

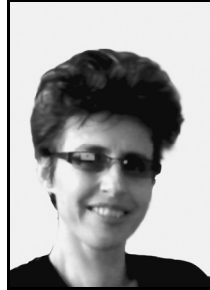
Teodora DIMITROVA-GREKOW, Mateusz ROGOWSKI

POLITECHNIKA BIAŁOSTOCKA,
ul. Wiejska 45A, 15-351 Białystok

Implementacja algorytmu śledzenia przy użyciu kamery oraz mobilnego robota edukacyjnego

Dr inż. Teodora DIMITROVA - GREKOW

Ukończyła studia na Wydziale Elektroniki Politechniki Sofijskiej, Bułgaria w 1991r., obroniła pracę doktorską na Uniwersytecie Technicznym w Wiedniu w 1997r. Jest adiunktem na Wydziale Informatyki przy Politechnice Białostockiej. Jej zainteresowania naukowe to synteza układów programowalnych, robotyka, mechatronika, analiza i przetwarzanie sygnałów.



e-mail: t.grekow@pb.edu.pl

Inż. Mateusz ROGOWSKI

Mateusz Rogowski ukończył studia na Wydziale Informatyki, Politechniki Białostockiej w 2013 z tytułem inżyniera na kierunku Informatyka, Politechnika Białostocka, gdzie aktualnie studiuje na kierunku Matematyka. Od 2012 pracuje jako programista w technologii Java. Zainteresowania naukowe to tworzenie aplikacji na urządzenia mobilne oraz analiza i przetwarzanie obrazów.



e-mail: m.rogowski@gmail.com

Streszczenie

Artykuł prezentuje algorytm śledzenia z wykorzystaniem filtra Kalmana. Do realizacji celu został wykorzystany smartfon z systemem Android i wbudowaną kamerą oraz robot z serii LEGO Mindstorms NXT. W ramach pracy został zaprojektowany i stworzony działający system śledzący obiekt. Zastosowanie filtra Kalmana w dużym stopniu poprawiło skuteczność śledzenia. Przedstawione zostały testy wraz z interpretacją ich wyników. Pokazują one skuteczność zaimplementowanych rozwiązań. Opisywany został także wpływ różnych parametrów na efektywność śledzenia.

Słowa kluczowe: detekcja i śledzenie obiektów, roboty edukacyjne, filtr Kalmana.

Implementation of tracking algorithm using cameras and mobile educational robot

Abstract

Information obtained through analysis of video sequences can have many uses, for instance in different video-game controllers or intelligent mobile safety systems [1]. Thanks to the miniaturization of integrated circuits, it is possible to use these solutions in intelligent mobile vehicles [2]. In order to keep track of an object, first there has to be found its existence and location. Many methods of detection and tracking are based on the specific object characteristic [3,4]. The observation of the object contour is also very popular [5,6]. In every case, however, the disadvantages are downside costs and possibility of losing an object by sudden changes of the selected feature. This work presents the implementation of a tracking algorithm with double detection-color and shape. Both parameters were analysed with a large tolerance, which simplified the calculation engine. There was used the Kalman filter [7]. An Android Smartphone with a built-in camera and a robot of LEGO Mindstorms NXT series were applied. The paper contains a description of the structure of the system (Fig.1) and the most important implementation details associated with a specific way of detecting and tracking the object (Fig.2). The experiments carried out show the efficiency of the system (Fig. 2,). The impact of the various parameters on the efficiency of the trace is described (Tab 1). The tests together with the interpretation of their results are presented.

Keywords: tracking algorithm, Kalman filter, object detection, educational robot.

1. Wprowadzenie

Jednym z podstawowych zadań robota mobilnego jest interakcja z otoczeniem. Aby mógł poprawnie działać powinien potrafić rejestrować środowisko w którym się znajduje i zmiany jakie w nim zachodzą. W ostatnich latach coraz częściej stosuje się rozwiązania wykorzystujące obraz z kamery. Przyczyniła się do tego dynamicznie zwiększająca się moc obliczeniowa komputerów, jak i jakość systemów wizyjnych. Informacje otrzymywane poprzez analizę sekwencji video mogą mieć wiele zastosowań. Bazując na nich różne kontrolery do gier komputerowych czy też systemy bezpieczeństwa [1].

Dzięki miniaturyzacji układów scalonych, możliwe jest stosowanie tych rozwiązań także w inteligentnych pojazdach mobilnych [2].

