IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 7, Issue 6 (Sep. - Oct. 2013), PP 58-62
www.iosrjournals.org

Field Programmable Gate Array for Data Processing in Medical Systems

Ch. Rama Mohan, Y. Devaraj, P.Prasanna Murali Krishna
M.Tech II Year ,Dr.Sgit, Markapur, Asst.Prof, Dr.SGIT,Markapur, Prof P.Prassanna Murali Krishna, H.O.D OF DECS Dr.SGIT,MARKAPUR

Abstract: *Two- dimensional &Three–dimensional (3-D) image segmentation is on of the most demanding tasks in image processing. It has been proven that only the 14-neighborship of a rhombic dodecahedron can satisfy the aforementioned requirements. The 3-D-GSC process is executed in the following three phases ,coding phases,linking phases,splitting phases. An FPGA-based digital signal processing board optimized for applications needing large memory with high bandwidth has been developed and successfully used for the parallelization of a modern image segmentation algorithm for medical and industrial real-time applications.The Use of this 128-bit coprocessor board is not limited to image segmentation.We propose the perfectly parallelizable 3-D Gray-Value Structure Code (3-D-GSC) for image segmentation on a new FPGA custom machine. This 128-Bit FPGA coprocessing board features an up-to-date Virtex-II Pro architecture, two large independent DDR-SDRAM channels, two fast independent ZBT-SRAM channels, and PCI-X bus and CameraLink interfaces.*

Key words: *Field Programmable Gate Array, Segme-ntation, Vogel bruch, Gray-valve structure code ,Homogeneous, Linking phase, Coding phase, Splitting phase, SDRAM.*

I. Introduction

If we look at medical applications, e.g., separating white and gray brain, the segmentation of these areas is a founding step in image analysis. Likewise, this type of problem arises in almost any image processing chain in nonscientific fields like industrial quality control, e.g., surface inspection.

In general, the image segmentation task works in the spatialdomain grouping together neighbored pixels or voxels to homogeneous regions if they can be considered to be similar according to a common feature. Often this will be simply the measured gray value.

Numerous techniques (see [1]–[5]) facing the elementary segmentation problem have been described during the last decades.

These are as follows:

- point oriented techniques (e.g., threshold methods) [6];
- edge detection [7] and watershed techniques [8];
- region growing techniques (e.g., region growing/merging [9], split and merge [10], pyramid linking [11]);
- deformable models [12];
- direct in-image-data classification methods [13];
- application-specific solutions.

Usually, the segmentation result (a region or label data set) is fed into a subsequent analysis, often realized by classification methods. The quality of various segmentation algorithms differs in many ways, so that for every application the best suitable
approach should be selected to match the requirements.

Not every algorithm can be hardware accelerated efficiently to benefit from the advantages of modern FPGA and DSP devices.Rehrmann [14] proposes a two-dimensional (2-D) segmentation
algorithm (CSC) based on a hierarchical approach of
Hartmann [15] to include local precision by keeping a global overview. Vogelbruch [16]–[18] extended this algorithm to 3-D(3-D-GSC) by identification of an appropriate 3-D island structure and proving this to be unique in 3-D. These approaches both are implicitly parallel because the island itself and islands of each hierarchy level can be processed independently. The main drawback is constituted by the large amount of memory needed and the high bandwidth requirements to keep all parallel processing units busy to make them most efficient. The general concept to adapt the GSC software implementation for use on hardware exploiting the GSC inherent parallelization capabilities is described later in this paper.

Therefore, specialized hardware optimized for maximum memory throughput is required. In the past 10 years many special hardware solutions (e.g., CNAPS [19], IMAP [20],SYMPHONIE, CC-IPP and SIMD architectures) have been proposed but none has been successfully brought to market. Today image processing tasks of low complexity are implemented on graphic controllers with DSP units (e.g., Matrox Genesis). For the solution of more complex tasks a significant trend to use standard FPGA architecturesis visible.

But a market analysis has shown available hardware architectures not be suitable for the complex requirements of the innovative algorithm used here. Therefore the hardware platform proposed in this paper features the special memory requirements ofthe 3-D-GSC algorithm described earlier. It contains two large banks of SDRAM and two banks of fast SRAM for caching purposes.The memory word width is set to 128 bit per bank to match the island structure of the GSC algorithm. Due to the use of an FPGA as processing core the board is suitable for manyother image or high memory intensive algorithms.

