/W processing performance. It has to be noted here also, that the accuracy require-

ments for the coefficients strongly limit the available spectrum of possible templates. In principle, with analog CNN computing core also the *Class1* templates can be executed. In the following comparison it is seen that if only simple B/W templates (*Class3*) are to be evaluated, the analog computation model is outperformed by the other approaches in all the considered fields.

4 Comparison

Here we will compare the figures of merit of different approaches. Because exact data is not available in detail from most of the circuits the numbers given in this part are intended to show an approximate magnitude instead of giving very precise results. It can also be noted that, e.g. the layout area taken by a single processor depends not only on the transistor configuration, but also on the available process and available number of metal layers. Moreover, the manual place and route of the layout can lead to different areas between electrically identical designs. The figures of merit we consider here are the cell density, speed of convergence and the consumed power. We will refer to the different design approaches as all digital (AD), threshold logic (TL) and others (OT) where the approach of [12] is followed.

4.1 Estimated layout area

First, a comparison is made between the estimated layout area of the cell using different approaches. We will assume a 0.25 micron digital CMOS process with six metal layers as the medium where all the approaches would be implemented. Because the local logic for storing intermediate results is, as a first approximation, the same with all the approaches, only the layout area taken by the active information processor gives differences in the estimations. We assume 1-neighborhood with 8-connected cells. The AD approach can be implemented with about three or four NAND/NOR gates together with eight direction controlled pull down branches. By using minimum size transistors we estimate the design to occupy an area of about $6 \times 6 \mu\text{m}^2$. This yields over 27000 cells/mm². Of course, the additional circuitry including local memories has to be taken into account. Based on the design in [8] we estimate an area for four local dynamic memories, communication switches and output buffer to equal $100 \mu\text{m}^2$ and therefore an estimated final cell density is 7350 cells per mm². For a TL approach the processor core for the same connectivity occupies approximately $10 \times 6 \mu\text{m}^2$ with the coefficient accuracy around 15%. This estimation is based on the

realization reported in [8] and it results a cell density of 6250 cells per mm². For OT realization we estimate that one analog multiplier occupies about $15 \mu\text{m}^2$ to achieve a 5% accuracy. We further estimate that the circuitry for limiting the cell output or state takes $300 \mu\text{m}^2$. This results to an area of $450 \mu\text{m}^2$ with ten coefficient multipliers and when the additional circuitry is taken into account an area estimate of $550 \mu\text{m}^2$ for analog cell is obtained. This would yield a cell density of 1810 cells per mm². When implementing a network of size e.g. 256×256 cells, the estimated cell densities would result to grid sizes of 8.9, 10.5 and 36.2 mm² for AD, TL and OT, respectively.

4.2 Estimated speed of convergence

Next, the speed of convergence will be investigated. To this speed we include also the time to load the corresponding neighborhood pattern or template. The AD approach will converge after a few gate delays if no propagation exists. This time is approximately four hundred picoseconds. In a global asynchronous propagation case the wave propagates to the neighboring pixel also in 400ps and therefore information propagates from one side of a 256 pixel wide image to the other side in approximately 102ns. The change of program can be estimated to take 5ns. For the TL realization we expect the speed of convergence for the non-propagating task or the speed of propagation in the propagating case to be approximately 20ns. This speed gives propagation time about $5 \mu\text{s}$ across 256 pixels. The speed advantage of the AD approach becomes even more clear when the programming time of the TL network is included. This time is 400ns in the design in [8]. The speed of programming can be possibly increased by using the techniques described in [14] where the template entries are allowed to have either a constant nonzero value or the value can be set to zero. In this case the programming speed is about the same as in the AD case. For OT we estimate the speed of convergence to be around $1 \mu\text{s}$ for non-propagating templates. This time is assumed also as the time for a propagating information to pass one pixel. Moreover, the programming of the network takes also about $1 \mu\text{s}$. To compare the speed of the structure the same task is evaluated as in the other cases. Now, for the OT approach, the network converges in $256 \mu\text{s}$. When the time taken to load the instruction to the network is taken into account the different approaches are seen to perform the same task in 107ns, $5.5 \mu\text{s}$ and $257 \mu\text{s}$. The speed advantage of AD over the TL and OT is around 50 and 2400, respectively.

