

mgr inż. Marek Kraft
 Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej

PERSPEKTYWY WYKORZYSTANIA REPROGRAMOWALNYCH UKŁADÓW CYFROWYCH W SYSTEMACH WIZYJNYCH ROBOTÓW MOBILNYCH

Rekonfigurowalne układy cyfrowe (głównie w formie układów FPGA) rozpowszechniają się w aplikacjach wymagających rozwiązywania problemów z dziedziny przetwarzania sygnałów, wizji komputerowej i innych. Możliwość wykonywania w sposób naturalny obliczeń w trybie równoległym, a także rekonfigurowalność i modułowość, umożliwiającą rozwiązywanie licznych problemów przez procesory zawarte w strukturze pojedynczego układu scalonego, nierzadko umożliwia wielokrotne przyspieszenie wykonywania obliczeń w stosunku do implementacji programowych, wykorzystujących jako platformę komputery klasy PC. Dodatkowo, zastosowanie układów FPGA może umożliwić zmniejszenie wymiarów, wagi i poboru prądu przez wykorzystujące taką platformę obliczeniową urządzenie.

Niniejszy artykuł zawiera analizę przydatności układów FPGA do zastosowań w systemach wizyjnych robotów mobilnych. Analizę poparto licznymi przykładami aplikacji, w których z powodzeniem wykorzystuje się układy reprogramowalne, a także przykładami doświadczeń autora w dziedzinie opracowywania takich aplikacji. Wnioski z analizy zawarto w podsumowaniu, wraz z sugestiami scenariuszy użycia układów FPGA w robotach mobilnych jako głównej platformy obliczeniowej, lub wspomagającego komputer pokładowy koprocatora.

THE PROSPECTS OF USING REPROGRAMMABLE DIGITAL CIRCUITS IN MOBILE ROBOT VISION SYSTEMS

Reconfigurable digital circuits (mainly in the form of FPGAs) are becoming increasingly popular in signal processing, computer vision and many other applications. Their natural ability to perform parallel computations, along with the reconfigurability and modularity often allow to increase the performance significantly, when compared to standard software implementations, using a standard PC as a platform. Additionally, the use of FPGA can allow to reduce the size, weight and power consumption of a complete system.

The following paper contains the analysis of usefulness of FPGA circuits as the computing platform in mobile robot vision systems. The analysis is backed up by numerous examples of applications, including author's experiences with using FPGAs as a part of computer vision system. The conclusions drawn from the analysis, along with suggestions for using FPGAs in robot vision systems (as a main hardware platform or a coprocessor) are given in the summary.

1. WSTĘP

Zmysł wzroku jest dla człowieka najważniejszym źródłem informacji o świecie zewnętrznym. Obraz jest bogatym źródłem treści, dostarczającym wielu danych o obserwowanych scenach i obiektach. Duża zawartość informacyjna powoduje również, że przetworzenie obrazu i wydobywanie z niego informacji istotnej dla użytkownika, czy realizacji zadania, jest

problemem nietrywialnym i bogatym w zagadnienia badawcze. Wizja komputerowa i przetwarzanie obrazów są w związku z tym przedmiotem badań wielu zespołów naukowych, co doprowadziło do szybkiego rozwoju w tej dziedzinie w ciągu ostatnich lat. Za rozwojem algorytmów i metod dla wizji komputerowej podążył również rozwój platform sprzętowych, których potencjalna wydajność jest współcześnie wystarczająca dla rozwiązywania złożonych problemów w tej dziedzinie, umożliwiając rozpowszechnienie się systemów wizji komputerowej w różnorodnych aplikacjach, w tym uwzględniających dynamikę pola widzenia [1], [2]. Dodatkowym czynnikiem, który decyduje o atrakcyjności wizji komputerowej jest relatywnie niska cena i niezawodność czujnika (kamery).

2. UKŁADY FPGA W SYSTEMACH WIZYJNYCH – ZALETY I WADY

Zakres zastosowań wizji maszynowej obejmuje szerokie spektrum problemów: od zastosowań przemysłowych, przez robotykę, zastosowania militarne, sztukę, monitoring i bezpieczeństwo, aż po interakcję człowiek-maszyna. Wśród wymienionych zastosowań, jednym z najbardziej wymagających wydaje się być robotyka mobilna. Pojazdy autonomiczne bardzo często wykorzystują systemy wizji komputerowej, np. do nawigacji. Wiąże się to z potrzebą wykrywania przeszkód, samolokalizacji, rekonstrukcji oraz wykrywania charakterystycznych obiektów. Pomimo postępu, jaki dokonał się w tej dziedzinie, aplikacje systemów wizyjnych oparte tradycyjnie o komputery klasy PC nadal napotykać na bariery w postaci ograniczeń energetycznych, wagowych i gabarytowych. W związku z powyższym, wysoce atrakcyjnym wydaje się rozwiązanie alternatywne, wykorzystujące inne założenia dla rozwiązania podobnych problemów.

