

Paweł WUJEK, Ryszard PEŁKA

WOJSKOWA AKADEMIA TECHNICZNA, WYDZIAŁ ELEKTRONIKI, INSTYTUT TELEKOMUNIKACJI, ZAKŁAD TECHNIKI CYFROWEJ
ul. Gen. Sylwestra Kaliskiego 2, 00-908 Warszawa 49

Implementacja algorytmu detekcji twarzy w obrazach cyfrowych z układem SoC Zynq

Mgr inż. Paweł WUJEK

Ukończył studia magisterskie na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej w 2009 roku. Obecnie jest asystentem w Zakładzie Techniki Cyfrowej w Instytucie Telekomunikacji w WAT. Jego zainteresowania naukowe dotyczą projektowania systemów cyfrowych z układami FPGA i SoC.



e-mail: pwujek@wat.edu.pl

Prof. dr hab. inż. Ryszard PEŁKA

Studiował na Wydziale Elektroniki Politechniki Warszawskiej. Doktor nauk technicznych (1984), doktor habilitowany (1997) i profesor (2004). Profesor zwyczajny na Wydziale Elektroniki w Wojskowej Akademii Technicznej. Kierownik Zakładu Techniki Cyfrowej w Instytucie Telekomunikacji WAT. Specjalizuje się w dziedzinie techniki cyfrowej, w obszarze precyzyjnej metrologii odcinków czasu oraz projektowania, optymalizacji i testowania systemów cyfrowych z układami FPGA i SoC.



e-mail: rpelka@wat.edu.pl

Streszczenie

W artykule przedstawiono koncepcję i projekt mikrosystemu do detekcji twarzy w obrazach cyfrowych z użyciem układu programowalnego SoC z rodziny Zynq firmy Xilinx [1]. Algorytm detekcji twarzy polega na wyodrębnieniu podstawowych cech twarzy i określeniu ich położenia w obrazie. Przedstawiono wyniki implementacji programowej w środowisku MATLAB/PC oraz implementacji sprzętowej. Obie implementacje przebadano pod względem złożoności oraz szybkości działania. W realizacji sprzętowej uzyskano porównywalną szybkość detekcji/lokalizacji twarzy i ponad 10-krotnie krótszy czas wyodrębniania cech twarzy.

Słowa kluczowe: detekcja twarzy, FPGA, SoC.

SoC Zynq-based implementation of a face detection algorithm in digital images

Abstract

In this paper there is presented the design of an integrated microsystem for face detection in digital images, based on a new SoC Zynq from Xilinx [1]. Zynq is a new class of SoCs which combines an industry-standard ARM dual-core Cortex-A9 processing system with 28 nm programmable logic. This processor-centric architecture delivers a comprehensive platform that offers ASIC levels of performance and power consumption, the ease of programmability and the flexibility of a FPGA. The proposed algorithm for face detection operates on images having the resolution of 640x480 pixels and 24-bit color coding. It uses three-stage processing: normalization, face detection/location [2] and feature extraction. We implemented the algorithm in a twofold way: (1) using MATLAB/PC, and (2) hardware platform based on ZedBoard from Avnet [3] with Zynq XC7Z020 SoC. Both implementations were examined in terms of complexity and speed. The hardware implementation achieved a comparable speed of face detection/location but was over 10-times faster while extracting the features of faces in digital images. A significant speedup of feature extraction results from the parallelized architecture of a hardware accelerator for calculation of mouth and eyes locations. The proposed microsystem may be used in low-cost, mobile applications for detection of human faces in digital images. Since the system is equipped with the Linux kernel, it can be easily integrated with other mobile applications, including www services running on handheld terminals with the Android operating system.

Keywords: face detection, FPGA, SoC.

1. Wstęp

Algorytmy służące do automatycznego wykrywania twarzy osób w obrazach cyfrowych są przedmiotem badań w wielu dziedzinach nauki takich jak elektronika, telekomunikacja, informatyka, automatyka i robotyka. Znajdują one zastosowanie w systemach bezpieczeństwa, w zautomatyzowanych systemach przetwarzania informacji multimedialnych (w tym zwłaszcza w serwisach WWW), w projektowaniu zautomatyzowanych systemów kierowania pojazdami, w nowych systemach komunikowania się człowieka z komputerami, a także w sprzęcie powszechnego użytku.

