

Artur Fogiel, Jacek Tkacz

UNIwersytet ZIELONOGÓRSKI, Wydział Elektrotechniki, Informatyki i Telekomunikacji, Instytut Informatyki i Elektroniki, ul. Podgórna 50, 65-246, Zielona Góra

Rozpoznawanie symboli języka migowego z wykorzystaniem algorytmów cyfrowego przetwarzania obrazów

Mgr inż. Artur FOGIEL

Absolwent Uniwersytetu Zielonogórskiego. Interesuje się zagadnieniami cyfrowego przetwarzania obrazu i algorytmami uczenia maszynowego. Pasjonat nowych technologii.

e-mail: fogiel.artur@gmail.com



Dr inż. Jacek TKACZ

Absolwent Uniwersytetu Zielonogórskiego. Od 2009 roku pracownik zatrudniony na stanowisku adiunkta w Zakładzie Inżynierii Komputerowej. Prowadzi badania nad symboliczną metodą dowodzenia twierdzeń i jej praktycznymi zastosowaniami w informatyce oraz elektronice. Interesuje się nowoczesnymi technologiami projektowania i wytwarzania aplikacji, w tym aplikacji mobilnych. W latach 1997-2005 zajmował się projektowaniem i rozwojem oprogramowania dla polskich bibliotek (PROLIB).

e-mail: J.Tkacz@iie.uz.zgora.pl



Streszczenie

W artykule został opisany proces projektowania i realizacji aplikacji, przeznaczonej do analizy w czasie rzeczywistym obrazów cyfrowych zawierających symbole języka migowego. Stworzone oprogramowanie zostało napisane w języku C++. Wykorzystuje ono bibliotekę OpenCV w połączeniu z sensorem Microsoft Kinect. Oprogramowanie przetwarza około 15 klatek obrazu w ciągu sekundy i rozpoznaje gesty z skutecznością 71%. Program jest także odporny w znacznym stopniu na wpływ warunków oświetleniowych.

Słowa kluczowe: Microsoft Kinect, OpenCV, OpenNI, język migowy.

Realtime sign language recognizing using digital video processing algorithms

Abstract

The paper describes implementation process of application that recognize sign language symbols which are presented by user in real time. First part of this paper presents software and libraries that are used by this program. It also contains description of Microsoft Kinect sensor with explanation of how it operates. Second part of this article is dedicated to KinectSL application. It presents structure of the application its algorithm and study that led to described method of recognizing sign language gestures. Presented application is multi-platform and can be run on Windows, Linux and MacOS operating systems. This software is able to recognize gestures for two hand at once that allows it to support advanced gestures. The tests carried out that KinectSL processes 15 frames per second and its performance of gestures recognizing is above 71 percent. Described system is good base to create program that will be able to fluent translate sign language. It can also be used at industry and entertainment to control processes with user gestures.

Keywords: Microsoft Kinect, OpenCV, OpenNI, sign language.

1. Wprowadzenie

Dialog w języku migowym polega całkowicie na wykorzystaniu narządu wzroku, rozmówcy obserwują ruch dłoni, ich położenie czy mimikę twarzy a następnie interpretują ich znaczenie. Zadaniem prezentowanej aplikacji jest odtworzenie tego procesu.

Problem rozpoznawania gestów języka migowego za pomocą komputera jest jednak wciąż nierozwiązany. Naukowcy, studenci i pasjonaci próbują stworzyć rozwiązanie uniwersalne, niezależne od warunków zewnętrznych, takich jak nasłonecznienie pomieszczenia, kolor ubrania czy tła osoby obserwowanej za pomocą kamery.

2. Cyfrowe przetwarzanie obrazów

Rosnąca jakość urządzeń wizyjnych, rozwój algorytmów oraz wzrastająca wydajność procesorów, sprawiły, że komputerowe przetwarzanie obrazów wykorzystywane jest coraz częściej w bardzo wielu dziedzinach nauki oraz przemysłu.

Prezentowany projekt korzysta z popularnych bibliotek i oprogramowania służącego do pracy z obrazami cyfrowymi, takich jak:

OpenCV (*Open Source Computer Vision Library*) jest zbiorem zawierającym ponad 500 funkcji przeznaczonych do przetwarzania obrazu w czasie rzeczywistym. Biblioteka jest darmowa i została udostępniona na licencji BSD (co pozwala na jej wykorzystanie również w celach komercyjnych). Szczegółowy opis możliwości biblioteki wraz z przykładami dostarczają pozycje [1,2]

OpenNI jest biblioteką dostarczającą zbiór interfejsów programistycznych udostępnianych na licencji open source.

