2000 6TH IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings

Object Oriented Image Segmentation on the CNNUC3 Chip

P. Földesy, G. Liñán, A. Rodríguez-Vázquez, S. Espejo and R. Domínguez-Castro

Instituto de Microelectrónica de Sevilla – CNM-CSIC, Edificio CICA-CNM, C/Tarfia s/n, 41012- Sevilla, SPAIN

Phone: +34 95 4239923, Fax: +34 95 4231832, E-mail: peter@imse.cnm.es

ABSTRACT: *In this paper we show how a complex object oriented image analysis algorithm can be implemented on a CNNUM chip for video-coding. Besides the applied linear operations, several gray-scale non-linear template operations are also emulated using algorithmic solutions.*

1. Introduction†

Cellular Neural Networks (CNNs) [1] exhibits outstanding image processing capabilities. With the extension of this processing core first to the CNN Universal Machine [2], and, then towards a complex image processing system – the CNN Chipset Architecture [3] – these capabilities can be utilized in real-life applications. However, feasibility of the technology is strongly dependent on the availability of high-performance customized mixed-signal chips like the one described in [5] and [6].

In this paper we demonstrate the use of CNN-UM chips for implementing object segmetation in real-time. Object-based image and video processing represents the latest revolution in the field of computer vision. Scenes are no more simply addressed as a set of pixels or block of pixels, but as a set of objects. This approach provides new solutions for a wide range of applications from automatic surveillance to video stream coding. The implemented algorithm is based on the work of [4] with several improvements.

The experimental results have been processed by the so-called CNNUC3 (or 64×64FPAPAP) CNN-UM chip [6]. The chip comprises a 64×64 pixel array with gray-scale input and output CNN core, extensions to direct optical input, fixed-state mask, arithmetic unit, etc. It has been manufactured in 0.5μm standard CMOS technology with almost 1million transistors 80% of which operate in analog mode; the remaining 20%, used for programming, memory and control operate in digital mode.

2. Implementation

2.1 Introduction

In this section we review the goals and the main features of the segmentation algorithm. The method employs luminance contrast (low-spatial freqencies), luminance gradient (high-spatial freqencies), and consequtive frame difference (or motion) information. Besides the realization on the CNNUC3 chip, the algorithm reported in [4] has been improved as follows:

- usage of robust operations and misusage of not-terminated transients (only dc outputs),
- segment any of the possible objects regardless to their motion by improved intraframe segmentation,
- mark the moving objects,
- restoring of the moving object contours without degradation,
- avoid the need of intermediate frames between the coded ones.

After gray-scale preprocessing, three types of information are gathered: (i) contour estimation by thresholded gradient and by (ii) edges of similar luminance level areas, and (iii) thresholded frame difference. Next, this information is merged and filtered by morphological operators. Then the smaller and larger objects are separated. The final segmentation contains the external contours of the larger objects, and the skeleton of the thinner ones. We tried to use as many contour information as possible and not to destruct them by the unaviodable binary filtering. The flow-chart of the whole process can be seen in Fig.1.

In the following sections, details of each step are described with special care to the algorithmic solutions of gray-scale nonlinear operations.

2.2 Edge-Enhancing Low-Pass Filtering, Thesholded Gradient

First, the high-frequency noise component is reduced by a linear low-pass filtering, which contained an image-smoothing **B** and a Laplacian-like **A** template. This operation besides the noise suppression, also blurs the

†. This work has been partially funded by ONR-NICOP N68171-98-C-9004, DICTAM IST-1999-19007 and TIC 990826.

0-7803-6344-2/00/$10.00 ©2000 IEEE

object edges. In order to enforce the noise reduction while maintaining the edge structure, a gradient controlled low-pass filtering is used (anisotropic diffusion). Since this operation is generally highly non-linear, a simple algorithmic replacement is applied (and can be renamed as nonlinear diffusion). The algorithm comprises blurring, gradient calculation utilizing the piecewise linear output tramsfer function, and extensive usage of the fixed-state map to handle separately the edge-like areas. The block diagram of the algorithm can be seen in Fig.2. and the processed two sample frames in Fig.3.