W celu śledzenia obiektu należy najpierw stwierdzić jego istnienie oraz położenie. Wiele metod detekcji i śledzenia jest opartych na konkretnej właściwości obiektu [3,4]. Bardzo popularna jest również obserwacja konturu [5,6]. Wszędzie jednak minusem zostają koszty obliczeniowe oraz możliwość utraty obiektu przy nagłej zmianie wybranej właściwości.

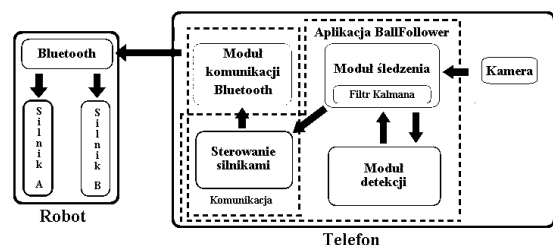
Niniejsza praca przedstawia implementację algorytmu śledzenia z podwójną detekcją – koloru oraz kształtu. Oba parametry rozpatrywano z dużą tolerancją co uprościło aparat obliczeniowy. Wykorzystano filtr Kalmana [7]. Do realizacji celu został wykorzystany smartfon z systemem Android i wbudowaną kamerą oraz robot z serii LEGO Mindstorms NXT.

W artykule zostały zawarte opis struktury systemu wraz z jego schematem oraz najbardziej istotne szczegóły implementacji związane z konkretnym sposobem realizacji detekcji i śledzenia obiektu.

Przeprowadzone eksperymenty pokazują dużą skuteczność zrealizowanego systemu. Opisywany został także wpływ różnych parametrów na efektywność śledzenia. Testy wraz z interpretacją ich wyników są pokazane pod koniec pracy. Przedstawione zostały możliwości rozwoju systemu.

2. Struktura systemu śledzenia

Sprzętowa warstwa systemu integruje robot i smartfon. Robot reprezentowany jest przez silniki i interfejs komunikacyjny. Smartfon realizuje rejestrację otoczenia, detekcję i śledzenie obiektu (aplikacja BallFollower) oraz komunikuje się z robotem w celu sterowania silników (rys. 1).



Rys. 1 Schemat blokowy systemu
Fig. 1. The system structure

Budowa robota ma duży wpływ na skuteczność działania systemu. Od niej zależy czy robot będzie odpowiednio skręcał oraz czy będzie możliwa realizacja śledzenia. Głównymi założeniami w trakcie projektowania była prostota, stabilność oraz łatwe mocowanie telefonu.

Skonstruowany robot jest pojazdem dwukołowym z niezależnymi napędami. Możliwe jest skręcanie w miejscu, czyli z zachowaniem osi skrętu między kołami, poprzez poruszanie kół w przeciwnie strony z tą samą prędkością.

Koła znajdują się blisko telefonu w celu zmniejszenia odległości urządzenia od osi obrotu. Dzięki temu, podczas skrętu obraz rejestrowany kamerą przesuwa się wolniej, niż gdyby telefon był bardziej wysunięty. Osiągnięto w ten sposób większą dokładność. Dodatkowo poprzez zbliżenie telefonu do kół zmniejszono drgania wywołane poruszaniem się robota. Kostka NXT jest zamontowana poziomo i dość nisko co stwarza przeciwwagę dla smartfona. Telefon ma stosunkowo sporą masę i gdyby był bardziej wysunięty lub przeciwwaga byłaby mniejsza, robot byłby niestabilny.

Śledzony obiekt to czerwona piłka z zestawu LEGO Mindstorms. Zapewnia ona łatwy do określenia kształt oraz prosty do rozpoznania kolor.

Ważnym elementem w inicjalizacji kamery jest ustawienie jej rozdzielczości oraz format danych. Do realizacji śledzenia została wybrana rozdzielczość 800x480. Do tej decyzji przyczyniły się trzy rzeczy: wymagana jakość obrazu, rozdzielczość ekranu oraz moc obliczeniowa procesora. Należało wybrać kompromis między jakością a ilością danych do analizy. Podczas testów stwierdzono, że ta rozdzielczość jest najbardziej odpowiednia.

Po zainicjalizowaniu podgląd kamery działa w tle. Aby pobrać obecną klatkę, należy przekazać do obiektu kamery bufor, który będzie przechowywał dane. Przy następnym odświeżeniu podglądu, zostaną sprawdzone dostępne bufory i jeżeli jakiś istnieje, zostaną do niego załadowane dane oraz wywołana jest metoda obsługująca zwrócenie buforu z danymi. Bufor ma postać tablicy.

Format obrazu zwracanego z kamery jest uzależniony od wykorzystywanego urządzenia. Wszystkie telefony z Androidem obsługują format YUV, niektóre natomiast udostępniają inne formaty. Takie jak RGB. Wykorzystywane urządzenie wykorzystuje model YUV z formatem zapisu NV21.

