

Bartłomiej ŚWIERCZEK, Jerzy KASPEREK, Paweł J. RAJDA
 AKADEMIA GÓRNICZO-HUTNICZA, KATEDRA ELEKTRONIKI

Implementacja algorytmu szkieletyzacji w układzie FPGA

Dr inż. Bartłomiej ŚWIERCZEK

Ukończył studia na Wydziale Elektrotechniki, Automatyki i Elektroniki i Informatyki w 2006. Obecnie jest pracownikiem firmy Netresearch Sp. z o. o. Jego zainteresowania naukowe to układy FPGA oraz ich zastosowanie w aplikacjach telekomunikacyjnych.



e-mail: bsimpson@poczta.fm

Dr inż. Jerzy KASPEREK

Ukończył studia na Wydziale Elektrotechniki, Automatyki i Elektroniki AGH, obronił pracę doktorską w 2001r. Jest adiunktem w Katedrze Elektroniki. Jego zainteresowania naukowe to sprzętowe przetwarzanie sygnałów, w szczególności obrazów biomedycznych.



e-mail: kasperek@agh.edu.pl

Dr inż. Paweł J. RAJDA

Ukończył studia na Wydziale Elektrotechniki, Automatyki i Elektroniki, obronił pracę doktorską w 2001r. Jest adiunktem w Katedrze Elektroniki. Jego zainteresowania naukowe to rekonfigurowalne systemy obliczeniowe oraz sprzętowe implementacje algorytmów przetwarzania sygnałów cyfrowych.



e-mail: p.rajda@agh.edu.pl

W literaturze znanych jest kilka podstawowych technik wyznaczania szkieletu danej figury: wyznaczanie grzbietów mapy odległości (*ang. distance map*) punktów brzegowych [11]; obliczanie diagramu Voronoi dla punktów brzegowych figury [9, 16] i wielokrotną sterowaną erozję figury - ścienianie. W niniejszym opracowaniu wykorzystano to ostatnie podejście – morfologiczny algorytm ścieniania.

Motywacją do podjęcia sprzętowej implementacji algorytmu szkieletyzacji był zamiar zbudowania systemu wizyjnego realizującego detekcję lewej komory serca na podstawie wyniku badania echokardiograficznego [1, 4, 10].

Streszczenie

Artykuł przedstawia sprzętową implementację algorytmu szkieletyzacji w układzie FPGA Virtex II firmy Xilinx. Szkieletyzacja jest operacją mającą na celu wyodrębnić osiowe punkty – szkielety figur w analizowanym obszarze – który można zdefiniować jako zbiór wszystkich punktów równoodległych od co najmniej dwóch punktów należących do jej brzegu. Operacja szkieletyzacji jest wykorzystywana w systemie wizyjnym realizującym detekcję granic lewej komory serca na podstawie wyników badania echokardiograficznego, do wyznaczania bazy i osi głównej lewej komory. W artykule przedstawiano szczegóły implementacji, wnioski z przeprowadzonych prac a także możliwości rozwiązań zrównoleglenia algorytmu.

Słowa kluczowe: szkieletyzacja, przetwarzanie obrazów, echokardiografia, układy rekonfigurowane.

Skeletonization hardware implementation in FPGA device

Abstract

The proposed paper presents the implementation of the image skeletonization algorithm in a high capacity Virtex II FPGA device. Presented work is a part of the image processing system dedicated to left ventricle parameters detection based on echocardiographic image data. Skeletonization is a transformation of a component of a digital image into a subset of the original component. This paper refers skeletonization algorithm defined by thinning approaches. Motivation for interest in skeletonization algorithm computation is the need to simplify the echocardiographic image shape in order to find the left ventricle main axis for further quantitative analysis – Fig. 3. Module hardware implementation details, achieved timing parameters as well as speed enhancement possibilities are discussed in the proposed text.

Keywords: skeletonisation, image processing, echocardiography, programmable devices

1. Wprowadzenie

Szkieletyzacja jest operacją z dziedziny przetwarzania obrazów, mającą na celu wyodrębnienie osiowych punktów figur – szkieletu, co umożliwia znaczną redukcję rozmiarów badanych obiektów, przy równoczesnym zachowaniu ich właściwości topologicznych.

