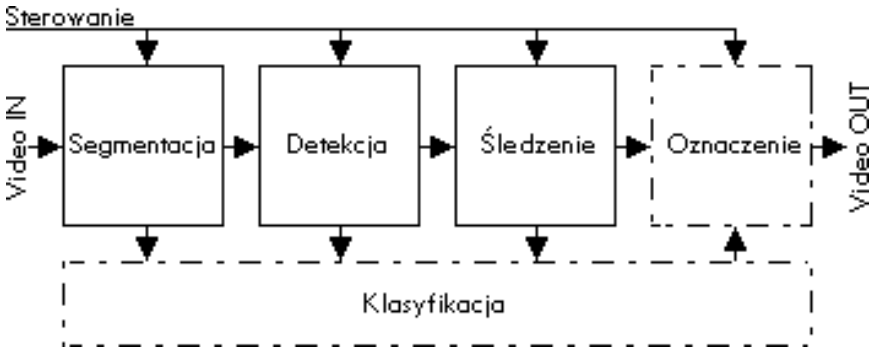
Damian Jackowski, Tomasz Marciniak, Adam Dąbrowski
Pracowania Układów Elektronicznych i Przetwarzania Sygnałów
Katedra Sterowania i Inżynierii Systemów
Politechnika Poznańska

IMPLEMENTACJA I ANALIZA MODELI DETEKCJI I ŚLEDZENIA OBIEKTÓW W SEKWENCJACH WIDEO Z ZASTOSOWANIEM MODUŁU Z PROCESOREM SYGNAŁOWYM

Streszczenie – W artykule przedstawiono analizę algorytmów detekcji i śledzenia obiektów w sekwencjach wideo zaimplementowanych z zastosowaniem bibliotek *Video and Image Processing Blockset*, *Target Support Package*, *Vision Library* oraz oprogramowania *Link for CCS* w środowisku Matlab / Simulink. Proces szybkiego projektowania i budowy modeli przetwarzania sekwencji wideo zrealizowano dla modułu laboratoryjnego EVM z procesorem sygnałowym DM6437.

1 Wprowadzenie do algorytmów detekcji i śledzenia obiektów

Wykrywanie obiektów w sekwencjach wideo składa się z następujących etapów: segmentacji, detekcji, śledzenia oraz ewentualnego oznaczenia obiektów (Rys. 1). Etapy te w istniejących rozwiązaniach systemów detekcji są często ściśle powiązane ze sobą. Metody detekcji można podzielić na algorytmy wykrywania ruchu, detekcji obiektów oraz śledzenia.



Rys. 1. Elementy modelu detekcji i śledzenia obiektów

Algorytmy wykrywania ruchu bazują na równaniu przepływu optycznego, metodach dopasowania, metodach różnicowych oraz analizie konturów [1]. Rezultatem ich działania jest binarny obraz, który jest następnie poddawany analizie „blob".

Cechą algorytmów detekcji obiektów jest analiza z wykorzystaniem pojedynczej ramki. Ich działanie jest możliwe dzięki wcześniej zbudowanym wzorcom lub określonym kryteriom. Do tej grupy algorytmów należą: metoda mapy krawędzi, analiza kształtu i barwy, metody oparte na wykrywaniu podobieństw ze wzorcami oraz klasyfikatory dokonujące analizy cech obiektów.

Algorytmy śledzenia obiektów są wykorzystywane jako uzupełnienie pozostałych algorytmów. Mogą bazować na metodach obliczania maksymalnego przesunięcia wykrytego obiektu pomiędzy kolejnymi ramkami sekwencji lub zachowaniu barwy i kształtu wykrytego obiektu reprezentowanych przez odpowiednio współczynniki barwy i kształtu. Często w celu poprawy osiąganych wyników stosuje się filtrację Kalmana.

W zależności od zastosowanego algorytmu możliwa jest również klasyfikacja. W przypadku ściśle określonych warunków spełnianych przez obserwowaną scenę możliwa jest klasyfikacja na podstawie wymiarów bądź barwy obiektu. W przypadku algorytmów wykrywających, klasyfikacja jest dokonywana bezpośrednio jako etap działania algorytmu.

