

Śledzenie obiektów dynamicznych z wykorzystaniem metod inteligencji obliczeniowej – implementacja sprzętowa

Przemysław Błaszowski, Michał Grochowski

Wydział Elektrotechniki i Automatyki, Politechnika Gdańska

Streszczenie: W artykule przedstawiono platformę sprzętową umożliwiającą detekcję, rozpoznawanie i śledzenie poruszających się obiektów przy wykorzystaniu technik inteligencji obliczeniowej. Projekt obejmuje budowę platformy sprzętowej wyposażonej w kamerę internetową oraz silniki umożliwiające jej ruch. Analiza obrazu odbywa się w komputerze klasy PC, komunikującym się z platformą za pomocą portu szeregowego USB. Jako przykład wykrywanych i śledzonych obiektów dynamicznych wykorzystano ludzkie twarze. Algorytmy obliczeniowe napisano w języku C++. Platforma doskonale integruje środowiska obliczeniowe typu *open source*, umożliwiając testowanie w praktyce opracowanych algorytmów.

Słowa kluczowe: przetwarzanie obrazu, rozpoznawanie twarzy, klasyfikator Haara, detekcja, sztuczne sieci neuronowe

1. Wprowadzenie

Nowoczesne systemy monitoringu, obejmujące zagadnienia detekcji ruchu czy identyfikacji użytkowników, są jedną z najbardziej dynamicznie rozwijających się gałęzi automatyki i informatyki, często określane mianem wizji aktywnej [1–2]. Analiza obrazu z kamery daje szeroką gamę możliwości, od wyszukiwania w trybie on-line osób czy obiektów (np. skradzionych samochodów), identyfikacji biometrycznej, aż po inteligentny monitoring wykrywający niebezpieczne zdarzenia [3]. Na rynku coraz częściej można znaleźć kolejne rozwiązania inteligentnych systemów realizujących detekcję i śledzenie potencjalnych zagrożeń. Pośród zróżnicowanych aplikacji wciąż można dostrzec jednak wspólną cechę – wysokie ceny, co sprawia, że na ogół nie znajdują one zastosowania w zwykłych systemach alarmowych. Opisywana platforma realizuje rozpoznawanie i śledzenie zadanych obiektów za pomocą obrazu z ruchomej kamery, wykorzystując i integrując oprogramowanie typu *open source*. Głównymi założeniami w trakcie jej tworzenia, były: zdolność do działania w czasie rzeczywistym z akceptowalną skutecznością rozpoznawania oraz utrzymywanie śledzonego obiektu w polu widzenia kamery, przy niskim koszcie budowy urządzenia. Wraz z systemem informatycznym platforma stanowi doskonały obiekt dydaktyczny umożliwiający testowanie wielu algorytmów przetwarzania obrazu, identyfikacji oraz metod sztucznej inteligencji.

Opis platformy można podzielić na dwie części: budowa platformy sprzętowej realizującej ruch kamery oraz

oprogramowanie i algorytmy służące analizie obrazu. Aby zapewnić dużą szybkość działania oprogramowania platformy, aplikację opracowano w języku C++. Do realizacji zadań detekcji i rozpoznawania obiektów wykorzystano metody inteligencji obliczeniowej, a mianowicie klasyfikator Haara [4, 7], zaimplementowany z użyciem funkcji biblioteki OpenCV, oraz sztuczne sieci neuronowe – stosując elementy biblioteki FANN (*Fast Artificial Neural Networks*) [8]. Do szybkiego prototypowania posłużyło środowisko MATLAB [9], w którym opracowano i dobrano najlepsze struktury i algorytmy, a następnie wyniki prac zaimplementowano w C++, znacznie skracając czas wykonywania obliczeń poszczególnych algorytmów. Jako przykład detekcji, rozpoznawania i śledzenia poruszających się obiektów, wykorzystano ludzkie twarze. Proponowane algorytmy umożliwiają również rozpoznawanie dowolnych, wyuczonych wcześniej obiektów.

Projekt jest efektem realizacji pracy inżynierskiej przez inż. Przemysława Błaszowskiego, pod kierunkiem dr. Michała Grochowskiego na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej.

