

Mariusz KAPRUZIAK, Bogdan OLECH, Mirosław ŁAZORYSZCZAK

POLITECHNIKA SZCZECIŃSKA, WYDZIAŁ INFORMATYKI, INSTYTUT ARCHITEKTURY KOMPUTERÓW I TELEKOMUNIKACJI

Koncepcja procesora rekonfigurowanego przeznaczonego do wspomagania widzenia maszynowego

Dr inż. Mariusz KAPRUZIAK

Obronił pracę doktorską na Politechnice Szczecińskiej w 2006r na temat "Opracowanie koncepcji i realizacja dedykowanego procesora radia programowalnego". Obecnie prowadzi badania dotyczące systemów rekonfigurowanych oraz widzenia maszynowego na Politechnice Szczecińskiej.



e-mail: mkapruziak@wi.ps.pl

Dr inż. Bogdan OLECH

Ukończył studia na Wydziale Elektrycznym Politechniki Szczecińskiej, obronił pracę doktorską w 1999 r. na Wydziale Elektroniki i Techniki Informatycznych Politechniki Warszawskiej. Jest adiunktem w Instytucie Architektury Komputerów i Telekomunikacji Politechniki Szczecińskiej. Jego zainteresowania naukowe to metodyka konstruowania systemów rozproszonego zbierania i przetwarzania danych w powiązaniu z techniką radiową, systemy rekonfigurowalne, zagadnienia modelowania warstwy fizycznej systemu.



e-mail: bolech@wi.ps.pl

Dr inż. Mirosław ŁAZORYSZCZAK

Ukończył studia na Wydziale Elektrycznym Politechniki Szczecińskiej, obronił pracę doktorską w 2005 r. Pracuje jako adiunkt w Instytucie Architektury Komputerów i Telekomunikacji. Jest członkiem IEEE. Zainteresowania naukowe dotyczą systemów i układów rekonfigurowalnych.



e-mail: mlazoryszczak@wi.ps.pl

Streszczenie

W artykule zawarto początkowe wyniki projektu rozpoczętego na Politechnice Szczecińskiej i dotyczącego nowatorskiej metody realizacji systemu wizualnej inspekcji maszynowej płytek PCB. Nowością, której dotyczy badanie, jest wykorzystanie elastycznych, dynamicznych algorytmów widzenia maszynowego, umożliwiających uwzględnienie w nich specyfiki rozpatrywanej aplikacji. Ze względu na dynamiczność algorytmów nie jest możliwe zaproponowanie w pełni strukturalnego rozwiązania, możliwego do realizacji w układach FPGA. Konieczne jest uwzględnienie procesu kosyntezy sprzętowo-programowej. Specyfika użytego procesu kosyntezy polega na tym, że oprogramowanie i sprzęt przeplatają się tutaj wzajemnie, nie tworząc konwencjonalnej struktury procesora i koprocesora ani nie wpisują się w znane architektury ISA współczesnych procesorów. Propozycja jednego takiego procesora, dedykowanego dla określonej klasy algorytmów widzenia maszynowego przedstawiona została w tym artykule.

Słowa kluczowe: rekonfigurowalność, FPGA, widzenie maszynowe.

Concept of reconfigurable processor dedicated for machine vision system

Abstract

This article describes initial results of the project started on Technical University of Szczecin and concerning novel method to implement visual inspection system for PCB. The novel part being under analysis is the usage of flexible, dynamic algorithms for machine vision, which allow to introduce specific parts of the considered application directly into the algorithm structure. However dynamic algorithms are impossible to implement in purely structural way on FPGA devices. It is necessary to incorporate hardware/software codesign process into design methodology. The specific part of codesign methodology in this application is, that no conventional processor/co-processor or ISA architecture could be used. The proposal of one of those processors, dedicated for particular class of machine vision algorithms has been presented in this paper.

Keywords: reconfigurability, FPGA, machine vision.

