

Bogdan KWOLEK  
Politechnika Rzeszowska

## SPRZĘTOWA REALIZACJA PRZEPLYWU OPTYCZNEGO W SYSTEMIE WIZYJNYM ROBOTA MOBILNEGO\*/

**Streszczenie.** W pracy przedstawiono wyniki prac związanych z wyznaczaniem przepływu optycznego w układach reprogramowanych. Przedstawiono sposób zaprogramowania układu FPGA firmy Altera. Ze względu na niewielkie opóźnienia wprowadzane w torze wizyjnym oraz stały czas przetwarzania układ szczególnie dobrze nadaje się do systemów czasu rzeczywistego.

## HARDWARE DESIGN FOR OPTICAL FLOW IN MOBILE ROBOT VISION SYSTEM

**Summary.** The paper presents results of work focused on determining of optical flow in programmable devices. The method of implementing of the optical flow algorithm in FPGA has been proposed. The structure of hardware processor has been shown as well. Because of small delays in vision system as well as constant processing time, the presented system is especially suitable for real-time applications.

### 1. Wprowadzenie

System wizyjny powinien umożliwiać robotowi mobilnemu orientację w przestrzeni oraz interakcję z otaczającym środowiskiem. Do programowania zadań robota wykorzystywany powinien być prosty język oparty między innymi na metodach niewerbalnych (pozasłownych, gestach, mimice). Do realizacji tych i innych pokrewnych zadań niezbędne są układy stereoskopowych kamer wykonujących pomiary z dokładnością w granicy jednego centymetra oraz systemy wizyjne dysponujące znaczną mocą obliczeniową. Moc obliczeniowa dostępnych obecnie systemów wizyjnych do realizacji zadań formułowanych w podobny sposób jest niewystarczająca lub koszt uzyskania jest zbyt duży. Celem zapewnienia odpowiedniego czasu reakcji systemów wizyjnych obecnie wykorzystywanych stosuje się odpowiednie akceleratory sprzętowe oraz obliczenia równoległe.

W niniejszej pracy nawiązano do przetwarzania obrazu w rekonfigurowanych układach cyfrowych. Omówiono problemy związane z wykryciem i lokalizacją poruszającego się obiektu. W szczególności omówiono sposób zaprogramowania układu reprogramowalnego wyznaczającego przepływ optyczny. W systemach wizyjnych przepływ optyczny stosuje się

\*/ Pracę wykonano w ramach projektu badawczego Nr 8T11A117 finansowanego przez KBN w latach 1999-2001.

celem określenia obszaru ruchu. Z wyznaczonego pola energii ruchu pochodzić mogą parametry określające prędkość obiektu.

Pole przepływu optycznego jest dwuwymiarowym rozkładem pozornych prędkości związanych ze zmianami jasności w obrazie. Wyznaczenie go na podstawie wzorów analitycznych jest bardzo pracochłonne. W niniejszej pracy posługiwano się metodą Horna-Schuncka i równaniem prędkości normalnej do linii stałej jasności [4]. Detekcja ruchu odbywa się w punktach obrazu, w których występuje wyraźna zmiana jasności.

Systemy wizyjne wykorzystujące przepływ optyczny powinny pracować opierając się na obrazach, na których zarejestrowano niewielkie, kilkupikselowe przemieszczenia obiektów. W wielu zastosowaniach praktycznych niezbędna jest zatem częstotliwość pracy systemu wizyjnego równa 25 Hz lub większa. Zapewnienie przez systemy wizji komputerowej odpowiedniego okresu próbkowania, wobec dużych wymagań obliczeniowych związanych z wyznaczeniem obszaru ruchu, wiąże się ze znacznymi kosztami lub wręcz jest niemożliwe w realizacji. Zaprezentowany w niniejszej pracy układ przetwarza na bieżąco obraz udostępniony na wyjściu przetwornika obrazu oraz wprowadza niewielkie opóźnienia w torze wizyjnym. Ze względu na stały czas przetwarzania układ szczególnie dobrze nadaje się do systemów czasu rzeczywistego.