Rozwiązania alternatywne, bazujące na przetwarzaniu równoległym strumieni danych (w miejsce szybkiego, szeregowego ich przetwarzania) oparte są najczęściej o dedykowane karty graficzne komputerów PC lub cyfrowe układy programowalne. Niestety, karty graficzne, choć atrakcyjne ze względu na dużą wydajność obliczeniową, zużywają zbyt dużo energii, aby mogły być atrakcyjne z punktu widzenia systemów mobilnych. Dodatkowym ograniczeniem jest fakt, iż nie mogą one w zasadzie pracować samodzielnie, bez komputera PC. W związku z powyższym, bardziej atrakcyjne dla zastosowań w robotyce mobilnej wydają się być dowolnie programowalne układy FPGA (ang. *Field Programmable Gate Array*). Oferują one dużą dowolność w zakresie kształtowania struktury potoku obliczeniowego, a przy odpowiednio zaplanowanej architekturze procesora umożliwiają przetwarzanie równoległe. Zalety te okupione są niestety nieco niższą (niż w przypadku kart graficznych) maksymalną częstotliwością taktowania – do czasów propagacji elementów tworzących architekturę obliczeniową dodać należy czasy propagacji programowalnych połączeń.

W tym miejscu nadmienić należy również, że zastosowanie układów FPGA nie jest panaceum, rozwiązującym wszelkie problemy z niedostatkami mocy obliczeniowej. Nakład pracy przy prototypowaniu sprzętowych implementacji algorytmów jest znacząco większy, niż w przypadku prototypowania tych samych algorytmów w sposób czysto programowy. Dodatkowo, specyfika pewnych algorytmów (np. złożonych algorytmów rekurencyjnych, czy algorytmów zawierających liczne instrukcje warunkowe) powoduje, że implementacja ich w sprzęcie nie daje korzyści, lub wręcz powoduje spowolnienie wykonywania. Algorytmy takie zwykle nie poddają się one w prosty sposób zrównolegleniu, bądź też wymagają niezwykle złożonej struktury sterującej. Pewnym rozwiązaniem tych problemów wydaje się być stosowanie zaimplementowanych w strukturze układów FPGA procesorów

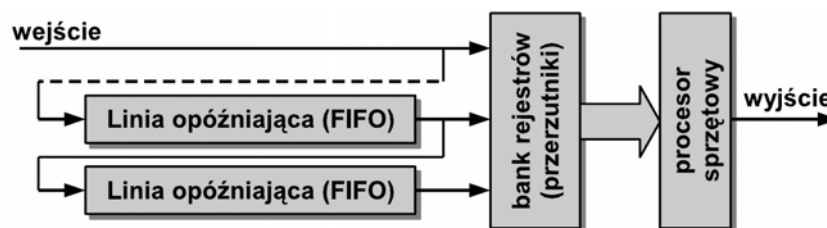
sekwencyjnych. Ich prędkość w porównaniu do stosowanych aktualnie w komputerach PC procesorów jest znacząco niższa, lecz współczesne układy FPGA udostępniają ilości zasobów umożliwiające pomieszczenie w ich strukturze nawet kilkudziesięciu takich jednostek obliczeniowych.

3. UKŁADY FPGA W SYSTEMACH WIZYJNYCH – PRZYKŁADY ZASTOSOWAŃ

Wizja komputerowa i przetwarzanie obrazów są przedmiotem intensywnych badań wielu licznych zespołów naukowych, co doprowadziło do znaczącego rozwoju w tych dziedzinach na przestrzeni ostatnich lat. Zdecydowana większość konstruowanych systemów wizji komputerowej oparta jest jednak na standardowych komputerach klasy PC. Rozwiązania alternatywne w postaci dedykowanych układów sprzętowych, wykorzystujących układy FPGA są wykorzystywane zdecydowanie rzadziej – jest to platforma trudniejsza do opanowania, o zupełnie innej filozofii działania. Korzyści z zastosowania układów FPGA są jednak wymierne – są to układy rekonfigurowalne, umożliwiające wprowadzanie zmian już na poziomie sprzętowym (nie tylko programowym), a dodatkowo w naturalny sposób przeznaczone są one do prowadzenia obliczeń równoległych, co umożliwia znaczące poprawienie wydajności obliczeniowej [3], [1], [2]. Większość rozwiązań z dziedziny robotyki przygotowywanych dla platform FPGA ma na celu jedynie weryfikację koncepcji (ang. *proof of concept*), a rozwiązania realizujące operacje bardziej zaawansowane niż wstępne przetwarzanie obrazu czy detekcja cech są stosunkowo nieliczne. Przedstawione poniżej przykłady systemów wizyjnych, w których cyfrowe układy programowalne zostały z powodzeniem zastosowane potwierdzają użyteczność tych układów w omawianej klasie aplikacji.

