Hardware Accelerated Watershed Based Echocardiographic Image Segmentation

D Drop[3], Z Latała[2], J Kasperek[1], R Patury[1], P J Rajda[1],
J Sadowski[3], L Szydłowski[3], L Wojnar[2]

[1]AGH University of Science and Technology, Poland
[2]Cracow University of Technology, Poland
[3]Cardiac Surgery and Transplantology Clinic, Medical College, Jagiellonian University, Poland

Abstract

The proposed poster presents a hardware accelerated left ventricle detection algorithm. Based on marker controlled watershed segmentation method, the algorithm core works sequentially: detection result from a given frame is taken as the segmentation marker to process the next frame. The complete algorithm is written in Matlab. It performs more then 20 elementary operations and needs tens of minutes to compute one movie. As this is a serious disadvantage in clinical application, hardware acceleration has been proposed. Authors built an experimental PC based system equipped with Data Translation frame-grabber and Alpha-Data ADM-XRC-II board with high capacity Virtex II Field Programmable Gate Array. Hardware implementation of particular algorithm stages shows significant speedup due to the possibility of computation concurrency in FPGA Virtex II device.

1. Introduction

Despite recent three-dimensional echocardiography progress, classic two-dimensional examinations seem to be still used in everyday clinical practice. The quantitative analysis of left ventricle volume and the ejection fraction are usually performed using Simpson method after a manual (or semi-automatic) tracking of the left ventricle during offline analysis. This approach has many disadvantages since it highly depends on the operator's experience and introduces significant delays.

Proposed experimental method implements real time image processing of the echocardiographic data and enables online visualisation of quantitative heart parameters. The detection algorithm originally developed and described in [5] has been modified and adapted for the new acquisition method and for hardware implementation. The left ventricle detection algorithm uses image processing methods and, in a pure software implementation, takes tens of minutes, even on a modern PC, to create the output movie from the full resolution input data. The algorithm fundamental step uses marker controlled watershed segmentation method. Left ventricle markers are obtained using many filtration operations from simple morphological erosion / dilation, through a histogram equalization and rank filtering, up to a hybrid radial gradient calculation.

2. System architecture

Currently, there are two versions of system architecture. The first, dedicated to offline work, reads acquired echocardiographic digital data and sends them to the specialised hardware acceleration module based on Xilinx FPGA Virtex II device. The authors used ADM-XRC-II-Lite board from AlphaData as the accelerator. The FPGA technology is well known for its high computational power especially in a signal processing field. Virtex II FPGA devices offer high density, fine granularity, massive parallelism and provide the opportunity for the run-time customisation of the performed computation task. Fig.1 presents the accelerator board schematic diagram. The main

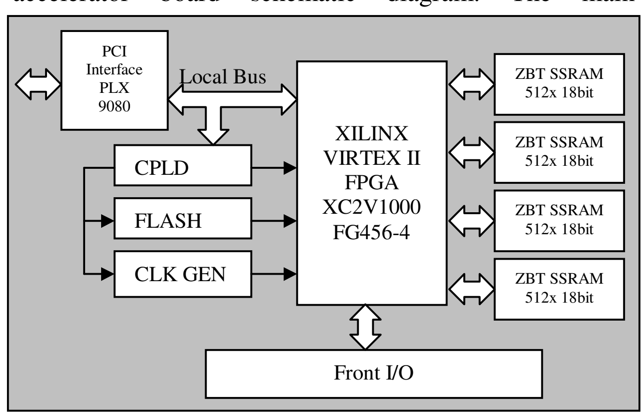

Fig. 1 ADM-XRC-II-Lite board block diagram

computation element is the one million gate XC2V1000 FPGA device, equipped with four independent ZBT RAM 512k × 18bits memory devices and connected with host PCI bus via the PLX 9080 PCI interface. The Software Development Kit (drivers & API C++ libraries) allows the hardware-software implementation of complex signal processing procedures.

The second version of the architecture uses the S-Video output from the echocardiographic device to the hardware acceleration platform through the video acquisition frame-grabber from Data Translation. Using such an interface impairs image quality and resolution, and fixes the data frame rate to the video standard, but definitely lowers the image data flow and therefore allows real time processing.