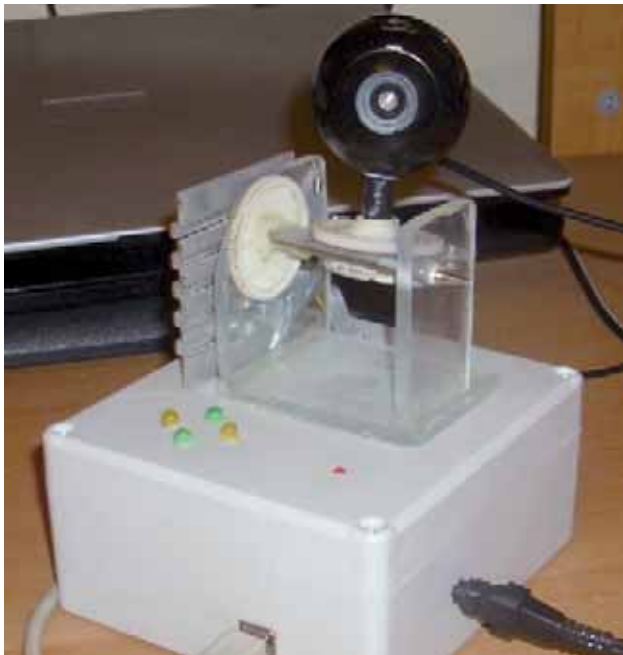
2. Platforma sprzętowa

Skonstruowaną platformę przedstawiono na rys. 1. Sercem układu jest mikrokontroler ATmega8, który komunikuje się z komputerem za pośrednictwem magistrali USB. W tym celu zastosowano konwerter USB UART: FT232. Założono, że do śledzenia celu wystarczą ruchy kamery wokół własnej osi oraz sterowanie w górę i w dół za pomocą drugiej osi prostopadłej, przylegającej do płaszczyzny podstawy.

Jako układy wykonawcze zastosowano dwa silniki skokowe bipolarne, sterowane przez mikrokontroler za pośrednictwem układów mocy L298N. Układ zasilany jest napięciem +12 V DC podawanym z zasilacza stabilizowanego. Schemat elektryczny systemu przedstawiono na rys. 2.

ATmega8 jest 8-bitowym mikrokontrolerem małej mocy CMOS, zbudowanym w architekturze AVR RISC. Przy budowie platformy wykorzystano 15 wejść/wyjść cyfrowych do sterowania silnikami oraz sygnalizacją, oraz port szeregowy USART do komunikacji z komputerem. Ustawiono tryb z oscylatorem wewnętrznym RC 8 MHz. Programowanie pamięci programu zrealizowano za pomocą magistrali szeregowej ISP.

Ze względu na brak implementacji w strukturze mikrokontrolera sterowników do obsługi USB, konieczne



Rys. 1. Urządzenie sterujące ruchami kamery
Fig. 1. The control device for camera movements

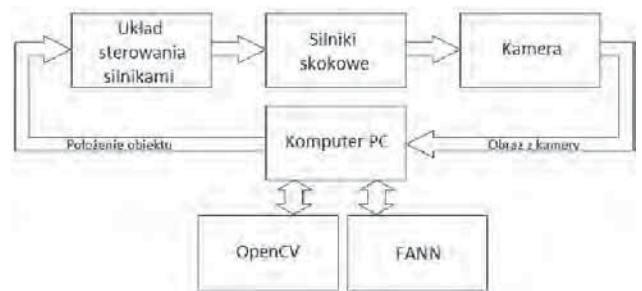
było zastosowanie dodatkowego konwertera między UART a USB. Wykorzystano układ FTDI – FT232R. Z poziomu komputera możliwa jest instalacja darmowych sterowników do obsługi interfejsu jako wirtualny port COM, co (z punktu widzenia programisty) znacznie upraszcza przesyłanie danych.