3.1. Wstępne przetwarzanie obrazu i filtracja

Operacje wstępnego przetwarzania obrazu, pomimo niewielkiej złożoności, często mają znaczący wpływ na jakość działania funkcji wyższego rzędu systemu wizyjnego. Operacje te wykonywane są na znaczącej ilości danych (najczęściej dla całego obrazu) – fakt ten ma decydujące znaczenie dla szybkości przetwarzania. Stosunkowa prostota operacji wstępnego przetwarzania obrazu (jak np. wyrównanie histogramu, korekcja barwna, adaptacyjne progowanie czy filtracja), a także możliwość ich dostosowania do równoległego przetwarzania powodują, że są to najczęściej implementowane w architekturze FPGA algorytmy. Przy użyciu stosunkowo niewielu zasobów i prostej architektury, opartej o zawarte w strukturze pamięci dwuportowe tworzące bufor FIFO, uzyskać można sporą wydajność, dzięki równoległemu dostępowi do pewnego wycinka obrazu (rys. 1).



Rys. 1. Architektura oparta na buforach FIFO, zapewniająca jednoczesny dostęp do wycinka obrazu

Systemy takie umożliwiają przetwarzanie z prędkościami znacznie przewyższające prędkości uzyskiwane przy użyciu komputerów klasy PC (przyspieszenie nawet do kilkudziesięciu razy). Przykładowe implementacje różnego rodzaju operacji filtrowania (filtry oparte o konwolucję, filtry medianowe) znaleźć można w pracach [4], [2] czy [5]. Przykłady

implementacji w FPGA innych operacji z zakresu wstępnego przetwarzania obrazu znaleźć można w pracach [6] i [3].

3.2. Detekcja cech

Operacja detekcji jest zwykle podstawą dla działania algorytmów wyższego rzędu, które zaimplementowane są na robocie. Jako przykład wymienić można algorytmy samolokalizacji, wizyjnej odometrii, śledzenia i wiele innych. Najczęściej wykorzystywanymi cechami są cechy punktowe (narożniki) oraz krawędzie i linie proste. W dziedzinie detekcji cech (zwłaszcza punktowych) obserwuje się stały postęp, mający na celu opracowanie detektora o dużej dokładności i powtarzalności. Nieodłączną częścią prac na detektorami cech jest opracowanie metod niezawodnego dopasowywania cechy widocznej na kilku obrazach. Tradycyjne metody oparte na sprawdzaniu wzajemnej korelacji wycinków obrazu (lub ich odmiany) wypierane są powoli przez metody wykorzystujące różnego rodzaju lokalne deskryptory. Przykłady nowoczesnych detektorów cech punktowych opisano w pracach [7], [8], [9].

Ponieważ algorytmy detekcji cech są w znakomitej większości algorytmami lokalnymi (obszar ich działania obejmuje jednorazowo rozpatrywany punkt i jego bezpośrednie otoczenie), możliwe jest wykorzystanie opisanej wcześniej architektury realizującej operację filtracji (lub podobnej).

Implementacje sprzętowe bazują zwykle na sprawdzonych, klasycznych algorytmach. Dla przykładu, system śledzenia opisany w pracy [10] wykorzystuje dobrze znany algorytm [11] do wykrywania cech punktowych. System wizyjny robota opisany w pracy [12] wykorzystuje z kolei klasyczny algorytm Harrisa [13]. Zdecydowanie mniej liczne są prace prowadzone nad implementacją bardziej nowoczesnych, odpornych algorytmów detekcji i dopasowywania cech. Praca [14] zawiera opis implementacji algorytmu Harrisa w wersji wieloskalowej, bardziej odpornej na zmiany kąta widzenia. W pracy [15] zawarto z kolei opis implementacji pewnych składowych algorytmu SIFT [7], jednakże obejmują one jedynie około 25% kompletnego algorytmu i były testowane jedynie w symulacji. Obszar badań obejmujący implementacje sprzętowe odpornych, wieloskalowych i opartych na deskryptorach algorytmów detekcji i dopasowywania cech wydaje się być wysoce atrakcyjny. W dziedzinie tej dominują próby przeniesienia istniejących algorytmów wprost z implementacji programowych, co stwarza pewne trudności. Znaczące polepszenie uzyskiwanych wyników (w sensie zwiększenia szybkości działania, a także zmniejszenia ilości zużywanych zasobów) można zdaniem autora uzyskać poprzez opracowanie nowych, oryginalnych algorytmów, dopasowanych do architektury FPGA.

Prace nad implementacjami w układach FPGA algorytmów wykrywania krawędzi oparte są zwykle na klasycznym algorytmie Canny'ego [16]. W pracy [17] opisano implementację zmodyfikowanej wersji tego algorytmu. W kolejnych etapach przetwarzania z wykrytych krawędzi ekstrahowane są zwykle linie proste. Wykorzystywany jest do tego najczęściej algorytm Hough'a. Istnieje wiele prac opisujących architekturę procesora realizującego ten algorytm lub jego modyfikacje – jako przykład podać można [18].

Uzyskiwane przyspieszenie (w stosunku do implementacji na średniej klasy komputerze PC) jest różne dla różnych algorytmów i wynosi od kilkudziesięciu razy dla operacji detekcji cech do kilkuset razy dla operacji wyznaczania prostych za pomocą transformaty Hough'a.