Przeważająca liczba publikacji dotyczących tego zagadnienia skupia się jednak na samej budowie algorytmów, które są zwykle badane z wykorzystaniem stacjonarnych komputerów o stosunkowo dużych mocach obliczeniowych. Przegląd ważniejszych osiągnięć w tej dziedzinie można znaleźć m.in. w pracy [4].

Znacznie mniej prac poświęconych jest problemowi zastosowania algorytmów detekcji twarzy w urządzeniach przenośnych, dysponujących niewielką mocą obliczeniową, mniejszą pamięcią i ograniczoną pojemnością źródła zasilania. Wyniki badań dotyczących komercyjnych projektów z układami ASIC są ponadto publikowane tylko w zarysie, a stosunkowo nieliczne prace poświęcone systemom opartym na układach programowalnych dotyczą starszych rodzin FPGA.

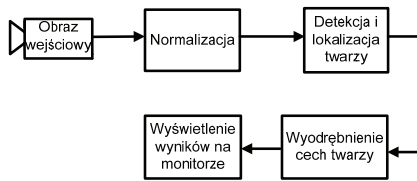
Tymczasem na rynku pojawiła się interesująca, nowa rodzina układów Zynq firmy Xilinx, która daje projektantom większe możliwości w zakresie budowy złożonych mikrosystemów [1]. Układy te zawierają logikę programowalną oraz dwurdzeniowy procesor ARM Cortex-A9 ze sprzętowymi peryferiami. Takie rozwiązanie umożliwia realizację kompletnego systemu mikroprocesorowego w jednym układzie – SoC. Wcześniejsze rodziny układów FPGA firmy Xilinx zawierały sprzętowe rdzeń procesora (PowerPC), konieczna była jednak implementacja układów peryferyjnych w logice programowalnej. Układy o podobnych możliwościach jak Zynq ma od niedawna w swojej ofercie również Altera, która produkuje układy z rodzin Arria V oraz Cyclone V wyposażone w sprzętowe rdzenie procesorów ARM Cortex-A9, określane przez producenta skrótem HPS (*Hard Processor System*).

Zastosowanie płyty uruchomieniowej ZedBoard firmy Avnet z układem Zynq XC7Z020 [3], pozwala na realizację kompletnego mikrosystemu do akwizycji obrazów, ich analizy i prezentacji wyników. Dzięki zastosowaniu procesora ARM możliwe jest uruchomienie systemu operacyjnego Linux, który ułatwia oprogramowanie fragmentów systemu odpowiedzialnych za akwizycję danych z kamery podłączonej przez interfejs USB oraz wyświetlanie danych wyjściowych na ekranie monitora i/lub wbudowanego wyświetlacza LCD.

2. Algorytm lokalizacji cech twarzy

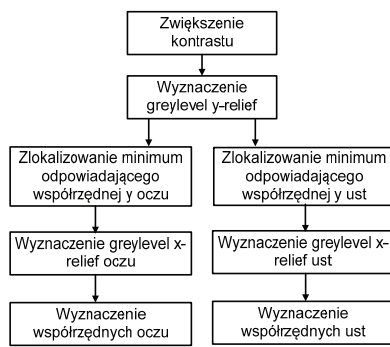
Algorytm lokalizacji cech twarzy zastosowany w opisywanym mikrosystemie odczytuje obraz z pliku lub kamery, poddaje go detekcji i procesowi lokalizacji twarzy, a następnie określa współrzędne cech twarzy. Spośród wielu możliwych cech twarzy analizowanych w literaturze, na podstawie wyników wcześniejszych własnych badań, zdecydowano się wybrać dwie podstawowe – oczy i usta.

Wybrany algorytm operuje na obrazie kolorowym o rozdzielczości 640x480 pikseli z 24-bitowym kodowaniem kolorów. W celu zweryfikowania poprawności działania algorytmu użyte zostało środowisko symulacyjne MATLAB.



Rys. 1. Schemat blokowy systemu detekcji twarzy w obrazach cyfrowych
Fig. 1. Block diagram of the face detection system in digital images

Po odczytaniu obrazu z kamery bądź pliku następuje detekcja i lokalizacja twarzy w obrazie zgodnie z algorytmem opisanym w pracy [2]. Uproszczony schemat blokowy systemu przedstawiono na rys. 1. Obraz z wykrytą twarzą uzyskany z tego bloku zapisywany jest do pliku wyjściowego i przekazywany do kolejnego bloku, w którym następuje zamiana palety kolorów na skalę monochromatyczną i normalizacja wymiarów obrazu do 100x150 pikseli. Następnie w tak przygotowanym obrazie wyszukiwane są współrzędne oczu i ust. Odbywa się to zgodnie z algorytmem przedstawionym na rys. 2.