2. Algorytm szkieletyzacji

Przekształcenia morfologiczne [8] możemy definiować dla obrazów w dowolnych modelach barw, tu skoncentrowano się na omówieniu ich działania na obrazie binarnym, w którym piksele tła mają wartość '0', a piksele obiektów wartość '1'. Jednym z podstawowych przekształceń morfologii matematycznej jest erozja. Za [14] operację erozji można zdefiniować jako operację na zbiorach: zakładamy istnienie pewnego obszaru X i koła K o promieniu r, który ma być elementem strukturalnym. Punktem środkowym elementu strukturalnego jest środek koła K.

- Erozja obszaru X jest to zbiór wszystkich kół o promieniu r, które w całości mieszczą się we wnętrzu obszaru X;
 - Przetaczając koło K po wewnętrznej stronie brzegu figury. Kolejne położenia środka koła K wyznaczają brzeg figury poddawanej operacji erozji;
- Przyjmujemy za A zbiór punktów obrazu należących do badanej figury, za B zbiór punktów należących do elementu strukturalnego. Operację erozji można zapisać:

$$A \ominus B = \{b \mid B_b \subset A\};$$

Erozja odpowiada na pytanie „na jakich pozycjach elementu B uzyskanego w wyniku translacji element B zawiera się w obszarze A”.

Wielokrotne użycie operatora erozji można wykorzystać do operacji ścieniania mającej na celu wyznaczenie brzegów określonych obszarów [14]. Do realizacji tego przekształcenia można zastosować wiele elementów strukturalnych - kilka z nich przedstawionych jest na rys.1. Elementy te, jak widać z ich konstrukcji, cechują się wyraźną kierunkowością.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| x | 1 | x | 0 | 1 | x | x | x | x | x | x | x |
| x | 1 | x | 1 | 1 | x | 1 | 1 | 0 | 0 | 1 | 1 |
| x | 0 | x | x | x | x | x | 1 | x | x | x | x |

Rys. 1. Przykładowe elementy strukturalne stosowane w operacji ścieniania
 Fig. 1. Structural elements patterns used in thinning operation

Wszystkie cechuje ta sama zasada polegająca na przykładaniu elementu kolejno do każdego punktu obrazu i sprawdzaniu, czy lokalizacja tego punktu zgadza się z rozkładem punktów w elemencie strukturalnym. I tu możliwe są dwie decyzje:

- niezmiennianiu wartości punktu, jeżeli otoczenie punktu zgadza się z układem elementu strukturalnego;
- zmianie wartości punktu na 0, jeżeli otoczenie punktu nie zgadza się z układem elementu strukturalnego;

Operacja ścieniania może być wykonywana wielokrotnie aż do momentu, gdy następny krok nie wprowadza żadnych zmian w obrazie. Najczęściej element strukturalny jest przekręcany o kąt 90° w przypadku siatki kwadratowej. Wynikiem ścieniania jest zawsze obraz binarny, obrazem wejściowym jest zazwyczaj również obraz binarny, chociaż nie jest to konieczne. Cechą ścieniania jest fakt, że figura po wykonaniu tej operacji zawiera się w figurze wyjściowej.

W celu usystematyzowania mnogości elementów strukturalnych wprowadzono tzw. alfabet Golay'a. Wiąże on litery alfabetu rzymskiego z pewnymi klasami elementów strukturalnych. Dokładny opis symboli alfabetu można znaleźć w [2].

Zastosowanie niektórych elementów strukturalnych może powodować niebezpieczeństwo wystąpienia nieciągłości szkieletu w wynikowym obrazie. Wady tej pozbawione są naprzemiennie stosowane elementy strukturalne z rys. 2.

| | | | | | |
|----|---|---|----|---|---|
| 1 | 1 | 1 | x | 0 | 0 |
| x | 1 | x | 1 | 1 | 0 |
| 0 | 0 | 0 | x | 1 | x |
| a) | | | b) | | |

Rys. 2. Elementy strukturalne gwarantujące powstanie ciągłego szkieletu
Fig. 2. Structural elements constructing the connected skeleton