3.3. Kompletne aplikacje

Kompletne aplikacje systemów wizyjnych, obejmujących wszystkie etapy przetwarzania zawarte w całości w architekturze FPGA są znacząco mniej liczne. Obejmują one głównie rekonstrukcję głębi za pomocą stereowizji, zadania śledzenia, jak również specyficzne aplikacje specjalne.

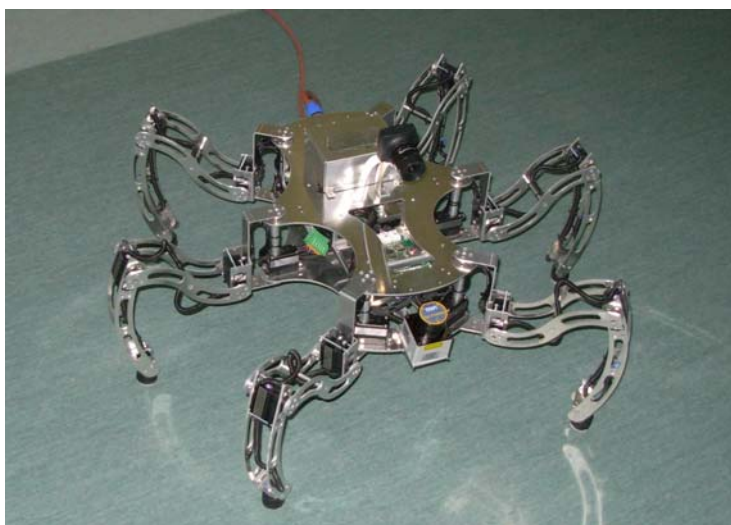
Zagadnienie generowania trójwymiarowej mapy otoczenia z wykorzystaniem metod stereowizyjnych jest interesujące ze względu na możliwość wykorzystania danych uzyskanych w wyniku działania algorytmu stereowizyjnego w wielu obszarach, między innymi w robotyce [19]. Zastosowanie układów FPGA umożliwia skonstruowanie wysoce zintegrowanego systemu stereowizyjnego o bardzo dobrych parametrach użytkowych. Dla przykładu, trzykamerowy system opisany w pracy [20] przetwarza z prędkością minimum 25 klatek na sekundę obrazy o rozdzielczości 320×240 pikseli, tworząc 32-poziomą mapę głębi i mapę otoczenia dla celów nawigacji. System wykorzystuje pojedynczy układ z rodziny Virtex II Pro. W pracy [21] opisano system oparty na czterech układach FPGA z rodziny Stratix, umożliwiający tworzenie map głębi z obrazów o rozdzielczości VGA. Kosztem zwiększenia kosztu i wymiarów systemu (poprzez użycie większej ilości jednostek obliczeniowych) uzyskano rozdzielczość 70 poziomów głębi przy prędkości przekraczającej 100 klatek na sekundę. Należy nadmienić, że akceptowalną dla wielu aplikacji wydajność i jakość przetwarzania systemu stereowizyjnego można uzyskać również przy wykorzystaniu tańszych układów FPGA. W pracy [22] opisano oparty na układzie Spartan 3 system stereowizyjny o dużych możliwościach w zakresie parametryzacji. Użytkownik może podejmować decyzje np. o ilości rozróżnianych poziomów głębi, rozmiarze okna, w którym liczona jest korelacja, aby uzyskać taki system, który przy założonym użyciu zasobów będzie zapewniał akceptowalną jakość danych wyjściowych. Prędkość wynosi w tym przypadku nawet do 300 klatek na sekundę dla obrazu o rozdzielczości 320×240 pikseli przy 20 poziomach głębi i oknie 7×7 pikseli.

Innym atrakcyjnym polem zastosowań układów FPGA jako części systemów wizyjnych dla robotów mobilnych są aplikacje związane ze śledzeniem. Nie istnieje w tej dziedzinie rozwiązanie uniwersalne, zdecydowanie przewyższające pozostałe we wszystkich możliwych scenariuszach zastosowań, w związku z czym wykorzystuje się wiele różnych algorytmów śledzenia. Podobnie zróżnicowane są istniejące rozwiązania sprzętowe. W pracy [23] opisano oparty o pojedynczy układ FPGA system realizujący wizyjne sprzężenie zwrotne z częstotliwością do 1000 Hz na obrazie o rozdzielczości megapikselowej. Implementacja ta oparta jest o filtry korelacyjne wykorzystujące prostą i odwrotną transformatę Fouriera. Z kolei system opisany w pracy [24] wykorzystuje pojedynczy układ FPGA do śledzenia kilku celów. Wykorzystuje on algorytm filtru cząsteczkowego i dla obrazów o rozdzielczości VGA umożliwia pracę z prędkością 56 klatek na sekundę. System hybrydowy, wykorzystujący do wykonania zadania śledzenia układ FPGA i procesor ogólnego przeznaczenia opisano w pracy [10]. Na obrazach o rozdzielczości 320×240 opisywana architektura umożliwia śledzenie z częstotliwością bliską 80 Hz. Rozwiązanie do śledzenia celu wykorzystuje algorytm oparty o odległość Hausdorffa. Na uwagę zasługuje w tym przypadku niewielkie użycie zasobów przez zaimplementowany procesor. Opis dedykowanego systemu wizyjnego z dedykowanymi procesorami zaimplementowanymi w FPGA do wykrywania i śledzenia obiektów ruchomych zawarto w pracy [25].