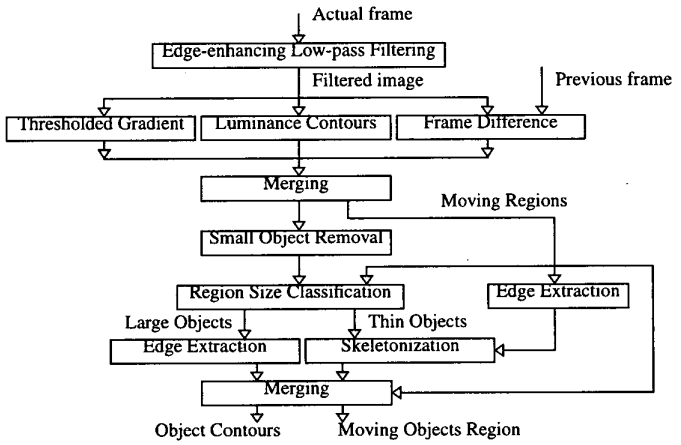

***Fig. 1:** The block diagram showing the implemented segmentation algorithm.*

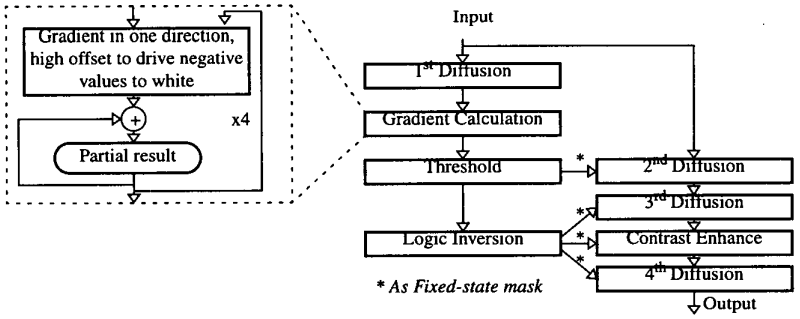

***Fig. 2:** The block diagram of the edge-preserving low-pass filter implementation can be seen in the figure. It supresses the separated edges and low intensity noise, while preserve the real edges. In the gradient calculation the sobel operator was used rotated in four directions.The role of the last three steps of diffusion and contrast enhance is to remove the noise from the edge areas and eliminate the unconsistancy between the edge and the remaining areas.*

2.3 Motion Detection

In order to invoke the motion information the pixelwise image different between two frames is calculated. In contrast to the published method, we used double thresholding on the difference instead of absolute value calculation and thresholding. In this way, the appearing and dissapearing light and dark areas can be distinguished and merged the proper one with other object information regarding to the current frame. This separation is useful because the raw difference contains information about two frames.

We found that the contours extracted by thresholded gradient can be correlated well with the appearing and dis-

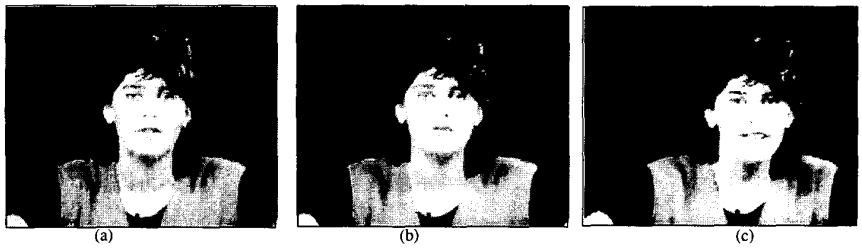

Fig. 3: In the images the results of the implemented nonlinear diffusion. Image (a) is the 65th frame of the "miss america" video sequence. Image (b) is the same frame after processing, and image (c) is the processed 85th frame of this sequnce.