Zastosowanie kilku algorytmów i zbudowanie wielopoziomowego systemu analizy sekwencji wejściowej mogłoby być skutecznym rozwiązaniem dla systemu wizyjnego do detekcji i śledzenia. Rozwiązanie to jest jednak trudne w realizacji oraz silnie zależne od zastosowanej platformy sprzętowej, na której działa system.

Duży wpływ na szybkość działania algorytmu mają parametry sygnału wejściowego takie jak szybkość próbkowania i rozdzielczość przetwarzanych ramek wejściowych. Parametry te powinny zostać odpowiednio dobrane do budowanego systemu i stosowanych algorytmów.

2 Przygotowanie modelu z zastosowaniem biblioteki *Video and Image Processing Blockset*

W celu zbudowania modelu przetwarzającego sekwencje wizyjne w środowisku Matlab / Simulink można zastosować bloki z biblioteki Video and Image Processing Blockset (VIPB) [2]. Biblioteka jest podzielona na grupy bloków według spełnianych funkcji:

- *Analysis & Enhacement* - wyznaczanie jednolitych obszarów, wykrywanie krawędzi i narożników, filtr medianowy, przepływ optyczny, dopasowanie bloków
- *Conversion* - konwersje pomiędzy przestrzeniami barw, zmiana typu danych, autoprogowanie, korekcja gamma

- *Filtering* – filtracja typu FIR i Kalmana
- *Geometric Transformations* - transformacje obrazu, pochylanie, obrót, przesunięcie, zmiana rozmiaru
- *Morphological Operations* - dylatacja, erozja, operacje otwarcia, domknięcia
- *Sinks* - wyświetlanie obrazów, reprezentacja wyników przetwarzania
- *Sources* - źródła obrazów i sekwencji wideo
- *Statistics* - wyliczanie histogramów, wyznaczanie minimum i maksimum, analiza blob, uśrednianie
- *Text & Graphics* - wstawianie tekstu i grafiki
- *Transforms* – DCT, FFT, piramida Gaussowska, HT.

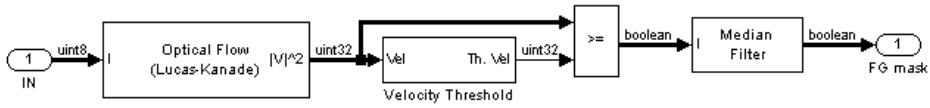
Podczas badań eksperymentalnych jako model bazowy wykorzystano model *People Tracking* z biblioteki *Video and Image Processing Blockset*. Model ten do przetwarzania wykorzystuje szybką metodę różnicową oraz analizę *Blob*.

Obrazem referencyjnym jest reprezentacja tła wyliczana na podstawie pierwszych ramek sekwencji, przykładowo 20. Liczbę ramek do obliczenia reprezentacji tła można dobrać dowolnie konfigurując model, umożliwia to blok *N-Sample Enable*. Podczas wyznaczania reprezentacji tła algorytm przetwarzania jest wyłączony. Do poprawy obrazu różnicowego użyto operacji morfologicznej domknięcia.

W etapie segmentacji poprawę wyników uzyskano poprzez zastosowanie autoprogowania (blok: *Autotreshold*). Proces autoprogowania dokonuje automatycznego dopasowania progu do obrazu wejściowego na podstawie analizy histogramu.

Algorytm pozwala nie tylko na detekcję ruchomych obiektów, ale również na śledzenie tego samego obiektu w kolejnych ramkach sekwencji wideo na podstawie współrzędnych wykrytego obiektu.

Dane wyjściowe dla danego obiektu (położenie, rozmiar) są poddawane filtracji Kalmana. Następnie dzięki blokowi *Draw Rectangles* i *Insert Text*, obiekty są oznaczane prostokątami i literą w górnym rogu każdego prostokąta na obrazie wyjściowym.