Za poruszanie kamerą odpowiadają silniki skokowe bipolarnie. Sterowanie kierunkiem ruchu odbywa się przez odpowiednie zadawanie sekwencji impulsów. Naj-

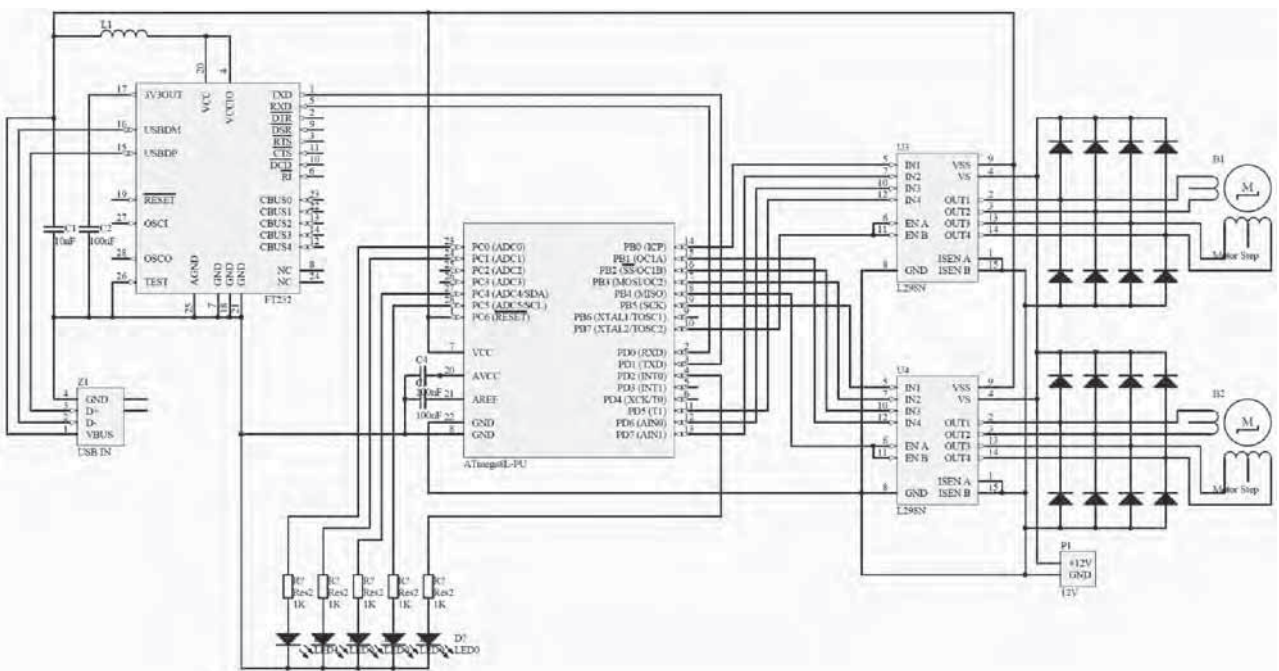
większą zaletą silników skokowych jest możliwość sterowania w pętli otwartej. Do zadawania precyzyjnego położenia wystarczy obliczyć potrzebną liczbę impulsów, bez informacji od sprzężenia zwrotnego. Zbędne stają się zatem wszelkie pomiary prędkości, których cena w wypadku sterowania klasycznych napędów elektrycznych zazwyczaj nie jest mała. Wykorzystano metodę sterowania półkrokowego – w jednej chwili zasilane są naprzemiennie jedno i dwa uzwojenia, co pozwala zmniejszyć krok dwukrotnie przy zachowaniu pełnej mocy. Cykl składa się z ośmiu półkroków.

Do sterowania silnikami skokowymi wykorzystano dwa układy L298N. Są to podwójne pełnomostkowe sterowniki zaprojektowane do współpracy z poziomami logicznymi standardu TTL, do sterowania obciążeniami indukcyjnymi, takimi jak przekaźniki, cewki powietrzne, silniki prądu stałego, czy silniki skokowe.

