AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

KATEDRA AUTOMATYKI I ROBOTYKI

Interfejsy człowiek komputer

Pomiar pulsu za pomocą kamery

Autorzy: *Głowacki Wojciech, Obalski Emil, Piekarz Arkadiusz*
Kierunek studiów: *Automatyka i robotyka*
Specjalizacja: *Inteligentne systemy sterowania*

Kraków, 2018

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Spis treści

1. Wprowadzenie.

Zmiana ukrwienia tkanki skóry ludzkiej podczas akcji serca może być wykrywana przez kamerę wizyjną. Dzięki temu, potencjalnie, możliwy jest zdalny pomiar pulsu jednej lub wielu osób jednocześnie. Efekt obserwowany jest w całym widzialnym spektrum jednak jest najmocniejszy w paśmie promieniowania odpowiadającego barwie zielonej. Podczas testowania algorytmów zdalnego pomiaru pulsu istotna jest rejestracja sygnału referencyjnego który może być gromadzony za pomocą czujnika dotykowego.

1.1. Cel i zakres projektu.

Celem projektu jest implementacja algorytmu detekcji pulsu w czasie rzeczywistym w oparciu o istniejące rozwiązania.

System powinien rejestrować puls estymowany na podstawie obrazów z kamery oraz jednocześnie gromadzić sygnał z dotykowego czujnika pulsu SEN0203 jako danych referencyjnych.

Wyniki pomiarów powinny być wyświetlane na ekranie komputera bądz zapisywane na dysku. Aplikacja powinna pozwalać na logowanie pomiarów pochodzących z algorytmu przetwarzania danych z kamery oraz odpowiadających im pomiarom z czujnika SEN0203 przysłanego przy użyciu kontolera Arduino Uno.

2. Algorytm wykrawania pulsu - opis teoretyczny

Wykrywanie ludzkiego tętna za pomocą kamery wymaga określenia rejonu ciałą osoby badanej. Dobrą strefą jest czoło człowieka. Głównie z powodu stosunkowo dużej przestrzeni skóry, która nie jest pokryta włosami, a ponadto jest mniej podatna na mimikę niż dla przykładu policzki.

Po ustaleniu i utrzymywaniu przez czas trwania pomiaru interesującego nas obszaru (ROI, czyli Region of Interest), należy następnie obliczyć średnią wartość składowych kolorów RBG w określonym obszarze.

Szczególnie analiza składowej zielonej okazuje się być wystarczająco wiarygodną do określenia pulsu (być może istnieją lepsze proporcje wyboru poszczególnych kanałów kolorów obrazu, lecz dla przykładu kolor niebieski ma skłonności do wprowadzania zaszumienia w analizie danych). Analiza i wydobycie znaczących danych jest możliwa, dzięki różnym charakterystyką absorpcji światła przez hemoglobinę (pozycja nr 5 w bibliografii). Następnie sygnał powinien zostać znormalizowany zgodnie ze wzorem:

$y'(t) = \frac{y(t)-\mu}{\sigma}$,

gdzie σ to odchylenie standardowe, natomiast μ jest wyliczoną średnią wartością.

Otrzymany sygnał powinien zostać przefiltorwany w wyszczególnienia jedynie tych częstotliwości które mogłyby odpowiadać za puls. Kolejnym krokiem wyliczenie transformaty widmowej przy użyciu transformacji Fouriera.

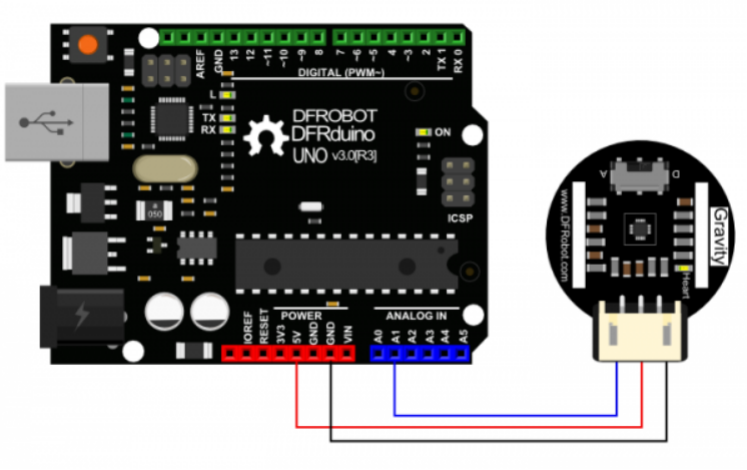
Finalnie, puls odpowiada częstotliwości odpowiadającej dla maksymalnej wartości energii w wyszczególnionym paśmie.