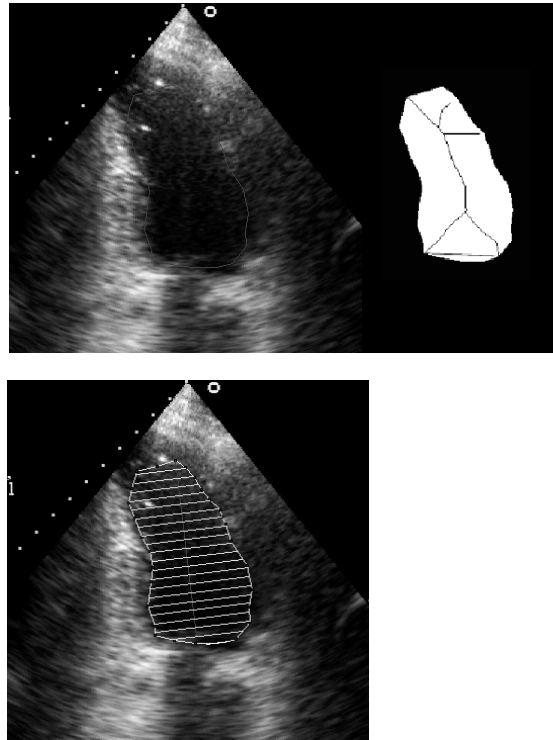
3. Platforma sprzętowa

Algorytm szkieletyzacji został zaimplementowany na platformie sprzętowej ADC-PMC2 firmy Alpha-Data [21]. Głównym elementem karty jest układ FPGA typu XC2V1000 z rodziny Virtex II [18] firmy Xilinx [21]. Karta, wykonana w standardzie PMC (*PCI Mezzanine Card*), w wersji *Lite* wyposażona jest w cztery niezależne banki pamięci ZBT RAM o organizacji 512kBx18 oraz układ interfejsu magistrali PCI, wykorzystujący układ PLX 9080. Dołączone środowisko programowe SDK (*Software Development Kit*) umożliwia realizację sprzętowo-programowego (ang. *hardware-software*) przetwarzania danych.

4. Zastosowanie algorytmu szkieletyzacji w ilościowym opisie lewej komory serca

Operację szkieletyzacji wykorzystano do wyznaczenia osi lewej komory serca przy analizie wyników segmentacji obrazu echokardiograficznego.

Rys. 3 prezentuje obrys rzeczywistej lewej komory serca w projekcji koniuszkowej -dwujamowej po dokonanej segmentacji i efekty wykonanej szkieletyzacji zarysu komory. Lewa część rysunku przedstawia obraz źródłowy z wykonanym obrysem LV, część środkowa efekt szkieletyzacji i wykonaną w najprostszy sposób detekcję „bazy” (jako punkty końcowe wyznaczonego szkieletu) a część prawa rysunku – zarys osi lewej komory (wyznaczonej jako linia łącząca środek „bazy” i najdalej od niej oddległy punkt na granicy komory) i proste wyznaczające poszczególne płaszczyzny ekstrapolujące zarys komory do wyznaczenia jej objętości metodą Simsona.



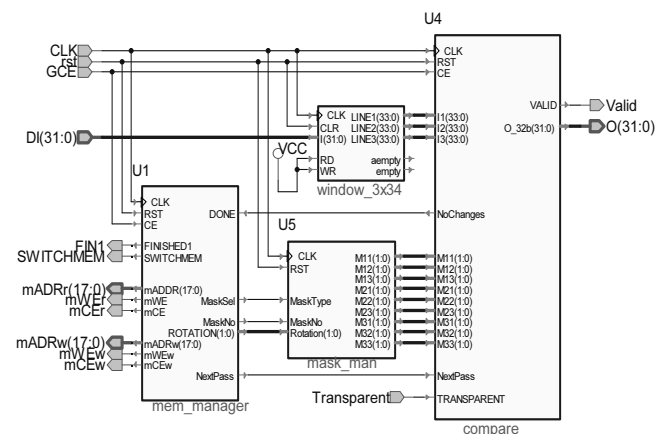
Rys. 3. Zastosowanie algorytmu szkieletyzacji do ilościowej analizy lewej komory serca

Fig. 3. Left ventricle echocardiographic image skeletonisation in quantitative analysis

5. Implementacja sprzętowa

Schemat blokowy jądra modułu realizującego algorytm szkieletyzacji przedstawia rys. 4. W jego skład wchodzi następujące komponenty:

- MEM_MANAGER – zarządzający pracą całego modułu,
- WINDOW 3x34 – generujący okno o wymiarach 3 x 34 bity,
- MASK_MAN – przechowujący i zarządzający elementami strukturalnymi,
- COMPARE – porównujący przetwarzane dane wraz z otoczeniem z elementem strukturalnym.