Do realizacji komunikacji robota LEGO NXT Mindstorms z smartfonem została wykorzystana aplikacja MINDdroid. Kod źródłowy programu jest udostępniany na licencji GPL v3. Komunikacja z robotem realizowana jest przez łączność Bluetooth z wykorzystaniem LCP (ang. *LEGO Communication Protocol*).

3. Implementacja detekcji i śledzenia

Do wykonania detekcji i śledzenia została wykorzystana wbudowana w telefonie kamera. Wszystkie operacje i obliczenia wykonują się w telefonie, do robota wysyłane są tylko instrukcje sterujące silnikami.

Implementacja detekcji

Śledzony obiekt określono dwoma cechami charakterystycznymi: kolor i kształt.

W celu dobrania wartości koloru, kilka klatek zarejestrowanych przez aplikację zostało zapisanych i przeanalizowanych na komputerze. Do interpretacji koloru wybrano model RGB, jako że jest bardziej popularny. Zakres wartości zwracane przez czujnik koloru jest:

$$Z \in [0; 255] \quad (1)$$

Na podstawie analizy danych z kamery powstał uproszczony zakres koloru poszukiwanej piłki $Z_p > 100$. Ograniczenie wartości z jednej tylko strony przyspieszyło sprawdzanie koloru, przy czym okazało się wystarczająco skuteczne.

W związku z tym, że analizowany jest kolor RGB, detekcja jest uzależniona od oświetlenia.

Każdy kształt jest sprawdzany pod względem proporcji Δ :

$$\Delta = H_p/W_p \quad (2)$$

gdzie H_p – wysokość, a W_p – szerokość obiektu.

Empirycznie ustalono zakres proporcji:

$$\Delta \in (0.8; 1.2) \quad (3)$$

Dodatkowo, żeby wyeliminować pojedyncze piksele, oraz zbyt małe obiekty, określona została minimalna wielkość piłki: po-

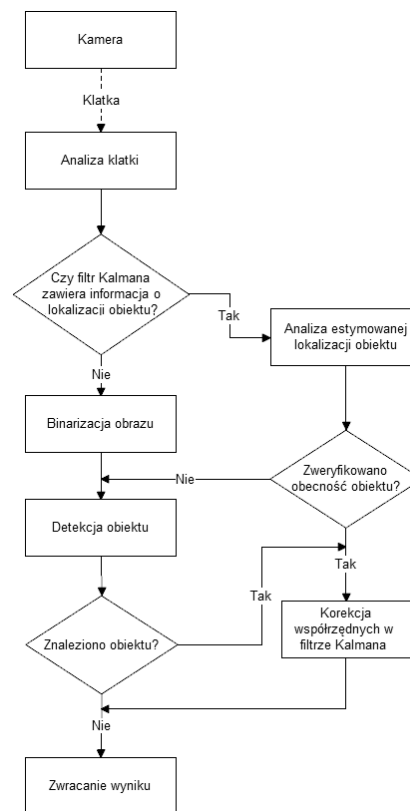
wierzchnia znalezionego obiektu musi stanowić przynajmniej 5% rejestrowanego obrazu.

Głównym punktem *detekcji na całym obrazie*, bez estymowanej wcześniej pozycji obiektu, jest przekazanie zbinaryzowanej tablicy pikseli do algorytmu. Tablica ta posiada informacje o połowie punktów zarejestrowanych przez kamerę. Na tym etapie rozdzielczość obrazu zostaje zmniejszona do 400x240 w celu przyspieszenia działania algorytmu. Dokonano tego poprzez pomijanie co drugiego piksela. Jest to sposób szybszy od uśredniania i jak wykazały testy, zupełnie wystarczający.

Binaryzacja polega na sprawdzeniu wszystkich pikseli w obrazie pod kątem poszukiwanego zakresu kolorów. Jeżeli kolor się zgadza, pikselowi zostaje przypisana wartość 1, w przeciwnym wypadku 0. Następnie zbinaryzowana tablica jest przeglądana w poszukiwaniu pozytywnej wartości. Oznacza ona istnienie poszukiwanego koloru, a więc możliwą lokalację obiektu. Współrzędne tego punktu są przekazywane do metody analizującej kształt zbioru punktów. Przy pomocy przeszukiwania BFS (ang. *Breadth-First Search*) sprawdzane są wszystkie sąsiadujące piksele. Dzięki temu otrzymujemy wymiary grupy punktów pasujących do koloru piłki. Wymiary te są weryfikowane pod kątem kształtu śledzonego obiektu. W związku z ograniczonym rozmiarem stosu na telefonie, algorytm musiał zostać zaimplementowany iteracyjnie.