W końcowej części pracy nawiązano do eksperymentów praktycznych polegających na śledzeniu ścieżki geometrycznej przez robot mobilny wyposażony w kamerę pokładową. Wykorzystano robot Khepera z miniaturową monochromatyczną kamerą CCD. Do detekcji linii ciągłej lub przerywanej zaznaczonej na jezdni wykorzystano metody przepływu optycznego. Zastosowanie przepływu optycznego zamiast metod wykorzystujących binaryzację pozwoliło zmniejszyć wpływ oświetlenia zewnętrznego i polepszyć skuteczność detekcji linii.

## 2. Układy reprogramowalne

Gwałtowny rozwój techniki cyfrowej zapoczątkowany kilka lat temu doprowadził do wyprodukowania układów reprogramowalnych składających się z dziesiątków milionów tranzystorów w jednym układzie. Struktura wewnętrzna tych układów może być szybko i wielokrotnie zmieniana programowo przez użytkownika w celu dopasowania jej do wykonywanych zadań. Cechą wspólną układów mikroprocesorowych i FPGA jest możliwość ich programowania. Zasadniczą różnicą jest zaś sposób realizacji obliczeń. Mikroprocesor wykonuje rozkazy pobrane z pamięci w znacznej mierze w sposób sekwencyjny. W układzie programowalnym algorytm wykonuje się równolegle w drodze realizacji operacji odwzorowanych w zasoby sprzętowe. Programowalność układu pozwala na odwzorowanie zadanych funkcji w sprzęcie w zależności od potrzeb, a także pozwala na eliminowanie błędów



zauważonych w trakcie funkcjonowania układu. W układach ASIC jakakolwiek modyfikacja algorytmu lub usunięcie błędu wiąże się ze znacznymi kosztami.

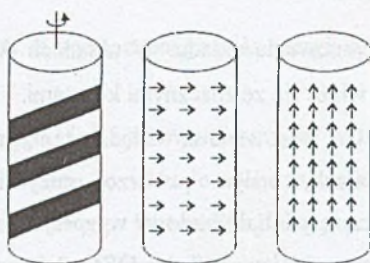
Układy programowane FPGA stosowane są wszędzie tam, gdzie zależy na funkcjonalności, elastyczności, małych wymiarach, małym poborze mocy i niezawodności. Głównie wykorzystuje się je jako akceleratory lub do budowy wyspecjalizowanych urządzeń. Urządzenia te mają w porównaniu do systemów opartych na DSP (lub na procesorach uniwersalnego przeznaczenia) najkorzystniejszy stosunek wydajności obliczeniowej do ceny. Jednak głównym zastosowaniem układów reprogramowanych są systemy czasu rzeczywistego. W rozwiązaniach opartych na układach specjalizowanych nie ma potrzeby stosowania kosztownych systemów operacyjnych czasu rzeczywistego. W systemie mikroprocesorowym rola systemu czasu rzeczywistego w istocie sprowadza się do dynamicznego przydzielania procesora realizowanym zadaniom w ten sposób, aby dane mogły być obsługane w gwarantowanym przez producenta czasie. W systemie opartym na układach specjalizowanych zasoby do realizowanych funkcji przydzielone są na stałe. Na wejście układu dane dostarczane są cyklicznie - możliwe jest zatem przetwarzanie potokowe. W wielu częściach algorytmów odwzorowanych w układzie programowalnym nie ma strat związanych z zapisem lub odczytem danych z pamięci. Wynik dostępny na wyjściu elementu przetwarzającego może być natychmiast przetworzony przez następny element znajdujący się w linii potokowej.