Interesujące, choć nieliczne są doniesienia o wykorzystaniu układów reprogramowalnych jako kompletnego komputera pokładowego dla robota mobilnego – jako przykład podać można prace [26] i [12].

4. PRZYKŁAD IMPLEMENTACJI SYSTEMU WIZYJNEGO W UKŁADZIE FPGA

Prace nad wykorzystaniem układów FPGA jako elementów systemu wizyjnego były podejmowane również w Instytucie Automatyki i Inżynierii Informatycznej (IAiI) Politechniki Poznańskiej [5], [27], [28], [29]. W strukturach FPGA zaimplementowano często używane w praktyce przetwarzania obrazów operacje – filtrowanie obrazu za pomocą maski Gaussa o wielkości 3×3 , filtrowanie za pomocą ulepszego, warunkowego filtru medianowego o rozmiarze maski 3×3 , podstawowe operatory morfologiczne (erozja i dylacja), a także detektory najczęściej używanych cech - morfologiczny detektor krawędzi, oraz detektor cech punktowych (narożników) FAST oraz detektor Harris'a. Planowane jest wykorzystanie zaimplementowanych operatorów jako części toru wizyjnego robotów mobilnych wykorzystywanych jako platformy badawcze w IAiI. Przykładem takiej konstrukcji jest robot Ragnó (patrz rys. 2)[30].



Rys. 2. Robot Ragnó, wykorzystywany jako platforma testowa dla zaimplementowanego systemu wizyjnego

Zaimplementowane dedykowane bloki sprzętowe charakteryzują się wysoką wydajnością. System, w którym jednocześnie wykonywane są operacje filtrowania za pomocą filtru Gaussa oraz ulepszego filtru medianowego, a także wykrywania narożników oraz krawędzi pracować może z maksymalną częstotliwością taktowania 120 MHz (jest to prędkość działania najwolniejszego z dedykowanych bloków sprzętowych). Po wypełnieniu potoku przetwarzającego w jednym takcie zegara pobierana jest jedna próbka danych wejściowych (podawanych w trybie *progressive scan*, linia po linii, bez przeplotu) oraz zwracana próbka danych wyjściowych. Umożliwia to przetwarzanie powyżej 300 klatek obrazu o rozdzielczości VGA i 8-bitowej skali szarości na sekundę. Wyniki takie uzyskano przy wykorzystaniu jako platformy sprzętowej niedrogich układów FPGA z rodziny Spartan 3 firmy Xilinx. Możliwe jest w związku z tym zwiększenie prędkości działania (nawet do 250 %) przez zastosowanie układów wydajniejszych (np. Virtex 6). Przykładowe obrazy źródłowe oraz wyniki przetwarzania ich za pomocą dedykowanych bloków sprzętowych przedstawiono na rys. 3 i rys. 4.

Uzyskanie wspomnianej wyżej prędkości przetwarzania nie wymaga użycia znaczącej ilości zasobów sprzętowych. W tabeli 1 wyszczególniono zapotrzebowanie procesorów sprzętowych na podstawowe zasoby układów FPGA, takie jak przerzutniki (FF), tablice

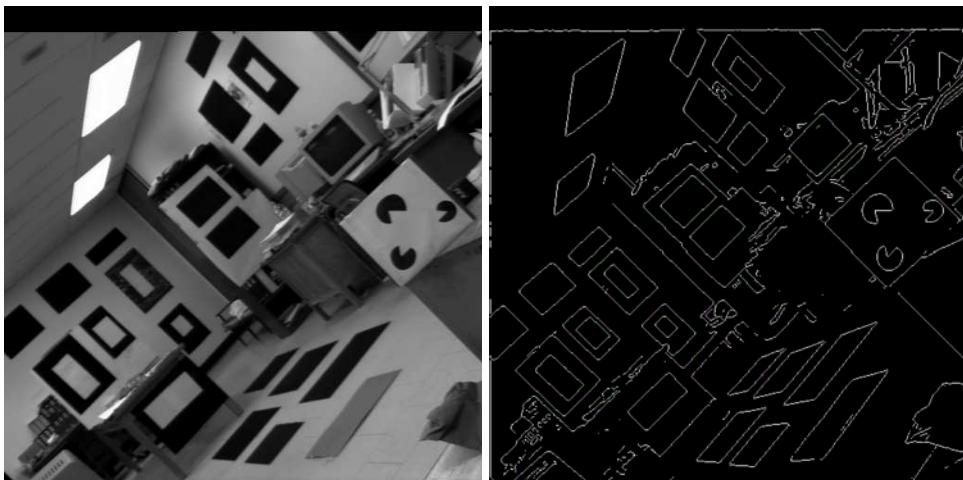
przełączniki, będące generatorami funkcji logicznych (LUT), a także szybkie dwuportowe pamięci (BlockRAM), wraz z maksymalnymi częstotliwościami taktowania. W ostatnim wierszu tabeli zamieszczono dla porównania ilość zasobów udostępnianą przez dostępny na rynku układ FPGA średniej klasy (cena przy zakupie dużej ilości wynosi około 70 \$).