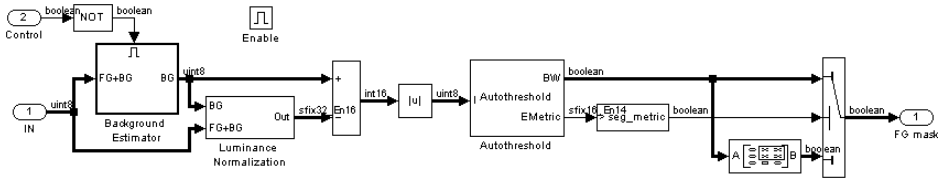


Rys. 2. Struktura *Segmentation* dla przepływu optycznego

W wyniku prac analitycznych oraz wykonanych eksperymentów w opracowanym i zastosowanym modelu zmodyfikowano źródło sekwencji wejściowej oraz sterowanie procesem segmentacji. Testowano trzy al-

gorytmy dokonujące wykrywania ruchu: przepływ optyczny (metody Lucas-Kande oraz Horn-Schunck - Rys. 2), dopasowanie bloków (dla wielkości bloków 5×5, 11×11, 17×17) oraz metodę różnicową.

Metodę różnicową przetestowano dla dwóch konfiguracji: dla reprezentacji tła wyliczonej jednorazowo oraz z ciągłą estymacją obrazu tła. Jej realizacja może zostać dokonana za pomocą gotowych bloków *Median Estimator Temporal Median*, można skorzystać ze zbudowanego modelu *Background Estimator* (Rys. 3)



Rys. 3. Struktura *Segmentation* dla metody różnicowej

Szybkość przetwarzania w ramkach na sekundę (FPS – *frame per second*) dla poszczególnych algorytmów zestawiono w Tabeli 1. W przypadku algorytmu *Block Matching* szybkość działania nie pozwalała na poprawne działanie modelu.

Z przeprowadzonych testów wynika, że najlepszym rozwiązaniem ze względu na prostotę implementacji, stosunkowo niskie wymagania obliczeniowe, a tym samym większą szybkość działania jest zastosowanie metody różnicowej. Metoda ta pozwala na wykrywanie ruchu na podstawie wykrywania różnic pomiędzy kolejnymi ramkami a modelowanym reprezentacją tła stanowiącego obraz referencyjny.

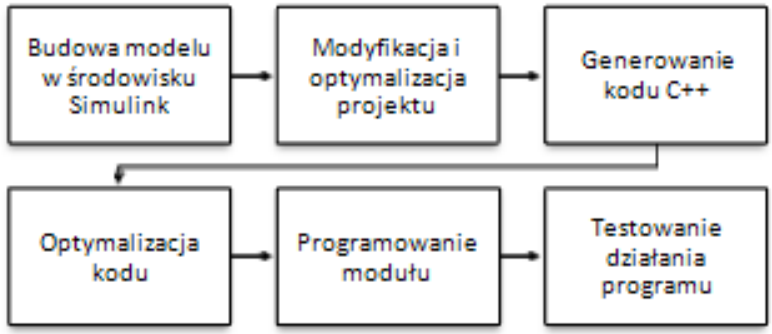
Tabela. 1. Wpływ wyboru metody wykrywania obiektów na FPS

Algorytm	**FPS**
Block Matchnig	0.7
Optical Flow	1.0
Metoda różnicowa z estymacją	3.2
Metoda różnicowa bez estymacji	6.8

3 Dostosowanie modelu do modułu DM6437 EVM z zastosowaniem biblioteki *Vision Library*

Opracowany w punkcie drugim model detekcji obiektów został dostosowany do wymagań pracy z modułem zawierającym procesor sygnałowy DM6437 [3].