Ogólny schemat działania układu pokazano na rys. 4. Przesyłanie informacji na temat położenia śledzonego obiektu poprzedzono ustaleniem konkretnych reguł komunikacji. Ustalono, że każdy przetwarzany obraz zostanie



Rys. 3. Ogólny schemat działania
Fig. 3. General scheme of the operations



Rys. 2. Schemat ideowy układu sterowania
Fig. 2. Control system schematics



Rys. 4. Widok układu sterowania
Fig. 4. View of the control system

	0	40	80	120	160	200	240	280	320
40	a	f	k		p	t	y	D	
80	b	g	l		q	u	z	E	
120	c	h	m	Strefa nieczułości		v	A	F	
160									
200	d	i	n		r	w	B	G	
240	e	j	o		s	x	C	H	

Rys. 5. Matryca sterowań
Fig. 5. Control matrix

podzielony na 35 pól i do mikrokontrolera będzie przesyłana informacja o tym, w którym z tych pól znajduje się rozpatrywany obiekt. Dodatkowo została ustalona pewna strefa nieczułości, obejmująca środkowy fragment obrazu. Śledzony obiekt nie musi znajdować się idealnie w środku pola widzenia kamery, tolerowany jest błąd w stanie ustalonym rzędu 12,5%. Na rys. 5 została przedstawiona siatka (matryca) sterowań, zgodnie z którą ustalane jest przybliżone położenie obiektu na analizowanym obrazie o rozdzielczości 320×240 .

Przykładowo, jeśli obiekt wykryty podczas analizy obrazu znajduje się w punkcie (246, 53), to do mikrokontrolera zostanie przesłana informacja o tym, że obiekt znajduje się w obszarze „z”. Dla wszystkich 35 pól dobrano proste sterowania typu: obszar „z” – 4 kroki w lewo, 2 kroki w górę, uzyskując tym samym, tabelę sterowań (ang. *look up table*) pozwalającą na szybkie przemieszczanie się kamery we wskazane miejsce.

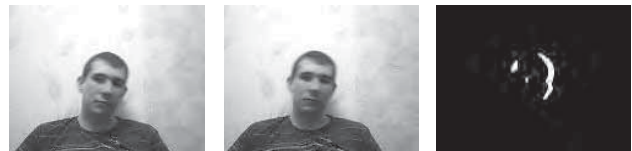
3. Schemat działania algorytmu

Detekcję i rozpoznawanie twarzy z wykorzystaniem technik inteligencji obliczeniowej zaimplementowano w języku C++ (Microsoft Visual C++ Express Edition). Programowanie w języku niższego poziomu pozwala na znacznie większy dostęp do poszczególnych funkcji, a co za tym idzie, znaczną redukcję czasu wykonywania algorytmów. Do podstawowych algorytmów przetwarzania obrazu oraz implementacji klasyfikatora Haara [4, 7] posłużyła biblioteka OpenCV 2.3 [5, 6]. Sieci neuronowe zostały zaimplementowane przy użyciu biblioteki FANN 2.1.0 beta [8].

Urządzenie może pracować w jednym z dwóch podstawowych trybów: wykrywanie ruchu oraz śledzenie zdefiniowanych obiektów (twarzy).

Wykrywanie ruchu

Śledzenie ruchu polega na znajdowaniu różnic dwóch kolejnych klatek, posługując się wyłącznie prostymi operacjami, takimi jak binaryzacja, erozja czy dyatacja (rys. 6).



Rys. 6. Metoda bazująca na różnicy obrazów
Fig. 6. Method based on the images differences

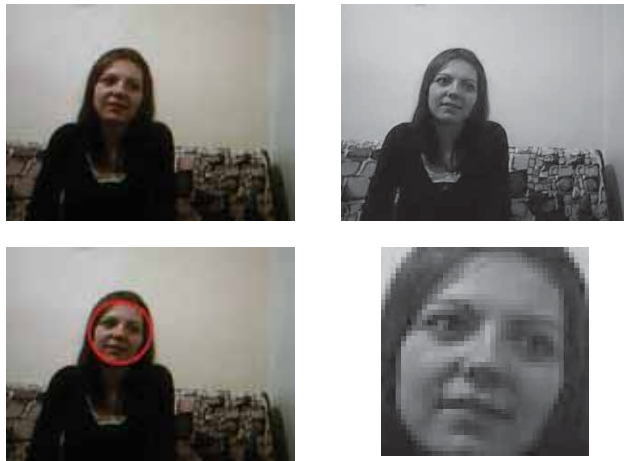
Wczytywany obraz jest odpowiednio skalowany oraz filtrowany. Program w każdym cyklu pobiera z kamery klatkę, pamiętając jednocześnie klatkę z cyklu poprzedniego. Po obliczeniu różnicy ich odjęciu od siebie uzyskuje się obraz zmian. Na podstawie binarnej reprezentacji wyznaczany jest środek ciężkości największego poruszającego się obiektu i w tym kierunku kierowana jest kamera.