Tab. 1. Porównanie zużycia zasobów układu FPGA przez poszczególne dedykowane procesory sprzętowe

	Block RAM	FF	LUT	F_{max} (MHz)
konwolucja (filtr Gaussa)	2	154	89	190
erozja/dylacja	2	208	443	155
warunkowy filtr medianowy	2	292	682	175
morfologiczny detektor krawędzi	6	516	689	150
detektor narożników FAST	10	1511	1291	150
detektor narożników Harrisa	10	1464	1040	120
XC3SD3400A	126	47744	47744	-



Rys. 3. Prezentacja działania sprzętowej implementacji algorytmu detekcji cech punktowych FAST – po lewej obraz oryginalny, po prawej obraz z naniesionymi wykrytymi cechami



Rys. 4. Prezentacja działania sprzętowej implementacji morfologicznego algorytmu detekcji krawędzi – po lewej obraz oryginalny, po prawej obraz z naniesionymi wykrytymi krawędziami

5. PERSPEKTYWY ROZWOJU

Moc obliczeniowa układów reprogramowalnych rośnie w takim samym tempie, co wydajność procesorów stosowanych w komputerach klasy PC. Główne ograniczenia związane są ze stosowaniem przy produkcji procesem technologicznym (rozmiarem pojedynczego tranzystora). Opisane powyżej rozwiązania wyraźnie wykazują, że układy FPGA mogą być wykorzystane jako platformy obliczeniowe w systemach wizyjnych robotów mobilnych. Czynnikiem hamującym ich upowszechnianie jest z pewnością zwiększona trudność implementacji algorytmów (w stosunku do implementacji czysto programowych), wynikająca głównie z fundamentalnych różnic w przebiegu projektowania i testowania, narzucanych przez zastosowaną architekturę. Dodatkowo, projektowanie wydajnych sprzętowych implementacji algorytmów wymaga sporej wiedzy z zakresu architektury układów cyfrowych i metod wykonywania obliczeń. Należy jednak nadmienić, że ciągły postęp dokonujący się w dziedzinie narzędzi projektowych ułatwia wykorzystanie potencjalnych możliwości układów reprogramowalnych i czyni je bardziej przystępnymi. Obserwowane są dwa dominujące trendy, mające na celu ułatwienie projektowania. Pierwszym z nich jest opracowywanie języków przypominających język C, jednak przekładających się wprost (w procesie kompilacji) na opis układu sprzętowego (tzw. języki *C-to-hardware*). Z doświadczeń Autora wynika, że jakkolwiek istnieją już w pełni funkcjonalne narzędzia tego typu, to poziom wydajności wynikowej architektury sprzętowej uzyskiwany przy pomocy dedykowanych języków opisu sprzętu (jak np. Verilog czy VHDL) jest lepszy, dzięki lepszej kontroli nad wykorzystywanymi zasobami układu. Problematykę tą szerzej opisano m.in. w publikacjach [31]. Przykład użycia języka opartego na C/C++ do implementacji systemu wizyjnego w FPGA opisano w pracy [32]. Drugim trendem jest równoległe projektowanie części sprzętowej i programowej systemu informatycznego w architekturze SoC (ang. *System on Chip*) i podział funkcji systemu pomiędzy nimi (ang. *hardware-software codesign*). Projektowanie według tej filozofii zakłada odpowiednie rozdzielenie zadań między komponenty zrealizowane w sposób czysto sprzętowy i komponenty wykonujące odpowiedni program tak, aby zadanie wykonywane było możliwie efektywnie. Opis metodologii oraz przykład projektu wykonanego według tej koncepcji opisano m. in. w pracy [33].

Szczególnie obiecujące dla aplikacji wizyjnych wydaje się być projektowanie z wykorzystaniem filozofii *hardware-software codesign*. Ilość zasobów logicznych udostępniana obecnie przez układy FPGA umożliwia implementowanie w nich z powodzeniem procesorów ogólnego przeznaczenia (nawet do kilkudziesięciu). Przykładem takiego mikroprocesora jest Microblaze, oferowany dla układów FPGA firmy Xilinx. Procesory takie – choć wolniejsze niż stosowane w komputerach PC procesory zgodne z architekturą x86 – wyposażone są w dedykowane interfejsy, umożliwiające dołączanie dedykowanych bloków sprzętowych, realizujących krytyczne sekcje kodu. Dodatkowo możliwe jest wykorzystanie wielu takich mikroprocesorów, w celu zrównoleglenia, a tym samym przyspieszenia wykonywania zadania zaimplementowanego w formie programu. Odpowiednie rozdzielenie zadań między sprzęt i program umożliwia zmaksymalizowanie efektywności wykorzystania zasobów i przyspieszenie wykonywania zadania [34]. Zagadnienie odpowiedniego podziału zadań ma również duże znaczenie w aplikacjach, w których układ reprogramowalny wykorzystywany jest jedynie jako tworzący część większego systemu dedykowany akcelerator.