3. Realizacja

Do realizacji projektu wykorzystano Arduino UNO, czujnik SEN0203 oraz komputer klasy PC. Do Arduino został podłączony czujnik, który mierzył puls w trakcie działania aplikacji i poprzez port szeregowy wysyłał dane do komputera. Na komputerze została zaimplementowana aplikacja, która odbierała te dane i zapisywała do pliku dołączając do tego czas ich przybycia. Dodatkowo aplikacja komputerowa pobierała dane w postaci obrazu z kamerki internetowej. Na tych danych w czasie rzeczywistym działał algorytm, który obliczał puls. Otrzymane w ten sposób wartości zostały zapisywane do pliku. Na podstawie logowanych danych utworzono wykresy wizualizujące błędy pomiarów pulsu przy zastosowaniu algorytmu wizyjnego, traktując pomiar z czujnika jako wartość prawidłową i referencyjną.

3.1. Arduino i czujnik pulsu SEN0203.

W zastosowanym rozwiązaniu zdecydowano się na Arduino ze względu na łatwość i szybkość implementacji. Wykorzystano środowisko Arduino IDE w wersji 1.8.5 oraz gotową bibliotekę (DFRobot_Heartrate ()2)), którą zaimportowano do projektu. Biblioteka ta umożliwia komunikację z czujnikiem pulsu w dwóch trybach pracy: cyfrowym oraz analogowym. Podczas wstępnych testów projektu okazało się, że w trybie cyfrowym prawidłowe dane docierają do czujnika znacznie częściej i dlatego zdecydowano się na jego wybór. Na rysunku 3.1 przedstawiono schemat podłączenia czujnika do mikrokontrolera.



Rys. 3.1. Schemat podłączenia.

Arduino podłączono przez złącze USB do komputera, na którym uruchomiono aplikację logującą dane. Gdy tylko czujnik SEN0203 posiadał istotne dane automatycznie przesyłał je na port szeregowy.

3.2. Aplikacja desktopowa.

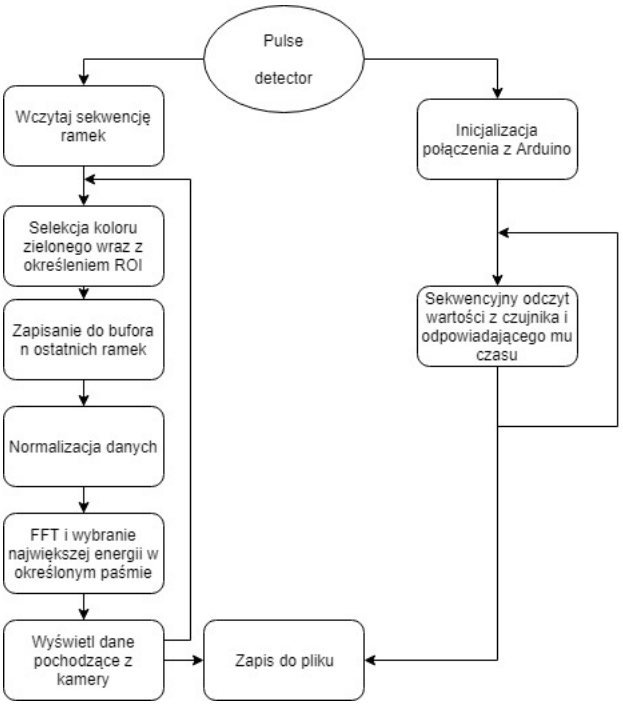
Przy tworzeniu aplikacji desktopowej wykorzystano następujące programy i biblioteki:

- Python 3.6,
- OpenCv 3.3.1,
- Python Numpy 1.14.3,
- Python Matplotlib 2.2.2,

– Python Pyserial 3.4.