1. Wstęp

W przeciągu wielu lat badań nad systemami widzenia maszynowego zaproponowano już wiele procesorów dedykowanych do

tego zadania [1, 2, 3]. Motywacją do podjęcia prac nad nową architekturą była pewna obserwacja dotycząca innej metody podejścia do problemu rozpoznawania obrazów. Metoda ta jest pewnym skrajnym przypadkiem metod rozpoznawania zależnego od kontekstu (*context-based recognition*) [4, 5]. Postuluje się w niej odwrócenie „kierunku” procesu rozpoznawania. Konwencjonalne systemy w pierwszej kolejności wykonują przetwarzanie obrazu. Najczęściej wykonywane jest rzutowanie przestrzeni obrazu na przestrzeń parametrów (*Hough Transform* [1]) lub inną wirtualną przestrzeń wygodą do przetwarzania (jak FMT [6]). Następnie na podstawie analizy wyników w tej przestrzeni podejmowana jest decyzja o obecności i lokalizacji obiektów.

W rozważanym przypadku natomiast hipotezy dotycząca wyglądu sceny znana jest już od początku. Dalsze przetwarzanie koncentruje się na potwierdzeniu lub odrzuceniu tych hipotez. Na podstawie wyników odrzuceń/potwierdzeń algorytm modyfikuje hipotezę początkową, tak aby dopasować ją do rozkładu sceny rzeczywistej.

W ogólności system taki może być mało stabilny. Może tworzyć i wyciągać konsekwencje z obiektów których nie widać na obrazie. Wszystko zależy od tego jak dobrze uda się określić początkową hipotezę i jak dobrze na podstawie historii wyników odrzuceń/potwierdzeń można ją modyfikować. Wymaga to bardzo dobrej znajomości kontekstu obrazu na etapie projektowania algorytmu i wpisania jej w algorytm rozpoznawania. To założenie dla niektórych aplikacji nie jest trudne do spełnienia. Aplikacje te to głównie aplikacje inspekcji maszynowej, na przykład inspekcji płytek PCB [7]. Rozważania takie, szczególnie nad inspekcją płytek PCB, stały się inspiracją dla autorów do podjęcia prac nad opisanymi wyżej algorytmami i nad propozycją dedykowanego dla nich procesora.

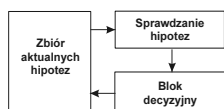
Opisana wyżej koncepcja algorytmu charakteryzuje się zdecydowanie mniejszą mocą obliczeniową konieczną do jego wykonania od rozwiązań konwencjonalnych. Problem polega tylko na tym, że struktura algorytmu zmienia się w zależności od wcześniejszych wyników przyjęcia/odrzućcia hipotezy. Taki algorytm relatywnie łatwo jest wykonać w sposób proceduralny na procesorze generalnego przeznaczenia, trudniej natomiast w sposób strukturalny, na przykład w układzie FPGA. Na tym zagadnieniu koncentruje się właśnie niniejszy artykuł.

2. Klasa algorytmów

W pracy rozważane są algorytmy zakładające dużą wiedzę a-priori o scenie zawartej w obrazie już przed rozpoczęciem analizy obrazu. Wiedza ta zawarta jest w konstrukcji hipotez początkowych oraz w algorytmie bloku decyzyjnego (rys. 1). Hipotezy początkowe są sprawdzane, poprzez uwiarygodnianie ich lub odrzucanie na podstawie obrazu. Wynik sprawdzenia następnie, na podstawie algorytmu zawartego w bloku decyzyjnym, zmienia parametry tych hipotez.

Pod pojęciem hipotezy rozumie się zbiór figur i ich parametrów. W najprostszym przypadku mogą to być linie proste, charakteryzowane przez punkt początkowy i końcowy (oraz w sposób jednoznaczny wynikający z tego kąta linii względem poziomu).

Blok sprawdzania hipotez to blok w którym stwierdzane jest występowanie danej figury na scenie. Odpowiedź jest liczbą z przedziału od zera do jeden. Zero oznacza definitywne stwierdzenie braku figury na scenie, jedynka definitywne potwierdzenie jego istnienia. Zakłada się w artykule, że sprawdzanie realizowane jest przez algorytm dopasowanie wzorca (*template matching*).