Zastosowanie układów FPGA do cyfrowego przetwarzania sygnałów wiąże się z uzyskaniem wielu innych ważnych zalet: wysokiego współczynnika wykorzystania zasobów układu, wydajności obliczeń przewyższającej wydajność dostępną w układach DSP, dużej precyzji obliczeń biorącej się z tego, że operacje mogą być na realizowane na danych reprezentowanych przez słowa o pożądanej długości, niekoniecznie jednakowej w całym algorytmie.

### 3. Przepływ optyczny

Obecnie w wielu dziedzinach stosowane są urządzenia rozpoznające i śledzące obiekty ruchome. Zarejestrowane przemieszczenie może stanowić dodatkowe źródło informacji dla systemu wizyjnego. W szczególności nieciągłości wykryte w przepływie optycznym mogą być użyteczne w trakcie segmentacji obrazu. Przepływ optyczny wykorzystuje się również do określenia parametrów ruchu [3].

Pole przepływu optycznego jest dwuwymiarowym rozkładem pozornych prędkości związanych ze zmianami jasności w obrazie. Wyznaczenie go na podstawie dokładnych wzorów analitycznych jest pracochłonne. Wykorzystano więc metodę zaproponowaną w [1], w której

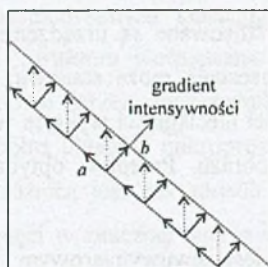


Rys. 1. Zjawisko przepływu optycznego. a) wirujący walec, b) pole prędkości, c) przepływ optyczny

Fig. 1. The motion field and optical flow. a) rotating cylinder, b) motion field, c) optical flow

oblicza się składową prędkości  $v_n$  normalną do linii stałej jasności. Detekcja ruchu odbywa się w punktach obrazu, w których występuje wyraźna zmiana stopnia szarości. Punkty te to zazwyczaj krawędzie obiektów będących w ruchu, zob. rys. 1.

Przepływ optyczny wyznacza się zakładając, że jasność  $I(x, y)$  punktu w chwili  $t$  oraz  $t + \delta t$  pozostaje niezmienna w myśl zależności:  $I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t)$ . Po rozwinięciu równania w szereg Taylora i odrzuceniu wyrazów wyższych rzędów otrzymuje się równanie  $I_x u + I_y v + I_t = 0$ , w którym  $I_x, I_y$  oznaczają pochodne przestrzenne obrazu w kierunkach odpowiednio poziomym i pionowym,  $I_t$  pochodną po czasie obrazu intensywności, zaś  $u, v$  prędkości w przestrzeni obrazu. Równanie wyraża w każdym punkcie obrazu związek między pochodnymi przestrzennymi  $I_x, I_y$  i po czasie  $I_t$ , a przepływem optycznym  $u, v$  w tym punkcie. Wyznaczenie wektora prędkości uzyskać można m.in. po wprowadzeniu ograniczenia równościowego redukującego do minimum błąd kwadratowy  $E^2(x, y) = (I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)$ , gdzie  $\lambda$  jest mnożnikiem Lagrange'a. Bezpośrednie wykorzystanie równania przepływu optycznego do wyznaczenia prędkości wiąże się z szeregiem trudności obliczeniowych, a w szczególności praktycznych, z których najbardziej uciążliwy jest tzw. problem apertury, zilustrowany w sposób poglądowy na rys. 2.



Rys.2. Ilustracja problemu apertury

Fig.2. Illustration of the aperture problem



W celu przemieszczenia linii prostej zaznaczonej na wspomnianym rysunku możliwy jest jedynie pomiar składowej  $b$  prędkości. Przyjmując szereg uproszczeń, prędkość można estymować na podstawie równania  $-I_t = \text{grad}(O) * c$ , gdzie  $\text{grad}(O)$  jest dwuwymiarowym gradientem obrazu, zaś  $c = (dx / dt, dy / dt) = (u, v)$ .