II. The 3-D-Gsc Segmentation Algorithm

The 3-D-GSC [16] merges local precision and global view by re-evaluating homogeneity decisions taken on a lower hierarchy level on the basis of the global view through a subsequent splitting of contiguous but non-similar regions. The GSC algorithmworks on a newly developed 3-D island structure havingthe properties introduced in the following due to the requirements indicated:

- homogeneous periodical lattice efficient algorithmic realization;
- covering of all lattice points complete region linkage;

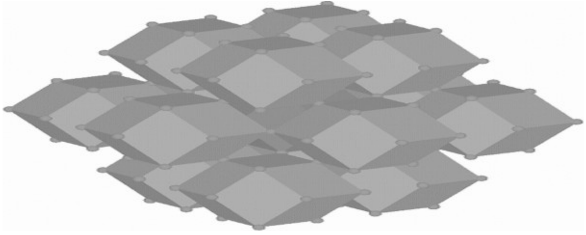

Fig. 1. The island structure of the 3-D GSC is a rhombic dodecahedron.

- central symmetrical islands isotropic region linkage;
- complete simple overlapping unique connectivity and splitting;
- simple hierarchy multiscale approach, recursive implementation.

In [16] it has been proven that only the 14-neighborship of a rhombic dodecahedron (see Fig. 1) can satisfy the aforementioned requirements, but not all overlapping points in a macro island can be examined during linking due to the inhomogeneous neighborhood structure. This requires an explicit splittingof these nonexamined overlappings.

The 3-D-GSC process is executed in the following three phases (for simplification, these are depicted in Fig. 2 for the 2-D case):

- In the **coding phase** neighboring and similar voxels are combined to local regions of the lowest hierarchy level with a resulting feature computed as a weighted mean of the contributing voxels.
- During the following **linking phase** these regions are linked hierarchically to global segments up to the highest hierarchy level. A region of one hierarchy level consists of contiguous and similar regions of the hierarchy level underneath. The gray value of this region once again is calculated by the weighted gray value mean of the participating regions.
- During the linking phase two regions can be non-similar, but overlapping. Therefore, in order to obtain a disjoint segmentation result, the overlapping area must be separated afterwards. This procedure is carried out recursively down to the lowest hierarchy level during the **splitting phase**, which is initiated immediately after the linking of an island. Thus global decisions are taken down to the local voxel level.

III. Parallelization And Modification Of The Algorithm For Hardware Implementation

For an efficient implementation of the algorithm on FPGA based platform several modifications had to be performed. A main aspect here is to reduce memory accesses with respect to the not very fast FPGA-DRAM connection (compared to current PC arch.). The modifications comprise as follows.

The indirect database scheme using a key table for database access to registered regions (code elements) has been dropped in favour of regular-sized database entries. Of course, this wastes a fair amount of memory but the number of memory accesses is significantly reduced. The regular-sized database entry format allows for replacing absolute region coding addresses by relative ones which can be calculated by the FPGA concurrently without performance penalty.

The coding phase has been optimized for minimal hardware resource consumption and maximum speed but still giving the same results as the software reference implementation.

To avoid time-consuming neighborhood searches by looking up position indices before database access, we developed a new linking scheme using overlapping lists which are generated during the registration of code elements.

The nondeterministic recursive splitting phases had to be replacedby a regular root-to-level-0 label propagation scheme including additional merging capabilities during the generation of the segmentation result (label dataset and segments' list).

In Fig. 2, the much more simpler 2-D case is depicted and shows that the best module utilization can be achieved by using two independent memory devices for database storage and two additional independent fast memory blocks for intermodule data exchange or caching purposes.

Likewise, this shows that three layers of the hierarchical A structure are processed in parallel covered by the coding array (20 coding modules) and the two linking arrays between the two memory banks proposed (first processing line from top). Additional a second processing line with two linking arrays can also start to work if the memory contains sufficient data for the modules. For larger images the two linking modules of the first processing lines will be reused. Please note, the arrows shows the data flow from the input image via different data base levels up to the resulting label image, and see the attached timing diagram that shows the process utilization of the processing lines.

The Splitting and Labeling module cannot be processed in parallel because it needs access to all hierarchical levels in both memory banks

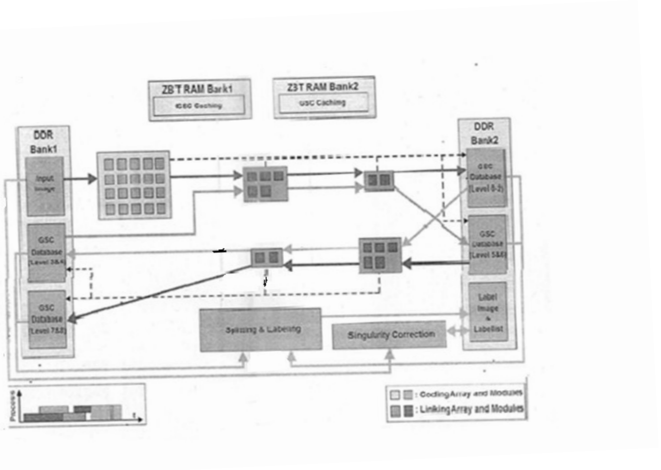

Fig.2. Flow diagram of the hardware algorithm; note that the two processing lines can work in parallel if they have sufficient data.