3. Watershed transform

The watershed transform is a well known tool for image segmentation. Its main drawback is the oversegmentation produced which can be overcome by using markers. In the left ventricle detection challenge an *apriori* knowledge of human heart physiology can be used to compute proper markers. So-called "entry markers" are obtained through several neighbourhood filtering operations as e.g. morphological erosion / dilation, ranking filtering, and histogram equalisation. After entry markers are detected, the main watershed transform is performed. Since the detection results depend to a great extend on local image minima number, the input image pre-processing is required. Authors used the so-called radial gradient hybrid filtering. The idea is presented on Fig.2. The filter is composed of four

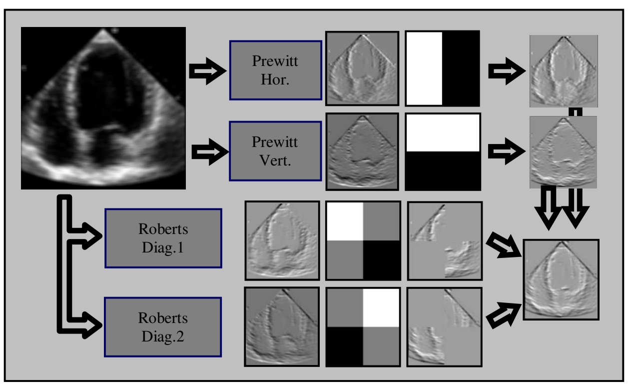

Fig.2 Radial gradient filtering concept

different masks: two of them are well known Prewitt masks (one enhances horizontal and the other – vertical details) and the two others are Roberts diagonal masks (one works at +45° and the other at -45°). Such a mask combination enhances left ventricle borders and eases the watershed segmentation task. The hardware implementation of the radial gradient filter is quite straightforward since all convolution masks coefficients are '1', '-1' or '0'. The output pixel value is just an algebraic sum of the neighbourhood pixels. All these four operations can be performed simultaneously in the FPGA implementation. Also the output value pixel normalization is performed in FPGA using a very fast and simple Look-Up Table solution. The final image is constructed by a union operation from partial convolutions outputs. The question is where to set the image center point. Originally, the "center of gravity" used to be computed from the entry marker – yet, since its hardware implementation is quite complex, the linear approximation was used instead. This assumption has been positively verified on tests images.

Although the watershed transform is a fairly well described method, implementation algorithms for discrete images may vary. Methods based on recursive algorithms proposed by Vincent & Soile [12] or on topographic distance by Meyer [6] are computationally expensive, especially when the hardware implementation is considered. FPGA devices offer massive parallelism and are not so effective when sequential operations are performed. For this reason, this work uses a method described in [2] by Bieniek and Moga, and is based on local connected components and markers. Fig 3. presents watershed results: input image (A), watershed created after radial gradient filtering without averaging (B), radial gradient with averaging filtering (C), markers (D and E) and the final watershed performed with markers (F).