W wielu zagadnieniach obliczeniowych związanych z wyznaczaniem przepływu wykorzystuje się metodę Horna-Schnucka [4, 1]. Przetwarza się w niej trzy kolejne obrazy  $I_1, I_2, I_3$  poddane filtracji Gaussa. Składową prędkości  $v_n$  normalną do linii stałej jasności wyznacza się na podstawie następującego równania:  $v_n = -I_t / \sqrt{I_x^2 + I_y^2}$ . W wyniku otrzymuje się obraz uwypuklający obszary w których nastąpiła zmiana jasności w przetworzonej sekwencji obrazów. Po zbinaryzowaniu pola energii ruchu uzyskuje się zaznaczone na obrazie fragmenty obiektów, dla których zarejestrowano pożądaną prędkość  $v_n$ .

Do wyznaczania wymaganych gradientów występujących w omawianej metodzie możliwe jest stosowanie technik *look up table* oraz metod równoległego przetwarzania obrazów. Przy założeniu, że w buforze zapamiętano dwa kolejne obrazy  $I_1, I_2$  poddane filtracji Gaussa, schemat algorytmu przedstawić można w sposób następujący:

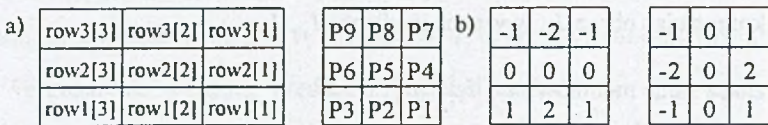
- pobierz obraz  $I_3$ , dokonaj filtracji Gaussa, zapamiętaj obraz,
- wyznacz różnicę obrazów  $I_3, I_1$ ,
- wykorzystując obraz  $I_2$  wyznacz gradienty  $I_x, I_y$ ,
- wyznacz sumę modułów  $m = |I_x| + |I_y|$ ,
- dla  $m \geq 0$  wyznacz pole prędkości  $v_n = -I_t / m$ ,
- dokonaj binaryzacji pola energii ruchu,
- opcjonalnie wyznacz środki ciężkości otrzymanego pola energii ruchu, tzn. współrzędne  $x, y$  przybliżające współrzędne środka ciężkości poruszającego się obiektu.

Wyznaczenie parametrów ruchu obiektów na podstawie obrazu, na którym zarejestrowano przemieszczenie więcej niż jednego obiektu, wymaga stosowania nieco bardziej złożonego podejścia. Przykładowo, obiekty poruszające się mogą być śledzone w oknie obrazu. Każdy z nich obejmowany jest wówczas przez niezależnie przemieszczane okno o wymiarach dobranych w ten sposób, by w razie osiągnięcia przez obiekt maksymalnych prędkości i przyspieszeń jego ucieczka poza granice okna nie była możliwa. Do predykcji położenia okna, a zarazem filtracji danych pomiarowych stosowany może być filtr Kalmana, którego budowę oparto na uproszczonych równaniach ruchu, przy założeniu stałej prędkości w okresie próbkowania. Przepływ optyczny może uzupełniać lub być uzupełniany przez inne sposoby przetwarzania

obrazu, w szczególności oparte na metodach momentów geometrycznych. Wydaje się on atrakcyjny, gdyż odnosi się do obiektów ruchomych o nieznanym kształcie. Ponadto stosunkowo dobrze nadaje się do zrównoleglenia. Można jednak wskazać wiele przyczyn, dla których metody przepływu optycznego nie udostępniają poprawnych wyników. Pierwsza grupa błędów wiąże się z leżącym u podstaw wszystkich metod przepływu optycznego założeniem o niezmienniej jasności korespondujących ze sobą punktów, ujętych przez kamerę na dwóch kolejnych obrazach. Podobnie zakłada się, że prędkość sąsiadujących punktów należących do poruszających się obiektów jest ta sama. Druga grupa błędów wynika z posługiwania się dyskretnymi aproksymacjami pochodnych dla danych wejściowych pochodzących z obrazów obarczonych szumem.