A market survey revealed that there was no suitable board available meeting the required independent memory layout for the algorithm as described above. Hence, we had to develop our own FPGA-based platform that fits the needs mentioned above and satisfies the requirements of the cooperating partners. This platform is described in Section 4.

IV. Hardware Acceleration Board

To achieve the goal of segmenting 2-D images up to pixel and 3-D images up to voxel (16 bpv) in real-time a PCI compliant extension board supporting standard PC systems featuring a Xilinx Virtex II Pro FPGA processor (gates)and equipped with sufficient local memory organized in four separate channels has been built (see Fig. 3).

It contains socketed DDR SDRAM modules up to 1 GB with 266 MSamples/s (PC2100) operating at 110 MHz to obtaindouble-word data (128 Bit) from the FPGA ports; this yields 1.6 GByte/s per channel. For caching purposes and fast random access two additional smaller SRAM memory banks (up to 8 MB) consisting of independent ZBT (zero bus turn-around) are available, operating at 110 MHz and a data width of 128 Bit.

Together, all four banks achieve an aggregated bandwidth of about 6.4 GByte/s.Industrial applications (e.g., for quality assurance) can be accomplished by connecting external cameras directly to the FPGA board or via the CameraLink interface. Image data can also be transferred from any other source via the PCI bus (64 Bit at 66 MHz). For future purposes, two 32 Bit extension connectors are available.

Industrial applications (e.g., for quality assurance) can be accomplished by connecting external cameras directly to the FPGA board or via the CameraLink interface. Image data can also be transferred from any other source via the PCI bus (64 Bit at 66 MHz). For future purposes, two 32 Bit extension connectors are available.

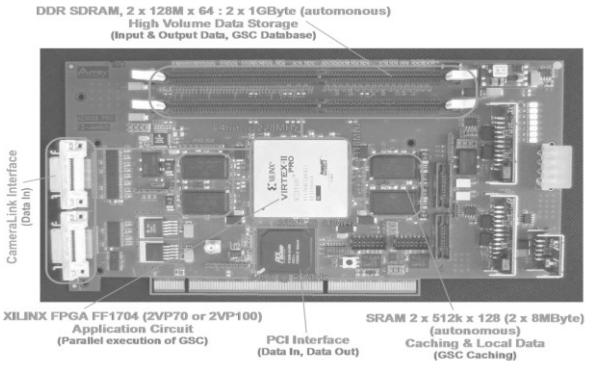
DDR SDRAM, 2 x 128M x 64 : 2 x 1GByte (automonous)
High Volume Data Storage
(Input & Output Data, GSC Database)
CameraLink Interface
(Data In)
XILINX FPGA FF1704 (2VP70 or 2VP100)
Application Circuit
(Parallel execution of GSC)
PCI Interface
(Data In, Data Out)
SRAM 2 x 512k x 128 (2 x 8MByte)
(autonomous)
Caching & Local Data
(GSC Caching)

Fig. 3. Photograph of the FPGA coprocessor board with the main parts depicted.

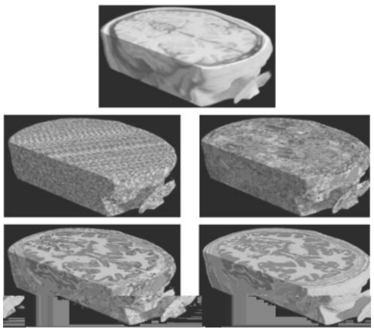

Fig. 4. Segmentation and classification results on a simulated 3-D MRI brain dataset. (a) Original image. (b) Initial region codings. (c) Intermediate codings. (d) Completely segmented. (e) Classified with n-to-1 classifier

V. Performance And Results

Fig. 4 shows the segmentation and classification result of the used GSC algorithm on a simulated 3-D MRI brain dataset. On the basis of the segmentation results 98.5% of all voxels in the MRI data set are detected properly as class members using a simple n-to-1 classifier.

Fig. 5 shows a runtime comparison in a logarithmic scale between nonoptimized/optimized SW implementation and the estimated HW performance based on memory access rates. The best case is given for a fully exploitation of the parallel laid out memory banks and the worst case indicates pure sequential data access. Proceeding on these assumptions, a speed-up factor of 30–100 times depending on resolution and image contents is expected to be achieved.

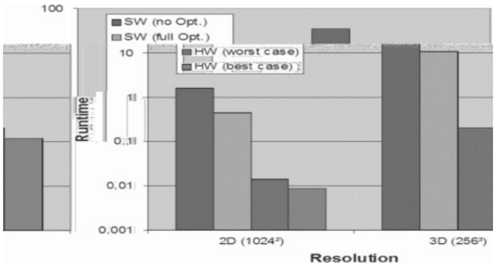

Fig. 5. Comparison of the run-time between the software implementation and the estimated hardware performance.