Rys. 2. Algorytm lokalizacji cech twarzy
Fig. 2. The algorithm for location of features

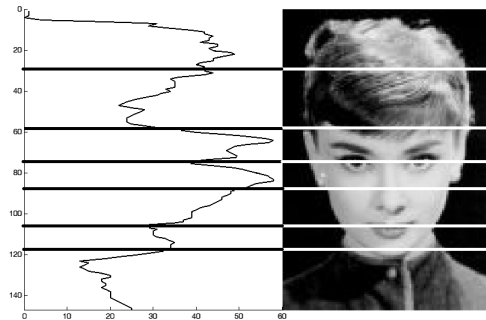
Zwiększenie kontrastu odbywa się z użyciem maski o rozmiarze 3x3, w obrębie której wyszukiwana jest największa (max) oraz najmniejsza (min) wartość jasności piksela $f(x,y)$. Następnie dla danego piksela obliczana jest nowa wartość jasności $g(x,y)$ według formuły podanej w pracy [5]:

$$g(x,y) = \begin{cases} min & \text{gdy } f(x,y) - min < max - f(x,y) \\ max & \text{w przeciwnym przypadku} \end{cases} \quad (1)$$

gdzie:

$f(x,y)$ – luminancja wejściowa piksela
 $g(x,y)$ – luminancja wyjściowa
 min, max – minimalna i maksymalna wartość luminancji w obrębie maski o rozmiarze 3x3 pikseli.

Do wyznaczenia współrzędnych oczu i ust można posłużyć się charakterystyką *greylevel relief* [5]. Wyznaczenie charakterystyki w kierunku poziomym *y-relief* polega na obliczeniu średniej wartości poziomu szarości w każdym wierszu, a następnie zaprezentowaniu tych wyników w kierunku pionowym. Charakterystyka ta umożliwia znalezienie współrzędnych y oczu i ust. Zgodnie z [6] obszar twarzy można podzielić na siatkę o wymiarach 4x5. Oczy powinny znajdować się w obszarze o współrzędnych 3,2 i 3,3, natomiast usta w obszarze o współrzędnych 4,2 i 4,3. Oznacza to, że tylko w tych obszarach należy przeszukiwać charakterystykę *y-relief* w celu znalezienia współrzędnych oczu. Zastosowanie wyżej wymienionych przedziałów znacząco skraca czas niezbędny do znalezienia współrzędnych cech. Przykładową charakterystykę *y-relief* zaznaczonymi obszarami potencjalnego występowania oczu i ust oraz obraz, dla którego została wykonana przedstawiono na rys. 3.



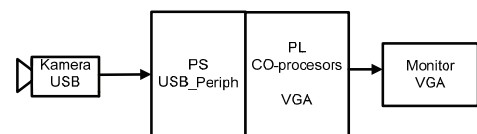
Rys. 3. Charakterystyka *y-relief* przykładowego obrazu twarzy
Fig. 3. *Y-relief* characteristic of exemplary face image

Wyszukiwanie współrzędnych y oczu i ust polega na znalezieniu lokalnego minimum funkcji *y-relief* w przeszukiwanym obszarze. Po znalezieniu współrzędnych y oczu i ust należy znaleźć współrzędne x . Są one wyznaczone z charakterystyki *x-relief* uzyskanej przez obliczenie średniej wartości luminancji pikseli w sąsiedztwie wyznaczonej współrzędnej y . Analogicznie jak poprzednio, algorytm wyszukuje lokalne minima na charakterystyce *x-relief* w odpowiednich przedziałach.

Na koniec, na obrazie wejściowym zaznaczane są współrzędne cech twarzy. Współrzędne oraz ich wzajemne relacje zapisywane są w bazie danych.

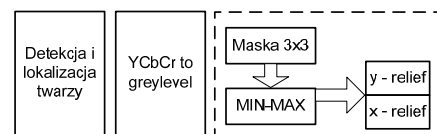
3. Implementacja w układzie SoC Zynq

Do implementacji algorytmu detekcji i lokalizacji twarzy oraz lokalizacji cech twarzy w obrazie cyfrowym użyto mikrosystemu SoC, na który składa się płyta uruchomieniowa ZedBoard firmy Avnet z układem Zynq XC7Z020. Schemat blokowy mikrosystemu przedstawiono na rys. 4.