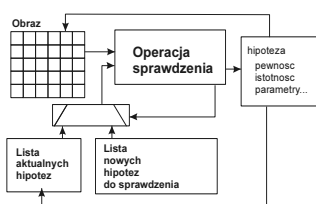
Rys. 1. Struktura algorytmów należących do rozpatrywanej klasy algorytmów
Fig. 1. Structure of considered algorithms

Wynik sprawdzania jest przekazywany do bloku decyzyjnego. Algorytm bloku decyzyjnego to najczęściej zbiór miękko decyzyjnych reguł, typu „JEŚLI dwie linie pod kątem 90 stopni są blisko danej linii i istnieje linia równoległa do danej TO stwórz nową hipotezę prostokąta” lub „JEŚLI prostokąt istnieje na scenie TO zwiększ stopień pewności dla linii z nim związanych”. Zmienne lub nowo dodane hipotezy są następnie sortowane względem współczynnika będącego sumą ważoną ich stopnia pewności i „istotności”. Istotność rozumiana jest tutaj, jako miara konsekwencji wynikających dla sceny z przyjęcia, że dana figura istnieje. Określana jest często jako liczba większych obiektów, które wykorzystały istnienie danej obiektu jako jedną z cech warunkujących ich dodanie.

3. Propozycja architektury procesora

Kluczem do propozycji architektury procesora przedstawionej w artykule, jest obserwacja, że przetwarzanie w bloku sprawdzania hipotezy i zmiany hipotez, przy odpowiedniej organizacji algorytmu, może zostać zrealizowane jako ta sama operacja. W takim przypadku głównym zadaniem procesora byłoby wykonywanie tej jednej operacji wiele razy dla różnych danych.

Blok sprawdzania realizuje operacje dopasowania do wzorca. Można zauważyć, że wzorzec może zostać zdefiniowany jako warunek „JEŚLI punkt o pozycji x_0, y_0 ma wartość 1 ORAZ punkt o pozycji x_1, y_1 ma wartość.... TO pewność istnienia figury = 1”. Podobnie zapisywane są warunki w bloku zmiany hipotez, jak na przykład „JEŚLI istnieje linia rozpoczynająca się w punkcie (x_0, y_0) , kończąca się (x_k, y_k) i mająca kąt α ORAZ znaleziono punkt (x_1, y_1) nie należący do linii ORAZ punkt (x_1, y_1) tworzy z punktem (x_0, y_0) linię o kącie α TO dodaj hipotezę o linii (x_0, y_0) do (x_1, y_1) z początkową pewnością = 0.5”. W obu warunkach należy sprawdzić przesłanki. Następnie, stosownie do stopnia ich spełnienia przypisywana jest pewność danej hipotezy.



Rys. 2. Koncepcja architektury procesora
Fig. 2. Concept of processor architecture

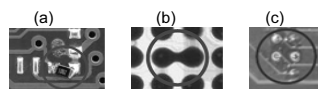
Operacja sprawdzania (rys.2) na wejściu pobiera hipotezę do przetwarzania. Na podstawie zawartych przesłanek tej hipotezy odczytuje wartości punktów z obrazu. Dla każdego punktu obrazu przechowywana jest informacja o hipotezach go dotyczących. Operacja sprawdzania może trwać dość długo i wymaga przesłania wielu danych. Czas jej wykonania jest zmienny, konfigurowalny w zależności od istotności badanej hipotezy. Im istotniejsza dla obrazu hipoteza tym dłużej jest przetwarzana. Ograniczenie czasu skutkuje zmniejszeniem liczby przesłanek, które są sprawdzane. Przesłanki do sprawdzenia wybierane są w sposób pseudo-

losowy. Wynik pewności hipotezy obarczony jest zatem pewnym rozrzutem, szacowanym przez parametr wariancji.