#### 4. Sprzętowa realizacja przepływu optycznego

Przedstawiony poniżej algorytm wyznaczania przepływu optycznego wykorzystuje dane pochodzące z półobrazów o wymiarach 256x256. Dane przetwarzane są na bieżąco w takt częstotliwości, z jaką przetwornik udostępnia treść obrazu na wyjściu. Sposób przetwarzania różni się zatem zasadniczo od algorytmu komputerowego, w którym w dyspozycji są całe półobrazy  $I_1$ ,  $I_2$ ,  $I_3$  zapamiętane w pamięci RAM.

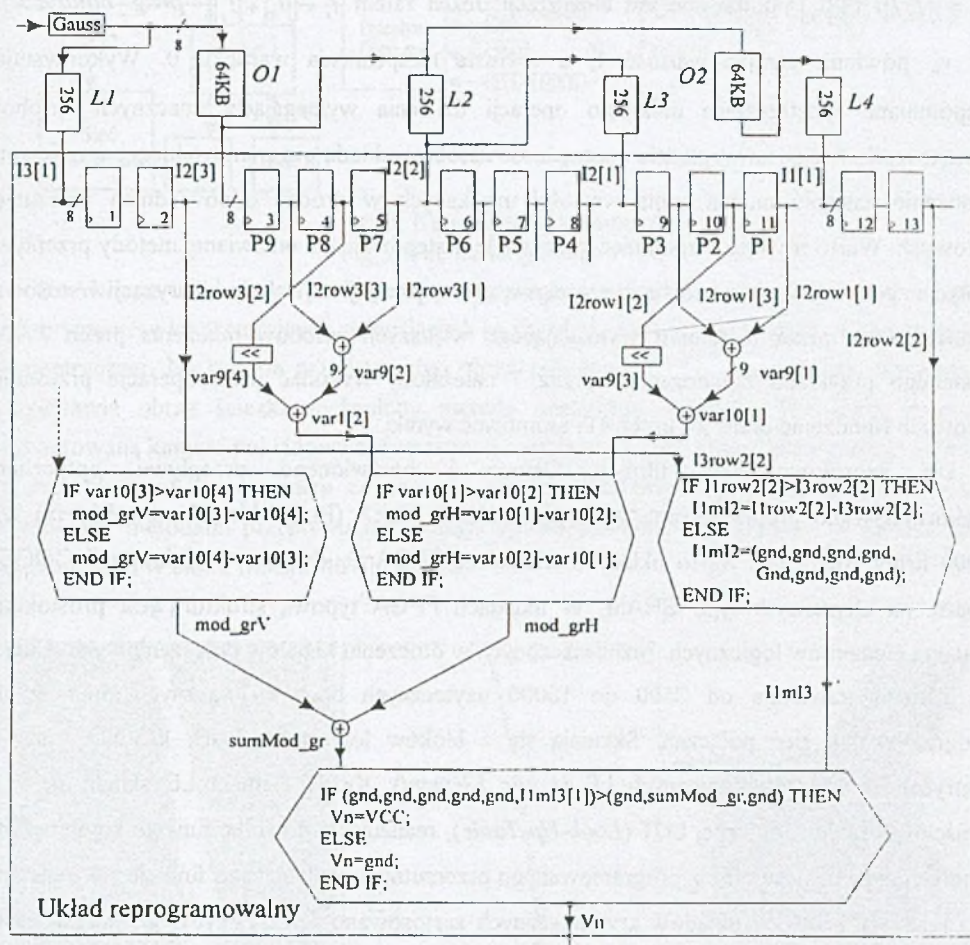


Rys.3. Wyznaczanie gradientu. a) oznaczenia przyjęte na rys. 4, b) maski Sobela  
Fig.3. Gradient determination. a) notation used on fig. 4, b) Sobel masks