Śledzenie twarzy

Algorytm śledzenia wybranych wcześniej obiektów (w tym przypadku twarzy) przedstawiono na rys. 7.



Rys. 7. Ogólny schemat przepływu informacji
Fig. 7. General flowchart of the data processing



Rys. 8 Efekt działania klasyfikatora Haara
Fig. 8. Result of Haar Classifier operating

Pobierane z kamery internetowej obrazy poddawane są wstępnej obróbce a następnie kierowane do właściwych algorytmów detekcji i rozpoznawania twarzy, klasyfikatora Haara oraz sztucznych sieci neuronowych. Klasyfikator Haara został nauczony wykrywania na obrazie twarzy (rys. 8). Skorzystano z szablonów detekcji twarzy dostępnych w bibliotece OpenCV. Ich wadą jest wykrywanie jedynie obiektów ustawionych frontalnie i pionowo w kierunku kamery. Aby zniwelować tą wadę zaimplementowano algorytm, który w przypadku niewykrycia na obrazie twarzy, obraca obraz o pewien kąt (np. $\pm 30^\circ$) i ponownie przeszukuje. W przypadku braku twarzy ponownie obracamy obraz itd. Dzięki temu w znacznym stopniu poprawiono skuteczność detekcji twarzy.

Wykrycie twarzy za pomocą klasyfikatora Haara pozwala na przejście do kolejnego etapu algorytmu, a mianowicie do jej identyfikacji. W tym procesie wykorzystano sztuczne sieci neuronowe, doskonale nadające się do zadań tego typu. Przy konstruowaniu sieci neuronowej konieczne było znalezienie kompromisu między zachowaniem odpowiedniej liczby szczegółów charakteryzujących twarz a możliwie małym nakładem obliczeniowym, ze względu na ograniczony czas działania. Ostatecznie wybrano zdjęcia w skali odcieni szarości, o rozdzielczości 40×40 . Projektowana sieć miała zatem 1600 wejść, z których każde mogło przyjąć jedną z 256 wartości. Liczba wyjść sieci zależy od liczby rozpoznawanych twarzy. Dodatkowo wyprowadzone zostało jedno wyjście, którym sieć sygnalizuje nierozpoznanie żadnego z wzorców twarzy. W projekcie wykorzystano sieć dwuwarstwową jednokierunkową, w warstwie ukrytej usytuowano 100 neuronów, natomiast liczba wyjść była zależna od liczby rozpoznawanych wzorców (np. 10

w przypadku rozpoznawania 9 osób). W obu warstwach przyjęto sigmoidalne funkcje przejść.

Aby sieć neuronowa była zdolna do rozpoznawania twarzy musi być wcześniej odpowiednio nauczona. Na rys. 9 przedstawiono zbiór uczący składający się ze 100 próbek dla dwóch rozpoznawanych osób. W celu usprawnienia procesu akwizycji danych uczących, w systemie zaimplementowano moduł automatycznego pobierania obrazów.

Zebranie danych do uczenia sztucznej sieci neuronowej jest realizowane po nakierowaniu kamery na dany obiekt, w tym przypadku na twarz. Program sam pobiera zadaną liczbę zdjęć i przygotowuje je do analizy. Wskazane jest, aby obiekt znajdował się w ruchu i zmieniał swoje charakterystyczne cechy. W przypadku twarzy mogą to być miny, pochylenie itd.