Wyniki wykonania operacji sprawdzania to uaktualnienie parametrów hipotezy lub dodanie nowej hipotezy na listę oraz przypisanie wskaźnika na tą hipotezę do dotyczących jej punktów obrazu. Dodatkowo określane jest prawdopodobieństwo wybrania w następnej operacji uaktualnienia parametrów istniejącej hipotezy lub sprawdzenia warunków tworzenia nowej. Zakończenie operacji sprawdzania wymaga przesłania wielu danych po szynie danych.

4. Środowisko uruchomieniowe

Procesor implementowany jest w układzie FPGA Xilinx Spartan3 XC3S500E zawarty w module firmy Digilent. Podłączony jest do modułu kamery Maritex CAM-PO2030NMSHPP03 umieszczonej na ramieniu nad płytą drukowaną. Ramie może przesuwać się w dwóch osiach x i y . Opis wyglądu płytki PCB wykonywany jest ręcznie, jako zbiór figur/hipotez, które powinny znaleźć się w widzianym obrazie. Taki zbiór stanowi jednocześnie początkową scenę. Scena ta jest następnie modyfikowana przez algorytm z dwóch powodów. Pierwszy to określenie przesunięcia płytki względem zdefiniowanej sceny, jej obrotu i w pewnym wąskim zakresie także skali. Drugi to identyfikacja błędów znajdujących się na tej płytce, rozumianych jako rozbieżności między obiektami zidentyfikowanymi na scenie a tymi, znajdującymi się w obrazie wzorcowym. Obiekty takie pokazywane na monitorze są podświetlone kolorem czerwonym.



Rys. 3. Szczególne przypadki usterek płytek PCB (a) przesunięcie elementów na płycie (b) połączenia pomiędzy sąsiednimi pinami (c) nie przyłutowane piny
Fig. 3. Particular PCB Boards fault cases (a) displacement of elements (b) bridge (c) not soldered pins

5. Podsumowanie

Prace nad projektem wciąż trwają i nie całość przedstawionych koncepcji została jeszcze zaimplementowana. Słabą stroną dotychczasowych rozwiązań jest operowanie na obrazach binarych. Obraz, przed przekazaniem do algorytmu, jest poddawany algorytmowi adaptacyjnego progowania (*thresholding*) [1]. Planuje się także rozwinięcie zbioru figur podstawowych, aby móc rozpoznawać i budować większe obiekty nie tylko ze zbioru linii. Planuje się także przeprowadzenie głębszych analiz wykonania bloku zmian hipotez, który na tym etapie nie pozwala na istnienie zbyt wielu reguł.

6. Literatura

- [1] E.R. Davis, Machine Vision, Theory, Algorithms, Practicalities, Elsevier 2005, ISBN: 0122060938.
- [2] S. Tagzout, K. Achour, K. Djekoune, Hough Transform Algorithm for FPGA Implementation, Signal Processing 84 (2001), pp. 1295-1301.
- [3] T.H. Draber, W.E. King, J.G. Tront, R.W. Conners, P.A. Araman, Using multiple FPGA architectures for real-time processing of low-level machine vision functions, Industrial Electronics, Control and Instrumentation 1995, pp. 1284-1289 vol.2.
- [4] T.M. Strat, M.A. Fischler, Context-Based Vision: Recognizing Objects Using Information from Both 2D and 3D Imagery, IEEE Transactions on Pattern Analysis and Machine Intelligence 1991, vol. 13, pp 1050-1065
- [5] J. Laaksonen, E. Oja, M. Koskela, S. Brandt, Analyzing Low-level Visual Features Using Content-Based Image Retrieval, 7-th International Conference on Neural Information Processing (ICONIP 2000) Korea.
- [6] Y. Sheng, H.H. Arsenault, Experiments on Pattern Recognition Using In-variant Fourier-Mellin Descriptors, J. Opt. Soc. Am. A 3, 771, 1986
- [7] M. Moganti, F. Ercal, C.H. Dagli, Automatic PCB Inspection Algorithms: A Survey, Computer Vision and Image Understanding 1996, vol. 63, pp 287-313.