Obraz pobierany na bieżąco z przetwornika obrazu poddawany jest filtracji Gaussa. Sposób zaprogramowania filtracji w układzie reprogramowanym przedstawiono w pracy [6]. Celem uzyskania treści składającej się na obrazy  $I_1$ ,  $I_2$  zastosowano dwie linie opóźniające o pojemności 256x256=64 KB każda. Wyjście linii opóźniającej  $O1$  udostępniającej treść obrazu  $I_2$  połączone jest z wejściem linii  $O2$  udostępniającej treść obrazu  $I_1$  (zob. rys.4). Jednocześnie treść obrazu udostępniana na wyjściu linii  $O1$  podawana jest na wejście 3 rejestrów udostępniających piksele P7, P8, P9, zob. rys. 3, składające się na obraz  $I_2$  (zob. rys.4). Po opóźnieniu w linii  $L2$  i kolejnych trzech rejestrach treści składającej się na obraz  $I_2$  otrzymuje się trzy dalsze piksele P4, P5, P6 wymagane w operacjach kontekstowych. Piksel P5 jest punktem środkowym maski i dla niego wyznaczany jest stosowny gradient. Do wymienionego piksela odnoszą się operacje realizowane na obrazach  $I_1$ ,  $I_3$ . Po dalszym opóźnieniu treści



obrazu  $I_2$  w linii  $L3$  oraz trzech rejestrach otrzymuje się piksele  $P1, P2, P3$ . Dzięki liniom  $L1$  i  $L4$  połączonym z rejestrami uzyskano wymagany piksel środkowy dla obrazów  $I_1, I_3$ .



Rys. 4. Procesor wyznaczający przepływ optyczny

Fig. 4. Optical flow processor implemented in hardware

Gradyenty wyznaczano posługując się maskami Sobela i Prewitta. Podany na rys. 4. schemat blokowy algorytmu przepływu optycznego odnosi się do maski Sobela (zob. rys.3). Moduł gradientu w kierunku poziomym<sup>1</sup> wyznaczany jest na podstawie następującej zależności:  $mod\_grH = \text{abs}([(I2row1[1] + I2row1[3]) + I2row1[2]*2] - [(I2row3[1] + I2row3[3]) + I2row3[2]*2])$ . Mnożenie przez dwa zrealizowane zostało w drodze przesunięcia bitowego. Omawianą operację zaznaczono na rys. 4 za pomocą symbolu <<. Pochodną  $I_x$  wyznaczono

<sup>1</sup> Na rys. 4. nie zaznaczono części schematu związanej z wyznaczeniem gradientu w kierunku pionowym.

posługując się treścią obrazów  $I_1$ ,  $I_3$  udostępnianą na wyjściu przerzutników 2 i 13 (zob. rys.4).

Jak wspomniano wcześniej, pole prędkości wyznaczone na podstawie wzoru  $v_n = -I_r / (|I_x| + |I_y|)$  poddawane jest binaryzacji. Jeżeli zatem  $I_r < (|I_x| + |I_y|) * \text{próg\_binaryzacji}$ , to  $v_n$  powinna przyjąć wartość 1, a w razie niespełnienia warunku 0. Wykorzystując wspomniane spostrzeżenie uniknięto operacji dzielenia wymagającej znacznych zasobów sprzętowych. Mając na względzie oszczędność zasobów układu programowalnego, wymienione mnożenie zastąpić można sumą wartości uzyskanych w drodze odpowiednich przesunięć bitowych. Warto również wspomnieć, że w wielu zastosowaniach omawianej metody przepływu optycznego można pogodzić się z częściową utratą precyzji w trakcie binaryzacji i stosować mnożenie np. przez 8 zamiast wymagającego większych zasobów mnożenia przez 7. Dla podanego przykładu z mnożeniem przez 7 należałoby wykonać dwie operacje przesunięć bitowych (mnożenie przez 2 i przez 4) i zsumować wyniki.