VI. Conclusion And Outlook

Industrial applications (e.g., for quality assurance) can be accomplished by connecting external cameras directly to the FPGA board or via the CameraLink interface. Image data can also be transferred from any other source via the PCI bus (64 Bit at 66 MHz). For future purposes, two 32 Bit extension connectors are available.

On the basis of the segmentation results 98.5% of all voxels in the MRI data set are detected properly as class members using a simple n-to-1 classifier.An FPGA-based digital signal processing board is used for large memory with high bandwidth has been developed and successfully used for the parallelization of a modern image segmentation algorithm for medical and industrial real-time applications.

Although the GSC algorithm is optimized for gray scaled images, this can be easily extended to ultidimensional data when an appropriate distance function is given. Use of this 128-bit coprocessor board is not limited to image segmentation but might also comprise applications such as FPGA-based project development and prototyping, simulation, reconstruction, FEM, etc.

References

[1] L. P. Clarke, R. P. Velthuizen, M. A. Camacho, J. J. Heine, M. Vaidyanathan, L. O. Hall, R. W. Thatcher, and M. L. Silbiger, "MRI segmentation: Methods and applications," *Magn. Reson. Imag.*, vol. 13, no. 3, pp. 343–368, 1995.

[2] R. C. Gonzalez and R. E.Woods, *Digital Image Processing*, 2nd ed.Reading, MA: Addison-Wesley, 1993.

[3] B. Jähne*, Digitale Bildverarbeitung*, 4th ed. New York: Springer, 1997.

[4] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognit.*, no. 26, pp. 1277–1294, 1993.

[5] J. C. Russ*, The Image Processing Handbook*, 3rd ed. Boca Raton, FL: CRC, 1999.

[6] S. U. Lee and S. Y. Chung, "A comparative performance study of several global thresholding techniques for segmentation," *Comput. Vis., Graphics Image Process.*, no. 52, pp. 171–190, 1990.

[7] D. Ziou and S. Tabbone, Edge Detection Techniques—An Overview Transl.: Tech. Rep. Dep. informatique, Univ. Sherbrooke, 195, 1997.

[8] L. Vincent and P. Soille, "Watersheds in digital spaces; An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 13, pp. 583–597, 1991. DILLINGER *et al.*: FPGA-BASED REAL-TIME IMAGE EGMENTATION 2101

[9] R. K. Justice and E. M. Stokely, "3D segmentation of MR brain images using seeded region growing," *IEEE Eng. Med. Biol. Soc.*, no. 18, pp. 1083–1084, 1996.

[10] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *J. Assoc. Comput. Machinery*, no. 23, pp. 368–388, 1976.

[11] P. J. Burt, "The pyramid as a structure for efficient computation," in *Multiresolution Image Processing and Analysis*, A. Rosenfeld, Ed New York: Springer-Verlag, 1984, pp. 6–35.

[12] T. McInerney and D. Terzopoulos, "Deformable models in medical images analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91–108, 1996.

[13] Y. Shimshonim, "Introduction to classification methods," in *School of Mathematical Sciences*. Tel-Aviv, Israel: Tel-Aviv :

[14] V. Rehrmann, "Stabile, echtzeitfähige Farbbildauswertung," *Koblenzer Schriften zur Informatik*, no. 1, 1994.

[15] G. Hartmann, "Recognition of hierarchically encoded images by technical and biological systems," *Biolog. Cybern.*, no. 57, pp. 73–84, 1987.

[16] J.-F. Vogelbruch, "Segmentierung von Volumendatensätzen mittels dreidimensionaler hierarchischer Inselstrukturen," inSchriften des Forschungszentrum Jülich/Informationstechnik Ph.D. dissertation, RWTH Aachen, vol. 3, 2002.

[17] J.-F. Vogelbruch, P. Sturm, R. Patzak, L. Priese, and H. Halling, "3D Segmentierung mittels hierarchischer Inselstrukturen," in *Bildverarbeitung für die Medizin*, Leipzig, 2002.

[18] J.-F.Vogelbruch, R. Patzak, and H. Halling, "3D segmentierung und visualisierung von volumendatensätzen," *Remagener Physiktage*, 2002.

[19] D. W. Hammerstrom and D. P. Lulich, "Image processing using onedimensional processor arrays," in *Proc. IEEE*, 1996, vol. 84, no. 7, pp. 1005–1018.

[20] M. W. van der Molen and S. Kyo*, Documentation for the IMAP-VISION Image Processing Card and the 1DC Language*. : NEC Incubation Center, 1997.