Rys. 4. Schemat blokowy jądra obliczeniowego modułu szkieletyzacji obrazu
Fig. 4. Image skeletonisation computation unit kernel

Komponent MEM_MANAGER steruje pracą części wykonawczej - generuje adresy i dla pamięci ZBT, oraz sygnały sterujące dla MASK_MAN i COMPARE. Dane wejściowe w postaci słowa 32bitowego (32piksele obrazu) podawane są na wejście układu WINDOW_3x34 generującego otoczenie badanych punktów.

Na jego wyjściach otrzymujemy dane dla bloku COMPARE, w postaci umożliwiającej przeprowadzenie 32 operacji szkieletyzacji z elementem strukturalnym o wymiarach 3x3. Element strukturalny przechowywany jest w MASK_MAN, gdzie następuje również jego obracanie i przełączanie w przypadku bardziej złożonych przekształceń.

Algorytm szkieletyzacji ma charakter iteracyjny. Każde kolejne przekształcenie bazuje na wynikach z poprzedniego kroku, wiąże się to z dużą ilością odwołań do zewnętrznej pamięci, w której przechowywane są dane i wielokrotnym odczytem całego obrazu. W przypadku obrazu o rozmiarach 352x436 podczas jednej iteracji wykonywanych jest po 4796 cykli zapisu i odczytu. Bardzo pomocna okazała się tutaj obecność na płycie prototypowej ADM-XRC-II Lite czterech niezależnych banków pamięci. Zostały one zgrupowane w dwa bloki z niezależnymi szynami adresowymi i danych. Każdy blok w danej iteracji służył tylko do zapisu lub tylko do odczytu, w następnej iteracji bloki były zamieniane. Rozwiązanie z rozdzielonymi strumieniami danych do zapisu i odczytu jest dwukrotnie wydajniejsze niż przypadek klasyczny, z dużą pamięcią i jednym strumieniem danych w postaci wspólnej magistrali adresowej i danych (prawie 9500 odwołań do pamięci, na przemian zapis i odczyt). Podczas analizy dostępnych materiałów nie napotkano na opracowanie jasno mówiące o możliwości dekompozycji obrazu wejściowego i zrównoleglenia obliczeń na mniejszych obrazach składowych. Wydaje się więc, że zwiększenia prędkości działania urządzenia można szukać na następujące sposoby:

- Zastosowanie szybszych pamięci,
- Ograniczenie liczby odwołań do pamięci przez wykorzystanie wewnętrznych zasobów pamięciowych układu FPGA do tymczasowego przetrzymywania danych w czasie działania algorytmu. Pamięci synchroniczne BlockRAM działają znacznie szybciej niż dostępne na płycie prototypowej ZBT SSRAM. Ponadto, można je zorganizować w struktury o szerszej niż 32bity magistrali danych,
- Wprowadzenie wstępnej analizy określającej obszary obrazu, które warto poddać działaniu algorytmu. W prezentowanym urządzeniu analizowane są po kolei wszystkie punkty obrazu niezależnie od tego czy niosą informację (są zapalone) czy nie (zgaszone). Dzięki wstępnej analizie można by ograniczyć działanie algorytmu tylko do istotnych części obrazu.

6. Zakończenie

Algorytm szkieletyzacji realizowany w postaci autonomicznego urządzenia spełnia kryterium pracy w czasie rzeczywistym, czas przetwarzania pojedynczego obrazu (5-18ms) jest mniejszy niż okres, z jakim mogą napływać kolejne klatki w większości prostych systemów wizyjnych USG. Prezentowane moduł nie jest ograniczony tylko i wyłącznie do wykonywania algorytmu szkieletyzacji i obcinania gałęzi. Możliwość zmiany wykonywanego przekształcenia, stosunkowo niewielka liczba wymaganych zasobów oraz uzyskane przyspieszenie pozwalają stwierdzić, że może ono znaleźć również zastosowanie w innych systemach przetwarzania i analizy obrazu.