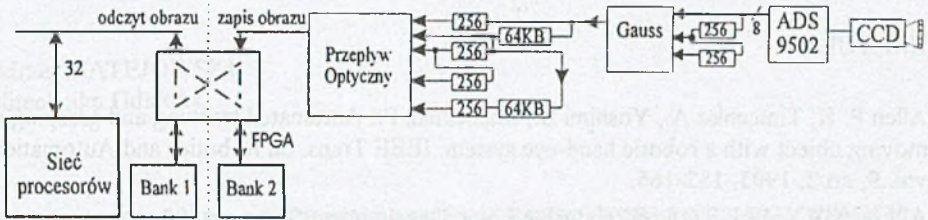
Do zaprogramowania filtracji Gaussa i omówionego przepływu optycznego wykorzystywano programowane w systemie układy FLEX (FLEXible Element Matrix) serii 8000 firmy Altera [2]. Są to układy o strukturze FPGA zbudowane w technologii CMOS i oparte na elementach typu SRAM. W układach FPGA typową strukturą jest prostokątna macierz elementów logicznych rozmieszczonych w otoczeniu kanałów połączeniowych. Układy tej rodziny zawierają od 2500 do 16000 użytecznych bramek logicznych oraz szybką programowalną sieć połączeń. Składają się z bloków logicznych LAB, które z kolei są macierzami elementów logicznych LE (*Logic Element*). Każdy element LE składa się z 4-węjsiowej tablicy logicznej LUT (*Look-Up Table*), realizującej dowolne funkcje kombinacyjne czterech argumentów oraz z programowanego przerzutnika realizującego funkcje sekwencyjne. Do realizacji szybkich układów arytmetycznych zastosowano ścieżki szybkiego przeniesienia. Mogą one łączyć sąsiednie komórki LE, bez potrzeby korzystania z połączeń globalnych. Komórki można łatwo łączyć w kaskady iloczynów lub sum, włączając komórki z sąsiednich bloków. W ten sposób możliwa jest realizacja funkcji o większej liczbie argumentów.

Omówiony algorytm zapisano w języku AHDL (zob. także fragmenty kodu zaznaczone na rys.4). Do zaprogramowania układów wykorzystywano system MAX+Plus II umożliwiający wykonywanie symulacji i weryfikacji rozwiązania.

Do zaprogramowania wybrano układ firmy Altera EPF8636, zawierający 6000 użytecznych bramek logicznych przy 636 elementach logicznych. Wspomniany projekt wykonano z myślą o wykorzystaniu w opracowanej karcie interfejsu kamery (zob. rys.5) zawierającej dwa banki pamięci umożliwiające równoległą pracę przetwornika obrazu i układu odczytującego dane do



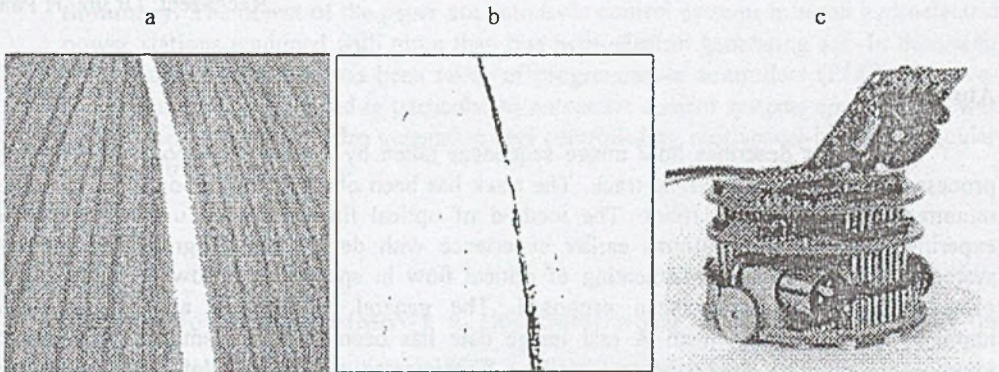
sieci procesorów. Zastosowane rozwiązania sprzętowe umożliwiły w szczególności przetwarzanie obrazów w czasie rzeczywistym.