Modyfikację modelu umożliwia wsparcie oprogramowania *Matlab/Simulink* dla środowiska *Code Composer Studio* (CCS) [4], w postaci specjalnie dedykowanej biblioteki *Target Support Package* (TSP) zawierającej bloki obsługujące peryferia modułu oraz oprogramowanie *Real-Time Workshop for Embedded IDE Link for CSS*, które pozwala na automatyczną budowę projektu i generację kodu programu w języku C dla środowiska CCS, programowanie procesora oraz sterowanie pracą programu [5].



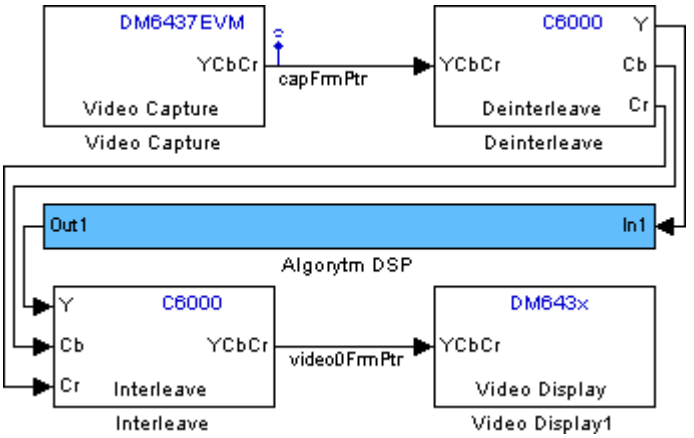
Rys. 4. Etapy budowy programu procesora

Etapy przygotowania oprogramowania dla modułu z procesorem sygnałowym pokazuje diagram na rys. 4. Ogólny schemat wykorzystany do budowy programu z zastosowaniem biblioteki TSP pokazano na rys. 5.

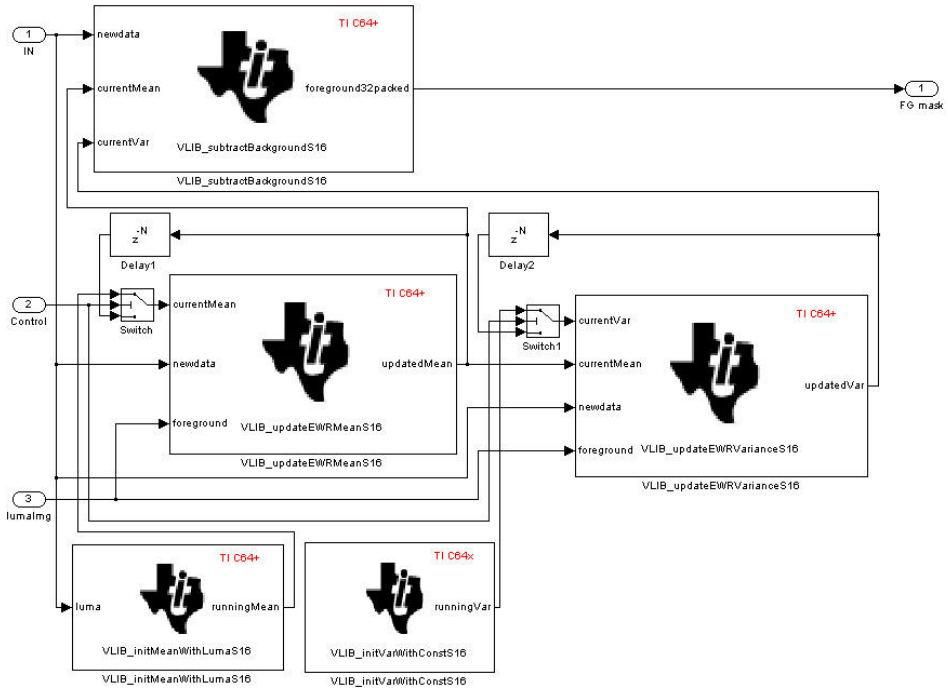
Do budowy modeli detekcji śledzenia obiektów można także zastosować specjalnie dedykowane biblioteki dostarczane przez producenta modułu, czyli firmę *Texas Instruments* (TI). W przypadku dla systemów wizyjnych możliwe jest korzystanie z biblioteki *Vision Library* (VLIB) [6].