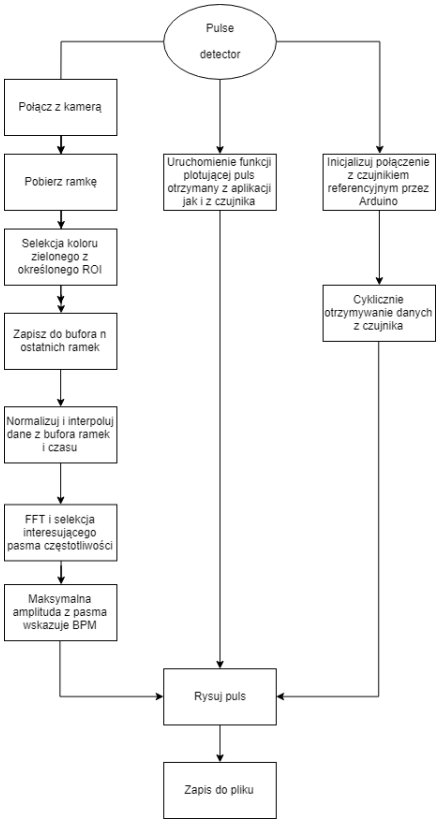
Aplikacja stworzona została zgodnie z algorytmem opisanym w rozdziale poświęconym teoretycznemu wyznaczaniu pulsu człowieka przy użyciu kamer, czy sekwencji wideo.

Stworzono dwie wersje aplikacji. Pierwsza z wersji odczytuje sekwencję ramek z określonego folderu oraz wylicza na ich podstawie wartości BPS. Ponadto łączy się przez port szeregowy z informacjami przychodzącymi z czujnika ciśnienia i zapisuje je wraz z odpowiadającym im czasom. Druga wersja jest bardziej rozbudowana. Inicjalizowane jest połączenie z kamerą ethernetową. Oprócz policzenia BPS dla nieregularnej częstotliwości FPS, jednocześnie wyrysowywane są aktualne wartości pulsu z czujnika jak i algorytmu. Nie daje ona aż tak dobrych efektów jak pierwsza z wersji, co z pewnością jest spowodowane dużym nakładem obliczeniowym co przekłada się między innymi na spadek FPS.



Rys. 3.2. Schemat działania pierwszej wersji programu.

Program rozpoczyna się od odczytania sekwencji ramek z pliku. Filmy zapisywane były w jakości FullHD z 30 FPS. Dodatkowo cyklicznie odczytywane są dane pochodzące z portu szeregowego, które wysyłane są przez czujnik SEN0203 Heart Rate Sensor podłączony do Arduino Uno.



Rys. 3.3. Schemat działania drugiej wersji programu.

Interesujący region na obrazie wyznaczony został na sztywno dla każdego z filmów. Wybrane ROI ma wielkość pokrywającą czoło osoby na obrazach. Otrzymana ramka wraz z odpowiadającym jej czasem zapisywana jest do bufora ramek. Długość bufora określona jest przy inicjalizacji pracy algorytmu. Przyjęto, że będzie to 100 ramek.

Najpierw jednak wybierany został kolor zielony, a średnia jego gęstości w określonym ROI zapamiętywana.

W wypadku czytania ramek z folderu nie ma problemu z zmienną liczbą FPS. W drugiej wersji, z powodu różnej częstotliwości ramek otrzymywanych z kamery internetowej (wideo narzuca ograniczenie nagłych spadków w liczbie FPS), należało interpolować brakujące ramki. Algorytm oczekiwał otrzymywania stałej liczby klatek na sekundę niezależnie od spadków FPS z kamery.

Kolejnym krokiem była normalizacja danych (kolor zielony) w ROI z każdej rameki.

Bufor obrazków przepuszczany był następnie przez szybką transformatę Fouriera. Wyszczególniana została z nich amplituda oraz częstotliwość.