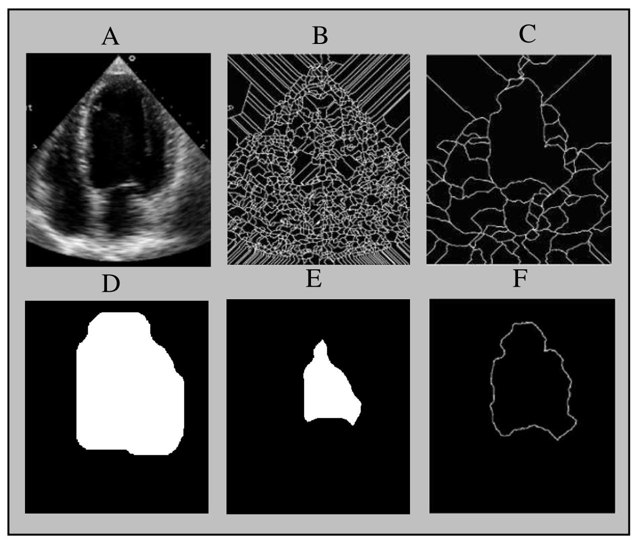

Fig.3 Watershed transform results

Implemented watershed has several stages. First, based on object and background markers supplied, input image is masked, and the label database is initialized by background and object labels. Then directed label graph is constructed. During the first image scan every pixel's value is compared to those of pixels in its neighborhood. In case of darker (i.e. lower pixel value) neighbor, its address is written to the label database (if more then one darker pixel is found, the lower pixel address is arbitrary taken). Absence of darker neighbors means that the pixel belongs to a plateau. At this moment, it is unknown whether it is a local minimum or non-minimum plateau. If it is not a background or object label, a new label is assigned to it in the labels database. The next step is the non-minimum plateau removal. Labels database is scanned, and for the pixel in the plateau with given label LAB_P the neighbor with different label is searched. The first different label neighbor found is put into the FIFO, and the pixel neighborhood scanning is stopped. The whole scanning is repeated for any pixel with LAB_P label. After this step, the FIFO holds addresses of pixels which are on non-minimum plateau borders and have darker neighbors. So, the next stage is the darker neighbors labels propagation on non-minimum plateau members. FIFO is read, and label propagation is performed for every pixel address from the FIFO. After this stage there are only minimum plateaus with no darker pixels around in the processed image. Directed graph construction is completed when minimal plateau labels are unified. This is achieved by re-addressing minimal plateau labels found in the previous step. At this point, the authors modified the algorithm idea presented in [2] by a special treatment of the object and of background labels. Once the directed graph label construction has been completed, segmentation is performed by label propagation according to graph walks. Note, that every non-local minimum pixel label '*p*' points to the pixel '*q*', from which it receives label in the label merging process. Thus, pixel labels point at the most rapid drowning paths and end at the local minimum pixels. During this label propagation stage a certain "path compression" method is used. If from a given image point 'P', minimum 'M' is reached, that if point 'Q' crosses point 'P' that implies that it also will reach point 'M'. Such a determinism can significantly speed up watershed computation. After the label propagation in the output image oversegmentation effects are often observed. They are caused by local minimum points located on the outside of the object marker. For this reason the final region merging was required, and "the lowest pass" method was used. This part of the whole procedure is very vulnerably for input image noise. That is why the source image is averaged after applying radial gradient.

4. Hardware implementation

The design was prepared in VHDL under Aldec's Active-HDL environment and synthesized using Xilinx ISE software. Radial gradient filter implementation in FPGA chip is highly effective. With only 66 MHz data clock, one image frame (366 × 436 pixel resolution with 8bits/pixel data representation) processing time is approx. 2.5ms. This can be compared to MATLAB implementation on PC: Pentium4 HT 3.0 GHz and 512 MB RAM gives 10.5 seconds processing time.

Fig.4 presents the watershed transform hardware implementation block diagram. The structure of the implementation was highly directed by the used APLHA DATA ADM-XRC-II-Lite FPGA board architecture. Control register interface allows software control of the designed unit, defines processed image size and holds status bits. CKU unit generates clock signals, initialises the whole unit and external memory blocks. CTRL unit is the main one and sequentially starts the five computation units:

- LSU – initialisation and labelling of the selected areas, neighbor pixel comparison and plateau pixel labelling,
- EPU – non-minimal plateau removal and minimal plateau unification,
- LPU – labels propagation,
- RMU – marker controlled region merging,
- EDU – final segmentation.

Processed data are stored in ZBT SSRAM modules. For computation time it is very important to maximise data throughput since the proposed implementation requires many data transfers between memory banks and internal computation units.