Jak wspomniano wcześniej, duże znaczenie ma również odpowiednie dopasowanie implementowanych algorytmów do architektury reprogramowalnej, w której przyspieszenie

uzyskuje się nie tylko przez zwiększanie częstotliwości taktowania, ale też przez zrównoleglenie wykonywanych zadań i wydłużanie potoku przetwarzającego. Przenoszenie wprost aplikacji z dziedziny programowej do sprzętowej często nie daje najlepszych wyników. Modyfikowanie istniejących algorytmów, lub też opracowywanie algorytmów dedykowanych dla aplikacji sprzętowych wydaje się być wysoce atrakcyjnym kierunkiem badań.

6. PODSUMOWANIE

W raporcie przedstawiono perspektywy wykorzystania reprogramowalnych układów cyfrowych FPGA w systemach wizyjnych. Zaprezentowane zostały liczne aplikacje, w których wykorzystano z powodzeniem tą platformę obliczeniową do realizacji zadań stawianych przed współczesnymi systemami wizyjnymi. Wykorzystanie możliwości układów FPGA niejednokrotnie umożliwia znaczące zwiększenie wydajności systemu wizyjnego, przy jednoczesnym zmniejszeniu poboru prądu, a także gabarytów kompletnego rozwiązania. Ma to zasadnicze znaczenie w przypadku zastosowań systemów wizyjnych w aplikacjach mobilnych, w tym robotyce. Układy reprogramowalne nie zawsze jednak nadają się do zastąpienia najczęściej stosowanych jako platforma obliczeniowa komputerów klasy PC – uzyskiwane przyspieszenie jest zależne od rodzaju implementowanego algorytmu. Istnieją również algorytmy, których implementacja w postaci sprzętowej jest nieopłacalna – nie powoduje przyspieszenia wykonywania obliczeń, a dodatkowo zużywa znaczącą ilość zasobów sprzętowych. Nawet w takim przypadku bardzo często możliwe jest jednak wyodrębnienie części algorytmu, której implementacja sprzętowa daje korzyści. Warto w takim przypadku zastosować układ FPGA w roli dedykowanego koprocesora.

Stały postęp w dziedzinie rozwoju narzędzi, a także metodyki projektowania hybrydowych systemów sprzętowo-programowych pozwala sądzić, iż w przyszłości nastąpi upowszechnienie rozwiązań sprzętowych w robotyce mobilnej. Stworzy to zapotrzebowanie na nowe, dedykowane do implementacji sprzętowej algorytmy, umożliwiające bardziej efektywne rozwiązywanie napotykaných w tej dziedzinie problemów.

Praca naukowa finansowana ze środków na naukę w latach 2008 - 2010 w ramach projektu badawczego N514 294635.

LITERATURA

- [1] W. James MacLean, An evaluation of the suitability of FPGAs for embedded vision systems. CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, str. 131, Washington, DC, USA, 2005. IEEE Computer Society.
- [2] B. A. Draper, J. R. Beveridge, A. P. W. Bohm, C. Ross, and M. Chawathe, Accelerated image processing on FPGAs. Image Processing, IEEE Transactions on, 12(12): p. 1543-1551, 2003.
- [3] Kazimierz Wiatr, Akceleracja obliczeń w systemach wizyjnych, Wydawnictwa Naukowo-Techniczne, 2003.