Zaprojektowany moduł spełnia wymóg skalowalności. Zaleca się aby wymiary przetwarzanego obrazu były definiowane w pakiecie konfiguracyjnym, istnieje również możliwość wprowadzania ich w czasie pracy urządzenia poprzez zapis do rejestrów kontrolno-sterujących. Ze względu na przyjętą organizację danych, szerokość wejściowego obrazu powinna być wielokrotnością 32 pikseli. Domyślne wymiary obrazu wynoszą: 352 x 436 punktów.

Urządzenie zajmuje około 920 elementów *slice*, co stanowi 18% zasobów sprzętowych układu XC2V1000FG456-4. Prędkość wskazywana przez narzędzia implementacji, przy której urządzenie powinno działać prawidłowo wynosi 120MHz i przekracza ona maksymalną prędkość działania pamięci ZBT SSRAM dostępnych na płycie ADM-XRC-II Lite wynoszącą 100MHz. Maksymalna częstotliwość z jaką projekt został uruchomiony wynosiła

50MHz. Planowana częstotliwość pracy, wynikająca z wydajności pamięci wynosi 100MHz. Z przeprowadzonych porównań z czasem wykonywania szkieletyzacji realizowanej w programie Matlab, z wykorzystaniem specjalizowanego zestawu narzędzi z optymalnie zrealizowanymi funkcjami morfologicznymi (SDC Morphology Toolbox for MATLAB firmy SDC Information Systems [12] na komputerze PC z procesorem AthlonXP 1.7GHz i 1024MB wynika, że implementacja algorytmu w układzie Virtex-II pozwala na uzyskanie przynajmniej 10-krotnego przyspieszenia w stosunku do platformy PC podanych parametrach. Kluczem do uzyskania jeszcze większego przyspieszenia wydaje się być również odpowiedni dobór elementów strukturalnych.

Praca naukowa finansowana ze środków Komitetu Badań Naukowych jako projekt badawczy 4 T11E 015 24.

Autrzy pragną podziękować firmie ALDEC za udostępnienie oprogramowania ActiveHDL [19].

7. Literatura

- [1] Flis A., Kasperek J., Rajda P.J., Implementacja filtru gradientu promieniowego w układzie FPGA, Materiały VIII krajowej konferencji RUC 2005 Szczecin 2005.
- [2] Golay M.J.E. Hexagonal Parallel Pattern Transformation", IEEE Transactions on Computers, pp.33-740, August 1969.
- [3] Jianning Xu, A Generalized Discrete Morphological Skeleton Transform With Multiple Structuring Elements for the Extraction of Structural Shape Components, IEEE Transactions on Image Processing, vol. 12, no. 12, December 2003
- [4] Latała Z., Zastosowanie komputerowej analizy obrazu ultrasonograficznego do badania serca, Rozprawa doktorska. Politechnika Krakowska, 2002.
- [5] MARAGOS P., SCHAFFER R., Morphological Skeleton Representation and Coding of Binary Image, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-34, no. 5, October 1986
- [6] Nieniewski M. Morfologia matematyczna w przetwarzaniu obrazów, PLJ, Warszawa 1998.
- [7] Ogniewicz R. and Ilg M., "Voronoi Skeletons: Theory and Applications", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6369,1992
- [8] Patury R., Kasperek J., Rajda P.J., Implementacja algorytmu segmentacji metodą detekcji działów wodnych w układzie FPGA, Pomiar Automatyka Kontrola 7bis 2006 str.3-5 Warszawa 2006.
- [9] Rosenfeld A. and Pfaltz J. Distance Functions in Digital Pictures, Pattern Recognition, Vol 1, 1968, pp 33-61.
- [10] SDC, Morphological Image Processing for MATLAB <http://www.mmorph.com/>
- [11] Tadeusiewicz R. Systemy wizyjne robotów przemysłowych, WNT, Warszawa 1992.
- [12] Tadeusiewicz R., Korohoda P.: Komputerowa analiza i przetwarzanie obrazów, WFPT, Kraków 1997.
- [13] Voronoi G. (1907). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Journal für die Reine und Angewandte Mathematik, 133:97-178, 1907
- [14] Wiatr K., Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego, Wydawnictwa AGH, Kraków 2001.
- [15] Xilinx, Virtex™-II Platform FPGAs: Complete Data Sheet, August 1, 2003 (DS031.pdf)
- [16] Aldec, <http://www.aldec.com>
- [17] Alpha-Data, <http://www.alpha-data.com>
- [18] Xilinx, <http://www.xilinx.com/>