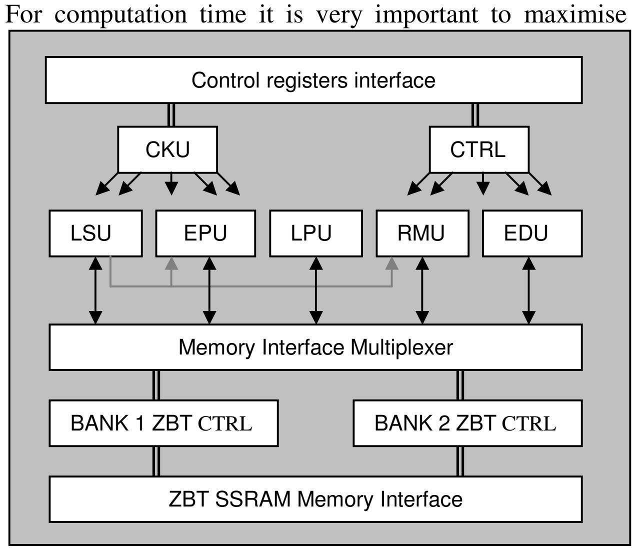

Fig.4 Watershed transform hardware implementation

5. Experimental results

The watershed transform has been implemented in XC2V1000FG456-4 Xilinx FPGA device and required roughly half its resources. Current image resolution limit is set up to 448 × 448 pixels. For larger images FIFO length in EPU module must be extended. During the investigations, images with 366 × 436 resolution were analysed and the average FIFO usage ranged from 3000 to 6000 words. The average processing time was 15-30ms per image frame and it was 10-20 times shorter then performed on the reference PC. Further increase in speed can be achieved by dividing the source image into sections and performing watershed on these sections concurrently. Such a method was successfully tested in MATLAB (division into 4 or 8 sections), but there were not enough independent memory blocks on ADM-XRC-II-Lite FPGA board to implement it at real platform.

Although authors' emphasis was mostly on hardware implementation, the algorithm verification was also performed. 63 patients with different LVEF data records were gathered and the special GUI tool for the manual left ventricle detection was prepared. Systolic and diastolic frames were selected by experienced cardiologists and ventricle borders were manually marked as well. Nineteen best quality data records were selected for comparison and showed that the referred automatic algorithm works in an acceptable way. For the rest of data records a further algorithm improvement is considered.

6. Conclusions

Watershed transform has been successfully implemented in hardware using FPGA programmable devices. The computation time shows that the real time visualisation of the left ventricle is possible. Then the real time computation of the quantitative heart parameters like LVEF (ejection fraction) and others, which can be derived from the detected left ventricle area, can be achieved as well. This opens new diagnostics possibilities for the real time heart monitoring.

Acknowledgements

This work is supported by Polish Committee of Scientific Research (project number 4T11E 01524). Authors would also like to thank Aldec, Inc. and Alatek, Inc. for Active-HDL & DVM-HES packages donations, which made this work possible.

References

[1] Beucher S., Meyer F., The morphological approach to segmentation: The watershed transformation, Mathematical Morphology in Image Processing, New York, 1993.

[2] Bieniek A., Moga A., An efficient watershed algorithm based on connected components, Pattern Recognition 2000;33: 907-916

[3] Bleau A., Joshua Leon L., Watershed-based segmentation and region merging, Computer Vision and Image Understanding 2000;77:317-370.

[4] Haris K., Efstratiadis S.N., Maglaveras N., Pappas C., Gourassas J., Louridas G., Model-based morphological segmentation and labeling of coronary angiograms, IEEE Transactions on Medical Imaging, 1999; Vol. 18, No. 10.

[5] Latała Z., Zastosowanie komputerowej analizy obrazu ultrasonograficznego do badania serca, PhD Thesis (In Polish). Cracow University of Technology, 2002.

[6] Meyer F., Topographic distance and watershed lines, Signal Processing 1994, 38:113-125.

[7] Meyer F., Vachier C., The viscous watershed transform, Journal Of Mathematical Imaging And Vision, 2005;22: 251-267.

[8] Moga A., Cramariuc B., Gabbouj.M., Parallel watershed transformation algorithms for image processing, Parallel Computing 1998;24:1981-2001

[9] Pitas I., Costaces C.I., Memory efficient propagation-based watershed and influence zone algorithms for large images, IEEE Transactions on Image Processing,2000;Vol. 9,No. 7.

[10] Pratikakis I.E., Sahli H., Cornalis J., Low level image partitioning guided by the gradient watershed hierarchy, Signal Processing 1999;75:173-195.

[11] Roerdink J.B.T.M., Meijester A., The Watershed Transform: Definitions, Algorithms and Parallelisation Strategies. Fundamenta Informaticae 2001;41:187-228.

[12] Vincent L., Soille P., Watersheds in digital space and efficient algorithm based on immersion simulations, IEEE Transactions on PAMI, 1991, 13, No. 6:583–598.

[13] www.aldec.com

[14] www.alatek.com

[15] www.xilinx.com

Address for correspondence:

Jerzy Kasperek
AGH University of Science and Technology
Al. Mickiewicza 30,
30-059 Kraków, Poland
kasperek@agh.edu.pl