OpenNI oferuje wsparcie dla:

- Rozpoznawania komend głosowych
- Obsługi gestów dłoni
- Śledzenia części ciała użytkownika

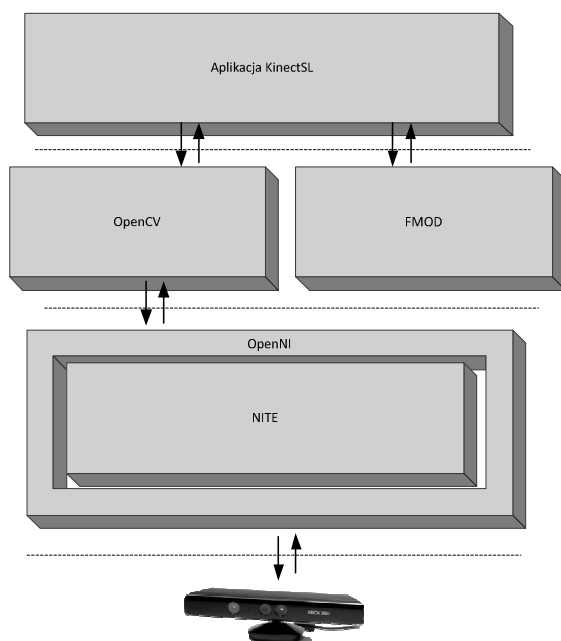
Realizacja tych zadań przebiega za pomocą middleware, np. NITE, z którym komunikuje się OpenNI, udostępniając programistom jednolity interfejs.

3. Aplikacja KinectSL

Zaprojektowana aplikacja umożliwia rozpoznawanie w czasie rzeczywistym, gestów języka migowego, prezentowanych przez osobę znajdującą się przed urządzeniem Microsoft Kinect. Program pozwala na pracę w trudnych warunkach oświetleniowych, np. wieczorem lub nocą. Prezentowany system obsługuje 22 gesty reprezentujące litery oraz 9 odpowiadających cyfr. Ważną cechą programu jest to, że pozwala on na wykrywanie i analizę dwóch dłoni jednocześnie, dzięki czemu obsługuje skomplikowane symbole a także nie wymaga od użytkownika trzymania dłoni w ściśle określonym obszarze, pozwala to na swobodne wykonywanie gestów i umożliwia aplikacji dostosowanie się do obsługującej ją osoby. Projekt jest wieloplatformowy i może zostać uruchomiony na wielu systemach operacyjnych, min. Windows, Linuks, MacOS.

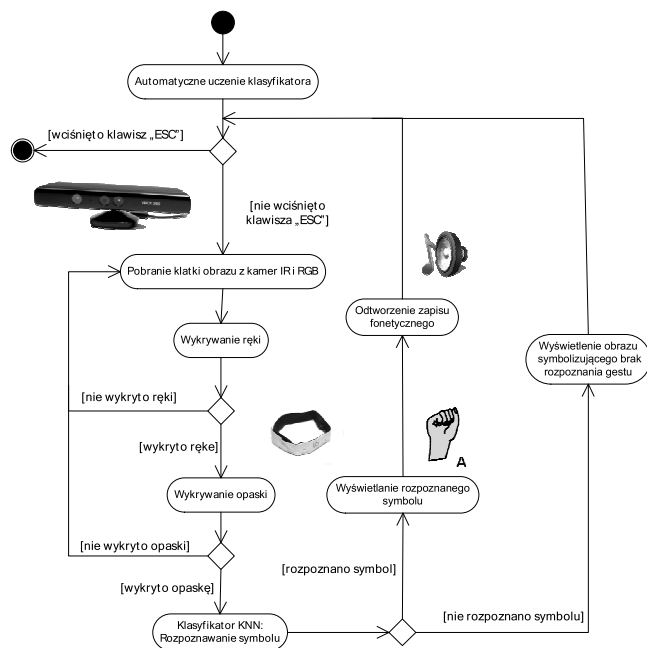
3.1. Struktura projektu

Na poniższym rysunku widoczna jest struktura aplikacji oraz sposób komunikacji poszczególnych elementów programu pomiędzy sobą. Można zauważyć, że biblioteka OpenCV współpracuje z OpenNI, która odpowiada min. za komunikację z urządzeniem Kinect. System wykorzystuje również bibliotekę FMOD do odtwarzania zapisu fonetycznego symboli języka migowego.