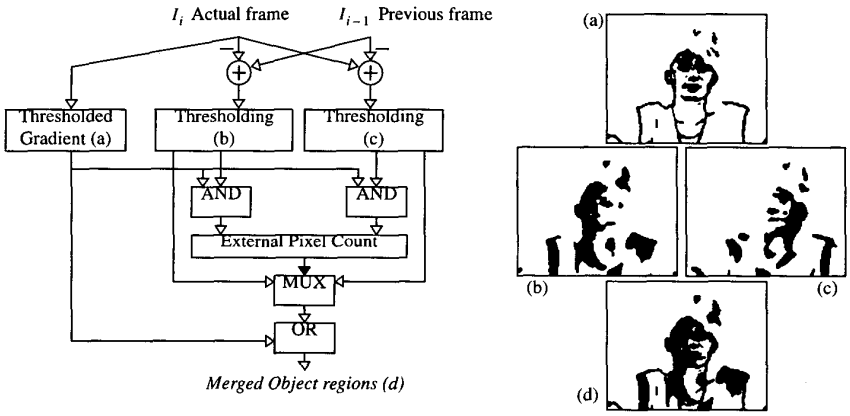

Fig. 4: The block diagram of the motion detection. The contour information of the threshhlded gradient operation is correlated with the appearing and dissappearing areas. Which area contains the more common information is merged with the contours. The images on the right side show partial results denoted by letters, which can be found also in the flow-chart.

sapearing regions. After binary correlation the evaluation is done externally by counting the black-and-white pixel ratio of the results. See Fig.4. for the flow-chart of this process.

2.4 Intensity Contour Detection

The contour estimation by the thresholded gradient is working only in cases where edge regions are sharp enough. It is not always true in natural environment, and the contours can be broken and not closed. On the other hand the luminance information diffused in regions can give this lack of information.

First, an extenal processor calculates the histogram of the incoming images dividing the luminance swing into 8-32 levels (this process is not need extensive calculations by the digital counterpart of the CNN chip). With this information some levels are choosen at the local minimums of the histogram where the preprocessed image is thresholed. With this threshold level choise, the similar large areas are not segmented.

After smoothing and edge detection on the binary results, closed and mostly not oversegmenting borders can be extracted. The corresponding results can be seen in Fig.5.

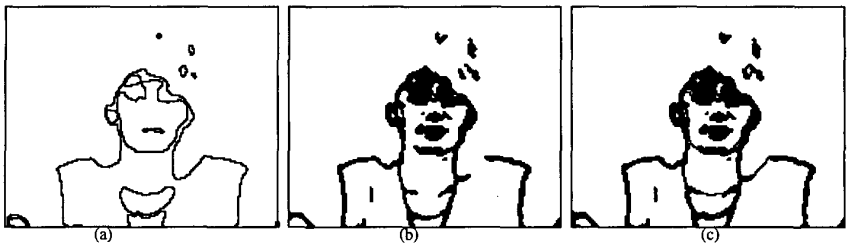

Fig. 5: In image (a) the result of the intesity based edge detection, in image (b) the result of the thresholded gradient, and in image (c) merged images can be seen.

2.5 Moving Areas, Filtering

The total area of movement is extracted as follows. The three main type of information is merged in this step and whole filling the outer parts of the frame is cleared. Using the binary contours of these image, the existing contour estimation can be enhanced.

The next step is the small object removal and the internal whole filling. In these steps morphological operators or hole filling with the commonly applied *"hollow"* function [7] cannot be used without some additional restriction because it may merge separable objects or destruct edge structures. To overcome this problem we use the fixed state map. This contains the combination of the enhanced contour estimation and the inverted moving area map (the still background). By freezing the existing contours and background the above mentioned operations can work safely.

Object size classification is used for small object removal, because the available one-template operations also could destruct the contour structure. In this step and in the laters, it is done by multiple morphological erosion and reconstruction.

The results of the moving area detection and this filtering can be seen in Fig.6.