Implementacja śledzenia

Obraz z kamery jest zwracany do aplikacji w postaci tablicy. Jeżeli w poprzednim przejściu obiekt został wykryty, zostaje pobrana przewidywana pozycja za pomocą filtru Kalmana (Rys. 2). W takim przypadku estymowane miejsce zostaje sprawdzone w uproszczony sposób, aby znacznie przyspieszyć proces.



Rys. 2 Algorytm śledzenia obiektu
Fig. 2 The tracking algorithm

Jeżeli obiekt nie zostanie znaleziony lub nie zostały pobrane dane z algorytmu śledzenia, zostaje wywołany algorytm detekcji na całym obrazie. W obu przypadkach wynik wyszukiwania zostaje przekazany do filtru Kalmana. Dalej informacja o pozycji celu zostaje przekazana do obiektu sterującego robotem. Podejmuje on decyzję o tym czy i z jaką prędkością mają się obracać silniki. Dane te zostają przesłane do robota za pomocą modułu komunikacyjnego.

Filtr Kalmana wykorzystany został do estymacji współrzędnych X oraz Y poszukiwanego obiektu. Pełna analiza obrazu z kamery jest wykonywana tylko wtedy, gdy nie powiodła się weryfikacja położenia, lub nie ma informacji o wcześniejszej lokacji piłki.

Pierwszym etapem jest sprawdzenie, czy filtr Kalmana został wcześniej zainicjalizowany. Oznaczało by to, że obiekt został już wcześniej zlokalizowany. W takim wypadku wywoływana jest metoda estymująca aktualne położenie piłki.

Do weryfikacji otrzymanych współrzędnych napisany został algorytm sprawdzający wybrane piksele w sąsiedztwie przewidzianej lokacji. Analizuje on dane prosto z kamery, czyli w formacie N21, przez co pomijana jest binaryzacja. Wszystkie potrzebne piksele są pobierane i konwertowane do systemu RGB na bieżąco. W ten sposób uzyskano znaczne przyspieszenie w porównaniu z algorytmem w module detekcji. Algorytm przechodzi do kolejnego etapu tylko, gdy weryfikacja obecności obiektu się powiedzie. W przeciwnym wypadku zostaje wywołany moduł detekcji, aby na nowo znaleźć piłkę. Jeżeli obiekt nie zostanie wykryty, następuje wyłączenie filtra Kalmana.

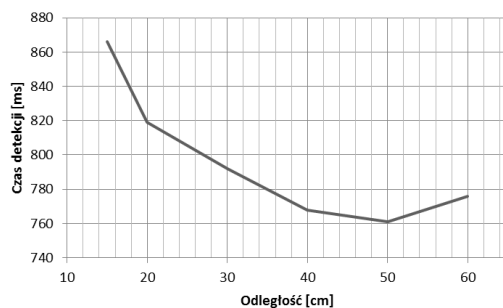
Po określeniu współrzędnych szukanego obiektu przez moduł detekcji lub ogólne sprawdzenie w module śledzenia, następuje korekcja parametrów filtra Kalmana. Gdy obiekt filtra nie istnieje, przed korekcją wywoływana jest jego inicjalizacja.

4. Testy

Eksperymenty przeprowadzono w trzech kategoriach: detekcja obiektu, detekcja za pomocą filtra Kalmana i śledzenie obiektu.

Detekcja obiektu

Przeprowadzonych zostało szereg testów sprawdzających działania detekcji obiektu. Dalej krótko zostanie opisanych kilka z nich.



Rys. 3 Zależność czasu wykrywania obiektu od odległości
Fig. 3. Dependency of the detection time on the distance to the object

Rys. 3 pokazuje zarejestrowaną zależność czasu wykrywania obiektu od odległości na jakiej znajduje się on od kamery. Zmierzono dziesięciokrotnie czas lokalizowania piłki znajdującej się w centralnym punkcie obrazu dla kilku różnych odległości.

Detekcja jest szybsza przy większym oddaleniu kamery od obiektu. Przyczynia się do tego mniejszy rozmiar piłki na obrazie, w związku z czym badanie jej kształtu trwa mniej czasu.

Przebadana została również sytuacja, w której w polu widzenia wprowadzony został przedmiot o podobnym kolorze. Spowodowało to średnio siedmiokrotne opóźnienie w detekcji.