VLIB jest biblioteką wspomagającą przetwarzanie obrazów lub sekwencji obrazów dedykowana dla procesorów z rodziny c64x. Biblioteka ta zawiera 40 funkcji napisanych w języku C oraz bloki umożliwiające zastosowanie ich w środowisku Matlab/Simulink. Do najważniejszych funkcji biblioteki należą: modelowanie i odejmowanie reprezentacji tła, ekstrakcja cech wykrytych obiektów, śledzenie i rozpoznawanie obiektów, niskopoziomowe przetwarzanie pikseli obrazu. Ważną cechą biblioteki jest zoptymalizowane działanie na laboratoryjnych modułach wideo

wyposażonych w procesor DSP firmy TI [6]. Dostosowanie struktury *Segmentation* do działania z modułem DM6437 EVM pokazano na rys. 6.



Rys. 5. Schemat modelu z zastosowaniem bloków z biblioteki *Target Support Package*



Rys. 6. Struktura *Segmentation* dla metody różnicowej z zastosowaniem biblioteki VLIB

Ekstrakcji tła dokonano z wykorzystaniem metody *Exponentially-Weighted Running Mean* i *Exponentially-Weighted Running Variance*. Użycie takich technik umożliwiają funkcje: *VLIB_subtract-BackgroundS16*, VLIB_updateEWRMeanS16, *VLIB_updateEWR-VarianceS16*, a do ich inicjalizacji niezbędne są bloki *VLIB_initMeanWithLumaS16*, *VLIB_initVarWithConstS16* z biblioteki VLIB.

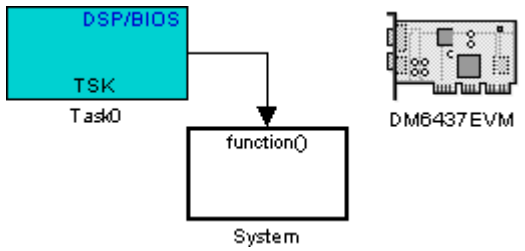
Do modułu dołączono kamerę pracującą w standardzie PAL jako źródło sekwencji wejściowej oraz monitor TV do wyświetlania wyników działania systemu (Rys. 7). Model systemu został zapisany jako funkcja wywoływana dla procesora DM6437 (Rys. 8), co pozwala na przetwarzanie z maksymalną szybkością.

Do akwizycji i prezentacji sygnału wideo model wykorzystuje bloki *Video Capture* i *Video Display* z biblioteki TSP. Sygnał wizyjny z bloku wejściowego jest w reprezentacji YCbCr 4:2:2 o wymiarach 720 × 576 pikseli. Konfigurowalne parametry to: szybkość próbkowania, wybór pomiędzy systemem PAL lub NTSC oraz wybór wejścia sygnału (*Composite* lub *Component*). Istotnym elementem modelu jest separacja składowych sygnału oraz ich ponowne scalenie po przetworzeniu. Służą do tego bloki *Deinterleave* oraz *Interleave*.

Rys. 7. Prezentacja wyniku etekcji obiektu ruchomego

Przetwarzaniu zostaje poddany jedynie sygnał luminancji, gdyż zawiera pełnowymiarową ramkę w reprezentacji *Intensity*, sygnały chromi-

nancji pozostają natomiast niezmienione. Przetwarzanie barwnej reprezentacji wymaga zastosowania konwersji pomiędzy przestrzeniami barw, co niestety powoduje spowolnienie algorytmu. Działanie algorytmu w czasie rzeczywistym z użyciem modułu DM6437 EVM wymagało także redukcji wymiarów przetwarzanej ramki wejściowej [7].