Rys. 1. Struktura aplikacji
Fig. 1. Application structure

3.2. Algorytm działania



Rys. 2. Diagram czynności aplikacji KinectSL
Fig. 2. Activity diagram for KinectSL application

W momencie uruchomienia, pierwszą czynnością wykonywaną przez program jest uczenie klasyfikatora KNN, na podstawie plików zbioru uczącego, znajdujących się w katalogu programu. Po zakończeniu tego procesu, uruchamiany jest sensor Kinect. Jeżeli wszystko zostało wykonane poprawnie, następuje odtworzenie komunikatu głosowego o uruchomieniu aplikacji.

Kolejnym krokiem jest pobranie obrazu z kamer RGB oraz IR. Następnie obraz jest przetwarzany i poszukiwana jest opaska LED umieszczona na nadgarstku użytkownika. W momencie znalezienia opaski następuje rozpoznawanie gestów użytkownika. Poprawnie rozpoznany gest jest wyświetlany w postaci graficznej oraz odtwarzany jest jego zapis fonetyczny.

4. Wydajność i skuteczność aplikacji

Głównym założeniem programu, było uzyskanie jak najwyższej skuteczności rozpoznawania gestów oraz skrócenie czasu przetwarzania obrazu do minimum.

Spełnienie warunku dotyczącego czasu jaki aplikacja potrzebuje na przetworzenie obrazu i analizę gestu udało się zweryfikować poprzez umieszczenie odpowiednich instrukcji w kodzie programu. Na podstawie danych zwróconych przez aplikację można zauważyć, że program, przetwarza około 15 klatek obrazu w ciągu sekundy. Do testu wykorzystano laptop Acer Aspire 6930g z procesorem Intel Core 2 Duo T5800 2 GHz oraz wyposażonym w 3 GB pamięci RAM.

Weryfikacja skuteczności rozpoznawania gestów, wymagała przeprowadzenia badania, które wykonano na grupie trzech osób. Zadaniem grupy było pokazywanie wybranych gestów, następnie dokonano zapisu ilości gestów rozpoznanych poprawnie, niepoprawnie lub nierozpoznanych w ogóle. Badanie zostało powtórzone pięciokrotnie, na tej samej grupie osób w celu uzyskania dokładnych danych o pracy programu.

Na podstawie przeprowadzonego badania można stwierdzić, że skuteczność rozpoznawania gestów przez zaprojektowany system przekracza 71 %. W trakcie wykonywania testu zauważono, że na końcowy wynik największy wpływ mają dwa czynniki. Pierwszym są zakłócenia sensora Kinect wynikające z metody jego działania. Drugim, niedokładne odwzorowanie gestów przez użytkownika.

5. Podsumowanie i wnioski

Celem artykułu było zaprezentowanie możliwości biblioteki OpenCV, sensora Kinect oraz algorytmów uczenia maszynowego. W pierwszej części dokumentu zostały opisane min. biblioteki służące do cyfrowego przetwarzania obrazu, budowa i zasada działania urządzenia Microsoft Kinect. W drugiej części na podstawie przedstawionych narzędzi i metod została zaprojektowana aplikacja pozwalająca na rozpoznawanie gestów w czasie rzeczywistym. System zapewnia akceptowalną szybkość przetwarzania obrazu oraz sprawność rozpoznawania gestów. Stanowi bardzo dobrą bazę do opracowania aplikacji będącej kompletnym tłumaczem języka migowego, obsługującym także gesty wykonywane w trzech płaszczyznach. Sensor Kinect dla konsoli Xbox 360 został zaprojektowany z myślą o rozrywce. Na rynku pojawiły się jednak nowe urządzenia, dedykowane obsłudze gestów, takie jak „Kinect for Windows” lub Asus Xtion Pro Live, których zastosowanie prawdopodobnie pozwoliłoby usprawnić pracę aplikacji.

6. Literatura

- [1] Gary Bradski, Adrian Kaehler, Learning OpenCV Computer Vision with the OpenCV Library, O'Reilly 2008
- [2] Robert Laganière, OpenCV 2 Computer Vision Application Programming Cookbook, Packt Publishing, 2011
- [3] Johannes Kilian, Simple Image Analysis By Moments, <http://public.cranfield.ac.uk/c5354/teaching/dip/openvc/SimpleImageAnalysisbyMoments.pdf> [Dostęp 24.03.2012]
- [4] V. A. Oliveira, A. Conci, Skin Detection using HSV color space, <http://www.matmidia.mat.puc-rio.br/sibgrapi2009/media/posters/59928.pdf> [Dostęp 24.03.2012]
- [5] Ryszard Tadeusiewicz, Przemysław Korohoda, Komputerowa analiza i przetwarzanie obrazów, Społeczeństwo Globalnej Informacji, Kraków 1997