4.3 Estimated power consumption

The final issue of comparison is the power consumption. Here, to get a fair comparison we use an estimated power per cell as the figure of merit. We estimate the power during the evaluation of one task. In the AD case the cell consumes power only when the cell changes its state (or output). The tasks that can be realized are such that the cell stays as initialized, or changes the state once and then stays in the changed polarity for the rest of the processing. In the following we assume a power supply 2.5V, three NAND gates consuming power when the AD cell changes polarity and the power consumed by one NAND gate to be 86nW/MHz. Now, if we further assume that about half of the cells change their polarity during the evaluation, a power consumption estimate of $0.5 \cdot 3 \cdot ((1/0.107) \cdot 86\text{nW}) = 1.21\mu\text{W}$ is obtained for evaluation in every 107ns. The power consumption for the other approaches is estimated from the reported realizations. To obtain a power consumed by one cell the total power is divided by the number of cells in the array. For TL we get $300\text{mW}/25344 = 11.8\mu\text{W}$. For the OT we assume that the 5V power supply is halved while maintaining the same current levels. With this assumption we get $1.1\text{W}/(2 \cdot 20 \cdot 22) = 1.25\text{mW}$ per cell.

5 Conclusions

In this paper we have shown that if only bipolar image processing tasks are considered to be executed on a parallel processor array the use of analog computation is not a good choice. The digital or logic approach outperforms the analog approach in all the considered measures, namely in layout area, processing speed and in power consumption. Therefore it can be concluded that only bipolar image processing tasks with non-monotonic outputs or gray scale operations are the ones that make the analog computation attractive.

References

- [1] T. Roska, L. O. Chua, "The CNN Universal Machine: An Analogic Array Computer", *IEEE Transactions on Circuits and Systems-II*, vol. 40, 1993, pp.163–146.
- [2] A. Stoffels, T. Roska, L. O. Chua, "Object-Oriented Image Analysis for Very-Low-Bitrate Video-Coding Systems Using the CNN Universal Machine", *International Journal of Circuit Theory and Applications*, vol. 25, 1997, 235–258.
- [3] A. Paasio, A. Kananen, K. Halonen, V. Porra, "Different Approaches for CNN VLSI Implementations", *the European Conference on Circuit Theory and Design*, Stresa, pp.1347–1350, 1999.
- [4] L.Koskinen, A.Paasio, A.Kananen, K.Halonen, "A MPEG-4 Shape Segmentation Algorithm", submitted to *ECCTD'01*
- [5] J.-E. Eklund, C. Svensson, A. Astrom, "VLSI Implementation of a Focal Plane Image Processor - A Realization of the Near-Sensor Image Processing Concept", *IEEE Transactions on Very Large Scale Integration Systems*, vol. 4, 1996, pp.322–335.
- [6] T. Roska, L. Kek, L. Nemes, A. Zarandy, "CNN Software Library", Version 7.0, DNS-CADET-15, Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, 1997.
- [7] A. Astrom, R. Forchheimer, J.-E. Eklund, "Global Feature Extraction Operations for Near-Sensor Image Processing", *IEEE Transactions on Image Processing*, vol. 5, 1996, pp.102–110.
- [8] A. Paasio, A. Kananen, K. Halonen, V. Porra, "A QCIF Resolution Binary I/O CNN-UM Chip", *Journal of VLSI Signal Processing Systems*, vol. 23, 1999, pp.281–290.
- [9] S. Wolfram, "Theory and Applications of Cellular Automata", World Scientific, 1986.
- [10] Y. Harada, "Threshold Logic Circuit", *US patent #5053645*
- [11] A. Paasio, A. Dawidziuk, "CNN Template Robustness with Different Output Nonlinearities", *International Journal of Circuit Theory and Applications*, vol. 27, 1999, pp.87–102.
- [12] R. Dominguez-Castro, S. Espejo, A. Rodriguez-Vazquez, R. Carmona, "A 0.8 μm CMOS 2-D Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instructions Storage", *IEEE Journal of Solid-State Circuits*, vol. 32, 1997, 1013–1026.
- [13] T. Ikenaga, T. Ogura, "A DTCNN universal machine based on highly parallel 2-D cellular automata CAM²", *IEEE Transactions on Circuits and Systems-I*, vol. 45, 1998, pp.538–546.
- [14] A. Kananen, A. Paasio, S. Lindfors, K. Halonen, "Cellular Nonlinear Network for Digital Error Correction", *IEEE International Symposium on Circuits and Systems*, Monterey, pp.255–259, 1998