Rys. 8. Schemat modelu

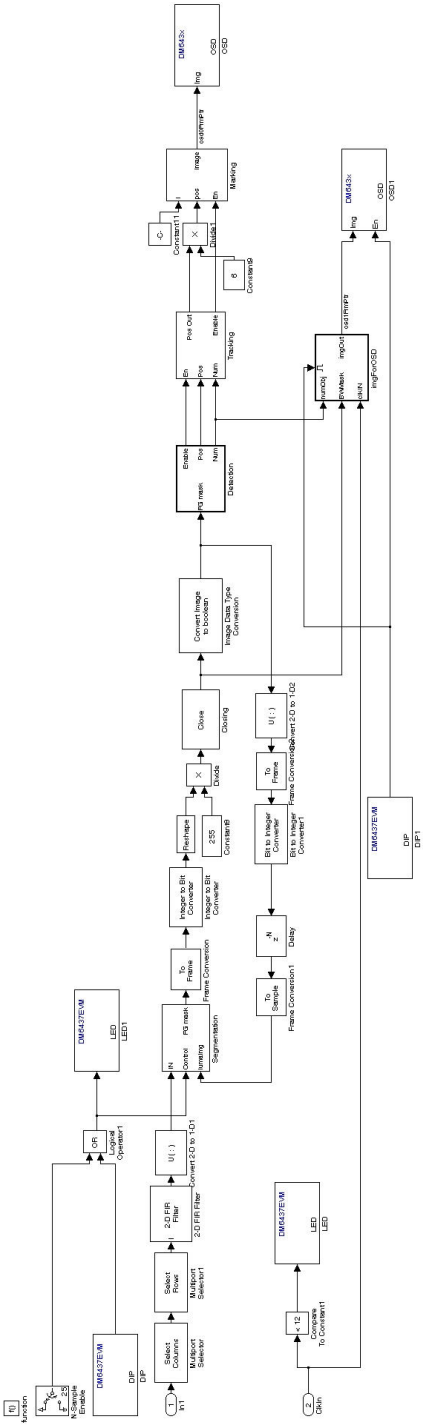
Model zawiera również blok DIP pozwalający na włączanie i wyłączanie estymacji tła podczas działania programu oraz bloki LED do sygnalizacji pojawienia się obiektu oraz przekroczenia wartości progowej powierzchni obiektów. Do prezentowania liczby obiektów oraz liczby przetworzonych ramek wykorzystano *On Screen Display* (OSD) (Rys. 7) [8]. Główną strukturę modelu przedstawia rys. 9.

Tabela. 2. Analiza szybkości przetwarzania i zajętości procesora

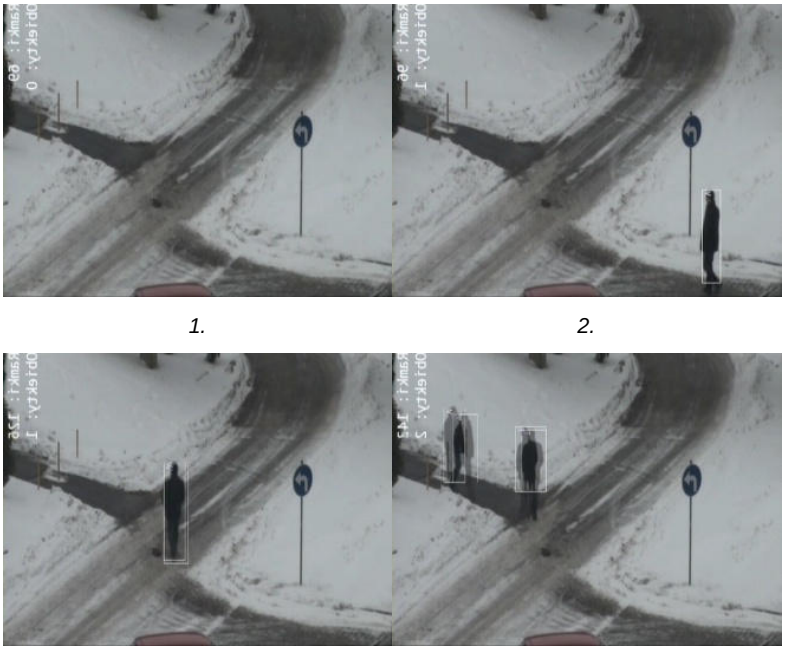
Metoda różnicowa	**FPS**	**CPU Load**
z zastosowaniem VIPB	>10	100 %
z zastosowaniem VLIB	24	~50 %