Program rozpoczyna działanie od wyświetlenia prostego interfejsu tekstowego, w którym użytkownik definiuje parametry wzorców, takie jak liczba osób i ich nazwy, liczba szablonów dla każdej osoby oraz tryb pobierania próbek. Po wprowadzeniu ustawień, następuje proces pobierania próbek. Program został napisany w ten sposób, że automatycznie wybiera szablony twarzy z obrazów znajdujących się w polu widzenia kamery. Dzięki temu nie jest konieczne żmudne robienie dużej liczby zdjęć i wyodrębnianie z nich twarzy. Najważniejszą zaletą tej metody jest pobieranie próbek przez ten sam algorytm, który wydziela twarze w algorytmie działającym w czasie rzeczywistym, co znacznie zwiększa szansę znalezienia właściwych zależności w stosunku do ręcznego wydzielania twarzy. Każde pobrane zdjęcie z kamery zostaje poddane analizie przez klasyfikator Haara, który odnajduje na obrazie twarz. Dzięki współrzędnym określającym położenie oraz szerokość i wysokość prostokąta obrysowującego twarz, w prosty sposób wycinany jest interesujący fragment zdjęcia.

Sieć neuronowa przyjmuje jako wejścia wektory o 1600 elementach, reprezentujące zdjęcia o rozdzielczości 40×40 . Z tego względu, każda wykryta próbka jest przekalowywana do tego rozmiaru. Przygotowany zbiór szablonów jest przetwarzany do postaci wektorów i zapisywany w pliku tekstowym, zgodnie ze specyfikacją biblioteki FANN. Na podstawie tego pliku następuje uczenie sieci neuronowej. W analizowanym przypadku założono, że



Rys. 9. Zestaw próbek uczących sieć neuronową
Fig. 9. Set of patterns for neural network learning

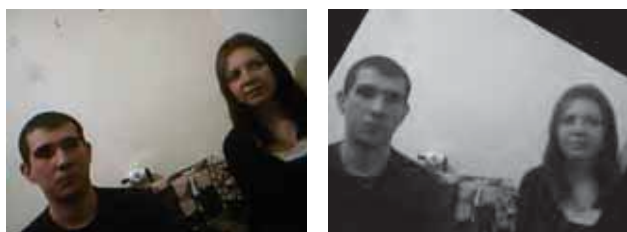
sieć ma uzyskać maksymalny błąd mniejszy od 10^{-4} , a czas uczenia nie może przekroczyć 1000 epok. Wybrany rodzaj sieci nie jest optymalnym wynikiem dogłębnej analizy różnych struktur, a raczej jednym z możliwych przykładów klasyfikatora dobrze spełniającego swoją rolę (struktura sieci dobrana eksperymentalnie).

Po zakończeniu fazy uczenia sieci neuronowej, możliwe jest rozpoczęcie działania programu w czasie rzeczywistym. Algorytm programu śledzącego twarze rozpoczyna się od fazy inicjalizacji. Nawiązywane i otwierane są połączenia z kamerą i mikroprocesorem oraz inicjalizowane wszystkie potrzebne zmienne. Prosty interfejs pobiera informacje od użytkownika o tym, która ze zdefiniowanych osób ma być śledzona.

Pętla główna programu rozpoczyna się od wczytania zdjęcia z kamery. Ustalono, że kamera pobiera zdjęcia o rozdzielczości 320×240 . Stanowi to kompromis między jakością obrazu a szybkością działania, co jest bardzo istotne w systemie rozpoznawania twarzy działającym w czasie rzeczywistym.

Po pobraniu klatki z kamery, następuje komunikacja z mikroprocesorem. Analiza nie może odbywać się, jeśli obraz został pobrany w trakcie ruchu kamery, dlatego po każdym wysłaniu polecenia sprawdzana jest odpowiedź mikroprocesora, potwierdzająca wykonanie ruchu. Jeśli możliwe jest dalsze przetwarzanie, rozpoczyna się proces przygotowujący zdjęcie do analizy przez klasyfikator Haara: przejście do skali szarości, skalowanie, filtrowanie obrazu oraz jego obrót (rys. 10).