- [4] Cesar Torres-Huitzil and Miguel Arias-Estrada, Real-time image processing with a compact FPGA-based systolic architecture, *Real-Time Imaging*, 10(3): p. 177-187, 2004.
- [5] M. Kraft and A. Kasiński, Improved median filter using conditional technique and its hardware implementation, *Proc. of 15th European Signal Processing Conf. EUSIPCO 2007*, p. 1649-1652. EURASIP, 2007.
- [6] A.M. Alsuwailem, S.A. Alshebeili, A new approach for real-time histogram equalization using FPGA. *Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems 2005*, p. 397-400, Washington, DC, USA, 2005. IEEE Computer Society.
- [7] David G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60(2): p. 91-110, 2004.
- [8] Herbert Bay, Tinne Tuytelaars, L. Van Gool, SURF: Speeded up robust features, *Proc. Of the 9th European Conference on Computer Vision*, Graz, Austria, May 2006.
- [9] Motilal Agrawal, Kurt Konolige, Morten Rufus Blas, CenSurE: Center surround extremas for realtime feature detection and matching, *Lecture Notes in Computer Science 5303(4)*, p. 102-115, Springer, 2008.
- [10] Miguel Arias-Estrada, Eduardo Rodríguez-Palacios, An FPGA coprocessor for real-time visual tracking, *Proceedings of the 12th International Conference on Field-Programmable Logic and Applications*, p. 710-719, London, UK, 2002. Springer-Verlag.
- [11] Steven M. Smith, A new class of corner finder, *Proceedings of the 3rd British Machine Vision Conference BMVC'92*, p. 139-148, Londyn, UK, 1992.
- [12] Wade S. Fife, James K. Archibald, Reconfigurable on-board vision processing for small autonomous vehicles. *EURASIP Journal on Embedded Systems*, 2007(1): p. 33, 2007.
- [13] C. Harris, M. Stephens, A combined corner and edge detector, *Proc. Of the 4th Alvey Vision Conference*, p. 147-151, 1988.
- [14] Cristina Cabani, W. James MacLean, A proposed pipelined architecture for FPGA-based affine-invariant feature detectors, *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, p. 121, Washington, DC, USA, 2006, IEEE Computer Society.
- [15] Harding Djakou Chati, Felix Mühlbauer, Tim Braun, Christophe Bobda, Karsten Berns. SOPC architecture for a key point detector, *Proceedings of the International Conference on Field-Programmable Logic and Applications*, p. 710-713. IEEE, 2007.
- [16] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) p. 679-698, November 1986.
- [17] Wenhao He, Kui Yuan, An improved Canny edge detector and its realization on FPGA, *Intelligent Control and Automation, 7th World Congress on*, 2008, p. 6561-6564, 2008.
- [18] S.-M. Karabernou, L. Kessal, F. Terranti. Real-time FPGA implementation of Hough transform using gradient and CORDIC algorithm, *Image and Vision Computing*, 23(11) p. 1009-1017, Oct., 2005.
- [19] J.P. Cyganek, B. Siebert, *An Introduction to 3D Computer Vision Techniques and Algorithms*, Wiley, 2009.

- [20] Mingxiang Li, Yunde Jia, Stereo vision system on programmable chip (SVSoC) for small robot navigation, In IROS'06.
- [21] Divyang K. Masrani, W. James MacLean, Expanding disparity range in an FPGA stereo system while keeping resource utilization low, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, p. 132, Washington, DC, USA, 2005. IEEE Computer Society.
- [22] Mark L. Chang, Stephen Longfield, Jr., A parameterized stereo vision core for FPGAs, Proc. of IEEE Symposium on Field-Programmable Custom Computing Machines, p. 1359-1365, 2009.
- [23] Kazuhiro Shimizu, Shinichi Hirai, Realtime and robust motion tracking by matched filter on CMOS+FPGA vision system. Proc. of the IEEE Conf. of Robotics and Automation, pages 788-793, 2007.
- [24] Jung Uk Cho, Seung Hun Jin, Dai Pham Xuan, Jae Wook Jeon, Multiple objects tracking circuit using particle filters with multiple features, Proc. of the IEEE Conf. of Robotics and Automation (ICRA), p. 4639-4644, 2007.
- [25] Pierre Chalimbaud, François Berry, Embedded active vision system based on an FPGA architecture, EURASIP Journal on Embedded Systems, 2007(1), p. 26, 2007.
- [26] Mark Maimone, Yang Cheng, Larry Matthies, Two years of visual odometry on the mars exploration rovers: Field reports, Journal of Field Robotics, 24(3): p. 169-186, 2007.
- [27] Marek Kraft, Adam Schmidt, Andrzej J. Kasinski, High-speed image feature detection using FPGA implementation of FAST algorithm, Proc. of the International Conference on Computer Vision and Applications VISAPP 2008 (1), p. 174-179, 2008.
- [28] Marek Kraft, Andrzej J. Kasinski, Morphological edge detection algorithm and its hardware implementation, Computer Recognition Systems 2, p. 132-139, 2008.
- [29] M. Kraft, M Fularz, Porównanie implementacji sprzętowych dwóch popularnych detektorów cech punktowych, Pomiary Automatyka Kontrola, vol. 55(8) 2009, p. 618-620.
- [30] D. Belter, K. Walas, P. Skrzypczyński, Autonomiczny robot sześcionożny – rozwój konstrukcji i systemu sterowania (Autonomous Six Legged Walking Robot - Design and Control Architecture Development), Pomiary Automatyka Robotyka, 2 (CD-ROM), 2009
- [31] Stephen A. Edwards, The challenges of hardware synthesis from C-like languages, Proc. Of Design, Automation and Test in Europe Conference and Exhibition, vol. 1: p. 66-67, 2005.
- [32] Najeem Lawal, Mattias O'Nils, Benny Thornberg, C++ based system synthesis of real-time video processing systems targeting FPGA implementation, Proc. of International Parallel and Distributed Processing Symposium, p. 203, 2007.
- [33] T. Wiantong, P. Y. K. Cheung, W. Luk, Hardware/software codesign: a systematic approach targeting data-intensive applications, IEEE Signal Processing Magazine, IEEE, 22(3): p. 14-22, 2005.
- [34] P. Huerta, J. Castillo, J. I. Martínez, V. López, Multi microblaze system for parallel computing. Proceedings of the 9th International Conference on Circuits, str. 1-6, Stevens Point, Wisconsin, USA, 2005.