Rys. 4. Schemat blokowy mikrosystemu
Fig. 4. Block diagram of the microsystem

Na rdzeniach procesora ARM uruchomiony został system operacyjny Linux, który umożliwia pobieranie obrazu z kamery, sterowanie całym mikrosystemem oraz wyświetlanie wyników na ekranie monitora. W logice programowalnej zaimplementowano bloki odpowiadające za lokalizację twarzy w obrazie, zmianę kodowania braw z YCbCr na skalę monochromatyczną oraz blok wyznaczający współrzędne cech w obrazie (rys. 5).

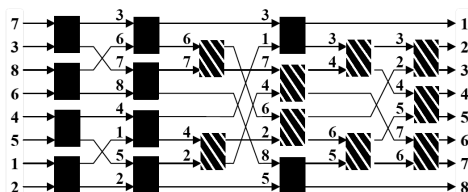


Rys. 5. Bloki mikrosystemu zaimplementowane w logice programowalnej
Fig. 5. Blocks implemented in programmable logic

Wszystkie bloki zrealizowane w logice programowalnej działają na zasadzie koprocesora uruchamianego w zależności od potrzeb. Dane między procesorem a logiką programowalną przekazywane są przez pamięć RAM. Rejestry wejściowe i wyjściowe logiki programowalnej odwzorowane są w pamięci operacyjnej procesora.

Blok detekcji i lokalizacji zrealizowano w sposób opisany w [2]. Blok poprawy kontrastu działa zgodnie z zależnością (1). Maska 3x3 została zbudowana, podobnie jak w [2], z użyciem linii opóźniających (rejestrów przesuwanych). Do wyznaczenia minimalnej i maksymalnej wartości luminancji w masce użyto ośmiowiejskiej sieci sortującej o strukturze pokazanej na rys. 6

[7]. Na podstawie badań stwierdzono, że środkowy piksel maski ma bardzo mało znaczący wpływ na efekt końcowy, dlatego został pominięty.

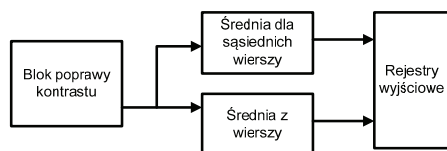


Rys. 6. Sieć sortująca (liczby oznaczają przykładowe wartości sortowanych elementów)

Fig. 6. Sorting network (numbers denote example values of sorted elements)

Prostokąty oznaczają dwuwejściowe ośmiobitowe układy porównujące, które na „górnym” wyjściu dają mniejszą, a na „dolnym” większą z dwóch porównywanych liczb. W projekcie zastosowano tylko bloki całkowicie wypełnione czarnym tłem. Pozostałe bloki byłyby niezbędne w przypadku konieczności pełnego posortowania wszystkich danych wejściowych.

Średnia luminancja w każdym wierszu jest obliczana z użyciem sumatora i rejestrów, w których zapisywana jest wartość dla każdego wiersza. Jeżeli wartość ta zawiera się w odpowiednim przedziale, wyszukiwana jest lokalna wartość maksymalna (minimalna), która następnie jest zapisywana w rejestrze wyjściowym. W tym samym czasie obliczana jest wartość średnia dla 4 sąsiednich wierszy i zapisywana do pamięci (wartości te służą następnie do określenia współrzędnych x cech). Na rys. 7 przedstawiono wzajemne relacje między blokami.



Rys. 7. Układ do równoległego przetwarzania luminancji w wierszach obrazu

Fig. 7. Parallel processing of luminance in rows of pixels

4. Wyniki badań eksperymentalnych

Przetestowano szybkość działania systemu oraz określono liczbę komórek zajmowanych przez poszczególne bloki zaimplementowane w układzie Zynq. Sprawdzenie poprawności działania algorytmu polegało na porównaniu, czy wyniki uzyskane z wersji programowej (MATLAB) i sprzętowej są identyczne, tzn. czy obliczone współrzędne oczu i ust są takie same. Zgodnie z przewidywaniami, efekty działania obydwu algorytmów są w pełni zgodne. Na rys. 8 przedstawiono przykładowy obraz twarzy z zaznaczonymi współrzędnymi badanych cech.