Po wykonaniu wszystkich wstępnych czynności, algorytm wchodzi w decydującą fazę detekcji. Po wyrównaniu



Rys. 10. Wynik wstępnego przetwarzania obrazu

Fig. 10. Result of the image pre-processing

histogramu i odpowiednim obrocie, obraz podawany jest na klasyfikator Haara. Algorytm ten analizuje kolejno małe wycinki obrazu poszukując w nich cech charakteryzujących



Rys. 11. Wynik działania klasyfikatora Haara

Fig. 11. Result of the Haar Classifier operating



Rys. 12. Wynik rozpoznawania twarzy

Fig. 12. Result of the face recognition

Rys. 13. Obraz po wykonaniu ruchu

Fig. 13. Image after the movement



poszukiwany obiekt. Po przeszukaniu obrazu, cykl rozpoczyna się na nowo, z inną wielkością badanych klatek, aż do osiągnięcia rozmiaru obrazu. Podstawowymi parametrami klasyfikatora są: minimalna klatka – opisująca, od jakiej wielkości wycinków ma rozpocząć się poszukiwanie oraz krok – współczynnik skali dla kolejnych wielkości wycinków. Podczas pracy w czasie rzeczywistym konieczne jest ustalenie parametrów, które nie będą znacznie obciążające obliczeniowo. Z tego względu wybrano krok równy 1,2, minimalną klatkę o wielkości 18×18 , konieczność wykrycia twarzy przynajmniej w dwóch kwadratach oraz pomijanie obszarów o małym prawdopodobieństwie wykrycia twarzy.

Wyodrębnione twarze zostają zapisane w postaci wektorów i są kolejno podawane na wejścia nauczonego wcześniej klasyfikatora neuronowego. Wyjścia sieci mogą przyjmować wartości rzeczywiste w zakresie od -1 do 1 . Program analizuje odpowiedzi sieci i wybiera wyjście, które przyjęło najwyższą wartość. Jeśli wartość ta jest większa od $0,9$, analizowany obiekt zostaje uznany za jedną z twarzy z bazy danych. Dodatkowe reguły uniemożliwiają rozpoznanie jednej osoby w dwóch miejscach obrazu.

Po ustaleniu współrzędnych śledzonej twarzy, algorytm przesyła informację do mikroprocesora o położeniu obiektu. Ten wykonuje odpowiedni ruch silnikami zgodnie z przyjętą matrycą sterowań.

Na rys. 10–13 przedstawiono kolejne kroki algorytmu wykrywania, rozpoznania i śledzenia twarzy. W demonstrowanym przykładzie kamera miała za zadanie odnaleźć, a następnie śledzić osobę oznaczoną na rys. 8 jako Aga.

4. Wyniki doświadczeń

Celem przeprowadzonych badań była ocena skuteczności detekcji, rozpoznawania i śledzenia zadanych obiektów. Badania przeprowadzono w ściśle określonych warunkach: przy świetle dziennym bądź w dobrze oświetlonym pomieszczeniu, zasięg rozpoznawania nie przekraczał 3 m. Ograniczenia te były spowodowane niską jakością i szybkością działania kamery. W przypadku śledzenia twarzy, konieczne było ustawienie frontalne człowieka względem kamery.

Detekcja zarówno ruchu, jak i twarzy, cechowała się wysoką skutecznością działania, bliską 100 % w opisanych warunkach. Najślabszym ogniwem całego algorytmu okazało się rozpoznawanie twarzy. Należy jednak podkreślić, że główny nacisk w projekcie został położony na budowę platformy sprzętowej i opracowanie metod detekcji i śledzenia obiektów. Zmienne warunki oświetlenia, przyjmowanie charakterystycznych pozycji i min, które nie były definiowane w procesie uczenia, czy bardzo podobne twarze mogą powodować błędy identyfikacji. Mimo to, rozpoznawanie działa z zadowalającą skutecznością, przekraczającą 70 %. Koszt części użytych do budowy platformy można oszacować na 150 zł. Wszelkie zastosowane algorytmy zaprogramowano w środowisku open source.

5. Podsumowanie

Skuteczność działania systemu rozpoznawania i śledzenia obiektów dynamicznych można ocenić jako zadowalającą. Projekt spełnia wszystkie założenia, umożliwiając poprawne śledzenie ruchu oraz twarzy w czasie rzeczywistym. Należy pamiętać, że jest to prototyp urządzenia, a opracowane algorytmy nie są optymalnymi, a jedynie przykładowymi rozwiązaniami problemu.