Testowanie rozpoczęto od modelu pełniącego funkcję *Loopback* [3], a więc bez zastosowania algorytmu przetwarzania oraz separacji składowych sygnałów. Pomiaru dokonywano mierząc czas przetworzenia 25 ramek. Tak zbudowany model osiąga 25 ramek na sekundę, zajętość procesora wynosi około 1,50 %.

Następnie testowaniu poddano algorytmy detekcji oraz śledzenia obiektów. Wyniki dla testu z wykorzystaniem bibliotek VIPB i VLIB (Rys. 9) zestawiono w Tabeli 2. Rysunek 10 prezentuje przykładowe rezultaty działania oprogramowania.



Rys. 9. Model detekcji i śledzenia obiektów dla modułu DM6437 EVM



1. *2.*

3. *4.*

Rys. 1. Przykładowa sekwencja z detekcją i śledzeniem obiektu

4 Podsumowanie

Celem artykułu było zaprezentowanie oprogramowania do detekcji i śledzenia obiektów w sekwencji wideo przy użyciu modułu EVM z procesorem sygnałowym DM6437, zaprogramowanego w środowisku Matlab / Simulink.

Dokonano wyboru algorytmu i przygotowano efektywny model analizy z zastosowaniem metody różnicowej, którą poddano modyfikacji w celu poprawy szybkości przetwarzania. Opracowany model zaimplementowano wykorzystując oprogramowanie Link for CCS. Działający w czasie rzeczywistym program pozwala na przetwarzanie sekwencji wejściowej z szybkością około 25 ramek na sekundę (FPS).

Na podstawie otrzymanych wyników można stwierdzić, że wykorzystanie środowiska MATLAB / Simulink pozwala na szybkie tworzenie, modyfikowanie i testowanie oprogramowania dla systemów wbudowanych z procesorami sygnałowymi.

Przeprowadzone eksperymenty pokazały także, że kody uzyskane za pomocą środowisk Matlab / Simulink i CCS wymagają optymalizacji w

celu przyspieszenia działania i poprawy funkcjonalności realizowanego systemu.

Literatura

[1] Jackowski D., Detekcja i śledzenie obiektów w sekwencjach wideo z wykorzystaniem modułu DM6437 DVDP, praca magisterska, Politechnika Poznańska, 2009.

[2] Video and Image Processing Blockset™ 2 User's Guide, The MathWorks, Inc., 2009.

[3] Texas Instruments, TMS320DM6437 DVDP Getting Started Guide, 2007.

[4] Code Composer Studio Development Tools v3.3 Getting Started Guide, Texas Instruments, 2006.

[5] Target Support Package™ TC6 3 User's Guide, The MathWorks, Inc., 2009.

[6] Texas Instruments, Reference Guide: Vision Library (VLIB) Application Programming Interface, November 2009.

[7] Marciniak T., Jackowski D., Pawłowski P., Dąbrowski A., Dobór parametrów modelu śledzenia we wbudowanym systemie wideo, *Elektronika*, 2010.

[8] He Z., How to Use the VPBE and VPFE Driver on TMS320DM643x Devices, 2007.

W pracy zaprezentowano wyniki osiągnięte w projektach PPBW, INDECT i DS.

IMPLEMENTATION AND ANALYSIS OF MODELS FOR DETECTION AND TRACKING OBJECTS IN VIDEO SEQUENCES USING DIGITAL SIGNAL PROCESSING MODULE

Summary – This paper presents the analysis of algorithms for detecting and tracking objects in video sequences using *Video and Image Processing Blockset*, *Target Support Package*, *Vision Library* and software *Link for CCS* in Matlab / Simulink environment. The rapid prototyping process of the video processing models was realized using the evaluation module EVM with the DM6437 digital signal processor.