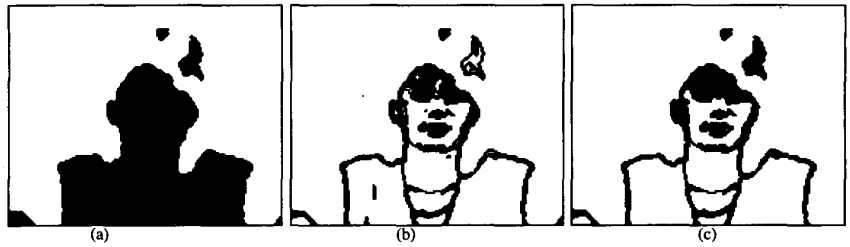

Fig. 6: The moving regions, the enhanced contour estimation, and this image after whole filling and small object removal can be seen in the images.

2.6 Final Contour Extraction

In this part of the algorithm, the goal is to maintain the external borders of the moving segments, create exact contours of the larger objects, and limit the processing time of the applied skeletonization cycles.

In order to distinguish the objects, size classification is used as was mentioned above. The smaller objects (see Fig.7a,b) are removed and stored for later edge extraction, while the remaining larger ones are processed next. During the object classification, after the morphological erosions, a so called *"core"* remains (see Fig.7c) before the reconstruction step. This core is increased (see Fig.7d) in the same amount then the erosion was applied, results in large, not connected objects. This result is also stored for later edge extraction.

If this core is removed from the original image, an edge-like image is the result with several pixel width (see Fig.7e). This image is the input of the following skeletonization process, granting the finite process time. In order to maintain the external borders of the moving region, the fixed state map is used. The input of the skeletonization is the logic combination of the thick edge map and the still background map. During the skeletonization, this background stops the peeling at the required borders. The skeleton in this way represent the internal edges, but follows the previously found external borders.

When the skeleton is ready (see Fig.7f), the background is removed, the previous small and large regions are added (see Fig.7g), and the last edge detection of this combination provide the final result (see Fig.7h).

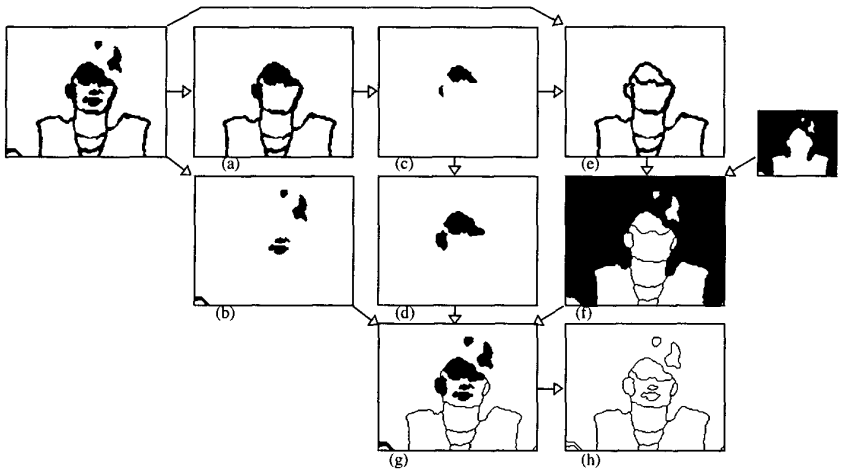

Fig. 7: Examples of the final contour detection can seen in the images. See text for detailed description.

2.7 Comments

As a result, the image containing the segment borders can be processed externally by later high-level labelling and tracking. The final segmented images can be seen in Fig.8. These example frames were choosen quite far from each other (representing a 3 frames/sec rate) in order to show the consistance of the segmentation.

The total number of template executions and logic operations in the algorithm is maximum 90 and 15, respectively. When the processed frames are the size of the chip (64×64 pixels), the required time of the processing without the I/O time is approximately 2msec. The memory management of the implementation was optimized, and since the chip contains 4 LAMs, 4 LLMs, and additional capacitances for memory interchange, all of the image processing steps of the algorithm can be executed within the chip without external storage.