Rys.5. Karta interfejsu kamery

Fig.5. The frame grabber

Omówiony algorytm przepływu optycznego wykorzystywano m.in. w trakcie eksperymentów laboratoryjnych polegających na śledzeniu przez robot mobilny Khepera ścieżki geometrycznej. Na rys. 6a przedstawiono obraz pobrany z kamery pokładowej. Rysunek 6b przedstawia obraz ścieżki wydzielony metodą omówioną w pracy. Fotografia robota z zamontowaną kamerą pokładową zamieszczona została na rys. 6c. Pojedyncze punkty widoczne na rys. 6b usuwano z obrazu za pomocą operacji morfologicznych [5]. Fragment ścieżki wydzielony metodami przepływu optycznego aproksymowano linią prostą i w dalszej części przetwarzano w bloku transformacji Hougha.



Rys. 6. Detekcja ścieżki geometrycznej. a) obraz pobrany z kamery pokładowej, b) wykryta ścieżka geometryczna, c) robot mobilny

Fig. 6. Detection of geometrical path. a) picture obtained from the onboard camera, b) detected geometrical path, c) mobile robot

## 5. Podsumowanie

W pracy omówiono sprzętową realizację algorytmu do wyznaczania przepływu optycznego. Przedstawiono szczegółowy schemat działania algorytmu w wersji dostosowanej do zaprogramowania w układzie reprogramowanym. Do zaprogramowania przepływu optycznego wykorzystywano język AHDL i układy programowalne firmy Altera. Rozwiązania sprzętowe w opracowanej karcie interfejsu kamery umożliwiają w szczególności synchronizację systemu



wizyjnego i kamery. Dzięki wspomnianemu rozwiązaniu możliwe jest sterowanie w czasie rzeczywistym. Układ przetwarza na bieżąco dane udostępniane na wyjściu przetwornika obrazu.

## LITERATURA

1. Allen P. K., Timcenko A., Yoshimi B., Michelman P.: Automated tracking and grasping of a moving object with a robotic hand-eye system. IEEE Trans. on Robotics and Automation, vol. 9, no.2, 1993, 152-165.
2. Altera, MAX+PLUS II Programmable Logic Development System, 1992.
3. Horn B. K. P.: Robot Vision, The MIT Press, McGraw-Hill Book Comp. Cambridge 1987.
4. Horn B. K. P., Schunck B. G.: Determining optical flow, Artificial Intelligence, 17, 1981, 185-203.
5. Kwolek B.: Zastosowanie układów reprogramowalnych do realizacji operacji morfologicznych w systemach wizyjnych robotyki, Mat. III KK Nauk. Reprogramowalne Układy Cyfrowe, Szczecin 2000, 27-34.
6. Kwolek B.: Zastosowanie obliczeń równoległych i układów programowalnych w robotyce, Mat. II Kraj. Konf. Metody i Systemy Komputerowe w Badaniach Naukowych i Projektowaniu Inżynierskim, Wyd. CCATIE, Kraków 1999, 525-530.

Recenzent: Dr inż. H.Palus

## Abstract

The paper describes how image sequences taken by a moving video camera may be processed to detect geometrical track. The track has been observed by onboard CCD camera mounted on the Khepera robot. The method of optical flow has been used in laboratory experiments. Using the author's earlier experience with design and programming real-time systems, the method of implementing of optical flow in specialised hardware based on the programmable FPGA has been proposed. The general architecture and the hardware implementation are described. A real image data has been used to demonstrate presented approach. The optical flow processor has been implemented using AHDL language. Because of small delays introduced in vision system as well as constant processing time, the presented system is especially suitable for real-time applications.