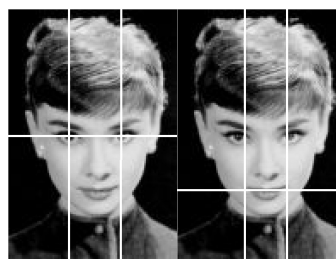
W tabeli 1 przedstawiono liczbę rejestrów oraz komórek LUT użytych przez poszczególne fragmenty algorytmu (przedstawione zostały tylko najważniejsze z nich).

Tab. 1. Wykorzystanie zasobów układu Zynq

Tab. 1. Utilization of Zynq resources

Blok	Rejestry	LUT
deteckcja i lokalizacja twarzy	115	409
poprawa kontrastu	255	254
sieć sortująca	112	138
y -relief	23	16
współrzędna x	70	50

Stosunkowo mała liczba zajętych rejestrów i relatywnie duża liczba wykorzystanych komórek LUT wynika z tego, że w projekcie użyto rejestry przesuwne, do realizacji których użyte zostały 6-wejściowe komórki LUT. W tablicy 2 podano czasy pracy algorytmu dla symulacji komputerowej oraz realizacji sprzętowej.



Rys. 8. Przykładowy obraz twarzy z zaznaczonymi cechami

Fig. 8. Sample image of the face with marked features

Tab. 2. Porównania czasów obliczeń dla implementacji MATLAB i SoC

Tab. 2. Comparison of processing time for MATLAB and SoC implementations

Fragment	MATLAB	SoC
deteckcja i lokalizacja	120 ms	200 ms
lokalizacja cech	340 ms	30 ms

Uzyskane wyniki potwierdzają, że zaprojektowany mikrosystem może konkurować pod względem szybkości działania z komputerem stacjonarnym. Czasy pracy uzyskane przy użyciu programu MATLAB mają na celu pokazanie szacunkowego czasu pracy algorytmu. Możliwe są, oczywiście sposoby jego skrócenia, np. z wykorzystaniem metod GPGPU [8] lub języka C/Asembler.

5. Podsumowanie

Wyniki badań eksperymentalnych potwierdziły, że w układzie SoC Zynq można zaprojektować kompletny mikrosystem do detekcji i lokalizacji twarzy w obrazach cyfrowych, którego szybkość działania jest porównywalna, a nawet lepsza niż w przypadku programowej realizacji algorytmu na komputerze stacjonarnym.

Dzięki zastosowaniu rdzeni procesora ARM możliwe jest użycie systemu operacyjnego oraz gotowych modułów obsługujących peryferia, co znacząco odciąża projektanta mikrosystemu. Zastosowanie sprzętowych układów peryferyjnych procesora ARM pozwala ograniczyć zajętość zasobów układu programowalnego.

Ważną zaletą opisanego mikrosystemu jest jego niewielki rozmiar, mały pobór mocy i niska cena, co ma istotne znaczenie w aplikacjach mobilnych.

6. Literatura

- [1] Xilinx: Zynq-7000 All Programmable SoC Overview, DS 190, v. 1.2, 21 August, 2012, on-line: <http://www.xilinx.com>.
- [2] Wujek P., Pelka R.: Układ SoC – FPGA do detekcji twarzy w obrazach cyfrowych, PAK, vol. 57, nr 8, 2011, ss. 889-891.
- [3] ZedBoard: Zynq™ Evaluation and Development - Hardware User's Guide", version 1.9, 29 Jan. 2013, on-line: www.zedboard.org.
- [4] Zhang C. Zhang Z.: A Survey of Recent Advances in Face Detection, Technical Report, MSR-TR-2010-66, Microsoft Corp. June 2010.
- [5] Sobottka K., Pitas I.: Extraction of Facial Regions and Features Using Color and Shape Information, Proc. of the 13th Int. Conf. on Pattern Recognition, vol. 3, 1996, pp. 421-425.
- [6] Seunghwan Yoo, Rae-Hong Park: Red-eye detection and correction using inpainting in digital photographs, IEEE Transactions on Consumer Electronics, Vol. 55, No. 3, August 2009, pp. 1006-1014.
- [7] Knuth D.E.: The Art of Computer Programming, Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1997. Section 5.3.4: Networks for Sorting, pp. 219-247.
- [8] Wujek P., Pelka R.: Analiza efektywności sprzętowych implementacji algorytmów detekcji twarzy w obrazach cyfrowych, PAK, vol. 58, nr 7, 2012, ss. 581-583.

otrzymano / received: 20.05.2013

przyjęto do druku / accepted: 03.07.2013

artykuł recenzowany / revised paper