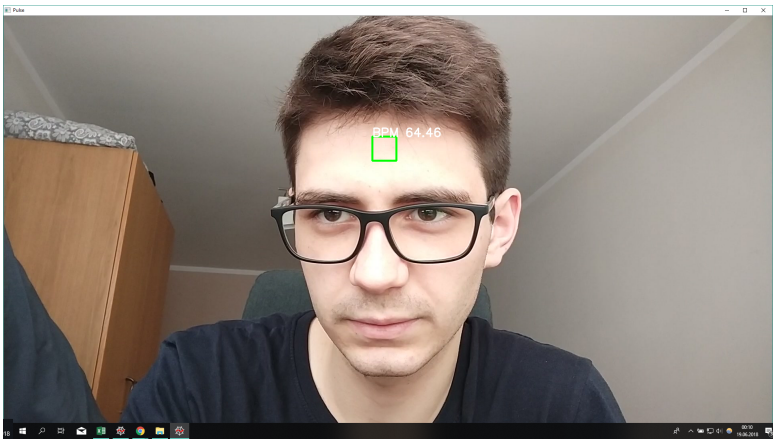
Następnie wybrano interesujące nas pasmo, dla którego szukano maksimum energii wyliczanej z transformaty Fouriera (czyli maksymalnej amplitudy). Pasmo ustaliliśmy w przedziale 50-180 BPM.

Częstotliwość odpowiadająca maksymalnej amplitudzie w danym paśmie jest wynikową wartością BPS.

Dane zwracane przez algorytm oraz te, które pochodzą z czujnika referecyjnego wyrysowywane są w czasie rzeczywistym oraz zapisywane do pliku *.txt w celu logowania postępów działania.

4. Działanie aplikacji.

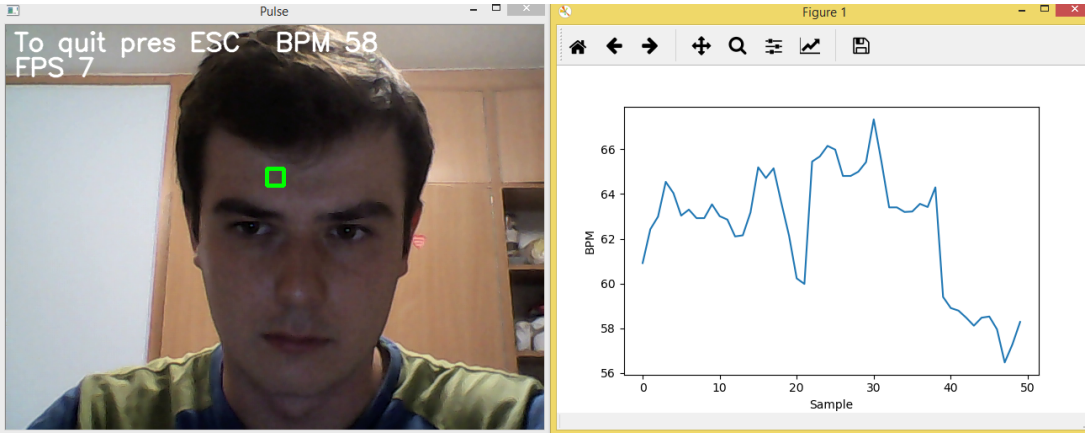
Wszystkie pomiary przeprowadzono w oświetleniu zarówno słonecznym jak i sztucznym. Pomiary zbierano około minuty. W tym czasie aplikacja na komputerze PC logowała dane w postaci obliczonej wartości pulsu przy pomocy algorytmu opisanego w rozdziale 3.2. Przeprowadzono badania na dwóch osobach. Dla każdej z osób zbierano pomiary, gdy ta od dłuższego czasu nie wykonywała pracy fizycznej, więc praca jej serca była spowolniona. Zebrano również pomiary tuż po wzmożonym wysiłku fizycznym lub w spoczynku, gdzie akcja serca jest spokojniejsza i wolniejsza.



Rys. 4.1. Wersja z czytaniem ramek z zapisanego filmu wideo.

Interfejsy aplikacji, która wczytuje serię ramek z zapisanego wideo, jak i tej otwierającej połączenie z kamerką internetową przedstawione są odpowiednio na rysunkach 4.1 i 4.2. W wypadku aplikacji 4.1 na obrazie zaznaczone jest ROI oraz aktualna wartość BPM, aplikacja 4.2 oprócz tego wyrysowuje zmianę wartości w czasie.