Eksperymenty pokazały, że obiekty o odmiennym kolorze nie wpływają na pracę systemu. Jeżeli natomiast kolor jest ten sam – rozpoznawalność spada kilkukrotnie pod warunkiem, że forma drugiego obiektu jest niesymetryczna. W przypadku, gdy obiekty posiadają formę podobną detekcja jest niemożliwa.

Detekcja z wykorzystaniem filtra Kalmana

Doświadczenie zweryfikowało jak algorytm się sprawdza przy różnych prędkościach przemieszczania się piłki.

W tym teście piłka była przesuwana ze stałą prędkością, zwiększoną odpowiednio do każdej próby. Wyniki eksperymentu przedstawione zostały w tabeli 1. Pokazują one, że filtr Kalmana przyspiesza detekcję obiektu kilkaset razy. Kolejne kolumny odpowia-

dają kolejnym rozpoznaniom dla ciągłej analizy. W momentach, gdy estymacja jest niedokładna, wykonywane jest przeszukiwanie całego obrazu i następuje korekcja parametrów filtra.

Tab. 1. Czasy detekcji [ms] z wykorzystaniem Filtru Kalmana
Tab. 1. The time of detection using the Kalman filter

V [cm/s]	1	2	3	4	5	6	7	8	9	10	11	12
2	801	792	915	1	791	798	2	4	792	2	1	2
4	790	793	798	804	2	810	4	6	804	2	777	2
8	788	796	802	814	780	6	804	785	6	4	2	2

Śledzenie obiektu przez robota

Ostatnie testy dotyczą sprawdzenia działania całego systemu. Eksperyment polegał na przesuwanie piłki przed kamerą z różną prędkością i analizie poprawności reakcji robota. Na podstawie poprzednich testów postanowiono ustawić odległość układu od poszukiwanego obiektu na 50 cm.

Przy prostym ruchu piłki, bez nagłych zmian prędkości, robot z powodzeniem ją rozpoznawał i śledził. Maksymalna szybkość obiektu, przy której układ poprawnie się zachowywał to w przybliżeniu 4 cm/s.

5. Wnioski

W ramach pracy został zaprojektowany i stworzony działający system śledzący obiekt. Skuteczność działania układu zależy w znacznym stopniu od warunków otoczenia. Podczas testów został sprawdzony wpływ niektórych z nich. Aplikacja kontrolująca działanie całego układu została napisana na system Android działający na telefonie.

Okazało się że tło rejestrowanego obrazu ma duże znaczenie przy detekcji. Jeżeli w obiektywie kamery znajdują się przedmioty o zbliżonym kolorze do poszukiwanego obiektu, algorytm działa znacznie wolniej. Jest to związane z sprawdzaniem każdego obszaru w celu ustalenia jego wymiarów. Można to poprawić, stosując dodatkowe filtry lub implementując inny algorytm detekcji.

Zastosowanie filtra Kalmana poprawiło skuteczność śledzenia. Nagłe zmiany prędkości oraz kierunku ruchu obiektu jednak powodują, że estymacje położenia dokonywane przez filtr są błędne.

Stworzone rozwiązanie działa poprawnie w określonych warunkach. W przyszłości można by było zastosować dodatkowe filtry podczas detekcji lub wykorzystać szybszy układ obliczeniowy. Warto by również przetestować kilkustopniową detekcję obiektu w celu usprawnienia komunikacji z robotem.

6. Literatura

- [1] Ramirez R. A., Rojas Castillo J., Chae, O.: Local directional number pattern for face analysis: face and expression recognition, Image Processing, IEEE Trans, Vol. 99, pp. 1-13, 2012.
- [2] Gao B., Han L.S.: Neural networks implementation of the visual information processing for an intelligent aerial vehicle, Cognitive Informatics, IEEE Trans, Vol. 5, pp. 394 – 399, 2006.
- [3] Cheng C. C., Kun-Hsin H., Hui-Ting L., Gwo-Long L.: Image following using the feature-based optical flow approach, Proceedings of the 2002 IEEE International Symposium, pp. 350–355, 2002.
- [4] Viola P., Jones M.J.: Rapid object detection using a boosted cascade of simple features, IEEE CVPR, 2001.
- [5] Allili M.S., Ziou D.: Adaptive appearance model for object contour tracking in videos, Computer and Robot Vision, IEEE CRV, pp. 510-518, 2007.
- [6] Yilmaz A., Li X., Shah M.: Object contour tracking using level sets, Asian Conference on Computer Vision, Jaju Island, Korea, 2004.
- [7] Welch G., Bishop G., An introduction to the Kalman Filter, University of North Carolina at Chapel Hill, 2006.