In case of QCIF (176x144) sized images the 30 frames/seconds rate can be achived. It should be mentioned that the segmentation of large images into chip sized parts also includes additional image transfers in order to maintain the consistancy of the frame. But this process occures in our case only for binary images, and the overhead is slight.

2.8 Future Work

In the future exhastive test is intended to be done. The algorithm is known to fail when the background has similar contrast and intesity information that the moving objects, and itself is also changing. The solution for a more general process requires motion estimation and the preliminary knowledge of the higher level algorithms, which use the information of the segmentation. See [8] for an other survey based on global optimization technics.

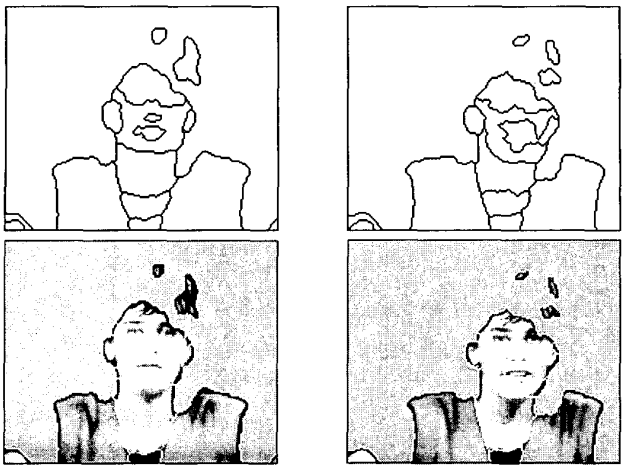

Fig. 8: Final segmentation of the moving objects in the 65th and 85th frame of the "miss america" sequence.

3. Conclusion

We implemented a object segmentation algorithm on the CNNUC3 chip. We used robust operations, image independent processing time, and solved several drawbacks of a known method. The estimated frame rate is 30 frames/sec on QCIF images.

It also became clear that the image processing capability of the CNN architecture can be optimized in the system level when conventional digital coprocessors are also present with the proper division of the tasks.

4. References

[1] L.O. Chua and L. Yang, "Cellular Neural Networks: Theory". *IEEE Trans. Circuits and Systems*, Vol. 35, pp. 1257-1272, Oct. 1988.

[2] T. Roska and L.O. Chua, "The CNN Universal Machine: An Analogic Array Computer". *IEEE Trans. Circuits and Systems II*, Vol. 40, pp 163-173, March 1993.

[3] T. Roska , "CNN Chip set Architectures and the Visual Mouse". *Proc. of the IEEE CNNA-96*, Seville, pp. 487-492, 1996.

[4] A. Stoffels, T. Roska, and L.O. Chua, "Object Oriented Image Analysis for Very-low-bitrate Video-Coding Systems, using the CNN Universal Machine", *Int. Journal on Circuit Theory and Applications*, Vol. 25, pp. 235-258, 1997.

[5] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. A. Carmona, P. Foldesy, A. Zarandy, P. Szolgay, T. Sziranyi and T. Roska, "A 0.8μm CMOS Two-Dimensional Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instructions Storage". *IEEE Journal of Solid-State Circuits*, Vol 32, pp 1013-1026, July 1997.

[6] G. Liñán, S.Espejo, R. Domínguez-Castro and A. Rodríguez-Vázquez., "The CNNUC3: An Analog I/O 64 x 64 CNN Universal Machine Chip Prototype with 7-bit Analog Accuracy". *Proc. of the CNN2000,* submitted.

[7] T. Roska, L. Kék, L. Nemes, Á. Zarándy, M. Brendel, *CSL - CNN Software Library, Version 7.2 DNS-CADET-15*. Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.

[8] T. Szirányi, K. László, L. Czúni and F. Ziliani, "Object oriented motion-segmentation for video-compression in the CNN-UM". *Journal of VLSI Signal Processing* November 1999.