Wyniki otrzymane z czytania sekwencji wideo wydają się być obiecujące. W załączonej płycie CD, w folderze data/dane_camera znajdują się wyniki wraz z filmami wideo. Wyniki zbierane były dla dwóch osób, które w pierwszej sekwencji były niezmęczone, a następnie po wykonaniu ćwiczeń. Wyniki predykcji na podstawie algorytmu z kamery pokrywały się z tymi pochodzącymi od czujnika. Puls był odchylony stosunkowo o wartość około 5BPM.



Rys. 4.2. Wersja z odczytem danych z kamerki internetowej

W wypadku danych pochodzących i logowanych zarówno z kamery jak i czujnika w tym samym procesie niestety dość mocno obniżona została jakość działania algorytmu. Dlatego o samej aplikacji jedynie wspomiamy, chociaż nie dawała ona oczekiwanych wynikików. Jest to głównie spowodowane obciążeniem, jakim jest jednoczesne czytanie z dwóch zródeł, działanie algorytmu (z dodatkową interpolacją wartości zgubionych ramek) oraz wyrysowywanie w czasie rzeczywistym wyników. W szczególności rysowanie zajmowało procesorowi dość długi czas, co znacznie obniżało liczbę FPS.

Obydwie wersje aplikacji oraz wyniki (w folderze data) znajdują się w dołączonej płycie CD.

5. Wnioski.

Zaproponowany i wdrożony algorytm detekcji pulsu poprzez analizę filmu wideo okazał się pracować poprawnie tylko dla niewielkiego zbioru danych. Podczas przeprowadzanych eksperymentów testowano różne warunki oświetlenia i okazało się, że algorytm jest bardzo niestabilny, a na jego wynik ogromny wpływ ma ilość ramek w buforze. Dla kilku zebranych danych udało się zebrać zadowalające wyniki. Puls obliczany na podstawie analizy obrazu nadążał za pulsem wskazywanym przez licznik.

Algorytm potrzebuje chwili, aby ustalić puls. Jest to spowodowane między innymi czasem związanym z wypełnieniem bufora próbek. Im dłuższy bufor, tym dokładniejszy jest bufor i teoretycznie lepszy pomiar.

Opracowano dwie wersje programu, jedna odczytująca sekwencję ramek z filmu video i drugą odczytującą ramki z kamerki internetowej. Szybkość odczytu ramek z kamerki był na poziomie 7-12 FPS co okazało się jedną z przyczyn, dla których algorytm nie obliczał poprawnie pulsu. Szybkość pracy algorytmu można by zwiększyć przez przeniesienie jednej z jego części na osobny rdzeń (np. rysowanie danych, zapisywanie do pliku i czytanie danych z czujnika na jednym rdzeniu, a cały algorytm wraz z połączeniem z kamerą na drugim). Kolejnym rozwinięciem projektu z pewnością byłaby implementacja algorytmu detekcji twarzy i fragmentaryzacji dużej przestrzeni skórnej (czyli czoła). Można również pokusić się o możliwość detekcji pulsu dla więcej niż jednej osoby jednocześnie.

Bibliografia.

1. Przybyło J., Jabłoński M., Kańtoch E., Augustyniak P., Distant Pulse Measurement System for Real-Time Surveillance Applications, AGH, Kraków, Poland, 2017

2. Heart Rate Sensor SKU: SEN0203, https://www.dfrobot.com/wiki/index.php/Heart_Rate_Sensor_SK

3. Rahman H., Ahmed M.H, Begum S., Funk P., Real Time Heart Rate Monitoring From Facial RGB Color Video Using Webcam, Swedish Artificial Intelligence Society, Malmo, Sweden, 2016.

4. Bai G., Huang J., Real-time Robust Noncontact Heart Rate Monitoring with a Camera, doi: 10.1109/ACCESS.2018.2837086.

5. Verkruysse W., Svaasand L., Remote plethysmographic imaging using ambient light, Optical Society of America, 2008, DOI:https://doi.org/10.1364/OE.16.021434