Najślabszym punktem sprzętowym urządzenia jest prosta kamera internetowa. Uniemożliwia ona pobieranie zdjęć wysokiej jakości w odpowiednim czasie. Nie radzi sobie także ze słabym oświetleniem. Mimo to, w zupełności wystarcza do zaprezentowania możliwości tego typu systemu. Algorytm detekcji oraz śledzenia także spełnia swoje zadanie w określonych warunkach. Wykrywanie ruchu jest bardzo skuteczne, ale nie nadaje się do rozpoznawania, natomiast przy wykrywaniu twarzy, konieczne jest frontalne ustawienie człowieka. Drugą wadę można wyeliminować przy odpowiednim nakładzie czasu, tworząc znacznie „bogatsze” szablony dla klasyfikatora Haara.

Perspektywy rozwoju platformy są duże. Lepsze czasy przetwarzania i większą dokładność można uzyskać przez zastosowanie profesjonalnej kamery, precyzyjnych urządzeń wykonawczych oraz większej mocy obliczeniowej. Konieczne jest również znalezienie metody detekcji nie tylko dla twarzy frontalnie ustawionej do kamery. Możliwe jest skorzystanie z dwóch kamer, co z jednej strony ułatwi proces rozpoznawania obiektu, a z drugiej znacznie skomplikuje algorytm. Programy można połączyć z przemysłowymi bazami danych oraz rozwinąć o dokładniejsze metody identyfikacji, wykorzystując np. algorytmy bazujące na PCA (np. *eigen-faces*) do przetwarzania wstępnego obrazów. Kolejnym krokiem jest implementacja całego algorytmu na platformie sprzętowej, bez konieczności podłączania zewnętrznego komputera. Prace nad implementacją tych usprawnień są obecnie prowadzone przez autorów artykułu.

Bibliografia

- Sommerlade E., *Active Visual Scene Exploration*. DPhil Thesis, University of Oxford, 2011.
- Christensen H., *Active Vision*. Kungl Tekniska Hogskolan [www.cas.kth.se/~hic/active-vision.pdf].
- [www.indect-project.eu/indect-partners-pl].
- Viola P., Jones M., *Robust real-time object detection*, "International Journal of Computer Vision", 57(2), 2004, 13–154.
- Bark M., *Wykrywanie twarzy z pomocą OpenC*, [http://napiszpr.webd.pl], data odczytu: 15.10.2011
- Bradski G., Kaehler A., *Learning OpenC*, YO'Reilly, 2008.
- Lienhart R., Maydt J., *An extended set of Haar-like features for rapid object detection*, Proc. IEEE Conference on Image Processing, 2002.
- Nissen S., *Neural Networks Made Simple*, Software 2.0, 2005.
- Wróbel Z., Koprowski R., *Praktyka przetwarzania obrazów w programie MATLAB*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2004. ■

Tracking of the moving objects with the use of the computational intelligence techniques – hardware implementation

Abstract: The paper presents the hardware platform for detection, recognition and tracking moving objects with the use of computational intelligence methods. Project includes construction of hardware platform equipped with webcam and motors allowing camera to move. Image analysis is carried out within PC computer that is communicated with the platform by USB serial port. Human faces have served as the examples of objects that were detected, recognized and then tracked. All the computational algorithms were written with C++. The platform skillfully integrates open source computational environments, allowing the testing of designed algorithms in practice.

Keywords: image processing, face recognition, Haar classifier, artificial neural network.

inż. Przemysław Błaszowski

W 2012 r. uzyskał tytuł inżyniera na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej. Główne dziedziny zainteresowania to języki średniego poziomu, zastosowania nowoczesnej elektroniki w automatyce i metody sztucznych inteligencji.

e-mail: blaszkowski.przemyslaw@gmail.com



dr inż. Michał Grochowski

Adiunkt w Katedrze Inżynierii Systemów Sterowania na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej. Jego zainteresowania naukowe obejmują metody detekcji i lokalizacji uszkodzeń, diagnostyki procesów, sterowania optymalizującego oraz inteligencji obliczeniowej.

e-mail: m.grochowski@eia.pg.gda.pl

